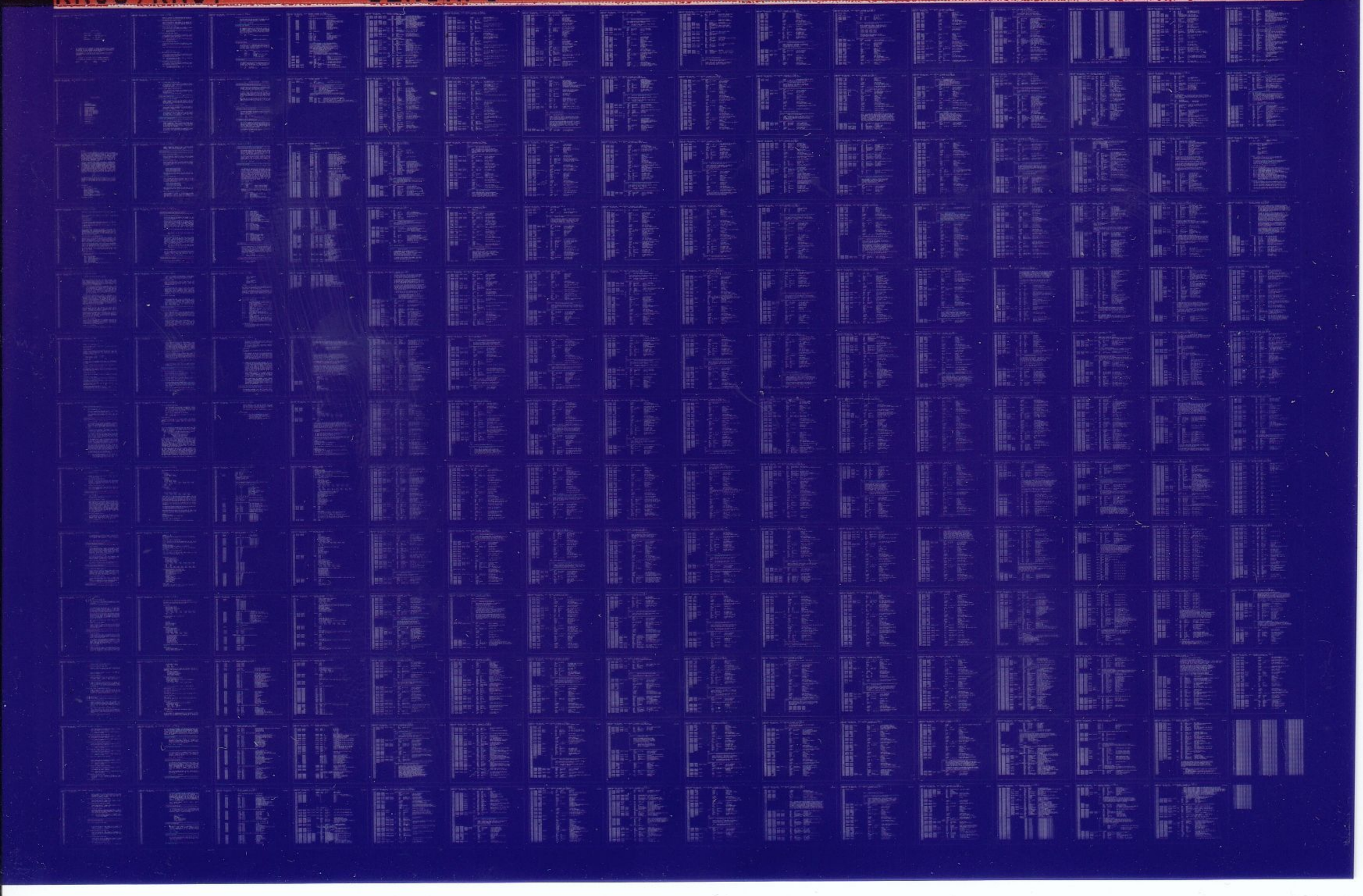
RK611
RK06,RK07

RK6 FCTNL CTRL
CZR6KF0

AH-9130F-MC
FICHE 1 OF 2



RK611 RK6 FCTNL CTRL AH-9130F-MC NOV 1980
RK06,RK07 CZR6KF0 FICHE 2 OF 2

```
.REM    %
```

IDENTIFICATION

PRODUCT CODE: AC-9128F-MC

PRODUCT NAME: CZR6KF0 RK6 FCTNL CTRL

DATE: JUNE 1980

MAINTAINER: DIAGNOSTIC GROUP

TABLE OF CONTENTS

1.0 ABSTRACT

THE RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC (CZR6K) IS A SERIES OF TESTS THAT COMPLETES THE TESTING OF AN RK611/RK06-RK07 SUBSYSTEM. THE DISKLESS CONTROLLER DIAGNOSTIC AND THE RK06-RK07 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-RK07 DRIVES
FORMATTED RK06K ON EACH DRIVE

2.2 PRELIMINARY PROGRAMS

THE RK611 DISKLESS CONTROLLER DIAGNOSTIC (ALL PARTS) AND THE

UNIBUS RK06-RK07 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 PROCESSCR IS HALTED.

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

3.3 CONSOLE SWITCH REGISTER

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

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

SW<7-0> - EXECUTE THIS TEST IF SW08 SET.

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

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

3.4 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E., AN 11/04 OR 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RK06 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

SWR = NNNNNN NEW ='

EACH TIME SWITCH SETTINGS ARE ENTERED, THE ENTIRE SWITCH REGISTER IMAGE MUST BE ENTERED. LEADING ZEROES ARE NOT REQUIRED. 'RUBOUT' AND 'CONTROL U' FUNCTIONS MAY BE USED TO CORRECT TYPING ERRORS DURING SWITCH ENTRY.

ON PROCESSORS WITH HARDWARE SWITCH REGISTERS, THE 'SOFTWARE' SWITCH REGISTER MAY BE USED. IF THE PROGRAM FINDS ALL 16 SWITCHES IN THE 'UP' POSITION, ALL SWITCH REGISTER REFERENCES WILL BE TO THE 'SOFTWARE' REGISTER AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.

3.5 CONTROL C (^C) OPERATION

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

3.6 CONTROL S (^S) OPERATION

IF ^S IS TYPED AT ANY TIME THE PROGRAM WILL GO INTO A STALL

LOOP UNTIL A CONTROL Q (^Q) IS TYPED.

3.7 CONTROL Q (^Q) OPERATION

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

3.8 UNIBUS ADDRESSES

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

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

3.9 EXECUTION TIME

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

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

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

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

4.0 PROGRAM DESCRIPTION

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

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

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

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

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

****

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

                        TEST 1  RK611 BASE ADDRESS TEST

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

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

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

                        TEST 3  SELECT ALL DRIVES

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

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

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

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

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

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

                                NOTE:  THIS TEST MUST BE RUN AT LEAST ONCE BEFORE  ANY
                                       OTHER TEST THAT FOLLOWS.
                                        THE DETERMINATION OF WHETHER EACH DRIVE IS AN
                                        RK06 OR RK07 IS MADE IN THIS TEST, AND
```

```
                                                                I  1
                                        A TABLE IS FILLED ACCORDINGLY.

                                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
        THIS TEST IS BYPASSED BY THE RK07 CONTROLLER BECAUSE
        IT DOES NOT RECOGNIZE ANY ILLEGAL DISK ADDRESS.

;*****************************************************************
;**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.
                                CYLINDER 1456 IS USED FOR THE RK07.



                                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



                        DATA TO CYLINDER 0, TRACK 0, SECTOR 0.  MAKE SURE ONLY
                        HEADER VRC ERROR SETS.

                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.

                TEST 26 OPI WITH HVRC ERROR

                        FORMAT CYLINDER 312, TRACK 0 SUCH THAT  A  HEADER  VRC
                        ERROR  IS  PRESENT AND SECTOR 17 HAS THE WRONG FORMAT.
                        ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK
                        0,  SECTOR  17.   THAT  BOTH  OPERATION INCOMPLETE AND
                        HEADER VRC SET.

                TEST 27 HVRC IGNORE ON NON-ADDRESSED SECTOR

                        FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR  20  HAS
                        AN  HVRC  ERROR.   ISSUE  A WRITE DATA OF 400 WORDS TO
                        CYLINDER 312, TRACK 0, AND SECTOR 21.  MAKE SURE  HVRC
                        IS NOT SET AT THE END OF THE OPERATION.



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

```
655
656                         WRITE CYLINDER 312, TRACK 0, SECTOR ZERO WITH A  KNOWN
657                         CONFIGURATION.   ISSUE  A  READ  DATA  OF 103 WORDS TO
658                         CYLINDER 312, TRACK 0, SECTOR 0.  MAKE SURE  ONLY  103
659                         WORDS GET TRANSFERRED TO MEMORY.
660
661                 TEST 42 WRITE DATA WITH NON-EXISTENT MEMORY
662
663                         ISSUE A WRITE DATA OF 1  WORD  USING  ADDRESS  776000.
664                         MAKE SURE NON-EXISTENT MEMORY SETS.
665
666                 TEST 43 READ DATA WITH NON-EXISTENT MEMORY
667
668                         ISSUE A READ DATA OF  1  WORD  USING  ADDRESS  776000.
669                         MAKE SURE NON-EXISTENT MEMORY SETS.
670
671                 TEST 44 UNIBUS PARITY ERROR
672
673                         INITIALIZE A MEMORY LOCATION WITH BAD PARITY.  ISSUE A
674                         WRITE  DATA  OF  400  WORDS STARTING AT A LOCATION 112
675                         WORDS BEFORE THE LOCATION WITH BAD PARITY.  MAKE  SURE
676                         THAT UNIBUS PARITY ERROR SETS.
677
678                         NOTE:  THIS TEST IS ONLY  EXECUTED  IF  MEMORY  PARITY
679                                OPTION EXISTS FOR BUFFER. IT WILL NOT BE EXECUTED
680                                 ON 'ECC' TYPE MEMORY.
681
682
683                 ********************************************************************
684                 **MULTIPLE SECTOR OPERATIONS
685                 ********************************************************************
686
687                 TEST 45 TWO SECTOR WRITE DATA (PART 1)
688
689
690
691                         ISSUE A WRITE DATA OF  1000  WORDS  TO  CYLINDER  312,
692                         TRACK  0,  SECTOR  0.   READ DATA BACK ONE SECTOR AT A
693                         TIME AND MAKE SURE IT IS CORRECT.
694
695                 TEST 46 TWO SECTOR WRITE DATA (PART 2)
696
697                         ISSUE A WRITE DATA OF  1000  WORDS  TO  CYLINDER  312,
698                         TRACK  0,  SECTOR  23.  READ DATA BACK ONE SECTOR AT A
699                         TIME AND MAKE SURE A MID-TRANSFER SEEK  DID  NOT  TAKE
700                         PLACE.
701
702                 TEST 47 TWO SECTOR WRITE DATA (PART 3)
703
704                         ISSUE A WRITE DATA OF 401 WORDS TO CYLINDER 312, TRACK
705                         0, SECTOR 10.  READ DATA BACK ONE SECTOR AT A TIME AND
706                         CHECK ZERO FILL OF SECOND SECTOR.
707
708                 TEST 50 MID-TRANSFER SEEK ON WRITE (PART 1)
709
710                         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.
                                CYLINDER 1456 IS USED FOR THE RK07.

                        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.
                                CYLINDER 1456 IS USED FOR THE RK07.

                        ********************************************************************
                        **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)

                                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 64 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 65 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 66 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 67 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 100C00

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

                TEST 70 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 71 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 72 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 73 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.
                                        CYLINDER 1456 IS USED FOR THE RK07.

                        TEST 74 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.
                                CYLINDER 1456 IS USED FOR THE RK07.

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

                        TEST 75 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 76 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 77 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 100 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 101 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 102 DRIVE OFF TRACK

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

```
                        TEST 103 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 104 DUMMY TEST FOR PREVIOUS TEST EXIT

                                THIS TEST IS PRESENT TO MAKE  $SW08TB  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 105 RESET ATTENTIONS WITH UNIBUS INIT

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

                        TEST 106 RESET ATTENTIONS WITH SUBSYSTEM CLEAR

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

                        TEST 107 SVAL AND ATTENTION FROM OTHER DRIVE

                                DO A RECALIBRATE ON ONE AVAILABLE DRIVE.  DO A  SELECT
                                ON ANOTHER AVAILABLF 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 110 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:



                        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
                        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 WORDS 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 MANNER 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
```

```
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
1124
1125
1126
1127                                        $TESTN  $ERRPC  DRVNUM
1128                                          RKCS1   RKCS2   RKDS    RKER    RKASOF  RKMR1
1129                                          T.CS1   T.CS2   T.DS    T.ER    T.ASF   T.MR1
1130
1131                                    ERROR ITEMS 4,5,6, AND 7 ARE USED TO REPORT  ERRORS  THAT  ARE
1132                                    THE  RESULT  OF  A HARDWARE ERROR INDICATOR BEING SET WHEN NOT
1133                                    EXPECTED, NOT SET WHEN IT IS EXPECTED,  OR  BOTH.   THE  ERROR
1134                                    REPORT WILL CONTAIN (1) ALL THE ERRORS THAT WERE DETECTED, (2)
1135                                    ALL THE EXPECTED ERRORS THAT DID NOT OCCUR,  OR  (3)  ALL  THE
1136                                    EXPEDTED BUT NOT SET ERRORS AND THE UNEXPECTED BUT SET ERRORS.
1137
1138                                    THE MESSAGE ITSELF EXPLAINS THE CIRCUMSTANCE FOR  THE  REPORT.
1139                                    INCLUDED  IN  THE  REPORT WILL BE ONE OR MORE OF THE FOLLOWING
1140                                    STATEMENTS:
1141
1142                                    "THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:"
1143                                    "THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:"
1144                                    "THE ABOVE ARE ERRORS SET IN OPERATION:"
1145
1146                                    PRECEEDING ANY ONE OF THESE LINES WILL BE ONE  OR  MORE  LINES
1147                                    THAT SPECIFY THE EXACT ERROR.  FOLLOWING THE LAST LINE WILL BE
1148                                    A LINE THAT IDENTIFIES THE OPERATION BEING PERFORMED.
1149
1150                                    EXAMPLE:
1151                                    NON-EXISTANT DRIVE
1152                                    THE ABOVE ARE ERRORS SET IN OPERATION:
1153                                    DRIVE SELECT
1154
1155                                    (ADDITIONAL LINES OF INFORMATION)
1156
1157                                    THIS IS THE RESULT OF AN ERROR SET IN A SELECT OPERATION.
1158
```

```
1159                                    EXAMPLE:
1160                                    NON-EXISTANT DRIVE
1161                                    THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:
1162                                    DRIVE SELECT
1163
1164                                    (ADDITIONAL LINES OF INFORMATION)
1165
1166                                    THIS IS THE RESULT OF AN EXPECTED ERROR THAT  DID  NOT  OCCUR,
1167                                    I.E.  A NON-EXISTANT DRIVE WAS ADDRESSED BUT NED WAS NOT SET.
1168
1169                                    EXAMPLE:
1170                                    NON-EXISTANT MEMORY
1171                                    THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:
1172                                    UNIBUS PARITY ERROR
1173                                    THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:
1174                                    WRITE DATA
1175
1176                                    (ADDITIONAL LINES OF INFORMATION)
1177
1178                                    THIS IS AN EXAMPLE  OF  NON-EXISTANT  MEMORY  BEING  SET  WHEN
1179                                    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)
```

```
                                        TST NUM ERR PC  DRIVE
                                        $TESTN  ERRPC   DRVNUM
                                          EXPT    IS
                                          $REG10  $REG11

                                    THE ERRORS REPORTED BY THIS FORMAT ARE  SOFTWARE  DETECTED  BY
                                    COMPARING  EXPECTED RESULTS WITH ACTUAL RESULTS.  THE SPECIFIC
                                    ERRORS ARE:
                                          WORD COUNT INCORRECT
                                          BUS ADDRESS INCORRECT
                                          CYLINDER ADDRESS INCORRECT
                                          TRACK ADDRESS INCORRECT
                                          SECTOR ADDRESS INCORRECT

                                    ERROR ITEM 11
                                        (ERROR INDICATOR OR STATUS BIT)
                                        NOT SET AS A RESULT OF
                                        (ANOTHER ERROR INDICATOR, STATUS BIT, OR OPERATION
                                        TST NUM ERR PC  DRIVE
                                        $TESTN  $ERRPC  DRVNUM
                                          RKCS1   RKCS2   RKDS    RKER    RKASOF  RKMR1



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

                                    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
```

```
1327                    OF  THE  FUNCTION  PERFORMED.  THE SUBROUTINE FUNCTION IS KEPT
1328                    SMALL AND IN GENERAL A SUBROUTINE ONLY PERFORMS ONE  FUNCTION,
1329                    I.E., LOAD THE RK611 REGISTER AND START AN OPERATION (TLOADRK)
1330                    OR WAIT A SPECIFIED NUMBER OF MILLISECONDS  FOR  AN  INTERRUPT
1331                    (TWATNN  WHERE  NN VARIES FROM CALL TO CALL AND IS THE TIME TO
1332                    WAIT).  THE FOLLOWING IS A DESCRIPTION OF THE SUBROUTINES  NOT
1333                    PROVIDED BY SYSMAC:
1334
1335                    ;*******************************************************************
1336                    ;LINE CLOCK CALIBRATE
1337                    ;*******************************************************************
1338
1339                            WAITS FOR A LINE  CLOCK  INTERRUPT  TO  CALIBRATE  THE
1340                            INTERRUPTS TO A MEANINGFUL TIME VALUE.  IN ADDITION IT
1341                            PRESETS THE TIMCNT IF THERE IS NO LINE CLOCK.   TIMCNT
1342                            IS USED IN THE LINE CLOCK SIMULATOR.
1343
1344                            CALL:   JSR     PC,CLKCAL
1345
1346                    ;*******************************************************************
1347                    ;OPTION PRESENT TEST AND SETUP
1348
1349
1350
1351                    ;*******************************************************************
1352
1353                            THIS ROUTINE CHECKS IF THE MEMORU  PARITY  OPTION  AND
1354                            THE  LINE  CLOCK  ARE ON THE SYSTEM.  FLAGS ARE SET IF
1355                            PRESENT;   CLEARED  OTHERWISE,  AND  THE   APPROPRIATE
1356                            INTERRUPT VECTORS ARE SET UP.
1357
1358                            CALL:   JSR     PC,OPTTST
1359
1360                    ;*******************************************************************
1361                    ;LINE CLOCK SIMULATION ROUTINE
1362                    ;*******************************************************************
1363
1364                            THIS ROUTINE IS USED TO SIMULATE THE LINE  CLOCK.   TO
1365                            DO THIS THE VALUE STORED IN MILCNT IS USED AS THE BASE
1366                            AND REPRESENTS THE NUMBER OF  TIMES  A  DECREMENT  AND
1367                            BRANCH  LOOP CAN BE DONE IN 1 MILLISECOND.  THE TIMCNT
1368                            VALUE IS DECREMENTED AND IF  IT  REACHED  0  THE  LINE
1369                            CLOCK  TICK  COUNTER  IS  BUMPED.   THEN THE TIMCNT IS
1370                            RESET TO 16 (REPRESENTS 16 MILLISECONDS PER LINE CLOCK
1371                            TICK).
1372
1373                            THUS  THE  ROUTINE  RETURNS  TO  THE  CALLER  AFTER  1
1374                            MILLISECOND  AND BUMPS THE LINE CLOCK TICK COUNTER FOR
1375                            EACH 16 CALLS.
1376
1377                            CALL:   JSR     PC,MYTIME
1378
1379                    ;*******************************************************************
1380                    ;WAIT FOR INTERRUPT ROUTINE
1381                    ;*******************************************************************
1382
```

```
1383                                    THE ROUTINE IS ENTERED BY ONE OF FOURTEEN TRAP  CALLS.
1384                                    THE  CALL  SPECIFIES  HOW MANY TICKS OF THE LINE CLOCK
1385                                    ARE  TO  ELAPSE  WHILE  WAITING  FOR  INTERRUPT.    IF
1386                                    INTERRUPT  DOES  NOT  OCCUR IN THAT PERIOD OF TIME. AN
1387                                    ERROR MESSAGE IS PREPARED  (BUT  NOT  PRINTED  IN  THE
1388                                    ROUTINE)AND THEN RETURNS TO THE LOCATION FOLLOWING THE
1389                                    CALL.  IF INTERRUPT OCCURS THE RETURN IS BUMPED BY  2.
1390                                    NORMALLY  AN  ERROR CALL WILL BE IN THE LOCATION AFTER
1391                                    THE CALL TO INTERRUPT WAIT.
1392
1393                                    CALL:   TWAT16  THROUGH   TWAT159,   TWAT1S,   TWAT2S,
1394                                    TWAT8S, AND TWAT1M
1395
1396                                ;*******************************************************************
1397                                ;"L" REGISTER LOADING ROUTINE
1398                                ;*******************************************************************
1399
1400                                    THE PARAMETERS FOLLOWING THE CALL ARE TRANSFERRED INTO
1401                                    THE  "L"  REGISTERS L.CS1-L.DCYL.  L.MR1 IS NOT LOADED
1402                                    IN THIS MANNER SINCE IT IS NOT COMMONLY LOADED FOR  AN
1403                                    OPERATION.  L.CS2 IS LOADED FROM DRVNUM.
1404
1405
1406
1407                                    CALL:   JSR     R4,LRLOAD
1408                                            COMMAND
1409                                            WORD COUNT
1410                                            BUS ADDRESS
1411                                            .BYTE   SECTOR ADDRESS
1412                                            .BYTE   TRACK ADDRESS
1413                                            CYLINDER ADDRESS
1414
1415                                ;*******************************************************************
1416                                ;LOAD RK611 FOR OPERATION
1417                                ;*******************************************************************
1418
1419                                    THE REGISTER SETUP STORAGE IS TRANSFERRED TO THE RK611
1420                                    REGISTER.    THIS  IS  A  STRAIGHT  TRANSFER  WITH  NO
1421                                    CHECKING OR MANIPULATION  OF  THE  REGISTER  CONTENTS.
1422                                    L.CS1  IS  TRANSFERRED  LAST AS IT SHOULD BE IF THE GO
1423                                    BIT IS SET.
1424
1425                                    CALL:   TLOADRK
1426
1427                                ;*******************************************************************
1428                                ;STORE RK611 REGISTERS
1429                                ;*******************************************************************
1430
1431                                    ALL THE RK611 REGISTERS ARE STORED IN THE TEST TABLE T
1432                                    WITH  THE  EXCEPTION  OF  THE DATA BUFFER WHICH IS NOT
1433                                    STORED IN THIS ROUTINE.
1434
1435                                    CALL:   TGETRK
1436
1437                                ;*******************************************************************
1438                                ;BIT COUNTER IN A WORD
```

```
1439                    ;*****************************************************************
1440                    ;
1441                    ;       THE WORD WHOSE BITS MUST BE COUNTED IS PLACED  ON  THE
1442                    ;       STACK  BY  THE  CALLING  ROUTINE.   THE NUMBER OF BITS
1443                    ;       FOUND IN THE WORD ARE PASSED BACK ON THE STACK.
1444                    ;
1445                    ;       CALL:   JSR     R4,BITCNT
1446                    ;
1447                    ;*****************************************************************
1448                    ;MAINTENANCE CLOCK ROUTINE
1449                    ;*****************************************************************
1450                    ;
1451                    ;       THE PARAMETERS PASSED TO THIS ROUTINE ARE  LOCATED  IN
1452                    ;       THE  ADDRESS  AFTER THE CALL.  THE FIRST BYTE CANTAINS
1453                    ;       THE NUMBER OF PHASES  ADDRESSES  THE  CALLING  ROUTINE
1454                    ;       WANTS  THE  CONTROLLER  CLOCKED THROUGH AND THE SECOND
1455                    ;       BYTE CONTAINS THE NUMBER OF CLOCK TRANSITIONS (PARTIAL
1456                    ;       PHASES) TO BE DONE.
1457                    ;
1458                    ;       CALL:   JSR     R4,MCLOCK
1459                    ;               .BYTE           ;NUMBER OF CLOCK TRANSISTIONS
1460                    ;
1461                    ;
1462                    ;
1463                    ;               .BYTE           ;NUMBER OF PHASE ADDRESSES
1464                    ;
1465                    ;*****************************************************************
1466                    ;READ AND SORT HEADERS
1467                    ;*****************************************************************
1468                    ;
1469                    ;       THE HEADERS IN THE CYLINDER AND TRACK SPECIFIED BY THE
1470                    ;       FIELDS  IN  THE  "L"  REGISTERS ARE READ AND STORED IN
1471                    ;       ASSCENDING ORDER.  CONTROLLER ERRORS  ARE  CHECKED  IN
1472                    ;       THE  READ  HEADER  OPERATION  AND DATA LATE IS CHECKED
1473                    ;       AFTER EACH READ OF THE DATA BUFFER.
1474                    ;
1475                    ;       CALL:   JSR     R4,RDSTHD
1476                    ;               TCHKOP          ;RETURN POINT IF CERR IN READ
1477                    ;                               ;HDR
1478                    ;               ERROR   4       ;OR 5, 6, 7
1479                    ;               ERROR   13      ;RETURN IF DATA LATE IN DB
1480                    ;                               ;UNLOAD
1481                    ;               ERROR   2       ;RETURN IF TO SLOW OR
1482                    ;                               ;IF HDR 0 NOT FOUND
1483                    ;
1484                    ;*****************************************************************
1485                    ;GET DRIVE STATUS
1486                    ;*****************************************************************
1487                    ;
1488                    ;       THIS ROUTINE GETS ALL THE DRIVE STATUS AND  PLACES  IT
1489                    ;       IN  $REG10  THROUGH $REG17.  THESE REGISTORS ARE FIRST
1490                    ;       CLEARED TO ALL ONES AND THEN  IF  ERROR  OCCURS  WHILE
1491                    ;       GETTING STATUS, THE 1'S ARE LEFT IN THE REGISTERS.
1492                    ;
1493                    ;       CALL:   JSR     R4,GETDRS
1494                    ;               BR      ERROR PROCESSING        ERROR RETURN
```

```
1495                                    BR      NO ERROR PROCESSING     GOOD RETURN
1496
1497                            ;*****************************************************************
1498                            ;SUBSYSTEM INITIALIZE AND INITIALIZE STATE TEST
1499                            ;*****************************************************************
1500
1501                                    THE SUBSYSTEM IS INITIALIZED WITH A SUBSYSTEM CLEAR
1502                                    COMMAND.  CERR AND DI ARE MONITORED FOR A SHORT PEIROD
1503                                    OF TIME DURING WHICH THEY SHOULD BOTH RESET.
1504
1505                                    IF THEY DO RESET, READY IS TESTED TO INSURE IF SETS.
1506
1507                                    IF ANY OF THESE THREE CONDITIONS ARE NOT MET AN
1508                                    APPROPRIATE ERROR MESSAGE IS PREPARED AND REPORTED
1509                                    WHEN THE ROUTINE RETURN TO THE CALL.  IF EVERY THING
1510                                    IS GOOD, THE RETURN SKIPS OVER THE ERROR CALL AND TEST
1511                                    ABORT.
1512
1513                                    THE USUAL CALL TO THIS ROUTINE WILL BE FOLLOWED BY AN
1514                                    ERROR MESSAGE AND BRANCH TO END OF TEST.  THIS IS DONE
1515                                    BECAUSE FAILURE TO INITIALIZE CORRECTLY IS FATAL TO
1516
1517
1518
1519                                    THE TEST.
1520
1521                                    CALL:   TSSINIT
1522
1523                            ;*****************************************************************
1524                            ;WORD COUNT AT END OF OPERATION CHECK
1525                            ;*****************************************************************
1526
1527                                    THIS ROUTINE COMPARES THE CONTENTS OF THE TEST STORAGE
1528                                    FOR THE WORD COUNT AGAINST THE SUPPLIED VALUE.  IF
1529                                    UNEQUAL, THE ERROR FLAG (WCERR) IS SET IN GROUP 4
1530                                    ERROR FLAGS (GRP4ER)
1531
1532                                    CALL:   JSR     R4,CHKWC
1533                                            .WORD           ;EXPECTED WC VALUE
1534
1535                            ;*****************************************************************
1536                            ;BUS ADDRESS AT END OF OPERATION CHECK
1537                            ;*****************************************************************
1538
1539                                    THIS ROUTINE COMPUTES THE EXPECTED BUS ADDRESS AT THE
1540                                    END OF A TRANSFER BY USING THE INITIAL BUS ADDRESS,
1541                                    ADDING IN THE INITIAL WORD COUNT, AND SUBTRACTING ANY
1542                                    RESIDUAL WORD COUNT.  IF THIS COMPUTED BA DOES NOT
1543                                    EQUAL THE CONTENTS OF RKBA AN ERROR FLAG (BAERR) IS
1544                                    SET IN GROUP 4 ERROR FIELD (GRP4ER)
1545
1546                                    IF BUS ADDRESS INCREMENT INHIBIT WAS SET, THE EXPECTED
1547                                    BUS ADDRESS IS THE STARTING BUS ADDRESS.
1548
1549                                    CALL:   JSR     R4,CHKBA
1550
```

```
                        ;*****************************************************************
                        ;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 INITIALIZATION.

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).

B. COMPARE THE CONTENTS OF IBUFF AND OBUFF (BIT 15). THE CONTENTS OF THE BUFFER MAY BE HEADERS OR DATA. A HEADER COMPARE OPERATION MAY BE SPECIFIED (BIT 7) WHICH WILL CAUSE THE COMPARE TO BE LIMITED TO 74(8) OR 102(8) WORDS OF HEADERS. THE LENGTH DEPENDS ON THE FORMAT BIT THAT WAS LAST SPECIFIED IN L.CS1. THE HEADERS MAY BE BUILT BEFORE THE COMPARE AS PART OF THE OPERATION (BIT 15 AND BIT 8 OR 9). DATA CAN ALSO BE GENERATED BEFORE THE COMPARE (NON-ZERO BITS 6-0).

C. RESUME COMPARE OPERATION. IF A COMPARE OPERATION DETECTS A MISCOMPARE, THE ROUTINE RETURNS TO CALLER BUT STORES PARAMETERS SUCH THAT THE COMPARE CAN BE RESUMED. THIS IS DONE BY CALLING GENCOM WITH BIT 14 SET IN THE CONTROL WORD.

D. DATA GENERATION OR COMPARE USING MEMORY MANAGEMENT. MEMORY MANAGEMENT CAN BE INVOKED FOR EITHER SOURCE OR DESTINATION BUT NOT FOR BOTH. IN THIS MANNER, DATA GENERATION CAN BE MADE TO PLACE DATA ANYWHERE IN AVAILABLE MEMORY. LIKEWISE DATA COMPARE WILL COMPARE THE CONTENTS OF IBUFF TO ANY AREA OF AVAILABLE MEMORY.

****
PHASE LOCKED LOOP CLOCK ADJUSTMENT ROUTINE
****

THIS ROUTINE IS ENTERED VIA START LOCATION 220(8). THE PROGRAM FIRST RUNS TEST 1, 2, AND 3 TO SET UP THE 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
1783                            7.0     REVISION HISTORY
1784
1785                                    REVF0   1) ADD CODE IN PROGRAM INITTIALIZATION TO
1786                                               CLEAR DATA BUFFER MEMORY LOCATIONS
1787                                            2) MODIFY CHAIN MODE LOOP ADDRESS IN $EOP
1788                                               FOR XXDP+ COMPATIBILITY
1789                                            3) MODIFY 'OPTTST' ROUTINE TO SUPPORT NEW
1790                                               PROCESSOR AND MEMORY OPTIONS
1791                                    %
```

```
                        ;       *** REV 007 ***
                        .NLIST  CND,MD,MC,TOC
                        .LIST   ME
                        .ENABL  ABS,AMA
167400                  $SWR=   167400
000001                  $TN=    1
                        .TITLE  CZR6KFO RK6 FCTNL CTRL DIAG
                        ;*COPYRIGHT (C) 1976,1980
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*
                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        12            ABORT PROGRAM AFTER 20 ERRORS
                        ;*        11            INHIBIT ITERATIONS
                        ;*        10            BELL ON ERROR
                        ;*         9            LOOP ON ERROR
                        ;*         8            LOOP ON TEST IN SWR<7:0>
                        .SBTTL  BASIC DEFINITIONS

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

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

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

```
1848    000006                  SP=     %6              ;;STACK POINTER
1849    000007                  PC=     %7              ;;PROGRAM COUNTER
1850
1851                            ;*PRIORITY LEVEL DEFINITIONS
1852    000000                  PR0=    0               ;;PRIORITY LEVEL 0
1853    000040                  PR1=    40              ;;PRIORITY LEVEL 1
1854    000100                  PR2=    100             ;;PRIORITY LEVEL 2
1855    000140                  PR3=    140             ;;PRIORITY LEVEL 3
1856    000200                  PR4=    200             ;;PRIORITY LEVEL 4
1857    000240                  PR5=    240             ;;PRIORITY LEVEL 5
1858    000300                  PR6=    300             ;;PRIORITY LEVEL 6
1859    000340                  PR7=    340             ;;PRIORITY LEVEL 7
1860
1861                            ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
1862    100000                  SW15=   100000
1863    040000                  SW14=   40000
1864    020000                  SW13=   20000
1865    010000                  SW12=   10000
1866    004000                  SW11=   4000
1867    002000                  SW10=   2000
1868    001000                  SW09=   1000
1869    000400                  SW08=   400
1870    000200                  SW07=   200
1871    000100                  SW06=   100
1872    000040                  SW05=   40
1873    000020                  SW04=   20
1874    000010                  SW03=   10
1875    000004                  SW02=   4
1876    000002                  SW01=   2
1877    000001                  SW00=   1
1878                            .EQUIV  SW09,SW9
1879                            .EQUIV  SW08,SW8
1880                            .EQUIV  SW07,SW7
1881                            .EQUIV  SW06,SW6
1882                            .EQUIV  SW05,SW5
1883                            .EQUIV  SW04,SW4
1884                            .EQUIV  SW03,SW3
1885                            .EQUIV  SW02,SW2
1886                            .EQUIV  SW01,SW1
1887                            .EQUIV  SW00,SW0
1888
1889                            ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
1890    100000                  BIT15=  100000
1891    040000                  BIT14=  40000
1892    020000                  BIT13=  20000
1893    010000                  BIT12=  10000
1894    004000                  BIT11=  4000
1895    002000                  BIT10=  2000
1896    001000                  BIT09=  1000
1897    000400                  BIT08=  400
1898    000200                  BIT07=  200
1899    000100                  BIT06=  100
1900    000040                  BIT05=  40
1901    000020                  BIT04=  20
1902    000010                  BIT03=  10
1903    000004                  BIT02=  4
```

```
1904            000002                  BIT01=  2
1905            000001                  BIT00=  1
1906                                    .EQUIV  BIT09,BIT9
1907                                    .EQUIV  BIT08,BIT8
1908                                    .EQUIV  BIT07,BIT7
1909                                    .EQUIV  BIT06,BIT6
1910                                    .EQUIV  BIT05,BIT5
1911                                    .EQUIV  BIT04,BIT4
1912                                    .EQUIV  BIT03,BIT3
1913                                    .EQUIV  BIT02,BIT2
1914                                    .EQUIV  BIT01,BIT1
1915                                    .EQUIV  BIT00,BIT0
1916
1917                                    ;*BASIC "CPU" TRAP VECTOR ADDRESSES
1918            000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
1919            000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
1920            000014                  TBITVEC=14              ;;"T" BIT
1921            000014                  TRTVEC= 14              ;;TRACE TRAP
1922            000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
1923            000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
1924            000024                  PWRVEC= 24              ;;POWER FAIL
1925            000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
1926            000034                  TRAPVEC=34              ;;"TRAP" TRAP
1927            000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
1928            000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
1929            000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
1930                                    .SBTTL  MEMORY MANAGEMENT DEFINITIONS
1931
1932                                    ;*KT11 VECTOR ADDRESS
1933
1934            000250                  MMVEC=  250
1935
1936                                    ;*KT11 STATUS REGISTER ADDRESSES
1937
1938            177572                  SR0=    177572
1939            177574                  SR1=    177574
1940            177576                  SR2=    177576
1941            172516                  SR3=    172516
1942
1943                                    ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
1944
1945            172300                  KIPDR0= 172300
1946            172302                  KIPDR1= 172302
1947            172304                  KIPDR2= 172304
1948            172306                  KIPDR3= 172306
1949            172310                  KIPDR4= 172310
1950            172312                  KIPDR5= 172312
1951            172314                  KIPDR6= 172314
1952            172316                  KIPDR7= 172316
1953
1954                                    ;*KERNEL "I" PAGE ADDRESS REGISTERS
1955
1956            172340                  KIPAR0= 172340
1957            172342                  KIPAR1= 172342
1958            172344                  KIPAR2= 172344
1959            172346                  KIPAR3= 172346
```

```
1960            172350          KIPAR4= 172350
1961            172352          KIPAR5= 172352
1962            172354          KIPAR6= 172354
1963            172356          KIPAR7= 172356
1964
1965            000210          AVECT1= 210                     ;DEFINE RK611 VECTOR INTERRUPT
1966            000240          APRIOR= PR5                     ;DEFINE RK611 PRIORITY
1967            177440          ABASE=  177440                  ;DEFINE RK611 BASE BUS ADDRESS
1968
1969                            .SBTTL  RK611 CONTROLLER REGISTER DEFINITION
1970
1971            000000          RKCS1=  0                       ;CONTROL AND STATUS REGISTER 1
1972            000002          RKWC=   2                       ;WORD COUNT REGISTER
1973            000004          RKBA=   4                       ;BUS ADDRESS REGISTER
1974            000006          RKDA=   6                       ;DESIRED TRACK SECTOR REGISTER
1975            000010          RKCS2=  10                      ;CONTROL AND STATUS REGISTER 2
1976            000012          RKDS=   12                      ;DRIVE STATUS REGISTER
1977            000014          RKER=   14                      ;ERROR REGISTER
1978            000016          RKASOF= 16                      ;ATTENTION SUMMARY AND OFFSET REGISTER
1979            000020          RKDCYL= 20                      ;DESIRED CYLINDER REGISTER
1980            000024          RKDB=   24                      ;DATA BUFFER
1981            000026          RKMR1=  26                      ;MAINTENANCE REGISTER 1
1982            000034          RKMR2=  34                      ;MAINTENANCE REGISTER 2
1983            000036          RKMR3=  36                      ;MAINTENANCE REGISTER 3
1984            000030          RKECPS= 30                      ;ECC POSITION INFORMATION
1985            000032          RKECPT= 32                      ;ECC PATTERN INFORMATION
1986            000022          RKSPAR= 22                      ;SPARE REGISTER
1987
1988                            .SBTTL  DRIVE COMMANDS
1989
1990            000101          SELDRV= 101                     ;SELECT DRIVE
1991            000103          PACK=   103                     ;PACK ACKNOWLEDGE
1992            000105          CLEAR=  105                     ;DRIVE CLEAR
1993            000107          UNLOAD= 107                     ;UNLOAD
1994            000111          SRTSPL= 111                     ;START SPINDLE
1995            000113          RECAL=  113                     ;RECALIBRATE
1996            000115          OFFSET= 115                     ;OFFSET
1997            000117          SEEK=   117                     ;SEEK
1998            000121          RDDATA= 121                     ;READ DATA
1999            000123          WRDATA= 123                     ;WRITE DATA
2000            000125          RDHEAD= 125                     ;READ HEADER
2001            000127          WRHEAD= 127                     ;WRITE HEADER AND DATA
2002            000131          WRTCHK= 131                     ;WRITE CHECK
2003            000300          INTR=   300                     ;GENERATE INTERRUPT TO CPU
2004
2005                            .SBTTL  CONTROL AND STATUS REGISTER 1 BITS
2006
2007            000001          GO=     BIT0                    ;GO BIT
2008            000100          IE=     BIT6                    ;INTERRUPT ENABLE
2009            000200          RDY=    BIT7                    ;CONTROLLER READY
2010            000400          BA16=   BIT8                    ;BUS ADDRESS BIT 16
2011            001000          BA17=   BIT9                    ;BUS ADDRESS BIT 17
2012            002000          CDT=    BIT10                   ;CONTROLLER DRIVE TYPE (0=RK06)
2013            004000          CTO=    BIT11                   ;CONTROLLER TIMED OUT WAITING FOR
2014                                                            ;  DRIVE RESPONSE
2015            010000          CFMT=   BIT12                   ;CONTROLLER DRIVE FORMAT (0=26 SECTOR, 1-24 SECTOR)
```

```
                        SPAR=   BIT13           ;DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
                        DI=     BIT14           ;DRIVE INTERRUPT
                        CERR=   BIT15           ;CONTROLLER ERROR
                        CCLR=   BIT15           ;CONTROLLER CLEAR

                        .SBTTL  CONTROL AND STATUS REGISTER 2 BITS

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

                        .SBTTL  ERROR REGISTER BIT DEFINITION

                        ILF=    BIT0            ;ILLEGAL FUNCTION CODE
                        SKI=    BIT1            ;SEEK INCOMPLETE
                        NXF=    BIT2            ;NON-EXECUTABLE DRIVE FUNCTION
                        DRPAR=  BIT3            ;DRIVE DETECTED DRIVE BUS PARITY ERROR
                        FMTE=   BIT4            ;FORMAT ERROR
                        DTYE=   BIT5            ;DRIVE TYPE ERROR
                        ECH=    BIT6            ;ECC HARD
                        BSE=    BIT7            ;BAD SECTOR ERROR
                        HVRC=   BIT8            ;HEADER VRC ERRROR
                        COE=    BIT9            ;CYLINDER ADDRESS OVERFLOW ERROR
                        IDAE=   BIT10           ;INVALID DISK ADDRESS ERROR
                        WLE=    BIT11           ;WRITE LOCK ERROR
                        DTE=    BIT12           ;DRIVE TIMING ERROR
                        OPI=    BIT13           ;OPERATION (SEARCH) INCOMPLETE
                        UNS=    BIT14           ;DRIVE UNSAFE
                        DCK-    BIT15           ;DATA CHECK

                        .SBTTL  STATUS REGISTER BIT DEFINITION

                        DRA=    BIT0            ;DRIVE AVAILABLE (CONTROLLER IS SET IF
                                                ;  THIS BIT IS RESET)
                        OFST=   BIT2            ;DRIVE OFFSET
                        ACLO=   BIT3            ;AC LOW
                        SPDLSS= BIT4            ;SPEED LOSS
                        DROT=   BIT5            ;DRIVE OFF TRACK
                        VV=     BIT6            ;VOLUME VALID
                        DRDY=   BIT7            ;DRIVE READY
                        DDT=    BIT8            ;DRIVE TYPE (0=RK06)
                        WRL=    BIT11           ;WRITE LOCK
                        PIP=    BIT13           ;POSITIONING IN PROGRESS
                        DSC=    BIT14           ;DRIVE STATUS CHANGE
                        SVAL=   BIT15           ;STATUS VALID
```

```
                        .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION

000017                  MESMSK= 17                      ;MESSAGE MASK

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

                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A

000040                  S.DRA=  BIT5                    ;DRIVE AVAILIABLE
000100                  S.VV=   BIT6                    ;VOLUME VALID
000200                  S.DRY=  BIT7                    ;DRIVE READY
000400                  S.TYPE= BIT8                    ;DRIVE TYPE
001000                  S.FORM= BIT9                    ;DRIVE FORMAT
002000                  S.OFF=  BIT10                   ;OFFSET
004000                  S.WRL=  BIT11                   ;WRITE LOCK
010000                  S.SPIN= BIT12                   ;SPINDLE ON
020000                  S.PIP=  BIT13                   ;POSITIONING IN PROGRESS
040000                  S.DSC=  BIT14                   ;DRIVE STATUS CHANGE

                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B

000040                  S.ICYL= BIT5                    ;ILLEGAL CYLINDER ADDRESS
000100                  S.ACLO= BIT6                    ;AC LOW
000200                  S.FLT=  BIT7                    ;DRIVE FAULT
000400                  S.ILF=  BIT8                    ;ILLEGAL FUNCTION
001000                  S.PAR=  BIT9                    ;DRIVE DETECTED SERCON PARITY ERROR
002000                  S.SKI=  BIT10                   ;SEEK INCOMPLETE
004000                  S.WLE=  BIT11                   ;WRITE LOCK ERROR
010000                  S.SPLS= BIT12                   ;SPEED LOSS
020000                  S.DROT= BIT13                   ;DRIVE OFF TRACK
040000                  S.UNS=  BIT14                   ;DRIVE UNSAFE

                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A

000020                  S.XDOK= BIT4                    ;TRANSDUCER OK
000040                  S.HDHM= BIT5                    ;HEADS HOME
000100                  S.BRHM= BIT6                    ;BRUSHES HOME
000200                  S.DOOR= BIT7                    ;DOOR INTERLOCKED
000400                  S.CART= BIT8                    ;CARTRAGE INTERLOCK
001000                  S.SPOK= BIT9                    ;SPEED OK
002000                  S.FWD=  BIT10                   ;FORWARD
004000                  S.REV=  BIT11                   ;REVERSE
010000                  S.LOAD= BIT12                   ;HEADS LOADING
020000                  S.RTZ=  BIT13                   ;RETURN TO ZERO
```

```
                        040000          S.UNLD= BIT14                   ;HEADS UNLOADING
                                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B
                        000020          S.SECT= BIT4                    ;SECTOR ERROR
                        000040          S.WCLK= BIT5                    ;WRITE CLOCK AND NO WRITE GATE
                        000100          S.WGAT= BIT6                    ;WRITE GATE AND NO TRANSISTIONS
                        000200          S.HDFL= BIT7                    ;HEAD FAULT
                        000400          S.MHD=  BIT8                    ;MULTIPLE HEAD SELECT
                        001000          S.XERR= BIT9                    ;INDEX ERROR
                        002000          S.DIB=  BIT10                   ;DIBIT ERROR
                        004000          S.PLO=  BIT11                   ;PLO ERROR
                        010000          S.NMOV= BIT12                   ;SEEK AND NO MOTION
                        020000          S.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
                        040000          S.BRKE= BIT14                   ;SERVO-BRAKE
                                        .SBTTL  COMMON MASKS
                        000007          M.DRV=  7                       ;DRIVE CODE
                        100000          M.PAR=  BIT15                   ;PARITY
                        000003          M.ID=   3                       ;BYTE ID
                        017760          M.CDIF= 17760                   ;CYLINDER DIFFERENCE/OFFSET
                        017760          M.CADD= 17760                   ;CYLINDER ADDRESS
                        077770          M.SER=  77770                   ;DRIVE SERIAL NUMBER
                        000760          M.SECT= 760                     ;SECTOR COUNT
                        007000          M.HEAD= 7000                    ;HEAD DECODE
                                        .SBTTL  TRAP CATCHER
                        000000                  .=0
                                        ;*ALL UNUSED LOCATIONS OF THE VECTOR AREA CONTAIN
                                        ;*A ".+2, IOT" SEQUENCE TO CATCH AND PROCESS ILLEGAL
                                        ;*TRAPS AND INTERRUPTS THAT MIGHT OCCUR.
                                        ;*THE IOT TRAP WHICH IS TAKEN ON THE ILLEGAL TRAP/INT
                                        ;*TRAPS TO THE $SCOPE ROUTINE WHICH (IF THE RETURN PC IS
                                        ;*LESS THAN 1002) JUMPS TO THE $ERROR ROUTINE.
                                        ;*THE $ERROR ROUTINE WILL REPORT THE ERROR AS FOLLOWS:
                                        ;*      PC=YYYYYY UNEXPECTED TRAP TO XXX
                                        ;*AND RETURN TO THE PROGRAM AT PC=YYYYYY+2
                                        ;*WHERE XXX=LOCATION OF ILLEGAL TRAP
                                        ;*      YYYYYY=PC AT TIME OF TRAP
                                        ;*NOTE: IF THE PROCESSOR IS NOT AN 11/05 THE PROGRAM
                                        ;*      CAN BE STARTED AT ADDRESS 0 AS WELL AS ADDRESS 200.
 000000  000000          $40CAT: HALT                    ;;HALT
 000002  000737                  BR      .-100           ;;BRANCH TO 177700 & TIME OUT (NOT ON
                                                         ;;11/05)
 000004  001772                  .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  001772         JMP     @#START ;;GO TO START OF PROGRAM
 000204  000137  001762         JMP     RESTRT          ;JUMP TO RESTART ROUTINE
         000214                  .=214
 000214  000137  001752         JMP     PARM            ;JUMP TO OPERATOR ASSIGNED PARMETERS
```

```
                    000220                      .=220
000220  000137  001742                          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.

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

```
                        .SBTTL  COMMON TAGS

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

            001100              .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                  .WORD   0
001102     000          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103     000          $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114     000          $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115     001          $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134     000          $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135     000          $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154     000          $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155     002          $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156     012          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A ''LINE FEED''
001157     000          $TPFLG: .BYTE   0               ;;''TERMINAL AVAILABLE'' FLAG (BIT<07>=0=YES)
001160  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001162  000000          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001164  000000          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001166  000000          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001170  000000          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001172  000000          $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001174  000000          $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001176  000000          $REG6:  .WORD   0               ;;CONTAINS (($REGAD)+14)
001200  000000          $REG7:  .WORD   0               ;;CONTAINS (($REGAD)+16)
001202  000000          $REG10: .WORD   0               ;;CONTAINS (($REGAD)+20)
001204  000000          $REG11: .WORD   0               ;;CONTAINS (($REGAD)+22)
001206  000000          $REG12: .WORD   0               ;;CONTAINS (($REGAD)+24)
001210  000000          $REG13: .WORD   0               ;;CONTAINS (($REGAD)+26)
001212  000000          $REG14: .WORD   0               ;;CONTAINS (($REGAD)+30)
001214  000000          $REG15: .WORD   0               ;;CONTAINS (($REGAD)+32)
001216  000000          $REG16: .WORD   0               ;;CONTAINS (($REGAD)+34)
001220  000000          $REG17: .WORD   0               ;;CONTAINS (($REGAD)+36)
001222  000000          $TMP0:  .WORD   0               ;;USER DEFINED
```

```
2264  001224  000000                $TMP1:   .WORD   0              ;;USER DEFINED
2265  001226  000000                $TMP2:   .WORD   0              ;;USER DEFINED
2266  001230  000000                $TMP3:   .WORD   0              ;;USER DEFINED
2267  001232  000000                $TMP4:   .WORD   0              ;;USER DEFINED
2268  001234  000000                $TMP5:   .WORD   0              ;;USER DEFINED
2269  001236  000000                $TMP6:   .WORD   0              ;;USER DEFINED
2270  001240  000000                $TMP7:   .WORD   0              ;;USER DEFINED
2271  001242  000000                $TMP10:  .WORD   0              ;;USER DEFINED
2272  001244  000000                $TMP11:  .WORD   0              ;;USER DEFINED
2273  001246  000000                $TMP12:  .WORD   0              ;;USER DEFINED
2274  001250  000000                $TMP13:  .WORD   0              ;;USER DEFINED
2275  001252  000000                $TMP14:  .WORD   0              ;;USER DEFINED
2276  001254  000000                $TMP15:  .WORD   0              ;;USER DEFINED
2277  001256  000000                $TMP16:  .WORD   0              ;;USER DEFINED
2278  001260  000000                $TMP17:  .WORD   0              ;;USER DEFINED
2279  001262  000000                $TIMES:  0                      ;;MAX. NUMBER OF ITERATIONS
2280  001264  000000                $ESCAPE: 0                      ;;ESCAPE ON ERROR ADDRESS
2281  001266  177607  000377        $BELL:   .ASCIZ  <207><377><377> ;;CODE FOR BELL
2282  001272     077                $QUES:   .ASCII  /?/            ;;QUESTION MARK
2283  001273     015                $CRLF:   .ASCII  <15>           ;;CARRIAGE RETURN
2284  001274  000012                $LF:     .ASCIZ  <12>           ;;LINE FEED
2285                                ;;**************************************************************
2286                                .SBTTL  APT MAILBOX-ETABLE
2287
2288                                ;;**************************************************************
2289                                .EVEN
2290  001276                        $MAIL:                          ;;APT MAILBOX
2291  001276  000000                $MSGTY:  .WORD   AMSGTY         ;;MESSAGE TYPE CODE
2292  001300  000000                $FATAL:  .WORD   AFATAL         ;;FATAL ERROR NUMBER
2293  001302  000000                $TESTN:  .WORD   ATESTN         ;;TEST NUMBER
2294  001304  000000                $PASS:   .WORD   APASS          ;;PASS COUNT
2295  001306  000000                $DEVCT:  .WORD   ADEVCT         ;;DEVICE COUNT
2296  001310  000000                $UNIT:   .WORD   AUNIT          ;;I/O UNIT NUMBER
2297  001312  000000                $MSGAD:  .WORD   AMSGAD         ;;MESSAGE ADDRESS
2298  001314  000000                $MSGLG:  .WORD   AMSGLG         ;;MESSAGE LENGTH
2299  001316                        $ETABLE:                        ;;APT ENVIRONMENT TABLE
2300  001316     000                $ENV:    .BYTE   AENV           ;;ENVIRONMENT BYTE
2301  001317     000                $ENVM:   .BYTE   AENVM          ;;ENVIRONMENT MODE BITS
2302  001320  000000                $SWREG:  .WORD   ASWREG         ;;APT SWITCH REGISTER
2303  001322  000000                $USWR:   .WORD   AUSWR          ;;USER SWITCHES
2304  001324  000000                $CPUOP:  .WORD   ACPUOP         ;;CPU TYPE,OPTIONS
2305                                ;*                              BITS 15-11=CPU TYPE
2306                                ;*                                      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
2307                                ;*                                      11/70=06,PDQ=07,Q=10
2308                                ;*                              BIT 10=REAL TIME CLOCK
2309                                ;*                              BIT  9=FLOATING POINT PROCESSOR
2310                                ;*                              BIT  8=MEMORY MANAGEMENT
2311  001326     000                $MAMS1:  .BYTE   AMAMS1         ;;HIGH ADDRESS,M.S. BYTE
2312  001327     000                $MTYP1:  .BYTE   AMTYP1         ;;MEM. TYPE,BLK#1
2313                                ;*                              MEM.TYPE BYTE   --  (HIGH BYTE)
2314                                ;*                                      900 NSEC CORE=001
2315                                ;*                                      300 NSEC BIPOLAR=002
2316                                ;*                                      500 NSEC MOS=003
2317  001330  000000                $MADR1:  .WORD   AMADR1         ;;HIGH ADDRESS,BLK#1
2318                                ;*                                MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
2319  001332     000                $MAMS2:  .BYTE   AMAMS2         ;;HIGH ADDRESS,M.S. BYTE
```

```
 2320  001333     000                $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
 2321  001334  000000                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
 2322  001336     000                $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
 2323  001337     000                $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
 2324  001340  000000                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
 2325  001342     000                $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
 2326  001343     000                $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
 2327  001344  000000                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
 2328  001346  000210                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
 2329  001350  000000                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
 2330  001352  177440                $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
 2331  001354  000000                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
 2332  001356  000000                $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
 2333  001360                        $ETEND:
 2334                                .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  057146         DH001
001364  060520         DT001
001366  060612         DF001

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

```
                                ;*        T.CS1   T.CS2   T.DS    T.ER    T.ASOF  T.DCYL  T.DA
                                ;*      RKBA    RKWC
                                ;*      T.BA    T.WC
                                ;
001370  000000                  EM2N:   .WORD   0
001372  000000                  DH2N:   .WORD   0
001374  060526                          DT002
001376  060616                          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  057174                  DH002A
001404  060476                          DT003
001406  060636                          DF003
                                ;
                                ;* ERROR ITEMS 4,5,6,87 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  060526                          DT004
001416  060646                          DF004

                                ;*      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
```

```
001424  060526                          DT005
001426  060676                          DF005

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

001430  000000                  EM6N:  .WORD   0
001432  000000                  DH6N:  .WORD   0
001434  060526                         DT006
001436  060732                         DF006

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

001440  000000                  EM7N:  .WORD   0
001442  000000                  DH7N:  .WORD   0
001444  060526                         DT007
001446  060756                         DF007

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

```
2559                                    ;*      ERROR AT COMPLETION OF OPERATION
2560                                    ;*      (DESCRIPITON OF OPERATION)
2561                                    ;*      TST NUM ERR PC  DRIVE
2562                                    ;*      $TESTN  $ERRPC  DRVNUM
2563                                    ;*        EXPT    1S
2564                                    ;*        $REG10  $REG11
2565                                    ;*
2566                                    ;*      THE ERRORS REPORTED BY THIS FORMAT ARE SOFTWARE DETECTED BY
2567                                    ;*      COMPARING EXPECTED RESULTS WITH ACTUAL RESULTS.  THE SPECIFIC
2568                                    ;*      ERRORS ARE;
2569                                    ;*      WORD COUNT INCORRECT
2570                                    ;*      BUS ADDRESS INCORRECT
2571                                    ;*      CYLINDER ADDRESS INCORRECT
2572                                    ;*      TRACK ADDRESS INCORRECT
2573                                    ;*      SECTOR ADDRESS INCORRECT
2574
2575  001450  000000                    EM10N:  .WORD   0
2576  001452  060007                            DH010
2577  001454  060576                            DT015
2578  001456  060776                            DF010
2579
2580                                    ;*      ERROR ITEM 11
2581                                    ;*      (ERROR INDICATOR OR STATUS BIT)
2582                                    ;*      NOT SET AS A RESULT OF
2583                                    ;*      (ANOTHER ERROR INDICATOR, STATUS BIT, OR OPERATION
2584                                    ;*      TST NUN ERR PC  DRIVE
2585                                    ;*      $TESTN  $ERRPC  DRVNUM
2586                                    ;*        RKCS1   RKCS2   RKDS    RKER    RKASOF   RKMR1
2587                                    ;*        T.CS1   T.CS2   T.DS    T.ER    T.ASOF   T.MR1
2588
2589  001460  000000                    EM11N:  .WORD   0
2590  001462  060133                            DH011
2591  001464  060526                            DT010
2592  001466  061016                            DF011
2593
2594                                    ;*      ERROR ITEM 12
2595                                    ;*      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
2596                                    ;*      "NOT RESET AS A RESULT OF"
2597
2598  001470  000000                    EM12N:  .WORD   0
2599  001472  060162                            DH012
2600  001474  060526                            DT010
2601  001476  061016                            DF011
2602
2603                                    ;*      ERROR ITEM 13
2604                                    ;*      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
2605                                    ;*      "SET AS A RESULT OF"
2606
2607  001500  000000                    EM13N:  .WORD   0
2608  001502  060213                            DH013
2609  001504  060526                            DT010
2610  001506  061016                            DF011
2611
2612                                    ;*      ERROR ITEM 14
2613                                    ;*      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
2614                                    ;*      "RESET AS A RESULT OF"
```

```
2615  001510  000000                  EM14N:  .WORD   0
2616  001512  060236                          DH014
2617  001514  060526                          DT010
2618  001516  061016                          DF011
2619
2620                                  ;*      ERROR ITEM 15
2621                                  ;*      (HEADER WORD MISCOMPARE) OR (DATA MISCOMPARE)
2622                                  ;*      TST NUM ERR PC  DRIVE
2623                                  ;*      $TESTN  $ERRPC  DRVNUM
2624                                  ;*        GOOD    BAD     WORD NUM
2625                                  ;*        $REG10  $REG11  $REG12
2626
2627  001520  000000                  EM15N:  .WORD 0
2628  001522  057174                          DH002A
2629  001524  060576                          DT015
2630  001526  061040                          DF015
2631
2632                                  ;*      ERROR ITEM 16
2633                                  ;*      ADDITIONAL LINES OF GOOD, BAD, WORD NUM FOR ERROR 15
2634
2635  001530  000000                          0
2636  001532  000000                          0
2637  001534  060604                          DT015A
2638  001536  061050                          DF016
2639
2640                                  .SBTTL   REGISTER STORAGE FOR TEST
2641
2642  001540  000000                  T.CS1:  .WORD   0
2643  001542  000000                  T.WC:   .WORD   0
2644  001544  000000                  T.BA:   .WORD   0
2645  001546  000000                  T.DA:   .WORD   0
2646  001550  000000                  T.CS2:  .WORD   0
2647  001552  000000                  T.DS:   .WORD   0
2648  001554  000000                  T.ER:   .WORD   0
2649  001556  000000                  T.ASOF: .WORD   0
2650  001560  000000                  T.DCYL: .WORD   0
2651  001562  000000                  T.SPAR: .WORD   0
2652  001564  000000                  T.DB:   .WORD   0
2653  001566  000000                  T.MR1:  .WORD   0
2654  001570  000000                  T.ECPS: .WORD   0
2655  001572  000000                  T.ECPT: .WORD   0
2656  001574  000000                  T.MR2:  .WORD   0
2657  001576  000000                  T.MR3:  .WORD   0
2658
2659                                  .SBTTL  REGISTER SETUP STORAGE
2660  001600  000100                  L.CS1:  .WORD   100              ;PRESET WITH INTERRUPT ENABLE
2661  001602  000000                  L.WC:   .WORD   0
2662  001604  000000                  L.BA:   .WORD   0
2663  001606                          L.DA:
2664  001606     000                  L.DS:   .BYTE   0
2665  001607     000                  L.DT:   .BYTE   0
2666  001610  000000                  L.CS2:  .WORD   0
2667  001612  000000                  L.ASOF: .WORD   0
2668  001614  000000                  L.DCYL: .WORD   0
2669  001616  000000                  L.MR1:  .WORD   0
2670                                  .SBTTL  PROGRAM DEFINED VARIABLES
```

```
2671
2672  001620  000000                RKVEC:  .WORD   0               ;RK VECTOR
2673  001622  000000                RKPRI:  .WORD   0               ;RK PRIORITY
2674  001624  000000                SRTFLG: .WORD   0               ;START FLAG
2675                                                                ;  0 = 200
2676                                                                ;  1 = 214
2677                                                                ; -1 = 204
2678  001626  000000                DRVNUM: .WORD   0               ;DRIVE UNDER TEST
2679  001630  000000                DRVBIT: .WORD   0               ;WORD TO STORE BIT TO INDICATE DRIVE UNDER TEST
2680  001632  000024                ERRCNT: .WORD   ^D20            ;ERROR COUNTER TO LIMIT ERROR
2681                                                                ;ERRORS REPORTED IN PROGRAM
2682  001634  000024                ERRLMT: .WORD   ^D20            ;DATA COMPARE ERROR LIMIT
2683  001636  061200                BSF24P: .WORD   BS24            ;POINTER TO BAD SECTORS 24 SECTOR MODE
2684                                                                ;  (FACTORY)
2685  001640  061054                BSF26P: .WORD   BS26            ;POINTER TO BAD SECTORS 26 SECTOR MODE
2686                                                                ;  (FACTORY)
2687  001642  000000                BSS24P: .WORD   0               ;POINTER TO BAD SECTORS 24 SECT MODE
2688                                                                ; (SOFTWARE)
2689  001644  000000                BSS26P: .WORD   0               ;POINTER TO BAD SECTORS 26 SECTOR MODE
2690                                                                ;  (SOFTWARE)
2691  001646  000000                BS26CT: .WORD   0               ;COUNT OF BAD SECTORS 26 SECTOR MODE
2692  001650  000000                BS24CT: .WORD   0               ;COUNT OF BAD SECTORS 24 SECTOR MODE
2693  001652  000764                MILCNT: .WORD   ^D500           ;COUNT TO APPROXIMATE 1 MILL SEC
2694  001654  000017                TIMCNT: .WORD   ^D15            ;COUNTER FOR MYTIME ROUTINE
2695           001660                        .=1660
2696  001660  000000                        .WORD   0
2697  001662  000000                INTSET: .WORD   0               ;NON-ZERO IF RK06 INTERRUPT SINCE
2698                                                                ;LAST CLEARED
2699  001664  000000                OPTFLG: .WORD   0               ;OPTION FLAGS
2700
2701           000001                        DOTST=  BIT0            ;DRIVE 0 TO BE TESTED FLAG
2702           000002                        MEMSZB= BIT1            ;MEMORY SIZE REPORT FLAG
2703           000004                        MEMPYB= BIT2            ;MEMORY PARITY REPORT FLAG
2704           000010                        SRTINS= BIT3            ;START UP INSTRUCTIONS REPORTED FLAG
2705           000100                        PARPRE= BIT6            ;PARITY PRESENT FLAG
2706           000200                        BSERPT= BIT7            ;BSE HAS BEEN REPORTED
2707           000400                        FPFMT=  BIT8            ;FIRST PASS FORMAT SWITCH
2708           002000                        CP1170= BIT10           ;CP IS 11/70 FLAG
2709           004000                        DRVRPT= BIT11           ;DRIVE NUMBERS REPORTED FLAG
2710           100000                        LCLKPR= BIT15           ;LINE CLOCK PRESENT
2711
2712  001666  000000                DRVDRP: .WORD   0               ;LIST OF DRIVES DROPPED
2713  001670  000000                MEMPAR: .WORD   0               ;WORD OF PARITY FLAGS
2714  001672  000000                CSRPTR: .WORD   0               ;POINTER TO CSR TO USE FOR SETTING BAD
2715                                                                ;  PARITY
2716  001674  000000                LCLKTK: .WORD   0               ;LINE CLOCK TICK COUNTER
2717  001676  000000                REFMT:  .WORD   0               ;REFORMAT SWITCH FOR HALT
2718
2719                                ;       THE FOLLOWING 4 VARIABLES ARE USED TO STORE PARAMETERS FOR
2720                                ;       HEADER OR DATA COMPARE CONTINUATION PROCESS.
2721  001700  000000                DESHLD: .WORD   0               ;DESTINATION HOLD
2722  001702  000000                SRCHLD: .WORD   0               ;SOURCE HOLD
2723  001704  000000                WRDNUM: .WORD   0               ;WORD NUMBER IN ERROR HOLD
2724  001706  000000                WRDCNT: .WORD   0               ;WORDS LEFT IN COMPARE HOLD
2725  001710  177546                KWLADD: .WORD   177546          ;KW11-L ADDRESS
2726  001712  000100                KWLVEC: .WORD   100             ;KW11-L VECTOR
```

```
2727  001714  172100                  MEMBAS: .WORD   172100          ;MM11 ADDRESS
2728  001716  000114                  MMVECA: .WORD   114             ;MM11 VECTOR
2729
2730  001720  000000                  DTYPE:  0                       ;DRV TYP  0=RK06, 2000=RK07
2731  001722  000000                  TYPTBL: 0                       ;DRV 0        0=RK06, 2000=RK07
2732  001724  000000                          0                       ;DRV 1
2733  001726  000000                          0                       ;DRV 2
2734  001730  000000                          0                       ;DRV 3
2735  001732  000000                          0                       ;DRV 4
2736  001734  000000                          0                       ;DRV 5
2737  001736  000000                          0                       ;DRV 6
2738  001740  000000                          0                       ;DRV7
2739
2740
2741
2742
2743
2744                                  .SBTTL  PROGRAM SETUP
2745
2746  001742  012737  000002  001624  SETCLK: MOV     #2,SRTFLG       ;SET START FLAG FOR CLOCK ADJUST
2747  001750  000412                  BR      START1
2748
2749  001752  012737  000001  001624  PARM:   MOV     #1,SRTFLG       ;SET START FLAG FOR PARMETER START
2750  001760  000406                  BR      START1
2751
2752  001762  012737  177777  001624  RESTRT: MOV     #-1,SRTFLG      ;LOAD START FLAG FOR PARMETER START
2753  001770  000402                  BR      START1
2754
2755  001772  005037  001624          START:  CLR     SRTFLG          ;CLEAR START FLAG
2756  001776  000005                  START1: RESET                   ;RESET THE WHOLE SYSTEM
2757  002000  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
2758  002004  012746  000340                  MOV     #PR7,-(SP)      ;LOAD STACK TO LOCK OUT ALL INTERRUPTS
2759  002010  012746  002016                  MOV     #1$,-(SP)       ;LOAD START OF PROGRAM
2760  002014  000002                  RTI                             ;LOAD PSW
2761
2762  002016  004737  045146          1$:     JSR     PC,$TKINT       ;INITIALIZE KEYBOARD
2763  002022  005037  001676                  CLR     REFMT           ;CLEAR REFORMAT SWITCH
2764                                  .SBTTL  INITIALIZE THE COMMON TAGS
2765                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
2766  002026  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
2767  002032  005026                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
2768  002034  022706  001140                  CMP     #SWR,R6 ;;DONE?
2769  002040  001374                  BNE     .-6             ;;LOOP BACK IF NO
2770  002042  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
2771                                  ;;INITIALIZE A FEW VECTORS
2772  002046  012737  032576  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
2773  002054  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
2774  002062  012737  033614  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
2775  002070  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
2776  002076  012737  047006  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
2777  002104  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
2778  002112  012737  046630  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
2779  002120  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
2780  002126  013737  032026  032020          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
2781  002134  005037  001262                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
2782  002140  005037  001264                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
```

```
2783  002144  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
2784  002152  012737  002152  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
2785  002160  012737  002160  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
2786                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
2787                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
2788  002166  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
2789  002172  012737  002226  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
2790  002200  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
2791  002206  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
2792  002214  022777  177777  176716          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
2793  002222  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
2794                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
2795  002224  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
2796  002226  012716  002234          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
2797  002232  000002                          RTI
2798  002234  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
2799  002242  012737  000174  001142          MOV     #DISPREG,DISPLAY
2800  002250  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
2801
2802  002254  005037  001304                  CLR     $PASS           ;;CLEAR PASS COUNT
2803  002260  132737  000200  001317          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
2804  002266  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
2805  002270  012737  001320  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
2806  002276                          67$:
2807                                  .SBTTL  TYPE PROGRAM NAME
2808                                  ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
2809  002276  005227  177777                  INC     #-1             ;;FIRST TIME?
2810  002302  001046                          BNE     68$             ;;BRANCH IF NO
2811  002304  022737  032162  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
2812  002312  001442                          BEQ     68$             ;;BRANCH IF YES
2813  002314  104401  002362                  TYPE    ,69$            ;;TYPE ASCIZ STRING
2814                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
2815  002320  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
2816  002324  001012                          BNE     70$             ;;BRANCH IF YES
2817  002326  123727  001316  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
2818  002334  001406                          BEQ     70$             ;;BRANCH IF YES
2819  002336  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
2820  002344  001005                          BNE     71$             ;;BRANCH IF NO
2821  002346  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
2822  002350  000403                          BR      71$
2823  002352  112737  000001  001134  70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
2824  002360                          71$:
2825  002360  000417                          BR      68$             ;;GET OVER THE ASCIZ
2826                                  ;;69$:  .ASCIZ  <CRLF>*CZR6KF0 RK6 FCTNL CTRL DIAG*<CRLF>
2827  002420                          68$:
2828  002420  132737  000200  001317          BITB    #BIT7,$ENVM     ;TEST IF DO NOT SIZE
2829  002426  001061                          BNE     3$              ;YES - SKIP
2830  002430  004737  032234                  JSR     PC,$SIZE
2831  002434  023727  032574  000740          CMP     $LSTBK,#740     ;MAKE SURE MEMORY IS SUFFICIENT
2832  002442  103007                          BHIS    2$              ;YES - SKIP
2833  002444  104401  051075                  TYPE    ,OPR005         ;MESSAGE (NOT ENOUGH MEMORY)
2834  002450  012737  000001  032020          MOV     #1,$EOPC1       ;FORCE END OF PROGRAM
2835  002456  000137  031772                  JMP     $EOP
2836  002462                          2$:
2837  002462  012700  070414                  MOV     #IBUFF,R0       ;SET UP TO CLEAR 2000
2838  002466  012701  002000                  MOV     #2000,R1        ;WORDS OF BUFFER SPACE
```

```
2839  002472  005020                  30$:    CLR     (R0)+           ;LOOP -
2840  002474  005301                          DEC     R1              ;UNTIL -
2841  002476  001375                          BNE     30$             ;DONE
2842  002500  122737  000013  000041          CMPB    #13,@#41        ;LOAD FROM XXDP
2843  002506  001003                          BNE     99$             ;BRANCH IF NOT
2844  002510  104401  051146                  TYPE    ,OPR007
2845  002514  000000                          HALT
2846  002516                          99$:
2847  002516  013700  032574                  MOV     $LSTBK,R0       ;GET LAST BANK
2848  002522  012701  000006                  MOV     #6,R1           ;SET SHIFT COUNT
2849  002526  013703  032572                  MOV     $LSTAD,R3       ;GET LAST ADDRESS
2850  002532  005004                          CLR     R4              ;CLEAR R4 FOR OVERFLOW
2851  002534  005737  032316                  TST     $KT11           ;MEM MANAGE PRESENT?
2852  002540  100005                          BPL     23$             ;NO - SKIP
2853  002542  006100                  22$:    ROL     R0              ;SHIFT BANK LEFT
2854  002544  006104                          ROL     R4              ;ADD IN CARRY
2855  002546  005301                          DEC     R1              ;DECREMENT COUNT
2856  002550  001374                          BNE     22$             ;LOOP IF NOT ZERO
2857  002552  050003                          BIS     R0,R3           ;SET BANK BITS IN LAST ADDRESS
2858  002554  112737  000001  001327  23$:    MOVB    #1,$MTYP1       ;FORCE MEMORY TYPE TO 1
2859  002562  110437  001326                  MOVB    R4,$MAMS1       ;STORE UPPER MEMORY ADDRESS
2860  002566  010337  001330                  MOV     R3,$MADR1       ;STORE LOWER ADDRESS
2861  002572  032737  000010  001664  3$:     BIT     #SRTINS,OPTFLG  ;TEST IF ALREDY REPORTED
2862  002600  001005                          BNE     24$             ;YES - SKIP
2863  002602  104401  051676                  TYPE    ,OPR016         ;TYPE STARTUP INSTRUCTIONS
2864  002606  052737  000010  001664          BIS     #SRTINS,OPTFLG  ;SET REPORTED FLAG
2865  002614                          24$:
2866  002614  022737  000001  001624          CMP     #1,SRTFLG       ;CHECK IF PARAMETER START
2867  002622  001122                          BNE     15$             ;NO, START TESTING
2868  002624  104401  050766          5$:     TYPE    ,OPR001         ;TYPE "RK611 BUS ADDRESS (  ) ="
2869  002630  013746  001352                  MOV     $BASE,-(SP)     ;;SAVE $BASE FOR TYPEOUT
2870  002634  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2871  002636  104401  051015                  TYPE    ,OPR002
2872  002642  104412                          RDOCT                   ;GET VALUE
2873  002644  012637  001222                  MOV     (SP)+,$TMP0
2874  002650  001407                          BEQ     7$              ;CHECK IF <CR>
2875  002652  022737  160000  001222          CMP     #160000,$TMP0   ;CHECK IF IN I/O PAGE
2876  002660  101361                          BHI     5$
2877  002662  013737  001222  001352          MOV     $TMP0,$BASE     ;LOAD NEW BUS ADDRESS
2878  002670  104401  051023          7$:     TYPE    ,OPR003         ;TYPE "RK611 VECTOR ADDRESS (  ) ="
2879  002674  013746  001346                  MOV     $VECT1,-(SP)    ;GET $VECT1 FOR TYPOUT
2880  002700  042716  160000                  BIC     #160000,(SP)    ;CLEAR PRIORITY BITS
2881  002704  104402                          TYPOC
2882  002706  104401  051015                  TYPE    ,OPR002
2883  002712  104412                          RDOCT                   ;GET VALUE
2884  002714  012637  001222                  MOV     (SP)+,$TMP0
2885  002720  001412                          BEQ     10$             ;CHECK IF <CR>
2886  002722  022737  001000  001222          CMP     #1000,$TMP0
2887  002730  101757                          BLOS    7$              ;CHECK IF LEGAL
2888  002732  042737  017777  001346          BIC     #17777,$VECT1   ;CLEAR OLD VECTOR
2889  002740  053737  001222  001346          BIS     $TMP0,$VECT1    ;LOAD NEW VECTOR ADDRESS
2890  002746  104401  051053          10$:    TYPE    ,OPR004         ;TYPE "RK611 PRIORITY (  ) ="
2891  002752  005046                          CLR     -(SP)
2892  002754  113716  001347                  MOVB    $VECT1+1,(SP)
2893  002760  006216                          ASR     (SP)            ;SHIFT 5 BITS RIGHT
2894  002762  006216                          ASR     (SP)
```

```
2895  002764  006216                          ASR     (SP)
2896  002766  006216                          ASR     (SP)
2897  002770  006216                          ASR     (SP)
2898  002772  104402                          TYPOC
2899  002774  104401  051015                  TYPE    ,OPR002
2900  003000  104412                          RDOCT                       ;GET VALUE
2901  003002  012637  001222                  MOV     (SP)+,$TMP0
2902  003006  001430                          BEQ     15$                 ;CHECK IF <CR>
2903  003010  022737  000007  001222          CMP     #7,$TMP0            ;CHECK IF LEGAL
2904  003016  103753                          BLO     10$
2905  003020  022737  000004  001222          CMP     #4,$TMP0
2906  003026  101347                          BHI     10$
2907  003030  006337  001222                  ASL     $TMP0               ;SHIFT 5 BITS LEFT
2908  003034  006337  001222                  ASL     $TMP0
2909  003040  006337  001222                  ASL     $TMP0
2910  003044  006337  001222                  ASL     $TMP0
2911  003050  006337  001222                  ASL     $TMP0
2912  003054  042737  160000  001347          BIC     #160000,$VECT1+1 ;CLEAR OLD PRIORITY
2913  003062  053737  001222  001347          BIS     $TMP0,$VECT1+1   ;LOAD RK611 PRIORITY
2914  003070  005037  001304          15$:    CLR     $PASS               ;SET PASS COUNT TO ZERO
2915  003074  005037  001666                  CLR     DRVDRP              ;CLEAR DROPPED DRIVES LIST
2916  003100  042737  004000  001664          BIC     #DRVRPT,OPTFLG      ;CLEAR DRIVE #'S REPORTED FLAG
2917  003106  004737  034522                  JSR     PC,OPTTST           ;SETUP PARITY CHECK & CLOCK
2918  003112  013700  001346                  MOV     $VECT1,R0           ;STORE VECTOR FOR USE
2919  003116  042700  160000                  BIC     #160000,R0          ;CLEAR PRIORITY BITS
2920  003122  010037  001620                  MOV     R0,RKVEC
2921  003126  012710  034442                  MOV     #INTHLR,(R0)        ;SETUP INTERRUPT ADDRESS
2922  003132  113737  001347  001622          MOVB    $VECT1+1,RKPRI      ;STORE PRIORITY FOR USE
2923  003140  013702  001352                  MOV     $BASE,R2            ;SET BASE ADDRESS
2924  003144  005037  001264                  CLR     $ESCAPE             ;CLEAR ESCAPE
2925  003150  012746  000000          NEWPAS: MOV     #PR0,-(SP)          ;SET PRIORITY
2926  003154  012746  003162                  MOV     #16$,-(SP)
2927  003160  000002                          RTI
2928  003162                          16$:
2929
2930                                  .SBTTL  **BASIC INTERFACE AND OPTION TESTS
2931                                  ;;*****************************************************************
2932                                  ;*TEST 1        RK611 BASE ADDRESS TEST
2933                                  ;*      CHECK THAT READING THE RK611 BASE ADDRESS (RKCS1) DOES NOT
2934                                  ;*      CAUSE A NON-EXISTANT MEMORY TRAP. IF A TRAP OCCURS
2935                                  ;*      THE PROGRAM IS HALTED.
2936
2937                                  ;;*****************************************************************
2938  003162  000004                  TST1:   SCOPE
2939  003164  012737  000100  001262          MOV     #100,$TIMES         ;;DO 100 ITERATIONS
2940  003172  012706  001100                  MOV     #STACK,SP           ;CLEAN OFF STACK
2941  003176  012701  000004                  MOV     #4,R1               ;SET POINTER TO VECTOR
2942  003202  012146                          MOV     (R1)+,-(SP)         ;STORE OLD VECTOR CONTENTS
2943  003204  011146                          MOV     (R1),-(SP)
2944  003206  012701  000004                  MOV     #4,R1               ;RESET POINTER
2945  003212  012721  034434                  MOV     #NEXINT,(R1)+       ;SET VECTOR TO NEM TEST HANDLER
2946  003216  012711  000340                  MOV     #PR7,(R1)           ;SET PRIORITY
2947  003222  013702  001352                  MOV     $BASE,R2            ;SET POINTER TO RK611 BASE ADDRESS
2948  003226  005037  001662                  CLR     INTSET              ;CLEAR INTERRUPT COUNTER
2949  003232  012762  000000  000000          MOV     #0,RKCS1(R2)        ;WRITE CS1 TO SEE IN NEM WILL SET
2950  003240  000240                          NOP
```

```
2951  003242  000240                          NOP
2952  003244  000240                          NOP
2953  003246  005737  001662                  TST     INTSET          ;TEST IF COUNTER IS 0
2954  003252  001406                          BEQ     1$              ;YES - SKIP ERROR REPORT
2955  003254  012737  053315  001360          MOV     #EM1,EM1N       ;MESSAGE (NON-EXISTANT MEMORY TRAP ERR)
2956  003262  104001                          ERROR   1
2957  003264  000137  043724                  JMP     CTRHLT          ;GO TO CONTROLLED HALT
2958  003270  012701  000006          1$:     MOV     #6,R1           ;RESTORE VECTOR
2959  003274  012611                          MOV     (SP)+,(R1)
2960  003276  012641                          MOV     (SP)+,-(R1)
2961
2962                                  ;;*****************************************************************
2963                                  ;*TEST 2        INTERRUPT VECTOR ADDRESS TEST
2964                                  ;*      CHECK THAT THE INTERRUPT VECTOR FOR THE RK611 IS SET TO THE
2965                                  ;*      EXPECTED ADDRESS. IF INTERRUPT VECTOR IS IN ERROR,
2966                                  ;*      THE PROGRAM IS HALTED.
2967
2968                                  ;;*****************************************************************
2969  003300  000004          TST2:   SCOPE
2970  003302  012737  000100  001262          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
2971  003310  012762  005000  000010          MOV     #CLR,RKCS2(R2) ;CLEAR SUBSYSTEM, SPECIFICALLY TO
2972                                                                  ;CLEAR ANY OLD INTERRUPTS
2973  003316  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT COUNTER
2974  003322  012762  000300  000000          MOV     #RDY!IE,RKCS1(R2)       ;WRITE CS1 TO FORCE INTERRUPT
2975  003330  000240                          NOP
2976  003332  000240                          NOP
2977  003334  000240                          NOP
2978  003336  005737  001662                  TST     INTSET          ;TEST IF INTERRUPT OCCURRED
2979  003342  001011                          BNE     3$              ;YES - SKIP ERROR REPORT
2980  003344  105737  001103                  TSTB    $ERFLG          ;TEST IF ERFLG ALREADY SET. IF SO THE
2981                                                                  ;INTERRUPT WENT TO THE WRONG VECTOR
2982                                                                  ;AND MESSAGE HAS BEEN REPORTED.
2983  003350  001004                          BNE     2$              ;THEREFORE - EXIT
2984  003352  012737  053315  001360          MOV     #EM1,EM1N       ;MESSAGE (NO INTERRUPT)
2985  003360  104001                          ERROR   1
2986  003362  000137  043724          2$:     JMP     CTRHLT          ;GO TO CONTROLLED HALT
2987  003366                          3$:
2988
2989                                  .SBTTL  **STATUS VALID TESTS
2990
2991
2992                                  ;;*****************************************************************
2993                                  ;*TEST 3        SELECT ALL DRIVES
2994                                  ;*
2995                                  ;*      IF NOT RUNNING IN APT AUTOMATIC ENVIRONMENT,
2996                                  ;*      DETERMINE WHAT DRIVES ARE ON-LINE BY
2997                                  ;*      SELECTING ALL DRIVES.  IF NON-EXISTENT DRIVE REPORTED
2998                                  ;*      MAKE SURE STATUS VALID IS RESET.  IF DRIVE
2999                                  ;*      PRESENT MAKE SURE NO ERROR EXISTS, DRIVE
3000                                  ;*      IS CYCLED UP, AND STATUS VALID SET, AND DSC RESET.
3001                                  ;*
3002                                  ;*      IF RUNNING IN APT AUTOMATIC ENVIRONMENT, THE DRIVES
3003                                  ;*      IDENTIFIED IN ETABLE ARE TESTED FOR NO ERROR, DRIVE
3004                                  ;*      CYCLED UP, AND STATUS VALID SET.
3005                                  ;*
3006                                  ;*      IF LOCATION 41 INDICATES THE XXDP MEDIA IS ON
```

```
                        ;*      THE RK06, DRIVE 0 WILL ONLY BE TESTED IF THE PARAM
                        ;*      START (214) WAS USED. IF THE AUTOMATIC START (200)
                        ;*      IS USED, DRIVE 0 IS NOT TESTED. THE RESTART (204)
                        ;*      WILL RETAIN THE TEST STATUS OF DRIVE 0.
                        ;*
                        ;*      IF THE PARAM START IS USED, THE OPERATOR MUST
                        ;*      EITHER PLACE DRIVE 0 OFF LINE IF IT IS NOT TO BE TESTED
                        ;*      OR UNLOADED AND A SCRATCH MEDIA MOUNTED IF IT IS TO
                        ;*      BE TESTED. THE PROGRAM WILL MONITOR OFF LINE AND VOLUME
                        ;*      VALID TO DETERMINE THE TEST STATUS OF DRIVE 0.
                        ;*
                        ;*      ALL DRIVES TO BE TESTED MUST BE ON-LINE, CYCLED UP, AND
                        ;*      WRITE LOCK RESET. ADDRESSES OF DRIVES THAT ARE NON-
                        ;*      EXISTANT EITHER BECAUSE THE DRIVE DOES NOT EXIST OR IS OFF-
                        ;*      LINE ARE USED TO VERIFY NON-EXISTANT DRIVE ERROR DETECTION.
                        ;*      DRIVES THAT ARE ON-LINE BUT NOT CYCLED UP OR ARE WRITE
                        ;*      LOCKED ARE NOT TESTED.
                        ;*
                        ;*      AT COMPLETION OF THE TEST A MESSAGE WILL BE GIVEN TO
                        ;*       IDENTIFY THE DRIVES TO BE USED IN TESTING.
                        ;*
                        ;*      NOTE: THIS TEST MUST BE RUN AT LEAST ONCE BEFORE
                        ;*              ANY OTHER TEST THAT FOLLOWS.
                        ;*              THE DETERMINATION OF WHETHER EACH DRIVE IS AN
                        ;*              RK06 OR AN RK07 IS MADE IN THIS TEST, AND A
                        ;*              TABLE IS MADE ACCORDINGLY.
                        ;;*****************************************************************
003366  000004         TST3:   SCOPE
003370  012737  000062  001262        MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
003376  005037  001720              CLR     DTYPE           ;ASSUME RK06
003402  104416                      TSSINIT                 ;CALL SUBSYSTEM CLEAR AND TEST
003404  104003                      ERROR   3

003406  012746  000000              MOV     #PR0,-(SP)      ;SET PROCESSOR PRIORITY TO ALLOW
003412  012746  003420              MOV     #1$,-(SP)       ;RK611 INTERRUPTS
003416  000002                      RTI

003420  013701  001620      1$:     MOV     RKVEC,R1        ;GET VECTOR
003424  012721  034442              MOV     #INTHLR,(R1)+ ;LOAD INTERRUPT VECTOR
003430  012711  000340              MOV     #PR7,(R1)
003434  012703  001354              MOV     #$DEVM,R3       ;GET ADDRESS OF DEVICE MAP
003440  132737  000200  001317      BITB    #BIT7,$ENVM     ;TEST IF SHOULD SIZE
003446  001007                      BNE     50$             ;NO - SKIP DRIVE SIZING.
003450  005737  001304              TST     $PASS           ;TEST IF FIRST PASS
003454  001004                      BNE     50$             ;NO - SKIP TO DRIVE SELECT TEST
003456  032737  004000  001664      BIT     #DRVRPT,OPTFLG  ;TEST IF DRIVE #'S REPORTED
003464  001402                      BEQ     92$             ;NO - GOTO SIZING
003466  000137  004142      50$:    JMP     11$
003472  005013              92$:    CLR     (R3)            ;CLEAR DEVICE MAP
003474  123727  000041  000013      CMPB    @#41,#13        ;TEST IF RK06 IS LOAD DEVICE
003502  001111                      BNE     77$             ;NO - SKIP
003504  022737  000001  001624      CMP     #1,SRTFLG       ;WAS START AT PARAM?
003512  001411                      BEQ     2$              ;YES - SKIP
003514  104401  051146              TYPE    ,OPR007         ;NO TEST OF DRIVE 0
003520  052737  000001  001666      BIS     #BIT0,DRVDRP    ;SET DRIVE 0 DROPPED
003526  042737  000001  001664      BIC     #DOTST,OPTFLG   ;DR FLAG - NO TEST DRIVE 0
```

```
3063  003534  000477                        BR      7$
3064  003536  104401  051146          2$:   TYPE    ,OPR006         ;MESSAGE - SWAP PACK ON DRIVE OFF LINE.
3065  003542  005037  001610                CLR     L.CS2           ;SET TO DRIVE 0
3066  003546  005037  001232                CLR     $TMP4           ;CLEAR FOR USE AS A SWITCH
3067  003552  012737  000101  001600  3$:   MOV     #SELDRV,L.CS1   ;LOAD FOR SELECT
3068  003560  012737  003646  001264        MOV     #4$,$ESCAPE     ;SET UP ESCAPE IN CASE OF ERR
3069  003566  104417                        TLOADRK                 ;LOAD RK & DO SELECT
3070
3071  003570  104423                        TWAT16                  ;WAIT 16MS FOR COMPLETION
3072  003572  104002                        ERROR   2               ;NOT DONE ON TIME
3073
3074  003574  104420                        TGETRK                  ;GET RK REGISTER
3075  003576  032737  100000  001540        BIT     #CERR,T.CS1     ;TEST IF CERR
3076  003604  001431                        BEQ     5$              ;NO - SKIP
3077  003606  032737  010000  001550        BIT     #NED,T.CS2      ;TEST IF NED
3078  003614  001014                        BNE     4$              ;YES - SKIP
3079  003616  032737  000040  001554        BIT     #DTYE,T.ER      ;ELSE SEE IF DRIVE TYPE ERROR
3080  003624  001406                        BEQ     26$             ;BR IF NO
3081
3082  003626  012737  002000  001720        MOV     #CDT,DTYPE      ;ELSE SETUP FOR RK07
3083
3084  003634  104416                        TSSINIT                 ;INIT SUBSYS
3085  003636  104003                        ERROR   3               ;NOT SUCCESSFUL
3086  003640  000744                        BR      3$              ;AND TRY AGAIN
3087
3088  003642  104421          26$:          TCHKOP                  ;CHECK THE OPERATION AND REPORT THE ERROR
3089  003644  104004                        ERROR   4 ;OR5,6,7      ;AFTER THE ERROR IS REPORTED THE TEST
3090                                                                ;IS ABORTED
3091  003646  104401  051146          4$:   TYPE    ,OPR007         ;TYPE NO TEST OF DRIVE 0
3092  003652  042737  000001  001664        BIC     #DOTST,OPTFLG   ;DR FLAG - NO TEST OF DRIVE 0
3093  003660  052737  000001  001666        BIS     #BIT0,DRVDRP    ;SET DRV 0 AS DROPPED
3094  003666  000422                        BR      7$              ;SKIP OVER WAIT FOR PACK MOUNT
3095  003670  005737  001232          5$:   TST     $TMP4           ;TEST FLAG DRIVE READY HAS RESET
3096  003674  001010                        BNE     6$              ;YES - SKIP TO CHECK IF IT IS SET AGAIN
3097  003676  032737  000200  001552        BIT     #DRDY,T.DS      ;ELSE CHECK READY
3098  003704  001322                        BNE     3$              ;STILL SET - GET STATUS AGAIN
3099  003706  012737  177777  001232        MOV     #-1,$TMP4       ;ELSE SET FLAG TO INDICATE READY WENT LOW
3100  003714  000716                        BR      3$              ;GO GET STATUS AGAIN
3101
3102  003716  032737  000200  001552  6$:   BIT     #DRDY,T.DS      ;TEST IF READY SET AGAIN
3103  003724  001712                        BEQ     3$              ;NO - GO GET STATUS AGAIN
3104  003726  052737  000001  001664  77$:  BIS     #DOTST,OPTFLG   ;ELSE SET DRV 0 TEST FLAG.
3105
3106  003734  012737  004042  001264  7$:   MOV     #35$,$ESCAPE    ;SET UP ESCAPE IN CASE OF ERR
3107  003742  005000                        CLR     R0              ;CLEAR FOR DRIVE NUMBER COUNTER
3108  003744  012701  000001                MOV     #1,R1           ;SET BIT 0 AS DRIVE SELECTOR
3109  003750  005037  001720                CLR     DTYPE           ;INIT TO RK06
3110
3111  003754  032737  000001  001664        BIT     #DOTST,OPTFLG   ;TEST DRIVE 0?
3112  003762  001440                        BEQ     9$              ;NO - SKIP
3113
3114  003764  104416          8$:           TSSINIT                 ;INITIALZE SUBSYSTEM
3115  003766  104003                        ERROR   3               ;ERROR IF NOT SUCCESSFUL
3116
3117  003770  010037  001610                MOV     R0,L.CS2        ;LOAD DRIVE NUMBER
3118  003774  012737  000101  001600        MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
```

```
3119  004002  104417                          TLOADRK                 ;LOAD RK REGS
3120
3121  004004  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3122  004006  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3123
3124  004010  104420                          TGETRK                  ;GET RK REGS
3125  004012  032737  100000  001540          BIT     #CERR,T.CS1     ;ERROR?
3126  004020  001031                          BNE     10$             ;YES - SKIP
3127  004022  032737  000200  001552          BIT     #DRDY,T.DS      ;ELSE TEST IF DRIVE READY
3128  004030  001404                          BEQ     35$             ;NO - SKIP
3129  004032  032737  004000  001552          BIT     #WRL,T.DS       ;ELSE TEST IF WRITE LOCKED
3130  004040  001403                          BEQ     36$             ;NO - SKIP
3131  004042  050137  001666          35$:    BIS     R1,DRVDRP       ;SET THIS BIT AS DROPPED DRIVE
3132  004046  000406                          BR      9$
3133  004050                          36$:
3134  004050  050113                          BIS     R1,(R3)         ;SET BIT - DRIVE PRESET IN MAP
3135  004052  006300                          ASL     R0
3136  004054  013760  001720  001722          MOV     DTYPE,TYPTBL(R0)        ;SET TABLE WITH DRIVE TYPE INFO
3137  004062  006200                          ASR     R0
3138
3139  004064  005200                  9$:     INC     R0              ;BUMPS TO NEXT DRIVE
3140  004066  006301                          ASL     R1              ;SHIFT DRIVE SELECTOR TO NEXT DRIVE.
3141  004070  032701  000400                  BIT     #BIT8,R1        ;WAS LAST DRIVE DONE?
3142  004074  001022                          BNE     11$             ;BR IF YES
3143  004076  005037  001720                  CLR     DTYPE           ;ELSE INIT FOR RK06 TO TEST NEXT DRV
3144  004102  000730                          BR      8$
3145
3146  004104  032737  010000  001550  10$:    BIT     #NED,T.CS2      ;WAS CERR DUE TO NED?
3147  004112  001364                          BNE     9$              ;YES - BUMP TO NEXT DRIVE
3148  004114  032737  000040  001554          BIT     #DTYE,T.ER      ;ELSE SEE IF DRIVE TYPE ERR
3149  004122  001404                          BEQ     24$             ;BR IF NO
3150
3151  004124  012737  002000  001720          MOV     #CDT,DTYPE      ;ELSE SET UP FOR RK07
3152  004132  000714                          BR      8$              ;AND TRY AGAIN
3153
3154  004134  104421                  24$:    TCHKOP                  ;ELSE REPORT THE ERRORS
3155  004136  104004                          ERROR   4 ;GR5,6,7
3156  004140  000000                  101$:   .WORD   0
3157  004142  032737  004000  001664  11$:    BIT     #DRVRPT,OPTFLG  ;TEST IF SHOULD REPORT
3158  004150  001037                          BNE     16$             ;NO - SKIP
3159
3160  004152  005713                          TST     (R3)            ;ANY DRIVE AVAILABLE?
3161  004154  001004                          BNE     12$             ;BR IF NOT ZERO
3162  004156  104401  051227                  TYPE    ,OPR008         ;ELSE REPORT NO DRIVES AVAILABLE
3163  004162  000137  043724                  JMP     CTRHLT          ;GO TO CONTROLLED HALT
3164
3165  004166  012701  000200          12$:    MOV     #BIT7,R1        ;SET DRIVE SELECTOR FOR DRIVE 7
3166  004172  012700  000007                  MOV     #7,R0           ;SET DRIVE NUMBER TO 7
3167  004176  104401  051312                  TYPE    ,OPR009         ;TYPE PREFIX TO DRIVE TEST MESSAGE
3168
3169  004202  030113                  13$:    BIT     R1,(R3)         ;TEST IF THIS DRIVE TO BE TESTED
3170  004204  001007                          BNE     15$             ;YES - SKIP
3171
3172  004206  005300                  14$:    DEC     R0              ;ELSE DECREMENT DRIVE NUMBER
3173  004210  006201                          ASR     R1              ;SHIFT BIT SELECTOR TO NEXT DRIVE DOWN
3174  004212  001373                          BNE     13$             ;IF NOT SHIFTED OUT - LOOP
```

```
3175  004214  052737  004000  001664          BIS     #DRVRPT,OPTFLG  ;SET DRIVE #'S REPORTED FLAG
3176  004222  000412                          BR      16$             ;ELSE GO TO STATUS VALID TEST
3177
3178  004224  010037  004140          15$:    MOV     R0,101$         ;PUT DRIVE NUMBER IN TYPE LOCATION
3179  004230  052737  000060  004140          BIS     #BIT4!BIT5,101$ ;MAKE IT ASCIZ
3180  004236  104401                          TYPE                    ;TYPE IT
3181  004240  004140                          101$
3182  004242  104401  050763                  TYPE    ,SPACE2         ;TYPE SOME SPACES
3183  004246  000757                          BR      14$             ;LOOP
3184
3185  004250  005000          16$:    CLR     R0              ;CLEAR DRIVE NUMBER COUNTER
3186  004252  012701  000001                  MOV     #1,R1           ;SET DRIVE SELECTOR TO DRIVE 0
3187  004256  012737  177777  001240          MOV     #-1,$TMP7       ;SET $TMP7 NEGATIVE
3188  004264  012737  177777  001630          MOV     #-1,DRVBIT      ;SET DRIVE SELECT BIT NEGATIVE
3189  004272  012737  004400  001264          MOV     #18$,$ESCAPE    ;SET ESCAPE IN CASE OF ERR
3190  004300  030137  001666                  BIT     R1,DRVDRP       ;HAS DRIVE 0 BEEN DROPPED?
3191  004304  001035                          BNE     18$             ;YES - SKIP TO DRIVE INC.
3192
3193  004306  104416          17$:    TSSINIT                 ;CLEAR SUBSYSTEM
3194  004310  104003                  ERROR   3               ;ERROR FOR BAD CLEAR
3195
3196  004312  010037  001610                  MOV     R0,L.CS2        ;SET DRIVE SELECT
3197  004316  010037  001626                  MOV     R0,DRVNUM       ;SET DRIVE NUMBER
3198  004322  004737  036134                  JSR     PC,GTYP0        ;GET DRV TYPE
3199  004326  012737  000101  001600          MOV     #SELDRV,L.CS1   ;SET FOR DRIVE SELECT
3200
3201  004334  104417                  TLOADRK                 ;LOAD RK REGS
3202  004336  104423                  TWAT16                  ;WAIT FOR INTERRUPT
3203  004340  104002                  ERROR   2               ;ERROR TO SLOW/NOT COMPLETE
3204  004342  030113                  BIT     R1,(R3)         ;WAS THAT DRIVE AVAILABLE
3205  004344  001026                  BNE     19$             ;YES - SKIP
3206
3207  004346  104422                  TCHKWE                  ;CHECK THAT ERROR OCCURRED
3208  004350  000000                  .WORD   0               ;NONE OF GROUP 1
3209  004352  000000                  .WORD   0               ;NONE OF GROUP 2
3210  004354  000001                  .WORD   1               ;NED IN GROUP 3
3211  004356  104004                  ERROR   4 ;OR5,6,7      ;ERROR IF NO ERROR OR WRONG ERROR
3212
3213  004360  032737  100000  001552          BIT     #SVAL,T.DS      ;DID STATUS VALID RESET?
3214  004366  001404                          BEQ     18$             ;YES - SKIP
3215  004370  012737  055437  001400          MOV     #EM47,EM3N      ;ELSE MESSAGE (SVAL NOT RESET WITH NED)
3216  004376  104003                          ERROR   3
3217  004400  005200          18$:    INC     R0              ;BUMP TO NEXT DRIVE
3218  004402  006301                  ASL     R1              ;SHIFT DRIVE SELECT BIT
3219  004404  032701  000400                  BIT     #BIT8,R1        ;ALL DRIVES CHECKED
3220  004410  001063                          BNE     21$             ;YES - EXIT
3221  004412  030137  001666                  BIT     R1,DRVDRP       ;TEST IF DRIVE HAS BEEN DROPPED
3222  004416  001370                          BNE     18$             ;YES - GET NEXT DRIVE
3223  004420  000732                          BR      17$             ;ELSE CHECK THIS DRIVE
3224  004422  104421          19$:    TCHKOP                  ;CHECK NO ERRORS SET
3225  004424  104004                  ERROR   4 ;OR5,6,7      ;REPORT ALL ERRORS
3226
3227  004426  032737  100000  001552          BIT     #SVAL,T.DS      ;CHECK SVAL SET
3228  004434  001004                          BNE     20$             ;YES - SKIP
3229  004436  012737  055520  001400          MOV     #EM48,EM3N      ;MESSAGE (NO SVAL FROM EXISTANT DR)
3230  004444  104003                          ERROR   3
```

```
3231
3232  004446  012737  000103  001600  20$:    MOV     #PACK,L.CS1     ;ELSE SET TO DO PACK ACK
3233  004454  104417                          TLOADRK                 ;LOAD RK
3234
3235  004456  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3236  004460  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3237
3238  004462  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
3239  004464  104004                          EFROR   4 ;OR5,6,7      ;YES - REPORT & ABORT TEST
3240
3241  004466  032737  000100  001552          BIT     #VV,T.DS        ;DID VV SET
3242  004474  001005                          BNE     22$             ;YES - SKIP
3243  004476  012737  055402  001400          MOV     #EM46,EM3N      ;MESSAGE (VV DID NOT SET/
3244  004504  104003                          ERROR   3
3245  004506  000734                          BR      18$
3246
3247  004510  032737  040000  001552  22$:    BIT     #DSC,T.DS       ;TEST IF DSC RESET
3248  004516  001410                          BEQ     23$             ;YES - SKIP
3249  004520  012737  056514  001470          MOV     #EMDSC,EM12N
3250  004526  012737  056754  061022          MOV     #EMSCLR,DF011A
3251  004534  104003                          ERROR   3               ;''DSC NOT RESET RESULT OF SUBSYS CLEAR''
3252  004536  000720                          BR      18$
3253
3254  004540  005737  001630          23$:    TST     DRVBIT          ;TEST IF DRVBIT IS NEGATIVE
3255  004544  100315                          BPL     18$             ;NO - SKIP
3256  004546  010137  001630                  MOV     R1,DRVBIT       ;STORE DRIVE SELECT BIT
3257  004552  010037  001240                  MOV     R0,$TMP7        ;STORE DRIVE NUMBER TO BE TESTED
3258  004556  000710                          BR      18$
3259
3260  004560                          21$:
3261  004560  005037  001264                  CLR     $ESCAPE         ;REV 006
3262  004564  013737  001240  001626          MOV     $TMP7,DRVNUM    ;LOAD LOWEST # DRIVE PRESENT INTO DRVNUM
3263  004572  013701  001626                  MOV     DRVNUM,R1
3264  004576  004737  036150                  JSR     PC,GTYP1        ;GET DRV TYP
3265
3266  004602  023727  001624  000002          CMP     SRTFLG,#2       ;TEST IF CLOCK ADJUST START
3267  004610  001002                          BNE     25$             ;NO - SKIP
3268  004612  000137  043614                  JMP     ADJCLK          ;GO TO ADJUST CLOCK ROUTINE
3269
3270  004616                          25$:
3271
3272                                  ;;*******************************************************************
3273                                  ;*TEST 4        RELEASE ALL DRIVES
3274                                  ;*
3275                                  ;*      RELEASE ALL DRIVES.  MAKE SURE NO ERROR
3276                                  ;*      SETS AND STATUS VALID IS RESET.
3277                                  ;*
3278                                  ;*******************************************************************
3279  004616  000004                  TST4:   SCOPE
3280  004620  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3281  004626  104416                          TSSINIT                 ;INITIALIZE SUBSYSTEM
3282  004630  104003                          ERROR   3               ;BAD INIT
3283
3284  004632  013737  001626  001610          MOV     DRVNUM,L.CS2    ;SET DRIVE NUMBER
3285  004640  012737  000101  001600          MOV     #SELDRV,L.CS1   ;SET DRIVE SELECT
3286
```

```
3287  004646  104417                          TLOADRK                 ;LOAD RK REGS
3288  004650  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3289  004652  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3290
3291  004654  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
3292  004656  104004                          ERROR   4 ;OR5,6,7      ;REPORT ANY ERRORS
3293
3294  004660  012737  000010  001610          MOV     #RLS,L.CS2      ;SET DRIVE RELEASE,STILL SET FOR SELECT
3295
3296  004666  104417                          TLOADRK                 ;LOAD RK REGS
3297  004670  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3298  004672  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3299
3300  004674  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
3301  004676  104004                          ERROR   4  ;OR 5, 6, OR 7 ;REPORT ALL ERRORS
3302  004700  032737  100000  001552          BIT     #SVAL,T.DS      ;DID SVAL RESET?
3303  004706  001404                          BEQ     1$              ;YES - SKIP
3304  004710  012737  055577  001400          MOV     #EM49,EM3N      ;MESSAGE (SVAL NOT RESET W/RELEASE)
3305  004716  104003                          ERROR   3
3306  004720                          1$:
3307
3308  004720  004737  036034          TSTLUP: JSR     PC,LDCON        ;LOAD RK06-RK07 CONSTANTS
3309                                  ;;*****************************************************************
3310                                  ;*TEST 5         NON-STANDARD MESSAGES AND SVAL
3311                                  ;*
3312                                  ;*      PICK ONE OF THE AVAILABLE DRIVES AND GET
3313                                  ;*      NON-STANDARD MESSAGES.  MAKE SURE NO
3314                                  ;*      ERROR OCCURS AND STATUS VALID DOES NOT SET
3315                                  ;*      AND THAT NON-STANDARD MESSAGES CAUSE STATUS
3316                                  ;*      VALID TO RESET.
3317                                  ;*
3318                                  ;;*****************************************************************
3319  004724  000004                  TST5:   SCOPE
3320  004726  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3321  004734  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
3322  004736  104003                          ERROR   3               ;BAD CLEAR MESSAGE
3323  004740  012701  000001                  MOV     #1,R1           ;PRESET R1 FOR MESSAGE PAIR 1
3324  004744  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRV NUMBER
3325  004752  012737  000101  001600          MOV     #SELDRV,L.CS1   ;LOAD SELECT COMMAND
3326  004760  005037  001616          1$:     CLR     L.MR1           ;LOAD FOR STANDARD STATUS
3327  004764  104417                          TLOADRK                 ;LOAD RK
3328  004766  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3329  004770  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3330  004772  104421                          TCHKOP                  ;CHECK OPERATION
3331  004774  104004                          ERROR   4 ;5,6 OR 7     ;REPORT ALL ERRORS
3332
3333  004776  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST STATUS VALID SET
3334  005004  001007                          BNE     2$              ;YES-SKIP
3335
3336  005006  012737  056354  001460          MOV     #EMSVAL,EM11N
3337  005014  012737  050160  061022          MOV     #EMSELD,DF011A
3338  005022  104011                          ERROR   11              ;"SVAL NOT SET RESULT Of DRIVE SELECT"
3339
3340  005024  010137  001616          2$:     MOV     R1,L.MR1        ;LOAD MESSAGE PAIR SELECT
3341
3342  005030  104417                          TLOADRK                 ;LOAD RK
```

```
3343  005032  104423                          TWAIT6                  ;WAIT FOR INTERRUPT
3344  005034  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3345
3346  005036  104421                          TCHKOP                  ;CHECK OPERATION
3347  005040  104004                          ERROR   4 ;5,6, OR 7    ;REPORT ALL ERRORS
3348
3349  005042  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST STATUS VALID RESET
3350  005050  001407                          BEQ     3$              ;YES-SKIP
3351
3352  005052  012737  056354  001470          MOV     #EMSVAL,EM12N
3353  005060  012737  056371  061022          MOV     #EMNZPR,DF011A
3354  005066  104012                          ERROR   12              ;''SVAL NOT RESET RESULT OF SEL W' NON-0 PAIR''
3355
3356  005070  022701  000003          3$:     CMP     #3,R1           ;WAS PAIR 3 SELECTED?
3357  005074  001402                          BEQ     4$              ;YES-SKIP
3358  005076  005201                          INC     R1              ;BUMP TO NEXT PAIR
3359  005100  000727                          BR      1$              ;SKIP TO DO IT.
3360  005102                          4$:
3361                                  ;;*****************************************************************
3362                                  ;*TEST 6        WRITING CS2 AND STATUS VALID
3363                                  ;*
3364                                  ;*      SELECT AN AVAILABLE DRIVE.  MAKE SURE STATUS
3365                                  ;*      VALID IS SET.  WRITE COMMAND AND STATUS REGISTER 2.
3366                                  ;*      MAKE SURE STATUS VALID RESETS.
3367                                  ;*
3368                                  ;;*****************************************************************
3369  005102  000004                  TST6:   SCOPE
3370  005104  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3371  005112  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
3372  005114  104003                          ERROR   3               ;BAD INIT ERROR
3373
3374  005116  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3375  005124  012737  000101  001600          MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
3376
3377  005132  104417                          TLOADRK                 ;LOAD RK
3378  005134  104423                          TWAIT6                  ;WAIT FOR INTERRUPT
3379  005136  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3380
3381  005140  104421                          TCHKOP                  ;CHFCK OPERATION
3382  005142  104004                          ERROR   4 ;5,6, OR 7    ;REPORT ALL ERRORS
3383
3384  005144  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST STATUS VALID SET
3385  005152  001007                          BNE     1$              ;YES-SKIP
3386
3387  005154  012737  056354  001460          MOV     #EMSVAL,EM11N
3388  005162  012737  050160  061022          MOV     #EMSELD,DF011A
3389  005170  104011                          ERROR   11              ;''SVAL NOT SET RESULT OF DRV SELECT''
3390
3391  005172  013762  001626  000010  1$:     MOV     DRVNUM,RKCS2(R2)  ;WRITE CS2 TO RESET SVAL
3392
3393  005200  104420                          TGETRK                  ;GET RK REGS.
3394
3395  005202  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST SVAL RESET
3396  005210  001407                          BEQ     2$              ;YES-SKIP
3397
3398  005212  012737  056354  001470          MOV     #EMSVAL,EM12N
```

```
3399  005220  012737  056432  061022          MOV     #EMWCS2,DF011A
3400  005226  104012                          ERROR   12              ;"SVAL NOT RESET BY WRITING CS2"
3401  005230                          2$:
3402
3403                                  .SBTTL  **CONTROLLER ERROR TESTS
3404
3405                                  ;;*******************************************************************
3406                                  ;*TEST 7         DRIVE TYPE ERROR
3407                                  ;*
3408                                  ;*      CREATE A DRIVE TYPE ERROR.  MAKE SURE DRIVE
3409                                  ;*      TYPE ERROR SETS AND STATUS VALID SETS.
3410                                  ;*
3411                                  ;;*******************************************************************
3412  005230  000004                  TST7:   SCOPE
3413  005232  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3414  005240  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
3415  005242  104003                          ERROR   3               ;BAD INIT ERROR
3416
3417  005244  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3418  005252  012737  000101  001600          MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
3419  005260  053737  036130  001600          BIS     HOLD2,L.CS1     ;LOAD DRIVE TYPE
3420
3421  005266  104417                          TLOADRK                 ;LOAD RK
3422  005270  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3423  005272  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3424
3425  005274  104422                          TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
3426  005276  000040                          .WORD   000040          ;DRIVE TYPE ERROR
3427  005300  000000                          .WORD   0
3428  005302  000000                          .WORD   0
3429  005304  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ANY DIFFERENCES (NO ERRORS,
3430                                                                  ;ADDITIONAL ERRORS, DIFFERENT ERRORS)
3431  005306  032737  100000  001552          BIT     #SVAL, T.DS     ;TEST IF SVAL SET
3432  005314  001007                          BNE     1$              ;YES-SKIP
3433
3434  005316  012737  056354  001460          MOV     #EMSVAL,EM11N
3435  005324  012737  055050  061022          MOV     #EMDTPE,DF011A
3436  005332  104011                          ERROR   11              ;"SVAL NOT SET RESULT OF DRV TYPE ERR"
3437  005334                          1$:
3438                                  ;;*******************************************************************
3439                                  ;*TEST 10        STATUS VALID AND PARITY ERROR
3440                                  ;*
3441                                  ;*      ISSUE A SELECT TO AN AVAILIABLE DRIVE WITH BAD PARITY.
3442                                  ;*      MAKE SURE SPAR, CONTROLLER ERROR, ATTENTION,
3443                                  ;*      DRIVE STATUS CHANGES, DRPAR, DRIVE INTERRUPT,
3444                                  ;*      AND STATUS VALID SET, ISSUE A CONTROLLER
3445                                  ;*      CLEAR.  MAKE SURE DRIVE INTERRUPT AND ATTENTION
3446                                  ;*      ARE STILL SET.  SELECT DRIVE AGAIN WITH GOOD
3447                                  ;*      PARITY.  MAKE SURE ATTENTION, DRIVE STATUS
3448                                  ;*      CHANGE, DRPAR, CONTROLLLER ERROR, DRIVE INTERRUPT,
3449                                  ;*      AND STATUS VALID ARE SET AND SPAR IS RESET.
3450                                  ;*      ISSUE A CONTROLLER CLEAR.  GET NON-STANDARD MESSAGES
3451                                  ;*      AND MAKE SURE ONLY DRIVE INTERRUPT AND ATTENTION
3452                                  ;*      ARE SET.  CLEAR ATTENTION WITH DRIVE CLEAR. REPEAT
3453                                  ;*      FOR ALL AVAILIABLE DRIVES.
3454                                  ;*
```

```
                                        ;;*****************************************************************
005334  000004                  TST10:  SCOPE
005336  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
005344  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
005346  104003                          ERROR   3               ;BAD INIT ERROR

005350  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
005356  012737  000101  001600          MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
005364  012737  000020  001616          MOV     #PAT,L.MR1      ;LOAD EVEN PARITY BIT

005372  104417                          TLOADRK                 ;LOAD RK REGS-SELECT W/EVEN PARITY
005374  104423                          TWAT16                  ;WAIT FOR INTERRUPT
005376  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

005400  104422                          TCHKWE                  ;CHECK OPERATION FOR EXPECTED ERROR
005402  000011                          DRPARERR!SPARERR        ;DRIVE SELECTED DRIVE BUS PARITY ERROR
005404  000000                          .WORD   0               ;CONTROLLER DETECTED DRIVE BUS PARITY ERROR
005406  000000                          .WORD   0
005410  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ANY DIFFERENCES

005412  012700  000400                  MOV     #BIT8,R0        ;ROUTINE TO DETERMINE WHICH BIT

005416  013701  001626                  MOV     DRVNUM,R1       ;SHOULD BE SET IN ASOF TO INDICATE
005422  001403                          BEQ     3$              ;DRIVE ATTENTION. R0 WILL HAVE THE
005424  006300                  2$:     ASL     R0              ;BIT THAT SHOULD BE SET FOR THE DRIVE
005426  005301                          DEC     R1              ;IN USE
005430  001375                          BNE     2$

005432  030037  001556          3$:     BIT     R0,T.ASOF       ;TEST ATTENTION SET
005436  001007                          BNE     4$              ;YES-SKIP
005440  012737  056540  001460          MOV     #EMDA,EM11N
005446  012737  054765  061022          MOV     #EMDPAR,DF011A
005454  104011                          ERROR   11              ;"DRV ATT NOT SET RESULT OF DRV PARITY ERR"
005456  032737  040000  001540  4$:     BIT     #DI,T.CS1       ;TEST DRIVE INTERRUPT SET
005464  001007                          BNE     5$              ;YES-SKIP
005466  012737  056474  001460          MOV     #EMDI,EM11N
005474  012737  054765  061022          MOV     #EMDPAR,DF011A
005502  104011                          ERROR   11              ;"DRV INT NOT SET RESULT OF DRV PARITY ERR"

005504  032737  040000  001552  5$:     BIT     #DSC,T.DS       ;TEST DRIVE STATUS CHANGE SET
005512  001007                          BNE     6$              ;YES-SKIP
005514  012737  056514  001460          MOV     #EMDSC,EM11N
005522  012737  054765  061022          MOV     #EMDPAR,DF011A
005530  104011                          ERROR   11              ;"DSC NOT SET RESULT OF DRV PARITY ERR"

005532  032737  100000  001552  6$:     BIT     #SVAL,T.DS      ;TEST STATUS VALID SET
005540  001007                          BNE     7$              ;YES-SKIP
005542  012737  056354  001460          MOV     #EMSVAL,EM11N
005550  012737  054765  061022          MOV     #EMDPAR,DF011A

005556  104011                          ERROR   11              ;"SVAL NOT SET RESULT OF DRV PAR ERR"

005560  005037  001616          7$:     CLR     L.MR1           ;CLEAR PAT IN MR1

005564  052737  100000  001600          BIS     #CCLR,L.CS1     ;CLEAR CONTROLLER
005572  104417                          TLOADRK                 ;LOAD RK REGS TO DO CLEAR
```

```
3511
3512
3513  005574  104421                          TCHKOP                  ;CHECK NO ERRORS SET
3514  005576  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS STILL SET
3515
3516  005600  030037  001556                  BIT     R0,T.ASOF       ;TEST ATTENTION STILL SET
3517  005604  001007                          BNE     8$              ;YES-SKIP
3518  005606  012737  056540  001510          MOV     #EMDA,EM14N
3519  005614  012737  056560  061022          MOV     #EMCCLR,DF011A
3520  005622  104014                          ERROR   14              ;''DRV ATT RESET RESULT OF CONT CLR''
3521
3522  005624  032737  040000  001540  8$:     BIT     #DI,T.CS1       ;TEST DRIVE INTERRUPT STILL SET
3523  005632  001007                          BNE     9$              ;YES-SKIP
3524  005634  012737  056474  001510          MOV     #EMDI,EM14N
3525  005642  012737  056560  061022          MOV     #EMCCLR,DF011A
3526  005650  104014                          ERROR   14              ;''DRV INT RESET RESULT OF CONTROLLER CLR''
3527
3528  005652  012737  000101  001600  9$:     MOV     #SELDRV,L.CS1   ;LOAD SELECT DRIVE
3529
3530  005660  104417                          TLOADRK                 ;LOAD RK REGS
3531  005662  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3532  005664  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3533
3534  005666  104422                          TCHKWE                  ;CHECK THAT ERRORS ARE SET
3535  005670  000010                          .WORD   000010          ;DPAR SET (NO SPAR 1, REPORT
3536  005672  000000                          .WORD   0               ;ANY DISCREPENCIES
3537  005674  000000                          .WORD   0
3538  005676  104004                          ERROR   4 ; OR 5,6,7
3539  005700  030037  001556                  BIT     R0,T.ASOF       ;TEST ATTENTION STILL SET.
3540  005704  001007                          BNE     10$             ;YES-SKIP
3541  005706  012737  056540  001460          MOV     #EMDA,EM11N
3542  005714  012737  054765  061022          MOV     #EMDPAR,DF011A
3543  005722  104011                          ERROR   11              ;''DRV ATT NOT SET RESULT OF DRV PARITY ERR''
3544
3545  005724  032737  040000  001540  10$:    BIT     #DI,T.CS1       ;TEST DRIVE INTERRUPT STILL SET
3546  005732  001007                          BNE     11$             ;YES-SKIP
3547  005734  012737  056474  001460          MOV     #EMDI,EM11N
3548  005742  012737  054765  061022          MOV     #EMDPAR,DF011A
3549  005750  104011                          ERROR   11              ;''DRV INT NOT SET RESULT OF DRV  PARITY ERR''
3550
3551  005752  032737  040000  001552  11$:    BIT     #DSC,T.DS       ;TEST STATUS CHANGE STILL SET
3552  005760  001007                          BNE     12$             ;YES-SKIP
3553  005762  012737  056514  001460          MOV     #EMDSC,EM11N
3554  005770  012737  054765  061022          MOV     #EMDPAR,DF011A
3555  005776  104011                          ERROR   11              ;''DSC NOT SET RESULT OF DRV PARITY ERR''
3556
3557  006000  032737  100000  001552  12$:    BIT     #SVAL,T.DS      ;TEST STATUS VALID STILL SET
3558  006006  001007                          BNE     13$             ;YES-SKIP
3559  006010  012737  056354  001460          MOV     #EMSVAL,EM11N
3560  006016  012737  054765  061022          MOV     #EMDPAR,DF011A
3561  006024  104011                          ERROR   11              ;''SVAL NOT SET RESULT OF DRV PARITY ERR''
3562
3563  006026  052737  100000  001600  13$:    BIS     #CCLR,L.CS1     ;CLEAR CONTROLLER
3564
3565  006034  104417                          TLOADRK                 ;LOAD RK REGS (DO CLEAR)
3566
```

```
3567
3568  006036  012701  000001            MOV     #1,R1           ;SET COUNTER TO SELECT STATUS PAIR
3569  006042  012737  000101  001600    MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
3570
3571  006050  010137  001616      14$:  MOV     R1,L.MR1        ;LOAD STATUS PAIR SELECTION
3572  006054  104417                    TLOADRK                 ;LOAD RK REGS
3573  006056  104423                    TWAT16                  ;WAIT FOR INTERRUPT
3574  006060  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3575
3576  006062  104421                    TCHKOP                  ;CHECK IF ANY ERRORS SET
3577  006064  104004                    ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS SET.
3578
3579  006066  030037  001556            BIT     R0,T.ASOF       ;TEST ATTENTION STILL SET
3580  006072  001007                    BNE     15$             ;YES-SKIP
3581  006074  012737  056540  001410    MOV     #EMDA,EM4N
3582  006102  012737  056371  061022    MOV     #EMNZPR,DF011A
3583  006110  104014                    ERROR   14              ;''ATTENTION RESET RESULT OF NON-0 PAIR SEL''
3584
3585  006112  032737  040000  001540  15$:  BIT     #DI,T.CS1
3586  006120  001007                    BNE     16$
3587  006122  012737  056474  001510    MOV     #EMDI,EM14N
3588  006130  012737  056371  061022    MOV     #EMNZPR,DF011A
3589  006136  104014                    ERROR   14              ;''DRV INT R' SET RESULT OF NON-0 PAIR SELECT''
3590
3591  006140  005201              16$:  INC     R1              ;BUMP PAIR SELECT
3592  006142  022701  000004            CMP     #4,R1           ;ALL PAIRS DONE?
3593  006146  001340                    BNE     14$             ;NO-LOOP
3594
3595  006150  005037  001616            CLR     L.MR1           ;CLEAR MR1
3596
3597  006154  012737  000105  001600    MOV     #CLEAR,L.CS1    ;LOAD DRIVE CLEAR
3598
3599  006162  104417                    TLOADRK                 ;DO DRIVE CLEAR
3600  006164  104423                    TWAT16                  ;WAIT FOR INTERRUPT
3601  006166  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3602
3603  006170  104421                    TCHKOP                  ;CHECK FOR ANY ERRORS
3604  006172  104004                    ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
3605
3606  006174  012701  000020            MOV     #20,R1          ;SET COUNT FOR SHORT WAIT
3607  006200  005301              17$:  DEC     R1              ;TO ALLOW CONTROLLER TIME TO POLL
3608  006202  001376                    BNE     17$             ;DRIVES
3609
3610  006204  104420                    TGETRK                  ;GET RK REGS
3611  006206  030037  001556            BIT     R0,T.ASOF       ;TEST ATTENTION RESET
3612  006212  001407                    BEQ     18$             ;YES-SKIP
3613  006214  012737  056540  001470    MOV     #EMDA,EM12N
3614  006222  012737  050206  061022    MOV     #EMDCLR,DF011A
3615  006230  104012                    ERROR   12              ;''ATTENTION NOT RESET RESULT OF DRV CLEAR
3616
3617  006232  032737  040000  001540  18$:  BIT     #DI,T.CS1       ;TEST DRIVE INTERRUPT RESET
3618  006240  001407                    BEQ     19$             ;YES-SKIP
3619  006242  012737  056474  001470    MOV     #EMDI,EM12N
3620  006250  012737  050206  061022    MOV     #EMDCLR,DF011A
3621  006256  104012                    ERROR   12              ;''DRV INT NOT RESET RESULT OF DRIVE CLR''
3622
```

```
3623  006260                          19$:
3624                                  ;;*****************************************************************
3625                                  ;*TEST 11       UNIT FIELD ERROR ON RELEASE
3626                                  ;*
3627                                  ;*       ISSUE A SUBSYSTEM CLEAR. SELECT AN AVAILABLE
3628                                  ;*       DRIVE.  PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE
3629                                  ;*       A RELEASE COMMAND.  CLOCK THROUGH PHASE ADDRESS 2.
3630                                  ;*       TURN OFF DIAGNOSTIC MODE.  MAKE SURE UNIT FIELD
3631                                  ;*       ERROR SETS.
3632                                  ;*
3633                                  ;;*****************************************************************
3634  006260  000004                  TST11:  SCOPE
3635  006262  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3636  006270  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
3637  006272  104002                          ERROR   2               ;BAD INIT ERROR
3638
3639  006274  013737  001626  001610          MOV     DRVNUM,L.CS2    ;SELECT A DRIVE
3640  006302  012737  000101  001600          MOV     #SELDRV,L.CS1   ;DO DRIVE SELECT
3641
3642  006310  104417                          TLOADRK                 ;LOAD RK
3643  006312  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3644  006314  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3645
3646  006316  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
3647  006320  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS.
3648
3649  006322  052737  000010  001610          BIS     #RLS,L.CS2      ;LOAD RELEASE
3650  006330  012737  000040  001616          MOV     #DMD,L.MR1      ;SET DIAGNOSTIC MODE
3651
3652  006336  104417                          TLOADRK                 ;LOAD RK
3653
3654  006340  004437  036316                  JSR     R4,MCLOCK       ;CALL MAINT CLOCK
3655  006344     023                          .BYTE   ^D19            ;NUMBER OF PHASES
3656  006345     002                          .BYTE   2               ;NUMBER OF CLOCK XISTIONS
3657
3658  006346  042762  000040  000026          BIC     #DMD,RKMR1(R2)  ;CLEAR DIAG MODE
3659
3660  006354  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3661  006356  104002                          ERROR   2               ;TO SLOW/NOT COMPLETED
3662
3663  006360  104422                          TCHKWE                  ;CHECK OPERATION WITH ERROR
3664  006362  000000                          .WORD   0
3665  006364  000000                          .WORD   0
3666  006366  000004                          .WORD   UFERR           ;UNIT FIELD ERROR
3667  006370  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ANY DISCREPENCIES
3668
3669  006372  104416                          TSSINIT                 ;CLEAR SUBSYSTEM TO INSURE UFE RESETS
3670  006374  104002                          ERROR   2
3671                                  ;;*****************************************************************
3672                                  ;*TEST 12       UNIT FIELD ERROR ON SELECT
3673                                  ;*
3674                                  ;*       ISSUE A SUBSYSTEM CLEAR.  SELECT AN AVAILABLE
3675                                  ;*       DRIVE.  PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE
3676                                  ;*       A SELECT COMMAND WITH MESSAGE ID = 3 AND DRIVE
3677                                  ;*       SELECTED = 0.  CLOCK THROUGH PHASE ADDRESS 6.
3678                                  ;*       TURN OFF DIAGNOSTIC MODE.  MAKE SURE UNIT FIELD
```

```
                                        ;*      ERROR SETS.
                                        ;*
                                        ;;*****************************************************************
006376  000004                          TST12:  SCOPE
006400  012737  000062  001262                  MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
006406  104416                          TSSINIT                         ;CLEAR SUBSYSTEM
006410  104003                          ERROR   3                       ;BAD INIT ERROR

006412  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
006420  012737  000101  001600          MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT

006426  104417                          TLOADRK                 ;LOAD RK
006430  104423                          TWAT16                  ;WAIT FOR INTERRUPT
006432  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE

006434  104421                          TCHKOP                  ;CHECK FOR ANY ERROR
006436  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS

006440  012737  000043  001616          MOV     #DMD!BIT1!BIT0,L.MR1    ;LOAD DIAG MODE & MSG PAIR 3
006446  005037  001610                  CLR     L.CS2           ;LOAD FOR DRIVE 0

006452  104417                          TLOADRK                 ;LOAD RK

006454  004437  036316                  JSR     R4,MCLOCK       ;CALL MAINTENANCE CLOCK
006460     026                          .BYTE   ^D22            ;THROUGH PHASE 6
006461     002                          .BYTE   2               ;PLUS 2 TRANSITIONS

006462  042762  000040  000026          BIC     #DMD,RKMR1(R2)  ;CLEAR DIAG MODE

006470  104423                          TWAT16                  ;WAIT FOR INTERRUPT
006472  104002                          ERROR   2               ;TO SLOW/NOT COMPLETED ERROR

006474  013737  036132  006504          MOV     HOLD3,1$        ;EXPECTED ERROR

006502  104422                          TCHKWE                  ;CHECK OPERATION WITH ERROR
006504  000000                  1$:     .WORD   0
006506  000000                          .WORD   0
006510  000004                          .WORD   UFERR           ;UNIT FIELD ERROR SHOULD SET
006512  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ANY DISCREPENCIES

                                .SBTTL  **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)
                                ;*
                                ;;*****************************************************************
006514  000004                  TST13:  SCOPE
```

```
3735  006516  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3736  006524  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
3737  006526  104003                          ERROR   3               ;BAD INIT ERROR
3738
3739  006530  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3740  006536  012737  000113  001600          MOV     #RECAL,L.CS1    ;LOAD RECAL
3741
3742  006544  104417                          TLOADRK                 ;LOAD RK
3743  006546  104423                          TWAT16                  ;WAIT FOR 1ST INTERRUPT
3744  006550  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
3745  006552  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
3746  006556  104420                          TGETRK                  ;GET RK REGS
3747  006560  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST SVAL SET
3748  006566  001010                          BNE     1$              ;YES-SKIP
3749  006570  012737  056354  001460          MOV     #EMSVAL,EM11N
3750  006576  012737  050247  061022          MOV     #EMRCAL,DF011A
3751  006604  104011                          ERROR   11              ;''SVAL NOT SET RESULT OF RECAL''
3752  006606  000463                          BR      50$             ;ABORT TEST
3753
3754  006610  104437                  1$      TWAT8S                  ;WAIT FOR INTERRUPT
3755  006612  000401                          BR      2$              ;NO INTERRUPT RETURN
3756  006614  000404                          BR      3$              ;INTERRUPT RETURN
3757
3758  006616  012737  055647  001370  2$:     MOV     #EM50,EM2N      ;ALTER MESSAGE ''NO 2ND INTERRUPT OR IT WAS LATE''
3759  006624  104002                          ERROR   2
3760
3761  006626  104420                  3$:     TGETRK                  ;GET RK REGS
3762  006630  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST SVAC SET NOW
3763  006636  001410                          BEQ     4$              ;NO-SKIP
3764  006640  012737  056354  001470          MOV     #EMSVAL,EM12N
3765  006646  012737  056601  061022          MOV     #EM2INT,DF011A
3766  006654  104012                          ERROR   12              ;''SVAL NOT RESET RESULT OF SECOND TEST''
3767  006656  000437                          BR      50$
3768
3769  006660  032737  040000  001552  4$:     BIT     #DSC,T.DS       ;TEST DSC SET BY ATTENTION
3770  006666  001010                          BNE     5$              ;YES-SKIP
3771  006670  012737  056514  001460          MOV     #EMDSC,EM11N
3772  006676  012737  056601  061022          MOV     #EM2INT,DF011A
3773  006704  104011                          ERROR   11              ;''DSC NOT SET RESULT OF SECOND INTERRUPT''
3774  006706  000423                          BR      50$
3775
3776  006710  012737  000101  001600  5$:     MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
3777
3778  006716  104417                          TLOADRK                 ;LOAD RK REGS
3779  006720  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3780  006722  104002                          ERROR   2       .       ;TO SLOW/NOT COMPLETE ERROR
3781
3782  006724  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
3783
3784  006726  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
3785
3786  006730  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST SVAC SET
3787  006736  001007                          BNE     50$             ;YES-SKIP
3788  006740  012737  056354  001460          MOV     #EMSVAL,EM11N
3789  006746  012737  050160  061022          MOV     #EMSELD,DF011A
3790  006754  104011                          ERROR   11              ;''SVAL NOT SET RESULT OF DRV SEL.
```

```
3791  006756                          50$:
3792                                  ;;*****************************************************************
3793                                  ;*TEST 14       SINGLE INTERRUPT FROM ATTENTION
3794                                  ;*
3795                                  ;*       DO A SEEK TO CYLINDER 0.  WAIT FOR INTERRUPT FROM
3796                                  ;*       DRIVE ATTENTION.  LOWER PRIORITY AGAIN AND MAKE
3797                                  ;*       SURE ANOTHER INTERRUPT DOES NOT OCCUR.  CLEAR DRIVE.
3798                                  ;*
3799                                  ;;*****************************************************************
3800  006756  000004                  TST14:  SCOPE
3801  006760  012737  000062  001262          MOV     #50.,$TIMES     ;;DO 50. ITERATIONS
3802  006766  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
3803  006770  104003                          ERROR   3               ;BAD INIT ERROR
3804
3805  006772  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3806  007000  012737  000117  001600          MOV     #SEEK,L.CS1     ;LOAD SEEK DCYL LEFT AT 0.
3807
3808  007006  104417                          TLOADRK                 ;LOAD RK REGS
3809  007010  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3810  007012  104002                          ERROR   2               ;TO SLOW/NOT COMPLETED ERROR
3811
3812
3813  007014  104420                          TGETRK                  ;GET RK REGS
3814
3815  007016  032737  040000  001540          BIT     #DI,T.CS1       ;TEST DI SET
3816  007024  001010                          BNE     2$              ;YES-SKIP
3817  007026  012737  056474  001460          MOV     #EMDI,EM11N
3818  007034  012737  056622  061022          MOV     #EMSKSF,DF011A
3819  007042  104011                          ERROR   11              ;"DI NOT SET RESULT OF SEEK TO SELF"
3820  007044  000417                          BR      50$
3821
3822
3823  007046  012700  000031          2$:     MOV     #25.,R0         ;LOAD AND DECREMENT A COUNT TO
3824  007052  005300                  3$:     DEC     R0              ;ZERO. GIVE CONTROLLER A CHANCE TO
3825  007054  001376                          BNE     3$              ;INTERRUPT AGAIN.  ERROR IF IT DOES.
3826
3827  007056  022737  000001  001662          CMP     #1,INTSET       ;CHECK ONLY ONE INTERRUPT OCCURRED
3828  007064  001407                          BEQ     50$             ;YES-SKIP
3829  007066  012737  056774  001500          MOV     #EMMI,EM13N
3830  007074  012737  056622  061022          MOV     #EMSKSF,DF011A
3831  007102  104013                          ERROR   13              ;"MULTIPLE INTERRUPTS RESULT OF SEEK TO SELF"
3832
3833  007104  104421                  50$:    TCHKOP                  ;CHECK FOR ANY ERRORS
3834  007106  104004                          ERROR   4  ;OR 5,6,7    ;REPORT ALL ERRORS
3835                                  ;;*****************************************************************
3836                                  ;*TEST 15       RESET ATTENTIONS WITH UNIBUS INIT
3837                                  ;*
3838                                  ;*       DO A SEEK TO CYLINDER 0 ON ALL AVAILIABLE DRIVES.
3839                                  ;*       ISSUE A RESET.  MAKE SURE ALL ATTENTION RESET.
3840                                  ;*
3841                                  ;;*****************************************************************
3842  007110  000004                  TST15:  SCOPE
3843  007112  012737  000005  001262          MOV     #5.,$TIMES      ;;DO 5. ITERATIONS
3844  007120  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
3845  007122  104003                          ERROR   3               ;BAD INIT ERROR
3846
```

```
3847  007124  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3848  007132  012737  000117  001600          MOV     #SEEK,L.CS1     ;LOAD SEEK (TO SELF-0)
3849
3850  007140  104417                          TLOADRK                 ;LOAD RK REGS
3851  007142  104423                          TWAT16                  ;WAIT FOR INTERRUPT
3852  007144  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
3853
3854  007146  104420                          TGETRK                  ;GET RK REGS
3855
3856  007150  032737  040000  001540          BIT     #DI,T.CS1       ;TEST DI SET
3857  007156  001010                          BNE     1$              ;YES-EXIT
3858  007160  012737  056474  001460          MOV     #EMDI,EM11N
3859  007166  012737  056622  061022          MOV     #EMSKSF,DF011A
3860  007174  104011                          ERROR   11              ;''DI NOT SET RESULT OF SEEK TO SELF
3861  007176  000450                          BR      50$
3862
3863  007200  005037  001662          1$:     CLR     INTSET          ;CLEAR INTERRUPT COUNTER
3864  007204  000005                          RESET                   ;DO UNIBUS RESET
3865  007206  004737  045146                  JSR     PC,$TKINT       ;RESET KEYBOARD INTERRUPT
3866
3867  007212  005037  001674                  CLR     LCLKTK          ;CLEAR TICK COUNTER
3868  007216  004737  035314          5$:     JSR     PC,MYTIME       ;CALL TIMER
3869  007222  022737  000012  001674          CMP     #10.,LCLKTK     ;COUNT 10 TICKS (MILLISECONDS)?
3870  007230  001372                          BNE     5$              ;NO - LOOP
3871
3872  007232  012762  000100  000000          MOV     #IE,RKCS1(R2)   ;SET IE FOR ANY STRAY INTERRUPTS
3873  007240  004737  034522                  JSR     PC,OPTTST       ;SET UP OPTIONS AGAIN
3874
3875  007244  104423                          TWAT16                  ;WAIT 16 MS FOR AN INTERRUPT
3876  007246  000410                          BR      2$              ;NONE IS EXPECTED SO RETURN SHOULD BE
3877                                                                  ;HERE-BR TO CONTINUE TEST.
3878  007250  012737  056637  001500          MOV     #EMUXIT,EM13N   ;INT OCCURRED ON RESET
3879  007256  012737  056701  061022          MOV     #EMRSET,DF011A
3880  007264  104013                          ERROR   13              ;''UNEXECUTED INTERRUPT RESULT OF RESET''
3881  007266  000414                          BR      50$
3882  007270  104420                  2$:     TGETRK                  ;GET RK REGS
3883  007272  032737  040000  001540          BIT     #DI,T.CS1       ;TEST DI RESET
3884  007300  001407                          BEQ     50$             ;YES-SKIP
3885  007302  012737  056474  001470          MOV     #EMDI,EM12N
3886  007310  012737  056701  061022          MOV     #EMRSET,DF011A
3887  007316  104012                          ERROR   12              ;''DI NOT RESET RESULT OF RESET''
3888
3889  007320                          50$:
3890
3891                                  .SBTTL  **ILLEGAL DISK ADDRESS ERROR TESTS
3892
3893                                  ;;*****************************************************************
3894                                  ;*TEST 16       ILLEGAL DISK ADDRESS (PART 1)
3895                                  ;*
3896                                  ;*      ISSUE A SEEK TO CYLINDER 0, HEAD 3. MAKE SURE
3897                                  ;*      ILLEGAL ADDRESS ERROR AND SEEK INCOMPLETE SETS.
3898                                  ;*      CLEAR CONTROLLER AND CLEAR DRIVE.  REPEAT FOR HEADS 4-7,
3899                                  ;*      CHECKING THAT BOTH IDAE AND SEEK INCOMPLETE SET FOR
3900                                  ;*      HEAD 7 AND IDAE ALONE SETS FOR HEADS 4, 5, AND 6.
3901                                  ;*
3902                                  ;;*****************************************************************
```

```
3903  007320  000004                    TST16:  SCOPE
3904  007322  012737  000012  001262            MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
3905  007330  012701  000003                    MOV     #3,R1           ;PRESET FOR SELECTING TRACK 3
3906
3907  007334  104416                            TSSINIT                 ;CLEAR SUBSYSTEM
3908  007336  104003                            ERROR   3
3909
3910  007340  012737  000113  001600            MOV     #RECAL,L.CS1    ;SET UP TO RECAL
3911  007346  013737  001626  001610            MOV     DRVNUM,L.CS2    ;LOAD DRIVE
3912
3913  007354  104417                            TLOADRK                 ;LOAD RK REGS
3914
3915  007356  104423                            TWAT16                  ;WAIT FOR 1ST INTERRUPT
3916  007360  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
3917
3918  007362  005037  001662                    CLR     INTSET          ;CLEAR INTERRUPT FLAG
3919
3920  007366  104437                            TWAT8S                  ;WAIT FOR INTERRUPT
3921  007370  104002                            ERROR   2
3922
3923  007372  012737  007400  001110            MOV     #1$,$LPERR      ;SET LOCAL LOOP ON ERROR
3924
3925  007400  104416                    1$:     TSSINIT                 ;CLEAR SUBSYSTEM
3926  007402  104003                            ERROR   3               ;BAD INIT ERROR
3927
3928  007404  013737  001626  001610            MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
3929  007412  012737  000117  001600            MOV     #SEEK,L.CS1     ;LOAD SEEK
3930  007420  110137  001607                    MOVB    R1,L.DT         ;LOAD TRACK
3931
3932  007424  104417                            TLOADRK                 ;LOAD RK REGS
3933  007426  104423                            TWAT16                  ;WAIT FOR INTERRUPT
3934  007430  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE
3935
3936  007432  032701  000001                    BIT     #BIT0,R1        ;TEST IF HEAD ADDRESS HAS BIT 0
3937  007436  001403                            BEQ     2$              ;NO - SKIP
3938  007440  032701  000002                    BIT     #BIT1,R1        ;TEST IF HEAD ADDRESS HAS BOTH 0 AND 1
3939  007444  001007                            BNE     3$              ;YES-GO CHECK BOTH IDAE AND SKI SET
3940
3941  007446  104422                    2$:     TCHKWE                  ;CHECK OPERATION WITH ERROR
3942  007450  002000                            IDAERR                  ;ILLEGAL DISK ADDRESS ERROR
3943  007452  000000                            0
3944  007454  000000                            0
3945  007456  104004                            ERROR   4 ; OR 5,6,7    ;REPORT ALL DISCREPANCIES
3946  007460  104415                            SCOP1                   ;LOCAL LOOP ON ERROR
3947  007462  000406                            BR      4$
3948
3949  007464  104422                    3$:     TCHKWE                  ;CHECK OPERATION WITH ERROR
3950  007466  002002                            IDAERR!SKIERR           ;ILLEGAL DISK ADDRESS ERROR
3951  007470  000000                            0                       ;SEEK INCOMPLETE
3952  007472  000000                            0
3953  007474  104004                            ERROR   4  ;OR 5,6,7    ;REPORT ANY DISCREPANCIES
3954  007476  104415                            SCOP1                   ;LOCAL LOOP ON ERROR TO 1$
3955
3956  007500  005201                    4$:     INC     R1              ;ELSE BUMP TO NEXT ILLEGAL TRACK
3957  007502  022701  000010                    CMP     #8.,R1          ;ALL ILLEGAL TRACKS SELECTED?
3958  007506  001334                            BNE     1$              ;NO-LOOP
```

```
                                ;;*****************************************************************
                                ;*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
                                ;*      THIS TEST IS BYPASSED FOR THE RK07 BECAUSE THE RK07
                                ;*      CONTROLLER DOES NOT RECOGNIZE THE ILLEGAL DISK ADDRESS
                                ;*
                                ;;*****************************************************************
007510  000004                TST17:  SCOPE
007512  012737  000012  001262        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
007520  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
007522  104003                        ERROR   3               ;BAD INIT ERROR

007524  012737  000113  001600        MOV     #RECAL,L.CS1    ;LOAD RECALIBRATE
007532  013737  001626  001610        MOV     DRVNUM,L.CS2    ;LOAD DRIVE

007540  104417                        TLOADRK                 ;LOAD RK REGS

007542  104423                        TWAT16                  ;WAIT FOR 1ST INTERRUPT
007544  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

007546  005037  001662                CLR     INTSET          ;CLEAR INTERRUPT FLAG

007552  104437                        TWAT8S                  ;WAIT FOR INTERRUPT
007554  104002                        ERROR   2

007556  012737  007564  001110        MOV     #1$,$LPERR      ;SET LOOP TO BYPASS RECAL

007564  104416                1$:     TSSINIT                 ;CLEAR SUBSYSTEM
007566  104003                        ERROR   3

007570  013737  001626  001610        MOV     DRVNUM, L.CS2   ;LOAD DRIVE NUMBER
007576  012737  000117  001600        MOV     #SEEK,L.CS1     ;LOAD SEEK
007604  005737  001720                TST     DTYPE           ;SEE IF RK06
007610  001402                        BEQ     2$              ;BR IF YES
                                ;       MOV     #1777,L.DCYL    ;ELSE LOAD RK07 ILL CYL
                                ;       MOV     #<IDAERR!SKIERR>,4$     ;EXP RK07 ERR
                                ;       BR      3$
007612  000137  007644                JMP     TST20           ;RK07 CONTROLLER
                                                                ;DOES NOT RECOGNIZE
                                                                ;ILLEGAL DISK ADDRESS
                                                                ;REV 006 THE ABOVE THREE LINES

007616  012737  001000  001614  2$:     MOV     #1000,L.DCYL    ;LOAD DK06 ILL CYL
                                ;       MOV     #IDAERR,4$      ;EXP RK06 ERR

007624  104417                3$:     TLOADRK                 ;LOAD RK REGS
007626  104423                        TWAT16                  ;WAIT FOR INTERRUPT
007630  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

007632  104422                        TCHKWE                  ;CHECK OPERATION WITH ERROR
007634  002000                4$:     .WORD   IDAERR          ;DISK ADDRESS ERROR
007636  000000                        .WORD   0
007640  000000                        .WORD   0
007642  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.
                                ;*
                                ;*      CYLINDER 1456 IS USED FOR THE RK07.
                                ;*
                                ;*      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.
                                ;*
                                ;*
                                ;;*****************************************************************
007644  000004                  TST20:  SCOPE
007646  012737  000001  001262          MOV     #1,$TIMES       ;;DO 1 ITERATION
                                ;
                                ;       TSSINIT                 ;CLEAR SUBSYS FROM LAST IDAERR
                                ;       ERROR   3               ;BAD INIT ERROR
                                ;
                                ;       MOV     #RECAL,L.CS1    ;DO RECAL CMD
                                ;       MOV     DRVNUM,L.CS2
                                ;
                                ;       TLOADRK                 ;LOAD "L" REGS INTO RK REGS
                                ;       TWAT16
                                ;       ERROR   2               ;NO INTR
                                ;
                                ;       CLR     INTSET          ;CLR INIT FLAG
                                ;
                                ;       TWAT8S                  ;WAIT FOR 2'N INTR
                                ;       ERROR   2               ;NOT THERE
                                ;
                                ;
007654  105037  007722                  CLRB    2$              ;CLEAR SECTOR POINTER
007660  005000                          CLR     R0
007662  005005                          CLR     R5              ;CLEAR R5 FOR BAD SECTOR COUNTING
007664  013703  001640                  MOV     BSF26P,R3       ;SET POINT IN TO STORE BS 26 SECT FORMAT
007670  012737  007676  001110          MOV     #1$,$LPERR      ;SET ERROR RETURN ADDRESS FOR INTERNAL LOOP
007676  104416                  1$:     TSSINIT                 ;CLEAR SUBSYSTEM
007700  104003                          ERROR   3               ;BAD INIT ERROR

007702  013737  036120  007724          MOV     LSTCYL,15$
```

```
 4071  007710  004437  035674                JSR     R4,LRLOAD       ;LOAD "L" REGS WITH
 4072  007714  000121                        RDDATA                  ; READ DATA
 4073  007716  177400                        -400                    ; WORD COUNT
 4074  007720  070414                        IBUFF                   ; BUFFER ADDRESS
 4075  007722     000              2$:      .BYTE   0               ; SECTOR ADDRESS
 4076  007723     002                       .BYTE   2               ; TRACK ADDRESS
 4077  007724  000000              15$:     0                       ; CYLINDER ADDRESS 632/1456
 4078
 4079  007726  104417                        TLOADRK                 ;LOAD "L" REGS INTO RK
 4080  007730  104431                        TWAT112                 ;WAIT FOR INTERRUPT
 4081  007732  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 4082  007734  104421                        TCHKOP                  ;CHECK FOR ANY ERRORS
 4083  007736  104004                        ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
 4084  007740  104415                        SCOP1                   ;LOOP TO 1$ IF SW 9 SET
 4085
 4086  007742  105737  001103                TSTB    $ERFLG          ;TEST FOR ERROR IN OPERATION
 4087  007746  001502                        BEQ     7$              ;NO-SKIP
 4088  007750  005700                        TST     R0              ;GETTING A BS FACTORY SECTOR 26 SECT FORMAT?
 4089  007752  001023                        BNE     3$              ;NO-SKIP
 4090  007754  062737  000002  007722        ADD     #2,2$           ;NEXT SECTOR ADDRESS
 4091  007762  122737  000012  007722        CMPB    #10.,2$         ;PAST APPLICABLE SECTORS?
 4092  007770  001066                        BNE     6$              ;NO-SKIP
 4093  007772  012737  055746  001360        MOV     #EM51,EM1N
 4094  010000  104001                        ERROR   1               ;"CANNOT READ BS FILES
 4095  010002  043737  001630  001354  25$:  BIC     DRVBIT,$DEVM    ;CLEAR DRIVE FROM DRIVE MAP
 4096  010010  053737  001630  001666        BIS     DRVBIT,DRVDRP   ;SE1 DRIVE DROPPED
 4097  010016  000137  027716                JMP     NEWDRV          ;ABORT TEST PASS.
 4098
 4099  010022  022700  000001          3$:   CMP     #1,R0           ;GETTING A BS SOFT SECTOR 26 SECT FORMAT?
 4100  010026  001014                        BNE     4$              ;NO-SKIP
 4101  010030  062737  000002  007722        ADD     #2,2$           ;NEXT SECTOR ADDRESS
 4102  010036  122737  000026  007722        CMPB    #22.,2$         ;PAST APPLICABLE SECTORS?
 4103  010044  001040                        BNE     6$              ;NO-SKIP
 4104  010046  012737  055746  001360        MOV     #EM51,EM1N
 4105  010054  104001                        ERROR   1               ;"CANNOT READ BS FILES"
 4106  010056  000751                        BR      25$
 4107
 4108  010060  022700  000002          4$:   CMP     #2,R0           ;GETTING A BS FACT SECTOR 24 SECTOR FORMAT?
 4109  010064  001014                        BNE     5$              ;NO-SKIP
 4110  010066  062737  000002  007722        ADD     #2,2$           ;NEXT SECTOR ADDRESS
 4111  010074  122737  000013  007722        CMPB    #11.,2$         ;PAST APPLICABLE SECTORS?
 4112  010102  001021                        BNE     6$              ;NO-SKIP
 4113
 4114  010104  012737  055746  001360        MOV     #EM51,EM1N
 4115  010112  104001                        ERROR   1               ;"CANNOT READ BS FILES"
 4116  010114  000732                        BR      25$
 4117
 4118  010116  062737  000002  007722  5$:   ADD     #2,2$           ;NEXT SECTOR (BS SOFT 24 SECT MODE)
 4119  010124  122737  000027  007722        CMPB    #23.,2$         ;PAST APPLICABLE SECTORS?
 4120  010132  001005                        BNE     6$              ;NO-SKIP
 4121  010134  012737  055746  001360        MOV     #EM51,EM1N
 4122  010142  104001                        ERROR   1               ;"CANNOT READ BS FILES
 4123  010144  000716                        BR      25$
 4124
 4125  010146  105037  001103          6$:   CLRB    $ERFLG          ;CLEAR ERROR FLAG
 4126  010152  000651                        BR      1$              ;DO NEXT READ
```

```
 4127
 4128  010154  005737  070422          7$:     TST     IBUFF+6         ;CHECK FOR ALIGNMENT PACK
 4129  010160  001405                          BEQ     8$              ;NO-SKIP
 4130  010162  012737  056041  001360          MOV     #EM52,EM1N
 4131  010170  104001                          ERROR   1               ;"ALIGNMENT PACK. DRIVE ABORTING"
 4132  010172  000703                          BR      25$
 4133
 4134  010174  012701  070424          8$:     MOV     #IBUFF+10,R1    ;SET TO START OF BAD SECTOR DATA
 4135
 4136  010200  022711  177777          9$:     CMP     #-1,(R1)        ;TEST IF WORD ALL ONES (END OF DATA)
 4137  010204  001422                          BEQ     11$             ;YES-SKIP
 4138  010206  012123                          MOV     (R1)+,(R3)+     ;STORE CYLINDER
 4139  010210  012123                          MOV     (R1)+,(R3)+     ;      TRACK AND SECTOR
 4140  010212  005205                          INC     R5              ;BUMP ERROR COUNTER
 4141  010214  022705  000025                  CMP     #21.,R5         ;DOES IT TOTAL 20 FOR THIS FORMAT?
 4142  010220  001367                          BNE     9$              ;NO-TEST AND MORE NEXT ADDRESS
 4143  010222  012737  056137  001360          MOV     #EM53,EM1N
 4144  010230  104001                          ERROR   1               ;TO MANY BAD SECTORS
 4145  010232  043737  001630  001354  10$:    BIC     DRVBIT,$DEVM    ;CLEAR DRIVE FROM TESTING
 4146  010240  053737  001630  001666          BIS     DRVBIT,DRVDRP   ;SET DRIVE DROPPED
 4147  010246  000137  027716                  JMP     NEWDRV          ;ABORT PASS
 4148
 4149  010252  005200                  11$:    INC     R0              ;BUMP TO NEXT
 4150  010254  022700  000001                  CMP     #1,R0           ;NOW TESTING BS SOFT 26 SECTOR MODE?
 4151  010260  001011                          BNE     12$             ;NO-SKIP
 4152  010262  012723  177777                  MOV     #-1,(R3)+       ;INSERT END OF FIELD FLAG
 4153  010266  010337  001644                  MOV     R3,BSS26P       ;SET POINTER TO BAD SECTOR SOFTWARE FIELD
 4154  010272  112737  000012  007722          MOVB    #12,2$          ;SET TO FIRST SECTOR THIS MODE
 4155  010300  000137  007676                  JMP     1$              ;GO READ IT.
 4156  010304  022700  000002          12$:    CMP     #2,R0           ;NOW TESTING BS FACT 24 SECTOR MODE?
 4157  010310  001014                          BNE     13$             ;NO-SKIP
 4158  010312  012723  177777                  MOV     #-1,(R3)+       ;INSERT END OF FIELD FLAG
 4159  010316  112737  000001  007722          MOVB    #1,2$           ;SET TO FIRST SECTOR THIS MODE
 4160  010324  010537  001646                  MOV     R5,BS26CT       ;STORE TOTAL BS COUNT 26 SECTOR MODE
 4161  010330  005005                          CLR     R5              ;CLEAR COUNTER FOR COUNTING 24 SECT BS
 4162  010332  013703  001636                  MOV     BSF24P,R3       ;SET POINTER FOR STORING BS
 4163  010336  000137  007676                  JMP     1$              ;GO READ
 4164
 4165  010342  022700  000003          13$:    CMP     #3,R0           ;NOW TESTING BS SOFT 24 SECTOR MODE
 4166  010346  001011                          BNE     14$             ;NO-SKIP
 4167  010350  012723  177777                  MOV     #-1,(R3)+       ;INSERT END OF FIELD FLAG
 4168  010354  010337  001642                  MOV     R3,BSS24P       ;STORE POINTER TO BSS 24 SECTOR MODE
 4169  010360  112737  000013  007722          MOVB    #13,2$          ;GET START OF FIELDS BSS 24 SECT MODE
 4170  010366  000137  007676                  JMP     1$              ;GO READ IT
 4171
 4172  010372  012723  177777          14$:    MOV     #-1,(R3)+       ;INSERT END OF FIELD FLAG
 4173  010376  010537  001650                  MOV     R5,BS24CT       ;STORE COUNT BSS 24 SECTOR MODE
 4174
 4175
 4176  010402  012700  061054                  MOV     #BS26,R0        ;GET START OF BAD SECTOR BUFFER
 4177                                  ;       MOV     #BS24+52.,R3    ;GET END OF BUFFER
 4178                                  ;
 4179                                  ;27$:   CMP     #312,(R0)+      ;TEST IF ANY SECTORS BAD IN CYL 312
 4180                                  ;       BEQ     26$             ;YES - GET OUT
 4181                                  ;
 4182                                  ;       CMP     R0,R3           ;CHECK IF ALL OF BUFFER TESTED
```

```
4183                                        ;       BNE     27$             ;NO - LOOP
4184                                        ;       BR      28$             ;EXIT
4185  010406  005003                                CLR     R3              ;COUNT
4186  010410  022710  177777                27$:    CMP     #-1,(R0)        ;AT THE END OF TABLE ?
4187  010414  001404                                BEQ     16$             ;BRANCH IF SO
4188  010416  022720  000312                        CMP     #312,(R0)+      ;CYL 312 BAD ?
4189  010422  001372                                BNE     27$             ;BRANCH IF NOT TRY AGAIN
4190  010424  000424                                BR      26$             ;ELSE ERROR
4191  010426  005203                            16$:INC     R3              ;
4192  010430  020327  000001                        CMP     R3,#1           ;SEE IF 1 ST HALF OF BS 26 TABLE
4193  010434  001003                                BNE     17$             ;BRANCH IF NOT
4194  010436  013700  001644                        MOV     BSS26P,R0       ;MOVE UP THE TABLE
4195  010442  000762                                BR      27$             ;
4196  010444  020327  000002                17$:    CMP     R3,#2           ;SEE IF DID 2 ND HALF OF BS26 TABLE
4197  010450  001003                                BNE     18$             ;BRANCH IF NOT
4198  010452  012700  061200                        MOV     #BS24,R0        ;ELSE MOVE UP 1 ST HALF
4199  010456  000754                                BR      27$             ;
4200  010460  020327  000003                18$:    CMP     R3,#3           ;SEE IF DID 1 ST HALF OF BS24 TABLE
4201  010464  001003                                BNE     19$             ;BRANCH IF NOT
4202  010466  013700  001642                        MOV     BSS24P,R0       ;ELSE MOVE UP 2ND HALF OF BS24
4203  010472  000746                                BR      27$
4204  010474  000406                            19$:BR      28$             ;ALL DONE
4205
4206  010476  012737  057020  001360      26$:    MOV     #DRVABT,EM1N    ;''BAD SECTOR IN AREA FOR TEST''
4207  010504  104001                                ERROR   1
4208  010506  000137  010002                        JMP     25$
4209
4210  010512                                    28$:
4211
4212                                        ;;**********************************************************************
4213                                        ;*TEST 21       FORMAT IN 26 SECTOR FORMAT
4214                                        ;*
4215                                        ;*      FORMAT CYLINDER 312, TRACK 0 AND TRACK 1 FOR 26 SECTOR
4216                                        ;*      FORMAT.  VERIFY FORMAT AND THAT DATA LATE DID NOT
4217                                        ;*      OCCUR WITH WRITE HEADER OR IN READING DATA
4218                                        ;*      BUFFER AFTER READ HEADER.
4219                                        ;*
4220                                        ;;**********************************************************************
4221  010512  000004                        TST21:  SCOPE
4222  010514  012737  000012  001262                MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
4223  010522  012737  000312  001676                MOV     #312,REFMT      ;SET REFORMAT SWITCH
4224  010530  105037  010653                        CLRB    10$             ;CLEAR SECTOR POINTER
4225  010534  105037  010557                        CLRB    11$             ;CLEAR TRACK POINTER
4226  010540  104416                                TSSINIT                 ;CLEAR SUBSYSTEM
4227  010542  104003                                ERROR   3               ;BAD INIT ERROR
4228
4229  010544  004437  035674                9$:     JSR     R4,LRLOAD       ;LOAD ''L'' REGS
4230  010550  000127                                WRHEAD                  ;WRITE HEADER
4231  010552  177676                                -1C2                    ;WORD CNT FOR 26 SECTOR MODE
4232  010554  072414                                OBUFF                   ;BUFFER
4233  010556     000                                .BYTE   0               ;SECTOR 0
4234  010557     000                            11$:.BYTE   0               ;TRACK 0
4235  010560  000312                                312                     ;CYL 0
4236
4237  010562  004437  042260                        JSR     R4,GENFOM       ;GENERATE DATA
4238  010566  000600                                000600                  ;BUILD HEADERS-NO BAD SECTORS
```

```
4239  010570  012737  010600  001110          MOV     #111$,$LPERR    ;SET LOCAL LOOP ON ERROR
4240  010576  000402                          BR      1$              ;SKIP INIT
4241  010600                          111$:
4242  010600  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
4243  010602  104003                          ERROR   3               ;BAD INIT ERROR
4244  010604  104417                  1$:     TLOADRK                 ;LOAD RK REGS
4245  010606  104431                          TWAT112                 ;WAIT FOR INTERRUPT
4246  010610  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4247
4248  010612  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
4249  010614  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
4250
4251  010616  104415                          SCOP1                   ;INTERNAL LOOP TO 111$
4252
4253  010620  012737  010660  001110          MOV     #112$,$LPERR    ;SET LOCAL LOOP ON ERROR
4254  010626  010203                          MOV     R2,R3           ;BUILD POINTER TO RKDB
4255  010630  062703  000024                  ADD     #RKDB,R3
4256  010634  012701  070414                  MOV     #IBUFF,R1       ;SET POINTER TO BUFFER
4257  010640  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
4258  010644  000125                          RDHEAD                  ;READ HEADER
4259  010646  000000                          0                       ;NO WORDS COUNT
4260  010650  000000                          0                       ;NO BUFFER
4261  010652     000                          .BYTE   0               ;SECTOR 0
4262  010653     000                  10$:    .BYTE   0               ;  TRACK 0
4263  010654  000312                          312                     ;  CYL 312
4264
4265  010656  000402                          BR      2$              ;SKIP INIT
4266  010660                          112$:
4267  010660  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
4268  010662  104003                          ERROR   3               ;BAD INIT ERROR
4269  010664  104417                  2$:     TLOADRK                 ;LOAD RK REGS
4270  010666  104423                          TWAT16                  ;WAIT FOR INTERRUPT
4271  010670  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4272
4273  010672  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
4274  010674  104004                          ERROR   4; OR 5,o,7     ;REPORT ALL ERRORS
4275  010676  012700  000003                  MOV     #3,R0           ;SET COUNT
4276  010702  011321                  5$:     MOV     (R3),(R1)+      ;GET RKDB
4277  010704  104420                          TGETRK                  ;GET RK REGS
4278  010706  032737  100000  001550          BIT     #DLT,T.CS2      ;TEST IF DATA LATE
4279  010714  001410                          BEQ     3$              ;NO-SKIP
4280  010716  012737  054541  054170          MOV     #EMDLT,EM13
4281  010724  012737  056707  061022          MOV     #EMRDB,DF011A
4282  010732  104013                          ERROR   13              ;"DATA LATE SET RESULT OF DB READ
4283  010734  104415                          SCOP1                   ;LOCAL LOOP TO 112$
4284
4285  010736  032737  100000  001540  3$:     BIT     #CERR,T.CS1     ;TEST IF CONT ERROR SET
4286  010744  001410                          BEQ     4$              ;NO-SKIP
4287  010746  012737  056733  001500          MOV     #EMCERR,EM13N
4288  010754  012737  056707  061022          MOV     #EMRDB,DF011A
4289  010762  104013                          ERROR   13              ;"CERR SET RESULT OF READ DB
4290  010764  104415                          SCOP1                   ;LOCAL LOOP TO 112$
4291  010766  005300                  4$:     DEC     R0              ;DEC COUNT
4292  010770  001344                          BNE     5$              ;LOOP IF NOT ZERO
4293  010772  012737  011002  001110          MOV     #117$,$LPERR    ;SET LOCAL LOOP 117$
4294  011000  000402                          BR      7$              ;SKIP INIT
```

```
4295  011002                          117$:
4296  011002  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
4297  011004  104003                  ERROR   3               ;BAD INIT ERROR
4298  011006  004437  036362    7$:   JSR     R4,RDSTHD       ;GO READ & SEQUENCE HEADERS
4299  011012  104421                  TCHKOP                  ;CONTROLLER ERROR RETURN
4300  011014  104004                  ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
4301  011016  104013                  ERROR   13              ;"DATA LATE SET RESULT OF DATA BUFFER READ"
4302  011020  104002                  ERROR   2               ;"OPERATION TO SLOW" MESSAGE
4303                                                          ;OR "HEADER 0 NOT FOUND" MESSAGE
4304
4305  011022  004437  042260          JSR     R4,GENCOM
4306  011026  100200                  100200                  ;COMPARE IBUF & OBUFF (HEADERS)
4307  011030  000414                  BR      6$              ;GOOD RETURN-NO MISCOMPARES
4308  011032  104015                  ERROR   15              ;REPORT 1ST MSCOMPARES
4309
4310  011034  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
4311  011040  005300          12$:    DEC     R0              ;DECREMENT IT
4312  011042  001407                  BEQ     6$              ;EXIT IF ZERO
4313  011044  004437  042260          JSR     R4,GENCOM
4314
4315  011050  040000                  040000                  ;RESUME COMPARE
4316  011052  000403                  BR      6$              ;GOOD RETURN-NO MORE ERRORS
4317  011054  104016                  ERROR   16              ;REPORT NEXT ERROR LINE
4318  011056  000770                  BR      12$             ;LOOP
4319  011060  104415                  SCOP1                   ;LOCAL ERROR LOOP TO 117$
4320
4321  011062  105737  001607    6$:   TSTB    L.DT            ;WAS TRACK 1 JUST DONE?
4322  011066  001010                  BNE     8$              ;YES-SKIP
4323
4324  011070  112737  000001  010557  MOVB    #1,11$          ;CHANGE PARAM TO LOAD "L" WITH
4325  011076  112737  000001  010653  MOVB    #1,10$          ;TRACK 2
4326  011104  000137  010544          JMP     9$              ;JUMP TO DO ENTIRE TEST ON TRK 1
4327
4328  011110                    8$:
4329
4330                            .SBTTL  **HEADER RECOGNITION TESTS
4331
4332                            ;;*****************************************************************
4333                            ;*TEST 22       BAD SECTOR ERROR
4334                            ;*
4335                            ;*      FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE
4336                            ;*      SECTOR 0 (BIT 15 OR WORD 2 OF HEADER RESET) AND SECTOR 1
4337                            ;*      (BIT 14 OR WORD 2 OF HEADER RESET) TO BE BAD SECTORS
4338                            ;*      AND ALL OTHER SECTORS GOOD.
4339                            ;*
4340                            ;*      ISSUE A WRITE DATA OR 400 WORDS TO CYLINDER 312, TRACK 0,
4341                            ;*      SECTOR 0.  MAKE SURE BAD SECTOR ERROR SETS.  ISSUE A
4342                            ;*      WRITE DATA TO CYLINDER 0, TRACK 0, SECTOR 1 OF 400 WORDS.
4343                            ;*      MAKE SURE BAD SECTOR ERROR SET.  ISSUE A WRITE DATA
4344                            ;*      OF 400 WORDS TO CYLINDER 0, TRACK 0, SECTOR 2.  MAKE
4345                            ;*      SURE NO ERROR SETS.
4346                            ;*
4347                            ;;*****************************************************************
4348  011110  000004            TST22:  SCOPE
4349  011112  012737  000012  001262  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
4350  011120  012737  000312  001676  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 SECTOR MODE
                                        OBUFF                   ;BUFFER ADDRESS
                                        .BYTE   0               ;SECTOR
                                        .BYTE   0               ;TRACK
                                        312                     ;CYLINDER


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

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

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

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

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

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

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

                                        SCOP1                   ;LOCAL ERROR LOOP TO 4$

                                        CMPB    #2,5$           ;WAS SECTOR SET TO 2
                                        BEQ     7$              ;YES-SKIP
```

```
4407  011310  105237  011242            INCB    5$              ;BUMP TO NEXT SECTOR
4408  011314  000743                    BR      4$              ;LOOP
4409
4410  011316  104421            6$:     TCHKOP                  ;CHECK FOR GOOD OPERATION
4411  011320  104004                    ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
4412
4413  011322                    7$:
4414                                    ;;*****************************************************************
4415                                    ;*TEST 23       HEADER VRC ERROR
4416                                    ;*
4417                                    ;*      FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE
4418                                    ;*      16 SECTORS WITH BAD HEADER VRC.  ISSUE A WRITE DATA
4419                                    ;*      OF EACH OF THE SECTORS WITH A BAD HEADER VRC.  MAKE
4420                                    ;*      SURE HEADER VRC ERROR SETS.  ISSUE A WRITE DATA TO
4421                                    ;*      A GOOD HEADER AND MAKE SURE NO ERROR OCCURS.
4422                                    ;*
4423                                    ;;*****************************************************************
4424  011322  000004            TST23:  SCOPE
4425  011324  012737  000012  001262    MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
4426  011332  012737  000312  001676    MOV     #312,REFMT      ;SET REFORMAT SWITCH
4427  011340  104416                    TSSINIT                 ;CLEAR SUBSYSTEM
4428  011342  104003                    ERROR   3               ;BAD INIT ERROR
4429
4430  011344  004437  035674            JSR     R4,LRLOAD       ;LOAD "L" REGS
4431  011350  000127                    WRHEAD                  ;WRITE HEADER
4432  011352  177676                    -102                    ;WORD COUNT
4433  011354  072414                    OBUFF                   ;BUFF ADD
4434  011356     000                    .BYTE   0               ;SECT
4435  011357     000                    .BYTE   0               ;TRACK
4436  011360  000312                    312                     ;CYL
4437
4438  011362  004437  042260            JSR     R4,GENCOM
4439  011366  000600                    600                     ;BUILD HEADERS NO BSE
4440
4441  011370  012700  072420            MOV     #OBUFF+4,R0     ;GET ADDRESS OF VRC HDR0
4442  011374  012703  000001            MOV     #BIT0,R3        ;SET FOR BIT CHANGE SELECT
4443  011400  030310            1$:     BIT     R3,(R0)         ;CHECK A VRC BIT
4444  011402  001402                    BEQ     2$              ;SKIP IF ZERO
4445  011404  040310                    BIC     R3,(R0)         ;ELSE CLEAR IT
4446  011406  000401                    BR      3$              ;SKIP
4447  011410  050310            2$:     BIS     R3,(R0)         ;IF ZERO SET IT
4448  011412  062700  000006    3$:     ADD     #6,R0           ;BUMP TO NEXT VRC WORD
4449
4450  011416  006303                    ASL     R3              ;SHIFT THE SELECT
4451  011420  001367                    BNE     1$              ;IF BIT NOT SHIFTED OUT-LOOP
4452
4453  011422  104417                    TLOADRK                 ;LOAD RK REGS
4454  011424  104431                    TWAT112                 ;WAIT FOR INTERRUPT
4455  011426  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4456  011430  104421                    TCHKOP                  ;CHECK OPERATION COMPLETE
4457  011432  104004                    ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
4458
4459  011434  012737  011442  001110    MOV     #4$,$LPERR      ;SET LOCAL LOOP
4460  011442  104416            4$:     TSSINIT                 ;CLEAR SUBSYSTEM
4461  011444  104003                    ERROR   3               ;BAD INIT ERROR
4462
```

```
4463  011446  004437  035674              JSR     R4,LRLOAD       ;LOAD "L" REGS
4464  011452  000123                      WRDATA                  ;WRITE DATA
4465  011454  177400                      -400                    ;WORD COUNT
4466  011456  072414                      OBUFF                   ;BUFFER ADD
4467  011460     000          5$:         .BYTE   0               ;SECT
4468  011461     000                      .BYTE   0               ;TRACK
4469  011462  000312                      312                     ;CYL
4470
4471  011464  104417                      TLOADRK                 ;LOAD RK REG
4472  011466  104424                      TWAT32                  ;WAIT FOR INTERRUPT
4473  011470  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4474
4475  011472  022737  000020  011460      CMP     #16.,5$         ;WAS THIS WRITE SECTOR 16?
4476  011500  001415                      BEQ     6$              ;YES-SKIP
4477                                                              ;ELSE
4478  011502  104422                      TCHKWE                  ;CHECK OPERATION WITH ERROR
4479  011504  000000                      0
4480  011506  000040                      40                      ;HVRC  EM EXPECTED
4481  011510  000000                      0
4482  011512  104004                      ERROR   4 ; OR 5,6,7    ;REPORT ANY DISCREPENCIES
4483
4484  011514  104415                      SCOP1                   ;LOCAL LOOP TO 4$
4485
4486  011516  105237  011460              INCB    5$              ;BUMP SECTOR IN "L" REG
4487  011522  022737  000016  011460      CMP     #16,5$          ;IF SECTOR IS 16 OR LESS
4488  011530  003744                      BLE     4$              ;LOOP
4489  011532  000402                      BR      7$              ;ELSE EXIT
4490  011534  104421              6$.     TCHKOP                  ;CHECK LAST OPERATION NO ERRORS
4491  011536  104004                      ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
4492
4493  011540                      7$:
4494                                      ;;*****************************************************************
4495                                      ;*TEST 24       BAD SECTOR ERROR AND HVRC ERROR
4496                                      ;*
4497                                      ;*      FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR ZERO HAS
4498                                      ;*      BOTH A BAD SECTOR ERROR AND HEADER VRC.  ISSUE A WRITE DATA
4499                                      ;*      TO CYLINDER 0, TRACK 0, SECTOR 0.  MAKE SURE ONLY HEADER VRC
4500                                      ;*      ERROR SETS.
4501                                      ;*
4502                                      ;;*****************************************************************
4503  011540  000004              TST24:  SCOPE
4504  011542  012737  000012  001262      MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
4505  011550  012737  000312  001676      MOV     #312,REFMT      ;SET REFORMAT SWITCH
4506  011556  104416                      TSSINIT                 ;CLEAR SUBSYSTEM
4507  011560  104003                      ERROR   3               ;BAD INIT ERROR
4508
4509  011562  004437  035674              JSR     R4,LRLOAD       ;LOAD "L" REG
4510  011566  000127                      WRHEAD                  ;WRITE HEADER
4511  011570  177676                      -102                    ;WORD CNT FOR 26 SECTOR MODE
4512  011572  072414                      OBUFF                   :BUFF ADD
4513  011574     000                      .BYTE   0               ;SECTOR
4514  011575     000                      .BYTE   0               ;TRACK
4515  011576  000312                      312                     ;CYLINDER
4516
4517  011600  004437  042260              JSR     R4,GENCOM
4518  011604  000600                      600                     ;BUILD HEADERS-NO BSE
```

```
4519
4520  011606  042737  100000  072416          BIC     #BIT15,OBUFF+2  ;CLEAR BIT TO SET BSE,LEAVE VRC BAD.
4521
4522  011614  104417                          TLOADRK                 ;LOAD RK REGS
4523  011616  104431                          TWAT112                 ;WAIT FOR INTERRUPT
4524  011620  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4525
4526  011622  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
4527  011624  104004                          ERROR   4 , OR 5,6,7    ;REPORT ALL ERRORS
4528
4529  011626  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
4530  011632  000123                          WRDATA                  ;WRITE DATA
4531  011634  177400                          -400                    ;WORD COUNT
4532  011636  072414                          OBUFF                   ;BUFF ADD
4533  011640     000                          .BYTE   0               ;SECTOR
4534  011641     000                          .BYTE   0               ;TRACK
4535  011642  000312                          312                     ;CYLINDER
4536
4537  011644  104417                          TLOADRK                 ;LOAD RK REGS
4538  011646  104424                          TWAT32                  ;WAIT FOR INTERRUPT
4539  011650  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4540
4541  011652  104422                          TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERR
4542  011654  000000                          0
4543  011656  000040                          40                      ;HVRC ERR EXPECTED
4544  011660  000000                          0
4545  011662  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL DISCREPENCIES
4546
4547                          ;;*****************************************************************
4548                          ;*TEST 25       OPERATION INCOMPLETE
4549                          ;*
4550                          ;*      FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 21 HAS THE
4551                          ;*      WRONG FORMAT.  ISSUE A WRITE DATA OF 400 TO CYLINER 0,
4552                          ;*      TRACK 0, SECTOR 21.  MAKE SURE OPI SET.
4553                          ;*
4554                          ;;*****************************************************************
4555  011664  000004          TST25:  SCOPE
4556  011666  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
4557  011674  012737  000312  001676          MOV     #312,REFMT      ;SET REFORMAT SWITCH
4558  011702  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
4559  011704  104003                          ERROR   3               ;BAD INIT ERROR
4560
4561  011706  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
4562  011712  000127                          WRHEAD                  ;WRITE HEADER
4563  011714  177676                          -102                    ;WORD COUNT FOR 26 SECT MODE
4564  011716  072414                          OBUFF                   ;BUFF ADD
4565  011720     000                          .BYTE   0               ;SECTOR
4566  011721     000                          .BYTE   0               ;TRACK
4567  011722  000312                          312                     ;CYLINDER
4568
4569  011724  004437  042260                  JSR     R4,GENCOM
4570  011730  000600                          600                     ;BUILD HEADERS-NO BSE ERRORS
4571
4572  011732  052737  001000  072614          BIS     #BIT9,OBUFF+200 ;CHANGE FORMAT IN SECTOR 25
4573  011740  052737  001000  072616          BIS     #BIT9,OBUFF+202 ;CORRECT THE VRC
4574
```

```
4575  011746  104417                          TLOADRK                 ;LOAD RK REGS
4576  011750  104431                          TWAT112                 ;WAIT FOR INTERRUPT
4577  011752  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
4578
4579  011754  104421                          TCHKOP                  ;CHECK FOR ANY ERRORS
4580  011756  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
4581
4582  011760  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
4583  011764  000123                          WRDATA                  ;WRITE DATA
4584  011766  177400                          -400                    ;400 WORDS
4585  011770  072414                          OBUFF                   ;BUFF ADD
4586  011772     025                          .BYTE   25              ;SECTOR 25
4587  011773     000                          .BYTE   0               ;TRACK 0
4588  011774  000312                          312                     ;CYL 312
4589
4590  011776  104417                          TLOADRK                 ;LOAD RK REGS
4591  012000  104425                          TWAT48                  ;WAIT FOR INTERRUPT
4592  012002  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE
4593
4594  012004  104422                          TCHKWE                  ;CHECK OPERATION EXPECTED ERROR
4595  012006  000000                          0
4596  012010  000020                          20                      ;OPI EXPECTED
4597  012012  000000                          0
4598  012014  104004                          ERROR   4 ; OR 5,6,7    ;REPORT ANY DISCREPENCIES
4599                                ;;*****************************************************************
4600                                ;*TEST 26       OPI WITH HVRC ERROR
4601                                ;*
4602                                ;*      FORMAT CYLINDER 312, TRACK 0 SUCH THAT A HEADER VRC
4603                                ;*      ERROR IS PRESENT AND SECTOR 17 HAS THE WRONG FORMAT.
4604                                ;*      ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312,
4605                                ;*      TRACK 0, SECTOR 17.  THAT BOTH OPERATION INCOMPLETE
4606                                ;*      AND HEADER VRC SET.
4607                                ;*
4608                                ;;*****************************************************************
4609  012016  000004                TST26:  SCOPE
4610  012020  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
4611  012026  012737  000312  001676          MOV     #312,REFMT      ;SET REFORMAT SWITCH
4612  012034  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
4613  012036  104003                          ERROR   3               ;BAD INIT ERROR
4614
4615  012040  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
4616  012044  000127                          WRHEAD                  ;WRITE HEADER
4617  012046  177676                          -102                    ;WORD COUNT FOR 26 SECT MODE
4618  012050  072414                          OBUFF                   ;BUS ADDRESS
4619  012052     000                          .BYTE   0               ;SECTOR
4620  012053     000                          .BYTE   0               ;TRACK
4621  012054  000312                          312                     ;CYLINDER
4622
4623  012056  004437  042260                  JSR     R4,GENCOM
4624  012062  000600                          600                     ;BUILD HEADER- NO BSE ERRORS
4625
4626  012064  012700  072550                  MOV     #OBUFF+134,R0   ;GET ADDRESS 2ND WORD HDR 17(8)
4627  012070  052720  001000                  BIS     #BIT9,(R0)+     ;SET FORMAT 24 SECT PER TRACK
4628  012074  052720  001000                  BIS     #BIT9,(R0)+     ;SET VRC BIT
4629  012100  062700  000004                  ADD     #4,R0           ;BUMP TO HVRC WORD HDR 20(8)
4630  012104  032710  000001                  BIT     #BIT0,(R0)      ;TEST BIT 0
```

```
4631  012110  001403                    BEQ     1$                ;RESET-SKIP
4632  012112  042710  000001            BIC     #BIT0,(R0)        ;CLEAR BIT
4633  012116  000402                    BR      2$
4634  012120  052710  000001    1$:     BIS     #BIT0,(R0)        ;SET BIT
4635                                                              ;FORCE OPI AND HVRC ERROR
4636  012124  104417            2$:     TLOADRK                   ;LOAD RK REGS
4637  012126  104431                    TWAT112                   ;WAIT FOR INTERRUPT
4638  012130  104002                    ERROR   2                 ;TO SLOW/NOT COMPLETE ERROR
4639
4640  012132  104421                    TCHKOP                    ;CHECK FOR ANY ERRORS
4641  012134  104004                    ERROR   4 ; OR 5,6,7      ;YES-REPORT ALL ERRORS
4642
4643  012136  004437  035674            JSR     R4,LRLOAD         ;LOAD "L" REGS
4644  012142  000123                    WRDATA                    ;WRITE DATA
4645  012144  177400                    -400                      ;400 WORDS
4646  012146  072414                    OBUFF                     ;BUFF ADDRESS
4647  012150     017                    .BYTE   17                ;SECT 17
4648  012151     000                    .BYTE   0                 ;TRACK 0
4649  012152  000312                    312                       ;CYLINDER 312
4650
4651  012154  104417                    TLOADRK                   ;LOAD RK REGS
4652  012156  104425                    TWAT48                    ;WAIT FOR INTERRUPT
4653  012160  104002                    ERROR   2                 ;TO SLOW/NOT COMPLETE
4654
4655  012162  104422                    TCHKWE                    ;CHECK WITH EXPECTED ERROR
4656  012164  000000                    0
4657  012166  000060                    60                        ;HVRC ERR & OPI EXPECTED
4658  012170  000000                    0
4659  012172  104004                    ERROR   4  ;OR 5,6,7
4660                            ;;*****************************************************************
4661                            ;*TEST 27       HVRC IGNORE ON NON-ADDRESSED SECTOR
4662                            ;*
4663                            ;*      FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 20 HAS AN HVRC
4664                            ;*      ERROR. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK 0,
4665                            ;*      AND SECTOR 21. MAKE SURE HVRC IS NOT SET AT THE
4666                            ;*      END OF THE OPERATION
4667                            ;*
4668                            ;;*****************************************************************
4669  012174  000004            TST27.  SCOPE
4670  012176  012737  000012  001262    MOV     #10.,$TIMES       ;;DO 10. ITERATIONS
4671  012204  012737  000312  001676    MOV     #312,REFMT        ;SET REFORMAT SWITCH
4672
4673  012212  104416                    TSSINIT                   ;CLEAR SUBSYSTEM
4674  012214  104003                    ERROR   3                 ;BAD INIT ERROR
4675
4676  012216  004437  035674            JSR     P4,LRLOAD         ;LOAD "L" REGISTERS
4677  012222  000127                    WRHEAD                    ;WRITE HEADER
4678  012224  177676                    -102                      ;WORD COUNT FOR 26 SECTOR MODE
4679  012226  072414                    OBUFF                     ;BUFF ADD
4680  012230     000                    .BYTE   0                 ;SECTOR
4681  012231     000                    .BYTE   0                 ;TRACK
4682  012232  000312                    312                       ;CYLINDER
4683
4684  012234  004437  042260            JSR     R4,GENCOM
4685  012240  000600                    600                       ;BUILD HEADERS-NO BSE ERRORS
4686
```

```
                                        MOV     #OBUFF+144,R0   ;ADDRESS OF HEAD 20 HVRC WORD
                                        MOV     #BIT1,R1        ;BIT 1 CONSTANT
                                        BIT     R1,(R0)         ;TEST BIT 1 SET
                                        BEQ     1$              ;RESET-SKIP
                                        BIC     R1,(R0)         ;ELSE CLEAR BIT 1
                                        BR      2$              ;SKIP
                                1$:     BIS     R1,(R0)         ;SET BIT 1

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

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

                                        JSR     R4,LRLOAD       ;LOAD "L" REGISTER
                                        WRDATA                  ;WRITE DATA
                                        -400                    ;WORD COUNT
                                        OBUFF                   ;BUFF ADD
                                        .BYTE   21              ;SECTOR
                                        .BYTE   0               ;TRACK
                                        312                     ;CYLINDER
                                        TLOADRK                 ;LOAD RK REGS
                                        TWAT32                  ;WAIT FOR INTERRUPT
                                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

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


                        .SBTTL  **DATA TRANSFER TESTS

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

                                MOV     #312,7$         ;PRESET CYL POINTER
                                CLRB    2$              ;CLEAR TRACK POINTER

                        1$:     JSR     R4,LRLOAD       ;LOAD "L" REG
                                WRHEAD                  ;WRITE HEADER
                                -102                    ;WORD COUNT FOR 26 SECTOR MODE
                                OBUFF                   ;BUFF ADDRESS
                                .BYTE   0               ;SECTOR
                        2$:     .BYTE   0               ;TRACK
```

```
4743  012376  000312          7$:   312                     ;CYLINDER
4744
4745  012400  004437  042260        JSR     R4,GENCOM
4746  012404  001200              1200                      ;BUILD HDRS-INCLUDE BAD SECTORS
4747
4748  012406  104417              TLOADRK                   ;LOAD RK REGS
4749  012410  104431              TWAT112                   ;WAIT FOR INTERRUPT
4750  012412  104002              ERROR   2                 ;TO SLOW/NOT COMPLETE ERROR
4751
4752  012414  104421              TCHKOP                    ;CHECK FOR ANY ERRORS
4753  012416  104004              ERROR   4 ;.OR 5,6,7      ;REPORT ALL ERRORS
4754
4755  012420  022737  000313  012376    CMP     #313,7$         ;TEST IF DONE 313 TK 0
4756  012426  001414              BEQ     3$              ;YES - SKIP
4757  012430  123727  012375  000002    CMPB    2$,#2           ;DID WE JUST FORMAT TRACK 2
4758  012436  001403              BEQ     8$              ;YES-SKIP
4759  012440  105237  012375          INCB    2$              ;BUMP TO NEXT TRACK
4760  012444  000746              BR      1$              ;GO FORMAT NEXT TRACK
4761
4762  012446  105037  012375    8$:   CLRB    2$              ;CLEAR TRACK POINTER
4763  012452  005237  012376          INC     7$              ;BUMP CYL TO 313
4764  012456  000741              BR      1$              ;GO FORMAT 313 TK 0
4765
4766  012460  004437  035674    3$:   JSR     R4,LRLOAD       ;LOAD "L" REGS
4767  012464  000123              WRDATA                  ;WRITE DATA
4768  012466  177400              -400                    ;ONE SECTOR WORD COUNT
4769  012470  072414              OBUFF                   ;BUFF ADDRESS
4770  012472     012              .BYTE   12              ;SECTOR 12
4771  012473     000              .BYTE   0               ;TRACK 0
4772  012474  000312              312                     ;CYLINDER 312
4773
4774  012476  004437  042260        JSR     R4,GENCOM
4775  012502  000001              1                       ;BUILD DATA PATTERN 1
4776  012504  000400              400                     ;400 WORDS LONG
4777  012506  012737  012514  001110    MOV     #4$,$LPERR      ;SET FOR LOCAL LOOP
4778  012514  104417          4$:   TLOADRK                 ;LOAD RK REGS
4779  012516  104431              TWAT112                 ;WAIT FOR INTERRUPT
4780  012520  104002              ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4781
4782  012522  104421              TCHKOP                  ;CHECK FOR ANY ERRORS
4783  012524  104004              ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
4784
4785  012526  004437  035674        JSR     R4,LRLOAD       ;LOAD "L" REGS
4786  012532  000121              RDDATA                  ;READ DATA
4787  012534  177400              -400                    ;400 WORDS
4788  012536  070414              IBUFF                   ;BUFF ADD
4789  012540     012              .BYTE   12              ;SECTOR 12
4790  012541     000              .BYTE   0               ;TRACK 0
4791  012542  000312              312                     ;CYL 312
4792
4793  012544  104417              TLOADRK                 ;LOAD RK
4794  012546  104424              TWAT32                  ;WAIT FOR INTERRUPT
4795  012550  104002              ERROR   2               ;TO SLOW/NOT COMPLETE
4796
4797  012552  104421              TCHKOP                  ;CHECK FOR ANY ERRORS
4798  012554  104004              ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
```

```
012556  004437  042260                  JSR     R4,GENCOM
012562  100001                          100001                  ;GO COMPARE DATA TO PATTERN 1
012564  000400                          400                     ;400 WORDS LONG
012566  000413                          BR      6$              ;GOOD RETURN-NO DATA ERRORS
012570  104015                          ERROR   15              ;ERROR RETURN

012572  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
012576  005300                  5$:     DEC     R0              ;DEC LIMIT
012600  001406                          BEQ     6$              ;EXIT IF 0
012602  004437  042260                  JSR     R4,GENCOM
012606  040000                          040000                  ;CONTINUE COMPARE
012610  000402                          BR      6$              ;EXIT IF NO MORE ERRORS
012612  104016                          ERROR   16              ;ELSE REPORT MISCOMPARE
012614  000770                          BR      5$              ;LOOP
012616  005037  001676          6$:     CLR     REFMT           ;CLEAR REFORMAT SWITCH

                                ;;*************************************************************
                                ;*TEST 31       WRITE DATA WITH BUS ADDRESS INCREMENT INHIBIT
                                ;*
                                ;*      ISSUE A WRITE DATA OF ONE SECTOR TO CYLINDER 312,
                                ;*      TRACK 2, SECTOR 12 WITH INHIBIT BUS
                                ;*      ADDRESS INCREMENT. READ DATA BACK TO MAKE SURE
                                ;*      EVERY WORD IS THE SAME AND CORRECT.
                                ;*
                                ;;*************************************************************
012622  000004                  TST31:  SCOPE
012624  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS

012632  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
012634  104003                          ERROR   3               ;BAD INIT ERROR

012636  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
012642  000123                          WRDATA                  ;WRDATA
012644  177400                          -400                    ;-400 WORDS
012646  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
012650     012                          .BYTE   12              ;SECTOR 12
012651     001                          .BYTE   1               ;TRACK 1
012652  000312                          312                     ;CYLINDER 312

012654  052737  000020  001610          BIS     #BAI,L.CS2      ;SET INCREMENT INHIBIT
012662  004437  042260                  JSR     R4,GENCOM       ;BUILD PATTERN
012666  000016                          16                      ;PATTERN 16
012670  000400                          400                     ;400 WORDS

012672  104417                          TLOADRK                 ;LOAD RK REGS
012674  104430                          TWAT96                  ;WAIT FOR INTERRUPT
012676  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

012700  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
012702  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

012704  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
012710  000121                          RDDATA                  ;RDDATA
012712  177400                          -400                    ;-400 WORDS
012714  070414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
```

```
4855  012716  012                       .BYTE   12              ;SECTOR 12
4856  012717  001                       .BYTE   1               ;TRACK 1
4857  012720  000312                    312                     ;CYLINDER 312
4858
4859  012722  012700  000377            MOV     #377,R0         ;SET COUNT TO SET OBUFF TO BE
4860  012726  012701  072416            MOV     #OBUFF+2,R1     ;ALL THE FIRST WORD OF PATTERN
4861  012732  012703  072414            MOV     #OBUFF,R3
4862
4863  012736  011321              1$:     MOV     (R3),(R1)+      ;MOV THE WORD
4864  012740  005300                      DEC     R0
4865  012742  001375                      BNE     1$              ;LOOP UNTIL ALL WORDS SET
4866
4867  012744  104417                      TLOADRK                 ;LOAD RK REGS
4868  012746  104424                      TWAT32                  ;WAIT FOR INTERRUPT
4869  012750  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
4870
4871  012752  104421                      TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
4872  012754  104004                      ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4873
4874  012756  004437  042260              JSR     R4,GENCOM       ;COMPARE THE DATA
4875  012762  100000                      100000
4876  012764  000400                      400
4877  012766  000413                      BR      2$
4878  012770  104015                      ERROR   15
4879  012772  013700  001634              MOV     ERRLMT,R0       ;GET ERROR LIMIT
4880  012776  005300              64$:    DEC     R0              ;DECREMENT COUNT
4881  013000  001406                      BEQ     65$             ;IF ZERO - EXIT
4882  013002  004437  042260              JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
4883  013006  040000                      40000
4884  013010  000402                      BR      65$             ;NO MORE ERRORS - EXIT
4885  013012  104016                      ERROR   16              ;REPORT NEXT ERROR
4886  013014  000770                      BR      64$             ;LOOP
4887  013016                      65$:
4888
4889  013016                      2$:
4890                                      ;;*****************************************************************
4891                                      ;*TEST 32       WRITE DATA ADDRESS GREATER THAN 32K
4892                                      ;*
4893                                      ;*      ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 177770.
4894                                      ;*      MAKE SURE CORRECT DATA IS ON DISK.
4895                                      ;*
4896                                      ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF MORE THAN 32K
4897                                      ;*              OF MEMORY IS PRESENT.
4898                                      ;*
4899                                      ;*****************************************************************
4900  013016  000004              TST32:  SCOPE
4901  013020  012737  000012  001262      MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
4902  013026  123727  001326  000001      CMPB    $MAMS1,#1       ;TEST IF >32K MEM
4903  013034  002016                      BGE     2$              ;YES-SKIP
4904
4905  013036  032737  000002  001664      BIT     #MEMSZB,OPTFLG  ;TEST IF REPORT ALREADY MADE
4906  013044  001011                      BNE     1$              ;YES -SKIP
4907
4908  013046  104401  051503              TYPE    ,OPR011         ;''INSUFFICIENT MEMORY DATA TRANSFER WITH
4909  013052  104401  051646              TYPE    ,OPR012         ;ADDRESS >32K
4910  013056  052737  000002  001664      BIS     #MEMSZB,OPTFLG  ;SET FLAG
```

```
                                                TYPE    ,OPR015         ;BYPASSED''
                                        1$:     BR      4$              ;EXIT
                                        2$:     MOV     #5$,$LPERR      ;SET LOCAL LOOP ON ERROR ADDRESS
                                        5$:
                                                TSSINIT                 ;CLEAR SUBSYSTEM
                                                ERROR   3               ;BAD INIT ERROR

                                                JSR     R4,LRLOAD       ;LOAD "L" REGS
                                                WRDATA                  ;WRITE DATA
                                                -400                    ;400 WORDS
                                                177770                  ;BUS ADDRESS IN 32K -10 BYTES
                                                .BYTE   16              ;SECTOR 16
                                                .BYTE   0               ;TRACK 0
                                                312                     ;CYLINDER 312
                                                JSR     R4,GENCOM       ;GENERATE DATA
                                                10000                   ;PATTERN 10, MEM. MANAGEMENT FOR DEST.
                                                1777                    ;RELOCATION ARGUMENT
                                                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
                                                SCOP1                   ;LOCAL LOOP ON ERROR TO 5$

                                                JSR     R4,GENCOM       ;CLEAR IBUFF TO 1'S.
                                                2007
                                                1000

                                                JSR     R4,LRLOAD       ;LOAD "L" REGS
                                                RDDATA                  ;RDDATA
                                                -400                    ;-400 WORDS
                                                IBUFF                   ;IBUFF IS BUFF ADDRESS
                                                .BYTE   16              ;SECTOR 16
                                                .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
                                                JSR     R4,GENCOM       ;COMPARE
                                                 10000                  ;MEMORY MANAGEMENT FOR DESTINATION
                                                1777                    ;RELOCATION ARGUMENT
                                                400                     ;400 WORDS
                                                BR      4$              ;NO ERROR-SKIP
                                                ERROR   15              ;REPORT FIRST MISCOMPARE
                                                MOV     ERRLMT,R0       ;GET ERROR LIMIT
                                        64$:    DEC     R0              ;DECREMENT COUNT
                                                BEQ     65$             ;IF ZERO - EXIT
                                                JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
                                                50000
                                                BR      65$             ;NO MORE ERRORS - EXIT
                                                ERROR   16              ;REPORT NEXT ERROR
```

```
 4967  013250  000770                        BR      64$             ;LOOP
 4968  013252                        65$:
 4969
 4970  013252                        4$:
 4971                                ;;*****************************************************************
 4972                                ;*TEST 33       READ DATA ADDRESS GREATER THAN 32K
 4973                                ;*
 4974                                ;*      ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 177770.
 4975                                ;*      CHECK MEMORY FOR CORRECT TRANSFER.
 4976                                ;*
 4977                                ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF MORE THAN 32K
 4978                                ;*              OF MEMORY IS PRESENT.
 4979                                ;*
 4980                                ;;*****************************************************************
 4981  013252  000004                TST33:  SCOPE
 4982  013254  012737  000012  001262        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
 4983  013262  123727  001326  000001        CMPB    $MAMS1,#1       ;CHECK IF >32K MEMORY
 4984  013270  002001                        BGE     2$              ;YES-SKIP
 4985
 4986  013272  000462                1$:     BR      5$              ;EXIT
 4987
 4988  013274  012737  013302  001110  2$:  MOV     #3$,$LPERR      ;SET LOCAL ERROR LOOP
 4989
 4990  013302                        3$:
 4991  013302  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
 4992  013304  104003                        ERROR   3               ;BAD INIT ERROR
 4993  013306  004437  035674                JSR     R4,LRLOAD       ;LOAD "L" REGS
 4994  013312  000123                        WRDATA                  ;WRDATA
 4995  013314  177400                        -400                    ;-400 WORDS
 4996  013316  072414                        OBUFF                   ;OBUFF IS BUFF ADDRESS
 4997  013320     017                        .BYTE   17              ;SECTOR 17
 4998  013321     000                        .BYTE   0               ;TRACK 0
 4999  013322  000312                        312                     ;CYLINDER 312
 5000  013324  004437  042260                JSR     R4,GENCOM       ;GENERATE DATA IN OBUFF
 5001  013330  000011                        11                      ;PATTERN 11
 5002  013332  000400                        400                     ;400 WORDS
 5003
 5004  013334  104417                        TLOADRK                 ;LOAD RK REGS
 5005  013336  104430                        TWAT96                  ;WAIT FOR INTERRUPT
 5006  013340  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 5007
 5008  013342  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 5009  013344  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
 5010  013346  004437  035674                JSR     R4,LRLOAD       ;LOAD "L" REG
 5011  013352  000121                        RDDATA                  ;READ DATA
 5012  013354  177400                        -400                    ;400 WORDS
 5013  013356  177770                        177770                  ;ACCROSS 32K BOUNDARY
 5014  013360     017                        .BYTE   17              ;SECTOR 17
 5015  013361     000                        .BYTE   0               ;TRACK 0
 5016  013362  000312                        312                     ;CYL 312
 5017
 5018  013364  104417                        TLOADRK                 ;LOAD RK REGS
 5019  013366  104424                        TWAT32                  ;WAIT FOR INTERRUPT
 5020  013370  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 5021  013372  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 5022  013374  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
```

```
5023  013376  004437  042260              JSR     R4,GENCOM       ;COMPARE DATA
5024  013402  120000                      120000                  ;MEMORY MANAGEMENT WITH SOURCE
5025  013404  001777                      1777                    ;RELOCATION ARGUMENT
5026  013406  000400                      400                     ;COMPARE 400 WORDS
5027  013410  000413              -       BR      5$              ;NO MISCOMPARE-EXIT
5028  013412  104015                      ERROR   15              ;REPORT FIRST MISCOMPARE
5029  013414  013700  001634              MOV     ERRLMT,R0       ;GET ERROR LIMIT
5030  013420  005300              64$:    DEC     R0              ;DECREMENT,COUNT
5031  013422  001406                      BEQ     65$             ;IF ZERO - EXIT
5032  013424  004437  042260              JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5033  013430  060000                      60000
5034  013432  000402                      BR      65$             ;NO MORE ERRORS - EXIT
5035  013434  104016                      ERROR   16              ;REPORT NEXT ERROR
5036  013436  000770                      BR      64$             ;LOOP
5037  013440                      65$:
5038  013440                      5$:
5039                              ;;*****************************************************************
5040                              ;*TEST 34       WRITE DATA ADDRESS GREATER THAN 64K
5041                              ;*
5042                              ;*      ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 377770.
5043                              ;*      MAKE SURE CORRECT DATA IS ON DISK.
5044                              ;*
5045                              ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF MORE THAN 64K
5046                              ;*              OF MEMORY IS PRESENT.
5047                              ;*
5048                              ;;*****************************************************************
5049  013440  000004              TST34:  SCOPE
5050  013442  012737  000012  001262      MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5051  013450  123727  001326  000002      CMPB    $MAMS1,#2       ;CHECK IF >64K MEMORY
5052  013456  002016                      BGE     2$              ;YES-SKIP
5053  013460  032737  000002  001664  6$: BIT     #MEMSZB,OPTFLG  ;TEST IF REPORT FLAG SET
5054  013466  001011                      BNE     1$              ;NO-SKIP
5055
5056  013470  104401  051503              TYPE    ,OPR011         ;''INSUFFICIENT MEMORY-DATA XFER WITH
5057  013474  104401  051652              TYPE    ,OPR013         ;ADDRESS >64K
5058  013500  104401  051662              TYPE    ,OPR015         ;BYPASSED''
5059  013504  052737  000002  001664      BIS     #MEMSZB,OPTFLG  ;SET FLAG
5060  013512  000467              1$:     BR      5$
5061
5062  013514  012737  013522  001110  2$: MOV     #3$,$LPERR      ;SET LOCAL LOOP ON ERROR
5063
5064  013522                      3$:
5065  013522  104416                      TSSINIT                 ;CLEAR SUBSYSTEM
5066  013524  104003                      ERROR   3               ;BAD INIT ERROR
5067  013526  004437  042260              JSR     R4,GENCOM       ;GENERATE DATA, PATTERN 11
5068  013532  010011                      10011                   ;MEM MANAGEMENT ON DESTINATION
5069  013534  003777                      3777                    ;RELOCATION ARGUMENT
5070  013536  000400                      400                     ;400 WORDS
5071
5072  013540  004437  035674              JSR     R4,LRLOAD       ;LOAD ''L'' REGS
5073  013544  000523                      WRDATA!BA16             ;WRITE DATA AND SET BA16
5074  013546  177400                      -400                    ;400 WORDS
5075  013550  177770                      177770                  ;ACCROSS 64K BOUNDARY
5076  013552     020                      .BYTE   20              ;SECTOR 20
5077  013553     000                      .BYTE   0               ;TRACK 0
5078  013554  000312                      312                     ;CYLINDER 312
```

```
5079
5080  013556  104417                          TLOADRK                 ;LOAD RK REGS
5081  013560  104430                          TWAT96                  ;WAIT FOR INTERRUPT
5082  013562  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5083
5084  013564  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5085  013566  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5086  013570  004437  042260                  JSR     R4,GENCOM       ;CLEAR IBUFF TO 1'S
5087  013574  002007                          2007
5088  013576  001000                          1000
5089
5090  013600  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
5091  013604  000121                          RDDATA                  ;RDDATA
5092  013606  177400                          -400                    ;-400 WORDS
5093  013610  070414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
5094  013612     020                          .BYTE   20              ;SECTOR 20
5095  013613     000                          .BYTE   0               ;TRACK 0
5096  013614  000312                          312                     ;CYLINDER 312
5097  013616  104417                          TLOADRK                 ;LOAD RK REGS
5098  013620  104424                          TWAT32                  ;WAIT FOR INTERRUPT
5099  013622  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5100
5101  013624  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5102  013626  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5103  013630  004437  042260                  JSR     R4,GENCOM       ;CHECK DATA
5104  013634  110000                          110000                  ;MEMORY MANAGEMENT WITH DESTINATION
5105  013636  003777                          3777                    ;RELOCATION ARGUMENT
5106  013640  000400                          400                     ;400 WORDS
5107  013642  000413                          BR      5$              ;NO MISCOMPARES-SKIP
5108  013644  104015                          ERROR   15              ;REPORT FIRST ERROR
5109
5110  013646  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
5111  013652  005300                  64$:    DEC     R0              ;DECREMENT COUNT
5112  013654  001406                          BEQ     65$             ;IF ZERO - EXIT
5113  013656  004437  042260                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5114  013662  050000                          50000
5115  013664  000402                          BR      65$             ;NO MORE ERRORS - EXIT
5116  013666  104016                          ERROR   16              ;REPORT NEXT ERROR
5117  013670  000770                          BR      64$             ;LOOP
5118  013672                          65$:
5119  013672                          5$:
5120                                  ;;*****************************************************************
5121                                  ;*TEST 35       READ DATA ADDRESS GREATER THAN 64K
5122                                  ;*
5123                                  ;*      ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 377770.
5124                                  ;*      CHECK MEMORY FOR CORRECT TRANSFER.
5125                                  ;*
5126                                  ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF MORE THAN 64K
5127                                  ;*              OF MEMORY IS PRESENT.
5128                                  ;*
5129                                  ;;*****************************************************************
5130  013672  000004                  TST35:  SCOPE
5131  013674  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5132  013702  123727  001326  000002          CMPB    $MAMS1,#2       ;CHECK IF >64K MEMORY
5133  013710  002001                          BGE     2$              ;YES-SKIP
5134  013712  000462                  1$:     BR      5$              ;EXIT
```

```
5135
5136  013714  012737  000032  001110  2$:     MOV     #32,$LPERR      ;SET LOCAL LOOP ON ERROR
5137
5138  013722                          3$:
5139  013722  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
5140  013724  104003                          ERROR   3               ;BAD INIT ERROR
5141  013726  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
5142  013732  000123                          WRDATA                  ;WRDATA
5143  013734  177400                          -400                    ;-400 WORDS
5144  013736  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
5145  013740     021                          .BYTE   21              ;SECTOR 21
5146  013741     000                          .BYTE   0               ;TRACK 0
5147  013742  000312                          312                     ;CYLINDER 312
5148  013744  004437  042260                  JSR     R4,GENCOM       ;GENERATE DATA
5149  013750  000012                          12                      ;PATTERN 12
5150  013752  000400                          400                     ;400 WORDS
5151
5152  013754  104417                          TLOADRK                 ;LOAD RK REGS
5153  013756  104430                          TWAT96                  ;WAIT FOR INTERRUPT
5154  013760  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5155
5156  013762  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5157  013764  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5158  013766  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
5159  013772  000521                          RDDATA.BA16             ;READ DATA AND SET BA16
5160  013774  177400                          -400                    ;400 WORDS
5161  013776  177770                          177770                  ;ACCROSS 64K BOUNDARY
5162  014000     021                          .BYTE   21              ;FROM SECTOR 21
5163  014001     000                          .BYTE   0               ;TRACK 0
5164  014002  000312                          312                     ;CYLINDER 312
5165
5166  014004  104417                          TLOADRK                 ;LOAD RK REGS
5167  014006  104424                          TWAT32                  ;WAIT FOR INTERRUPT
5168  014010  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5169
5170  014012  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5171  014014  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5172  014016  004437  042260                  JSR     R4,GENCOM       ;COMPARE DATA
5173  014022  120000                          120000                  ;MEM MANAGEMENT WITH. SOURCE
5174  014024  003777                          3777                    ;RELOCATION ARGUMENT
5175  014026  000400                          400                     ;400 WORDS
5176  014030  000413                          BR      5$              ;NO MISCOMPARES-SKIP
5177  014032  104015                          ERROR   15              ;REPORT FIRST ERROR
5178
5179  014034  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
5180  014040  005300                  64$:    DEC     R0              ;DECREMENT COUNT
5181  014042  001406                          BEQ     65$             ;IF ZERO - EXIT
5182  014044  004437  042260                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5183  014050  060000                          60000 .
5184  014052  000402                          BR      65$             ;NO MORE ERRORS - EXIT
5185  014054  104016                          ERROR   16              ;REPORT NEXT ERROR
5186  014056  000770                          BR      64$             ;LOOP
5187  014060                          65$:
5188
5189  014060                          5$:
5190                                  ;;**********************************************************************
```

```
                                        ;*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.
                                        ;*
                                        ;;*****************************************************************
014060  000004                TST36:  SCOPE
014062  012737  000012  001262        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
014070  123727  001326  000003        CMPB    $MAMS1,#3       ;CHECK IF >96K MEMORY
014076  002016                        BGE     3$              ;YES-SKIP
014100  032737  000002  001664  1$:   BIT     #MEMSZB,OPTFLG  ;TEST IF REPORT FLAG SET
014106  001011                        BNE     2$              ;NO-SKIP
                                
01411C  104401  051503                TYPE    ,OPR011         ;''INSUFFICIENT MEMORY-DATA TRANSFET WITH
014114  104401  051656                TYPE    ,OPR014         ;ADDRESS >96K BYPASSED''
014120  104401  051662                TYPE    ,OPR015
014124  052737  000002  001664        BIS     #MEMSZB,OPTFLG  ;SET REPORT FLAG
014132  000463                2$:     BR      6$
                                
014134  012737  014142  001110  3$:   MOV     #4$,$LPERR      ;SET LOCAL LOOP ON ERROR
                                
014142                          4$:
014142  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
014144  104003                        ERROR   3               ;BAD INIT ERROR
014146  004437  035674                JSR     R4,LRLOAD       ;LOAD ''L'' REG
014152  001123                        WRDATA!BA17             ;WRITE DATA AND BA17
014154  177400                        -400                    ;400 WORDS FROM
014156  177770                        177770                  ;ACCROSS 96K BOUNDARY
014160     022                        .BYTE   22              ;TO SECTOR 22
014161     000                        .BYTE   0               ;TRACK 0
014162  000312                        312                     ;CYL 312
014164  004437  042260                JSR     R4,GENCOM       ;GENERATE DATA
014170  010013                        10013                   ;PATTERN 13 MEM MAN WITH DEST.
014172  005777                        5777                    ;RELOCATION ARGUMENT
014174  000400                        400                     ;400 WORDS
                                
014176  104417                        TLOADRK                 ;LOAD RK REGS
014200  104430                        TWAT96                  ;WAIT FOR INTERRUPT
014202  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
                                
014204  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
014206  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
                                
014210  004437  035674                JSR     R4,LRLOAD       ;LOAD ''L'' REGS
014214  000121                        RDDATA                  ;RDDATA
014216  177400                        -400                    ;-400 WORDS
014220  070414                        IBUFF                   ;IBUFF IS BUFF ADDRESS
014222     022                        .BYTE   22              ;SECTOR 22
014223     000                        .BYTE   0               ;TRACK 0
014224  000312                        312                     ;CYLINDER 312
                                
014226  104417                        TLOADRK                 ;LOAD RK REGS
014230  104424                        TWAT32                  ;WAIT FOR INTERRUPT
```

```
5247  014232  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5248
5249  014234  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5250  014236  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5251
5252  014240  004437  042260                JSR     R4,GENCOM       ;COMPARE DATA
5253  014244  110000                        110000                  ;MEM MANAGEMENT WITH DESTINATION
5254  014246  005777                        5777                    ;RELOCATION ARGUMENT
5255  014250  000400                        400                     ;400 WORDS
5256  014252  000413                        BR      6$              ;NO MISCOMPARES-BRANCH
5257  014254  104015                        ERROR   15              ;REPORT 1ST ERROR
5258
5259  014256  013700  001634                MOV     ERRLMT,R0       ;GET ERROR LIMIT
5260  014262  005300              64$:      DEC     R0              ;DECREMENT COUNT
5261  014264  001406                        BEQ     65$             ;IF ZERO - EXIT
5262  014266  004437  042260                JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5263  014272  050000                        50000
5264  014274  000402                        BR      65$             ;NO MORE ERRORS - EXIT
5265  014276  104016                        ERROR   16              ;REPORT NEXT ERROR
5266  014300  000770                        BR      64$             ;LOOP
5267  014302              65$:
5268
5269  014302              6$:
5270                      ;;*****************************************************************
5271                      ;*TEST 37        READ DATA ADDRESS GREATER THAN 96K
5272                      ;*
5273                      ;*      ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 577770.
5274                      ;*      CHECK MEMORY FOR CORRECT TRANSFER.
5275                      ;*
5276                      ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF MORE THAN 96K
5277                      ;*              OF MEMORY IS PRESENT.
5278                      ;*
5279                      ;*****************************************************************
5280  014302  000004              TST37:  SCOPE
5281  014304  012737  000012  001262      MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5282  014312  123727  001326  000003      CMPB    $MAMS1,#3       ;CHECK IF >96K MEMORY
5283  014320  002001                      BGE     3$              ;YES-SKIP
5284  014322  000462              2$:     BR      6$
5285
5286  014324  012737  014332  001110  3$: MOV     #4$,$LPERR      ;SET LOCAL LOOP ON ERROR
5287
5288  014332              4$:
5289  014332  104416                      TSSINIT                 ;CLEAR SUBSYSTEM
5290  014334  104003                      ERROR   3               ;BAD INIT ERROR
5291
5292  014336  004437  035674              JSR     R4,LRLOAD       ;LOAD "L" REGS
5293  014342  000123                      WRDATA                  ;WRDATA
5294  014344  177400                      -400                    ;-400 WORDS
5295  014346  072414                      OBUFF                   ;OBUFF IS BUFF ADDRESS
5296  014350     005                      .BYTE   5               ;SECTOR 5
5297  014351     000                      .BYTE   0               ;TRACK 0
5298  014352  000312                      312                     ;CYLINDER 312
5299  014354  004437  042260              JSR     R4,GENCOM       ;GENERATE DATA
5300  014360  000014                      14                      ;PATTERN 14
5301  014362  000400                      400                     ;400 WORDS
5302
```

```
5303  014364  104417                          TLOADRK                 ;LOAD RK REGS
5304  014366  104430                          TWAT96                  ;WAIT FOR INTERRUPT
5305  014370  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5306
5307  014372  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5308  014374  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5309  014376  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
5310  014402  001121                          RDDATA.BA17             ;READ DATA WITH BA17 SET
5311  014404  177400                          -400                    ;400 WORDS
5312  014406  177770                          177770                  ;ACCROSS 96K BOUNDARY
5313  014410     005                          .BYTE   5               ;FROM SECTOR 5
5314  014411     000                          .BYTE   0               ;TRACK 0
5315  014412  000312                          312                     ;CYL 312
5316
5317  014414  104417                          TLOADRK                 ;LOAD RK REGS
5318  014416  104424                          TWAT32                  ;WAIT FOR INTERRUPT
5319  014420  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5320
5321  014422  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5322  014424  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5323  014426  004437  042260                  JSR     R4,GENCOM       ;COMPARE DATA
5324  014432  120000                          120000                  ;MEM MANAGEMENT WITH SOURCE
5325  014434  005777                          5777                    ;RELOCATION ARGUMENT
5326  014436  000400                          400                     ;400 WORDS
5327  014440  000413                          BR      6$              ;NO MISCOMPARES-SKIP
5328  014442  104015                          ERROR   15              ;REPORT FIRST ERROR
5329
5330  014444  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
5331  014450  005300                  64$:    DEC     R0              ;DECREMENT COUNT
5332  014452  001406                          BEQ     65$             ;IF ZERO - EXIT
5333  014454  004437  042260                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5334  014460  060000                          60000
5335  014462  000402                          BR      65$             ;NO MORE ERRORS - EXIT
5336  014464  104016                          ERROR   16              ;REPORT NEXT ERROR
5337  014466  000770                          BR      64$             ;LOOP
5338  014470                          65$:
5339
5340  014470                          6$:
5341                                  ;;*****************************************************************
5342                                  ;*TEST 40       PARTIAL SECTOR WRITE DATA
5343                                  ;*
5344                                  ;*      ISSUE A WRITE DATA OF 103 WORDS TO CYLINDER 312,
5345                                  ;*      HEAD 0, SECTOR 0.  MAKE SURE THE SECTOR WAS
5346                                  ;*      ZERO FILLED CORRECTLY.
5347                                  ;*
5348                                  ;;*****************************************************************
5349  014470  000004                  TST40:  SCOPE
5350  014472  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5351  014500  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
5352  014502  104003                          ERROR   3               ;BAD INIT ERROR
5353
5354  014504  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REG
5355  014510  000123                          WRDATA                  ;WRITE DATA
5356  014512  177675                          -103                    ;WORD COUNT PARTIAL SECTOR
5357  014514  072414                          OBUFF                   ;BUFF ADDRESS
5358  014516     007                          .BYTE   7               ;SECTOR 7
```

```
5359  014517  000                       .BYTE   0               ;TRACK 0
5360  014520  000312                    312                     ;CYLINDER 312
5361
5362  014522  004437  042260            JSR     R4,GENCOM       ;GENERATE DATA
5363  014526  000003                    3                       ;PATTERN 3
5364  014530  000400                    400                     ;400 WORDS
5365
5366  014532  104417                    TLOADRK                 ;LOAD RK REGS
5367  014534  104430                    TWAT96                  ;WAIT FOR INTERRUPT
5368  014536  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5369
5370  014540  104421                    TCHKOP                  ;CHECK FOR ANY ERROR
5371  014542  104004                    ERROR   4 ; OR 5,6,7    ;REPORT ALL ERROR
5372
5373  014544  004437  035674            JSR     R4,LRLOAD       ;LOAD "L" REGS
5374
5375  014550  000121                    RDDATA                  ;READ DATA
5376  014552  177400                    -400                    ;ONE FULL SECTOR
5377  014554  070414                    IBUFF                   ;BUFF ADDRESS
5378  014556  007                       .BYTE   7               ;SECTOR 7
5379  014557  000                       .BYTE   0               ;TRACK 0
5380  014560  000312                    312                     ;CYLINDER 312
5381
5382  014562  004437  042260            JSR     R4,GENCOM
5383  014566  002007                    2007                    ;CLEAR IBUFF TO ALL ONES
5384  014570  000400                    400
5385
5386  014572  104417                    TLOADRK                 ;LOAD RK REGS
5387  014574  104424                    TWAT32                  ;WAIT FOR INTERRUPT
5388  014576  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE
5389
5390  014600  104421                    TCHKOP                  ;CHECK FOR ANY ERRORS
5391  014602  104004                    ERROR   4 ; OR 5,6,7    ;REPORT ALL ERRORS
5392
5393  014604  012701  072622            MOV     #OBUFF+206,R1   ;CLEAR THE LAST 205 WORDS
5394  014610  012700  000275            MOV     #275,R0         ;OF THE OUTPUT BUFFER TO ZERO
5395  014614  005021            1$:     CLR     (R1)+           ;TO VERIFY THE PARTIAL SECTOR
5396  014616  005300                    DEC     R0              ;WRITE 0 FILLED THE SECTOR
5397  014620  001375                    BNE     1$
5398  014622  004437  042260            JSR     R4,GENCOM
5399  014626  100000                    100000                  ;COMPARE OBUFF & IBUFF.
5400  014630  000400                    400                     ;ALL 400 WORDS
5401  014632  000413                    BR      3$              ;NO ERRORS-EXIT
5402  014634  104015                    ERROR   15              ;REPORT FIRST COMPARE ERROR
5403
5404  014636  013700  001634            MOV     ERRLMT,R0       ;GET ERROR LIMIT
5405  014642  005300            2$:     DEC     R0              ;DECREMENT IT
5406  014644  001406                    BEQ     3$              ;IF ZERO-EXIT
5407  014646  004437  042260            JSR     R4,GENCOM
5408  014652  040000                    40000                   ;CONTINUE COMPARE
5409  014654  000402                    BR      3$              ;NO MORE ERRORS-EXIT
5410  014656  104016                    ERROR   16              ;REPORT NEXT COMPARE ERROR
5411  014660  000770                    BR      2$              ;LOOP
5412
5413  014662            3$:
5414                    ;;*****************************************************************
```

```
                                        ;*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.
                                        ;*
                                        ;;*****************************************************************
014662  000004                  TST41:  SCOPE
014664  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
014672  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
014674  104003                          ERROR   3               ;BAD INIT ERROR

014676  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
014702  000123                          WRDATA                  ;WRDATA
014704  177400                          -400                    ;-400 WORDS
014706  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
014710     017                          .BYTE   17              ;SECTOR 17
014711     000                          .BYTE   0               ;TRACK 0
014712  000312                          312                     ;CYLINDER 312
014714  004437  042260                  JSR     R4,GENCOM       ;GENERATE DATA
014720  000004                          4                       ;PATTERN 4
014722  000400                          400                     ;400 WORDS

014724  104417                          TLOADRK                 ;LOAD RK REGS
014726  104430                          TWAT96                  ;WAIT FOR INTERRUPT
014730  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

014732  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
014734  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

014736  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
014742  000121                          RDDATA                  ;RDDATA
014744  177675                          -103                    ;-103 WORDS
014746  070414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
014750     017                          .BYTE   17              ;SECTOR 17
014751     000                          .BYTE   0               ;TRACK 0
014752  000312                          312                     ;CYLINDER 312
014754  004437  042260                  JSR     R4,GENCOM
014760  002007                          2007                    ;CLEAR IBUFF
014762  000400                          400

014764  104417                          TLOADRK                 ;LOAD RK REGS
014766  104424                          TWAT32                  ;WAIT FOR INTERRUPT
014770  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
014772  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
014774  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

014776  012700  072622                  MOV     #OBUFF+206,R0   ;AFTER THE LAST 205 WORDS OF
015002  012701  000275                  MOV     #275,R1         ;THE OUTPUT BUFFER TO ALL ONES.
015006  012720  177777          1$:     MOV     #-1,(R0)+       ;THESE SHOULD ALL BE ONES IN
015012  005301                          DEC     R1              ;IBUFF BECAUSE THE PARTIAL
015014  001374                          BNE     1$              ;READ FILLED ONLY 103 WORDS.
015016  004437  042260                  JSR     R4,GENCOM       ;GO COMPARE IBUFF & OBUFF
015022  100000                          100000
```

```
 5471  015024  000400                          400                     ;ALL 400 WORDS
 5472  015026  000413                          BR      3$              ;NO ERRORS-EXIT
 5473  015030  104015                          ERROR   15              ;REPORT FIRST COMPARE ERROR
 5474
 5475  015032  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
 5476  015036  005300                  64$:    DEC     R0              ;DECREMENT COUNT
 5477  015040  001406                          BEQ     65$             ;IF ZERO - EXIT
 5478  015042  004437  042260                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
 5479  015046  040000                          40000
 5480  015050  000402                          BR      65$             ;NO MORE ERRORS - EXIT
 5481  015052  104016                          ERROR   16              ;REPORT NEXT ERROR
 5482  015054  000770                          BR      64$             ;LOOP
 5483  015056                          65$:
 5484
 5485  015056                          3$:
 5486                                  ;;*******************************************************************
 5487                                  ;*TEST 42       WRITE DATA WITH NON-EXISTENT MEMORY
 5488                                  ;*
 5489                                  ;*      ISSUE A WRITE DATA OF 1 WORD USING ADDRESS 776000.
 5490                                  ;*      MAKE SURE NON-EXISTENT MEMORY SETS.
 5491                                  ;*
 5492                                  ;;*******************************************************************
 5493  015056  000004                  TST42:  SCOPE
 5494  015060  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
 5495  015066  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
 5496  015070  104003                          ERROR   3               ;BAD INIT ERROR
 5497
 5498  015072  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REG
 5499  015076  001523                          BA16!BA17!WRDATA        ;BA16 & 17 SET WITH WRITE DATA
 5500  015100  177777                          -1                      ;WORD COUNT OF 1
 5501  015102  176000                          176000                  ;BUFF ADDRESS=IO PAGE BASE
 5502  015104     013                          .BYTE   13              ;SECT 13
 5503  015105     000                          .BYTE   0               ;TRACK 0
 5504  015106  000312                          312                     ;CYLINDER 312
 5505
 5506  015110  104417                          TLOADRK                 ;LOAD RK REGS
 5507  015112  104430                          TWAT96                  ;WAIT FOR INTERRUPT
 5508  015114  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 5509  015116  104422                          TCHKWE                  ;CHECK OPERATION WITH ERROR
 5510  015120  000000                          0
 5511  015122  000000                          0
 5512  015124  000040                          NEMERR                  ;NON-EXISTENT MEMORY ERROR
 5513  015126  104004                          ERROR   4 ;OR5,6,7      ;REPORT ANY DISCREPENCIES
 5514  015130  012737  054061  001450          MOV     #EM11A,EM10N    ;SET UP ERROR MESSAGE
 5515  015136  012737  050560  061002          MOV     #OPER42,DF010A  ;WITH SUPPORT MESSAGE
 5516  015144  113700  001541                  MOVB    T.CS1+1,R0      ;GET UPPER CS1
 5517  015150  042700  177774                  BIC     #177774,R0      ;CLEAR UNUSED BITS
 5518  015154  022700  000003                  CMP     #3,R0           ;TEST IF BOTH UPPER BUS BITS SET
 5519  015160  001406                          BEQ     1$              ;YES - SKIP
 5520  015162  010037  001204                  MOV     R0,$REG11       ;SET UP FOR ERROR REPORT
 5521  015166  012737  000003  001202          MOV     #3,$REG10
 5522  015174  104010                          ERROR   10
 5523  015176  022737  176002  001544  1$:     CMP     #176002,T.BA    ;TEST IF BUSS ADDRESS LOW OKAY
 5524  015204  001412                          BEQ     2$              ;YES - SKIP
 5525  015206  012737  054033  001450          MOV     #EM11,EM10N     ;SET UP MESSGAE
 5526  015214  012737  176002  001202          MOV     #176002,$REG10  ;STORE VALUE FOR REPORT
```

```
 5527  015222  013737  001544  001204          MOV     T.BA,$REG11
 5528  015230  104010                          ERROR   10
 5529
 5530  015232  005737  001542          2$:     TST     T.WC            ;TEST IF WORD COUNT CORRECT
 5531  015236  001411                          BEQ     3$              ;YES - SKIP
 5532  015240  012737  054006  001450          MOV     #EM10,EM10N     ;SET UP MESSAGE
 5533  015246  005037  001202                  CLR     $REG10
 5534  015252  013737  001542  001204          MOV     T.WC,$REG11
 5535  015260  104010                          ERROR 10
 5536
 5537  015262                          3$:
 5538
 5539                                  ;;*****************************************************************
 5540                                  ;*TEST 43       READ DATA WITH NON-EXISTENT MEMORY
 5541                                  ;*
 5542                                  ;*      ISSUE A READ DATA OF 1 WORD USING ADDRESS 776000.
 5543                                  ;*      MAKE SURE NON-EXISTENT MEMORY SETS.
 5544                                  ;*
 5545                                  ;*****************************************************************
 5546  015262  000004                  TST43:  SCOPE
 5547  015264  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
 5548  015272  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
 5549  015274  104003                          ERROR   3               ;BAD INIT ERROR
 5550
 5551  015276  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REG
 5552  015302  001521                          BA16!BA17!RDDATA        ;BA16 & 17 WITH READ DATA
 5553  015304  177777                          -1                      ;WORD COUNT OF 1
 5554  015306  176000                          176000                  ;BUFF ADDRESS=IO PAGE BASE
 5555  015310     013                          .BYTE   13              ;SECTOR 13
 5556  015311     000                          .BYTE   0               ;TRACK 0
 5557  015312  000312                          312                     ;CYL 312
 5558
 5559  015314  104417                          TLOADRK                 ;LOAD RK REGS
 5560  015316  104430                          TWAT96                  ;WAIT FOR INTERRUPT
 5561  015320  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 5562  015322  104422                          TCHKWE                  ;CHECK OPERATION WITH ERRORS
 5563  015324  000000                          0
 5564  015326  000000                          0
 5565  015330  000040                          NEMERR                  ;NON-EXISTENT MEMORY ERROR
 5566  015332  104004                          ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES
 5567  015334  012737  054061  001450          MOV     #EM11A,EM10N    ;SET MESSAGE
 5568  015342  012737  050634  061002          MOV     #OPER43,DF010A  ;SET SUPPORT MESSAGE
 5569  015350  113700  001541                  MOVB    T.CS1+1,R0              ;GET UPPER CS1
 5570  015354  042700  177774                  BIC     #177774,R0      ;CLEAR UNWANTED BITS
 5571  015360  022700  000003                  CMP     #3,R0           ;TEST BOTH BUS 16 & 17 SET
 5572  015364  001406                          BEQ     1$              ;YES - SKIP
 5573  015366  012737  000003  001202          MOV     #3,$REG10       ;SET VALUES FOR REPORT
 5574  015374  C10037  001204                  MOV     R0,$REG11
 5575  015400  104010                          ERROR   10
 5576
 5577  015402  022737  176002  001544  1$:     CMP     #176002,T.BA    ;TEST IF BUS ADDRESS CORRECT
 5578  015410  001412                          BEQ     2$              ;YES - SKIP
 5579  015412  012737  054033  001450          MOV     #EM11,EM10N     ;SET MESSAGE
 5580  015420  012737  176002  001202          MOV     #176002,$REG10  ;SET VALUES FOR REPORT
 5581  015426  013737  001544  001204          MOV     T.BA,$REG11
 5582  015434  104010                          ERROR   10
```

```
5583
5584  015436  005737  001542          2$:     TST     T.WC            ;TEST IF WORD COUNT CORRECT
5585  015442  001411                          BEQ     3$              ;YES - SKIP
5586  015444  012737  054006  001450          MOV     #EM10,EM10N     ;SET MESSAGE
5587  015452  005037  001202                  CLR     $REG10          ;SET VALUES
5588  015456  013737  001542  001204          MOV     T.WC,$REG11
5589  015464  104010                          ERROR   10
5590
5591  015466                          3$:
5592
5593                                  ;;**********************************************************************
5594                                  ;*TEST 44       UNIBUS PARITY ERROR
5595                                  ;*
5596                                  ;*      INITIALIZE A MEMORY LOCATION WITH BAD
5597                                  ;*      PARITY. ISSUE A WRITE DATA OF 400 WORDS
5598                                  ;*      STARTING AT A LOCATION 112 WORDS BEFORE
5599                                  ;*      THE LOCATION WITH BAD PARITY.  MAKE SURE
5600                                  ;*      THAT UNIBUS PARITY ERROR SETS.
5601                                  ;*
5602                                  ;*      NOTE:   THIS TEST IS ONLY EXECUTED IF
5603                                  ;*              MEMORY PARITY OPTION EXISTS FOR
5604                                  ;*              BUFFER. IT WILL NOT BE EXECUTED
5605                                  ;*              ON 'ECC' TYPE MEMORY.
5606
5607                                  ;*
5608                                  ;;**********************************************************************
5609  015466  000004                  TST44:  SCOPE
5610  015470  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5611  015476  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
5612  015500  104003                          ERROR   3               ;BAD INIT ERROR
5613
5614  015502  032737  000100  001664          BIT     #PARPRE,OPTFLG  ;TEST IF PARITY OPTION PRESENT
5615  015510  001013                          BNE     1$              ;YES-SKIP
5616  015512  032737  000004  001664          BIT     #MEMPYB,OPTFLG  ;TEST IF PARITY OPTION REPORTED
5617  015520  001005                          BNE     25$             ;YES-SKIP TO EXIT
5618  015522  104401  051362                  TYPE    ,OPR010         ;PRINT BYPASS MESSAGE
5619  015526  052737  000004  001664          BIS     #MEMPYB,OPTFLG  ;SET OPTION REPORTED BIT
5620  015534  000137  016326          25$:    JMP     4$              ;SKIP TO EXIT
5621
5622  015540                          1$:
5623  015540  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
5624  015544  000123                          WRDATA                  ;WRDATA
5625  015546  177400                          -400                    ;-400 WORDS
5626  015550  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
5627  015552     010                          .BYTE   10              ;SECTOR 10
5628  015553     000                          .BYTE   0               ;TRACK 0
5629  015554  000312                          312                     ;CYLINDER 312
5630
5631  015556  032737  002000  001664          BIT     #CP1170,OPTFLG  ;TEST IF 11/70
5632  015564  001002                          BNE     3$              ;YES - SKIP
5633  015566  005077  164100                  CLR     @CSRPTR         ;CLEAR PARITY IE
5634
5635  015572  004437  042260          3$:     JSR     R4,GENCOM       ;GENERATE DATA
5636  015576  000007                          7                       ;PATTERN 7
5637  015600  000400                          400                     ;400 WORDS
5638
```

```
5639  015602  012746  000340          MOV     #PR7,-(SP)        ;PUT PRIORITY 7 ON STACK
5640  015606  012746  015614          MOV     #10$,-(SP)        ;PUT ADDRESS ON STACK
5641  015612  000002                  RTI                       ;SET PRI
5642  015614                  10$:
5643
5644  015614  012737  100200  072640  MOV     #100200,OBUFF+224 ;SET WORD IN BUFFER
5645  015622  032737  002000  001664  BIT     #CP1170,OPTFLG    ;TEST IF 11/70
5646  015630  001011                  BNE     5$                ;YES - SKIP
5647  015632  012777  000004  164032  MOV     #BIT2,@CSRPTR     ;SET WRITE WRONG PARITY BIT
5648  015640  012737  100200  072640  MOV     #100200,OBUFF+224  ;SET BAD PARITY IN MEMORY
5649  015646  012777  000001  164016  MOV     #BIT0,@CSRPTR     ;CLEAR CONTROL BIT, SET IE BIT
5650
5651  015654  013746  001622    5$:   MOV     RKPRI,-(SP)       ;SET OLD PRIORITY
5652  015660  012746  015666          MOV     #11$,-(SP)        ;SET ADDRESS
5653  015664  000002                  RTI                       ;RESTORE PRI
5654  015666  013704  001672    11$:  MOV     CSRPTR,R4         ;SET R4 WITH CSR POINTER
5655  015672  005000                  CLR     R0                ;SET R0 FOR COUNTER
5656  015674  012701  001662          MOV     #INTSET,R1        ;SET R1 FOR POINTER TO INTERRUPT FLAG
5657  015700  012777  070400  163712  MOV     #SPCHLR,@RKVEC    ;SET UP INTERRUPT VECTORS FOR
5658  015706  012777  070410  164002  MOV     #SPCPAR,@MMVECA   ; RK611 AND PARITY ERROR
5659  015714  012737  016302  001264  MOV     #2$,$ESCAPE       ;SET UP ESCAPE FOR ERROR
5660  015722  104417                  TLOADRK                   ;LOAD RK REGS
5661  015724  032737  002000  001664  BIT     #CP1170,OPTFLG    ;TEST IF 11/70
5662  015732  001434                  BEQ     45$               ;NO - SKIP
5663  015734  012737  000016  177746  MOV     #16,177746        ;SET TO DISABLE CACHE AND UNIBUS ERROR
5664  015742  012777  000000  163740  MOV     #0,@KWLADD        ;TURN OFF CLOCK INTERRUPTS
5665  015750  012777  000000  163166  MOV     #0,@$TKS          ;TURN OFF KEYBOARD
5666  015756  012737  170000  177750  MOV     #170000,177750    ;SET FOR ERROR FORCE
5667
5668                          ;**********************
5669  015764  105711          40$:    TSTB    (R1)              ;LOOP TO WAIT FOR INTERRUPT OR ABORT
5670  015766  003005                  BGT     43$               ;WAIT. THE CODE BETWEEN THE STARS IS SET
5671  015770  005300                  DEC     R0                ;SET UP SO ALL BYTES HAVE PARITY OF 1.
5672  015772  100774                  BMI     40$               ;IF THIS CODE IS CHANGED, REMEMBER ALL
5673  015774  000240                  NOP                       ;BYTES MUST HAVE AN EVEN NUMBER OF
5674  015776  003372                  BGT     40$               ;BITS.
5675  016000  000240                  NOP
5676  016002  005014          43$:    CLR     (R4)              ;CLEAR ERROR FORCE
5677                          ;**************************
5678
5679  016004  005037  177746          CLR     177746            ;ENABLE CACHE
5680  016010  012777  000100  163672  MOV     #BIT6,@KWLADD     ;TURN ON CLOCK INTERRUPTS
5681  016016  012777  000100  163120  MOV     #100,@$TKS        ;TURN ON KEYBOARD INTERRUPTS
5682  016024  104430          45$:    TWAT96                    ;WAIT FOR INTERRUPT
5683  016026  000414                  BR      46$               ;TO SLOW/NOT COMPLETE ERROR
5684  016030  032737  002000  001664  BIT     #CP1170,OPTFLG    ;TEST IF 11/70
5685  016036  001024                  BNE     48$               ;YES - SKIP
5686  016040  005077  163626          CLR     @CSRPTR           ;ELSE CLEAR CSR
5687  016044  005037  072640          CLR     OBUFF+224         ;CLEAR THE BAD PARITY WORD
5688  016050  012777  000001  163614  MOV     #1,@CSRPTR        ;SET PARITY DETECT AGAIN
5689  016056  000414                  BR      48$               ;SKIP
5690
5691  016060  032737  002000  001664  46$:    BIT     #CP1170,OPTFLG    ;TEST IF 11/70
5692  016066  001007                  BNE     47$               ;YES - SKIP
5693  016070  005077  163576          CLR     @CSRPTR           ;CLEAR CSR
5694  016074  005037  072640          CLR     OBUFF+224         ;CLEAR BAD PARITY WORD
```

```
5695  016100  012777  000001  163564          MOV     #1,@CSRPTR      ;SET UP PARITY DETECT AGAIN
5696  016106  104002                  47$:    ERROR   2               ;REPORT TO SLOW ERROR
5697  016110                          48$:
5698  016110  104422                          TCHKWE                  ;CHECK OPERATION WITH ERROR
5699  016112  000000                          0
5700  016114  000000                          0
5701  016116  000100                          UPERR                   ;UNIBUS PARITY ERROR
5702  016120  104004                          ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES
5703
5704  016122  005037  001264                  CLR     $ESCAPE         ;CLEAR ESCAPE
5705  016126  012737  050707  061002          MOV     #OPER44,DF010A  ;SET MESSAGES FOR REPORT
5706  016134  012737  054033  001450          MOV     #EM11,EM10N
5707  016142  023727  001544  072642          CMP     T.BA,#OBUFF+226 ;CHECK IF BA IN RANGE
5708  016150  103010                          BHIS    14$             ;NOT TO LOW - SKIP
5709  016152  012737  072642  001202          MOV     #OBUFF+226,$REG10 ;SET VALUES FOR REPORT
5710  016160  013737  001544  001204          MOV     T.BA,$REG11
5711  016166  104010                          ERROR   10
5712  016170  000413                          BR      16$
5713
5714  016172  023727  001544  072646  14$:    CMP     T.BA,#OBUFF+232 ;CHECK IF BA IN RANGE
5715  016200  101407                          BLOS    16$             ;YES - SKIP
5716  016202  012737  072646  001202          MOV     #OBUFF+232,$REG10 ;SET VALUES
5717  016210  012737  001544  001204          MOV     #T.BA,$REG11
5718  016216  104010                          ERROR   10
5719
5720  016220  012737  054006  001450  16$:    MOV     #EM10,EM10N     ;SET MESSAGE
5721  016226  023727  001542  177513          CMP     T.WC,#-265      ;CHECK IF WORD COUNT WITHIN RANGE
5722  016234  103007                          BHIS    20$             ;YES - SKIP
5723  016236  012737  177513  001202          MOV     #-265,$REG10    ;SET VALUES
5724  016244  013737  001542  001204          MOV     T.WC,$REG11
5725  016252  104010                          ERROR   10
5726
5727  016254  023727  001542  177515  20$:    CMP     T.WC,#-263      ;STILL CHECKING IF WC IN RANGE
5728  016262  101407                          BLOS    2$              ;YES - SKIP
5729  016264  012737  177515  001202          MOV     #-263,$REG10    ;SET VALUES
5730  016272  013737  001542  001204          MOV     T.WC,$REG11
5731  016300  104010                          ERROR   10
5732
5733  016302                          2$:
5734  016302  005037  001264                  CLR     $ESCAPE         ;REV 006
5735  016306  012777  034442  163304          MOV     #INTHLR,@RKVEC  ;RESET INT. VECTORS FOR RK06
5736  016314  004737  034522                  JSR     PC,OPTTST
5737  016320  012777  000100  162616          MOV     #100,@$TKS      ;INSURE KEYBOARD IS ENABLED
5738
5739  016326                          4$:
5740
5741
5742                                  .SBTTL  **MULTIPLE SECTOR OPERATIONS
5743
5744                                  ;;******************************************************************
5745                                  ;*TEST 45        TWO SECTOR WRITE DATA (PART 1)
5746                                  ;*
5747                                  ;*      ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5748                                  ;*      TRACK 0, SECTOR 0.  READ DATA BACK ONE SECTOR
5749                                  ;*      AT A TIME AND MAKE SURE IT IS CORRECT.
5750                                  ;*
```

```
                                ;;*****************************************************************
016326  000004                 TST45:  SCOPE
016330  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
016336  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
016340  104003                          ERROR   3               ;BAD INIT ERROR

016342  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
016346  000123                          WRDATA                  ;WRDATA
016350  177000                          -1000                   ;-1000 WORDS
016352  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
016354     000                          .BYTE   0               ;SECTOR 0
016355     000                          .BYTE   0               ;TRACK 0
016356  000312                          312                     ;CYLINDER 312

016360  004437  042260                  JSR     R4,GENCOM       ;GENERATE DATA
016364  000015                          15                      ;PATTERN 15
016366  001000                          1000                    ;1000 WORDS

016370  104417                          TLOADRK                 ;LOAD RK REGS
016372  104430                          TWAT96                  ;WAIT FOR INTERRUPT
016374  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

016376  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
016400  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

016402  004437  042260                  JSR     R4,GENCOM       ;CLEAR IBUFF
016406  002007                          2007                    ;TO ALL 1'S
016410  001000                          1000

016412  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
016416  000121                          RDDATA                  ;RDDATA
016420  177400                          -400                    ;-400 WORDS
016422  070414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
016424     000                          .BYTE   0               ;SECTOR 0
016425     000                          .BYTE   0               ;TRACK 0
016426  000312                          312                     ;CYLINDER 312

016430  104417                          TLOADRK                 ;LOAD RK REGS
016432  104424                          TWAT32                  ;WAIT FOR INTERRUPT
016434  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

016436  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
016440  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

016442  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
016446  000121                          RDDATA                  ;RDDATA
016450  177400                          -400                    ;-400 WORDS
016452  071414                          IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
016454     001                          .BYTE   1               ;SECTOR 1
016455     000                          .BYTE   0               ;TRACK 0
016456  000312                          312                     ;CYLINDER 312

016460  104417                          TLOADRK                 ;LOAD RK REGS
016462  104424                          TWAT32                  ;WAIT FOR INTERRUPT
016464  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
```

```
5807  016466  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5808  016470  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5809
5810  016472  004437  042260                  JSR     R4,GENCOM       ;COMPARE DATA
5811  016476  100000                          100000                  ;
5812  016500  001000                          1000                    ;1000 WORDS
5813  016502  000413                          BR      2$              ;NO MISCOMPARES-EXIT
5814  016504  104015                          ERROR   15              ;REPORT FIRST ERROR
5815
5816  016506  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
5817  016512  005300                  64$:    DEC     R0              ;DECREMENT COUNT
5818  016514  001406                          BEQ     65$             ;IF ZERO - EXIT
5819  016516  004437  042260                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5820  016522  040000                          40000
5821  016524  000402                          BR      65$             ;NO MORE ERRORS - EXIT
5822  016526  104016                          ERROR   16              ;REPORT NEXT ERROR
5823  016530  000770                          BR      64$             ;LOOP
5824  016532                          65$:
5825
5826  016532                          2$:
5827                                  ;;*******************************************************************
5828                                  ;*TEST 46       TWO SECTOR WRITE DATA (PART 2)
5829                                  ;*
5830                                  ;*      ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5831                                  ;*      TRACK 0, SECTOR 23.  READ DATA BACK ONE SECTOR
5832                                  ;*      AT A TIME AND MAKE SURE A MID-TRANSFER
5833                                  ;*      SEEK DID NOT TAKE PLACE.
5834                                  ;*
5835                                  ;;*******************************************************************
5836  016532  000004                  TST46:  SCOPE
5837  016534  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
5838  016542  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
5839  016544  104003                          ERROR   3               ;BAD INIT ERROR
5840
5841  016546  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
5842  016552  000123                          WRDATA                  ;WRDATA
5843  016554  177000                          -1000                   ;-1000 WORDS
5844  016556  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
5845  016560     023                          .BYTE   23              ;SECTOR 23
5846  016561     000                          .BYTE   0               ;TRACK 0
5847  016562  000312                          312                     ;CYLINDER 312
5848
5849  016564  004437  042260                  JSR     R4,GENCOM       ;GENERATE DATA
5850  016570  000016                          16                      ;PATTERN 16
5851  016572  001000                          1000                    ;1000 WORDS
5852
5853  016574  104417                          TLOADRK                 ;LOAD RK REGS
5854  016576  104430                          TWAT96                  ;WAIT FOR INTERRUPT
5855  016600  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5856
5857  016602  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5858  016604  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5859
5860                                  ;       IF THE TRACK ADDRESS AT THE END OF OPERATION IS IN ERROR
5861                                  ;       THE CONTROLLER DID A MID-TRANSFER SEEK AS THOUGH IT
5862                                  ;       WERE IN 24(8) SECTORS PER TRACK MODE.
```

```
5863
5864  016606  004437  035674              JSR     R4,LRLOAD       ;LOAD "L" REGS
5865  016612  000121                      RDDATA                  ;RDDATA
5866  016614  177400                      -400                    ;-400 WORDS
5867  016616  070414                      IBUFF                   ;IBUFF IS BUFF ADDRESS
5868  016620     023                      .BYTE   23              ;SECTOR 23
5869  016621     000                      .BYTE   0               ;TRACK 0
5870  016622  000312                      312                     ;CYLINDER 312
5871
5872  016624  004437  042260              JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
5873  016630  002007                      2007
5874  016632  001000                      1000
5875
5876  016634  104417                      TLOADRK                 ;LOAD RK REGS
5877  016636  104424                      TWAT32                  ;WAIT FOR INTERRUPT
5878  016640  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5879
5880  016642  104421                      TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5881  016644  104004                      ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5882
5883  016646  004437  035674              JSR     R4,LRLOAD       ;LOAD "L" REGS
5884  016652  000121                      RDDATA                  ;RDDATA
5885  016654  177400                      -400                    ;-400 WORDS
5886  016656  071414                      IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
5887  016660     024                      .BYTE   24              ;SECTOR 24
5888  016661     000                      .BYTE   0               ;TRACK 0
5889  016662  000312                      312                     ;CYLINDER 312
5890
5891  016664  104417                      TLOADRK                 ;LOAD RK REGS
5892  016666  104424                      TWAT32                  ;WAIT FOR INTERRUPT
5893  016670  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5894
5895  016672  104421                      TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
5896  016674  104004                      ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5897
5898  016676  004437  042260              JSR     R4,GENCOM       ;COMPARE DATA
5899  016702  100000                      100000
5900  016704  001000                      1000                    ;1000 WORDS
5901  016706  000413                      BR      1$              ;NO ERRORS-SKIP
5902  016710  104015                      ERROR   15              ;REPORT FIRST ERROR
5903
5904  016712  013700  001634              MOV     ERRLMT,R0       ;GET ERROR LIMIT
5905  016716  005300              64$:    DEC     R0              ;DECREMENT COUNT
5906  016720  001406                      BEQ     65$             ;IF ZERO - EXIT
5907  016722  004437  042260              JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5908  016726  040000                      40000
5909  016730  000402                      BR      65$             ;NO MORE ERRORS - EXIT
5910  016732  104016                      ERROR   16              ;REPORT NEXT ERROR
5911  016734  000770                      BR      64$             ;LOOP
5912  016736              65$:
5913  016736              1$:
5914                      ;;*****************************************************************
5915                      ;*TEST 47       TWO SECTOR WRITE DATA (PART 3)
5916                      ;*
5917                      ;*      ISSUE A WRITE DATA OF 401 WORDS TO CYLINDER 312,
5918                      ;*      TRACK 0, SECTOR 10.  READ DATA BACK ONE SECTOR AT
```

```
                        ;*      A TIME AND CHECK ZERO FILL OF SECOND SECTOR.
                        ;*
                        ;;*****************************************************************
016736  000004         TST47:  SCOPE
016740  012737  000012  001262         MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
016746  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
016750  104003                        ERROR   3               ;BAD INIT ERROR
016752  004437  035674                JSR     R4,LRLOAD       ;LOAD "L" REGS
016756  000123                        WRDATA                  ;WRDATA
016760  177377                        -401                    ;-401 WORDS
016762  072414                        OBUFF                   ;OBUFF IS BUFF ADDRESS
016764     010                        .BYTE   10              ;SECTOR 10
016765     000                        .BYTE   0               ;TRACK 0
016766  000312                        312                     ;CYLINDER 312
016770  004437  042260                JSR     R4,GENCOM       ;GENERATE DATA
016774  000002                        2                       ;PATTERN 2
016776  001000                        1000                    ;1000 WORDS
017000  104417                        TLOADRK                 ;LOAD RK REGS
017002  104430                        TWAT96                  ;WAIT FOR INTERRUPT
017004  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
017006  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
017010  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
                                    CLEAR LAST 377 WORDS OF OBUFF FOR EXPECTED ZEROS FROM ZERO FILL
017012  012700  073416        .       MOV     #OBUFF+1002,R0  ;GET STARTING ADDRESS TO BE CLEARED
017016  012701  000377                MOV     #377,R1         ;NUMBER OF WORDS
017022  005020                1$:     CLR     (R0)+           ;CLEAR WORD
017024  005301                        DEC     R1              ;DEC COUNT
017026  001375                        BNE     1$              ;LOOP UNTIL COUNT ZERO
017030  004437  042260                JSR     R4,GENCOM       ;CLEAR IBUFF TO ONES
017034  002007                        2007
017036  001000                        1000
017040  004437  035674                JSR     R4,LRLOAD       ;LOAD "L" REGS
017044  000121                        RDDATA                  ;RDDATA
017046  177400                        -400                    ;-400 WORDS
017050  070414                        IBUFF                   ;IBUFF IS BUFF ADDRESS
017052     010                        .BYTE   10              ;SECTOR 10
017053     000                        .BYTE   0               ;TRACK 0
017054  000312                        312                     ;CYLINDER 312
017056  104417                        TLOADRK                 ;LOAD RK REGS
017060  104424                        TWAT32                  ;WAIT FOR INTERRUPT
017062  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
017064  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
017066  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
017070  004437  035674                JSR     R4,LRLOAD       ;LOAD "L" REGS
017074  000121                        RDDATA                  ;RDDATA
017076  177400                        -400                    ;-400 WORDS
017100  071414                        IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
017102     011                        .BYTE   11              ;SECTOR 11
017103     000                        .BYTE   0               ;TRACK 0
```

```
5975  017104  000312                          312                     ;CYLINDER 312
5976
5977  017106  104417                          TLOADRK                 ;LOAD RK REGS
5978  017110  104424                          TWAT32                  ;WAIT FOR INTERRUPT
5979  017112  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
5980
5981  017114  004437  042260                  JSR     R4,GENCOM       ;DATA COMPARE
5982  017120  100000                          100000
5983  017122  001000                          1000                    ;1000 WORDS
5984  017124  000413                          BR      2$              ;NO ERROR-SKIP
5985  017126  104015                          ERROR   15              ;REPORT FIRST ERROR
5986
5987  017130  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
5988  017134  005300                  64$:    DEC     R0              ;DECREMENT COUNT
5989  017136  001406                          BEQ     65$             ;IF ZERO - EXIT
5990  017140  004437  042260                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
5991  017144  040000                          40000
5992  017146  000402                          BR      65$             ;NO MORE ERRORS - EXIT
5993  017150  104016                          ERROR   16              ;REPORT NEXT ERROR
5994  017152  000770                          BR      64$             ;LOOP
5995  017154                          65$:
5996
5997  017154                          2$:
5998                                  ;;*****************************************************************
5999                                  ;*TEST 50       MID-TRANSFER SEEK ON WRITE (PART 1)
6000                                  ;*
6001                                  ;*      ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
6002                                  ;*      TRACK 0, SECTOR 25.  READ DATA BACK ONE SECTOR
6003                                  ;*      AT A TIME AND MAKE SURE A MID-TRANSFER SEEK
6004                                  ;*      DID TAKE PLACE.
6005                                  ;*
6006                                  ;;*****************************************************************
6007  017154  000004                  TST50:  SCOPE
6008  017156  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
6009  017164  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
6010  017166  104003                          ERROR   3               ;BAD INIT ERROR
6011
6012  017170  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
6013  017174  000123                          WRDATA                  ;WRDATA
6014  017176  177000                          -1000                   ;-1000 WORDS
6015  017200  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
6016  017202     025                          .BYTE   25              ;SECTOR 25
6017  017203     000                          .BYTE   0               ;TRACK 0
6018  017204  000312                          312                     ;CYLINDER 312
6019
6020  017206  004437  042260                  JSR     R4,GENCOM       ;GENERATE DATA
6021  017212  000003                          3                       ;PATTERN 3
6022  017214  001000                          1000                    ;1000 WORDS
6023
6024  017216  104417                          TLOADRK                 ;LOAD RK REGS
6025  017220  104430                          TWAT96                  ;WAIT FOR INTERRUPT
6026  017222  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6027
6028  017224  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6029  017226  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6030                                  ;       A TRACK ERROR PRINTED OUT AT THE END OF THE OPERATION INDICATES A
```

```
                                        ;       MID-TRANSFER HEAD SWITCH DID NOT OCCUR.
017230  004437  042260                        JSR     R4,GENCOM
017234  002007                                2007
017236  001000                                1000
017240  004437  035674                        JSR     R4,LRLOAD       ;LOAD "L" REGS
017244  000121                                RDDATA                  ;RDDATA
017246  177400                                -400                    ;-400 WORDS
017250  070414                                IBUFF                   ;IBUFF IS BUFF ADDRESS
017252     025                                .BYTE   25              ;SECTOR 25
017253     000                                .BYTE   0               ;TRACK 0
017254  000312                                312                     ;CYLINDER 312
017256  104417                                TLOADRK                 ;LOAD RK REGS
017260  104425                                TWAT48                  ;WAIT FOR INTERRUPT
017262  104002                                ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
017264  104421                                TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
017266  104004                                ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
017270  004437  035674                        JSR     R4,LRLOAD       ;LOAD "L" REGS
017274  000121                                RDDATA                  ;RDDATA
017276  177400                                -400                    ;-400 WORDS
017300  071414                                IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
017302     000                                .BYTE   0               ;SECTOR 0
017303     001                                .BYTE   1               ;TRACK 1
017304  000312                                312                     ;CYLINDER 312
017306  104417                                TLOADRK                 ;LOAD RK REGS
017310  104425                                TWAT48                  ;WAIT FOR INTERRUPT
017312  104002                                ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
017314  104421                                TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
017316  104004                                ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
017320  004437  042260                        JSR     R4,GENCOM       ;COMPARE DATA
017324  100000                                100000
017326  001000                                1000                    ;1000 WORDS
017330  000413                                BR      1$              ;NO ERRORS-SKIP
017332  104015                                ERROR   15              ;REPORT FIRST ERROR
017334  013700  001634                        MOV     ERRLMT,R0       ;GET ERROR LIMIT
017340  005300                        64$:    DEC     R0              ;DECREMENT COUNT
017342  001406                                BEQ     65$             ;IF ZERO - EXIT
017344  004437  042260                        JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
017350  040000                                40000
017352  000402                                BR      65$             ;NO MORE ERRORS - EXIT
017354  104016                                ERROR   16              ;REPORT NEXT ERROR
017356  000770                                BR      64$             ;LOOP
017360                                  65$:
                                        1$:
                                        ;;*****************************************************************
                                        ;*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.
                                ;*
                                ;;*****************************************************************
017360  000004                  TST51:  SCOPE
017362  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
017370  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
017372  104003                          ERROR   3               ;BAD INIT ERROR

017374  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
017400  000123                          WRDATA                  ;WRDATA
017402  177000                          -1000                   ;-1000 WORDS
017404  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
017406     025                          .BYTE   25              ;SECTOR 25
017407     002                          .BYTE   2               ;TRACK 2
017410  000312                          312                     ;CYLINDER 312

017412  004437  042260                  JSR     R4,GENCOM       ;GENERATE DATA
017416  000004                          4                       ;PATTERN 4
017420  001000                          1000                    ;1000 WORDS

017422  104417                          TLOADRK                 ;LOAD RK REGS
017424  104430                          TWAT96                  ;WAIT FOR INTERRUPT
017426  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
                                ;       A CYLINDER ERROR REPORTED AT THE END OF THE OPERATION INDICATES A
                                ;       MID-TRANSFER SEEK DID NOT OCCUR.
017430  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
017432  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

017434  004437  042260                  JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
017440  002007                          2007
017442  001000                          1000

017444  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
017450  000121                          RDDATA                  ;RDDATA
017452  177400                          -400                    ;-400 WORDS
017454  070414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
017456     025                          .BYTE   25              ;SECTOR 25
017457     002                          .BYTE   2               ;TRACK 2
017460  000312                          312                     ;CYLINDER 312

017462  104417                          TLOADRK                 ;LOAD RK REGS
017464  104425                          TWAT48                  ;WAIT FOR INTERRUPT
017466  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

017470  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
017472  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

017474  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
017500  000121                          RDDATA                  ;RDDATA
017502  177400                          -400                    ;-400 WORDS
017504  071414                          IBUFF+1000                      ;IBUFF+1000 IS BUFF ADDRESS
017506     000                          .BYTE   0               ;SECTOR 0
017507     000                          .BYTE   0               ;TRACK 0
017510  000313                          313                     ;CYLINDER 313
```

```
                                        TLOADRK                 ;LOAD RK REGS
                                        TWAT48                  ;WAIT FOR INTERRUPT
                                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

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

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

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

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

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

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

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

                                        JSR     R4,LRLOAD       ;LOAD "L" REGS
                                        RDDATA                  ;RDDATA
                                        -1000                   ;-1000 WORDS
                                        IBUFF                   ;IBUFF IS BUFF ADDRESS
```

```
6199  017632  000000                  .BYTE   0               ;SECTOR 0
6200  017633  000000                  .BYTE   0               ;TRACK 0
6201  017634  000312                  312                     ;CYLINDER 312
6202
6203  017636  104417                  TLOADRK                 ;LOAD RK REGS
6204  017640  104430                  TWAT96                  ;WAIT FOR INTERRUPT
6205  017642  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6206
6207  017644  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6208  017646  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6209
6210  017650  004437  042260          JSR     R4,GENCOM       ;COMPARE DATA
6211  017654  100000                  100000
6212  017656  001000                  1000                    ;1000 WORDS
6213  017660  000413                  BR      1$              ;NO MISCOMPARES-SKIP
6214  017662  104015                  ERROR   15
6215
6216  017664  013700  001034          MOV     ERRLMT,R0       ;GET ERROR LIMIT
6217  017670  005300          64$:    DEC     R0              ;DECREMENT COUNT
6218  017672  001406                  BEQ     65$             ;IF ZERO - EXIT
6219  017674  004437  042260          JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
6220  017700  040000                  40000
6221  017702  000402                  BR      65$             ;NO MORE ERRORS - EXIT
6222  017704  104016                  ERROR   16              ;REPORT NEXT ERROR
6223  017706  000770                  BR      64$             ;LOOP
6224  017710                  65$:
6225
6226  017710                  1$:
6227                                  ;;*****************************************************************
6228                                  ;*TEST 53       TWO SECTOR READ DATA (PART 2)
6229                                  ;*
6230                                  ;*      ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
6231                                  ;*      TRACK 0, SECTOR 23.  VERIFY THAT CORRECT DATA IS
6232                                  ;*      READ AND A MID-TRANSFER SEEK DOES NOT OCCUR.
6233                                  ;*
6234                                  ;*      NOTE:   TWO SECTOR WRITE DATA (PART 2) MUST BE
6235                                  ;*              EXECUTED BEFORE THIS TEST.
6236                                  ;*
6237                                  ;;*****************************************************************
6238  017710  000004          TST53:  SCOPE
6239  017712  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
6240  017720  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
6241  017722  104003                  ERROR   3               ;BAD INIT ERROR
6242
6243                                  ;GENERATE SAME DATA AS USED IN TWO SECTOR WRITE (PART 2)
6244
6245  017724  004437  042260          JSR     R4,GENCOM       ;GENERATE DATA
6246  017730  000016                  16                      ;PATTERN 16
6247  017732  001000                  1000                    ;1000 WORDS
6248
6249  017734  004437  042260          JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
6250  017740  002007                  2007
6251  017742  001000                  1000
6252  017744  004437  035674          JSR     R4,LRLOAD       ;LOAD "L" REGS
6253  017750  000121                  RDDATA                  ;RDDATA
6254  017752  177000                  -1000                   ;-1000 WORDS
```

```
6255  017754  070414                        IBUFF                   ;IBUFF IS BUFF ADDRESS
6256  017756     023                        .BYTE   23              ;SECTOR 23
6257  017757     000                        .BYTE   0               ;TRACK 0
6258  017760  000312                        312                     ;CYLINDER 312
6259
6260  017762  104417                        TLOADRK                 ;LOAD RK REGS
6261  017764  104430                        TWAT96                  ;WAIT FOR INTERRUPT
6262  017766  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6263
6264  017770  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6265  017772  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6266
6267  017774  004437  042260                JSR     R4,GENCOM       ;COMPARE DATA
6268  020000  100000                        100000
6269  020002  001000                        1000                    ;1000 WORDS
6270  020004  000413                        BR      1$              ;NO MISCOMPARES-SKIP
6271  020006  104015                        ERROR   15              ;REPORT 1ST ERROR
6272
6273  020010  013700  001634                MOV     ERRLMT,R0       ;GET ERROR LIMIT
6274  020014  005300                64$:    DEC     R0              ;DECREMENT COUNT
6275  020016  001406                        BEQ     65$             ;IF ZERO - EXIT
6276  020020  004437  042260                JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
6277  020024  040000                        40000
6278  020026  000402                        BR      65$             ;NO MORE ERRORS - EXIT
6279  020030  104016                        ERROR   16              ;REPORT NEXT ERROR
6280  020032  000770                        BR      64$             ;LOOP
6281  020034                        65$:
6282
6283  020034                        1$:
6284                                ;;*****************************************************************
6285                                ;*TEST 54       TWO SECTOR READ DATA (PART 3)
6286                                ;*
6287                                ;*      ISSUE A READ DATA OF 401 WORDS TO CYLINDER 312,
6288                                ;*      TRACK 0, SECTOR 10.  VERIFY THAT ALL 401 WORDS
6289                                ;*      ARE PLACED IN MEMORY.
6290                                ;*
6291                                ;*      NOTE:   TWO SECTOR WRITE DATA (PART 3) MUST BE
6292                                ;*              EXECUTED BEFORE THIS TEST.
6293                                ;*
6294                                ;;*****************************************************************
6295  020034  000004                TST54:  SCOPE
6296  020036  012737  000012  001262        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
6297  020044  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
6298  020046  104003                        ERROR   3               ;BAD INIT ERROR
6299
6300                                ;       GENERATE SAME DATA AS USED IN TWO SECTOR WRITE (PART 3)
6301
6302  020050  004437  042260                JSR     R4,GENCOM       ;GENERATE DATA
6303  020054  000002                        2                       ;PATTERN 2
6304  020056  000401                        401                     ;401 WORDS
6305
6306  020060  004437  042260                JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
6307  020064  002007                        2007
6308  020066  001000                        1000
6309
6310  020070  004437  035674                JSR     R4,LRLOAD       ;LOAD "L" REGS
```

```
6311  020074  000121                  RDDATA                  ;RDDATA
6312  020076  177377                  -401                    ;-401 WORDS
6313  020100  070414                  IBUFF                   ;IBUFF IS BUFF ADDRESS
6314  020102     010                  .BYTE   10              ;SECTOR 10
6315  020103     000                  .BYTE   0               ;TRACK 0
6316  020104  000312                  312                     ;CYLINDER 312
6317
6318  020106  104417                  TLOADRK                 ;LOAD RK REGS
6319  020110  104430                  TWAT96                  ;WAIT FOR INTERRUPT
6320  020112  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6321
6322  020114  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6323  020116  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6324
6325  020120  004437  042260          JSR     R4,GENCOM       ;COMPARE DATA
6326  020124  100000                  100000
6327  020126  000401                  401                     ;401 WORDS
6328  020130  000413                  BR      1$              ;NO MISCOMPARES-SKIP
6329  020132  104015                  ERROR   15              ;PRINT FIRST ERROR
6330
6331  020134  013700  001634          MOV     ERRLMT,R0       ;GET ERROR LIMIT
6332  020140  005300          64$:    DEC     R0              ;DECREMENT COUNT
6333  020142  001406                  BEQ     65$             ;IF ZERO - EXIT
6334  020144  004437  042260          JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
6335  020150  040000                  40000
6336  020152  000402                  BR      65$             ;NO MORE ERRORS - EXIT
6337  020154  104016                  ERROR   16              ;REPORT NEXT ERROR
6338  020156  000770                  BR      64$             ;LOOP
6339  020160                  65$:
6340  020160                  1$:
6341                          ;;*****************************************************************
6342                          ;*TEST 55       MID-TRANSFER SEEK ON READ (PART 1)
6343                          ;*
6344                          ;*      ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
6345                          ;*      TRACK 0, SECTOR 25.  VERIFY THAT CORRECT DATA IS
6346                          ;*      READ AND A MID-TRANSFER SEEK DOES OCCUR.
6347                          ;*
6348                          ;*      NOTE:   MID-TRANSFER SEEK ON WRITE (PART 1) MUST BE
6349                          ;*              EXECUTED BEFORE THIS TEST.
6350                          ;*
6351                          ;;*****************************************************************
6352  020160  000004          TST55:  SCOPE
6353  020162  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
6354  020170  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
6355  020172  104003                  ERROR   3               ;BAD INIT ERROR
6356
6357                          ;       GENERATE SAME DATA AS USED IN MID TRANSFER SEEK ON WRITE (PART 1)
6358  020174  004437  042260          JSR     R4,GENCOM       ;GENERATE DATA
6359  020200  000003                  3                       ;PATTERN 3
6360  020202  001000                  1000                    ;1000 WORDS
6361
6362  020204  004437  042260          JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
6363  020210  002007                  2007
6364  020212  001000                  1000
6365
6366  020214  004437  035674          JSR     R4,LRLOAD       ;LOAD "L" REGS
```

```
6367  020220  000121                        RDDATA                  ;RDDATA
6368  020222  177000                        -1000                   ;-1000 WORDS
6369  020224  070414                        IBUFF                   ;IBUFF IS BUFF ADDRESS
6370  020226     025                        .BYTE   25              ;SECTOR 25
6371  020227     000                        .BYTE   0               ;TRACK 0
6372  020230  000312                        312                     ;CYLINDER 312
6373
6374  020232  104417                        TLOADRK                 ;LOAD RK REGS
6375  020234  104430                        TWAT96                  ;WAIT FOR INTERRUPT
6376  020236  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6377
6378  020240  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6379  020242  104004                        ERROF       ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6380
6381  020244  004437  042260                JSR     R4,GENCOM       ;COMPARE DATA
6382  020250  100000                        100000
6383  020252  001000                        1000                    ;1000 WORDS
6384  020254  000413                        BR      1$              ;NO MISCOMPARES-SKIP
6385  020256  104015                        ERROR   15              ;PRINT FIRST ERROR
6386
6387  020260  013700  001634                MOV     ERRLMT,R0       ;GET ERROR LIMIT
6388  020264  005300              64$:      DEC     R0              ;DECREMENT COUNT
6389  020266  001406                        BEQ     65$             ;IF ZERO - EXIT
6390  020270  004437  042260                JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
6391  020274  040000                        40000
6392  020276  000402                        BR      65$             ;NO MORE ERRORS - EXIT
6393  020300  104016                        ERROR   16              ;REPORT NEXT ERROR
6394  020302  000770                        BR      64$             ;LOOP
6395  020304              65$:
6396  020304              1$:
6397                      ;;*****************************************************************
6398                      ;*TEST 56       MID-TRANSFER SEEK ON READ (PART 2)
6399                      ;*
6400                      ;*      ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
6401                      ;*      TRACK 2, SECTOR 25.  VERIFY THAT CORRECT DATA IS
6402                      ;*      READ AND A MID-TRANSFER SEEK DOES OCCUR.
6403                      ;*
6404                      ;*      NOTE:   MID-TRANSFER SEEK ON WRITE (PART 2) MUST BE
6405                      ;*              EXECUTED BEFORE THIS TEST.
6406                      ;*
6407                      ;;*****************************************************************
6408  020304  000004      TST56:  SCOPE
6409  020306  012737  000012  001262        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
6410  020314  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
6411  020316  104003                        ERROR   3               ;BAD INIT ERROR
6412
6413                      ;       GENERATE SAME DATA AS USED IN MID TRANSFER SEEK ON WRITE (PART 2)
6414  020320  004437  042260                JSR     R4,GENCOM       ;GENERATE DATA
6415  020324  000004                        4                       ;PATTERN 4
6416  020326  001000                        1000                    ;1000 WORDS
6417
6418  020330  004437  042260                JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
6419  020334  002007                        2007
6420  020336  001000                        1000
6421
6422  020340  004437  035674                JSR     R4,LRLOAD       ;LOAD "L" REGS
```

```
6423  020344  000121                          RDDATA                  ;RDDATA
6424  020346  177000                          -1000                   ;-1000 WORDS
6425  020350  070414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
6426  020352     025                          .BYTE   25              ;SECTOR 25
6427  020353     002                          .BYTE   2               ;TRACK 2
6428  020354  000312                          312                     ;CYLINDER 312
6429
6430  020356  104417                          TLOADRK                 ;LOAD RK REGS
6431  020360  104430                          TWAT96                  ;WAIT FOR INTERRUPT
6432  020362  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6433
6434  020364  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6435  020366  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6436
6437  020370  004437  042260                  JSR     R4,GENCOM       ;COMPARE DATA
6438  020374  100000                          100000
6439  020376  001000                          1000                    ;1000 WORDS
6440  020400  000413                          BR      1$              ;NO MISCOMPARES-SKIP
6441  020402  104015                          ERROR   15              ;REPORT FIRST ERROR
6442
6443  020404  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
6444  020410  005300                  64$:    DEC     R0              ;DECREMENT COUNT
6445  020412  001406                          BEQ     65$             ;IF ZERO - EXIT
6446  020414  004437  042260                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
6447  020420  040000                          40000
6448  020422  000402                          BR      65$             ;NO MORE ERRORS - EXIT
6449  020424  104016                          ERROR   16              ;REPORT NEXT ERROR
6450  020426  000770                          BR      64$             ;LOOP
6451  020430                          65$:
6452  020430                          1$:
6453                                  ;;*****************************************************************
6454                                  ;*TEST 57       CYLINDER ADDRESS OVERFLOW (PART 1)
6455                                  ;*
6456                                  ;*      ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632,
6457                                  ;*      TRACK 2, SECTOR 25.  MAKE SURE CYLINDER ADDRESS
6458                                  ;*      OVERFLOW ERROR DOES NOT OCCUR.
6459                                  ;*      CYLINDER 1456 IS USED FOR THE RK07.
6460                                  ;*
6461                                  ;;*****************************************************************
6462  020430  000004                  TST57:  SCOPE
6463  020432  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
6464  020440  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
6465  020442  104003                          ERROR   3               ;BAD INIT ERROR
6466
6467  020444  013737  036120  020466          MOV     LSTCYL,1$
6468  020452  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
6469  020456  000121                          RDDATA
6470  020460  177400                          -400                    ;400 WORDS
6471  020462  070414                          IBUFF                   ;IBUFF IS BUFF ADDR
6472  020464     025                          .BYTE   25              ;SECTOR 25
6473  020465     002                          .BYTE   2               ;TRK 2
6474  020466  000000                  1$:     0                       ;CYL 632/1456
6475
6476  020470  104417                          TLOADRK                 ;LOAD RK REGS
6477  020472  104434                          TWAT159                 ;WAIT FOR INTERRUPT
6478  020474  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
```

```
6479
6480  020476  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6481  020500  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6482                                  ;;*****************************************************************
6483                                  ;*TEST 60       CYLINDER ADDRESS OVERFLOW (PART 2)
6484                                  ;*
6485                                  ;*      ISSUE A READ DATA OF 401 WORDS TO CYLINDER 632,
6486                                  ;*      TRACK 2, SECTOR 25.  MAKE SURE CYLINDER ADDRESS
6487                                  ;*      OVERFLOW ERROR DOES OCCUR.
6488                                  ;*      CYLINDER 1456 IS USED FOR THE RK07.
6489                                  ;*
6490                                  ;;*****************************************************************
6491  020502  000004                  TST60:  SCOPE
6492  020504  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
6493  020512  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
6494  020514  104003                          ERROR   3               ;BAD INIT ERROR
6495  020516  013737  036120  020540          MOV     LSTCYL,1$
6496  020524  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
6497  020530  000121                          RDDATA
6498  020532  177377                          -401                    ;401 WORDS
6499  020534  070414                          IBUFF                   ;IBUFF IS BUFF ADDR
6500  020536     025                          .BYTE   25              ;SEC 25
6501  020537     002                          .BYTE   2               ;TRK 2
6502  020540  000000                  1$:     0                       ;CYL 632/1456
6503
6504  020542  104417                          TLOADRK                 ;LOAD RK REGS
6505  020544  104434                          TWAT159                 ;WAIT FOR INTERRUPT
6506  020546  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6507
6508  020550  104422                          TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
6509  020552  001000                          COERR                   ;CYLINDER OVERFLOW
6510  020554  000000                          0
6511  020556  000000                          0
6512  020560  104004                          ERROR   4; OR 5,6,7     ;REPORT ANY DISCREPANCIES
6513
6514
6515                                  .SBTTL  **18 BIT DATA TRANSFER TESTS
6516
6517                                  ;;*****************************************************************
6518                                  ;*TEST 61       FORMAT IN 24 SECTOR FORMAT
6519                                  ;*
6520                                  ;*      FORMAT CYLINDER 312, TRACK 0, AND TRACK 1 FOR 24 SECTOR
6521                                  ;*      FORMAT.  VERIFY FORMAT AND THAT DATA LATE DID NOT
6522                                  ;*      OCCUR WITH WRITE HEADER ON IN READING DATA BUFFER
6523                                  ;*      AFTER READ HEADER.
6524                                  ;*
6525                                  ;;*****************************************************************
6526  020562  000004                  TST61:  SCOPE
6527  020564  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
6528  020572  012737  000312  001676          MOV     #312,REFMT      ;SET REFORMAT SWITCH
6529  020600  012737  020610  001110          MOV     #1$,$LPERR      ;SET LOCAL LOOP ON ERROR
6530  020606  005001                          CLR     R1              ;CLEAR R1 FOR TRACK COUNTER
6531  020610                          1$:
6532  020610  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
6533  020612  104003                          ERROR   3               ;BAD INIT ERROR
6534  020614  012737  010127  001600          MOV     #WRHEAD!CFMT,L.CS1  ;SET UP FOR WRITE HEADER 24(8) SECTOR MODE
```

```
6535  020622  053737  001720  001600          BIS     DTYPE,L.CS1     ;SET DRV TYP
6536  020630  013737  001626  001610          MOV     DRVNUM,L.CS2    ;SET DRIVE NUMBER
6537  020636  012737  000074  001602          MOV     #74,L.WC        ;SET WORD COUNT
6538  020644  110137  001607                  MOVB    R1,L.DT         ;LOAD TRACK ADDRESS
6539  020650  012737  072414  001604          MOV     #OBUFF,L.BA     ;       SET BUS ADDRESS
6540  020656  012737  000312  001614          MOV     #312,L.DCYL     ;     CYLINDER ADDRESS
6541
6542  020664  004437  042260                  JSR     R4,GENCOM       ;GENERATE HEADER
6543  020670  001200                          1200                    ;INCLUDE BAD SECTOR BITS
6544
6545  020672  104417                          TLOADRK                 ;LOAD RK REGS
6546  020674  104434                          TWAT159                 ;WAIT FOR INTERRUPT
6547  020676  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6548
6549  020700  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6550  020702  104004                          ERROR   4  ,OR 5, 6, 7  ;REPORT ALL ERRORS
6551
6552  020704  104415                          SCOP1                   ;LOCAL LOOP ON ERROR TO 1$
6553
6554  020706  005701                          TST     R1              ;R1 POINTING TO TRACK 0
6555  020710  001002                          BNE     2$              ;NO-SKIP
6556  020712  005201                          INC     R1              ;BUMP TO TRACK 1
6557  020714  000735                          BR      1$              ;LOOP
6558  020716  012737  020726  001110  2$:     MOV     #3$,$LPERR      ;SET LOCAL LOOP ON ERROR
6559  020724  005001                          CLR     R1              ;CLEAR TRACK POINTER
6560  020726                          3$:
6561  020726  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
6562  020730  104003                          ERROR   3               ;BAD INIT ERROR
6563  020732  012737  010125  001600          MOV     #RDHEAD!CFMT,L.CS1  ;LOAD READ 24(8) SECTOR FORMAT
6564  020740  013737  001626  001610          MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
6565  020746  110137  001607                  MOVB    R1,L.DT         ;LOAD TRACK
6566  020752  012737  000312  001614          MOV     #312,L.DCYL     ;LOAD CYLINDER
6567
6568  020760  004437  036362                  JSR     R4,RDSTHD       ;GO READ STANDARD HEADER
6569  020764  104421                          TCHKOP                  ;RETURN IF CERR W/O DATA LATE SET
6570  020766  104004                          ERROR   4; OR 5,6,7     ;REPORT ALL OTHER ERRORS
6571  020770  104013                          ERROR   13              ;REPORT DATA LATE
6572  020772  104002                          ERROR   2               ;REPORT ''OPERATION TO SLOW'' OR ''HEADER
6573                                                                  ;0 NOT FOUND'' MESSAGE
6574
6575  020774  104415                          SCOP1                   ;LOCAL LOOP TO 3$ ON ERROR
6576  020776  004437  042260                  JSR     R4,GENCOM       ;GENERATE & COMPARE HEADERS
6577  021002  101200                          101200                  ;INCLUDING BAD SECTOR LISTS
6578  021004  000413                          BR      4$              ;NO MISCOMPARES-SKIP
6579  021006  104015                          ERROR   15              ;REPORT FIRST MISCOMPARE
6580
6581  021010  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
6582  021014  005300                  64$:    DEC     R0              ;DECREMENT COUNT
6583  021016  001406                          BEQ     65$             ;IF ZERO - EXIT
6584  021020  004437  042260                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
6585  021024  040000                          40000
6586  021026  000402                          BR      65$             ;NO MORE ERRORS - EXIT
6587  021030  104016                          ERROR   16              ;REPORT NEXT ERROR
6588  021032  000770                          BR      64$             ;LOOP
6589  021034                          65$:
6590
```

```
6591  021034  104415                  4$:     SCOP1                   ;LOCAL LOOP TO 3$
6592  021036  005701                          TST     R1              ;POINTING TO TRACK 1
6593  021040  001002                          BNE     5$              ;NO-EXIT
6594  021042  005201                          INC     R1              ;BUMP TO TRACK 1
6595  021044  000730                          BR      3$              ;LOOP
6596
6597  021046                          5$:
6598                                  ;;*****************************************************************
6599                                  ;*TEST 62       24 SECTOR FORMAT DATA TRANSFER (PART 1)
6600                                  ;*
6601                                  ;*      ISSUE A WRITE DATA OF 400 WORDS IN 24 SECTOR FORMAT
6602                                  ;*      TO CYLINDER 312, TRACK 0, SECTOR 0.  READ SECTOR BACK
6603                                  ;*      AND MAKE SURE IT IS CORRECT.
6604                                  ;*
6605                                  ;;*****************************************************************
6606  021046  000004                  TST62:  SCOPE
6607  021050  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
6608  021056  012737  000312  001676          MOV     #312,REFMT      ;SET REFORMAT SWITCH
6609  021064  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
6610  021066  104003                          ERROR   3               ;BAD INIT ERROR
6611
6612  021070  004437  042260                  JSR     R4,GENCOM       ;GENERATE DATA
6613  021074  000013                          13                      ;PATTERN 13
6614  021076  000400                          400                     ;400 WORDS
6615
6616  021100  004437  042260                  JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
6617  021104  002007                          2007
6618  021106  000400                          400
6619
6620  021110  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
6621  021114  010123                          WRDATA!CFMT                     ;WRDATA!CFMT
6622  021116  177400                          -400                    ;-400 WORDS
6623  021120  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
6624  021122     000                          .BYTE   0               ;SECTOR 0
6625  021123     000                          .BYTE   0               ;TRACK 0
6626  021124  000312                          312                     ;CYLINDER 312
6627
6628  021126  104417                          TLOADRK                 ;LOAD RK REGS
6629  021130  104430                          TWAT96                  ;WAIT FOR INTERRUPT
6630  021132  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6631
6632  021134  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6633  021136  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6634
6635  021140  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
6636  021144  010121                          RDDATA!CFMT                     ;RDDATA!CFMT
6637  021146  177400                          -400                    ;-400 WORDS
6638  021150  070414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
6639  021152     000                          .BYTE   0               ;SECTOR 0
6640  021153     000                          .BYTE   0               ;TRACK 0
6641  021154  000312                          312                      CYLINDER 312
6642
6643  021156  104417                          TLOADRK                 ;LOAD RK REGS
6644  021160  104424                          TWAT32                  ;WAIT FOR INTERRUPT
6645  021162  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6646
```

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

                                        JSR     R4,GENCOM       ;COMPARE DATA
                                        100000
                                        400                     ;400 WORDS
                                        BR      1$              ;NO MISCOMPARES-SKIP
                                        ERROR   15              ;REPORT 1ST ERROR

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

                                1$:


                                ;;*****************************************************************
                                ;*TEST 63       24 SECTOR FORMAT DATA TRANSFER (PART 2)
                                ;*
                                ;*      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.
                                ;*
                                ;;*****************************************************************
                                TST63:  SCOPE
                                        MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
                                        MOV     #312,REFMT      ;SET REFORMAT SWITCH
                                        JSR     PC,OPTTST       ;SET UP OPTIONS
                                        TSSINIT                 ;CLEAR SUBSYSTEM
                                        ERROR   3               ;BAD INIT ERROR

                                        JSR     R4,GENCOM       ;GENERATE DATA
                                        15                      ;PATTERN 15
                                        1000                    ;1000 WORDS

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

                                        JSR     R4,LRLOAD       ;LOAD "L" REGS
                                        WRDATA!CFMT                     ;WRDATA!CFMT
                                        -1000                   ;-1000 WORDS
                                        OBUFF                   ;OBUFF IS BUFF ADDRESS
                                        .BYTE   23              ;SECTOR 23
                                        .BYTE   0               ;TRACK 0
                                        312                     ;CYLINDER 312

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

```
 6703  021320  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 6704
 6705  021322  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 6706  021324  104004                    ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
 6707
 6708  021326  004437  035674            JSR     R4,LRLOAD       ;LOAD "L" REGS
 6709  021332  010121                    RDDATA!CFMT                     ;RDDATA!CFMT
 6710  021334  177000                    -1000                   ;-1000 WORDS
 6711  021336  070414                    IBUFF                   ;IBUFF IS BUFF ADDRESS
 6712  021340     023                    .BYTE   23              ;SECTOR 23
 6713  021341     000                    .BYTE   0               ;TRACK 0
 6714  021342  000312                    312                     ;CYLINDER 312
 6715
 6716  021344  104417                    TLOADRK                 ;LOAD RK REGS
 6717  021346  104426                    TWAT64                  ;WAIT FOR INTERRUPT
 6718  021350  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 6719
 6720  021352  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 6721  021354  104004                    ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
 6722
 6723  021356  004437  042260            JSR     R4,GENCOM       ;COMPARE DATA
 6724  021362  100000                    100000
 6725  021364  001000                    1000                    ;1000 WORDS
 6726  021366  000413                    BR      1$              ;NO MISCOMPARES-SKIP
 6727  021370  104015                    ERROR   15              ;REPORT FIRST ERROR
 6728
 6729  021372  013700  001634            MOV     ERRLMT,R0       ;GET ERROR LIMIT
 6730  021376  005300                64$:DEC     R0              ;DECREMENT COUNT
 6731  021400  001406                    BEQ     65$             ;IF ZERO - EXIT
 6732  021402  004437  042260            JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
 6733  021406  040000                    40000
 6734  021410  000402                    BR      65$             ;NO MORE ERRORS - EXIT
 6735  021412  104016                    ERROR   16              ;REPORT NEXT ERROR
 6736  021414  000770                    BR      64$             ;LOOP
 6737  021416                        65$:
 6738
 6739  021416                        1$:
 6740
 6741                                  .SBTTL  **SPECIAL DATA TRANSFER TESTS
 6742
 6743                                  ;;*****************************************************************
 6744                                  ;*TEST 64       MULTI-SECTOR DATA TRANSFER AND BSE
 6745                                  ;*
 6746                                  ;*      FORMAT CYLINDER 312, TRACK 0 IN 26 SECTOR FORMAT WITH
 6747                                  ;*      SECTOR 1 MARKED BAD.  ISSUE A WRITE DATA OF 1000 WORDS
 6748                                  ;*      TO CYLINDER 312, TRACK 0, SECTOR 0.  MAKE SURE BAD SECTOR
 6749                                  ;*      ERROR SETS AND RKDA IS CORRECT.  READ SECTOR 0 AND
 6750                                  ;*      MAKE SURE IT IS CORRECT.
 6751                                  ;*
 6752                                  ;*      ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0,
 6753                                  ;*      SECTOR 0.  MAKE SURE BAD SECTOR ERROR SETS AND THE
 6754                                  ;*      PREVIOUS SECTOR IS LOADED CORRECTLY INTO MEMORY.
 6755                                  ;*
 6756                                  ;;*****************************************************************
 6757  021416  000004                TST64:  SCOPE
 6758  021420  012737  000012  001262        MOV     #10.,$TIMES     ;:DO 10. ITERATIONS
```

```
 6759  021426  012737  000312  001676          MOV     #312,REFMT      ;SET REFORMAT SWITCH
 6760  021434  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
 6761  021436  104003                          ERROR   3               ;BAD INIT ERROR
 6762
 6763  021440  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
 6764  021444  000127                          WRHEAD                  ;WRHEAD
 6765  021446  177676                          -102                    ;-102 WORDS
 6766  021450  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
 6767  021452     000                          .BYTE   0               ;SECTOR 0
 6768  021453     000                          .BYTE   0               ;TRACK 0
 6769  021454  000312                          312                     ;CYLINDER 312
 6770
 6771  021456  004437  042260                  JSR     R4,GENCOM       ;BUILD HEADERS
 6772  021462  000600                          600
 6773
 6774  021464  042737  040000  072424          BIC     #BIT14,OBUFF+10 ;MARK SECTOR 1 BAD
 6775  021472  042737  040000  072426          BIC     #BIT14,OBUFF+12 ;CORRECT HURC
 6776
 6777  021500  104417                          TLOADRK                 ;LOAD RK REGS
 6778  021502  104431                          TWAT112                 ;WAIT FOR INTERRUPT
 6779  021504  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 6780
 6781  021506  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 6782  021510  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 6783
 6784  021512  004437  042260                  JSR     R4,GENCOM       ;GENERATE DATA
 6785  021516  000016                          16                      ;PATTERN 16
 6786  021520  001000                          1000                    ;1000 WORDS
 6787
 6788  021522  004437  042260                  JSR     R4,GENCOM       ;CLEAR IBUFF TO ALL ONES
 6789  021526  002007                          2007
 6790  021530  001000                          1000
 6791
 6792  021532  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
 6793  021536  000123                          WRDATA                  ;WRDATA
 6794  021540  177000                          -1000                   ;-1000 WORDS
 6795  021542  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
 6796  021544     000                          .BYTE   0               ;SECTOR 0
 6797  021545     000                          .BYTE   0               ;TRACK 0
 6798  021546  000312                          312                     ;CYLINDER 312
 6799
 6800  021550  104417                          TLOADRK                 ;LOAD RK REGS
 6801  021552  104424                          TWAT32                  ;WAIT FOR INTERRUPT
 6802  021554  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 6803
 6804  021556  104422                          TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERR
 6805  021560  000000                          0
 6806  021562  000100                          BSERR                   ;BAD SECTOR ERROR
 6807  021564  000000                          0
 6808  021566  104004                          ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES
 6809  021570  005037  050002                  CLR     GRP4ER          ;CLEAR GROUP 4 ERRORS
 6810  021574  004437  037642                  JSR     R4,CHKCTS       ;CHECK CYL, TRK, SECT CORRECT AFTER ABORTED WRITE
 6811  021600  032737  000020  050002          BIT     #TRKERR,GRP4ER  ;TRK IN ERROR?
 6812  021606  001416                          BEQ     1$              ;NO-SKIP
 6813  021610  012737  054170  001450          MOV     #EM13,EM10N     ;"TRACK ADDRESS INCORRECT"
 6814  021616  013737  047756  001202          MOV     EXPTRK,$REG10   ;EXPECTED VALUE
```

```
6815  021624  013737  047770  001204          MOV     REALTRK,$REG11  ;REAL VALUE
6816  021632  012737  050467  061002          MOV     #OPER37,DF010A  ;"AFTER WRITE DATA TERMINATED WITH BSE"
6817  021640  104010                          ERROR   10
6818  021642  000527                          BR      5$              ;EXIT
6819
6820  021644  032737  000040  050002  1$:     BIT     #SECERR,GRP4ER  ;SECTOR IN ERROR?
6821  021652  001422                          BEQ     3$              ;NO-SKIP
6822  021654  012737  054220  001450          MOV     #EM14,EM10N     ;"SECTOR ADDRESS INCORRECT"
6823  021662  012737  050467  061002          MOV     #OPER37,DF010A  ;"AFTER WRITE DATA ABORTED WITH BSE"
6824  021670  013737  047754  001202          MOV     EXPSEC,$REG10   ;EXPECTED VALUE
6825  021676  013737  047772  001204          MOV     REALSEC,$REG11  ;REAL VALUE
6826  021704  104010                          ERROR   10
6827  021706  000505                          BR      5$              ;EXIT
6828  021710  104415                          SCOP1                   ;LOCAL LOOP TO BEGINNING OF TEST
6829  021712  012737  021720  001110          MOV     #3$,$LPERR      ;SET LOCAL LOOP ON ERROR
6830  021720                          3$:
6831  021720  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
6832  021722  104003                          ERROR   3               ;BAD INIT ERROR
6833  021724  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
6834  021730  000121                          RDDATA                  ;RDDATA
6835  021732  177400                          -400                    ;-400 WORDS
6836  021734  070414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
6837  021736     000                          .BYTE   0               ;SECTOR 0
6838  021737     000                          .BYTE   0               ;TRACK 0
6839  021740  000312                          312                     ;CYLINDER 312
6840
6841  021742  104417                          TLOADRK                 ;LOAD RK REGS
6842  021744  104424                          TWAT32                  ;WAIT FOR INTERRUPT
6843  021746  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6844
6845  021750  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6846  021752  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6847
6848  021754  004437  042260                  JSR     R4,GENCOM       ;COMPARE DATA
6849  021760  100000                          100000
6850  021762  000400                          400                     ;400 WORDS
6851  021764  000413                          BR      4$              ;NO MISCOMPARES-EXIT
6852  021766  104015                          ERROR   15              ;REPORT FIRST ERROR
6853
6854  021770  013700  001634                  MOV     ERRLMT,R0       ;GET ERROR LIMIT
6855  021774  005300                  64$:    DEC     R0              ;DECREMENT COUNT
6856  021776  001406                          BEQ     65$             ;IF ZERO - EXIT
6857  022000  004437  042260                  JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
6858  022004  040000                          40000
6859  022006  000402                          BR      65$             ;NO MORE ERRORS - EXIT
6860  022010  104016                          ERROR   16              ;REPORT NEXT ERROR
6861  022012  000770                          BR      64$             ;LOOP
6862  022014                          65$:
6863
6864  022014  004437  042260          4$:     JSR     R4,GENCOM       ;CLEAR IBUFF
6865  022020  002007                          2007
6866  022022  001000                          1000
6867
6868  022024  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
6869  022030  000121                          RDDATA                  ;RDDATA
6870  022032  177000                          -1000                   ;-1000 WORDS
```

```
 6871  022034  070414                        IBUFF                   ;IBUFF IS BUFF ADDRESS
 6872  022036     000                        .BYTE   0               ;SECTOR 0
 6873  022037     000                        .BYTE   0               ;TRACK 0
 6874  022040  000312                        312                     ;CYLINDER 312
 6875
 6876  022042  104417                        TLOADRK                 ;LOAD RK REGS
 6877  022044  104424                        TWAT32                  ;WAIT FOR INTERRUPT
 6878  022046  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 6879
 6880  022050  104422                        TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
 6881  022052  000000                        0
 6882  022054  000100                        BSERR                   ;BAD SECTOR ERROR
 6883  022056  000000                        0
 6884  022060  104004                        ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES
 6885
 6886  022062  004437  042260                JSR     R4,GENCOM       ;COMPARE DATA AGAIN
 6887  022066  100000                        100000
 6888  022070  000400                        400                     ;400 WORDS
 6889  022072  000413                        BR      5$              ;NO MISCOMPARES
 6890  022074  104015                        ERROR   15              ;REPORT FIRST ERROR
 6891
 6892  022076  013700  001634                MOV     ERRLMT,R0       ;GET ERROR LIMIT
 6893  022102  005300                66$:    DEC     R0              ;DECREMENT COUNT
 6894  022104  001406                        BEQ     67$             ;IF ZERO - EXIT
 6895  022106  004437  042260                JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
 6896  022112  040000                        40000
 6897  022114  000402                        BR      67$             ;NO MORE ERRORS - EXIT
 6898  022116  104016                        ERROR   16              ;REPORT NEXT ERROR
 6899  022120  000770                        BR      66$             ;LOOP
 6900  022122                        67$:
 6901
 6902  022122                        5$:
 6903                                ;;*****************************************************************
 6904                                ;*TEST 65       FORMAT TEST
 6905                                ;*
 6906                                ;*      FORMAT CYLINDER 312, TRACKS 0 AND 1 IN 26 SECTOR FORMAT.
 6907                                ;*      MAKE SURE NO ERRORS SET.  READ SECTORS 0-25 AND MAKE
 6908                                ;*      SURE DATA CHECK DOES NOT OCCUR.
 6909                                ;*
 6910                                ;;*****************************************************************
 6911  022122  000004                TST65:  SCOPE
 6912  022124  012737  000001  001262        MOV     #1,$TIMES       ;;DO 1 ITERATION
 6913  022132  005000                        CLR     R0              ;CLEAR TRACK COUNTER
 6914  022134  012737  022142  001110        MOV     #1$,$LPERR      ;SET LOCAL LOOP
 6915
 6916  022142                        1$:
 6917  022142  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
 6918  022144  104003                        ERROR   3               ;BAD INIT ERROR
 6919
 6920  022146  013737  001626  001610        MOV     DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
 6921  022154  012737  000127  001600        MOV     #WRHEAD,L.CS1   ;LOAD WRITE HEADER
 6922  022162  053737  001720  001600        BIS     DTYPE,L.CS1     ;SET DRV TYP
 6923  022170  110037  001607                MOVB    R0,L.DT         ;LOAD DESIRED TRACK FROM TRACK COUNTER
 6924  022174  012737  072414  001604        MOV     #OBUFF,L.BA     ;LOAD BUS ADDRESS
 6925  022202  012737  177676  001602        MOV     #-102,L.WC      ;       WORD COUNT
 6926  022210  012737  000312  001614        MOV     #312,L.DCYL     ;       CYLINDER
```

```
6927
6928  022216  004437  042260              JSR     R4,GENCOM       ;BUILD HEADER
6929  022222  001200                      1200                    ;WITH BSE FLAGGED
6930
6931  022224  104417                      TLOADRK                 ;LOAD RK REGS
6932  022226  104431                      TWAT112                 ;WAIT FOR INTERRUPT
6933  022230  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6934
6935  022232  104421                      TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6936  022234  104004                      ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
6937
6938  022236  104415                      SCOP1                   ;LOCAL LOOP TO 1$
6939
6940  022240  005700                      TST     R0              ;R0 AT ZERO?
6941  022242  001002                      BNE     2$              ;NO-EXIT
6942  022244  005200                      INC     R0              ;BUMP COUNTER
6943  022246  000735                      BR      1$              ;LOOP
6944  022250  005000              2$:     CLR     R0              ;CLEAR SECTOR COUNTER
6945  022252  012737  022260  001110      MOV     #113$,$LPERR    ;SET LOCAL LOOP ON ERROR
6946
6947  022260                      113$:
6948  022260  104416                      TSSINIT                 ;CLEAR SUBSYSTEM
6949  022262  104003                      ERROR   3               ;BAD INIT ERROR
6950  022264                      3$:
6951  022264  004437  035674              JSR     R4,LRLOAD       ;LOAD "L" REGS
6952  022270  000121                      RDDATA                  ;RDDATA
6953  022272  177400                      -400                    ;-400 WORDS
6954  022274  070414                      IBUFF                   ;IBUFF IS BUFF ADDRESS
6955  022276     000                      .BYTE   0               ;SECTOR 0
6956  022277     000                      .BYTE   0               ;TRACK 0
6957  022300  000312                      312                     ;CYLINDER 312
6958
6959  022302  110037  001606              MOVB    R0,L.DS         ;LOAD SECTOR COUNTER INTO DESIRED SECTOR
6960
6961  022306  104417                      TLOADRK                 ;LOAD RK REGS
6962  022310  104424                      TWAT32                  ;WAIT FOR INTERRUPT
6963  022312  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
6964
6965  022314  104421                      TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
6966  022316  104004                      ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6967
6968  022320  104415                      SCOP1                   ;LOCAL LOOP TO 3$ ON ERROR
6969
6970  022322  022700  000024              CMP     #24,R0          ;LAST SECTOR READ?
6971  022326  001402                      BEQ     4$              ;YES-EXIT
6972  022330  005200                      INC     R0              ;BUMP SECTOR COUNTER
6973  022332  000754                      BR      3$              ;LOOP
6974
6975  022334  005037  001676      4$:     CLR     REFMT           ;CLEAR REFORMAT SWITCH
6976
6977                              .SBTTL  **WRITE CHECK TESTS
6978
6979                              ;;*****************************************************************
6980                              ;*TEST 66       WRITE-CHECK WITH NO ERROR
6981                              ;*
6982                              ;*      WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH A KNOWN PATTERN.
```

```
                                        ;*      DO A WRITE-CHECK OF 400 WORDS.  MAKE SURE NO
                                        ;*      ERROR OCCURS.
                                        ;*
                                        ;;*****************************************************************
022340  000004                          TST66:  SCOPE
022342  012737  000012  001262                  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
022350  104416                                  TSSINIT                 ;CLEAR SUBSYSTEM
022352  104003                                  ERROR   3               ;BAD INIT ERROR

022354  004437  035674                          JSR     R4,LRLOAD       ;LOAD "L" REGS
022360  000123                                  WRDATA                  ;WRDATA
022362  177400                                  -400                    ;-400 WORDS
022364  072414                                  OBUFF                   ;OBUFF IS BUFF ADDRESS
022366     000                                  .BYTE   0               ;SECTOR 0
022367     000                                  .BYTE   0               ;TRACK 0
022370  000312                                  312                     ;CYLINDER 312

022372  004437  042260                          JSR     R4,GENCOM       ;GENERATE DATA
022376  000002                                  2                       ;PATTERN 2
022400  000400                                  400                     ;400 WORDS

022402  104417                                  TLOADRK                 ;LOAD RK REGS
022404  104430                                  TWAT96                  ;WAIT FOR INTERRUPT
022406  104002                                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

022410  104421                                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
022412  104004                                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

022414  004437  035674                          JSR     R4,LRLOAD       ;LOAD "L" REGS
022420  000131                                  WRTCHK                  ;WRTCHK
022422  177400                                  -400                    ;-400 WORDS
022424  072414                                  OBUFF                   ;OBUFF IS BUFF ADDRESS
022426     000                                  .BYTE   0               ;SECTOR 0
022427     000                                  .BYTE   0               ;TRACK 0
022430  000312                                  312                     ;CYLINDER 312

022432  104417                                  TLOADRK                 ;LOAD RK REGS
022434  104424                                  TWAT32                  ;WAIT FOR INTERRUPT
022436  104002                                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

022440  104421                                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
022442  104004                                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
                                        ;;*****************************************************************
                                        ;*TEST 67       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.
                                        ;*
                                        ;;*****************************************************************
022444  000004                  TST67:  SCOPE
022446  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
022454  012700  000001                  MOV     #BIT0,R0        ;SET LO ORDER BIT IN R0 FOR
                                                                ;CAUSING WRITE CHECK ERROR
022460  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
022462  104003                          ERROR   3               ;BAD INIT ERROR
022464  004437  042260                  JSR     R4,GENCOM       ;GENERATE DATA, ALL 0'S
022470  000001                          1
022472  000400                          400

022474  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
022500  000123                          WRDATA                  ;WRDATA
022502  177400                          -400                    ;-400 WORDS
022504  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
022506     000                          .BYTE   0               ;SECTOR 0
022507     000                          .BYTE   0               ;TRACK 0
022510  000312                          312                     ;CYLINDER 312

022512  104417                          TLOADRK                 ;LOAD RK REGS
022514  104430                          TWAT96                  ;WAIT FOR INTERRUPT
022516  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

022520  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
022522  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

022524  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
022530  000131                          WRTCHK                  ;WRTCHK
022532  177400                          -400                    ;-400 WORDS
022534  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
022536     000                          .BYTE   0               ;SECTOR 0
022537     000                          .BYTE   0               ;TRACK 0
022540  000312                          312                     ;CYLINDER 312

022542  104417                          TLOADRK                 ;LOAD RK REGS
022544  104424                          TWAT32                  ;WAIT FOR INTERRUPT
022546  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

022550  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
022552  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

022554  104415                          SCOP1                   ;LOCAL LOOP ON WRITE CHECK
022556  012737  022564  001110          MOV     #1$,$LPERR      ;SET LOCAL LOOP
022564  010037  072634          1$:     MOV     R0,OBUFF+220    ;CAUSE ERROR BIT IN BUFFER
022570  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
022572  104003                          ERROR   3               ;BAD INIT ERROR
022574  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
022600  000131                          WRTCHK                  ;WRTCHK
022602  177400                          -400                    ;-400 WORDS
022604  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
022606     000                          .BYTE   0               ;SECTOR 0
022607     000                          .BYTE   0               ;TRACK 0
022610  000312                          312                     ;CYLINDER 312
```

```
7095
7096  022612  104417                          TLOADRK                 ;LOAD RK REGS
7097  022614  104424                          TWAT32                  ;WAIT FOR INTERRUPT
7098  022616  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7099
7100  022620  104422                          TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
7101  022622  000000                          0
7102  022624  000004                          WCKERR                  ;WRITE CHECK ERROR
7103  022626  000000                          0
7104  022630  104004                          ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES
7105
7106  022632  104415                          SCOP1                   ;LOCAL LOOP ON ERROR TO 1$
7107
7108                                ;         NOTE:   THE WORD COUNT AND BUS ADDRESS CAN BE EITHER OF THREE
7109                                ;         VALUES AND BE CORRECT. THE DIFFERENCE IS CAUSED BY WHEN THE
7110                                ;         WCE OCCURRED. IF IT OCCURRED ON THE FIRST WORD OF A DOUBLE
7111                                ;         NPR CYCLE, WC AND BA WILL BE TWO PAST WHERE THE ERROR ACTUALLY
7112                                ;         OCCURRED. IF WCE OCCURRED ON A SINGLE NPR CYCLE OR THE LAST
7113                                ;         NPR CYCLE OF A DOUBLE CYCLE, WC AND BA CONTENTS ARE ONE PAST
7114                                ;         THE ACTUAL WORD WHERE THE ERROR WAS. IN SOME CASES, WC AND BA
7115                                ;         CAN BE THREE PAST WHERE THE ERROR WAS AND STILL BE ACCEPTABLE.
7116
7117
7118  022634  023727  001544  072636          CMP     T.BA,#OBUFF+222 ;CHECK BA HALT AT PROPER PLACE
7119  022642  001416                          BEQ     2$              ;YES-SKIP
7120  022644  101040                          BHI     6$              ;IF TO HI - SKIP
7121  022646  012737  054033  001450          MOV     #EM11,EM10N     ;"INCORRECT BA"
7122  022654  012737  072636  001202          MOV     #OBUFF+222,$REG10  ;GOOD VALUE
7123  022662  013737  001544  001204          MOV     T.BA,$REG11     ;BAD VALUE
7124  022670  012737  050523  061002          MOV     #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
7125  022676  104010                          ERROR   10
7126
7127  022700  023727  001542  177511  2$:     CMP     T.WC,#-267      ;CHECK WORD COUNT AT CORRECT VALUE
7128  022706  001461                          BEQ     3$              ;YES-SKIP
7129  022710  101037                          BHI     7$              ;IF HIGHER SKIP
7130  022712  012737  054006  001450          MOV     #EM10,EM10N     ;"INCORRECT WC"
7131  022720  012737  050523  061002          MOV     #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
7132  022726  012737  177511  001202          MOV     #-267,$REG10    ;GOOD VALUE
7133  022734  013737  001542  001204          MOV     T.WC,$REG11     ;ERROR VALUE
7134  022742  104010                          ERROR   10
7135  022744  000442                          BR      3$              ;EXIT
7136
7137  022746  023727  001544  072642  6$:     CMP     T.BA,#OBUFF+226 ;TEST IF BA AT HI SIDE
7138  022754  101415                          BLOS    7$              ;YES - SKIP
7139  022756  012737  054033  001450          MOV     #EM11,EM10N     ;SET MESSAGE
7140  022764  012737  072642  001202          MOV     #OBUFF+226,$REG10 ;GOOD VALUE
7141  022772  013737  001544  001204          MOV     T.BA,$REG11     ;ERROR VALUE
7142  023000  012737  050523  061002          MOV     #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
7143  023006  104010                          ERROR   10
7144
7145  023010  023727  001542  177513  7$:     CMP     T.WC,#-265      ;TEST IF WORD COUNT AT HI SIDE
7146  023016  101415                          BLOS    3$              ;YES - SKIP
7147  023020  012737  054006  001450          MOV     #EM10,EM10N     ;SET MESSAGE
7148  023026  012737  050523  061002          MOV     #OPER41,DF010A  ;"WC ABORTED WITH WCE"
7149  023034  012737  177513  001202          MOV     #-265,$REG10    ;GOOD VALUE
7150  023042  013737  001542  001204          MOV     T.WC,$REG11     ;ERROR VALUE
```

```
                                                ERROR   10

                                        3$:     SCOP1                   ;LOCAL LOOP ON ERROR TO 1$

023054  032700  100000                        BIT     #BIT15,R0       ;BIT 15 SET?
023060  001002                                BNE     4$              ;YES-EXIT
023062  006300                                ASL     R0              ;SHIFT ERROR BIT
023064  000637                                BR      1$              ;LOOP

023066                                  4$:
                                        ;;*****************************************************************
                                        ;*TEST 70       WRITE CHECK ERROR (PART 2)
                                        ;*
                                        ;*      WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH 17777
                                        ;*      IN ALL WORDS.  WRITE CHECK CYLINDER 312, TRACK 0,
                                        ;*      SECTOR 0 WITH THE SAME DATA EXCEPT WORD 120 HAS
                                        ;*      ONE OF THE FOLLWOING CONFIGURATIONS:
                                        ;*
                                        ;*      177776  177757  177377  167777
                                        ;*      177775  177737  176777  157777
                                        ;*      177773  177677  175777  137777
                                        ;*      177767  177577  173777  077777
                                        ;*
                                        ;*      MAKE SURE WRITE CHECK ERROR SET FOR EACH
                                        ;*      OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS
                                        ;*      AND WORD COUNT IS CORRECT.
                                        ;*
                                        ;;*****************************************************************
023066  000004                        TST70:  SCOPE
023070  012737  000012  001262                MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
023076  012700  177776                        MOV     #177776,R0      ;LOAD R0 FOR CAUSING WRITE CHECK ERROR

023102  104416                                TSSINIT                 ;CLEAR SUBSYSTEM
023104  104003                                ERROR   3               ;BAD INIT ERROR
023106  004437  042260                        JSR     R4,GENCOM       ;GENERATE DATA
023112  000007                                7                       ;ALL 1'S
023114  000400                                400                     ;400 WORDS

023116  004437  035674                        JSR     R4,LRLOAD       ;LOAD "L" REGS
023122  000123                                WRDATA                  ;WRDATA
023124  177400                                -400                    ;-400 WORDS
023126  072414                                OBUFF                   ;OBUFF IS BUFF ADDRESS
023130     000                                .BYTE   0               ;SECTOR 0
023131     000                                .BYTE   0               ;TRACK 0
023132  000312                                312                     ;CYLINDER 312

023134  104417                                TLOADRK                 ;LOAD RK REGS
023136  104430                                TWAT96                  ;WAIT FOR INTERRUPT
023140  104002                                ERROR   2               ;TO ^LOW/NOT COMPLETE ERROR

023142  104421                                TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
023144  104004                                ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

023146  004437  035674                        JSR     R4,LRLOAD       ;LOAD "L" REGS
023152  000131                                WRTCHK                  ;WRTCHK
023154  177400                                -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

                                        SCOP1           ;LOCAL LOOP TO START OF TEST
                                        MOV     #1$,$LPERR      ;SET LOCAL LOOP

                                1$:     MOV     R0,OBUFF+220    ;PUT WORD IN OBUFF TO CAUSE WCE
                                        TSSINIT                 ;CLEAR SUBSYSTEM
                                        ERROR   3               ;BAD INIT ERROR

                                        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

                                        TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
                                        0
                                        WCKERR                  ;WRITE CHECK ERROR
                                        0
                                        ERROR   4; OR 5,6,7     ;REPORT ALL DISCREPANCIES

                                        SCOP1                   ;LOCAL LOOP TO 1$

                                ;       NOTE:   THE WORD COUNT AND BUS ADDRESS CAN BE EITHER OF THREE
                                ;       VALUES AND BE CORRECT. THE DIFFERENCE IS CAUSED BY WHEN THE
                                ;       WCE OCCURRED. IF IT OCCURRED ON THE FIRST WORD OF A DOUBLE
                                ;       NPR CYCLE, WC AND BA WILL BE TWO PAST WHERE THE ERROR ACTUALLY
                                ;       OCCURRED. IF WCE OCCURRED ON A SINGLE NPR CYCLE OR THE LAST
                                ;       NPR CYCLE OF A DOUBLE CYCLE, WC AND BA CONTENTS ARE ONE PAST
                                ;       THE ACTUAL WORD WHERE THE ERROR WAS. IN SOME CASES, WC AND BA
                                ;       CAN BE THREE PAST WHERE THE ERROR WAS AND STILL BE ACCEPTABLE.

                                        CMP     T.BA,#OBUFF+222 ;CHECK BA HALT AT PROPER PLACE
                                        BEQ     2$              ;YES-SKIP
                                        BHI     6$              ;IF TO HI - SKIP
                                        MOV     #EM11,EM10N     ;"INCORRECT BA"
                                        MOV     #OBUFF+222,$REG10  ;GOOD VALUE
                                        MOV     T.BA,$REG11     ;BAD VALUE
                                        MOV     #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
```

```
7263  023320  104010                          ERROR   10
7264
7265  023322  023727  001542  177511  2$:     CMP     T.WC,#-267      ;CHECK WORD COUNT AT CORRECT VALUE
7266  023330  001461                          BEQ     3$              ;YES-SKIP
7267  023332  101037                          BHI     7$              ;IF HIGHER - SKIP
7268  023334  012737  054006  001450          MOV     #EM10,EM10N     ;"INCORRECT WC"
7269  023342  012737  050523  061002          MOV     #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
7270  023350  012737  177511  001202          MOV     #-267,$REG10    ;GOOD VALUE
7271  023356  013737  001542  001204          MOV     T.WC,$REG11     ;ERROR VALUE
7272  023364  104010                          ERROR   10
7273  023366  000442                          BR      3$              ;EXIT
7274
7275  023370  023727  001544  072642  6$:     CMP     T.BA,#OBUFF+226 ;TEST IF BA AT HI SIDE
7276  023376  101415                          BLOS    7$              ;YES - SKIP
7277  023400  012737  054033  001450          MOV     #EM11,EM10N     ;SET MESSAGE
7278  023406  012737  072642  001202          MOV     #OBUFF+226,$REG10 ;GOOD VALUE
7279  023414  013737  001544  001204          MOV     T.BA,$REG11     ;ERROR VALUE
7280  023422  012737  050523  061002          MOV     #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
7281  023430  104010                          ERROR   10
7282
7283  023432  023727  001542  177513  7$:     CMP     T.WC,#-265      ;TEST IF WORD COUNT AT HI SIDE
7284  023440  101415                          BLOS    3$              ;YES - SKIP
7285  023442  012737  054006  001450          MOV     #EM10,EM10N     ;SET MESSAGE
7286  023450  012737  050523  061002          MOV     #OPER41,DF010A  ;"WC ABORTED WITH WCE"
7287  023456  012737  177513  001202          MOV     #-265,$REG10    ;GOOD VALUE
7288  023464  013737  001542  001204          MOV     T.WC,$REG11     ;ERROR VALUE
7289  023472  104010                          ERROR   10
7290
7291  023474  104415              3$:         SCOP1
7292
7293  023476  032700  100000                  BIT     #BIT15,R0       ;BIT 15 SET? (ALL PATTERNS TESTED)
7294  023502  001002                          BNE     4$              ;YES-EXIT
7295  023504  006300                          ASL     R0              ;SHIFT FOR NEXT TEST
7296  023506  000637                          BR      1$              ;LOOP
7297
7298  023510              4$:
7299                      ;;*******************************************************************
7300                      ;*TEST 71        WRITE CHECK OF PARTIAL SECTOR
7301                      ;*
7302                      ;*      WRITE CYLINDER 312, TRACK 0, SECTOR WITH A KNOWN
7303                      ;*      CONFIGURATIONS.  ISSUE A WRITE CHECK COMMAND OF
7304                      ;*      110 WORDS MAKING SURE THE 111TH WORD IS
7305                      ;*      DIFFERENT THAN DATA ON DISK.  MAKE SURE
7306                      ;*      WRITE CHECK ERROR DOES NOT SET.
7307                      ;*
7308                      ;*******************************************************************
7309  023510  000004              TST71:  SCOPE
7310  023512  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
7311  023520  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7312  023522  104003                          ERROR   3               ;BAD INIT ERROR
7313
7314  023524  004437  042260                  JSR     R4,GENCOM       ;GENERATE DATA
7315  023530  000007                          7                       ;ALL 1'S
7316  023532  000400                          400                     ;400 WORDS
7317
7318  023534  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
```

```
7319  023540  000123                    WRDATA                  ;WRDATA
7320  023542  177400                    -400                    ;-400 WORDS
7321  023544  072414                    OBUFF                   ;OBUFF IS BUFF ADDRESS
7322  023546     000                    .BYTE   0               ;SECTOR 0
7323  023547     000                    .BYTE   0               ;TRACK 0
7324  023550  000312                    312                     ;CYLINDER 312
7325
7326  023552  104417                    TLOADRK                 ;LOAD RK REGS
7327  023554  104430                    TWAT96                  ;WAIT FOR INTERRUPT
7328  023556  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7329
7330  023560  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7331  023562  104004                    ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7332
7333  023564  005037  072636            CLR     OBUFF+222
7334
7335  023570  004437  035674            JSR     R4,LRLOAD       ;LOAD "L" REGS
7336  023574  000131                    WRTCHK                  ;WRTCHK
7337  023576  177670                    -110                    ;-110 WORDS
7338  023600  072414                    OBUFF                   ;OBUFF IS BUFF ADDRESS
7339  023602     000                    .BYTE   0               ;SECTOR 0
7340  023603     000                    .BYTE   0               ;TRACK 0
7341  023604  000312                    312                     ;CYLINDER 312
7342
7343  023606  104417                    TLOADRK                 ;LOAD RK REGS
7344  023610  104424                    TWAT32                  ;WAIT FOR INTERRUPT
7345  023612  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7346
7347  023614  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7348  023616  104004                    ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7349
7350                            .SBTTL  **MAXIMUM DATA TRANSFER AND CONTROLLER TIME OUT
7351
7352                            ;;*****************************************************************
7353                            ;*TEST 72       MAXIMUM DATA TRANSFER (PART 1)
7354                            ;*
7355                            ;*      IN THE FIRST PASS OF THE PROGRAM, THE HEADERS OF
7356                            ;*      THE FIRST 4 CYLINDERS ARE WRITTEN. THIS IS DONE TO
7357                            ;*      INSURE THE FORMAT IS CORRECT.
7358                            ;*
7359                            ;*      ZERO OUT THE FIRST 256 SECTORS OF THE DISK WITH
7360                            ;*      ONE SECTOR WRITES.  ISSUE A SEEK TO CYLINDER 0, TRACK 0.
7361                            ;*      ISSUE A WRITE DATA OF MAXIMUM DATA TRANSFER 200000 WORDS
7362                            ;*      TO CYLINDER 0, TRACK 0, SECTOR 0.  MAKE SURE CONTROLLER
7363                            ;*      TIME OUT IS NOT SET.  CHECK CYLINDER ADDRESS,
7364                            ;*      DISK ADDRESS, BUS ADDRESS AND WORD COUNT.  READ
7365                            ;*      EACH SECTOR TO MAKE SURE IT WAS WRITTEN CORRECTLY.
7366                            ;*
7367                            ;*      NOTE:   THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT
7368                            ;*              IN THE FIRST 256 SECTORS ON THE PACK.
7369                            ;*
7370                            ;;*****************************************************************
7371  023620  000004            TST72:  SCOPE
7372  023622  012737  000002  001262    MOV     #2,$TIMES       ;;DO 2 ITERATIONS
7373  023630  032737  000400  001664    BIT     #FPFMT,OPTFLG   ;TEST IF FIRST PASS SWITCH SET
7374  023636  001043                    BNE     24$             ;YES - SKIP FORMAT
```

```
7375
7376  023640  105037  023663          CLRB    21$             ;CLEAR ADDRESS POINTERS
7377  023644  005037  023664          CLR     22$
7378
7379  023650  004437  035674  20$:    JSR     R4,LRLOAD       ;LOAD "L" REGISTERS
7380  023654  000127                  WRHEAD                  ;WRITE HEADER
7381  023656  177676                  -102                    ;102 WORDS
7382  023660  072414                  OBUFF                   ;OBUFF IS BUFF ADDRESS
7383  023662     000                  .BYTE   0               ;SECTOR 0
7384  023663     000          21$:    .BYTE   0               ;TRACK ADDRESS (VARIABLE)
7385  023664  000000          22$:    0                       ;CYLINDER 0 (VARIABLE)
7386
7387  023666  004437  042260          JSR     R4,GENCOM       ;GO GENERATE HEADERS
7388  023672  001200                  1200                    ;WITH BAD SECTOR ERRORS
7389
7390  023674  104417                  TLOADRK                 ;LOAD RK REGS
7391  023676  104431                  TWAT112                 ;WAIT FOR INTERRUPT
7392  023700  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7393
7394  023702  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7395  023704  104004                  ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7396
7397  023706  123727  023663  000002  CMPB    21$,#2          ;TEST IF LAST TRACK
7398  023714  001403                  BEQ     23$             ;YES - SKIP
7399  023716  105237  023663          INCB    21$             ;ELSE BUMP TRACK
7400  023722  000752                  BR      20$             ;LOOP
7401
7402  023724  105037  023663  23$:    CLRB    21$             ;CLEAR TRACK POINTER
7403  023730  023727  023664  000003  CMP     22$,#3          ;TEST IF LAST CYLINDER WRITTEN
7404  023736  001403                  BEQ     24$             ;YES - SKIP
7405  023740  005237  023664          INC     22$             ;ELSE BUMP CYLINDER
7406  023744  000741                  BR      20$             ;LOOP
7407
7408  023746  013737  023760  001110  24$:    MOV     1$,$LPERR       ;SET LOCAL LOOP ON ERROR
7409  023754  012703  000400          MOV     #400,R3         ;SET COUNT FOR SECTOR CLEARING
7410  023760                  1$:
7411  023760  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
7412  023762  104003                  ERROR   3               ;BAD INIT ERROR
7413  023764  004437  035674          JSR     R4,LRLOAD       ;LOAD "L" REGS
7414  023770  000123                  WRDATA                  ;WRDATA
7415  023772  177400                  -400                    ;-400 WORDS
7416  023774  072414                  OBUFF                   ;OBUFF IS BUFF ADDRESS
7417  023776     000                  .BYTE   0               ;SECTOR 0
7418  023777     000                  .BYTE   0               ;TRACK 0
7419  024000  000000                  0                       ;CYLINDER 0
7420
7421  024002  104417                  TLOADRK                 ;LOAD RK REGS
7422  024004  104434                  TWAT159                 ;WAIT FOR INTERRUPT
7423  024006  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7424
7425  024010  104421                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7426  024012  104004                  ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7427
7428  024014  104415                  SCOP1                   ;LOCAL LOOP ON ERROR TO 1$
7429  024016  005303                  DEC     R3              ;DECREMENT COUNT
7430  024020  012762  072414  000004  2$:      MOV     #OBUFF,RKBA(R2) ;SET BA
```

```
7431  024026  012762  177400  000002          MOV     #-400,RKWC(R2)  ;AND WC AGAIN
7432  024034  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
7433  024040  013762  001626  000010          MOV     DRVNUM,RKCS2(R2)        ;SET DRIVE NUMBER
7434  024046  012737  000123  036124          MOV     #WRDATA,HOLD
7435  024054  053737  001720  036124          BIS     DTYPE,HOLD
7436  024062  013762  036124  000000          MOV     HOLD,RKCS1(R2) ;DO WRITE DATA
7437
7438  024070  104425                          TWAT48                  ;WAIT FOR INTERRUPT
7439  024072  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR     .
7440
7441  024074  032762  000200  000014          BIT     #BSE,RKER(R2)   ;BAD SECTOR ERROR?
7442  024102  001415                          BEQ     3$              ;NO-SKIP
7443  024104  032737  000200  001664          BIT     #BSERPT,OPTFLG  ;TEST IF BSE TO MANY HAS BEEN REPORTED
7444  024112  001007                          BNE     5$              ;YES - SKIP
7445  024114  052737  000200  001664          BIS     #BSERPT,OPTFLG  ;SET FLAG
7446  024122  012737  052457  001360          MOV     #OPR017,EMIN    ;SET MESSAGE
7447  024130  104001                          ERROR   1               ;"FIRST 256 SECTOR NOT BSE FREE"
7448  024132  000137  024466          5$:     JMP     14$             ;GO TO EXIT
7449                                  ;       CHECK FOR ANY OTHER ERRORS
7450  024136                          3$:
7451  024136  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7452  024140  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7453  024142  104415                          SCOP1                   ;LOCAL LOOP TO 1$ (RESTART SECTOR CLEAR)
7454
7455  024144  005303                          DEC     R3              ;DECREMENT COUNT
7456  024146  001324                          BNE     2$              ;LOOP IF NOT ZERO
7457
7458  024150  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7459  024154  000117                          SEEK                    ;SEEK
7460  024156  000000                          0                       ;0 WORDS
7461  024160  000000                          0                       ;0 IS BUFF ADDRESS
7462  024162     000                          .BYTE   0               ;SECTOR 0
7463  024163     000                          .BYTE   0               ;TRACK 0
7464  024164  000000                          0                       ;CYLINDER 0
7465
7466  024166  104417                          TLOADRK                 ;LOAD RK REGS
7467  024170  104423                          TWAT16                  ;WAIT FOR INTERRUPT
7468  024172  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7469  024174  005037  001662                  CLR     INTSET          ;CLEAR FIRST INTERRUPT
7470  024200  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7471  024202  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7472  024204  104427                          TWAT80                  ;WAIT FOR SECOND INTERRUPT
7473  024206  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7474  024210  004437  042260                  JSR     R4,GENCOM       ;GENERATE DATA
7475  024214  004006                          4006                    ;PATTERN 6, 1ST WORD REPEATED
7476  024216  000400                          400                     ;400 WORDS
7477
7478  024220                          4$:
7479  024220  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7480  024222  104003                          ERROR   3               ;BAD INIT ERROR
7481  024224  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7482  024230  000123                          WRDATA                  ;WRDATA
7483  024232  000000                          0                       ;0 WORDS
7484  024234  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7485  024236     000                          .BYTE   0               ;SECTOR 0
7486  024237     000                          .BYTE   0               ;TRACK 0
```

```
7487  024240  000000                        0                       ;CYLINDER 0
7488
7489  024242  052737  000020  001610        BIS     #BAI,L.CS2
7490
7491  024250  104417                        TLOADRK                 ;LOAD RK REGS
7492  024252  104437                        TWAT8S                  ;WAIT FOR SECOND INTERRUPT
7493  024254  104002                        ERROR   2               ;ELSE REPORT TO SLOW/NOT COMPLETE ERROR
7494
7495  024256                        7$:
7496  024256  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7497  024260  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7498
7499  024262  104415                        SCOP1                   ;INTERNAL LOOP ON ERROR TO 4$
7500  024264  012703  000400                MOV     #400,R3         ;SET COUNTER FOR READ-COMPARE LOOP
7501  024270  005037  024332                CLR     10$             ;CLEAR SECTOR AND TRACK VALUES
7502  024274  005037  024334                CLR     12$             ;CLEAR CYL VALUE
7503  024300  013737  024314  001110        MOV     8$,$LPERR       ;SET LOCAL LOOP ON ERROR
7504  024306  042737  000020  001610        BIC     #BAI,L.CS2      ;CLEAR BAI
7505
7506  024314                        8$:
7507  024314  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
7508  024316  104003                        ERROR   3               ;BAD INIT ERROR
7509  024320  004437  035674        9$:     JSR     R4,LRLOAD       ;LOAD RK REGS
7510  024324  000121                        RDDATA                  ;READ DATA
7511  024326  177400                        -400                    ;400 WORDS
7512  024330  070414                        IBUFF                   ;INTO IBUFF
7513  024332     000                10$:     .BYTE   0               ;SECTOR (VARIABLE)
7514  024333     000                11$:     .BYTE   0               ;TRACK  (VARIABLE)
7515  024334  000000                12$:     0                       ;CYL    (VARIABLE)
7516
7517  024336  104417                        TLOADRK                 ;LOAD RK REGS
7518  024340  104425                        TWAT48                  ;WAIT FOR INTERRUPT
7519  024342  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7520
7521  024344  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7522  024346  104004                        ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7523
7524  024350  104415                        SCOP1                   ;LOCAL LOOP ON ERROR TO 8$
7525
7526  024352  004437  042260                JSR     R4,GENCOM       ;COMPARE DATA
7527  024356  100000                        100000
7528  024360  000400                        400                     ;400 WORDS
7529  024362  000413                        BR      13$             ;NO MISCUMPARE-EXIT LOOP
7530  024364  104015                        ERROR   15              ;REPORT FIRST ERROR
7531
7532  024366  013700  001634                MOV     ERRLMT,R0       ;GET ERROR LIMIT
7533  024372  005300                64$:    DEC     R0              ;DECREMENT COUNT
7534  024374  001406                        BEQ     65$             ;IF ZERO - EXIT
7535  024376  004437  042260                JSR     R4,GENCOM       ;CONTINUE DATA COMPARE
7536  024402  040000                        40000
7537  024404  000402                        BR      65$             ;NO MORE ERRORS - EXIT
7538  024406  104016                        ERROR   16              ;REPORT NEXT ERROR
7539  024410  000770                        BR      64$             ;LOOP
7540  024412                        65$:
7541
7542  024412  104415                13$:    SCOP1                   ;LOCAL LOOP TO 8$
```

```
7543
7544  024414  005303                          DEC     R3              ;DEC READ LOOP COUNT
7545  024416  001423                          BEQ     14$             ;IF ZERO-EXIT
7546
7547  024420  105237  024332                  INCB    10$             ;BUMP SECTOR
7548  024424  123727  024332  000026          CMPB    10$,#26         ;FINISHED WITH TRACK?
7549  024432  001332                          BNE     9$              ;NO-LOOP
7550  024434  105037  024332                  CLRB    10$             ;CLEAR SECTOR
7551  024440  105237  024333                  INCB    11$             ;BUMP TRACK
7552  024444  123727  024333  000003          CMPB    11$,#3          ;FINISHED WITH CYLINDER?
7553  024452  001322                          BNE     9$              ;NO-LOOP
7554  024454  105037  024333                  CLRB    11$             ;CLEAR TRACK
7555  024460  005237  024334                  INC     12$             ;BUMP CYL.
7556  024464  000715                          BR      9$              ;LOOP
7557
7558  024466                          14$:
7559                                  ;;*****************************************************************
7560                                  ;*TEST 73       MAXIMUM DATA TRANSFER (PART 2)
7561                                  ;*
7562                                  ;*      ZERO OUT FIRST 256 SECTORS OF THE DISK WITH
7563                                  ;*      200000 WORD WRITE.  SEEK TO CYLINDER 632.
7564                                  ;*      ISSUE A WRITE OF MAXIMUM DATA TRANSFER
7565                                  ;*      200000 WORD WRITE.  MAKE SURE CONTROLLER TIME
7566                                  ;*      OUT IS NOT SET.  CHECK CYLINDER ADDRESS
7567                                  ;*      DISK ADDRESS, BUS ADDRESS AND WORD COUNT.
7568                                  ;*      SEEK TO CYLINDER 632.  ISSUE A WRITE CHECK
7569                                  ;*      OF 200000 WORDS.  MAKE SURE NO ERROR SETS.
7570                                  ;*
7571                                  ;*      NOTE:   THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT
7572                                  ;*              IN THE FIRST 256 SECTORS ON THE PACK.
7573                                  ;*              CYLINDER 1456 IS USED FOR RK07.
7574                                  ;*
7575                                  ;*****************************************************************
7576  024466  000004                  TST73:  SCOPE
7577  024470  012737  000002  001262          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
7578  024476  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7579  024500  104003                          ERROR   3               ;BAD INIT ERROR
7580  024502  012700  000620                  MOV     #400.,R0        ;SET COUNT FOR INTERRUPT WAIT
7581  024506  005037  072414                  CLR     OBUFF
7582
7583  024512  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7584  024516  000123                          WRDATA                  ;WRDATA
7585  024520  000000                          0                       ;0 WORDS
7586  024522  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
7587  024524     000                          .BYTE   0               ;SECTOR 0
7588  024525     000                          .BYTE   0               ;TRACK 0
7589  024526  000000                          0                       ;CYLINDER 0
7590  024530  052737  000020  001610          BIS     #BAI,L.CS2      ;SET BAI
7591  024536  104417                          TLOADRK                 ;LOAD RK REGS
7592  024540  104434                  1$:     TWAT159                 ;WAIT FOR INTERRUPT
7593  024542  000401                          BR      2$              ;NO INTERRUPT-SKIP
7594  024544  000403                          BR      3$              ;INTERRUPT-SKIP
7595
7596  024546  005300                  2$:     DEC     R0              ;DEC WAIT COUNTER
7597  024550  001373                          BNE     1$              ;NO ZERO-LOOP
7598  024552  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
```

```
 7599
 7600  024554  032762  000200  000014  3$:     BIT     #BSE,RKER(R2)   ;DID BSE SET
 7601  024562  001415                          BEQ     4$              ;NO-SKIP
 7602
 7603  024564  032737  000200  001664          BIT     #BSERPT,OPTFLG  ;TEST IF TO MANY BAD SECTORS REPORTED
 7604  024572  001007                          BNE     12$             ;YES - SKIP
 7605  024574  052737  000200  001664          BIS     #BSERPT,OPTFLG  ;SET FLAG
 7606  024602  012737  052457  001360          MOV     #OPR017,EM1N    ;SET MESSAGE
 7607  024610  104001                          ERROR   1               ;"FIRST 256 SECTORS NOT BSE FREE"
 7608  024612  000137  025160          12$:    JMP     11$             ;EXIT
 7609
 7610  024616                          4$:
 7611  024616  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 7612  024620  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
 7613  024622  013737  036120  024652          MOV     LSTCYL,13$
 7614  024630  013737  036120  025026          MOV     LSTCYL,14$
 7615
 7616  024636  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
 7617  024642  000117                          SEEK
 7618  024644  000000                          0                       ;0 WORDS
 7619  024646  000000                          0                       ;0 IS BUFF ADDR
 7620  024650     000                          .BYTE   0               ;SEC 0
 7621  024651     000                          .BYTE   0               ;TRK 0
 7622  024652  000000          13$:    0                               ;CYL 632/1456
 7623
 7624  024654  104417                          TLOADRK                 ;LOAD RK REGS
 7625  024656  104423                          TWAT16                  ;WAIT FOR INTERRUPT
 7626  024660  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 7627  024662  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
 7628
 7629  024666  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 7630  024670  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 7631
 7632  024672  104427                          TWAT80                  ;WAIT FOR 2ND INTERRUPT
 7633  024674  104002                          ERROR   2
 7634
 7635  024676  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 7636  024700  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 7637
 7638  024702  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
 7639  024706  000105                          CLEAR                   ;CLEAR
 7640  024710  000000                          0                       ;0 WORDS
 7641  024712  000000                          0                       ;0 IS BUFF ADDRESS
 7642  024714     000                          .BYTE   0               ;SECTOR 0
 7643  024715     000                          .BYTE   0               ;TRACK 0
 7644  024716  000000                          0                       ;CYLINDER 0
 7645
 7646  024720  104417                          TLOADRK                 ;LOAD RK REGS
 7647  024722  104423                          TWAT16                  ;WAIT FOR INTERRUPT
 7648  024724  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 7649
 7650  024726  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 7651  024730  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 7652
 7653  024732  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
 7654  024736  000123                          WRDATA                  ;WRDATA
```

```
 7655  024740  000000                          0                       ;0 WORDS
 7656  024742  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
 7657  024744     000                          .BYTE   0               ;SECTOR 0
 7658  024745     000                          .BYTE   0               ;TRACK 0
 7659  024746  000000                          0                       ;CYLINDER 0
 7660
 7661  024750  012737  135143  072414          MOV     #135143,OBUFF   ;SET WORD FOR OUTPUT
 7662  024756  012700  000621                  MOV     #401.,R0        ;SET COUNT FOR INTERRUPT WAIT
 7663  024762  052737  000020  001610          BIS     #BAI,L.CS2      ;SET BUS ADDRESS INC INHIBIT
 7664
 7665  024770  104417                          TLOADRK                 ;LOAD RK REGS
 7666  024772  104434                  5$:     TWAT159                 ;WAIT FOR INTERRUPT
 7667  024774  000401                          BR      6$              ;NO INTERRUPT-BRANCH
 7668  024776  000403                          BR      7$              ;INTERRUPT-BRANCH
 7669
 7670  025000  005300                  6$:     DEC     R0              ;DEC WAIT COUNT
 7671  025002  001373                          BNE     5$              ;LOOP IF NOT ZERO
 7672  025004  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 7673
 7674  025006                          7$:
 7675  025006  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 7676  025010  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
 7677
 7678  025012  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
 7679  025016  000117                          SEEK
 7680  025020  000000                          0                       ;0 WORDS
 7681  025022  000000                          0                       ;0 IS BUFF ADDR
 7682  025024     000                          .BYTE   0               ;SEC 0
 7683  025025     000                          .BYTE   0               ;TRK 0
 7684  025026  000000                  14$:    0                       ;CYL 632/1456
 7685  025030  104417                          TLOADRK                 ;LOAD RK REGS
 7686  025032  104423                          TWAT16                  ;WAIT FOR INTERRUPT
 7687  025034  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 7688  025036  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
 7689
 7690  025042  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 7691  025044  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 7692
 7693  025046  104427                          TWAT80                  ;WAIT FOR SECOND INIT
 7694  025050  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 7695  025052  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 7696  025054  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 7697
 7698  025056  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
 7699  025062  000105                          CLEAR                   ;CLEAR
 7700  025064  000000                          0                       ;0 WORDS
 7701  025066  000000                          0                       ;0 IS BUFF ADDRESS
 7702  025070     000                          .BYTE   0               ;SECTOR 0
 7703  025071     000                          .BYTE   0               ;TRACK 0
 7704  025072  000000                          0                       ;CYLINDER 0
 7705
 7706  025074  104417                          TLOADRK                 ;LOAD RK REGS
 7707  025076  104423                          TWAT16                  ;WAIT FOR INTERRUPT
 7708  025100  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 7709
 7710  025102  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
```

```
 7711  025104  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 7712
 7713  025106  004437  035674                   JSR     R4,LRLOAD       ;LOAD "L" REGS
 7714  025112  000131                          WRTCHK                  ;WRTCHK
 7715  025114  000000                          0                       ;0 WORDS
 7716  025116  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
 7717  025120     000                          .BYTE   0               ;SECTOR 0
 7718  025121     000                          .BYTE   0               ;TRACK 0
 7719  025122  000000                          0                       ;CYLINDER 0
 7720  025124  052737  000020  001610          BIS     #BAI,L.CS2      ;SET BAI FLAG
 7721  025132  012700  000621                   MOV     #401.,R0        ;SET WAIT COUNT
 7722
 7723  025136  104417                          TLOADRK                 ;LOAD RK REGS
 7724  025140  104434                  8$:     TWAT159                 ;WAIT FOR INTERRUPT
 7725  025142  00040?                          BR      9$              ;NO INTERRUPT-SKIP
 7726  025144  000403                          BR      10$             ;INTERRUPT-SKIP
 7727
 7728  025146  005300                  9$:     DEC     R0              ;DEC WAIT COUNT
 7729  025150  001373                          BNE     8$              ;NOT ZERO-LOOP
 7730  025152  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 7731
 7732  025154                          10$:
 7733  025154  10442?                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 7734  025156  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
 7735
 7736  025160                          11$:
 7737                                  ;;*****************************************************************
 7738                                  ;*TEST 74       CONTROLLER TIME OUT
 7739                                  ;*
 7740                                  ;*      SEEK TO CYLINDER 632.  ISSUE A RECALIBRATE AND DO NOT
 7741                                  ;*      WAIT FOR SECOND INTERRUPT. NOW ISSUE A READ HEADER
 7742                                  ;*      OF CYLINDER 0, TRACK 0.  MAKE SURE CONTROLLER TIME
 7743                                  ;*      OUT SETS.
 7744                                  ;*      CYLINDER 1456 IS USED FOR THE RK07.
 7745                                  ;*
 7746                                  ;;*****************************************************************
 7747  025160  000004                  TST74:  SCOPE
 7748  025162  012737  000005  001262          MOV     #5.,$TIMES      ;;DO 5. ITERATIONS
 7749  025170  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
 7750  025172  104003                          ERROR   3               ;BAD INIT ERROR
 7751  025174  013737  036120  025216          MOV     LSTCYL,1$
 7752
 7753  025202  004437  035674                   JSR     R4,LRLOAD       ;LOAD "L" REGS
 7754  025206  000117                          SEEK
 7755  025210  000000                          0                       ;0 WORDS
 7756  025212  000000                          0                       ;0 IS BUFF ADDR
 7757  025214     000                          .BYTE   0               ;SEC 0
 7758  025215     000                          .BYTE   0               ;TRK 0
 7759  025216  000000                  1$:     0                       ;CYL 632/1456
 7760
 7761  025220  104417                          TLOADRK                 ;LOAD RK REGS
 7762  025222  104423                          TWAT16                  ;WAIT FOR INTERRUPT
 7763  025224  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 7764
 7765  025226  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 7766  025230  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
```

```
                                        ;CLEAR INTERRUPT FLAG
                                        ;WAIT FOR SECOND INTERRUPT
                                        ;TO SLOW/NOT COMPLETE ERROR
                                        ;CHECK OPERATION FOR ANY ERRORS
                                        ;REPORT ALL ERRORS

                                        ;LOAD '_" REGS
                                        ;CLEAR
                                        ;0 WORDS
                                        ;0 IS BUFF ADDRESS
                                        ;SECTOR 0
                                        ;TRACK 0
                                        ;CYLINDER 0

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

                                        ;CHECK OPERATION FOR ANY ERRORS
                                        ;REPORT ALL ERRORS

                                        ;LOAD "L" REGS
                                        ;RECAL
                                        ;0 WORDS
                                        ;0 IS BUFF ADDRESS
                                        ;SECTOR 0
                                        ;TRACK 0
                                        ;CYLINDER 0

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

                                        ;LOAD "L" REGS
                                        ;RDHEAD
                                        ;0 WORDS
                                        ;0 IS BUFF ADDRESS
                                        ;SECTOR 0
                                        ;TRACK 0
                                        ;CYLINDER 0

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

                                        ;CHECK OPERATION WITH EXPECTED ERROR

                                        ;CONTROLLER TIME OUT
                                        ;REPORT ANY DISCREPANCIES
                                        ;CLEAR SUBSYSTEM
                                        ;BAD INIT ERROR
                                        ;CLEAR INT FLAG
                                        ;SET INT ENABLE
                                        ;WAIT FOR SECOND INT
```

```
                                .SBTTL  **ERRORS DURING DATA TRANSFER
                                ;;*****************************************************************
                                ;*TEST 75        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
                                ;;*     LIMIT SHOULD BE THE ONLY ERROR SET.
                                ;;*****************************************************************
025402  000004                  TST75:  SCOPE
025404  012737  000003  001262          MOV     #3.,$TIMES      ;;DO 3. ITERATIONS

025412  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
025414  104003                          ERROR   3               ;BAD INIT ERROR

025416  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
025422  000113                          RECAL                   ;RECAL
025424  000000                          0                       ;0 WORDS
025426  000000                          0                       ;0 IS BUFF ADDRESS
025430     000                          .BYTE   0               ;SECTOR 0
025431     000                          .BYTE   0               ;TRACK 0
025432  000000                          0                       ;CYLINDER 0

025434  104417                          TLOADRK                 ;LOAD RK REGS
025436  104423                          TWAT16                  ;WAIT FOR INTERRUPT
025440  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

025442  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
025446  104437                          TWAT8S                  ;WAIT FOR SECOND INTERRUPT
025450  104002                          ERROR   2

025452  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
025454  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS

025456  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
025462  000117                          SEEK                    ;SEEK
025464  000000                          0                       ;0 WORDS
025466  000000                          0                       ;0 IS BUFF ADDRESS
025470     000                          .BYTE   0               ;SECTOR 0
025471     000                          .BYTE   0               ;TRACK 0
025472  000002                          2                       ;CYLINDER 2
025474  012737  000020  001616          MOV     #PAT,L.MR1      ;SET EVEN PARITY BIT
025502  104417                          TLOADRK                 ;LOAD RK REGS
025504  104423                          TWAT16                  ;WAIT FOR INTERRUPT
025506  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

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

```
7879  025514  004437  035674            JSR     R4,LRLOAD       ;LOAD "L" REGS
7880  025520  00012?                    WRDATA                  ;WRDATA
7881  025522  177400                    -400                    ;-400 WORDS
7882  025524  072414                    OBUFF                   ;OBUFF IS BUFF ADDRESS
7883  025526     000                    .BYTE   0               ;SECTOR 0
7884  025527     000                    .BYTE   0               ;TRACK 0
7885  025530  000001                    1                       ;CYLINDER 1
7886
7887  025532  104417                    TLOADRK                 ;LOAD RK REGS
7888  025534  104423                    TWAT16                  ;WAIT FOR INTERRUPT
7889  025536  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7890
7891  025540  104422                    TCHKWE                  ;CHECK OPERATION WITH ERROR
7892  025542  000002                    SKIERR                  ;SEEK INCOMPLETE
7893  025544  000000                    0
7894  025546  000000                    0
7895  025550  104004                    ERROR   4  ;OR 5,6,OR7  ;REPORT ALL DISCREPANCIES
7896
7897  025552  104416                    TSSINIT                 ;CLEAR SUBSYSTEM
7898  025554  104003                    ERROR   3               ;BAD INIT ERROR
7899
7900
7901  025556                    3$:
7902  025556  004437  035674            JSR     R4,LRLOAD       ;LOAD "L" REGS
7903  025562  000101                    SELDRV                  ;SELDRV
7904  025564  000000                    0                       ;0 WORDS
7905  025566  000000                    0                       ;0 IS BUFF ADDRESS
7906  025570     000                    .BYTE   0               ;SECTOR 0
7907  025571     000                    .BYTE   0               ;TRACK 0
7908  025572  000000                    0                       ;CYLINDER 0
7909
7910  025574  012737  000001  001616    MOV     #1,L.MR1        ;SET TO GET STATUS PAIR 1
7911  025602  104417                    TLOADRK                 ;LOAD RK REGS
7912  025604  104423                    TWAT16                  ;WAIT FOR INTERRUPT
7913  025606  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7914
7915  025610  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
7916  025612  104004                    ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
7917
7918  025614  032737  000040  001574    BIT     #S.HDHM,T.MR2   ;TEST IF HEADS HOME
7919  025622  001755                    BEQ     3$              ;NO - GET STATUS AGAIN
7920
7921  025624  104416                    TSSINIT                 ;CLEAR SUBSYSTEM
7922  025626  104003                    ERROR   3               ;BAD INIT ERROR
7923
7924  025630  012762  000100  000000    MOV     #IE,RKCS1(R2)   ;SET IE
7925
7926  025636  005037  001662            CLR     INTSET          ;CLEAR INT FLAG
7927  025642  104437                    TWAT8S                  ;WAIT FOR SECOND INTERRUPT
7928  025644  000401                    BR      1$
7929  025646  000404                    BR      2$
7930
7931  025650  012737  060314  001372  1$:  MOV   #DH016,DH2N    ;"SUBSYSTEM CLEAR TO RESET LIMIT ERROR
7932                                            ;ALLOWING HEADS TO RELOAD"
7933  025656  104002                    ERROR   2
7934
```

```
7935  025660                          2$:
7936
7937                                  ;;*****************************************************************
7938                                  ;*TEST 76       PROGRAMMING ERROR
7939                                  ;*      ISSUE A SUBSYSTEM CLEAR. ISSUE
7940                                  ;*      A READ DATA OF 400 WORDS ON CYLINDER 0,
7941                                  ;*      TRACK 0, SECTOR 0. DURING READ ISSUE A
7942                                  ;*      WRITE TO THE SPARE REGISTER. MAKE SURE
7943                                  ;*      PROGRAMMING ERROR SETS.
7944                                  ;;*****************************************************************
7945  025660  000004                  TST76:  SCOPE
7946  025662  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
7947  025670  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7948  025672  104003                          ERROR   3               ;BAD INIT ERROR
7949
7950  025674  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
7951  025700  000121                          RDDATA                  ;RDDATA
7952  025702  177400                          -400                    ;-400 WORDS
7953  025704  070414                          IBUFF                   ;IBUFF IS BUFF ADDRESS
7954  025706     000                          .BYTE   0               ;SECTOR 0
7955  025707     000                          .BYTE   0               ;TRACK 0
7956  025710  000000                          0                       ;CYLINDER 0
7957
7958  025712  104417                          TLOADRK         ;LOAD RK REGS
7959
7960  025714  012762  000001  000022          MOV     #1,RKSPAR(R2)   ;WRITE SPARE REGISTER
7961
7962  025722  104423                          TWAT16          ;WAIT FOR INTERRUPT
7963  025724  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
7964
7965  025726  104422                          TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
7966  025730  000000                          0
7967  025732  000000                          0
7968  025734  000020                          PGERR                   ;PROG ERROR
7969  025736  104004                          ERROR   4  ;OR 5,6,7    ;REPORT ALL DISCREPANCIES
7970
7971
7972                                  ;;*****************************************************************
7973                                  ;*TEST 77       ECC HARD
7974                                  ;*      ISSUE A SUBSYSTEM CLEAR. ISSUE
7975                                  ;*      A WRITE DATA WORDS CONSISTING OF 177777 TO
7976                                  ;*      CYLINDER 0, TRACK 0, SECTOR 0. NOW WRITE
7977                                  ;*      ALL ZEROS TO CYLINDER 0, TRACK 0, SECTOR 0.
7978                                  ;*      DURING WRITE ISSUE CONTROLLER CLEAR. MAKE
7979                                  ;*      SURE PROGRAMMING ERROR IS RESET. NOW
7980                                  ;*      ISSUE A READ DATA TO CYLINDER 0, TRACK 0,
7981                                  ;*      HEAD 0 AND AN ECC HARD ERROR SHOULD SET.
7982                                  ;;*****************************************************************
7983  025740  000004                  TST77:  SCOPE
7984  025742  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
7985  025750  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
7986  025752  104003                          ERROR   3               ;BAD INIT ERROR
7987
7988  025754  004437  042260                  JSR     R4,GENCOM       ;GENERATE DATA OF ALL ONES
7989  025760  000001                          1
7990  025762  000400                          400
```

```
7991
7992  025764  004437  035674              JSR     R4,LRLOAD       ;LOAD "L" REGS
7993  025770  000123                      WRDATA                  ;WRDATA
7994  025772  177400                      -400                    ;-400 WORDS
7995  025774  072414                      OBUFF                   ;OBUFF IS BUFF ADDRESS
7996  025776     000                      .BYTE   0               ;SECTOR 0
7997  025777     000                      .BYTE   0               ;TRACK 0
7998  026000  000000                      0                       ;CYLINDER 0
7999
8000  026002  104417                      TLOADRK                 ;LOAD RK REGS
8001  026004  104430                      TWAT96                  ;WAIT FOR INTERRUPT
8002  026006  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8003
8004  026010  104421                      TCHKOP                  ;CHECK OPERATION FOP ANY ERRORS
8005  026012  104004                      ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
8006
8007  026014  004437  042260              JSR     R4,GENCOM       ;GENERATE DATA OF ZEROS
8008  026020  000002                      2
8009  026022  000400                      400
8010
8011  026024  004437  035674              JSR     R4,LRLOAD       ;LOAD "L" REGS
8012  026030  000123                      WRDATA                  ;WRDATA
8013  026032  177630                      -150                    ;-150 WORDS
8014  026034  072414                      OBUFF                   ;OBUFF IS BUFF ADDRESS
8015  026036     000                      .BYTE   0               ;SECTOR 0
8016  026037     000                      .BYTE   0               ;TRACK 0
8017  026040  000000                      0                       ;CYLINDER 0
8018
8019  026042  104417                      TLOADRK                 ;START OPERATION
8020
8021  026044  005737  001662        1$:   TST     INTSET          ;CHECK IF INTERRUPT HAS OCCURRED
8022  026050  001026                      BNE     2$              ;YES - MUCH TO SOON. REPORT ERROR
8023  026052  005762  000002              TST     RKWC(R2)        ;TEST IF NPR'S DONE
8024  026056  001372                      BNE     1$              ;NO - LOOP
8025
8026  026060  052762  100000  000000      BIS     #CCLR,RKCS1(R2) ;CLEAR CONTROLLER (CROWBAR WRITE)
8027
8028  026066  004437  035674              JSR     R4,LRLOAD       ;LOAD "L" REGS
8029  026072  000121                      RDDATA                  ;RDDATA
8030  026074  177400                      -400                    ;-400 WORDS
8031  026076  070414                      IBUFF                   ;IBUFF IS BUFF ADDRESS
8032  026100     000                      .BYTE   0               ;SECTOR 0
8033  026101     000                      .BYTE   0               ;TRACK 0
8034  026102  000000                      0                       ;CYLINDER 0
8035
8036  026104  104417                      TLOADRK                 ;LOAD RK REGS
8037  026106  104425                      TWAT48                  ;WAIT FOR INTERRUPT
8038  026110  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8039
8040  026112  104422                      TCHKWE                  ;CHECK OPERATION WITH ERROR
8041  026114  000000                      0
8042  026116  000003                      DCKERR!ECHERR           ;DATA CHECK AND ECC HARD
8043  026120  000000                      0
8044  026122  104004                      ERROR   4 ;OR 5,6,7     ;REPORT ALL DISCREPANCIES
8045
8046  026124  000402                      BR      3$              ;SKIP TO EXIT
```

```
                                2$:
026126                                  TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
026130  104004                          ERROR   4  ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
026132                          3$:


                                ;;*****************************************************************
                                ;*TEST 100      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.
                                ;;*****************************************************************
026132  000004                  TST100: SCOPE
026134  012737  000005  001262          MOV     #5.,$TIMES      ;;DO 5. ITERATIONS

026142  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
026144  104003                          ERROR   3               ;BAD INIT ERROR

026146  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
026152  000117                          SEEK                    ;SEEK
026154  000000                          0                       ;0 WORDS
026156  000000                          0                       ;0 IS BUFF ADDRESS
026160     000                          .BYTE   0               ;SECTOR 0
026161     000                          .BYTE   0               ;TRACK 0
026162  000632                          632                     ;CYLINDER 632

026164  104417                          TLOADRK                 ;LOAD RK REGS
026166  104423                          TWAT16                  ;WAIT FOR INTERRUPT
026170  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
026172  005037  001662                  CLR     INTSET          ;CLEAR INT FLAG
026176  104430                          TWAT96                  ;WAIT FOR SECOND INTERRUPT
026200  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

026202  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
026206  000113                          RECAL                   ;RECAL
026210  000000                          0                       ;0 WORDS
026212  000000                          0                       ;0 IS BUFF ADDRESS
026214     000                          .BYTE   0               ;SECTOR 0
026215     000                          .BYTE   0               ;TRACK 0
026216  000000                          0                       ;CYLINDER 0

026220  104417                          TLOADRK                 ;LOAD RK REGS
026222  104423                          TWAT16                  ;WAIT FOR INTERRUPT
026224  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR

026226  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
026232  000125                          RDHEAD                  ;RDHEAD
026234  000000                          0                       ;0 WORDS
026236  000000                          0                       ;0 IS BUFF ADDRESS
026240     000                          .BYTE   0               ;SECTOR 0
026241     000                          .BYTE   0               ;TRACK 0
026242  000000                          0                       ;CYLINDER 0
```

```
8103  026244  012737  000040  001616          MOV     #DMD,L.MR1      ;SET DIAG MODE
8104  026252  104417                          TLOADRK                 ;LOAD RK REGS
8105
8106  026254  004437  036316                  JSR     R4,MCLOCK       ;CLOCK CONTROLLER THROUGH SEEK
8107  026260  001062                          1062                    ;AND CLEAR TO READ
8108
8109  026262  005062  000026                  CLR     RKMR1(R2)       ;RESET DIAG MODE, LET RD HDRS COMPLETE
8110
8111  026266  104424                          TWAT32                  ;WAIT FOR INTERRUPT
8112  026270  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8113  026272  104422                          TCHKWE                  ;CHECK OPERATION WITH EXP ERROR
8114  026274  010000                          DTERR                   ;DRIVE TIMING ERROR
8115  026276  000000                          0
8116  026300  000000                          0
8117  026302  104004                          ERROR   4  ;OR 5,6,7    ;REPORT ALL DISCREPANCIES
8118
8119  026304                          1$:
8120  026304  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
8121  026306  104003                          ERROR   3               ;BAD INIT ERROR
8122  026310  012762  000100  000000          MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
8123  026316  005037  001662                  CLR     INTSET          ;CLEAR INT FLAG
8124
8125  026322  104437                          TWAT8S                  ;WAIT FOR INTERRUPT FOR END OF RECAL
8126  026324  104002                          ERROR   2
8127
8128  026326  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
8129  026332  000105                          CLEAR                   ;CLEAR
8130  026334  000000                          0                       ;0 WORDS
8131  026336  000000                          0                       ;0 IS BUFF ADDRESS
8132  026340     000                          .BYTE   0               ;SECTOR 0
8133  026341     000                          .BYTE   0               ;TRACK 0
8134  026342  000000                          0                       ;CYLINDER 0
8135
8136  026344  104417                          TLOADRK                 ;LOAD RK REGS
8137  026346  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8138  026350  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8139  026352  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8140  026354  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8141
8142                          .SBTTL  **ERROR FORCING IN DRIVE
8143
8144                          ;;*****************************************************************
8145                          ;*TEST 101      INITIALIZE CLEARING SACK
8146                          ;*
8147                          ;*      ISSUE A SUBSYSTEM CLEAR.  SELECT AN AVAILABLE
8148                          ;*      DRIVE.  ISSUE A SUBSYSTEM CLEAR.  PUT CONTROLLER IN
8149                          ;*      DIAGNOSTIC MODE.  ISSUE A SELECT COMMAND WITH
8150                          ;*      MESSAGE ID = 3 AND DRIVE SELECTED = 0.  CLOCK THROUGH
8151                          ;*      PHASE ADDRESS 6.  TURN OFF DIAGNOSTIC MODE.  MAKE
8152                          ;*      SURE UNIT FIELD ERROR DOES NOT SET.
8153                          ;*
8154                          ;;*****************************************************************
8155  026356  000004          TST101: SCOPE
8156  026360  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
8157  026366  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
8158  026370  104003                          ERROR   3               ;BAD INIT ERROR
```

```
8159
8160  026372  004437  035674            JSR     R4,LRLOAD       ;LOAD "L" REGS
8161  026376  000101                    SELDRV                  ;SELDRV
8162  026400  000000                    0                       ;0 WORDS
8163  026402  000000                    0                       ;0 IS BUFF ADDRESS
8164  026404     000                    .BYTE   0               ;SECTOR 0
8165  026405     000                    .BYTE   0               ;TRACK 0
8166  026406  000000                    0                       ;CYLINDER 0
8167
8168  026410  104417                    TLOADRK                 ;LOAD RK REGS
8169  026412  104423                    TWAT15                  ;WAIT FOR INTERRUPT
8170  026414  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8171
8172  026416  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8173  026420  104004                    ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8174
8175  026422  104416                    TSSINIT                 ;CLEAR SUBSYSTEM
8176  026424  104003                    ERROR   3               ;BAD INIT ERROR
8177
8178  026426  004437  035674            JSR     R4,LRLOAD       ;LOAD "L" REGS
8179  026432  000101                    SELDRV                  ;SELDRV
8180  026434  000000                    0                       ;0 WORDS
8181  026436  000000                    0                       ;0 IS BUFF ADDRESS
8182  026440     000                    .BYTE   0               ;SECTOR 0
8183  026441     000                    .BYTE   0               ;TRACK 0
8184  026442  000000                    0                       ;CYLINDER 0
8185  026444  012737  000043  001616    MOV     #3!DMD,L.MR1    ;SET DIAG MODE AND MESSAGE PAIR 3
8186  026452  005037  001610            CLR     L.CS2           ;SELECT DRIVE 0
8187
8188  026456  104417                    TLOADRK                 ;LOAD RK REGS
8189
8190  026460  004437  036316            JSR     R4,MCLOCK       ;CLOCK THROUGH PHASE ADDRESS 6
8191  026464  001027                    1027
8192
8193  026466  042762  000040  000026    BIC     #DMD,RKMR1(R2)  ;CLEAR MAINTENANCE MODE
8194
8195  026474  104424                    TWAT32                  ;WAIT FOR INTERRUPT
8196  026476  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE
8197
8198  026500  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8199  026502  104004                    ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8200                          ;;*****************************************************************
8201                          ;*TEST 102      DRIVE OFF TRACK
8202                          ;*
8203                          ;*      ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECALIBRATE.  ISSUE
8204                          ;*      OFFSET OF +1200 MICRO-INCHES.  PUT CONTROLLER IN DIAGNOSTIC
8205                          ;*      MODE.  ISSUE A WRITE DATA OF 1 WORD TO CYLINDER 0,
8206                          ;*      TRACK 0, SECTOR 0.  CLOCK THROUGH SEEK AND DRIVE CLEAR
8207                          ;*      MESSAGES.  TURN OFF DIAGNOSTIC MODE.  DRIVE OFF TRACK
8208                          ;*      SHOULD SET IN DRIVE.  REPEAT FOR ALL AVAILIABLE DRIVES.
8209                          ;*
8210                          ;;*****************************************************************
8211  026504  000004          TST102: SCOPE
8212  026506  012737  000005  001262    MOV     #5.,$TIMES      ;;DO 5. ITERATIONS
8213  026514  104416                    TSSINIT                 ;CLEAR SUBSYSTEM
8214  026516  104003                    ERROR   3               ;BAD INIT ERROR
```

```
8215
8216  026520  004437  035674              JSR     R4,LRLOAD       ;LOAD "L" REGS
8217  026524  000113                      RECAL                   ;RECAL
8218  026526  000000                      0                       ;0 WORDS
8219  026530  000000                      0                       ;0 IS BUFF ADDRESS
8220  026532     000                      .BYTE   0               ;SECTOR 0
8221  026533     000                      .BYTE   0               ;TRACK 0
8222  026534  000000                      0                       ;CYLINDER 0
8223
8224  026536  104417                      TLOADRK                 ;LOAD RK REGS
8225  026540  104423                      TWAT15                  ;WAIT FOR INTERRUPT
8226  026542  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8227  026544  005037  001662              CLR     INTSET          ;CLEAR INTERRUPT FLAG
8228
8229  026550  104437                      TWAT8S                  ;WAIT FOR INTERRUPT #2
8230  026552  104002                      ERROR   2
8231
8232  026554  004437  035674              JSR     R4,LRLOAD       ;LOAD "L" REGS
8233  026560  000105                      CLEAR                   ;CLEAR
8234  026562  000000                      0                       ;0 WORDS
8235  026564  000000                      0                       ;0 IS BUFF ADDRESS
8236  026566     000                      .BYTE   0               ;SECTOR 0
8237  026567     000                      .BYTE   0               ;TRACK 0
8238  026570  000000                      0                       ;CYLINDER 0
8239
8240  026572  104417                      TLOADRK                 ;LOAD RK REGS
8241  026574  104423                      TWAT16                  ;WAIT FOR INTERRUPT
8242  026576  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8243
8244  026600  104421                      TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8245  026602  104004                      ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8246
8247  026604  004437  035674              JSR     R4,LRLOAD       ;LOAD "L" REGS
8248  026610  000115                      OFFSET                  ;OFFSET
8249  026612  000000                      0                       ;0 WORDS
8250  026614  000000                      0                       ;0 IS BUFF ADDRESS
8251  026616     000                      .BYTE   0               ;SECTOR 0
8252  026617     000                      .BYTE   0               ;TRACK 0
8253  026620  000000                      0                       ;CYLINDER 0
8254  026622  112737  000060  001612      MOVB    #60,L.ASOF      ;SET OFFSET AT +1200
8255
8256  026630  104417                      TLOADRK                 ;LOAD RK REGS
8257  026632  104423                      TWAT16                  ;WAIT FOR INTERRUPT
8258  026634  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8259
8260  026636  104421                      TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8261  026640  104004                      ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8262
8263  026642  005037  001662              CLR     INTSET          ;CLEAR INT FLAG
8264
8265  026646  104424                      TWAT32                  ;WAIT FOR INT #2
8266  026650  104002                      ERROR   2
8267
8268  026652  104421                      TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8269  026654  104004                      ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8270
```

```
 8271  026656  004437  035674            JSR     R4,LRLOAD       ;LOAD "L" REGS
 8272  026662  000105                    CLEAR                   ;CLEAR
 8273  026664  000000                    0                       ;0 WORDS
 8274  026666  000000                    0                       ;0 IS BUFF ADDRESS
 8275  026670     000                    .BYTE   0               ;SECTOR 0
 8276  026671     000                    .BYTE   0               ;TRACK 0
 8277  026672  000000                    0                       ;CYLINDER 0
 8278
 8279  026674  104417                    TLOADRK                 ;LOAD RK REGS
 8280  026676  104423                    TWAT16                  ;WAIT FOR INTERRUPT
 8281  026700  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8282
 8283  026702  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8284  026704  104004                    ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8285
 8286  026706  004437  035674            JSR     R4,LRLOAD       ;LOAD "L" REGS
 8287  026712  000023                    23                      ;WRITE DATA W/O INTERRUPT ENAPLE
 8288  026714  177777                    -1                      ;-1 WORDS
 8289  026716  072414                    OBUFF                   ;OBUFF IS BUFF ADDRESS
 8290  026720     000                    .BYTE   0               ;SECTOR 0
 8291  026721     000                    .BYTE   0               ;TRACK 0
 8292  026722  000000                    0                       ;CYLINDER 0
 8293  026724  012737  000040  001616    MOV     #DMD,L.MR1      ;SET DIAGNOSTIC MODE
 8294
 8295  026732  104417                    TLOADRK
 8296
 8297  026734  004437  036316            JSR     R4,MCLOCK       ;CLOCK THROUGH SEEK & DRIVE CLEAR
 8298  026740  001064                    1064
 8299
 8300  026742  005062  000026            CLR     RKMR1(R2)       ;CLEAR DIAGNOSTIC MODE
 8301  026746  104426                    TWAT64                  ;WAIT FOR INTERRUPT
 8302  026750  012762  100000  000000    MOV     #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
 8303  026756  104423                    TWAT16                  ;STALL FOR 16 MS
 8304  026760  000240                    NOP
 8305  026762  000240                    NOP
 8306
 8307  026764  004437  035674            JSR     R4,LRLOAD       ;LOAD "L" REGS
 8308  026770  000101                    SELDRV                  ;SELDRV
 8309  026772  000000                    0                       ;0 WORDS
 8310  026774  000000                    0                       ;0 IS BUFF ADDRESS
 8311  026776     000                    .BYTE   0               ;SECTOR 0
 8312  026777     000                    .BYTE   0               ;TRACK 0
 8313  027000  000000                    0                       ;CYLINDER 0
 8314  027002  005037  001616            CLR     L.MR1           ;RESET DIAG MODE
 8315
 8316  027006  104417                    TLOADRK                 ;LOAD RK REGS
 8317  027010  104423                    TWAT16                  ;WAIT FOR INTERRUPT
 8318  027012  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8319
 8320  027014  104422                    TCHKWE                  ;CHECK OPERATION WITH ERROR EXPECTED
 8321  027016  000400                    DROTERR                 ;DRIVE OFF TRACK
 8322  027020  000000                    0
 8323  027022  000000                    0
 8324  027024  104004                    ERROR   4; OR 5,6,7     ;REPORT ANY DISCREPANCIES
 8325                          ;;*****************************************************************
 8326                          ;*TEST 103       FILE UNSAFE
```

```
8327                                    ;*
8328                                    ;*      ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECLAIBRATE.  ISSUE
8329                                    ;*      A READ HEAD OF CYLINDER 0, TRACK 0 IN 24 SECTOR
8330                                    ;*      FORMAT.  DO A SELECT COMMAND IN 26 SECTOR FORMAT.
8331                                    ;*      PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A WRITE
8332                                    ;*      HEADER TO CYLINDER 0, TRACK 0, ONE WORD IN 26 SECTOR
8333                                    ;       FORMAT.  CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES.
8334                                    ;*      SIMULATE INDEX PULSE.  TURN OFF DIAGNOSTIC MODE.  FILE
8335                                    ;*      UNSAFE SHOULD SET BECAUSE OF ATTEMPTING TO WRITE
8336                                    ;*      THROUGH SECTOR PULSE.  REPEAT FOR ALL AVAILIABLE DRIVES.
8337                                    ;*
8338                                    ;*****************************************************************
8339  027026  000004                    TST103: SCOPE
8340  027030  012737  000005  001262            MOV     #5.,$TIMES      ;;DO 5. ITERATIONS
8341  027036  012737  177777  001676            MOV     #-1,REFMT       ;SET REFORMAT SWITCH
8342  027044  104416                            TSSINIT                 ;CLEAR SUBSYSTEM
8343  027046  104003                            ERROR   3               ;BAD INIT ERROR
8344
8345  027050  004437  035674                    JSR     R4,LRLOAD       ;LOAD "L" REGS
8346  027054  000113                            RECAL                   ;RECAL
8347  027056  000000                            0                       ;0 WORDS
8348  027060  000000                            0                       ;0 IS BUFF ADDRESS
8349  027062     000                            .BYTE   0               ;SECTOR 0
8350  027063     000                            .BYTE   0               ;TRACK 0
8351  027064  000012                            12                      ;CYLINDER 12
8352
8353  027066  104417                            TLOADRK                 ;LOAD RK REGS
8354  027070  104423                            TWAT16                  ;WAIT FOR INTERRUPT
8355  027072  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8356
8357  027074  104421                            TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8358  027076  104004                            ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8359
8360  027100  005037  001662                    CLR     INTSET          ;CLEAR INT FLAG
8361  027104  104437                            TWAT8S                  ;WAIT FOR SECOND INT
8362  027106  104002                            ERROR   2
8363
8364  027110  104421                            TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8365  027112  104004                            ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8366
8367  027114  004437  035674                    JSR     R4,LRLOAD       ;LOAD "L" REGS
8368  027120  000105                            CLEAR                   ;CLEAR
8369  027122  000000                            0                       ;0 WORDS
8370  027124  000000                            0                       ;0 IS BUFF ADDRESS
8371  027126     000                            .BYTE   ^               ;SECTOR 0
8372  027127     000                            .BYTE   0               ;TRACK 0
8373  027130  000012                            12                      ;CYLINDER 12
8374
8375  027132  104417                            TLOADRK                 ;LOAD RK REGS
8376  027134  104423                            TWAT16                  ;WAIT FOR INTERRUPT
8377  027136  104002                            ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8378
8379  027140  104421                            TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8380  027142  104004                            ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8381
8382  027144  004437  035674                    JSR     R4,LRLOAD       ;LOAD "L" REGS
```

```
8383  027150  010125                  RDHEAD!CFMT                     ;RDHEAD!CFMT
8384  027152  000000                  0                               ;0 WORDS
8385  027154  000000                  0                               ;0 IS BUFF ADDRESS
8386  027156     000                  .BYTE   0                       ;SECTOR 0
8387  027157     000                  .BYTE   0                       ;TRACK 0
8388  027160  000012                  12                              ;CYLINDER 12
8389
8390  027162  104417                  TLOADRK                         ;LOAD RK REGS
8391  027164  104424                  TWAT32                          ;WAIT FOR INTERRUPT
8392  027166  104002                  ERROR   2                       ;TO SLOW/NOT COMPLETE ERROR
8393
8394  027170  104421                  TCHKOP                          ;CHECK OPERATION FOR ANY ERRORS
8395  027172  104004                  ERROR   4  ;OR 5, 6, 7          ;REPORT ALL ERRORS
8396
8397  027174  004437  035674          JSR     R4,LRLOAD               ;LOAD "L" REGS
8398  027200  000101                  SELDRV                          ;SELDRV
8399  027202  000000                  0                               ;0 WORDS
8400  027204  000000                  0                               ;0 IS BUFF ADDRESS
8401  027206     000                  .BYTE   0                       ;SECTOR 0
8402  027207     000                  .BYTE   0                       ;TRACK 0
8403  027210  000012                  12                              ;CYLINDER 12
8404
8405  027212  104417                  TLOADRK                         ;LOAD RK REGS
8406  027214  104423                  TWAT16                          ;WAIT FOR INTERRUPT
8407  027216  104002                  ERROR   2                       ;TO SLOW/NOT COMPLETE ERROR
8408
8409  027220  004437  035674          JSR     R4,LRLOAD               ;LOAD "L" REGS
8410  027224  000127                  WRHEAD                          ;WRHEAD
8411  027226  177777                  -1                              ;-1 WORDS
8412  027230  072414                  OBUFF                           ;OBUFF IS BUFF ADDRESS
8413  027232     000                  .BYTE   0                       ;SECTOR 0
8414  027233     000                  .BYTE   0                       ;TRACK 0
8415  027234  000012                  12                              ;CYLINDER 12
8416  027236  012737  000040  001616  MOV     #DMD,L.MR1              ;SET DIAGNOSTIC-MODE
8417
8418  027244  104417                  TLOADRK                         ;LOAD RK REGS
8419  027246  004437  036316          JSR     R4,MCLOCK               ;CLOCK THROUGH SEEK AND DRIVE CLEAR
8420  027252  001064                  1064
8421
8422  027254  052762  000200  000026  BIS     #MIND,RKMR1(R2)         ;SET INDEX
8423
8424  027262  004437  036316          JSR     R4,MCLOCK               ;CLOCK INDEX
8425  027266  001001                  1001
8426
8427  027270  042762  000200  000026  BIC     #MIND,RKMR1(R2)         ;CLEAR INDEX
8428
8429  027276  004437  036316          JSR     R4,MCLOCK               ;CLOCK CLEAR
8430  027302  001001                  1001
8431
8432  027304  005062  000026          CLR     RKMR1(R2)               ;CLEAR DIAGNOSTIC MODE
8433
8434  027310  104426                  TWAT64                          ;WAIT FOR INTERRUPT
8435  027312  104002                  ERROR   2                       ;TO SLOW/NOT COMPLETE ERROR
8436
8437  027314  104421                  TCHKOP                          ;CHECK OPERATION FOR ANY ERRORS
8438  027316  104004                  ERROR   4  ;OR 5, 6, 7, 10      ;REPORT ALL ERRORS
```

```
 8439
 8440  027320  004437  035674              JSR     R4,LRLOAD       ;LOAD "L" REGS
 8441  027324  000101                      SELDRV                  ;SELDRV
 8442  027326  000000                      0                       ;0 WORDS
 8443  027330  000000                      0                       ;0 IS BUFF ADDRESS
 8444  027332     000                      .BYTE   0               ;SECTOR 0
 8445  027333     000                      .BYTE   0               ;TRACK 0
 8446  027334  000012                      12                      ;CYLINDER 12
 8447
 8448  027336  005037  001616              CLR     L.MR1           ;CLEAR DIAG MODE
 8449
 8450  027342  104417                      TLOADRK                 ;LOAD RK REGS
 8451  027344  104423                      TWAT16                  ;WAIT FOR INTERRUPT
 8452  027346  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8453
 8454  027350  104422                      TCHKWE                  ;CHECK OPERATION WITH EXPECTED ERROR
 8455  027352  040400                      UNSERR!DROTERR          ;UNSAFE AND DRIVE OFF TRACK
 8456  027354  000000                      0
 8457  027356  000000                      0
 8458  027360  104004                      ERROR   4; OR 5,6,7     ;REPORT ANY DISCREPANCIES
 8459
 8460  027362  104416                      TSSINIT                 ;CLEAR SUBSYSTEM
 8461  027364  104003                      ERROR   3               ;BAD INIT ERROR
 8462
 8463  027366  004437  035674              JSR     R4,LRLOAD       ;LOAD "L" REGS
 8464  027372  000101                      SELDRV                  ;SELDRV
 8465  027374  000000                      0                       ;0 WORDS
 8466  027376  000000                      0                       ;0 IS BUFF ADDRESS
 8467  027400     000                      .BYTE   0               ;SECTOR 0
 8468  027401     000                      .BYTE   0               ;TRACK 0
 8469  027402  000012                      12                      ;CYLINDER 12
 8470
 8471  027404  012737  000001  001616      MOV     #1,L.MR1        ;SET MESSAGE SELECT ONE
 8472
 8473  027412                      1$:
 8474  027412  104417                      TLOADRK                 ;LOAD RK REGS
 8475  027414  104423                      TWAT16                  ;WAIT FOR INTERRUPT
 8476  027416  104002                      ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8477
 8478  027420  104421                      TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8479  027422  104004                      ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8480
 8481  027424  032737  000040  001574      BIT     #S.HDHM,T.MR2   ;TEST IF HEADS HOME
 8482  027432  001767                      BEQ     1$
 8483
 8484  027434  104416                      TSSINIT                 ;CLEAR SUBSYSTEM
 8485  027436  104003                      ERROR   3               ;BAD INIT ERROR
 8486
 8487  027440  005037  001662              CLR     INTSET          ;CLEAR INT FLAG
 8488  027444  104434                      TWAT159                 ;WAIT FOR APPROX 160 MS
 8489  027446  000240                      NOP                     ;DON'T CARE ERROR RETURN
 8490
 8491  027450  104416                      TSSINIT                 ;CLEAR SUBSYSTEM
 8492  027452  104003                      ERROR   3               ;BAD INIT ERROR
 8493
 8494  027454  012762  000100  000000      MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
```

```
 8495
 8496  027462  104437                    TWAT8S                  ;WAIT FOR SECOND INTERRUPT FROM HDS LOADED
 8497  027464  104002                    ERROR   2
 8498
 8499  027466  005037  001616            CLR     L.MR1           ;CLEAR MR1
 8500
 8501  027472  004437  035674            JSR     R4,LRLOAD       ;LOAD "L" REGS
 8502  027476  000105                    CLEAR                   ;CLEAR
 8503  027500  000000                    0                       ;0 WORDS
 8504  027502  000000                    0                       ;0 IS BUFF ADDRESS
 8505  027504     000                    .BYTE   0               ;SECTOR 0
 8506  027505     000                    .BYTE   0               ;TRACK 0
 8507  027506  000012                    12                      ;CYLINDER 12
 8508
 8509  027510  104417                    TLOADRK                 ;LOAD RK REGS
 8510  027512  104423                    TWAT16                  ;WAIT FOR INTERRUPT
 8511  027514  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8512
 8513  027516  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8514  027520  104004                    ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8515
 8516  027522  004437  042260            JSR     R4,GENCOM       ;BUILD HEADERS
 8517  027526  001200                    1200
 8518
 8519  027530  004437  035674            JSR     R4,LRLOAD       ;LOAD "L" REGS
 8520  027534  000127                    WRHEAD                  ;WRHEAD
 8521  027536  177676                    -102                    ;-102 WORDS
 8522  027540  072414                    OBUFF                   ;OBUFF IS BUFF ADDRESS
 8523  027542     000                    .BYTE   0               ;SECTOR 0
 8524  027543     000                    .BYTE   0               ;TRACK 0
 8525  027544  000012                    12                      ;CYLINDER 12
 8526
 8527  027546  104417                    TLOADRK                 ;LOAD RK REGS
 8528  027550  104426                    TWAT64                  ;WAIT FOR INTERRUPT
 8529  027552  104002                    ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8530
 8531  027554  104421                    TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8532  027556  104004                    ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8533
 8534                          ;;*****************************************************************
 8535                          ;*TEST 104      DUMMY TEST FOR PREVIOUS TEST EXIT
 8536                          ;*      THIS TEST IS PRESENT TO MAKE $SW08TB TABLE HAVE AN ENTRY
 8537                          ;*      WHICH RELATES TO "NEWDRV". THIS IS NECESSARY IF AN ERROR OCCURS
 8538                          ;*      IN THE PRECEEDING TEST AND THAT ERROR ABORTS THE TEST.
 8539                          ;*      IF THIS TEST WERE NOT PRESENT, THE PROGRAM WOULD SKIP THE
 8540                          ;*      "NEWDRV" ROUTINE AND GO TO THE TEST FOLLOWING "NEWDRV".
 8541                          ;*
 8542                          ;*      IN ADDITION, THE DRIVE IS CLEARED AND THE HEADS ARE ALLOWED
 8543                          ;*      TO RELOAD. THIS MUST BE DONE TO PREVENT UNEXPECTED INTERRUPTS
 8544                          ;*      FROM THE DRIVE COMING READY AT A LATER TIME.
 8545                          ;;*****************************************************************
 8546  027560  000004         TST104: SCOPE
 8547  027562  012737  000001  001262  MOV     #1,$TIMES       ;;DO 1 ITERATION
 8548
 8549  027570  104416                    TSSINIT                 ;CLEAR SUBSYSTEM
 8550  027572  104003                    ERROR   3               ;BAD INIT ERROR
```

```
 8551
 8552  027574  013762  001626  000010          MOV     DRVNUM,RKCS2(R2) ;LOAD DRIVE NUMBER
 8553  027602  005C37  036126                  CLR     HOLD1
 8554  027606  012737  000001  036124          MOV     #1,HOLD
 8555  027614  053737  001720  036124          BIS     DTYFE,HOLD
 8556  027622  013762  036124  000000          MOV     HOLD,RKCS1(R2)  ;SELECT THE DRIVE
 8557  027630  032762  000200  000012  1$:     BIT     #DRDY,RKDS(R2)  ;TEST IF DRIVE READY
 8558  027636  001004                          BNE     2$              ;BR IF YES
 8559  027640  005237  036126                  INC     HOLD1           ;ELSE TRY AGAIN
 8560  027644  001371                          BNE     1$
 8561  027646  104002                          ERROR   2               ;NO RDY
 8562
 8563  027650                          2$:
 8564  027650  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
 8565  027652  104003                          ERROR   3               ;BAD INIT ERROR
 8566
 8567  027654  004437  042260                  JSR     R4,GENCOM       ,GENERATE HEADERS FOR CYL 0
 8568  027660  001200                          1200
 8569
 8570  027662  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
 8571  027666  000127                          WRHEAD                  ;WRHEAD
 8572  027670  177676                          -102                    ;-102 WORDS
 8573  027672  072414                          OBUFF                   ;OBUFF IS BUFF ADDRESS
 8574  027674     000                          .BYTE   0               ;SECTOR 0
 8575  027675     000                          .BYTE   0               ;TRACK 0
 8576  027676  000000                          0                       ;CYLINDER 0
 8577
 8578  027700  104417                          TLOADRK                 ;LOAD RK REGS
 8579  027702  104426                          TWAT64                  ;WAIT FOR INTERRUPT
 8580  027704  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8581
 8582  027706  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8583  027710  104004                          ERROR   4  ,OR 5, 6, 7  ;REPORT ALL ERRORS
 8584  027712  005037  001676                  CLR     REFMT           ;CLEAR REFORMAT SWITCH
 8585                          .SBTTL  **MULTI-DRIVE OPERATIONS
 8586  027716  000004                  NEWDRV: SCOPE
 8587  027720  012737  000001  001262          MOV     #1,$TIMES       ;DO ONLY ONCE
 8588  027726  032737  000200  001630          BIT     #BIT7,DRVBIT    ;WERE WE TESTING DRIVE 7?
 8589  027734  001026                          BNE     3$              ;YES-SKIP
 8590
 8591  027736  005237  001626          1$:     INC     DRVNUM          ;BUMP TO NEXT SEQUENTIAL ADDRESS
 8592  027742  006337  001630                  ASL     DRVBIT          ;BUMP DRIVEBIT TO THAT POSITION
 8593  027746  033737  001630  001354          BIT     DRVBIT,$DEVM    ;IS THIS DRIVE TO BE TESTED?
 8594  027754  001005                          BNE     2$              ;YES-EXIT
 8595  027756  032737  000400  001630          BIT     #BIT8,DRVBIT    ;ALL DRIVES TESTED?
 8596  027764  001012                          BNE     3$              ;YES-EXIT
 8597  027766  000763                          BR      1$              ;ELSE CHECK NEXT DRIVE AVAILABLE
 8598
 8599  027770  112737  000004  001102  2$:     MOVB    #4,$TSTNM       ;SET TEST NUMBER FOR REPORTS
 8600  027776  013701  001626                  MOV     DRVNUM,R1
 8601  030002  004737  036150                  JSR     PC,GTYP1        ;GET DRV TYP FOR NEXT DRV TO TST
 8602  030006  000137  004720                  JMP     TSTLUP          ;GO TO TEST LOOP TO CHECK THIS DRIVE
 8603  030012  005037  001630          3$:     CLR     DRVBIT          ;CLEAR DRIVE BIT
 8604  030016  005037  001626                  CLR     DRVNUM          ;CLEAR DRIVE NUMBER
 8605
 8606                          ;;*****************************************************************
```

```
 8607                                   ;*TEST 105      RESET ATTENTIONS WITH UNIBUS INIT
 8608                                   ;*
 8609                                   ;*      DO A RECALIBRATE ON ALL AVAILIABLE DRIVES.
 8610                                   ;*      ISSUE A RESET.  MAKE SURE ALL ATTENTION RESET.
 8611                                   ;*
 8612                                   ;;*****************************************************************
 8613  030022  000004                  TST105: SCOPE
 8614  030024  012737  000012  001262          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
 8615  030032  005000                          CLR     R0              ;CLEAR DRIVE POSITION COUNTER
 8616  030034  012701  000001                  MOV     #1,R1           ;PRESET BIT FOR POSITION 0 IN TESTING FOR AVAIL
 8617  030040  013703  001354                  MOV     $DEVM,R3        ;GET DEVICE MAP
 8618  030044  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
 8619  030046  104003                          ERROR   3               ;BAD INIT ERROR
 8620  030050  030103                  1$:     BIT     R1,R3           ;TEST IF THIS DRIVE AVAILABLE
 8621  030052  001006                          BNE     2$              ;YES-SKIP TO SEEK
 8622  030054  006301                  3$:     ASL     R1              ;SHIFT DRIVE SELECT BIT
 8623  030056  005200                          INC     R0              ;BUMP DRIVE POSITION COUNTER
 8624  030060  032701  000400                  BIT     #BIT8,R1        ;ALL DRIVE POSITIONS CHECKED
 8625  030064  001771                          BEQ     1$              ;NO-LOOP
 8626  030066  000443                          BR      4$              ;SKIP TO RESET
 8627
 8628  030070  010037  001610          2$:     MOV     R0,L.CS2        ;LOAD DRIVE NUMBER
 8629  030074  004737  036134                  JSR     PC,GTYPO
 8630  030100  012737  000113  001600          MOV     #RECAL,L.CS1    ;LOAD RECALIBRATE
 8631
 8632  030106  104417                          TLOADRK                 ;LOAD RK REGS
 8633  030110  104423                          TWAT16                  ;WAIT FOR INTERRUPT
 8634  030112  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 8635
 8636  030114  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
 8637  030120  012705  000764                  MOV     #500.,R5        ;SET COUNT FOR 8 SECONDS
 8638  030124  012762  000012  000000          MOV     #12,RKCS1(R2)   ;RESET INTERRUPT ENABLE
 8639  030132  016237  000016  001556  12$:    MOV     RKASOF(R2),T.ASOF ;GET ATTENTION REGISTER
 8640  030140  113704  001557                  MOVB    T.ASOF+1,R4     ;ADJUST FOR CHECK OF ATTENTION
 8641  030144  042704  177400                  BIC     #177400,R4      ;CLEAR UNUSED BITS
 8642  030150  030104                          BIT     R1,R4           ;CHECK IF ATT SET FROM DRIVE RECAL'ED
 8643  030152  001006                          BNE     10$             ;YES - SKIP
 8644
 8645  030154  104423                          TWAT16                  ;WAIT FOR 16 MS
 8646  030156  000240                          NOP                     ;DON'T CARE RETURNS
 8647  030160  000240                          NOP
 8648  030162  005305                          DEC     R5              ;TATOL WAIT TIME IS 8 SECONDS
 8649  030164  001362                          BNE     12$             ;CHECK ATTENTION EACH 16 MS
 8650  030166  104002                          ERROR   2               ;REPORT IF NO ATTENTION IN 8 SEC
 8651
 8652  030170                          10$:
 8653  030170  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 8654  030172  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 8655
 8656  030174  000727                          BR      3$              ;LOOP FOR NEXT DRIVE
 8657  030176  000005                  4$:     RESET                   ;UNIBUS RESET
 8658  030200  004737  045146                  JSR     PC,$TKINI       ;RESET KEYBOARD INTERRUPT ENABLE
 8659
 8660  030204  012701  000031                  MOV     #25.,R1         ;DO A SHORT DELAY
 8661  030210  005301                  5$:     DEC     R1
 8662  030212  001376                          BNE     5$
```

```
8663  030214  004737  034522          JSR     PC,OPTTST       ;SET UP OPTIONS
8664
8665  030220  104420                  TGETRK                  ;GET RK611 REGS
8666
8667  030222  105737  001557          TSTB    T.ASOF+1        ;ALL ATTENTION RESET?
8668  030226  001407                  BEQ     6$              ;YES-SKIP
8669
8670  030230  012737  056540  001470  MOV     #EMDA,EM12N     ;'DRIVE ATT NOT RESET RESULT OF
8671  030236  012737  056664  061022  MOV     #EMUR,DF011A    ;UNIBUS RESET''
8672  030244  104012                  ERROR   12
8673
8674  030246                      6$:
8675                              ;;*****************************************************************
8676                              ;*TEST 106      RESET ATTENTIONS WITH SUBSYSTEM CLEAR
8677                              ;*
8678                              ;*      DO A RECALIBRATE ON ALL AVAILABLE DRIVES.
8679                              ;*      ISSUE A SUBSYSTEM CLEAR.  MAKE SURE ALL ATTENTIONS
8680                              ;*      RESET.
8681                              ;*
8682                              ;;*****************************************************************
8683  030246  000004              TST106: SCOPE
8684  030250  012737  000012  001262  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
8685  030256  005000                  CLR     R0              ;CLEAR DRIVE POSITION COUNTER
8686  030260  012701  000001          MOV     #1,R1           ;PRESET TO TEST POSITION 0
8687  030264  013703  001354          MOV     $DEVM,R3        ;CUT DEVICE MAP
8688  030270  104416                  TSSINIT                 ;CLEAR SUBSYSTEM
8689  030272  104003                  ERROR   3               ;BAD INIT ERROR
8690  030274  030103              1$: BIT     R1,R3           ;THIS DRIVE AVAILABLE?
8691  030276  001006                  BNE     2$              ;YES-SKIP TO SEEK
8692  030300  006301              3$: ASL     R1              ;SHIFT TO NEXT DRIVE POSITION
8693  030302  005200                  INC     R0              ;DUMP POSITION COUNTER
8694  030304  032701  000400          BIT     #BIT8,R1        ;ALL POSITIONS CHECKED
8695  030310  001771                  BEQ     1$              ;NO-LOOP
8696  030312  000443                  BR      4$              ;YES-SKIP TO CLEAR
8697
8698  030314  010037  001610      2$: MOV     R0,L.CS2        ;LOAD DRIVE NUMBER
8699  030320  004737  036134          JSR     PC,GTYP0
8700  030324  012737  000113  001600  MOV     #RECAL,L.CS1    ;LOAD RECALIBRATE
8701  030332  104417                  TLOADRK                 ;LOAD RK REGS
8702  030334  104423                  TWAT16                  ;WAIT FOR INTERRUPT
8703  030336  104002                  ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8704
8705  030340  005037  001662          CLR     INTSET          ;CLEAR INT FLAG
8706  030344  012705  000764          MOV     #500.,R5        ;SET COUNT FOR 8 SECONDS
8707  030350  012762  000012  000000  MOV     #12,RKCS1(R2)   ;RESET INTERRUPT ENABLE
8708  030356  016237  000016  001556  12$: MOV  RKASOF(R2),T.ASOF ;GET ATTENTION REGISTER
8709  030364  113704  001557          MOVB    T.ASOF+1,R4     ;ADJUST FOR CHECK OF ATTENTION
8710  030370  042704  177400          BIC     #177400,R4      ;CLEAR UNUSED BITS
8711  030374  030104                  BIT     R1,R4           ;CHECK IF ATT SET FROM DRIVE RECAL'ED
8712  030376  001006                  BNE     10$             ;YES - SKIP
8713
8714  030400  104423                  TWAT16                  ;WAIT FOR 16 MS
8715  030402  000240                  NOP                     ;DON'T CARE RETURNS
8716  030404  000240                  NOP
8717  030406  005305                  DEC     R5              ;TATOL WAIT TIME IS 8 SECONDS
8718  030410  001362                  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$:     BIS     #SCLR,RKCS2(R2) ;DO SUBSYSTEM CLEAR
                                MOV     #25.,R1         ;DO A SHORT DELAY
                        5$:     DEC     R1
                                BNE     5$

                               'TGETRK                  ;GET RK611 REGS

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

                                MOV     #EMDA,EM12N     ;"DRIVE ATT NOT RESET AS RESULT OF
                                MOV     #EMSCLR,DF011A  ;SUBSYSTEM CLEAR"
                                ERROR   12

                        6$:
                        ;;*****************************************************************
                        ;*TEST 107      SVAL AND ATTENTION FROM OTHER DRIVE
                        ;*
                        ;*      DO A RECALIBRATE ON ONE AVAILABLE DRIVE.  DO A SELECT
                        ;*      ON ANOTHER AVAILABLE DRIVE.  MAKE SURE STATUS VALID
                        ;*      IS SET.  WAIT FOR SECOND INTERRUPT FROM RECALIBRATE
                        ;*      MAKE SURE STATUS VALID REMAINS SET AND DRIVE STATUS
                        ;*      CHANGE REMAINS RESET.
                        ;*
                        ;*      REPEAT FOR ALL COMBINATIONS OF TWO AVAILIABLE DRIVES.
                        ;*
                        ;*      NOTE:   THIS TEST WILL ONLY BE DONE IF AT LEAST
                        ;*              TWO DRIVES ARE AVAILABLE.
                        ;*
                        ;*****************************************************************
000004                  TST107: SCOPE
000012  001262                 MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
001354                          MOV     $DEVM,-(SP)     ;PUT DEVICE MAP ON STACK
036242                          JSR     R4,BITCNT       ;COUNT NUMBER OF BITS(# OF DRIVES)
000001                          CMP     (SP)+,#1        ;COMPARE TO 1
                                BHI     2$              ;SKIP IF MORE THAN 1
001304                          TST     $PASS           ;CHECK IF PASS 0
                                BNE     1$              ;NO-SKIP

052574                          TYPE    ,OPR018         ;"OVERLAPPED OPERATION BYPASSED"
031772                  1$:     JMP     $EOP            ;GET OUT.

177777  001224          2$:     MOV     #-1,$TMP1       ;SET LOOP CONTROL FLAG
001354                          MOV     $DEVM,R5        ;GET DEVICE MAP
                                CLR     R0              ;CLEAR FOR DRIVE #A
                                CLR     R1              ;CLEAR FOR DRIVE #B
000001                          MOV     #1,R3           ;SET DRIVE POSITION A
000001                          MOV     #1,R4           ;SET DRIVE POSITION B
```

```
8775  030560  012737  030570  001110          MOV     #3$,$LPERR      ;SET LOCAL LOOP ON ERROR
8776  030566  000503                          BR      11$             ;GO SET UP POINTERS
8777
8778  030570                          3$:
8779  030570  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
8780  030572  104003                          ERROR   3               ;BAD INIT ERROR
8781
8782  030574  010037  001610                  MOV     R0,L.CS2        ;LOAD DRIVE A
8783  030600  010037  001626                  MOV     R0,DRVNUM       ;LOAD FOR REPORT
8784  030604  004737  036134                  JSR     PC,GTYP0
8785  030610  012737  000113  001600          MOV     #RECAL,L.CS1    ;LOAD RECAL
8786
8787  030616  104417                          TLOADRK                 ;LOAD RK REGS
8788  030620  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8789  030622  104002                          ERROR   2               ;TL SLOW/NOT COMPLETE ERROR
8790  030624  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERROPS
8791  030626  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8792
8793  030630  005037  001662                  CLR     INTSET          ;CLEAR INT FLAG
8794
8795  030634  010137  001610                  MOV     R1,L.CS2        ;LOAD DRIVE B
8796  030640  010137  001626                  MOV     R1,DRVNUM       ;LOAD FOR REPORT
8797  030644  004737  036150                  JSR     PC,GTYP1
8798  030650  012737  000101  001600          MOV     #SELDRV,L.CS1   ;LOAD DRIVE SELECT
8799
8800  030656  104417                          TLOADRK                 ;LOAD RK REGS
8801  030660  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8802  030662  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8803
8804  030664  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8805  030666  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8806
8807  030670  032737  100000  001552          BIT     #SVAL,T.DS      ;CHECK IF STATUS VALID SET
8808  030676  001007                          BNE     4$              ;YES - SKIP
8809  030700  012737  056354  001460          MOV     #EMSVAL,EM11N   ;"STATUS VALID NOT SET RESULT OF
8810  030706  012737  050160  061022          MOV     #OPER00,DF011A  ;DRIVE SELECT"
8811  030714  104011                          ERROR   11
8812
8813  030716  005037  001662          4$:     CLR     INTSET          ;CLEAR INT FLAG
8814  030722  104436                          TWAT2S                  ;WAIT FOR SEEK COMPLETE INTERRUPT
8815  030724  000401                          BR      44$             ;NONE RECEIVED - SKIP
8816  030726  000406                          BR      55$             ; RECEIVED - SKIP
8817
8818  030730  010037  001626          44$:    MOV     R0,DRVNUM       ;SET DRIVE FOR REPORT
8819  030734  012737  060411  001372          MOV     #DH017,DH2N     ;"COMMAND - SELECT AFTER RECAL"
8820  030742  104002                          ERROR   2
8821
8822  030744  104420                  55$:    TGETRK                  ;GET RK REGS
8823  030746  032737  100000  001552          BIT     #SVAL,T.DS      ;TEST IF SVAL STILL SET
8824  030754  001007                          BNE     5$              ;YES - SKIP
8825
8826  030756  012737  056354  001510          MOV     #EMSVAL,EM14N   ;"STATUS VALID RESET RESULT OF
8827  030764  012737  060434  061022          MOV     #DH018,DF011A   ;RECAL COMPLETE ATTENTION AFTER SEL"
8828  030772  104014                          ERROR   14
8829
8830  030774  104415                  5$:     SCOP1                   ;LOCAL LOOP TO 3$
```

```
                                ;       THE FOLLOWING CODE CAUSES THE TEST TO BE RUN ON EVERY COMBINATION
                                ;       OF DRIVES AVAILABLE.  THE FIRST PASS OF THE PROGRAM WILL USE THE
                                ;       LOWEST NUMBER DRIVE AS A AND THE NEXT HIGHER NUMBER DRIVE AS
                                ;       B.  THE SECOND PASS SWAPS DRIVE A & B.  THE THIRD PASS USES
                                ;       THE LOWEST NUMBER DRIVE AS B AND THE 3RD HIGHEST NUMBER DRIVE
                                ;       AS A.  THE FORTH PASS SWAPS A & B AGAIN.  THIS CONTINUES
                                ;       UNTIL ALL DRIVES HAVE BEEN TESTED WITH THE LOWEST NUMBER
                                ;       DRIVE.
                                ;
                                ;       THE SECOND HIGHEST NUMBER DRIVE IS THEN USED AS A AND THE
                                ;       THIRD HIGHEST AS B.  THEY ARE SWAPPED ON THE NEXT PASS.
                                ;
                                ;       THIS TECHNIQUE IS CONTINUED UNTIL ALL COMBINATIONS ARE
                                ;       CHECKED.

030776 005237 001224            11$:    INC     $TMP1           ;INCREMENT PASS CONTROL
031002 001024                           BNE     16$             ;SKIP IF NOT ZERO
                                                                ;(IT WILL BE ZERO ON THE 1ST PASS)

031004 030305                   12$:    BIT     R3,R5           ;TEST IF BIT POSITION FOR A AT AVAIL DRIVE
031006 001006                           BNE     13$             ;YES-SKIP

031010 005200                   22$:    INC     R0              ;BUMP R0 (DRIVE A)
031012 006303                           ASL     R3              ;SHIFT DRIVE SELECT BIT ONE POSITION
031014 032703 000400                    BIT     #BIT8,R3        ;IF BIT 8 IS SET, ALL DRIVES HAVE
031020 001771                           BEQ     12$             ;BEEN CHECKED; IF NOT CHECK NEXT POSITION
031022 000464                           BR      50$             ;DONE-EXIT

031024 010001                   13$:    MOV     R0,R1           ;SET DRIVE B TO THE SAME AS A
031026 010304                           MOV     R3,R4
031030 005201                   14$:    INC     R1              ;BUMP R1 (DRIVE B)
031032 006304                           ASL     R4              ;SHIFT SELECTOR BIT ONE POSITION
031034 030405                           BIT     R4,R5           ;IS THIS DRIVE AVAIL?
031036 001004                           BNE     15$             ;YES-SKIP
031040 032704 000400                    BIT     #BIT8,R4        ;WERE ALL POSITIONS CHECKED?
031044 001771                           BEQ     14$             ;NO-LOOP
031046 000452                           BR      50$             ;DONE-EXIT

031050 000137 030570            15$:    JMP     3$              ;GO DO THE TEST ON THE DRIVE A & B
                                                                ;CONTAINED IN R0 & R1
031054 032737 000001 001224     16$:    BIT     #BIT0,$TMP1     ;IS PASS FLAGS ODD?
031062 001410                           BEQ     17$             ;NO-SKIP

031064 010046                           MOV     R0,-(SP)        ;
031066 010346                           MOV     R3,-(SP)        ;SWAP R0 & R1, R3 & R4
031070 010403                           MOV     R4,R3           ;TO EXCHANGE DRIVE A & B
031072 010100                           MOV     R1,R0
031074 012604                           MOV     (SP)+,R4
031076 012601                           MOV     (SP)+,R1
031100 000137 030570                    JMP     3$              ;REPEAT TEST ON THIS COMBO.

031104 032737 000002 001224     17$:    BIT     #BIT1,$TMP1     ;TEST IF PASS FLAGS AT HALF MODULE 4?
031112 001410                           BEQ     19$             ;NO-SKIP TO BUMP DRIVE B
031114 005200                   18$:    INC     R0              ;BUMP DRIVE A
031116 006303                           ASL     R3              ;SHIFT DRIVE SELECT BIT
```

```
8887  031120  030305                    BIT     R3,R5           ;AVAILABLE?
8888  031122  001014                    BNE     20$             ;YES-SKIP
8889  031124  032703  000400            BIT     #BIT8,R3        ;ALL CHECKED?
8890  031130  001771                    BEQ     18$             ;NO-SKIP
8891  031132  000412                    BR      21$             ;GO TO NEXT PASS
8892
8893  031134  005201              19$:  INC     R1              ;BUMP DRIVE B
8894  031136  006304                    ASL     R4              ;SHIFT DRIVE SELECT BIT
8895  031140  030405                    BIT     R4,R5           ;AVAILABLE?
8896  031142  001004                    BNE     20$             ;YES-SKIP
8897  031144  032704  000400            BIT     #BIT8,R4        ;ALL CHECKED?
8898  031150  001771                    BEQ     19$             ;NO-LOOP
8899  031152  000404                    BR      23$             ;YES-SKIP TO NEXT PASS
8900
8901  031154  000137  030570      20$:  JMP     3$              ;GO TEST THIS COMBO
8902
8903  031160  010100              21$:  MOV     R1,R0           ;SET DRIVE 0 TO LOW POSITION THIS PASS
8904  031162  010403                    MOV     R4,R3           ;SET SELECT BITS TO AGREE
8905  031164  005037  001224      23$:  CLR     $TMP1           ;CLEAR PASS FLAGS
8906  031170  000137  031010            JMP     22$             ;GO SET UP A & B
8907  031174                      50$:
8908                              ;;*****************************************************************
8909                              ;*TEST 110      OVERLAPPED OPERATIONS
8910                              ;*
8911                              ;*      DO A RECALIBRATE ON BOTH DRIVE A AND DRIVE B.  ISSUE  A
8912                              ;*      SEEK ON DRIVE A TO CYLINDER 1.  IMMEDIATELY ISSUE A WRITE
8913                              ;*      DATA TO CYLINDER 100, TRACK 0, HEAD 0 ON DRIVE B.
8914                              ;*      AT THE END OF THE DATA TRANSFER NO ERRORS SHOULD
8915                              ;*      BE SET AND DRIVE A HAS ATTENTION SET.
8916                              ;*
8917                              ;*      REPEAT FOR ALL COMBINATIONS OF TWO DRIVES.
8918                              ;*
8919                              ;*      NOTE:   IF ONLY ONE DRIVE IS AVAILABLE THE
8920                              ;*              TEST WILL NOT BE DONE.
8921                              ;*
8922                              ;*****************************************************************
8923  031174  000004              TST110: SCOPE
8924  031176  012737  000005  001262      MOV     #5.,$TIMES      ;;DO 5. ITERATIONS
8925
8926  031204  012737  177777  001224  2$: MOV     #-1,$TMP1       ;SET LOOP CONTROL FLAG
8927  031212  013705  001354            MOV     $DEVM,R5        ;GET DEVICE MAP
8928  031216  005000                    CLR     R0              ;CLEAR FOR DRIVE #A
8929  031220  005001                    CLR     R1              ;CLEAR FOR DRIVE #B
8930  031222  012703  000001            MOV     #1,R3           ;SET DRIVE POSITION A
8931  031226  012704  000001            MOV     #1,R4           ;SET DRIVE POSITION B
8932  031232  012737  031242  001110    MOV     #3$,$LPERR      ;SET LOCAL LOOP ON ERROR
8933  031240  000555                    BR      11$             ;GO SET UP POINTERS
8934  031242                      3$:
8935  031242  104416                    TSSINIT                 ;CLEAR SUBSYSTEM
8936  031244  104003                    ERROR   3               ;BAD INIT ERROR
8937
8938  031246  010037  001626            MOV     R0,DRVNUM       ;STORE DRIVE FOR REPORT
8939  031252  010037  001610            MOV     R0,L.CS2        ;SETUP DRIVE A TO RECAL
8940  031256  004737  036134            JSR     PC,GTYP0
8941  031262  012737  000113  001600    MOV     #RECAL,L.CS1
8942
```

```
8943  031270  104417                          TLOADRK                 ;LOAD RK REGS
8944  031272  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8945  031274  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8946  031276  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
8947
8948  031302  104437                          TWAT8S                  ;WAIT FOR SECOND INTERRUPT
8949  031304  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8950
8951  031306  012737  000105  001600          MOV     #CLEAR,L.CS1    ;SET UP TO CLEAR DRIVE
8952  031314  104417                          TLOADRK                 ;LOAD RK REGS
8953  031316  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8954  031320  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8955  031322                          4$:
8956  031322  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8957  031324  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8958  031326  032737  040000  001540          BIT     #DI,T.CS1       ;TEST IF DI STILL SET
8959  031334  001372                          BNE     4$              ;YES - LOOP
8960  031336  010137  001626                  MOV     R1,DRVNUM       ;STORE DRIVE FOR REPORT
8961  031342  010137  001610                  MOV     R1,L.CS2        ;SETUP DRIVE B TO RECAL
8962  031346  004737  036150                  JSR     PC,GTYP1
8963  031352  012737  000113  001600          MOV     #RECAL,L.CS1
8964
8965  031360  104417                          TLOADRK                 ;LOAD RK REGS
8966  031362  104423                          TWAT16                  ;WAIT  OR INTERRUPT
8967  031364  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8968  031366  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT FLAG
8969  031372  104437                          TWAT8S                  ;WAIT FOR SECOND INTERRUPT
8970  031374  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8971  031376  012737  000105  001600          MOV     #CLEAR,L.CS1    ;SET UP DRIVE CLEAR
8972  031404  104417                          TLOADRK                 ;LOAD RK REGS
8973  031406  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8974  031410  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8975  031412                          5$:
8976  031412  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
8977  031414  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
8978  031416  032737  040000  001540          BIT     #DI,T.CS1       ;TEST IF DI STILL SET
8979  031424  001372                          BNE     5$              ;YES - LOOP
8980
8981  031426  010037  001626                  MOV     R0,DRVNUM       ;STORE DRIVE FOR REPORT
8982  031432  010037  001610                  MOV     R0,L.CS2        ;SETUP DRIVE A TO SEEK
8983  031436  004737  036134                  JSR     PC,GTYP0
8984  031442  012737  000001  001614          MOV     #1,L.DCYL       ;TO CYL 1
8985  031450  012737  000117  001600          MOV     #SEEK,L.CS1
8986
8987  031456  104417                          TLOADRK                 ;LOAD RK REGS
8988  031460  104423                          TWAT16                  ;WAIT FOR INTERRUPT
8989  031462  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
8990
8991  031464  010137  001626                  MOV     R1,DRVNUM       ;STORE DRIVE FOR REPORT
8992  031470  010137  001610                  MOV     R1,L.CS2        ;SETUP DRIVE B TO WRITE DATA
8993  031474  004737  036150                  JSR     PC,GTYP1
8994  031500  012737  000100  001614          MOV     #100,L.DCYL     ;AT CYL 100
8995  031506  012737  177400  001602          MOV     #-400,L.WC      ;400 WORDS
8996  031514  012737  072414  001604          MOV     #OBUFF,L.BA
8997  031522  012737  000123  001600          MOV     #WRDATA,L.CS1
8998
```

```
 8999  031530  104417                          TLOADRK                 ;LOAD RK REGS-DO WRITE
 9000
 9001  031532  104427                          TWAT8C                  ;WAIT FOR INTERRUPT
 9002  031534  104002                          ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
 9003
 9004  031536  104421                          TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
 9005  031540  104004                          ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
 9006
 9007  031542  010037  001626                  MOV     R0,DRVNUM       ;STORE DRIVE FOR REPORT
 9008  031546  130337  001557                  BITB    R3,T.ASOF+1     ;CHECK IF DRIVE ATTENTION IS DRIVE A
 9009  031552  001007                          BNE     10$             ;YES-SKIP
 9010  031554  012737  056540  001460          MOV     #EMDA,EM11N     ;"DRIVE ATTENTION NOT SET RESULT OF
 9011  031562  012737  050272  061022          MOV     #EMSK,DF011A    ;SEEK"
 9012  031570  104011                          ERROR   11
 9013
 9014  031572  104415                  10$:    SCOP1                   ;LOCAL LOOP TO 3$
 9015
 9016                                  ;       THE FOLLOWING CODE CAUSES THE TEST TO BE RUN ON EVERY COMBINATION
 9017                                  ;       OF DRIVES AVAILABLE.  THE FIRST PASS OF THE PROGRAM WILL USE THE
 9018                                  ;       LOWEST NUMBER DRIVE AS A AND THE NEXT HIGHER NUMBER DRIVE AS
 9019                                  ;       B.  THE SECOND PASS SWAPS DRIVE A & B.  THE THIRD PASS USES
 9020                                  ;       THE LOWEST NUMBER DRIVE AS B AND THE 3RD HIGHEST NUMBER DRIVE
 9021                                  ;       AS A.  THE FORTH PASS SWAPS A & B AGAIN.  THIS CONTINUES
 9022                                  ;       UNTIL ALL DRIVES HAVE BEEN TESTED WITH THE LOWEST NUMBER
 9023                                  ;       DRIVE.
 9024                                  ;
 9025                                  ;       THE SECOND HIGHEST NUMBER DRIVE IS THEN USED AS A AND THE
 9026                                  ;       THIRD HIGHEST AS B.  THEY ARE SWAPPED ON THE NEXT PASS.
 9027                                  ;
 9028                                  ;       THIS TECHNIQUE IS CONTINUED UNTIL ALL COMBINATIONS ARE
 9029                                  ;       CHECKED.
 9030
 9031  031574  005237  001224          11$:    INC     $TMP1           ;INCREMENT PASS CONTROL
 9032  031600  001024                          BNE     16$             ;SKIP IF NOT ZERO
 9033                                                                  ;(IT WILL BE ZERO ON THE 1ST PASS)
 9034
 9035  031602  030305                  12$:    BIT     R3,R5           ;TEST IF BIT POSITION FOR A AT AVAIL DRIVE
 9036  031604  001006                          BNE     13$             ;YES-SKIP
 9037
 9038  031606  005200                  22$:    INC     R0              ;BUMP R0 (DRIVE A)
 9039  031610  006303                          ASL     R3              ;SHIFT DRIVE SELECT BIT ONE POSITION
 9040  031612  032703  000400                  BIT     #BIT8,R3        ;IF BIT 8 IS SET, ALL DRIVES HAVE
 9041  031616  001771                          BEQ     12$             ;BEEN CHECKED; IF NOT CHECK NEXT POSITION
 9042  031620  000464                          BR      50$             ;DONE-EXIT
 9043
 9044  031622  010001                  13$:    MOV     R0,R1           ;SET DRIVE B TO THE SAME AS A
 9045  031624  010304                          MOV     R3,R4
 9046  031626  005201                  14$:    INC     R1              ;BUMP R1 (DRIVE B)
 9047  031630  006304                          ASL     R4              ;SHIFT SELECTOR BIT ONE POSITION
 9048  031632  030405                          BIT     R4,R5           ;IS THIS DRIVE AVAIL?
 9049  031634  001004                          BNE     15$             ;YES-SKIP
 9050  031636  032704  000400                  BIT     #BIT8,R4        ;WERE ALL POSITIONS CHECKED?
 9051  031642  001771                          BEQ     14$             ;NO-LOOP
 9052  031644  000452                          BR      50$             ;DONE-EXIT
 9053
 9054  031646  000137  031242          15$:    JMP     3$              ;GO DO THE TEST ON THE DRIVE A & B
```

```
 9055                                                           ;CONTAINED IN R0 & R1
 9056  031652  032737  000001  001224  16$:    BIT     #BIT0,$TMP1     ;IS PASS FLAGS ODD?
 9057  031660  001410                         BEQ     17$             ;NO-SKIP
 9058
 9059  031662  010046                         MOV     R0,-(SP)        ;
 9060  031664  010346                         MOV     R3,-(SP)        ;SWAP R0 & R1, R3 & R4
 9061  031666  010403                         MOV     R4,R3           ;TO EXCHANGE DRIVE A & B
 9062  031670  010100                         MOV     R1,R0
 9063  031672  012604                         MOV     (SP)+,R4
 9064  031674  012601                         MOV     (SP)+,R1
 9065  031676  000137  031242                 JMP     3$              ;REPEAT TEST ON THIS COMBO.
 9066
 9067  031702  032737  000002  001224  17$:    BIT     #BIT1,$TMP1     ;TEST IF PASS FLAGS AT HALF MODULE 4?
 9068  031710  001410                         BEQ     19$             ;NO-SKIP TO BUMP DRIVE B
 9069  031712  005200                 18$:    INC     R0              ;BUMP DRIVE A
 9070  031714  006303                         ASL     R3              ;SHIFT DRIVE SELECT BIT
 9071  031716  030305                         BIT     R3,R5           ;AVAILABLE?
 9072  031720  001014                         BNE     20$             ;YES-SKIP
 9073  031722  032703  000400                 BIT     #BIT8,R3        ;ALL CHECKED?
 9074  031726  001771                         BEQ     18$             ;NO-SKIP
 9075  031730  000412                         BR      21$             ;GO TO NEXT PASS
 9076
 9077  031732  005201                 19$:    INC     R1              ;BUMP DRIVE B
 9078  031734  006304                         ASL     R4              ;SHIFT DRIVE SELECT BIT
 9079  031736  030405                         BIT     R4,R5           ;AVAILABLE?
 9080  031740  001004                         BNE     20$             ;YES-SKIP
 9081  031742  032704  000400                 BIT     #BIT8,R4        ;ALL CHECKED?
 9082  031746  001771                         BEQ     19$             ;NO-LOOP
 9083  031750  000404                         BR      23$             ;YES-SKIP TO NEXT PASS
 9084
 9085  031752  000137  031242         20$:    JMP     3$              ;GO TEST THIS COMBO
 9086
 9087  031756  010100                 21$:    MOV     R1,R0           ;SET DRIVE 0 TO LOW POSITION THIS PASS
 9088  031760  010403                         MOV     R4,R3           ;SET SELECT BITS TO AGREE
 9089  031762  005037  001224         23$:    CLR     $TMP1           ;CLEAR PASS FLAGS
 9090  031766  000137  031606                 JMP     22$             ;GO SET UP A & B
 9091  031772                         50$:
 9092                                 ;ADD WAITING LOOP FOR APT OPERATION ON JAN-9-78 REV 007
 9093                                 .SBTTL  END OF PASS ROUTINE
 9094
 9095                                 ;;**********************************************************************
 9096                                 ;*INCREMENT THE PASS NUMBER ($PASS)
 9097                                 ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
 9098                                 ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
 9099                                 ;*IF THERES A MONITOR GO TO IT
 9100                                 ;*IF THERE ISN'T JUMP TO TSTAPT
 9101
 9102  031772                         $EOP:
 9103  031772  000004                         SCOPE
 9104  031774  005037  001102                 CLR     $TSTNM          ;;ZERO THE TEST NUMBER
 9105  032000  005037  001262                 CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
 9106  032004  005237  001304                 INC     $PASS           ;;INCREMENT THE PASS NUMBER
 9107  032010  042737  100000  001304         BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 9108  032016  005327                         DEC     (PC)+           ;;LOOP?
 9109  032020  000001                 $EOPCT: .WORD   1
 9110  032022  003063                         BGT     $DOAGN          ;;YES
```

```
 9111  032024  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 9112  032026  000001                  $ENDCT: .WORD   1
 9113  032030  032020                          $EOPCT
 9114  032032  104401  032040                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 9115  032036  000407                          BR      64$             ;;GET OVER THE ASCIZ
 9116                                  ;;65$:  .ASCIZ  <12><15>/END PASS #/
 9117  032056                          64$:
 9118  032056  013746  001304                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
 9119                                                                  ;;TYPE PASS NUMBER
 9120  032062  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 9121  032064  104401  032072                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 9122  032070  000421                          BR      66$             ;;GET OVER THE ASCIZ
 9123                                  ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
 9124  032134                          66$:
 9125  032134  013746  001112                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
 9126                                                                  ;;TOTAL NUMBER OF ERRORS
 9127  032140  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 9128  032142  104401  001273                  TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
 9129  032146  005037  001112                  CLR     $ERTTL          ;;CLEAR ERROR TOTAL
 9130  032152  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
 9131  032156  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 9132  032160  000005                          RESET                   ;;CLEAR THE WORLD
 9133  032162  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
 9134  032164  000240                          NOP                     ;;SAVE ROOM
 9135  032166  000240                          NOP                     ;;FOR
 9136  032170  000240                          NOP                     ;;ACT11
 9137  032172                          $DOAGN:
 9138  032172  000137                          JMP     @(PC)+          ;;RETURN
 9139  032174  032202                  $RTNAD: .WORD   TSTAPT
 9140  032176     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 9141             032202                       .EVEN
 9142  032202  122737  000001  001316  TSTAPT: CMPB    #APTENV,$ENV    ;RUNNING UNDER APT MODE ?
 9143  032210  001007                          BNE     2$              ;BRANCH IF NOT
 9144  032212  023727  001304  000002          CMP     $PASS,#2        ;TWO PASS DONE ?
 9145  032220  103403                          BLO     2$              ;BRANCH IF NOT
 9146  032222  005237  001102          1$:     INC     $TSTNM          ;INCREMENT TEST NUMBER
 9147  032226  000775                          BR      1$              ;WAITING LOOP FOR APT DUMP THE NEXT PROG
 9148  032230  000137  003150          2$:     JMP     NEWPAS          ;REPEAT THE SAME TEST IF NOT UNDER APT
 9149                                  .SBTTL  ROUTINE TO SIZE MEMORY
 9150
 9151                                  ;;******************************************************************
 9152                                  ;*CALL:
 9153                                  ;*      JSR     PC,$SIZE
 9154                                  ;*      RETURN
 9155                                  ;*$LSTAD WILL CONTAIN:
 9156                                  ;*      WITH KT11 OPTION        -- LAST VIRTUAL ADDRESS OF THE LAST BANK
 9157                                  ;*      WITHOUT KT11 OPTION     -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
 9158                                  ;*$LSTBK WILL CONTAIN THE LAST BANK AS A SAF
 9159                                  ;*$KT11 IS THE MEMORY MANAGEMENT KEY
 9160                                  ;*BIT07 = 0 DON'T USE MEMORY MANAGEMENT
 9161                                  ;*      MUST BE SETUP BEFORE THE CALL
 9162                                  ;*BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
 9163                                  ;*      DETERMINED BY ROUTINE
 9164
 9165  032234  010046          $SIZE:  MOV     R0,-(SP)        ;;SAVE R0 ON THE STACK
 9166  032236  010146                  MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
```

```
 9167  032240  010246                          MOV     R2,-(SP)        ;;SAVE R2 ON THE STACK
 9168  032242  010346                          MOV     R3,-(SP)        ;;SAVE R3 ON THE STACK
 9169  032244  013746  000114                  MOV     @#114,-(SP)     ;;SAVE MEMORY ERROR VECTOR PS & PC
 9170  032250  013746  000116                  MOV     @#116,-(SP)
 9171  032254  012737  000116  000114          MOV     #116,@#114      ;;IGNORE PARITY ERRORS WHILE SIZING
 9172  032262  012737  000002  000116          MOV     #RTI,@#116
 9173  032270  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE PRESENT ERROR VECTOR PS & PC
 9174  032274  013746  000006                  MOV     @#ERRVEC+2,-(SP)
 9175  032300  010600                          MOV     SP,R0           ;;SAVE THE STACK POINTER
 9176                          ;;SET THE ERRVEC PS TO THE PRESENT PS
 9177  032302  104400                          TRAP                    ;;PUSH OLD PSW AND PC ON STACK
 9178  032304  012637  000006                  MOV     (SP)+,@#ERRVEC+2        ;;SAVE THE PSW IN @#ERRVEC+2
 9179  032310  012701  003776                  MOV     #3776,R1        ;;SETUP ADDRESS
 9180  032314  105727                          TSTB    (PC)+           ;;USE MEMORY MANAGEMENT?
 9181  032316  000200                $KT11:    .WORD   200             ;;SET TO USE MEMORY MANAGEMENT
 9182  032320  100062                          BPL     $CORE           ;;BR IF NO
 9183  032322  012737  032460  000004          MOV     #$KTNEX,@#ERRVEC ;;SET FOR TIMEOUT
 9184  032330  005737  177572                  TST     @#SR0           ;;KT11 ARE YOU THERE?
 9185  032334  052737  100000  032316          BIS     #100000,$KT11   ;;YES--SET KT11 KEY
 9186  032342  005046                          CLR     -(SP)           ;;INITIALIZE FOR "PAR" LOADING
 9187  032344  012702  172340                  MOV     #KIPAR0,R2      ;;ADDRESS OF FIRST "PAR"
 9188  032350  012703  000010                  MOV     #^D8,R3         ;;LOAD EIGHT "PAR.'S" AND EIGHT "PDR.'S"
 9189  032354  012762  077406  177740  1$:     MOV     #77406,-40(R2)  ;;PDR = 4K, UP, READ/WRITE
 9190  032362  011622                          MOV     (SP),(R2)+      ;;LOAD "PAR"
 9191  032364  062716  000200                  ADD     #200,(SP)       ;;UPDATE FOR NEXT "PAR"
 9192  032370  077307                          SOB     R3,1$           ;;LOOP UNTIL ALL EIGHT ARE LOADED
 9193  032372  012742  177600                  MOV     #177600,-(R2)   ;;SETUP KIPAR7 FOR I/O
 9194  032376  005042                          CLR     -(R2)           ;;SETUP KIPAR6 FOR TESTING
 9195  032400  012737  032416  000004          MOV     #2$,@#ERRVEC    ;;CATCH TIMEOUT IF NO SR3
 9196  032406  012737  000020  172516          MOV     #20,@#SR3       ;;ENABLE 22 BIT MODE
 9197  032414  000401                          BR      3$              ;;THIS PDP-11 HAS A SR3 REGISTER
 9198  032416  022626                  2$:     CMP     (SP)+,(SP)+     ;;CLEAN OFF THE STACK--NO SR3
 9199  032420  005237  177572          3$:     INC     @#SR0           ;;TURN ON MEMORY MANAGEMENT
 9200  032424  012737  032450  000004          MOV     #$KTOUT,@#ERRVEC ;;SET FOR TIME OUT
 9201  032432  005737  143776          4$:     TST     @#143776        ;;TRAP ON NON-EX-MEM
 9202  032436  062712  000040                  ADD     #40,(R2)        ;;MAKE A 1K STEP
 9203  032442  023712  172356                  CMP     @#KIPAR7,(R2)   ;;LAST ONE?
 9204  032446  101371                          BHI     4$              ;;NO--TRY IT
 9205  032450  011202                $KTOUT:   MOV     (R2),R2         ;;GET LAST BANK+1
 9206  032452  005037  177572                  CLR     @#SR0           ;;TURN OFF MEMORY MANAGEMENT
 9207  032456  000421                          BR      $SIZEX
 9208  032460  042737  100000  032316  $KTNEX: BIC     #100000,$KT11   ;;KT11 NON-EXISTENT
 9209  032466  012737  032516  000004  $CORE:  MOV     #$CROUT,@#ERRVEC ;;SET FOR TIMEOUT
 9210  032474  005002                          CLR     R2              ;;SET UP BANK
 9211  032476  062701  004000          1$:     ADD     #4000,R1        ;;INCREMENT BY 1K
 9212  032502  062702  000040                  ADD     #40,R2          ;;1K STEP
 9213  032506  005711                          TST     (R1)            ;;TRAP ON TIME OUT
 9214  032510  022701  177776                  CMP     #177776,R1      ;;LAST ONE
 9215  032514  001370                          BNE     1$              ;;NO--TRY AGAIN
 9216  032516  162701  004000          $CROUT: SUB     #4000,R1
 9217  032522  162702  000040          $SIZEX: SUB     #40,R2          ;;DROP BACK
 9218  032526  010006                          MOV     R0,SP           ;;RESTORE THE STACK
 9219  032530  012637  000006                  MOV     (SP)+,@#ERRVEC+2 ;;RESTORE ERROR VECTOR
 9220  032534  012637  000004                  MOV     (SP)+,@#ERRVEC
 9221  032540  012637  000116                  MOV     (SP)+,@#116     ;;RESTORE MEMORY ERROR VECTOR
 9222  032544  012637  000114                  MOV     (SP)+,@#114
```

```
 9223  032550  010137  032572                MOV     R1,$LSTAD       ;;LAST ADDRESS
 9224  032554  010237  032574                MOV     R2,$LSTBK       ;;LAST BANK
 9225  032560  012603                        MOV     (SP)+,R3        ;;RESTORE R3
 9226  032562  012602                        MOV     (SP)+,R2        ;;RESTORE R2
 9227  032564  012601                        MOV     (SP)+,R1        ;;RESTORE R1
 9228  032566  012600                        MOV     (SP)+,R0        ;;RESTORE R0
 9229  032570  000207                        RTS     PC
 9230  032572  000000              $LSTAD:   .WORD   0               ;;CONTAINS THE LAST ADDRESS
 9231  032574  000000              $LSTBK:   .WORD   0               ;;CONTAINS THE LAST BANK
 9232                                .SBTTL  SCOPE HANDLER ROUTINE
 9233
 9234                                ;;**************************************************************************
 9235                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
 9236                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
 9237                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
 9238                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 9239                                ;*SW14=1        LOOP ON TEST
 9240                                ;*SW11=1        INHIBIT ITERATIONS
 9241                                ;*SW09=1        LOOP ON ERROR
 9242                                ;*SW08=1        LOOP ON TEST IN SWR<7:0>
 9243                                ;*CALL
 9244                                ;*      SCOPE           ;;SCOPE=IOT
 9245
 9246  032576                        $SCOPE:
 9247  032576  104407                        CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
 9248                                ;;GO TO ERROR ROUTINE IF RETURN PC LESS THAN 1002
 9249                                ;;OTHERWISE CONTINUE
 9250  032600  021627  001002                CMP     (SP),#1002      ;;UNEXPECTED TRAP OR INTERRRUPT
 9251  032604  101002                        BHI     1$              ;;ARE TRAPPED HERE VIA IOT
 9252  032606  000137  033614                JMP     $ERROR          ;;GO PROCESS UNEXPECTED TRAP
 9253  032612  032777  040000  146320  1$:   BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
 9254  032620  001131                        BNE     $OVER           ;;YES IF SW14=1
 9255                                ;#####START OF CODE FOR THE XOR TESTER#####
 9256  032622  000416              $XTSTR:   BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
 9257                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
 9258  032624  013746  000004                MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
 9259  032630  012737  032650  000004        MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
 9260  032636  005737  177060                TST     @#177060        ;;TIME OUT ON XOR?
 9261  032642  012637  000004                MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 9262  032646  000500                        BR      $SVLAD          ;;GO TO THE NEXT TEST
 9263  032650  022626              5$:       CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
 9264  032652  012637  000004                MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 9265  032656  000440                        BR      7$              ;;LOOP ON THE PRESENT TEST
 9266  032660                        6$:;#####END OF CODE FOR THE XOR TESTER#####
 9267  032660  032777  000400  146252        BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
 9268  032666  001421                        BEQ     2$              ;;BR IF NO
 9269  032670  005046                        CLR     -(SP)           ;;CLEAR A TEMP. LOCATION
 9270  032672  117716  146242                MOVB    @SWR,(SP)       ;;PICKUP THE DESIRED TEST NUMBER
 9271  032676  001414                        BEQ     8$              ;;BRANCH IF BAD TEST NUMBER IN SWR
 9272  032700  022716  000110                CMP     #110,(SP)       ;;CHECK THE NUMBER IN THE SWR
 9273  032704  002411                        BLT     8$              ;;BRANCH IF TEST NUMBER IS OUT OF RANGE
 9274  032706  011637  001102                MOV     (SP),$TSTNM     ;;UPDATE THE TEST NUMBER
 9275  032712  005316                        DEC     (SP)            ;;BACKUP BY ONE
 9276  032714  006316                        ASL     (SP)            ;;SCALE THE TEST NUMBER AS AN INDEX
 9277  032716  062716  033122                ADD     #$SW08TBL,(SP)  ;;FORM THE ADDRESS OF TEST POINTER
 9278  032722  013637  001106                MOV     @(SP)+,$LPADR   ;;SET LOOP ADDRESS TO DESIRED TEST
```

```
 9279  032726  000466                          BR      $OVER           ;;GO LOOP ON THE TEST
 9280  032730  005726                  8$:     TST     (SP)+           ;;CLEAN THE BAD TEST NUMBER OFF OF THE STACK
 9281  032732  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
 9282  032736  001421                          BEQ     3$              ;;BR IF NO
 9283  032740  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
 9284  032746  101C15                          BHI     3$              ;;BR IF NO
 9285  032750  032777  001000  146162          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
 9286  032756  001404                          BEQ     4$              ;;BR IF NO
 9287  032760  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
 9288  032766  000446                          BR      $OVER
 9289  032770  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
 9290  032774  005037  001262                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
 9291  033000  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
 9292  033002  032777  004000  146130  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
 9293  033010  001011                          BNE     1$              ;;BR IF YES
 9294  033012  005737  001304                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
 9295  033016  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
 9296  033020  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
 9297  033024  023737  001262  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
 9298  033032  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
 9299  033034  012737  000C01  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
 9300  033042  013737  033120  001262          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
 9301  033050  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
 9302  033054  113737  001102  001302          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
 9303  033062  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 9304  033066  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 9305  033072  005037  001264                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 9306  033076  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 9307  033104  013777  001102  146030  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 9308  033112  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 9309  033116  000002                          RTI                     ;;FIXES PS
 9310  033120  003720                  $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
 9311  033122                          $SW08TBL:
 9312  033122  003164                          .WORD   TST1+2          ;;STARTING ADDRESS OF TEST 1
 9313  033124  003302                          .WORD   TST2+2          ;;STARTING ADDRESS OF TEST 2
 9314  033126  003370                          .WORD   TST3+2          ;;STARTING ADDRESS OF TEST 3
 9315  033130  004620                          .WORD   TST4+2          ;;STARTING ADDRESS OF TEST 4
 9316  033132  004726                          .WORD   TST5+2          ;;STARTING ADDRESS OF TEST 5
 9317  033134  005104                          .WORD   TST6+2          ;;STARTING ADDRESS OF TEST 6
 9318  033136  005232                          .WORD   TST7+2          ;;STARTING ADDRESS OF TEST 7
 9319  033140  005336                          .WORD   TST10+2         ;;STARTING ADDRESS OF TEST 10
 9320  033142  006262                          .WORD   TST11+2         ;;STARTING ADDRESS OF TEST 11
 9321  033144  006400                          .WORD   TST12+2         ;;STARTING ADDRESS OF TEST 12
 9322  033146  006516                          .WORD   TST13+2         ;;STARTING ADDRESS OF TEST 13
 9323  033150  006760                          .WORD   TST14+2         ;;STARTING ADDRESS OF TEST 14
 9324  033152  007112                          .WORD   TST15+2         ;;STARTING ADDRESS OF TEST 15
 9325  033154  007322                          .WORD   TST16+2         ;;STARTING ADDRESS OF TEST 16
 9326  033156  007512                          .WORD   TST17+2         ;;STARTING ADDRESS OF TEST 17
 9327  033160  007646                          .WORD   TST20+2         ;;STARTING ADDRESS OF TEST 20
 9328  033162  010514                          .WORD   TST21+2         ;;STARTING ADDRESS OF TEST 21
 9329  033164  011112                          .WORD   TST22+2         ;;STARTING ADDRESS OF TEST 22
 9330  033166  011324                          .WORD   TST23+2         ;;STARTING ADDRESS OF TEST 23
 9331  033170  011542                          .WORD   TST24+2         ;;STARTING ADDRESS OF TEST 24
 9332  033172  011666                          .WORD   TST25+2         ;;STARTING ADDRESS OF TEST 25
 9333  033174  012020                          .WORD   TST26+2         ;;STARTING ADDRESS OF TEST 26
 9334  033176  012176                          .WORD   TST27+2         ;;STARTING ADDRESS OF TEST 27
```

```
 9335  033200  012330                    .WORD   TST30+2         ;;STARTING ADDRESS OF TEST 30
 9336  033202  012624                    .WORD   TST31+2         ;;STARTING ADDRESS OF TEST 31
 9337  033204  013020                    .WORD   TST32+2         ;;STARTING ADDRESS OF TEST 32
 9338  033206  013254                    .WORD   TST33+2         ;;STARTING ADDRESS OF TEST 33
 9339  033210  013442                    .WORD   TST34+2         ;;STARTING ADDRESS OF TEST 34
 9340  033212  013674                    .WORD   TST35+2         ;;STARTING ADDRESS OF TEST 35
 9341  033214  014062                    .WORD   TST36+2         ;;STARTING ADDRESS OF TEST 36
 9342  033216  014304                    .WORD   TST37+2         ;;STARTING ADDRESS OF TEST 37
 9343  033220  014472                    .WORD   TST40+2         ;;STARTING ADDRESS OF TEST 40
 9344  033222  014664                    .WORD   TST41+2         ;;STARTING ADDRESS OF TEST 41
 9345  033224  015060                    .WORD   TST42+2         ;;STARTING ADDRESS OF TEST 42
 9346  033226  015264                    .WORD   TST43+2         ;;STARTING ADDRESS OF TEST 43
 9347  033230  015470                    .WORD   TST44+2         ;;STARTING ADDRESS OF TEST 44
 9348  033232  016330                    .WORD   TST45+2         ;;STARTING ADDRESS OF TEST 45
 9349  033234  016534                    .WORD   TST46+2         ;;STARTING ADDRESS OF TEST 46
 9350  033236  016740                    .WORD   TST47+2         ;;STARTING ADDRESS OF TEST 47
 9351  033240  017156                    .WORD   TST50+2         ;;STARTING ADDRESS OF TEST 50
 9352  033242  017362                    .WORD   TST51+2         ;;STARTING ADDRESS OF TEST 51
 9353  033244  017566                    .WORD   TST52+2         ;;STARTING ADDRESS OF TEST 52
 9354  033246  017712                    .WORD   TST53+2         ;;STARTING ADDRESS OF TEST 53
 9355  033250  020036                    .WORD   TST54+2         ;;STARTING ADDRESS OF TEST 54
 9356  033252  020162                    .WORD   TST55+2         ;;STARTING ADDRESS OF TEST 55
 9357  033254  020306                    .WORD   TST56+2         ;;STARTING ADDRESS OF TEST 56
 9358  033256  020432                    .WORD   TST57+2         ;;STARTING ADDRESS OF TEST 57
 9359  033260  020504                    .WORD   TST60+2         ;;STARTING ADDRESS OF TEST 60
 9360  033262  020564                    .WORD   TST61+2         ;;STARTING ADDRESS OF TEST 61
 9361  033264  021050                    .WORD   TST62+2         ;;STARTING ADDRESS OF TEST 62
 9362  033266  021232                    .WORD   TST63+2         ;;STARTING ADDRESS OF TEST 63
 9363  033270  021420                    .WORD   TST64+2         ;;STARTING ADDRESS OF TEST 64
 9364  033272  022124                    .WORD   TST65+2         ;;STARTING ADDRESS OF TEST 65
 9365  033274  022342                    .WORD   TST66+2         ;;STARTING ADDRESS OF TEST 66
 9366  033276  022446                    .WORD   TST67+2         ;;STARTING ADDRESS OF TEST 67
 9367  033300  023070                    .WORD   TST70+2         ;;STARTING ADDRESS OF TEST 70
 9368  033302  023512                    .WORD   TST71+2         ;;STARTING ADDRESS OF TEST 71
 9369  033304  023622                    .WORD   TST72+2         ;;STARTING ADDRESS OF TEST 72
 9370  033306  024470                    .WORD   TST73+2         ;;STARTING ADDRESS OF TEST 73
 9371  033310  025162                    .WORD   TST74+2         ;;STARTING ADDRESS OF TEST 74
 9372  033312  025404                    .WORD   TST75+2         ;;STARTING ADDRESS OF TEST 75
 9373  033314  025662                    .WORD   TST76+2         ;;STARTING ADDRESS OF TEST 76
 9374  033316  025742                    .WORD   TST77+2         ;;STARTING ADDRESS OF TEST 77
 9375  033320  026134                    .WORD   TST100+2                ;;STARTING ADDRESS OF TEST 100
 9376  033322  026360                    .WORD   TST101+2                ;;STARTING ADDRESS OF TEST 101
 9377  033324  026506                    .WORD   TST102+2                ;;STARTING ADDRESS OF TEST 102
 9378  033326  027030                    .WORD   TST103+2                ;;STARTING ADDRESS OF TEST 103
 9379  033330  027562                    .WORD   TST104+2                ;;STARTING ADDRESS OF TEST 104
 9380  033332  030024                    .WORD   TST105+2                ;;STARTING ADDRESS OF TEST 105
 9381  033334  030250                    .WORD   TST106+2                ;;STARTING ADDRESS OF TEST 106
 9382  033336  030470                    .WORD   TST107+2                ;;STARTING ADDRESS OF TEST 107
 9383  033340  031176                    .WORD   TST110+2                ;;STARTING ADDRESS OF TEST 110
 9384  033342  031774                    .WORD   $EOP+2          ;ADDRESS OF END OF PASS
 9385  033344  044172                    .WORD   ABTFAIL+2       ;ADDRESS OF ABORT FAILURE HANDLER
 9386
 9387                          .SBTTL  APT COMMUNICATIONS ROUTINE
 9388
 9389                          ;;**************************************************************************
 9390  033346  112737  000001  033612  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
```

```
 9391  033354  112737  000001  033610  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
 9392  033362  000403                  BR      $ATYC
 9393  033364  112737  000001  033612  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
 9394  033372                          $ATYC:
 9395  033372  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9396  033374  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9397  033376  105737  033610                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
 9398  033402  001450                          BEQ     5$              ;;IF NOT:  BR
 9399  033404  122737  000001  001316          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
 9400  033412  001031                          BNE     3$              ;;IF NOT:  BR
 9401  033414  132737  000100  001317          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
 9402  033422  001425                          BEQ     3$              ;;IF NOT:  BR
 9403  033424  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
 9404  033430  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 9405  033436  005737  001276          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
 9406  033442  001375                          BNE     1$              ;;IF NOT:  WAIT
 9407  033444  010037  001312                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
 9408  033450  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
 9409  033452  001376                          BNE     2$
 9410  033454  163700  001312                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
 9411  033460  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
 9412  033462  010037  001314                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
 9413  033466  012737  000004  001276          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
 9414  033474  000413                          BR      5$
 9415  033476  017637  000004  033522  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
 9416  033504  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
 9417  033512  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
 9418  033516  004737  044202                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
 9419  033522  000000                  4$:     .WORD   0
 9420  033524                          5$:
 9421  033524  105737  033612          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
 9422  033530  001416                          BEQ     12$             ;;IF NOT:  BR
 9423  033532  005737  001316                  TST     $ENV            ;;RUNNING UNDER APT?
 9424  033536  001413                          BEQ     12$             ;;IF NOT:  BR
 9425  033540  005737  001276          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
 9426  033544  001375                          BNE     11$             ;;IF NOT:  WAIT
 9427  033546  017637  000004  001300          MOV     @4(SP),$FATAL   ;;GET ERROR #
 9428  033554  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 9429  033562  005237  001276                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
 9430  033566  105037  033612          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
 9431  033572  105037  033611                  CLRB    $LFLG           ;;CLEAR LOG FLAG
 9432  033576  105037  033610                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
 9433  033602  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 9434  033604  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 9435  033606  000207                          RTS     PC              ;;RETURN
 9436  033610     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
 9437  033611     000          $LFLG:  .BYTE   0               ;;LOG FLAG
 9438  033612     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
 9439          033614                  .EVEN
 9440          000200          APTSIZE=200
 9441          000001          APTENV=001
 9442          000100          APTSPOOL=100
 9443          000040          APTCSUP=040
 9444                          .SBTTL  ERROR HANDLER ROUTINE
 9445
 9446                          ;;**************************************************************
```

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

033614                                  $ERROR:
033614  104407                                CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
033616  105237  001103                  7$:   INCB    $ERFLG            ;;SET THE ERROR FLAG
033622  001775                                BEQ     7$                ;;DON'T LET THE FLAG GO TO ZERO
033624  013777  001102  145310                MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER AND ERROR FLAG
033632  032777  002000  145300                BIT     #BIT10,@SWR       ;;BELL ON ERROR?
033640  001402                                BEQ     1$                ;;NO - SKIP
033642  104401  001266                        TYPE    ,$BELL            ;;RING BELL
033646  005237  001112                  1$:   INC     $ERTTL            ;;COUNT THE NUMBER OF ERRORS
033652  011637  001116                        MOV     (SP),$ERRPC       ;;GET ADDRESS OF ERROR INSTRUCTION
033656  162737  000002  001116                SUB     #2,$ERRPC
033664  117737  145226  001114                MOVB    @$ERRPC,$ITEMB    ;;STRIP AND SAVE THE ERROR ITEM CODE
033672  032777  020000  145240                BIT     #BIT13,@SWR       ;;SKIP TYPEOUT IF SET
033700  001055                                BNE     20$               ;;SKIP TYPEOUTS
033702  021627  001002                        CMP     (SP),#1002        ;;IF RETURN PC LESS THAN 1002
033706  101046                                BHI     12$               ;;ERROR IS ILLEGAL TRAP
                                        ;;PROCESS UNEXPECTED TRAP OR INTERRUPT
033710  016637  000004  001116                MOV     4(SP),$ERRPC      ;;GET PC AT TIME OF FALSE TRAP
033716  162737  000002  001116                SUB     #2,$ERRPC         ;;ADJUST PC
033724  104401  033770                        TYPE    ,10$              ;;TYPE HEADER
033730  013746  001116                        MOV     $ERRPC,-(SP)      ;;SAVE $ERRPC FOR TYPEOUT
033734  104402                                TYPOC                     ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
033736  104401  033776                        TYPE    ,11$
033742  162716  000004                        SUB     #4,(SP)           ;;GET FALSE TRAP VECTOR ADDR
033746  011637  001116                        MOV     (SP),$ERRPC
033752  013746  001116                        MOV     $ERRPC,-(SP)      ;;SAVE $ERRPC FOR TYPEOUT
033756  104402                                TYPOC                     ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
033760  104401  001273                        TYPE    ,$CRLF
033764  022626                                CMP     (SP)+,(SP)+       ;;POP FALSE TRAP VECTOR PC&ADDR
033766  000422                                BR      20$
033770  050200  036503  000040        10$:    .ASCIZ  <200>'PC= '
033776  020040  047125  054105        11$:    .ASCIZ  '  UNEXPECTED TRAP TO '
034004  042520  052103  042105
034012  052040  040522  020120
034020  047524  000040
                                        .EVEN
034024                                  12$:
034024  004737  034136                        JSR     PC,TYPERR         ;;GO TO USER ERROR ROUTINE
034030  104401  001273                        TYPE    ,$CRLF
034034                                  20$:
034034  122737  000001  001316                CMPB    #APTENV,$ENV      ;;RUNNING IN APT MODE
034042  001007                                BNE     2$                ;;NO,SKIP APT ERROR REPORT
034044  113737  001114  034056                MOVB    $ITEMB,21$        ;;SET ITEM NUMBER AS ERROR NUMBER
034052  004737  033364                        JSR     PC,$ATY4          ;;REPORT FATAL ERROR TO APT
034056     000                  21$:    .BYTE   0
```

```
 9503  034057  000                        .BYTE   0
 9504  034060  000777              22$:    BR      22$             ;;APT ERROR LOOP
 9505  034062  005777  145052      2$:     TST     @SWR            ;;HALT ON ERROR
 9506  034066  100002                      BPL     3$              ;;SKIP IF CONTINUE
 9507  034070  000000                      HALT                    ;;HALT ON ERROR!
 9508  034072  104407                      CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 9509  034074  032777  001000  145036  3$: BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 9510  034102  001402                      BEQ     4$              ;;BR IF NO
 9511  034104  013716  001110              MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 9512  034110  005737  001264      4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
 9513  034114  001402                      BEQ     5$              ;;BR IF NONE
 9514  034116  013716  001264              MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
 9515  034122                      5$:
 9516  034122  022737  032162  000042      CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
 9517  034130  001001                      BNE     6$              ;;BRANCH IF NO
 9518  034132  000000                      HALT                    ;;YES
 9519  034134                      6$:
 9520  034134  000002                      RTI                     ;;RETURN
 9521                          ;;**************************************************************
 9522                          .SBTTL  TYPE ERROR ROUTINE
 9523                          ;*ENTRY JSR PC,TYPERR
 9524                          ;*RETURN RTS PC
 9525                          ;*
 9526                          ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
 9527                          ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
 9528                          ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
 9529                          ;*THE ERROR.
 9530                          ;;**************************************************************
 9531  034136  104413          TYPERR: SAVREG
 9532  034140  113737  001102  001302  MOVB    $TSTNM,$TESTN   ;GET TEST NUMBER FOR REPORT
 9533  034146  042737  177400  001302  BIC     #177400,$TESTN  ;CLEAR UNUSED BITS
 9534  034154  113700  001114          MOVB    $ITEMB,R0       ;ENTER ERROR NUMBER
 9535  034160  042700  177400          BIC     #177400,R0      ;CLEAR UNUSED BITS
 9536  034164  005300                  DEC     R0              ;FORM INDEX FOR ERROR TABLE
 9537  034166  006300                  ASL     R0
 9538  034170  006300                  ASL     R0
 9539  034172  006300                  ASL     R0
 9540  034174  062700  001360  1$:     ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERROR ENTRY
 9541  034200  012037  034214          MOV     (R0)+,2$        ;GET EM POINTER
 9542  034204  001404                  BEQ     3$              ;BRANCH IF THERE ISN'T ONE
 9543  034206  104401  001273          TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
 9544  034212  104401                  TYPE                    ;TYPE ERROR MESSAGE (EM)
 9545  034214  000000          2$:     .WORD   0               ;EM POINTER GOES HERE
 9546  034216  012037  034232  3$:     MOV     (R0)+,4$        ;GET DH POINTER
 9547  034222  001404                  BEQ     5$              ;BRANCH IF THERE ISN'T ONE
 9548  034224  104401  001273          TYPE    ,$CRLF          ;TYPE CR-LF
 9549  034230  104401                  TYPE                    ;TYPE DATA HEADER
 9550  034232  000000          4$:     .WORD   0               ;DH POINTER GOES HERE
 9551  034234  012001          5$:     MOV     (R0)+,R1        ;GET DT POINTER
 9552  034236  001445                  BEQ     20$             ;BRANCH IF THERE ARE NONE
 9553  034240  005004                  CLR     R4              ;SET INDENT SWITCH
 9554  034242  012000                  MOV     (R0)+,R0        ;GET DF POINTER
 9555  034244  012002                  MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
 9556  034246  104401  001273          TYPE    ,$CRLF
 9557  034252  112003          10$:    MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
 9558  034254  105720                  TSTB    (R0)+           ;BUMP PAST FORMAT WORD
```

```
 9559  034256  005703                        TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
 9560  034260  001416                        BEQ     14$             ;NO - SKIP DATA PRINT
 9561  034262  005704                        TST     R4              ;CHECK FOR INDENT
 9562  034264  001004                        BNE     12$             ;YES, GO INDENT
 9563  034266  013146                11$:    MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
 9564  034270  104402                        TYPOC                   ;TYPE IT
 9565  034272  005303                        DEC     R3              ;MORE DATA WORDS
 9566  034274  001403                        BEQ     13$             ;NO-BRANCH
 9567  034276  104401  050763        12$:    TYPE    ,SPACE2         ;TYPE SEPARATORS
 9568  034302  000771                        BR      11$             ;LOOP
 9569  034304  104401  001273        13$:    TYPE    ,$CRLF          ;TYPE <CR><LF>
 9570  034310  005710                        TST     (R0)            ;CHECK IF NEXT HEADER AVAILABLE
 9571  034312  001401                        BEQ     14$             ;NO, DO NOT CHANGE INDENT
 9572  034314  005104                        COM     R4              ;CHANGE INDENT
 9573  034316  005302                14$:    DEC     R2              ;MORE DH'S?
 9574  034320  003414                        BLE     20$             ;NO-BRANCH
 9575  034322  012037  034342        15$:    MOV     (R0)+,18$       ;GET NEXT DH POINTER
 9576  034326  001751                        BEQ     10$             ;IF 0 GET DATA
 9577  034330  005704                        TST     R4              ;INDENT?
 9578  034332  001402                        BEQ     17$             ;NO, BRANCH
 9579  034334  104401  050763                TYPE    ,SPACE2         ;TYPE INDENT
 9580  034340  104401                17$:    TYPE                    ;TYPE DH
 9581  034342  000000                18$:    .WORD   0               ;DH POINTER GOES HERE
 9582  034344  104401  001273                TYPE    ,$CRLF
 9583  034350  000740                        BR      10$             ;GET DATA
 9584  034352  104414                20$:    RESREG
 9585  034354  005237  001632                INC     ERRCNT          ;INCREMENT THE ERROR COUNT
 9586  034360  032777  010000  144552        BIT     #SW12,@SWR      ;CHECK IF SWITCH 12 SET
 9587  034366  001421                        BEQ     25$             ;NO, RETURN
 9588  034370  022737  000024  001632        CMP     #20.,ERRCNT     ;CHECK IF ERROR THRESHOLD EXCEEDED
 9589  034376  103015                        BHIS    25$             ;NO, RETURN
 9590  034400  104401  053227                TYPE    ,ABORT          ;TYPE "PROGRAM ABORTED BECAUSE
 9591                                                                ;ERROR THRESHOLD EXCEEDED"
 9592  034404  005737  000042                TST     42              ;CHECK IF CHAIN MODE
 9593  034410  001407                        BEQ     22$             ;NO, HALT PROCESSOR
 9594  034412  012706  001100                MOV     #STACK,SP       ;INITIALIZE STACK
 9595  034416  012737  000001  032020        MOV     #1,$EOPCT       ;FORCE END OF PROGRAM
 9596  034424  000137  031772                JMP     $EOP
 9597  034430  000000                22$:    HALT
 9598  034432  000207                25$:    RTS     PC
 9599                                .SBTTL  NON-EXISTANT MEMORY AND INTERRUPT CHECK HANDLER
 9600                                ;*      THIS ROUTINE SETS THE INTERRUPT FLAG AND DOES AN RTI.
 9601                                ;*      THIS IS THE INDICATION TO THE ROUTINE CHECKING
 9602                                ;*      NON-EXIXTANT MEMORY THAT AN INTERRUPT DID OCCUR.
 9603
 9604  034434  005237  001662        NEXINT: INC     INTSET          ;BUMP THE INTERRUPT COUNTER
 9605  034440  000002                        RTI
 9606
 9607                                .SBTTL  RK611 INTERRUPT HANDLER
 9608                                ;*      MOST INTERRUPTS FROM THE RK611 ARE HANDLED BY THIS ROUTINE.  ACTUAL
 9609                                ;*      PROCESSING AS A RESULT OF THE INTERRUPT IS LEFT TO THE MAIN
 9610                                ;*      PROGRAM.  THE HANDLER JUST SETS A FLAG TO INDICATE THE
 9611                                ;*      INTERRUPT OCCURRED.
 9612
 9613  034442  000240                INTHLR: NOP
 9614  034444  005237  001662                INC     INTSET          ;BUMP THE INTERRUPT FLAG
```

```
 9615  034450  000002                          RTI                     ;RETURN.
 9616
 9617                                  .SBTTL  MEMORY PARITY ERROR TRAP HANDLER
 9618                                  ;*      MEMORY PARITY TRAPS WILL BE REPORTED BY THIS ROUTINE. THE REPORT
 9619                                  ;*      WILL INCLUDE THE PC AT TIME OF FAILURE AND ABORT THE PROGRAM.
 9620
 9621  034452  032777  020000  144460  PERHLR: BIT     #BIT13,@SWR     ;TEST IF INHIBIT REPORT
 9622  034460  001003                          BNE     1$              ;YES - SKIP
 9623  034462  104401  053456                  TYPE    ,EM3            ;TYPE PARITY ERROR MESSAGE
 9624  034466  104402                          TYPOC                   ;AND PC VALUE
 9625  034470  012737  000001  032020  1$:     MOV     #1,$EOPCT       ;FORCE END OF PROGRAM
 9626  034476  012706  001100                  MOV     #STACK,SP       ;CLEAN OFF STACK
 9627  034502  000137  031772                  JMP     $EOP
 9628
 9629                                  .SBTTL  LINE CLOCK INTERRUPT HANDLER
 9630                                  ;*      THE LINE CLOCK INTERRUPT HANDLER WILL INCREMENT THE LCLKTK
 9631                                  ;*      (LINE CLOCK TICK COUNTER) EACH TIME AN INTERRUPT OCCURS.
 9632
 9633  034506  005237  001674          LCKHLR: INC     LCLKTK          ;INCREMENT CLOCK TICK COUNTER
 9634  034512  042777  000200  145170          BIC     #BIT7,@KWLADD   ;CLEAR MONITOR BIT
 9635  034520  000002                          RTI
 9636
 9637                                  ;;*******************************************************************
 9638                                  .SBTTL  OPTION PRESENT TEST AND SETUP
 9639                                  ;*      THIS ROUTINE CHECKS IF THE MEMORY PARITY OPTION AND THE
 9640                                  ;*      LINE CLOCK ARE ON THE SYSTEM. FLAGS ARE SET IF PRESENT; CLEARED
 9641                                  ;*      OTHERWISE, AND THE APPROPRIATE INTERRUPT VECTORS ARE SET UP.
 9642  034522  104413                  OPTTST: SAVREG
 9643  034524  013746  000004                  MOV     ERRVEC,-(SP)    ;STORE OLD NEM CONTENTS
 9644  034530  013746  000006                  MOV     ERRVEC+2,-(SP)
 9645  034534  012737  034754  000004          MOV     #20$,ERRVEC     ;DET INTERRUPT FOR NEM
 9646  034542  012737  000340  000006          MOV     #PR7,ERRVEC+2   ;SET PRIORITY
 9647  034550  005037  001670                  CLR     MEMPAR          ;CLEAR PARITY WORD FLAGS
 9648  034554  042737  000100  001664          BIC     #PARPRE,OPTFLG  ;CLEAR PARITY PRESENT FLAG
 9649  034562  005737  170200                  TST     170200          ;TEST IF EITHER 11/70 OR 11/44
 9650  034566  000240                          NOP
 9651  034570  000240                          NOP
 9652  034572  005737  177770                  TST     177770          ;NOW SEE IF IT'S 11/70
 9653  034576  000240                          NOP
 9654  034600  000240                          NOP
 9655  034602  012737  177750  001672          MOV     #177750,CSRPTR  ;SET POINTER FOR 11/70
 9656  034610  052737  002000  001664          BIS     #CP1170,OPTFLG  ;SET 11/70 FLAG
 9657  034616  052737  000100  001664          BIS     #PARPRE,OPTFLG  ;SET PARITY PRESENT FLAG
 9658  034624  000507                          BR      35$             ;GO TO VECTOR SETUP
 9659
 9660  034626  013703  001714          3$:     MOV     MEMBAS,R3       ;SET UP POINTER TO FIRST PARITY CSR
 9661  034632  012704  000001                  MOV     #1,R4           ;INIT MASK
 9662  034636  012713  000001          6$:     MOV     #1,(R3)         ;SET PARITY DETECT IN THAT CSR
 9663  034642  005713                          TST     (R3)
 9664  034644  052713  000002                  BIS     #BIT1,(R3)      ;IS THIS ECC MEMORY?
 9665  034650  032713  000002                  BIT     #BIT1,(R3)      ;
 9666  034654  001061                          BNE     47$             ;YES - EXIT
 9667  034656  050437  001670                  BIS     R4,MEMPAR       ;SET PARITY PRESENT BIT
 9668                                  ;       BIT     #PARPRE,OPTFLG  ;WAS PARITY PRESENT SET BEFORE
 9669                                  ;       BNE     10$             ;YES - SKIP
 9670  034662  013700  001716                  MOV     MMVECA,R0       ;SET UP FOR PARITY TRAPS
```

```
 9671  034666  012720  034766                  MOV     #40$,(R0)+      ;  TO 40$
 9672  034672  012710  000340                  MOV     #PR7,(R0)
 9673  034676  012700  072640                  MOV     #OBUFF+224,R0   ;SET POINTER TO BUFFER WHERE BAD PARITY
 9674                                                                  ;  IS USED IN THE TESTS
 9675  034702  012713  000004                  MOV     #BIT2,(R3)      ;SET TO WRITE WRONG PARITY
 9676  034706  012710  177777                  MOV     #-1,(R0)        ;WRITE WORD BAD
 9677  034712  012713  000001                  MOV     #1,(R3)         ;SET TO DETECT PARITY ERROR
 9678  034716  005710                          TST     (R0)            ;READ BAD WORD
 9679  034720  042737  000100  001664          BIC     #PARPRE,OPTFLG  ;REV 005
 9680  034726  012713  000002                  MOV     #2,(R3)         ;TRUN OFF CZR ERROR ENABLE
 9681  034732  012710  000000                  MOV     #0,(R0)         ;CLEAR BAD PARITY
 9682                                  ;       BIC     #7,(R3)         ;REV 006
 9683                                  ;       MOV     #0,(R0)         ;REV 006
 9684  034736  000442                          BR      35$             ;REV 005
 9685  034740  062703  000002          10$:    ADD     #2,R3           ;BUMP TO NEXT CSR
 9686  034744  000241                          CLC
 9687  034746  006104                          ROL     R4              ;SHIFT MASK
 9688  034750  001332                          BNE     6$              ;TEST IF ALL DONE
 9689  034752  000434                          BR      35$             ;YES - SKIP
 9690
 9691  034754  022626                  20$:    CMP     (SP)+,(SP)+     ;CLEAR STACK
 9692  034756  012737  035040  000004          MOV     #30$,ERRVEC     ;SET NEW NEM TRAP ADDRESS
 9693  034764  000720                          BR      3$              ;GO TO CSR CHECKS
 9694
 9695  034766  022626                  40$:    CMP     (SP)+,(SP)+     ;CLEAR STACK
 9696  034770  010337  001672                  MOV     R3,CSRPTR       ;SET CSR POINTER FOR CSR TO BE USED
 9697  034774  052737  000100  001664          BIS     #PARPRE,OPTFLG  ;SET PARITY PRESENT FLAG
 9698  035002  012713  000000                  MOV     #0,(R3)         ;TURN OFF CSR ERROR ENABLE
 9699  035006  012710  000000                  MOV     #0,(R0)         ;WRITE GOOD PARITY
 9700  035012  012713  000001                  MOV     #1,(R3)         ;SET TO DETECT PARITY ERRORS
 9701  035016  000412                          BR      35$             ;EXIT
 9702
 9703  035020  012713  000000          47$:    MOV     #0,(R3)         ;CLEAR CSR
 9704  035024  042737  000100  001664          BIC     #PARPRE,OPTFLG  ;FORGET THEM ALL
 9705  035032  005037  001670                  CLR     MEMPAR          ;FORGET THEM ALL
 9706  035036  000402                          BR      35$             ;EXIT
 9707  035040  022626                  30$:    CMP     (SP)+,(SP)+     ;CLEAR STACK
 9708  035042  000736                          BR      10$             ;GO CHECK NEXT CSR
 9709  035044  013700  001716          35$:    MOV     MMVECA,R0       ;SET UP POINTER TO PARITY VECTOR
 9710  035050  005737  001670                  TST     MEMPAR          ;TEST IF ANY PARITY PRESENT
 9711  035054  001004                          BNE     37$             ;YES - SKIP
 9712  035056  032737  002000  001664          BIT     #CP1170,OPTFLG  ;TEST IF 11/70
 9713  035064  001405                          BEQ     39$             ;NO - SKIP
 9714  035066  012720  034452          37$:    MOV     #PERHLR,(R0)+   ;SET UP PARITY VECTOR
 9715  035072  012710  000340                  MOV     #PR7,(R0)       ;SET PRIORITY
 9716  035076  000403                          BR      38$
 9717  035100  012720  000116          39$:    MOV     #116,(R0)+      ;ELSE RESTORE TRAP CATCHER
 9718  035104  005010                          CLR     (R0)
 9719  035106  012737  034434  000004  38$:    MOV     #NEXINT,ERRVEC  ;SET UP NEM VECTOR FOR LINE CLOCK TEST
 9720  035114  005037  001662                  CLR     INTSET          ;CLEAR INTERRUPT COUNTER
 9721  035120  013700  001712                  MOV     KWLVEC,R0       ;SET POINTER TO VECTOR
 9722  035124  012720  034506                  MOV     #LCKHLR,(R0)+   ;INSERT ADDRESS OF INTERRUPT HDLR
 9723  035130  012710  000340                  MOV     #PR7,(R0)       ;INSERT PRIORITY
 9724  035134  012777  000100  144546          MOV     #BIT6,@KWLADD   ;LOAD KW11-L FOR INTERRUPT ENABLE
 9725  035142  000240                          NOP
 9726  035144  000240                          NOP
```

```
 9727  035146  000240                          NOP
 9728  035150  005737  001662                  TST     INTSET          ;TEST IN NEM ON KW11-P REFERENCE
 9729  035154  001003                          BNE     4$              ;THIS BRANCH WILL BYPASS SET UP OF
 9730                                                                  ;CLOCK OPTION
 9731  035156  052737  100000  001664          BIS     #LCLKPR,OPTFLG  ;SET CLOCK PRESENT FLAG
 9732  035164  012701  000006          4$:     MOV     #6,R1           ;RESTORE OLD VECTOR
 9733  035170  005037  001662                  CLR     INTSET          ;CLEAR INT FLAG
 9734  035174  012637  000006                  MOV     (SP)+,ERRVEC+2
 9735  035200  012637  000004                  MOV     (SP)+,ERRVEC
 9736  035204  012746  000000                  MOV     #PR0,-(SP)      ;RESTORE PRIORITY TO 0
 9737  035210  012746  035216                  MOV     #12$,-(SP)
 9738  035214  000002                          RTI
 9739  035216                          12$:
 9740  035216  104414                          RESREG
 9741  035220  000207                          RTS     PC
 9742
 9743                                  ;;*****************************************************************
 9744                                  .SBTTL  LOOP ON INTERNAL ERROR
 9745                                  ;*      THIS ROUTINE IS USED TO PROVIDE TIGHT SCOPE LOOPS. THE CALLER
 9746                                  ;*      IS EXPECTED TO SET $LPERR TO THE START OF THE SCOPE LOOP
 9747                                  ;*      TO BE EXECUTED ON ERROR.
 9748
 9749  035222  032777  001000  143710  SCOP1$: BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
 9750  035230  001405                          BEQ     5$              ;NO, CONTINUE
 9751  035232  105737  001103                  TSTB    $ERFLG          ;CHECK IF ERROR OCCURRED
 9752  035236  001402                          BEQ     5$
 9753  035240  013716  001110                  MOV     $LPERR,(SP)     ;LOAD ERROR RETURN
 9754  035244  000002                  5$:     RTI                     ;RETURN
 9755                                  .SBTTL  LINE CLOCK CALIBRATE
 9756                                  ;*      WAITS FOR A LINE CLOCK INTERRUPT TO CAL'BRATE THE INTERRUPTS
 9757                                  ;*      TO A  MEANINGFUL TIME VALUE. IN ADDITION IT PRESETS
 9758                                  ;*      THE TIMCNT IF THERE IS NO LINE CLOCK. TIMCNT IS USED IN THE
 9759                                  ;*      LINE CLOCK SIMULATOR.
 9760
 9761  035246  005037  001674          CLKCAL: CLR     LCLKTK          ;CLEAR TICK COUNTER
 9762  035252  032737  100000  001664          BIT     #LCLKPR,OPTFLG  ;TEST IF CLOCK PRESENT
 9763  035260  001004                          BNE     1$              ;YES - SKIP
 9764  035262  012737  000020  001654          MOV     #16.,TIMCNT     ;ELSE PRESET TIMCNT
 9765  035270  000410                          BR      2$              ;AND EXIT
 9766  035272  005737  001662          1$:     TST     INTSET          ;TEST IF INTERRUPT HAS OCCURRED
 9767  035276  001005                          BNE     2$              ;YES- ABORT CALIBRATION
 9768  035300  005737  001674                  TST     LCLKTK          ;WAIT FOR CLOCK TICK
 9769  035304  001772                          BEQ     1$              ;NOT YET - LOOP
 9770  035306  005037  001674                  CLR     LCLKTK          ;CLEAR TICK COUNT
 9771  035312  000207                  2$:     RTS     PC      ;RETURN
 9772
 9773                                  .SBTTL  LINE CLOCK SIMULATION ROUTINE
 9774                                  ;*      THIS ROUTINE IS USED TO SIMULATE THE LINE CLOCK. TO
 9775                                  ;*      DO THIS THE VALUE STORED IN MILCNT IS USED AS THE
 9776                                  ;*      BASE AND REPRESENTS THE NUMBER OF TIMES A DECREMENT
 9777                                  ;*      AND BRANCH LOOP CAN BE DONE IN 1 MILLISECOND. THE
 9778                                  ;*      TIMCNT VALUE IS DECREMENTED AND IF IT REACHED 0 THE
 9779                                  ;*      LINE CLOCK TICK COUNTER IS BUMPED. THEN THE TIMCNT
 9780                                  ;*      IS RESET TO 16 (REPRESENTS 16 MILLISECONDS PER LINE CLOCK
 9781                                  ;*      TICK).
 9782                                  ;*
```

```
                                        ;*      THUS THE ROUTINE RETURNS TO THE CALLER AFTER 1 MILLISECOND
                                        ;*      AND BUMPS THE LINE CLOCK TICK COUNTER FOR EACH 16 CALLS.
                                        ;*
035314  010046                 MYTIME: MOV     R0,-(SP)        ;SAVE R0
035316  013700  001652                 MOV     MILCNT,R0       ;SET COUNT
035322  005737  001662         1$:     TST     INTSET          ;TEST IF INTERRUPT SET
035326  001012                         BNE     2$              ;YES - SKIP
035330  005300                         DEC     R0              ;DECREMENT COUNT TO ZERO
035332  001373                         BNE     1$
035334  005337  001654                 DEC     TIMCNT          ;DECREMENT TIMCNT
035340  001005                         BNE     2$              ;IF NOT ZERO - EXIT
035342  005237  001674                 INC     LCLKTK          ;ELSE BUMP TICK COUNTER
035346  012737  000020  001654         MOV     #16.,TIMCNT     ;RESET TIME COUNT
035354  012600                 2$:     MOV     (SP)+,R0        ;RESTORE R0
035356  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.

035360  104413                 IWAT8S: SAVREG          ;ENTRY FOR 8 SECOND WAIT
035362  012700  000764                 MOV     #500.,R0
035366  000463                         BR      WATSRT
035370  104413                 IWAT1M: SAVREG          ;ENTRY FOR 1 MIN WAIT
035372  012700  007246                 MOV     #3750.,R0
035376  000457                         BR      WATSRT
035400  104413                 IWAT2S: SAVREG  ;ENTRY FOR 2 SECOND WAIT
035402  012700  000200                 MOV     #128.,R0
035406  000453                         BR      WATSRT
035410  104413                 IWAT1S: SAVREG                  ;ENTRY FOR 1 SECOND WAIT
035412  012700  000077                 MOV     #63.,R0
035416  000447                         BR      WATSRT
035420  104413                 IWAT159:        SAVREG          ;ENTRY FOR 160 MS WAIT
035422  012700  000012                 MOV     #10.,R0
035426  000443                         BR      WATSRT
035430  104413                 IWAT144:        SAVREG          ;ENTRY FOR 144 MS WAIT
035432  012700  000011                 MOV     #9.,R0
035436  000437                         BR      WATSRT
035440  104413                 IWAT128:        SAVREG          ;ENTRY FOR 128 MS WAIT
035442  012700  000010                 MOV     #8.,R0
035446  000433                         BR      WATSRT
035450  104413                 IWAT112:        SAVREG          ;ENTRY FOR 112 MS WAIT
035452  012700  000007                 MOV     #7,R0
035456  000427                         BR      WATSRT
035460  104413                 IWAT96: SAVREG                  ;ENTRY FOR 96 MS WAIT
035462  012700  000006                 MOV     #6,R0
035466  000423                         BR      WATSRT
035470  104413                 IWAT80: SAVREG                  ;ENTRY FOR 80 MS WAIT
035472  012700  000005                 MOV     #5,R0
```

```
 9839  035476  000417                          BR      WATSRT
 9840  035500  104413                  IWAT64: SAVREG                  ;ENTRY FOR 64 MS WAIT
 9841  035502  012700  000004                  MOV     #4,R0
 9842  035506  000413                          BR      WATSRT
 9843  035510  104413                  IWAT48: SAVREG                  ;ENTRY FOR 48 MS WAIT
 9844  035512  012700  000003                  MOV     #3,R0
 9845  035516  000407                          BR      WATSRT
 9846  035520  104413                  IWAT32: SAVREG                  ;ENTRY FOR 32 MS WAIT
 9847  035522  012700  000002                  MOV     #2,R0
 9848  035526  000403                          BR      WATSRT
 9849  035530  104413                  IWAT16: SAVREG                  ;ENTRY FOR 16 MS WAIT
 9850  035532  012700  000001                  MOV     #1,R0
 9851  035536  012746  000000          WATSRT: MOV     #PR0,-(SP)      ;LOAD PRIORITY 0 ON STACK
 9852  035542  012746  035550                  MOV     #5$,-(SP)       ;LOAD ADDRESS
 9853  035546  000002                          RTI
 9854
 9855  035550  012737  000020  001654  5$:     MOV     #16.,TIMCNT     ;PRESET TIME COUNTER
 9856  035556  004737  035246                  JSR     PC,CLKCAL       ;GO CALIBRATE THE CLOCK
 9857  035562  005737  001662          1$:     TST     INTSET          ;TEST IF INTERRUPT OCCURRED
 9858  035566  001036                          BNE     3$              ;YES - EXIT
 9859  035570  032737  100000  001664          BIT     #LCLKPR,OPTFLG  ;TEST IF KW11-L AVAILABLE
 9860  035576  001002                          BNE     2$              ;YES - SKIP
 9861  035600  004737  035314                  JSR     PC,MYTIME       ;ELSE CALL SIMULATOR
 9862  035604  023700  001674          2$:     CMP     LCLKTK,R0       ;TEST IF ENOUGH TICKS COUNTED
 9863  035610  103764                          BLO     1$              ;NO - LOOP
 9864  035612  104420                          TGETRK                  ;ELSE GET '611 REGS
 9865  035614  013701  001540                  MOV     T.CS1,R1        ;PUT CS1 IN R1- STRIP ALL BUT
 9866  035620  042701  177741                  BIC     #177741,R1      ;COMMAND CODE; INDEX INTO TABLE
 9867  035624  016137  050120  001372          MOV     CMNDLB(R1),DH2N ;AND SELECT HEADER TO IDENTIFY OPERATION
 9868  035632  012737  053526  001370          MOV     #EM4,EM2N       ;MESSAGE (NO INTERRUPT OR INTERRUPT LATE)
 9869  035640  013700  001302                  MOV     $TESTN,R0       ;GET NUMBER OF PRESENT TEST
 9870  035644  006300                          ASL     R0              ;SHIFT FOR INDEX
 9871  035646  016037  033122  001264          MOV     $SW08TB(R0),$ESCAPE  ;LOAD ESCAPE TO ABORT TESTS
 9872  035654  162737  000002  001264          SUB     #2,$ESCAPE      ;BUT GO TO NEXT SCOPE CALL
 9873  035662  000402                          BR      4$
 9874  035664  062716  000002          3$:     ADD     #2,(SP)         ;BUMP RETURN AROUND ERROR
 9875  035670  104414                  4$:     RESREG                  ;RESTORE REGS
 9876  035672  000002                          RTI                     ;RETURN
 9877
 9878                                  .SBTTL  "L" REGISTER LOADING ROUTINE
 9879                                  ;*      THE PARAMETERS FOLLOWING THE CALL ARE TRANSFERRED INTO
 9880                                  ;*      THE "L" REGISTERS L.CS1-L.DCYL. L.MR1 IS NOT
 9881                                  ;*      LOADED IN THIS MANNER SINCE IT IS NOT COMMONLY LOADED
 9882                                  ;*      FOR AN OPERATION. L.CS2 IS LOADED FROM DRVNUM.
 9883                                  ;*
 9884                                  ;*      CALL:   JSR     R4,LRLOAD
 9885                                  ;*              COMMAND
 9886                                  ;*              WORD COUNT
 9887                                  ;*              BUS ADDRESS
 9888                                  ;*              .BYTE           SECTOR ADDRESS
 9889                                  ;*              .BYTE           TRACK ADDRESS
 9890                                  ;*              CYLINDER ADDRESS
 9891
 9892  035674                          LRLOAD:
 9893  035674  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9894  035676  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
```

```
 9895  035700  012701  001600                MOV     #L.CS1,R1       ;GET ADDRESS OF L REGS
 9896  035704  012700  000004                MOV     #4,R0           ;PRESET COUNT
 9897  035710  012421                1$:     MOV     (R4)+,(R1)+     ;MOVE FIRST FOUR WORDS INTO "L" REGS
 9898  035712  005300                        DEC     R0              ;CS1, WC, BA, DA
 9899  035714  001375                        BNE     1$
 9900  035716  013721  001626                MOV     DRVNUM,(R1)+    ;LOAD DRIVE NUMBER
 9901  035722  005721                        TST     (R1)+           ;BUMP PAST ASOF
 9902  035724  012411                        MOV     (R4)+,(R1)      ;LOAD DCYL
 9903  035726  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 9904  035730  012600                        MOV     (SP)+,R0        ;,POP STACK INTO R0
 9905  035732  000204                        RTS     R4
 9906
 9907                            .SBTTL  LOAD RK611 FOR OPERATION
 9908                            ;*      THE REGISTER SETUP STORAGE IS TRANSFERRED TO THE RK611 REGISTER.
 9909                            ;*      THIS IS A STRAIGHT TRANSFER WITH NO CHECKING OR MANIPULATION
 9910                            ;*      OF THE REGISTER CONTENTS.  L.CS1 IS TRANSFERRED LAST AS IT
 9911                            ;*      SHOULD BE IF THE GO BIT IS SET.
 9912  035734  005037  001662    LOADRK: CLR     INTSET          ;CLEAR INTERRUPT FLAG.
 9913  035740  010046                    MOV     R0,-(SP)        ;STORE REGISTER
 9914  035742  010146                    MOV     R1,-(SP)
 9915  035744  012700  001602            MOV     #L.WC,R0        ;GET ADDRESS OF SETUP STORAGE WC
 9916  035750  010201                    MOV     R2,R1           ;GET BASE ADDRESS
 9917  035752  062701  000002            ADD     #2,R1           ;PUT R1 PAST RKCS1
 9918  035756  012021                    MOV     (R0)+,(R1)+     ;LOAD WORD COUNT
 9919  035760  012021                    MOV     (R0)+,(R1)+     ;LOAD BUS ADDRESS
 9920  035762  012021                    MOV     (R0)+,(R1)+     ;LOAD DISK ADDRESS
 9921  035764  012011                    MOV     (R0)+,(R1)      ;LOAD CS2
 9922  035766  062701  000006            ADD     #6,R1           ;BUMP R1 TO ASOF
 9923  035772  012021                    MOV     (R0)+,(R1)+     ;LOAD OFFSET
 9924  035774  012021                    MOV     (R0)+,(R1)+     ;LOAD CYLINDER
 9925  035776  062701  000004            ADD     #4,R1           ;BUMP R1 TO MR1
 9926  036002  011011                    MOV     (R0),(R1)       ;LOAD MR1
 9927  036004  123727  001102  000007    CMPB    $TSTNM,#7       ;SEE IF TEST 7
 9928  036012  001403                    BEQ     1$              ;BR IF YES
 9929  036014  053737  001720  001600    BIS     DTYPE,L.CS1     ;ELSE SET CORRECT DRIVE TYPE
 9930  036022  013712  001600    1$:     MOV     L.CS1,(R2)      ;LOAD CS1
 9931  036026  012601                    MOV     (SP)+,R1        ;RESTORE REGISTER
 9932  036030  012600                    MOV     (SP)+,R0
 9933  036032  000002                    RTI                     ;RETURN
 9934
 9935                            ;
 9936                            ;THIS ROUTINE LOADS CONSTANTS EITHER FOR THR RK06 OR THE RK07
 9937                            ;
 9938  036034  005737  001720    LDCON:  TST     DTYPE           ;SEE IF RK06
 9939  036040  001414                    BEQ     1$              ;BR IF YES
 9940  036042  012737  001456  036120    MOV     #1456,LSTCYL    ;ELSE LOAD RK07 CONSTANTS
 9941  036050  012737  001000  036122    MOV     #1000,TSTCYL    ;MAY NOT BE NEEDED
 9942  036056  005037  036130            CLR     HOLD2
 9943  036062  012737  000040  036132    MOV     #DTYE,HOLD3
 9944  036070  000207                    RTS     PC
 9945
 9946  036072  012737  000632  036120    MOV     #632,LSTCYL     ;RK06 CONSTANTS
 9947  036100  005037  036122    1$:     CLR     TSTCYL          ;MAY NOT BE NEEDED
 9948  036104  012737  002000  036130    MOV     #CDT,HOLD2
 9949  036112  005037  036132            CLR     HOLD3
 9950  036116  000207                    RTS     PC
```

```
                                LSTCYL: 0
                                TSTCYL: 0                       ;FOR T20,57,60 MAY NOT BE NEEDED
                                HOLD:   0                       ;FOR T73,105
                                HOLD1:  0                       ;FOR T105
                                HOLD2:  0                       ;FOR T7
                                HOLD3:  0                       ;FOR T12

                                ;
                                ;THESE ROUTINES LOAD THE CORRECT DRIVE TYPE INTO DTYPE
                                ;
036134  006300                  CTYP0:  ASL     R0
036136  016037  001722  001720          MOV     TYPTBL(R0),DTYPE
036144  006200                          ASR     R0
036146  000207                          RTS     PC

036150  006301                  GTYP1:  ASL     R1
036152  016137  001722  001720          MOV     TYPTBL(R1),DTYPE
036160  006201                          ASR     R1
036162  000207                          RTS     PC


                                .SBTTL  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.

036164  010046                  GETRK:  MOV     R0,-(SP)        ;STORE REGISTERS TO BE USED
036166  010146                          MOV     R1,-(SP)
036170  010346                          MOV     R3,-(SP)
036172  012700  001540                  MOV     #T.CS1,R0       ;SET POINTER TO TEST TABLE
036176  010201                          MOV     R2,R1           ;SET POINTER TO RK611 BASE
036200  012703  000012                  MOV     #10.,R3         ;SET COUNT FOR 1ST 10 REGS
036204  012120                  1$:     MOV     (R1)+,(R0)+     ;STORE RKCS1 THROUGH RKSPAR
036206  005303                          DEC     R3
036210  001375                          BNE     1$
036212  062701  000002                  ADD     #2,R1           ;BUMP POINTER PAST RKDB
036216  005720                          TST     (R0)+           ;BUMP POINTER PAST T.RKDB
036220  012703  000004                  MOV     #4,R3           ;SET COUNT FOR LAST 5 REGS
036224  012120                  2$:     MOV     (R1)+,(R0)+     ;STORE RKMR1 THROUTH RKMR3
036226  005303                          DEC     R3
036230  001375                          BNE     2$
036232  012603                          MOV     (SP)+,R3        ;RESTORE REGISTERS
036234  012601                          MOV     (SP)+,R1
036236  012600                          MOV     (SP)+,R0
036240  000002                          RTI                     ;RETURN
                                ;SBTTL  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.

036242  016637  000002  001256  BITCNT: MOV     2(SP),$TMP16    ;GET WORD WHOSE BITS ARE TO BE COUNTED
036250  010346                          MOV     R3,-(SP)        ;STORE R3
036252  005037  001260                  CLR     $TMP17          ;CLEAR $TMP16 FOR COUNTING
036256  012703  000021                  MOV     #17.,R3         ;SET A SHIFT COUNTER
036262  000241                          CLC                     ;CLEAR CARRY
```

```
10007  036264  006037  001256          1$:     ROR     $TMP16          ;ROTATE WORD.
10008  036270  103407                  BCS     3$              ;WAS BIT SHIFTED OUT A 1?
10009  036272  005303                  2$:     DEC     R3              ;NO - DEC COUNT
10010  036274  001373                  BNE     1$              ;LOOP IF NOT ZERO
10011  036276  012603                  MOV     (SP)+,R3        ;RESTORE R3
10012  036300  013766  001260  000002          MOV     $TMP17,2(SP)    ;PUT COUNT OF BITS ON STACK
10013  036306  000204                  RTS     R4              ;RETURN
10014  036310  005237  001260          3$:     INC     $TMP17          ;BUMP COUNT
10015  036314  000766                  BR      2$              ;LOOP
10016
10017                                  .SBTTL  MAINTENANCE CLOCK ROUTINE
10018                                  ;*      THE PARAMETERS PASSED TO THIS ROUTINE ARE LOCATED IN THE
10019                                  ;*      ADDRESS OFTER THE CALL. THE FIRST BYTE CANTAINS THE NUMBER
10020                                  ;*      OF PHASES ADDRESSES THE CALLING ROUTINE WANTS THE CONTROLLER
10021                                  ;*      CLOCKED THROUGH AND THE SECOND BYTE CONTAINS THE NUMBER OF
10022                                  ;*      CLOCK TRANSITIONS(PARTIAL PHASES) TO BE DONE.
10023
10024  036316                          MCLOCK:
10025  036316  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10026  036320  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10027  036322  112400                  MOVB    (R4)+,R0        ;GET NUMBER OF CONTROLLER PHASE ADDRESSES
10028  036324  112401                  MOVB    (R4)+,R1        ;GET PARTIAL PHASE ADDRESS COUNT
10029
10030  036326  006300                  ASL     R0              ;MULTIPLY PHASE ADDRESS  COUNT BY 4
10031  036330  006300                  ASL     R0
10032  036332  060100                  ADD     R1,R0           ;ADD IN PARTIALS
10033  036334  052762  000400  000026  1$:     BIS     #MCLK,RKMR1(R2) ;SET CLOCK
10034  036342  042762  000400  000026          BIC     #MCLK,RKMR1(R2) ;CLEAR MCLK
10035  036350  005300                  DEC     R0              ;DECREMENT COUNT
10036  036352  001370                  BNE     1$              ;LOOP IF NOT ZERO
10037  036354  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
10038  036356  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
10039  036360  000204                  RTS     R4
10040                                  .SBTTL  READ AND SORT HEADERS
10041                                  ;*      THE HEADERS IN THE CYLINDER AND TRACK SPECIFIED BY
10042                                  ;*      THE FIELDS IN THE "L" REGISTERS ARE READ AND STORED IN
10043                                  ;*      ASSCENDING ORDER. CONTROLLER ERRORS ARE CHECKED IN THE
10044                                  ;*      READ HEADER OPERATION AND DATA LATE IS CHECKED AFTER
10045                                  ;*      EACH READ OF THE DATA BUFFER.
10046                                  ;*
10047                                  ;*      CALL:   JSR     R4,RDSTHD
10048                                  ;*              TCHKOP                  ;RETURN POINT IF CERR IN READ HDR
10049                                  ;*              ERROR   4  ;OR 5, 6, 7
10050                                  ;*              ERROR   13              ;RETURN IF DATA LATE IN DB UNLOAD
10051                                  ;*              ERROR   2               ;RETURN IF TO SLOW OR
10052                                  ;*                                      ;IF HDR 0 NOT FOUND
10053
10054  036362  104413                  RDSTHD: SAVREG
10055  036364  032737  100000  001664          BIT     #LCLKPR,OPTFLG  ;TEST IF CLOCK PRESENT
10056  036372  001402                  BEQ     20$             ;NO - SKIP
10057  036374  005077  143310          CLR     @KWLADD ;RESET INTERRUPT
10058  036400  012700  000026          20$:    MOV     #26,R0          ;PRESET FOR 26 SECTOR FORMAT
10059  036404  032737  010000  001600          BIT     #CFMT,L.CS1     ;IS 24 SECTOR MODE SET?
10060  036412  001402                  BEQ     1$              ;NO - SKIP
10061  036414  012700  000024          MOV     #24,R0          ;ELSE CHANGE TO 24 SECTOR MODE
10062  036420  012701  070414          1$:     MOV     #IBUFF,R1       ;SET POINTER TO INPUT BUFFER
```

```
10063  036424  010005                        MOV     R0,R5           ;SAVE NUMBER OF SECTORS
10064  036426  010104                        MOV     R1,R4           ;SAVE IBUFF ADDRESS
10065  036430  010203                        MOV     R2,R3           ;SET UP POINTER TO RKDB
10066  036432  062703  000024                ADD     #RKDB,R3
10067  036436  013762  001626  000010        MOV     DRVNUM,RKCS2(R2) ;LOAD DRIVE NUM
10068  036444  013762  001614  000020        MOV     L.DCYL,RKDCYL(R2) ;LOAD CYLINDER NUM
10069  036452  013762  001606  000006        MOV     L.DA,RKDA(R2)   ;LOAD TRACK AND SECTOR
10070
10071  036460  012737  000020  001654  2$:   MOV     #16.,TIMCNT     ;SET TIME COUNTER
10072  036466  005037  001662                CLR     INTSET          ;CLEAR INTERRUPT FLAG
10073  036472  005037  001674                CLR     LCLKTK          ;CLEAR TICK COUNTER
10074  036476  053737  001720  001600        BIS     DTYPE,L.CS1     ;SET DRV TYP
10075  036504  013762  001600  000000        MOV     L.CS1,RKCS1(R2) ;LOAD COMMAND
10076
10077  036512  005737  001662          3$:   TST     INTSET          ;TEST IF INT OCCURRED
10078  036516  001020                        BNE     4$              ;YES - SKIP
10079  036520  004737  035314                JSR     PC,MYTIME       ;WAIT 1 MS
10080  036524  005737  001674                TST     LCLKTK          ;HAVE WE WAITED 16 MS?
10081  036530  001770                        BEQ     3$              ;NO - LOOP ON WAIT
10082
10083  036532  062766  000006  000006        ADD     #6,6(SP)        ;SET RETURN FOR TO SLOW
10084  036540  104420                        TGETRK                  ;GET RK REGS
10085  036542  012737  053526  001370        MOV     #EM4,EM2N       ;LOAD MESSAGE ''TO SLOW/NOT COMPLETE''
10086  036550  012737  050324  001372        MOV     #OPER24,DH2N    ;LOAD COMMAND ''READ HEADER'' FOR REPORT
10087  036556  000466                        BR      10$             ;SKIP
10088
10089  036560  005762  000000          4$:   TST     RKCS1(R2)       ;TEST FOR CONTROLLER ERROR
10090  036564  100474                        BMI     11$             ;YES - SKIP
10091
10092  036566  011324                        MOV     (R3),(R4)+      ;STORE HEADERS
10093  036570  011324                        MOV     (R3),(R4)+
10094  036572  011324                        MOV     (R3),(R4)+
10095
10096  036574  005762  000010                TST     RKCS2(R2)       ;TEST IF DATA LATE
10097  036600  100443                        BMI     8$              ;YES - SKIP
10098
10099  036602  005300                        DEC     R0              ;DEC SECTOR COUNT
10100  036604  001325                        BNE     2$              ;IF NOT ZERO - LOOP
10101
10102  036606  032737  100000  001664        BIT     #LCLKPR,OPTFLG  ;TEST IF CLOCK PRESENT
10103  036614  001403                        BEQ     5$              ;NO - SKIP
10104  036616  012777  000100  143064        MOV     #BIT6,@KWLADD   ;SET INTERRUPT ENABLE
10105  036624  032761  000037  000002  5$:   BIT     #37,2(R1)       ;HEADER AT TOP OF BUFF=HEAD 0?
10106  036632  001413                        BEQ     6$              ;YES - SKIP
10107  036634  012124                        MOV     (R1)+,(R4)+     ;ELSE MOV THIS HEADER TO BOTTOM
10108  036636  012124                        MOV     (R1)+,(R4)+
10109  036640  012124                        MOV     (R1)+,(R4)+
10110
10111  036642  005305                        DEC     R5              ;TEST FO INSURE HEAD 0 IS FOUND
10112  036644  001367                        BNE     5$              ;IF ALL HEADERS NOT CHECKED - LOOP
10113  036646  012737  056276  001370        MOV     #EM56,EM2N      ;ELSE ''HEADER 0 NOT FOUND'' MESSAGE
10114  036654  005037  001372                CLR     DH2N
10115  036660  000421                        BR      9$              ;SKIP
10116
10117  036662  012700  070414          6$:   MOV     #IBUFF,R0       ;GET TOP OF IBUFF
10118  036666  012120                  7$:   MOV     (R1)+,(R0)+     ;MOVE HEADERS SO THEY START AT TOP OF IBUFF
```

```
10119  036670  012120                          MOV     (R1)+,(R0)+
10120  036672  012120                          MOV     (R1)+,(R0)+
10121  036674  020004                          CMP     R0,R4           ;ALL HEADERS MOVED?
10122  036676  001373                          BNE     7$              ;NO - LOOP
10123
10124  036700  062766  000010  000006        ADD     #10,6(SP)       ;SET UP FOR GOOD RETURN
10125  036706  000423                          BR      11$
10126
10127  036710  012737  054541  001500  8$:     MOV     #EMDLT,EM13N    ;"DATA LATE SET RESULT OF READ DB"
10128  036716  012737  056707  061022          MOV     #EMRDB,DF011A
10129  036724  062766  000004  000006  9$:     ADD     #4,6(SP)        ;SET ERROR RETURN
10130  036732  104420                          TGETRK                  ;GET RK REGS
10131  036734  013700  001302          10$:    MOV     $TESTN,R0       ;GET TEST NUMBER
10132  036740  006300                          ASL     R0              ;SHIFT FOR INDEX
10133  036742  016037  033122  001264          MOV     $SW08TB(R0),$ESCAPE  ;SET ESCAPE
10134  036750  162737  000002  001264          SUB     #2,$ESCAPE      ;TO NEXT SCOPE CALL
10135
10136  036756  104414                  11$:    RESREG
10137  036760  000204                          RTS     R4
10138
10139                                  .SBTTL  GET DRIVE STATUS
10140                                  ;*      THIS ROUTINE GETS ALL THE DRIVE STATUS AND PLACES IT IN $REG10
10141                                  ;*      THROUGH $REG17.   THESE REGISTORS ARE FIRST CLEARED TO ALL ONES AND
10142                                  ;*      THEN IF ERROR OCCURS WHILE GETTING STATUS, THE 1'S ARE LEFT
10143                                  ;*      IN THE REGISTERS.
10144                                  ;*
10145                                  ;*CALL: JSR     R4,GETDRS
10146                                  ;*      BR      ERROR PROCESSING        ERROR RETURN
10147                                  ;*      BR      NO ERPOR PROCESSING     GOOD RETURN
10148
10149  036762  104413          GETDRS: SAVREG
10150  036764  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR ANY OLD ERRORS LAYING AROUND
10151  036772  012700  001202                  MOV     #$REG10,R0      ;PRESET ALL STATUS STORAGE TO
10152  036776  012701  000010                  MOV     #8.,R1          ;ALL ONES
10153  037002  012720  177777          1$:     MOV     #177777,(R0)+
10154  037006  005301                          DEC     R1
10155  037010  001374                          BNE     1$
10156  037012  012700  001206                  MOV     #$REG12,R0      ;SET POINTER TOREG12 FOR A01 & B01
10157  037016  012701  000001                  MOV     #1,R1           ;PRESET FOR PAIR ONE.
10158  037022  005037  001230                  CLR     $TMP3           ;CLEAR ERROR SWITCH
10159  037026  013762  001610  000010  2$:     MOV     L.CS2,RKCS2(R2) ;LOAD DRIVE #
10160  037034  010162  000026                  MOV     R1,RKMR1(R2)    ;LOAD MR1
10161  037040  012762  000001  000000          MOV     #BIT0,RKCS1(R2) ;DO SELECT
10162  037046  012703  000050                  MOV     #40.,R3         ;WAIT FOR A FEW MICRO RECORDS TO
10163  037052  005303                  3$:     DEC     R3              ;BIT SELECT FINISH
10164  037054  001376                          BNE     3$
10165  037056  032762  100000  000000          BIT     #CERR,RKCS1(R2) ;ANY ERROR SET AS A RESULT OF SELECT?
10166  037064  001415                          BEQ     4$              ;NO - SKIP
10167  037066  032762  024000  000000          BIT     #CTO!SPAR,RKCS1(R2) ;CHECK IF TIMEOUT OR PARITY ERROR
10168  037074  001004                          BNE     8$              ;YES - SKIP
10169  037076  032762  037400  000010          BIT     #37400,RKCS2(R2) ;TEST FOR ERRORS:
10170                                                                  ;   NED!UPE!MDS!UFE!NEM!PGE
10171  037104  001405                          BEQ     4$              ;NO - SKIP
10172  037106  012737  000001  001230  8$:     MOV     #1,$TMP3        ;SET ERROR FLAG
10173  037114  022020                          CMP     (R0)+,(R0)+     ;BUMP TO LET THAT PAIR STAY ALL 1'S.
10174  037116  000404                          BR      5$              ;SKIP
```

```
10175  037120  016220  000034          4$:     MOV     RKMR2(R2),(R0)+ ;STORE A WORD
10176  037124  016220  000036                  MOV     RKMR3(R2),(R0)+ ;STORE B WORD
10177  037130  012762  100000  000000  5$:     MOV     #CCLR,RKCS1(R2) ;CLEAR ANY OLD ERROR IN CONTROLLER
10178  037136  005701                          TST     R1              ;IS R1 A 0 (LAST TRANSFER, PAIR 0)
10179  037140  001410                          BEQ     6$              ;YES - SKIP
10180  037142  005201                          INC     R1              ;ELSE BUMP TO NEXT PAIR
10181  037144  022701  000004                  CMP     #4,R1           ;PAIR 3  JUST STORED?
10182  037150  001326                          BNE     2$              ;NO - SKIP
10183  037152  005001                          CLR     R1              ;ELSE SET TO PAIR 0
10184  037154  012700  001202                  MOV     #$REG10,R0      ;PRESET POINTER FOR PAIR 0
10185  037160  000722                          BR      2$              ;GO GET THEM
10186  037162  104414                  6$:     RESREG                  ;EXIT HERE
10187  037164  005737  001230                  TST     $TMP3           ;ANY ERROR IN STATUS GETTING
10188  037170  001001                          BNE     7$              ;YES - SKIP
10189  037172  005724                          TST     (R4)+           ;ELSE BUMP PART ERROR
10190  037174  000204                  7$:     RTS     R4              ;RETURN
10191
10192                                  .SBTTL  SUBSYSTEM INITIALIZE AND INITIALIZE STATE TEST
10193                                  ;*      THE SUBSYSTEM IS INITIALIZED WITH A SUBSYSTEM CLEAR
10194                                  ;*      COMMAND.  CERR AND DI ARE MONITORED FOR A SHORT
10195                                  ;*      PERIOD OF TIME DURING WHICH THEY SHOULD BOTH RESET.
10196                                  ;*
10197                                  ;*      IF THEY DO RESET, READY IS TESTED TO INSURE IF SETS.
10198                                  ;*
10199                                  ;*      IF ANY OF THESE THREE CONDITIONS ARE NOT MET AN APPROPRIATE
10200                                  ;*      ERROR MESSAGE IS PREPARED AND REPORTED WHEN THE ROUTINE
10201                                  ;*      RETURN TO THE CALL. IF EVERY THING IS GOOD, THE RETURN
10202                                  ;*      SKIPS OVER THE ERROR CALL AND TEST ABORT.
10203                                  ;*
10204                                  ;*      THE USUAL CALL TO THIS ROUTINE WILL BE FOLLOWED BY
10205                                  ;*      AN ERROR MESSAGE AND BRANCH TO END OF TEST.  THIS
10206                                  ;*      IS DONE BECAUSE FAILURE TO INITIALIZE CORRECTLY IS FATAL TO
10207                                  ;*      THE TEST.
10208
10209  037176                          SSINIT:
10210  037176  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10211  037200  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10212  037202  012701  000007                  MOV     #7,R1           ;SET CLEAR COUNT
10213  037206  012700  001600                  MOV     #L.CS1,R0       ;GET ADDRESS OF "L" REGS
10214  037212  012720  000100                  MOV     #100,(R0)+      ;PRESET CS1  INTR ENABLE
10215  037216  005020                  7$:     CLR     (R0)+           ;CLEAR THE NEXT
10216  037220  005301                          DEC     R1              ;COUNT 0?
10217  037222  001375                          BNE     7$              ;NO - LOOP
10218  037224  012762  000040  000010          MOV     #SCLR,RKCS2(R2) ;CLEAR SUBSYSTEM
10219  037232  012737  000012  001222          MOV     #10.,$TMP0      ;SET A COUNTER
10220  037240  016237  000000  001540  1$:     MOV     RKCS1(R2),T.CS1 ;GET CS1
10221  037246  032737  140000  001540          BIT     #CERR!DI,T.CS1  ;TEST IF ERROR OR DI SET
10222  037254  001433                          BEQ     2$              ;NO - SKIP TO READY TEST
10223  037256  005337  001222                  DEC     $TMP0           ;ELSE DECREMENT COUNTER
10224  037262  001366                          BNE     1$              ;AND LOOP
10225  037264  032737  100000  001540          BIT     #CERR,T.CS1     ;TEST - IS IT CERR STILL SET
10226  037272  001404                          BEQ     3$              ;NO  - SKIP TO DI MESSAGE
10227  037274  012737  053621  001400          MOV     #EM5,EM3N       ;MESSAGE (SUBSYS CLR NOT RESET ERROR)
10228  037302  000403                          BR      4$
10229  037304  012737  053665  001400  3$:     MOV     #EM6,EM3N       ;MESSAGE (SUBSYS CLEAR NOT RESET DI)
10230  037312  104420                  4$:     TGETRK
```

```
10231  037314  013700  001302                  MOV     $TESTN,R0       ;GET PRESENT TEST NUMBER
10232  037320  006300                          ASL     R0              ;SHIFT FOR INDEX
10233  037322  016037  033122  001264          MOV     $SW08TBL(R0),$ESCAPE ;LOAD ESCAPE TO ABORT TEST
10234  037330  162737  000002  001264          SUB     #2,$ESCAPE      ;SET TO NEXT SCOPE CALL
10235  037336  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10236  037340  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10237  037342  000414                          BR      6$              ;SKIP TO EXIT
10238  037344  032737  000200  001540  2$:     BIT     #RDY,T.CS1      ;TEST READY SET
10239  037352  001004                          BNE     5$              ;YES - GOOD EXIT
10240  037354  012737  053744  001400          MOV     #EM7,EM3N       ;MESSAGE (SUBSYS CLR NOT SET READY)
10241  037362  000753                          BR      4$
10242  037364  012601                  5$:     MOV     (SP)+,R1        ;RESTORE REGS
10243  037366  012600                          MOV     (SP)+,R0
10244  037370  062716  000002                  ADD     #2,(SP)         ;GOOD RETURN
10245  037374  000002                  6$:     RTI
10246
10247                                  .SBTTL  WORD COUNT AT END OF OPERATION CHECK
10248                                  ;*      THIS ROUTINE COMPARES THE CONTENTS OF THE TEST STORAGE FOR
10249                                  ;*      THE WORD COUNT AGAINST THE SUPPLIED VALUE.  IF UNEQUAL, THE
10250                                  ;*      ERROR FLAG (WCERR) IS SET IN GROUP 4 ERROR FLAGS (GRP4ER)
10251                                  ;*
10252                                  ;*CALL: JSR     R4,CHKWC
10253                                  ;*      .WORD                   ;EXPECTED WC VALUE
10254
10255  037376  012437  047744          CHKWC:  MOV     (R4)+,EXPWC     ;STORE EXPECTED VALUE
10256                                                                  ;
10257  037402  023737  047744  001542          CMP     EXPWC,T.WC      ;COMPARE
10258  037410  001406                          BEQ     1$              ;EQUAL - SKIP
10259  037412  052737  000001  050002          BIS     #WCERR,GRP4ER   ;SET ERROR FLAG
10260  037420  013737  001542  047760          MOV     T.WC,REALWC     ;STORE REAL WORD COUNT
10261  037426  000204                  1$:     RTS     R4              ;RETURN.
10262
10263                                  .SBTTL  BUS  ADDRESS AT END OF OPERATION CHECK
10264                                  ;*      THIS ROUTINE COMPUTES THE EXPECTED BUS ADDRESS AT THE END OF
10265                                  ;*      A TRANSFER BY USING THE INITIAL BUS ADDRESS, ADDING IN THE
10266                                  ;*      INITIAL WORD COUNT, AND SUBTRACTING ANY RESIDUAL WORD COUNT.
10267                                  ;*      IF THIS COMPUTED BA DOES NOT EQUAL THE CONTENTS OF RKBA
10268                                  ;*      AN ERROR FLAG (BAERR) IS SET IN GROUP 4 ERROR FIELD (GRP4ER)
10269                                  ;*
10270                                  ;*      IF BUS ADDRESS INCREMENT INHIBIT WAS SET, THE EXPECTED BUS
10271                                  ;*      ADDRESS IS THE STARTING BUS ADDRESS.
10272                                  ;*CALL: JSR     R4,CHKBA
10273
10274  037430  010046          CHKBA:  MOV     R0,-(SP)
10275  037432  010146                  MOV     R1,-(SP)
10276  037434  010346                  MOV     R3,-(SP)
10277  037436  032737  000020  001610          BIT     #BAI,L.CS2      ;TEST IF BAI SET
10278  037444  001404                          BEQ     4$              ;NO - SKIP
10279  037446  013737  001604  047750          MOV     L.BA,EXPBA      ;STORE EXPECTED BA (SAME AS STARTING BA)
10280  037454  000441                          BR      3$
10281  037456  013700  001602          4$:     MOV     L.WC,R0         ;GET INITIAL WORD COUNT
10282  037462  005400                          NEG     R0
10283  037464  113703  001601                  MOVB    L.CS1+1,R3      ;GET BA16 & BA17
10284  037470  042703  177774                  BIC     #177774,R3      ;CLEAR UNWANTED BITS
10285
10286  037474  005700                          TST     R0              ;TEST IF INITIAL WORD COUNT 0
```

```
10287  037476  001003                        BNE     6$              ;NO - SKIP
10288  037500  062703  000002                ADD     #2,R3           ;ADD 2 TO BA16,17 (65K WORD XFER)
10289  037504  000407                        BR      9$              ;SKIP
10290  037506  005700                6$:     TST     R0              ;TEST IF INITIAL WC BIT 15 SET
10291  037510  100001                        BPL     5$              ;NO - SKIP
10292  037512  005203                        INC     R3              ;BUMP BA16,17 (32K WORD XFER)
10293  037514  006300                5$:     ASL     R0              ;SHIFT WORD COUNT TO MAKE MEM ADD CNT
10294  037516  063700  001604                ADD     L.BA,R0         ;ADD IN START BUFFER ADD
10295  037522  005503                        ADC     R3              ;IF CARRY - ADD INTO BA16,17
10296  037524  013701  001542          9$:   MOV     T.WC,R1         ;GET END OF OPERATION WORD COUNT
10297  037530  001411                        BEQ     1$              ;BRANCH IF ZERO
10298  037532  005401                        NEG     R1
10299  037534  005701                        TST     R1              ;TEST END OPERATION WC BIT 15 SET
10300  037536  100001                        BPL     7$              ;NO - SKIP
10301  037540  005303                        DEC     R3              ;DEC BA 16,17 (32K WC LEFT)
10302  037542  006301                7$:     ASL     R1              ;SHIFT WC TO MAKE MEM ADD CNT
10303  037544  160100                        SUB     R1,R0           ;SUB FROM COMPUTED BUS ADDRESS
10304  037546  005603                        SBC     R3              ;SUB CARRY FROM BA16,17
10305  037550  010337  047746                MOV     R3,EXPUBA       ;STORE EXPECTED UPPER BA BITS
10306  037554  010037  047750          1$:   MOV     R0,EXPBA
10307  037560  020037  001544          3$:   CMP     R0,T.BA         ;EQUAL TO COMPUTED?
10308  037564  001406                        BEQ     2$              ;YES - SKIP
10309  037566  052737  000004  050002        BIS     #BAERR,GRP4ER   ;ELSE SET BAERR FLAG
10310  037574  013737  001544  047764        MOV     T.BA,REALBA     ;STORE REAL BUS ADDRESS
10311  037602  113703  001541          2$:   MOVB    T.CS1+1,R3      ;GET REAL UPPER BA
10312  037606  042703  177774                BIC     #177774,R3      ;CLEAR UNWANTED BITS
10313  037612  020337  047746                CMP     R3,EXPUBA       ;CHECK IF EQUAL
10314  037616  001405                        BEQ     8$              ;YES - SKIP
10315  037620  052737  000002  050002        BIS     #UBAERR,GRP4ER  ;ELSE SET UBA ERROR
10316  037626  010337  047762                MOV     R3,REALUB       ;STORE REAL UPPER BA
10317  037632  012603                8$:     MOV     (SP)+,R3
10318  037634  012601                        MOV     (SP)+,R1
10319  037636  012600                        MOV     (SP)+,R0
10320  037640  000204                        RTS     R4
10321
10322                                  .SBTTL  CYLINDER,TRACK,SECTOR TEST AT END OF OPERATION
10323                                  ;*      THIS ROUTINE CHECKS THAT THE CONTENTS OF THE RKDCYL AND RKDA
10324                                  ;*      ARE CORRECT FOR ANY SIZE DATA TRANSFER AT THE END OF THE
10325                                  ;*      OPERATION.  THE CONTENTS OF THE LOAD REGISTER STORAGE ARE
10326                                  ;*      COUNTED ON TO HAVE THE INITIAL VALUES TO MAKE THE
10327                                  ;*      NECESSARY CALCULATION.
10328                                  ;*
10329                                  ;*      ALL THREE VALUES ARE GENERATED AND STORED IN EXPECTED VALUES
10330                                  ;*      STORAGE EXPCYL, EXPTRK, EXPSEC.  ALL 3 ARE CHECKED AND
10331                                  ;*      IF ONE OR MORE ARE WRONG, THE CORRESPONDING BIT IN THE
10332                                  ;*      ERROR FLAGS FIELD (GRP4ER) IS SET.
10333                                  ;*
10334                                  ;*CALL: JSR     R4,CHKCTS
10335
10336  037642  104413                  CHKCTS: SAVREG
10337  037644  013700  001602                  MOV     L.WC,R0         ;GET SPECIFIED WORD COUNT
10338  037650  005400                          NEG     R0              ;NEGATE IT
10339  037652  013701  001542                  MOV     T.WC,R1         ;GET END OF OPERATION WORD COUNT
10340  037656  001401                          BEQ     10$             ;IF ZERO - SKIP
10341  037660  005401                          NEG     R1              ;NEGATE IT
10342  037662  160100                  10$:    SUB     R1,R0           ;COMPUTE ACTUAL WORDS TRANSFERRED
```

```
10343  037664  005001                          CLR     R1              ;CLEAR R1 FOR COUNTING
10344                                  ;       THE FOLLOWING CODE DETERMINES HOW MAY SECTORS OF DATA HAS BEEN
10345                                  ;       TRANSFERRED IN THE OPERATION. ONCE IT HAS COMPUTED THAT, THE
10346                                  ;       END OF OPERATION VALUES FOR THE CYLINDER, TRACK, AND SECTOR
10347                                  ;       IS CALCULATED.
10348  037666  022700  000400          1$:     CMP     #400,R0
10349  037672  003004                          BGT     2$
10350  037674  005201                          INC     R1
10351  037676  162700  000400                  SUB     #400,R0
10352  037702  000771                          BR      1$
10353  037704  005700                  2$:     TST     R0
10354  037706  001401                          BEQ     3$
10355  037710  005201                          INC     R1
10356                                  ;       AT THIS POINT R1 HAS A COUNT OF THE
10357                                  ;       NUMBER OF FULL SECTOR TRANSFER + 1 IF A
10358                                  ;       PARTIAL SECTOR WAS TRANSFERRED.
10359  037712  012703  000026          3$:     MOV     #26,R3
10360  037716  032737  010000  001600          BIT     #CFMT,L.CS1
10361  037724  001402                          BEQ     4$
10362  037726  012703  000024                  MOV     #24,R3
10363                                  ;       R3 HAS BEEN SET UP WTH THE NUMBER
10364                                  ;       OF SECTORS IN A TRACK FOR THE FORMAT USED
10365  037732  013737  001614  047752  4$:     MOV     L.DCYL,EXPCYL   ;GET STARTING VALUES FOR CYLINDER
10366  037740  113704  001607                  MOVB    L.DT,R4         ;TRACK
10367  037744  042704  177400                  BIC     #177400,R4
10368  037750  113705  001606                  MOVB    L.DS,R5         ;SECTOR
10369  037754  042705  177400                  BIC     #177400,R5
10370  037760  005301                          DEC     R1              ;ADJUST COUNT FOR ZERO DETECT
10371  037762  005205                  5$:     INC     R5              ;BUMP SECTOR COUNT
10372  037764  020503                          CMP     R5,R3           ;DID THIS MAKE SECTOR COUTN > 1 TRACK?
10373  037766  001010                          BNE     6$              ;NO - SKIP
10374  037770  005005                          CLR     R5              ;ELSE CLEAR SECTOR COUNT
10375  037772  005204                          INC     R4              ;BUMP TRACK COUNT
10376  037774  022704  000003                  CMP     #3,R4           ;DID THIS MAKE TRK COUNT > 1 CYLINDER?
10377  040000  001003                          BNE     6$              ;NO - SKIP
10378  040002  005004                          CLR     R4              ;ELSE CLEAR TRACK COUNT
10379  040004  005237  047752                  INC     EXPCYL          ;BUMP CYLINDER COUNT
10380  040010  005301                  6$:     DEC     R1              ;DEC COUNT
10381  040012  001363                          BNE     5$              ;IF ZERO - EXIT
10382  040014  010437  047756                  MOV     R4,EXPTRK       ;STORE EXPECTED TRACK
10383  040020  010537  047754                  MOV     R5,EXPSEC       ;STORE EXPECTED SECTOR (CYL ALREADY SLOW)
10384  040024  023737  001560  047752          CMP     T.DCYL,EXPCYL   ;TEST IF CYLINDER OK
10385  040032  001403                          BEQ     7$              ;YES - SKIP
10386  040034  052737  000010  050002          BIS     #CYLERR,GRP4ER  ;NO - SET ERROR FLAG
10387  040042  120437  001547          7$:     CMPB    R4,T.DA+1       ;TEST TRACK OK
10388  040046  001403                          BEQ     8$              ;YES - SKIP
10389  040050  052737  000020  050002          BIS     #TRKERR,GRP4ER  ;NO - SET ERROR FLAG
10390  040056  120537  001546          8$:     CMPB    R5,T.DA         ;TEST SECTOR COUNT OK
10391  040062  001403                          BEQ     9$              ;YES - SKIP
10392  040064  052737  000040  050002          BIS     #SECERR,GRP4ER  ;USE SET ERROR FLAG
10393  040072  012700  047760          9$:     MOV     #REALWC,R0
10394  040076  013720  001542                  MOV     T.WC,(R0)+      ;STORE REAL WORD COUNT
10395  040102  013720  001544                  MOV     T.BA,(R0)+      ;STORE REAL BUS ADDRESS
10396  040106  013720  001560                  MOV     T.DCYL,(R0)+    ;STORE REAL CYLINDER ADDRESS
10397  040112  113710  001547                  MOVB    T.DA+1,(R0)     ;STORE REAL TRACK ADDRESS
10398  040116  005720                          TST     (R0)+
```

```
10399  040120  113710  001546                  MOVB    T.DA,(R0)       ;STORE REAL SECTOR ADDRESS
10400  040124  104414                          RESREG
10401  040126  000204                          RTS     R4
10402
10403                                  .SBTTL  OPERATION CHECK ROUTINE
10404                                  ;*      THIS IS WHERE ALL HARDWARE ERROR INDICATORS AND SOME  SOFTWARE
10405                                  ;*      ERRORS ARE CHECKED. THE GENERAL PROCEDURE FLOW IS AS FOLLOWS:
10406                                  ;*      THE ROUTINE IS CALLED WITH A TRAP (TCHKOP).  THE LOCATION
10407                                  ;*      FOLLOWING THE TRAP CALL WILL HAVE AN ERROR TRAP WHICH
10408                                  ;*      THE ROUTINE WILL BYPASS IF NO ERROR IS FOUND. IF AN
10409                                  ;*      ERROR IS DETECTED, THE ERROR TRAP CALL IS MODIFIED
10410                                  ;*      BY THIS ROUTINE SUCH THAT THE ERROR TABLE ITEM WILL
10411                                  ;*      BE THE PROPER ITEM FOR  THE FORMAT REQUIRED BY THIS
10412                                  ;*      ERROR. THE ERROR TRAP WILL BE MADE EITHER ERROR 4,5,6,
10413                                  ;*      7, OR 10. REFER TO THE ERROR ITEM TABLE FOR A DESCRIPTION
10414                                  ;*      OF THE FORMAT AND WHICH ERRORS ARE DISPLAYED IN WHAT
10415                                  ;*      FORMAT.
10416
10417  040130  104413                  CHKWE:  SAVREG
10418  040132  011600                          MOV     (SP),R0         ;GET POINTER TO ERROR WORDS
10419  040134  012037  001242                  MOV     (R0)+,$TMP10    ;STORE EXPECTED ERROR GROUP 1
10420  040140  012037  001244                  MOV     (R0)+,$TMP11    ;STORE EXPECTED ERROR GROUP 2
10421  040144  012037  001246                  MOV     (R0)+,$TMP12    ;STORE EXPECTED ERROR GROUP 3
10422  040150  010016                          MOV     R0,(SP)         ;STORE RETURN
10423  040152  012737  177777  001250          MOV     #-1,$TMP13      ;SET FLAG - EXPECTED ERROR
10424  040160  000403                          BR      CHKST
10425
10426  040162  104413                  CHKOP:  SAVREG
10427  040164  005037  001250                  CLR     $TMP13          ;RESET EXPECTED ERROR FLAG
10428
10429  040170  104420                  CHKST:  TGETRK                  ;GET 611 REGS IO TRAP
10430  040172  005037  047774                  CLR     GRP1ER          ;CLEAR ERROR FLAGS
10431  040176  005037  047776                  CLR     GRP2ER
10432  040202  005037  050000                  CLR     GRP3ER
10433  040206  005037  050002                  CLR     GRP4ER
10434  040212  005037  050116                  CLR     GPSUMF          ;CLEAR SUMMARY FLAGS
10435  040216  032737  024000  001540          BIT     #CS1ERBIT,T.CS1 ;TEST IF ERROR SET IN CS1
10436  040224  001111                          BNE     4$              ;YES - SKIP
10437  040226  032737  177400  001550          BIT     #CS2ERBIT,T.CS2 ;TEST IF ERROR SET IN CS2
10438  040234  001105                          BNE     4$              ;YES - SKIP
10439  040236  032737  000070  001552          BIT     #DSERBIT,T.DS   ;TEST IF ERROR SET IN DS
10440  040244  001101                          BNE     4$              ;YES - SKIP
10441  040246  005737  001554                  TST     T.ER            ;TEST IF ERROR SET IN ER
10442  040252  001076                          BNE     4$              ;YES - SKIP
10443  040254  032737  100000  001540          BIT     #CERR,T.CS1     ;COMBINED ERROR SET?
10444  040262  001405                          BEQ     9$              ;NO - SKIP
10445  040264  052737  100000  047774          BIS     #CERNER,GRP1ER  ;SET ERROR FLAG IN GROUP 1
10446  040272  000137  041022                  JMP     25$             ;SKIP
10447
10448                                  ;       CODE TO CHECK WORD COUNT, BUFFER ADDRESS, CYLINDER, TRACK,
10449                                  ;       AND SECTOR AT THE END OF THE OPERATION. THIS IS DONE ONLY
10450                                  ;       IF CERR NOT SET BY THE OPERATION.
10451                                  ;
10452                                  ;       ALL OF THE ABOVE CONDITIONS ARE CHECKED AND A BIT SET FOR
10453                                  ;       EACH CHECK THAT FAILS. HOWEVER, ONLY ONE ERROR IS REPORTED.
10454                                  ;       THE ORDER OF PRIORITY FOR REPORTING THE ERROR IS THE ORDER
```

```
10455                                   ;       LISTED ABOVE.
10456
10457  040276  005737  001250           9$:     TST     $TMP13          ;TEST IF ERROR EXPECTED
10458  040302  001402                           BEQ     62$             ;NO - SKIP
10459  040304  000137  041022                   JMP     25$             ;YES - JUMP
10460  040310  013700  001540           62$:    MOV     T.CS1,R0        ;GET CS1
10461  040314  042700  177741                   BIC     #177741,R0      ;CHECK IF OPERATION IS READ DATA,
10462  040320  022700  000020                   CMP     #20,R0          ;WRITE DATA, OR WRITE CHECK. IF
10463  040324  002445                           BLT     3$              ;NOT, SKIP ALL CHECKING IN GROUP
10464  040326  022700  000030                   CMP     #30,R0          ;FOUR
10465  040332  003042                           BGT     3$
10466  040334  004437  037376                   JSR     R4,CHKWC        ;CHECK WORD COUNT
10467  040340  000000                           .WORD   0               ;EXPECTED WORD COUNT
10468  040342  004437  037430                   JSR     R4,CHKBA        ;CHECK BUS ADDRESSS
10469  040346  004437  037642                   JSR     R4,CHKCTS       ;CHECK CYL, TRACK, & SECTOR
10470  040352  005737  050002                   TST     GRP4ER          ;ANY GROUP 4 ERRORS?
10471  040356  001430                           BEQ     3$              ;NO - SKIP
10472  040360  016037  050120  061002           MOV     CMNDLB(R0),DF010A ;LOAD ADDRESS OF COMMAND MESSAGE
10473  040366  013700  050002                   MOV     GRP4ER,R0       ;PUT GROUP 4 ERROR FLAG IN R0
10474  040372  005001                           CLR     R1              ;CLEAR R1 FOR INDEX COUNTER
10475  040374  006200                   1$:     ASR     R0              ;SHIFT FLAGS - FIRST ONE ON RIGHT IS ERROR TO
10476  040376  103402                           BCS     2$              ;BE REPORTED, REST ARE IGNORED
10477  040400  005720                           TST     (R0)+           ;WHEN AN ERROR BIT IS FOUND,
10478  040402  000774                           BR      1$              ;GET THE ERROR MESSAGE ASSOCIATED
10479  040404  016037  050004  001450   2$.     MOV     GRP4MS(R0),EM10N ;WITH IT AND SET ERROR TABLE ITEM TO
10480  040412  016037  047744  001202           MOV     EXPWC(R0),$REG10 ;POINT TO THE MESSAGE. LOAD REG10 & 11
10481  040420  016037  047760  001204           MOV     REALWC(R0),$REG11 ;WITH EXPECTED & IS VALUES
10482  040426  104414                           RESREG                  ;RESTORE REGISTER
10483  040430  012776  000010  000000           MOV     #10,@(SP)       ;MAKE THE ERROR CALL POINT TO THE
10484  040436  000002                           RTI                     ;RIGHT TABLE ENTRY & RETURN.
10485  040440                           3$:
10486  040440  104414                           RESREG
10487  040442  062716  000002                   ADD     #2,(SP)         ;BUMP RETURN PAST ERROR
10488  040446  000002                           RTI                     ;RETURN
10489
10490                                   ;       THE FOLLOWING CODE BUILDS THE GROUP 1,2, & 3 ERROR WORDS.
10491
10492  040450  012700  047774           4$:     MOV     #GRP1ER,R0      ;SET UP GENERAL REGISTER AS POINTER
10493  040454  012701  001540                   MOV     #T.CS1,R1       ;CS1
10494  040460  012703  001550                   MOV     #T.CS2,R3       ;CS2
10495  040464  012704  001552                   MOV     #T.DS,R4        ;DS
10496  040470  012705  001554                   MOV     #T.ER,R5        ;AND ER
10497
10498  040474  051510                           BIS     (R5),(R0)       ;SET ALL BITS IN GRP1ER THAT
10499                                                                   ;CORRESPOND TO ERROR BITS IN RKER
10500  040476  042710  120701                   BIC     #ILF!ECH!BSE!HVRC!OPI!DCK,(R0) ; CLEAR ALL THAT DON'T BELONG GRP1
10501
10502  040502  032711  020000                   BIT     #SPAR,(R1)      ;TEST IF SPAR SET
10503  040506  001402                           BEQ     5$              ;NO - SKIP
10504  040510  052710  000001                   BIS     #SPARERR,(R0)   ;SET SPAR ERROR FLAG
10505
10506  040514  032714  000010           5$:     BIT     #ACLO,(R4)      ;TEST ACLO SET
10507  040520  001402                           BEQ     6$              ;NO - SKIP
10508  040522  052710  000100                   BIS     #ACLOERR,(R0)   ;SET ACLO ERROR FLAG
10509
10510  040526  032714  000020           6$:     BIT     #SPDLSS,(R4)    ;TEST SPEED LOSS SET
```

```
10511  040532  001402                    BEQ     7$              ;NO - SKIP
10512  040534  052710  000200            BIS     #SPDERR,(R0)    ;SET SPEED LOSS ERROR FLAG
10513
10514  040540  032714  000040    7$:     BIT     #DROT,(R4)      ;TEST IF DROT SET
10515  040544  001402                    BEQ     8$              ;NO - SKIP
10516  040546  052710  000400            BIS     #DROTERR,(R0)   ;SET DROT ERROR FLAG
10517
10518  040552  032711  100000    8$:     BIT     #CERR,(R1)      ;TEST CERR ITSELF SET
10519  040556  001002                    BNE     10$             ;YES - SKIP
10520  040560  032710  020000            BIT     #NCERWE,(R0)    ;SET NO CERR WITH ERROR FLAG
10521
10522  040564  012700  047776    10$:    MOV     #GRP2ER,R0      ;SET POINTER TO GROUP 2 ERROR FLAGS
10523
10524  040570  032715  000100            BIT     #ECH,(R5)       ;TEST IF ECH SET
10525  040574  001402                    BEQ     11$             ;NO - SKIP
10526  040576  052710  000001            BIS     #ECHERR,(R0)    ;SET ECH FLAG
10527
10528  040602  032715  100000    11$:    BIT     #DCK,(R5)       ;TEST DCK SET
10529  040606  001402                    BEQ     12$             ;NO - SKIP
10530  040610  052710  000002            BIS     #DCKERR,(R0)    ;SET DCK ERROR FLAG.
10531
10532  040614  032713  040000    12$:    BIT     #WCE,(R3)       ;TEST WRITE CHECK ERROR
10533  040620  001402                    BEQ     120$            ;NO - SKIP
10534  040622  052710  000004            BIS     #WCKERR,(R0)    ;SET WCE BIT
10535  040626  032713  100000    120$:   BIT     #DLT,(R3)       ;TEST DATA LATE
10536  040632  001402                    BEQ     13$             ;NO - SKIP
10537  040634  052710  000010            BIS     #DLTERR,(R0)    ;SET DLT ERROR FLAG
10538
10539  040640  032715  020000    13$:    BIT     #OPI,(R5)       ;TEST OPI SET
10540  040644  001402                    BEQ     14$             ;NO - SKIP
10541  040646  052710  000020            BIS     #OPIERR,(R0)    ;SET OPI ERROR FLAG
10542
10543  040652  032715  000400    14$:    BIT     #HVRC,(R5)      ;TEST HVRC SET
10544  040656  001402                    BEQ     16$             ;NO - SKIP
10545  040660  052710  000040            BIS     #HVRCERR,(R0)   ;SET HVRC FLAG
10546
10547  040664  032715  000200    16$:    BIT     #BSE,(R5)       ;TEST BSE ERROR FLAG
10548  040670  001402                    BEQ     17$             ;NO - SKIP
10549  040672  052710  000100            BIS     #BSERR,(R0)     ;SET BSE FLAG
10550
10551  040676  012700  050000    17$:    MOV     #GRP3ER,R0      ;SET POINTER TO GROUP 3 FLAGS
10552
10553  040702  032713  010000            BIT     #NED,(R3)       ;TEST NED SET
10554  040706  001402                    BEQ     18$             ;NO - SKIP
10555  040710  052710  000001            BIS     #NEDERR,(R0)    ;SET NED FLAG
10556
10557  040714  032711  004000    18$:    BIT     #CTO,(R1)       ;TEST CTO SET
10558  040720  001402                    BEQ     19$             ;NO - SKIP
10559  040722  052710  000002            BIS     #CTOERR,(R0)    ;SET CTO FLAG
10560
10561  040726  032713  000400    19$:    BIT     #UFE,(R3)       ;TEST UFE SET
10562  040732  001402                    BEQ     20$             ;NO - SKIP
10563  040734  052710  000004            BIS     #UFERR,(R0)     ;SET UFE FLAG
10564
10565  040740  032713  001000    20$:    BIT     #MDS,(R3)       ;TEST MDS SET
10566  040744  001402                    BEQ     21$             ;NO - SKIP
```

```
10567  040746  052710  000010                BIS     #MDSERR,(R0)    ;SET MDE FLAG
10568
10569  040752  032713  002000        21$:    BIT     #PGE,(R3)       ;TEST PGE SET
10570  040756  001402                        BEQ     22$             ;NO - SKIP
10571  040760  052710  000020                BIS     #PGERR,(R0)     ;SET PGE FLAG
10572
10573  040764  032713  004000        22$:    BIT     #NEM,(R3)       ;TEST NEM SET
10574  040770  001402                        BEQ     23$             ;NO - SKIP
10575  040772  052710  000040                BIS     #NEMERR,(R0)    ;SET NEM FLAG
10576
10577  040776  032713  020000        23$:    BIT     #UPE,(R3)       ;TEST UPE SET
10578  041002  001402                        BEQ     24$             ;NO - SKIP
10579  041004  052710  000100                BIS     #UPERR,(R0)     ;SET UPE FLAG
10580
10581  041010  032715  000001        24$:    BIT     #ILF,(R5)       ;TEST ILF SET
10582  041014  001402                        BEQ     25$             ;NO - SKIP
10583  041016  052710  000200                BIS     #ILFERR,(R0)    ;SET ILF FLAG.
10584
10585                                  ;       THE FOLLOWING CODE IS EXECUTED ONLY IF ERRORS WERE EXPECTED.
10586                                  ;       THE FLAG IN $TMP13 INDICATES IF
10587                                  ;       AN ERROR WAS EXPECTED AND THE CONTENTS OF TMP10,
10588                                  ;       TEMP11, & TEMP12 SPECIFY WHICH ERRORS. THESE ARE COMPARED AGAINST
10589                                  ;       THE ERRORS FOUND AND STORED IN GRP1ER, GRP2ER, AND GRP3ER.
10590                                  ;       THE CONTENTS OF GRP1,2, & 3 ARE MODIFIED TO INDICATE ERRORS THAT
10591                                  ;       OCCURRED BUT WERE NOT EXPECTED. THE CONTENTS OF $TMP10,11,
10592                                  ;       & 12 ARE MODIFIED TO INDICATE EXPECTED ERRORS THAT DID NOT
10593                                  ;       OCCUR. BOTH CONDITIONS CAN EXIST AT THE SAME TIME AND MUST
10594                                  ;       BE REPORTED.
10595
10596  041022  005737  001250        25$:    TST     $TMP13          ;CHECK IF AN ERROR WAS EXPECTED
10597  041026  001423                        BEQ     110$            ;NO - SKIP
10598  041030  012704  047774                MOV     #GRP1ER,R4      ;GET ADDRESS OF ERROR
10599  041034  012705  001242                MOV     #$TMP10,R5      ;GET ADDRESS OF EXPECTED ERRORS
10600
10601  041040  011500                26$:    MOV     (R5),R0         ;GET EXPECTED ERROR
10602  041042  011401                        MOV     (R4),R1         ;GET GROUP ERROR FLAGS
10603  041044  020001                        CMP     R0,R1           ;ARE THEY EQUAL?
10604  041046  001003                        BNE     27$             ;NO - SKIP
10605  041050  005000                        CLR     R0              ;CLEAR EXPECTED ED
10606  041052  005001                        CLR     R1              ;CLEAR OCCURED
10607  041054  000403                        BR      28$
10608
10609  041056  010003                27$:    MOV     R0,R3           ;STORE EXPECTED ERRORS
10610  041060  040100                        BIC     R1,R0           ;RESET EXPECTED THAT OCCURRED
10611  041062  040301                        BIC     R3,R1           ;RESET OCCURRED THAT EXPECTED
10612  041064  010025                28$:    MOV     R0,(R5)+        ;STORE EXPECTED THAT DID NOT OCCUR
10613  041066  010124                        MOV     R1,(R4)+        ;STORE OCCURRED THAT WERE NOT EXPECTED
10614  041070  022705  001250                CMP     #$TMP13,R5      ;ALL GROUPS CHECKED.
10615  041074  001361                        BNE     26$             ;NO - LOOP
10616
10617                                  ;       THE FOLLOWING CODE:
10618                                  ;
10619                                  ;       A. DETERMINES WHICH FORMAT IS TO BE USED
10620                                  ;       B. LOADS THE ADDRESSES OF THE ASCIZ TEXT INTO THE SELECTED
10621                                  ;          ERROR TABLE ITEM AND FORMAT FIELD
10622                                  ;       C. COUNTS THE NUMBER OF ERRORS THAT MUST BE REPORTED
```

```
10623                                   ;       D. GETS DRIVE STATUS IF GROUP 1 ERROR.
10624                                   ;
10625                                   ;       THE DECISION OF WHICH ERROR IS TO BE USED
10626                                   ;       IS BASED ON THE ERROR GROUP (OR GROUPS) THAT HAVE
10627                                   ;       FLAGS SET. IF ANY BIT IS SET IN GROUP 1,2, OR 3, GROUP 1 FORMAT (ERROR 4 OR 5)
10628                                   ;       WILL BE USED; ANY SET IN GROUP 2 OR 3, GROUP 2 (ERROR 6) WILL BE USED; AND A
10629                                   ;       FLAG SET IN GROUP 3 ONLY, GROUP 3 (ERROR 7) IS USED.
10630                                   ;
10631                                   ;       THE FORMAT TO BE USED IN THE CONTROLLING FACTOR IN HOW THE
10632                                   ;       ERROR TRAP IS CHANGED IN THE MAIN CALL. IF GROUP 1 FORMAT IS
10633                                   ;       USED THE ERROR TRAP WILL BE CHANGED TO ERROR 4 OR 5(DEPENDING ON
10634                                   ;       AVAILABILITY OF DRIVE STATUS), GROUP 2 FORMAT WILL BE ERROR
10635                                   ;       6, AND GROUP 3 WILL BE ERROR 7. ONLY THE LOW ORDER BYTE
10636                                   ;       OF THE ERROR TRAP WILL BE ALTERED. THE SP WILL BE
10637                                   ;       POINTING TO THE LOCATION THAT CONTAINS THE ERROR CALL TRAP.
10638                                   ;
10639                                   ;       IF THE STATUS IS READ FROM THE DRIVE WITH NO PROBLEM,
10640                                   ;       ERROR 4 IS USED. IF ANY ERROR IS ENCOUNTERED READING
10641                                   ;       STATUS, ERROR 5 IS USED. ERROR 5 INCLUDES A WARNING MESSAGE.
10642
10643  041076  005004                   110$:   CLR     R4              ;CLEAR COUNTERS
10644  041100  005005                           CLR     R5
10645  041102  012700  001224                   MOV     #$TMP1,R0       ;LOAD POINTERS FOR TEMPORARY STORAGE OF ADDRESS
10646  041106  012701  001226                   MOV     #$TMP2,R1       ;WHERE ASCIZ ADDRESSES GO
10647  041112  012703  050116                   MOV     #GPSUMF,R3      ;POINTERS TO GROUP SUMMARY FLAGS
10648  041116  012710  060762                   MOV     #DF007A,(R0)    ;PRESET FOR GRP3 ERR MESSAGE BUILD
10649  041122  012711  001442                   MOV     #DH7N,(R1)
10650  041126  013746  050000                   MOV     GRP3ER,-(SP)    ;GET GROUP 3 ERRORS, PUT ON STACK
10651  041132  004437  036242                   JSR     R4,BITCNT       ;GO COUNT NUMBER AT ERRORS
10652  041136  005716                           TST     (SP)            ;ANY ERRORS?
10653  041140  001403                           BEQ     29$             ;NO - SKIP
10654  041142  061605                           ADD     (SP),R5         ;ADD IN ERROR TOTAL
10655  041144  052713  000004                   BIS     #GRP3ST,(R3)    ;SET BIT TO INDICATE GROUP 3 ERROR
10656
10657  041150  005726                   29$:    TST     (SP)+           ;CLEAR OFF STACK
10658  041152  005737  001250                   TST     $TMP13          ;ERROR EXPECTED
10659  041156  001412                           BEQ     31$             ;NO - SKIP
10660  041160  013746  001246                   MOV     $TMP12,-(SP)    ;PUT GROUP 3 NOT RECEIVED ERRORS ON STACK
10661  041164  004437  036242                   JSR     R4,BITCNT       ;COUNT NUMBER OF ERRORS.
10662  041170  005716                           TST     (SP)            ;WERE THERE ANY
10663  041172  001403                           BEQ     30$             ;NO - SKIP
10664  041174  052713  000040                   BIS     #GP3NR,(R3)     ;SET GROUP 3 NOT RECEIVED ERROR FLAG
10665  041200  061604                           ADD     (SP),R4         ;ADD COUNT TO TOTAL THESE
10666
10667  041202  005726                   30$:    TST     (SP)+           ;CLEAR OFF STACK
10668  041204  013746  047776           31$:    MOV     GRP2ER,-(SP)    ;GET GROUP 2 ERRORS FOR COUNTING
10669  041210  004437  036242                   JSR     R4,BITCNT       ;COUNT BITS
10670  041214  005716                           TST     (SP)            ;ANY SET?
10671  041216  001407                           BEQ     32$             ;NO - SKIP
10672  041220  052713  000002                   BIS     #GRP2ST,(R3)    ;SET FLAG FOR GROUP 2 ERRORS
10673  041224  061605                           ADD     (SP),R5         ;ADD INTO TOTAL
10674  041226  012710  060736                   MOV     #DF006A,(R0)    ;STORE ADDRESS FOR BUILDING REPORT
10675  041232  012711  001432                   MOV     #DH6N,(R1)
10676
10677  041236  005726                   32$:    TST     (SP)+           ;CLEAR OFF STACK
10678  041240  005737  001250                   TST     $TMP13          ;ANY EXPECTED ERRORS
```

```
10679  041244  001416                    BEQ     34$             ;NO - SKIP
10680  041246  013746  001244            MOV     $TMP11,-(SP)    ;GET GROUP 2 NOT RECEIVED ERRORS
10681  041252  004437  036242            JSR     R4,BITCNT       ;COUNT NUMBER OF BITS
10682  041256  005716                    TST     (SP)            ;ANY SET?
10683  041260  001407                    BEQ     33$             ;NO - SKIP
10684  041262  052713  000020            BIS     #GP2NR,(R3)     ;SET FLAG FOR GROUP 2 NOT RECEIVED
10685  041266  061604                    ADD     (SP),R4         ;ADD INTO TOTAL
10686  041270  012710  060736            MOV     #DF006A,(R0)    ;STORE ADDRESS FOR BUILDING REPORT
10687  041274  012711  001432            MOV     #DH6N,(R1)
10688
10689  041300  005726            33$:    TST     (SP)+           ;CLEAR OFF STACK
10690  041302  013746  047774    34$:    MOV     GRP1ER,-(SP)    ;GET GROUP 1 ERROR FLAGS
10691  041306  004437  036242            JSR     R4,BITCNT       ;COUNT THE NUMBER OF BITS
10692  041312  005716                    TST     (SP)            ;ANY SET?
10693  041314  001407                    BEQ     35$             ;NO - SKIP
10694  041316  052713  000001            BIS     #GRP1ST,(R3)    ;SET FLAG FOR GROUP 1 ERRORS SET
10695  041322  061605                    ADD     (SP),R5         ;ADD INTO TOTAL
10696  041324  012710  060652            MOV     #DF004A,(R0)    ;LOAD ADDRESS FOR BUILDING REPORT
10697  041330  012711  001412            MOV     #DH4N,(R1)
10698  041334  005726            35$:    TST     (SP)+           ;CLEAR OFF STACK
10699  041336  005737  001250            TST     $TMP13          ;ANY EXPECTED ERRORS?
10700  041342  001416                    BEQ     60$             ;NO - SKIP
10701  041344  013746  001242            MOV     $TMP10,-(SP)    ;GET GROUP 1 NO RECEIVED ERROR
10702  041350  004437  036242            JSR     R4,BITCNT       ;COUNT # OF BITS
10703  041354  005716                    TST     (SP)            ;ANY SET?
10704  041356  001407                    BEQ     36$             ;NO - SKIP
10705  041360  052713  000010            BIS     #GP1NR,(R3)     ;SET FLAG FOR GROUP 1 NOT RECEIVED
10706  041364  061604                    ADD     (SP),R4         ;ADD INTO TOTAL
10707  041366  012710  060652            MOV     #DF004A,(R0)    ;LOAD ADDRESS FOR BUILDING REPORT
10708  041372  012711  001412            MOV     #DH4N,(R1)
10709  041376  005726            36$:    TST     (SP)+           ;CLEAR OFF STACK.
10710  041400  032713  000011    60$:    BIT     #GRP1ST!GP1NR,(R3) ;ANY GROUP 1 ERROR
10711  041404  001414                    BEQ     52$             ;NO - SKIP
10712  041406  042713  040000            BIC     #DRSTER,(R3)
10713  041412  004437  036762            JSR     R4,GETDRS
10714  041416  000401                    BR      51$             ;ERROR RETURN
10715  041420  000406                    BR      52$             ;NO ERROR RETURN
10716  041422  012710  060702    51$:    MOV     #DF005A,(R0)    ;CHANGE TO FORMAT 5 - STORE ADDRESS
10717  041426  012711  001422            MOV     #DH5N,(R1)      ;FOR BUILDING REPORT
10718  041432  052713  040000            BIS     #DRSTER,(R3)    ;SET DRIVE STATUS ERROR
10719  041436                    52$:
10720
10721                            ;       THE ERRORS ARE COUNTED, FLAGS SET TO INDICATE WHICH ERRORS
10722                            ;       ARE TO BE REPORTED, AND THE ERROR FORMAT HAS BEEN SELECTED.
10723                            ;       THE FOLLOWING CODE WILL TYPE ALL THE ERRORS, LOAD THE
10724                            ;       PROPER HEADER MESSAGE ADDRESS IN THE ERROR ITEM TABLE
10725                            ;       AND LOAD THE PROPER HEADER MESSAGE ADDRESS IN THE PROPER
10726                            ;       DF TABLE.
10727                            ;
10728                            ;       AT THIS TIME
10729                            ;           R5 CONTAINS EITHER THE NUMBER OF ERRORS THAT OCCURRED BUT WERE
10730                            ;               NOT EXPECTED OR
10731                            ;               THE NUMBER OF ERRORS THE OCCURRED IF NONE WERE EXPECTED
10732                            ;           R4 CONTAINS THE NUMBER OF ERRORS THAT WERE EXPECTED BUT
10733                            ;               DID NOT OCCUR.
10734                            ;           $TMP10 CONTAINS GROUP 1 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
```

```
10735                                   ;       $TMP11 CONTAINS GROUP 2 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
10736                                   ;       $TMP12 CONTAINS GROUP 3 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
10737                                   ;       GRP1ER CONTAINS GROUP 1 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
10738                                   ;       GRP2ER CONTAINS GROUP 2 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
10739                                   ;       GRP3ER CONTAINS GROUP 3 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
10740                                   ;     (R1)=#$TMP2 CONTAINS THE ADDRESS OF THE HEADER MESSAGE ADDRESS IN
10741                                   ;           OF THAT MUST BE ALTERED TO IDENTIFY THE OPERATION
10742                                   ;     (R0)=#$TMP1 CONTAINS THE ADDRESS OF THE HEADER MESSAGE ADDRESS IN
10743                                   ;           THE ERROR ITEM TABLE THAT MUST BE ALTERED TO PROVIDE A
10744                                   ;           PROPER MESSAGE TO REPORT.
10745                                   ;     (R3)=#GRSUMF CONTAIN FLAGS TO INDICATE WHICH OF THE GROUP
10746                                   ;           ERROR FLAG FIELDS HAVE ERROR BITS STORED.
10747  041436  032777  020000  137474          BIT     #SW13,@SWR      ;IS REPORT INHIBITED?
10748  041444  001402                          BEQ     37$             ;NO - SKIP
10749  041446  000137  042034                JMP     49$             ;ELSE EXIT
10750  041452  005737  001250          37$:  TST     $TMP13          ;WERE ERRORS EXPECTED?
10751  041456  001004                          BNE     38$             ;YES - SKIP
10752
10753                                   ;     IF NO ERRORS WERE EXPECTED, $TMP10,11, &12 ARE NOT MEANINGFUL
10754
10755  041460  012771  057655  000000          MOV     #DH007,@(R1)    ;HEADER = ERROR IN OPERATION
10756  041466  000411                          BR      39$
10757  041470  012771  057476  000000  38$:   MOV     #DH005,@(R1)    ;PRESET HDRMSG = EXPECTED NOT SET
10758  041476  032713  000070                BIT     #GP1NR!GP2NR!GP3NR,(R3) ;ANY NOT RECEIVED ERRORS?
10759  041502  001003                          BNE     39$             ;YES - SKIP
10760  041504  012771  057573  000000          MOV     #DH006,@(R1)    ;SET MESSAGE TO UNEXPECTED ERROR SET
10761  041512  013701  001540          39$:  MOV     T.CS1,R1        ;GET CS1
10762  041516  042701  177741                BIC     #177741,R1      ;CLEAR ALL BUT COMMAND
10763
10764  041522  016170  050120  000000          MOV     CMNDLB(R1),@(R0) ;MOVE ADDRESS OF COMMAND MESSAGE
10765                                                                  ;INTO REPORT
10766  041530  032713  000007                BIT     #GRP1ST!GRP2ST!GRP3ST,(R3) ;ANY GPR ERRORS?
10767  041534  001462                          BEQ     46$             ;NO - SKIP GPR REPORT
10768
10769                                   ;     PRINT ALL THE ERRORS CONTAINED IN THE GPR1,2,3ER(UNEXPECTED ERRORS)
10770
10771  041536  013701  050000                MOV     GRP3ER,R1       ;GET GROUP 3 ERROR FLAGS
10772  041542  012700  050020                MOV     #GRP3MS,R0      ;SET POINTER TO GRP3 ERROR MESSAGES
10773  041546  005037  001252          140$: CLR     $TMP14          ;CLEAR GROUP PRINTING INDICATOR
10774  041552  012737  000021  001254  40$:  MOV     #17.,$TMP15     ;PRESET SHIFT COUNT
10775  041560  000241                          CLC                     ;CLEAR CARRY
10776  041562  006001                    41$:  ROR     R1              ;ROTATE ERROR FLAGS
10777  041564  103406                          BCS     42$             ;WAS BIT SHIFTED OUT SET?
10778  041566  062700  000002          141$: ADD     #2,R0           ;NO - BUMP POINTER
10779  041572  005337  001254                DEC     $TMP15          ;DEC SHIFT COUNT
10780  041576  001371                          BNE     41$             ;LOOP IF SHIFT NOT ZERO
10781  041600  000411                          BR      44$
10782  041602  011037  041614          42$:  MOV     (R0),43$        ;GET ERROR MESSAGE ADDRESS FROM TABLE
10783  041606  104401  001273                TYPE    ,$CRLF          ;TYPE CRLF
10784  041612  104401                          TYPE                    ;TYPE ERROR MESSAGE
10785  041614  000000                    43$:  .WORD                   ;MESSAGE ADDRESS GOES HERE
10786  041616  005305                          DEC     R5              ;DECEREMENT TOTAL ERROR COUNT.
10787  041620  001362                          BNE     141$            ;LOOP IF ZERO
10788  041622  000427                          BR      46$             ;FLSE EXIT GPR ERROR PRINT LOOP
10789
10790  041624  005713                    44$:  TST     (R3)            ;TEST GPSUMF FLAG FOR PRINTING ERROR NOT RECEIVED
```

```
10791  041626  100455                        BMI     47$             ;YES - SKIP
10792  041630  005737  001252                TST     $TMP14          ;PRINTING GROUP 3?
10793  041634  001007                        BNE     45$             ;NO -SKIP
10794  041636  013701  047776                MOV     GRP2ER,R1       ;ELSE SET TO GROUP 2, GET GRP2ER
10795  041642  012700  050040                MOV     #GRP2MS,R0      ;& SET POINTER TO GROUP 2 ERROR MESSAGE TABLE
10796  041646  005237  001252                INC     $TMP14          ;BUMP TO INDICATE PRINTING GROUP 2
10797  041652  000737                        BR      40$             ;GO RESTART PRINT LOOP
10798  041654  022737  000002  001252  45$:  CMP     #2,$TMP14       ;PRINTING GROUP 1?
10799  041662  001407                        BEQ     46$             ;YES - EXIT GPR ERROR PRINT LOOP.
10800  041664  013701  047774                MOV     GRP1ER,R1       ;ELSE SET TO GROUP 1, GET GROUP 1 ERROR
10801  041670  012700  050056                MOV     #GRP1MS,R0      ;SET POINTER TO GROUP 1 ERROR MESSAGE TABLE
10802  041674  005237  001252                INC     $TMP14          ;BUMP TO INDICATE PRINTING GROUP 1
10803  041700  000724                        BR      40$             ;RESTART PRINT LOOP.
10804
10805  041702  005737  001250          46$:  TST     $TMP13          ;EXPECTING ERRORS?
10806  041706  001452                        BEQ     49$             ;NO - SKIP
10807
10808                                  ;      PRINT ALL ERRORS CONTAINED IN $TMP10, 11, 12(NOT RECEIVED ERRORS)
10809
10810  041710  005713                        TST     (R3)            ;TEST IF PRINTING NOT RECEIVED ERRORS
10811  041712  100423                        BMI     47$             ;YES - SKIP
10812  041714  032713  000070                BIT     #GP1NR!GP2NR!GP3NR,(R3) ;ANY NOT RECEIVED ERRORS
10813  041720  001445                        BEQ     49$             ;NO - SKIP
10814  041722  032713  000007                BIT     #GRP1ST!GRP2ST!GRP3ST,(R3) ;ANY NOT RECEIVED ERRORS?
10815  041726  001404                        BEQ     146$            ;NO - SKIP LABEL FOR UNEXPECTED ERRORS
10816  041730  104401  001273                TYPE    ,$CRLF          ;TYPE CRLF
10817  041734  104401  057573                TYPE    ,DH006          ;TYPE HEADER FOR PREVIOUS ERRORS
10818  041740  052737  100000  050116  146$:  BIS     #REPNR,GPSUMF   ;SET PRINTING NOT RECEIVED ERRORS SWITCH
10819  041746  010405                        MOV     R4,R5           ;MOVE TOTAL ERRORS TO R5
10820  041750  013701  001246                MOV     $TMP12,R1       ;GET GRP3 NOT RECEIVED ERRORS
10821  041754  012700  050020                MOV     #GRP3MS,R0      ;SET POINTER TO GROUP 3 MESSAGES
10822  041760  000672                        BR      140$            ;GO START PRINT LOOP
10823  041762  005737  001252          47$:  TST     $TMP14          ;PRINTING GROUP 3?
10824  041766  001007                        BNE     48$             ;NO - SKIP
10825  041770  013701  001244                MOV     $TMP11,R1       ;ELSE SETUP TO PRINT GROUP 2 - GET ERRORS
10826  041774  012700  050040                MOV     #GRP2MS,R0      ;& SET POINTER TO GROUP 2 MESSAGE TABLE
10827  042000  005237  001252                INC     $TMP14          ;BUMP TO INDICATE GROUP 2 PRINTING
10828  042004  000662                        BR      40$             ;GO START PRINT LOOP
10829  042006  022737  000002  001252  48$:  CMP     #2,$TMP14       ;PRINTING GROUP 1?
10830  042014  001407                        BEQ     49$             ;YES - EXIT LOOP
10831  042016  013701  001242                MOV     $TMP10,R1       ;SET POINTER TO GROUP 1 MESSAGE
10832  042022  012700  050056                MOV     #GRP1MS,R0      ;TABLE AND GET GROUP 1 ERRORS.
10833  042026  005237  001252                INC     $TMP14          ;BUMP TO INDICATE GROUP 1 PRINTING
10834  042032  000647                        BR      40$             ;START LOOP AGAIN.
10835
10836  042034  032713  000077          49$:  BIT     #77,(R3)        ;TEST IF ANY ERRORS TO BE REPORTED
10837                                                                 ;  GRP1ST!GRP2ST!GRP3ST
10838                                                                 ;  GP1NR!GP2NR!GP3NR
10839  042040  001004                        BNE     61$             ;YES - SKIP
10840  042042  104414                        RESREG                  ;ELSE EXIT
10841  042044  062716  000002                ADD     #2,(SP)         ;BUMP FOR GOOD RETURN
10842  042050  000002                        RTI
10843
10844  042052  112776  000007  000000  61$:  MOVB    #7,@(SP)        ;PRESET FOR GROUP 3 ERROR RETURN.
10845  042060  032713  000022                BIT     #GRP2ST!GP2NR,(R3) ;ANY GROUP 2 ERRORS?
10846  042064  001403                        BEQ     50$             ;NO - SKIP
```

```
10847  042066  112776  000006  000000          MOVB    #6,@(SP)        ;ELSE SET FOR GROUP 2 ERROR RETURN
10848
10849  042074  032713  000011          50$:    BIT     #GRP1ST!GP1NR,(R3) ;ANY GROUP 1 ERRORS?
10850  042100  001411                          BEQ     53$             ;NO - SKIP
10851  042102  112776  000004  000000          MOVB    #4,@(SP.        ;ELSE SET FOR GROUP 1 ERROR RETURN.
10852  042110  032713  040000                  BIT     #DRSTER,(R3)    ;CHECK IF ERROR GETTING DRIVE STATUS
10853  042114  001403                          BEQ     53$             ;NO - SKIP
10854  042116  112776  000005  00000C          MOVB    #5,@(SP)        ;ELSE CHANGE RETURN FORM GROUP 1
10855
10856  042124  005737  001264          53$:    TST     $ESCAPE         ;CHECK IF ESCAPE ALREADY SET
10857  042130  001011                          BNE     54$             ;YES - SKIP
10858  042132  013700  001302                  MOV     $TESTN,R0       ;SET UP $ESCAPE TO FORCE
10859  042136  006300                          ASL     R0              ;ABORT TO PRESENT TEST AFTER
10860  042140  016037  033122  001264          MOV     $SW08TB(R0),$ESCAPE ;ERROR IS REPORTED
10861  042146  162737  000002  001264          SUB     #2,$ESCAPE      ;BUT GO TO NEXT SCOPE STATEMENT
10862  042154  104414                  54$:    RESREG
10863  042156  000002                          RTI                     ;RETURN
10864
10865                                  ;;*****************************************************************
10866                                  .SBTTL  BAD SECTOR CHECK
10867                                  ;*      THE FIELD WHOSE ADDRESS IS IN THE LOCATION AFTER THE
10868                                  ;*      CALL IS CHECKED TO SEE IF ANY SECTORS ARE LISTED THEREIN
10869                                  ;*      THAT HAVE THE CYLINDER AND TRACK ADDRESS SPECIFIED IN
10870                                  ;*      L.DCYL AND L.DT. IF A SECTOR IS FOUND IN THIS FIELD
10871                                  ;*      THAT IS BAD FOR THAT CYLINDER AND TRACK, THE SECTOR NUMBER
10872                                  ;*      IS PLACED ON THE STACK. THE TOTAL NUMBER OF BAD SECTORS
10873                                  ;*      IS PLACED ON THE STACK AFTER THE ENTIRE
10874                                  ;*      FIELD IS SEARCHED.
10875                                  ;*
10876                                  ;*      CALL:   JSR     R4,BDSRCK
10877                                  ;*              <ADDRESS OF FIELD TO BE SEARCHED>
10878                                  ;;*****************************************************************
10879
10880  042160  012637  001236  BDSRCK: MOV     (SP)+,$TMP6     ;STORE OLD R4 CONTENTS
10881  042164  010437  001240          MOV     R4,$TMP7        ;GET RETURN ADDRESS
10882  042170  011404                  MOV     (R4),R4         ;GET POINTER TO FIELD TO BE CHECKED
10883  042172  005037  001234          CLR     $TMP5           ;CLEAR A COUNTER
10884  042176  005714          1$:     TST     (R4)            ;TEST IF FIELD HAS NO (OR NO MORE)ENTRIES
10885  042200  100417                  BMI     4$              ;YES - EXIT
10886  042202  023724  001614          CMP     L.DCYL,(R4)+    ;IS THIS ENTRY FOR THIS CYLINDER?
10887  042206  001012                  BNE     3$              ;NO - SKIP
10888  042210  005204                  INC     R4              ;BUMP TO TRACK
10889  042212  123714  001607          CMPB    L.DT,(R4)       ;IS ENTRY FOR THIS TRACK?
10890  042216  001005                  BNE     2$              ;NO - SKIP
10891  042220  005046                  CLR     -(SP)           ;CLEAR STACK LOCATION
10892  042222  114416                  MOVB    -(R4),(SP)      ;PUT SECTOR NUMBER ON STACK
10893  042224  005237  001234          INC     $TMP5           ;BUMP COUNTER
10894  042230  000401                  BR      3$              ;BRANCH
10895
10896  042232  005304          2$:     DEC     R4              ;DECREMENT POINTER TO WORD ALIGN
10897  042234  005724          3$:     TST     (R4)+           ;BUMP TO NEXT ENTRY
10898  042236  000757                  BR      1$              ;TEST NEXT ENTRY
10899
10900  042240  013746  001234  4$:     MOV     $TMP5,-(SP)     ;PUT COUNT ON STACK
10901  042244  013746  001236          MOV     $TMP6,-(SP)     ;PUT OLD R4 CONTENTS BACK ON STACK
10902  042250  013704  001240          MOV     $TMP7,R4        ;SET UP RETURN
```

```
10903  042254  005724                          TST     (R4)+           ;BUMP PAST PARAMETER
10904  042256  000204                          RTS     R4              ;RETURN
10905
10906                                  ;;*****************************************************************
10907                                  .SBTTL  DATA GENERATION AND COMPARE ROUTINE
10908                                  ;*  CALLS:      JSR     R4,GENCOM
10909                                  ;*              CONTROL WORD
10910                                  ;*
10911                                  ;*              JSR     R4,GENCOM
10912                                  ;*              CONTROL WORD
10913                                  ;*              LENGTH
10914                                  ;*
10915                                  ;*              JSR     R4,GENCOM
10916                                  ;*              CONTROL WORD
10917                                  ;*              RELOCATION CONSTANT
10918                                  ;*              LENGTH
10919                                  ;*
10920                                  ;*  RETURN:     RTS     R4
10921                                  ;*      R4 IS ADJUSTED IN THE CODE FOR THE FOLLOWING RETURNS:
10922                                  ;*      THE FIRST CALL RETURNS TO THE LOCATION FOLLOWING THE
10923                                  ;*      CONTROL WORD. THIS IS UNCONDITIONAL.
10924                                  ;*
10925                                  ;*      THE SECOND CALL RETURNS TO THE LOCATION FOLLOWING THE LENGTH IF
10926                                  ;*      THE OPERATION REQUIRES DATA COMPARE AND DATA MISCOMPARED.
10927                                  ;*      IF DATA IS TO BE GENERATED ONLY OR NO DATA COMPARE
10928                                  ;*      ERRORS OCCURRED, THE RETURN IS TO LENGTH +4.
10929                                  ;*
10930                                  ;*      THE THIRD CALL IS IDENTICAL TO THE SECOND.
10931                                  ;*
10932                                  ;*  DEFINITION OF CONTROL WORD:
10933                                  ;*      BIT 15 - DO COMPARE OPERATION OF IBUFF (SOURCE) TO OBUFF
10934                                  ;*               (DESTINATION). EXPECTED VALUES ARE IN OBUFF (DESTINATION.
10935                                  ;*      BIT 14 - RESUME COMPARE OPERATION FROM POINT LEFT BY LAST COMPARE.
10936                                  ;*      BIT 13 - INVOKE MEMORY MANAGEMENT FOR SOURCE (IBUFF).
10937                                  ;*      BIT 12 - INVOKE MEMORY MANAGEMENT FOR DESTINATION (OBUFF).
10938                                  ;*      BIT 11 - REPEAT FIRST WORD OF SELECTED PATTERN THROUGHOUT OBUFF.
10939                                  ;*      BIT 10 - CLEAR IBUFF TO PATTERN SELECTED.
10940                                  ;*      BIT 9 - BUILD HEADERS, CONSIDERING BS FILES
10941                                  ;*      BIT 8 - BUILD HEADERS, ALL SECTORS INDICATE GOOD SECTORS.
10942                                  ;*      BIT 7 - HEADER OPERATION SPECIFIED (EITHER COMPARE OR BUILD).
10943                                  ;*      BIT 6 TO 0 - PATTERN SELECT FIELD, OCTAL ENCODED. 0 INDICATES
10944                                  ;*               NO DATA GENERATION, 1 IS ALL ZEROS, AND 7 IS ALL ONES.
10945                                  ;*               OTHER PATTERNS PROVIDED ARE PATTERNS 2-6, 8-16.
10946                                  ;*
10947                                  ;*  EXPLANATION OF CALLS:
10948                                  ;*      THE CALL WITH CONTROL WORD THE ONLY PARAMETER IS USED FOR
10949                                  ;*      BUILDING OR COMPARING HEADERS OR RESUMING A COMPARE OPERATION.
10950                                  ;*
10951                                  ;*      THE CALL WITH CONTROL WORD AND LENGTH AS PARAMETERS IS USED
10952                                  ;*      FOR DATA GENERATION OR COMPARE AND FOR IBUFF INITALIZATION.
10953                                  ;*
10954                                  ;*      THE CALL WITH CONTROL WORD, RELOCATION CONSTANT, AND LENGTH IS
10955                                  ;*      USED FOR DATA GENERATION OR COMPARE WITH MEMORY MANAGEMENT.
10956                                  ;*
10957                                  ;*
10958                                  ;*  DESCRIPTION:
```

```
10959                                   ;*
10960                                   ;*      THIS ROUTINE IS MULTI-PURPOSE AND WILL PERFORM THE FOLLOWING:
10961                                   ;*        A.    BUILD HEADERS, EITHER 20 OR 22 SECTORS/TRACK MODE. THE
10962                                   ;*              ROUTINE WILL BUILD THE HEADERS AS ALL GOOD SECTORS (BIT 8)
10963                                   ;*              OR TAKE THE BAD SECTOR FILES (HARDWARE OR SOFTWARE) FOR
10964                                   ;*               EITHER FORMAT) INTO ACCOUNT AND BUILD THE HEADERS WITH THE
10965                                   ;*              SECTORS MARKED BAD IF ANY SECTORS FOR THE CYLINDER - TRACK
10966                                   ;*              ARE LISTED THEREIN (BIT 9).
10967                                   ;*
10968                                   ;*        B.    COMPARE THE CONTENTS OF IBUFF AND OBUFF (BIT 15). THE
10969                                   ;*              CONTENTS OF THE BUFFER MAY BE HEADERS OR DATA. A
10970                                   ;*              HEADER COMPARE OPERATION MAY BE SPECIFIED (BIT 7) WHICH
10971                                   ;*              WILL CAUSE THE COMPARE TO BE LIMITED TO 74(8) OR 102(8)
10972                                   ;*              WORDS OF HEADERS. THE LENGTH DEPENDS ON THE FORMAT
10973                                   ;*              BIT THAT WAS LAST SPECIFIED IN L.CS1. THE HEADERS
10974                                   ;*              MAY BE BUILT BEFORE THE COMPARE AS PART OF THE
10975                                   ;*              OPERATION (BIT 15 AND BIT 8 OR 9). DATA CAN ALSO BE
10976                                   ;*              GENERATED BEFORE THE COMPARE (NON-ZERO BITS 6-0).
10977                                   ;*
10978                                   ;*        C.    RESUME COMPARE OPERATION. IF A COMPARE OPERATION
10979                                   ;*              DETECTS A MISCOMPARE, THE ROUTINE RETURNS TO CALLER
10980                                   ;*              BUT STORES PARAMETERS SUCH THAT THE COMPARE CAN BE
10981                                   ;*              RESUMED. THIS IS DONE BY CALLING GENCOM WITH BIT 14
10982                                   ;*              SET IN THE CONTROL WORD.
10983                                   ;*
10984                                   ;*        D.    DATA GENERATION OR COMPARE USING MEMORY MANAGEMENT.
10985                                   ;*              MEMORY MANAGEMENT CAN BE INVOKED FOR EITHER
10986                                   ;*              SOURCE OR DESTINATION BUT NOT FOR BOTH. IN THIS
10987                                   ;*              MANNER, DATA GENERATION CAN BE MADE TO
10988                                   ;*              PLACE DATA ANYWHERE IN AVAILABLE MEMORY. LIKEWISE
10989                                   ;*              DATA COMPARE WILL COMPARE THE CONTENTS OF IBUFF TO
10990                                   ;*              ANY AREA OF AVAILABLE MEMORY.
10991
10992   042260                          GENCOM:
10993   042260  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10994   042262  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10995   042264  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10996   042266  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10997   042270  012400                          MOV     (R4)+,R0        ;GET PARAMETER WORD
10998   042272  012737  056222  001520          MOV     #EM54,EM15N     ;PRESET FOR HEADER COMPARE ERROR
10999   042300  032700  000200                  BIT     #BIT7,R0        ;HEADER OPERATION SPECIFIED?
11000   042304  001005                          BNE     18$             ;YES - SKIP
11001   042306  012737  056251  001520          MOV     #EM55,EM15N     ;CHANGE FOR DATA COMPARE ERROR
11002   042314  000137  043046                  JMP     17$             ;ELSE JUMP TO DATA ROUTINE
11003   042320                          18$:
11004   042320  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
11005   042322  032700  001400                  BIT     #BIT8!BIT9,R0   ;MUST HEADERS BE BUILT?
11006   042326  001002                          BNE     19$             ;YES - SKIP
11007   042330  000137  042630                  JMP     11$             ;ELSE JUMP TO HEADER COMPARE
11008   042334  113701  001607          19$:    MOVB    L.DT,R1         ;START HEADER BUILD ROUTINE
11009   042340  013703  001614                  MOV     L.DCYL,R3       ;GET TRACK AND CYL
11010   042344  012705  000005                  MOV     #5,R5           ;SET COUNT TO SHIFT TRACK FOR HDR WORD
11011
11012   042350  006301                  1$:     ASL     R1              ;SHIFT TRACK
11013   042352  005305                          DEC     R5              ;DECREMENT TRACK
11014   042354  001375                          BNE     1$              ;LOOP UNTIL COUNT 0
```

```
11015
11016  042356  012704  000026              MOV     #26,R4          ;PRESET FOR 26 SECTOR MODE
11017  042362  032737  010000  001600      BIT     #CFMT,L.CS1     ;IS IT 24 SECTOR MODE?
11018  042370  001404                      BEQ     2$              ;NO - SKIP
11019  042372  012704  000024              MOV     #24,R4          ;CHANGE COUNT FOR 24 SECTOR MODE
11020  042376  052701  001000              BIS     #BIT9,R1        ;SET 24 SECTOR MODE BIT IN WRD 2 OF HDR
11021
11022  042402  052701  140000      2$:     BIS     #BIT15!BIT14,R1 ;SET BS BITS TO INDICATE GOOD SECTOR
11023  042406  012705  072414              MOV     #OBUFF,R5       ;SET POINTER TO ADDRESS WHERE HEADERS GO
11024  042412  010325              3$:     MOV     R3,(R5)+        ;INSERT CYLINDER
11025  042414  010125                      MOV     R1,(R5)+        ;INSERT TRACK AND SECTOR
11026  042416  010337  001224              MOV     R3,$TMP1        ;CALCULATE HVRC WORD
11027  042422  010115                      MOV     R1,(R5)
11028  042424  040137  001224              BIC     R1,$TMP1
11029  042430  040315                      BIC     R3,(R5)
11030  042432  053725  001224              BIS     $TMP1,(R5)+     ;COMPLETE HVRC WORD INSERTION
11031
11032  042436  005304                      DEC     R4              ;DECREMENT HEADER COUNT
11033  042440  001402                      BEQ     4$              ;DONE? - YES, SKIP
11034  042442  005201                      INC     R1              ;BUMP SECTOR
11035  042444  000762                      BR      3$              ;LOOP
11036
11037  042446  032700  001000      4$:     BIT     #BIT9,R0        ;MUST HEADERS BE CORRECTED FOR TABLE ENTRIES?
11038  042452  001003                      BNE     5$              ;YES - SKIP
11039  042454  005700              10$:    TST     R0              ;IS THIS A COMPARE OPERATION?
11040  042456  100464                      BMI     11$             ;YES-GO DO HDR COMPARE
11041  042460  000534                      BR      50$             ;ELSE GET OUT
11042
11043  042462  013737  001640  042520  5$:     MOV     BSF26P,6$       ;PRESET FOR BS FACTORY LIST
11044  042470  012737  100000  001224          MOV     #BIT15,$TMP1    ;SET BIT TO BE RESET IN BAD HEADER
11045  042476  032737  010000  001600          BIT     #CFMT,L.CS1     ;IS THIS 26 SECTOR MODE?
11046  042504  001403                          BEQ     8$              ;YES - SKIP
11047  042506  013737  001636  042520          MOV     BSF24P,6$       ;ELSE CHANGE FOR 24 SECTOR MODE
11048
11049  042514  004437  042160          8$:     JSR     R4,BDSRCK       ;GO CHECK FOR BAD SECTOR THIS ADDRESS
11050  042520  000000              6$:     .WORD   0               ;POINTER TO FILE TO BE CHECKED GOES HERE
11051  042522  012605                          MOV     (SP)+,R5        ;GET # OF BAD SECTORS THIS PACK ADDRESS
11052  042524  001417                          BEQ     9$              ;SKIP IF ZERO
11053
11054  042526  011601              7$:     MOV     (SP),R1         ;GET 1ST BAD SECTOR NUMBER
11055  042530  006301                          ASL     R1              ;MULTIPLY SECTOR NUMBER BY 6 TO
11056  042532  006301                          ASL     R1              ;LOACATE SECTOR TO BE MARKED BAD
11057  042534  061601                          ADD     (SP),R1
11058  042536  062601                          ADD     (SP)+,R1
11059  042540  062701  000002                  ADD     #2,R1           ;ADD 2 FOR 2ND WORD THAT SECTOR
11060  042544  043761  001224  072414          BIC     $TMP1,OBUFF(R1) ; CLEAR BIT FOR BAD SECTOR IN HDR
11061  042552  043761  001224  072416          BIC     $TMP1,OBUFF+2(R1) ; CORRECT THE HVRC BIT
11062  042560  005305                          DEC     R5              ;DECREMENT BAD SECTOR COUNT
11063  042562  001361                          BNE     7$              ;LOOP IF NOT ZERO
11064
11065  042564  032737  100000  001224  9$:     BIT     #BIT15,$TMP1    ;WERE WE DOING BS FACTORY LIST?
11066  042572  001730                          BEQ     10$             ;NO - GO CHECK IF COMPARE MUST BE DONE
11067  042574  012737  040000  001224          MOV     #BIT14,$TMP1    ;ELSE SET BIT TO BE RESET IN BAD HDR
11068  042602  013737  001644  042520          MOV     BSS26P,6$       ;PRESET POINTER FOR 26 SECTOR MODE
11069  042610  032737  010000  001600          BIT     #CFMT,L.CS1     ;TEST IF WE ARE DOING 26 SECTOR MODE
11070  042616  001736                          BEQ     8$              ;YES - SKIP TO START CHECK
```

```
11071  042620  013737  001642  042520          MOV     BSS24P,6$       ;CHANGE POINTER TOR 24 SECTOR MODE
11072  042626  000732                          BR      8$              ;SKIP TO START CHECK
11073
11074                                  ;       START OF COMPARE
11075  042630  012701  000102          11$:    MOV     #102,R1         ;PRESET FOR 102 WORDS OF HEADER
11076  042634  032737  010000  001600          BIT     #CFMT,L.CS1     ;CHECK IF 26 SECTOR MODE
11077  042642  001402                          BEQ     12$             ;YES - SKIP
11078  042644  012701  000074                  MOV     #74,R1          ;CHANGE TO 74 WORDS OF HEADER
11079
11080  042650  012704  070414          12$:    MOV     #IBUFF,R4       ;SET START OF HEADERS TO BE COMPARED
11081  042654  012705  072414                  MOV     #OBUFF,R5       ;SET START OF GOOD HEADERS
11082  042660  005003                          CLR     R3              ;CLEAR COUNTER
11083  042662  032700  040000                  BIT     #BIT14,R0       ;IS THIS A CONTINUATION OF EARLIER COMPARE
11084  042666  001412                          BEQ     13$             ;NO - SKIP
11085  042670  013705  001700          28$:    MOV     DESHLD,R5       ;GET VALUES WHERE PREVIOUS CHECK STOPPED
11086  042674  013704  001702                  MOV     SRCHLD,R4       ;  DESTINATION AND SOURCE
11087  042700  013703  001704                  MOV     WRDNUM,R3       ;  WORD NUMBER IN ERROR
11088  042704  013701  001706                  MOV     WRDCNT,R1       ;  WORD COUNT LEFT IN COMPARE
11089  042710  005701                          TST     R1              ;TEST IF WORD COUNT LEFT - 0
11090  042712  001417                          BEQ     50$             ;YES - EXIT
11091
11092  042714  032700  030000          13$:    BIT     #BIT12 BIT13,R0 ;MEM MANAGE REQUIRED:
11093  042720  001402                          BEQ     25$             ;NO - SKIP
11094  042722  005237  177572                  INC     @#SR0           ;TURN IT ON
11095  042726  022425                  25$:    CMP     (R4)+,(R5)+     ;COMPARE THE WORDS
11096  042730  001012                          BNE     14$             ;SKIP IF NOT EQUAL
11097  042732  005203                          INC     R3              ;BUMP WORD NUMBER IN ERROR
11098  042734  005301                          DEC     R1              ;DEC WORD COUNT LEFT IN COMPARE
11099  042736  001373                          BNE     25$             ;LOOP IF NOT ZERO
11100  042740  032700  030000                  BIT     #BIT12 BIT13,R0 ;MEM MANAGE IN USE?
11101  042744  001402                          BEQ     5 $             ;NO - SKIP
11102  042746  005337  177572                  DEC     @#SR0           ;TURN IT OFF
11103  042752                          50$.
11104  042752  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
11105  042754  000427                          BR      16$
11106
11107                                  ;       ERROR REPORT PREP AND PARAMETER STORAGE FOR CONTINUATION
11108
11109  042756  010537  001700          14$:    MOV     R5,DESHLD       ;STORE DESTINATION
11110  042762  010437  001702                  MOV     R4,SRCHLD       ;       SOURCE
11111  042766  014537  001202                  MOV     -(R5),$REG10    ;LOAD GOOD WORD FOR REPORT
11112  042772  014437  001204                  MOV     -(R4),$REG11    ;       BAD WORD
11113  042776  010337  001206                  MOV     R3,$REG12       ;       WORD NUMBER
11114  043002  005301                          DEC     R1              ;DEC COUNT LEFTFOR CONTINUATION
11115  043004  005203                          INC     R3              ;BUMP BAD WORD NUMBER
11116  043006  010137  001706                  MOV     R1,WRDCNT       ;STORE COUNT LEFT
11117  043012  010337  001704                  MOV     R3,WRDNUM       ;       WORD NUM IN ERROR
11118  043016  032700  030000                  BIT     #BIT12 BIT13,R0 ;MEM MANAGE IS USE?
11119  043022  001402                          BEQ     15$             ;NO - SKIP
11120  043024  005337  177572                  DEC     @#SR0           ;TURN IT OFF
11121
11122  043030                          15$:
11123  043030  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
11124  043032  005724                          TST     (R4)+           ;ERROR RETURN
11125
11126  043034                          16$:
```

```
11127  043034  012605                    MOV     (SP)+,R5        ;;POP STACK INTO R5
11128  043036  012603                    MOV     (SP)+,R3        ;;POP STACK INTO R3
11129  043040  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
11130  043042  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
11131  043044  000204                    RTS     R4
11132
11133                          ;         DATA PATTERN PROCESSING ROUTINE
11134
11135  043046  032700  040000    17$:    BIT     #BIT14,R0       ;CONTINUE WITH COMPARE?
11136  043052  001402                    BEQ     29$             ;NO - SKIP
11137  043054  010446                    MOV     R4,-(SP)        ;STORE RETURN
11138  043056  000704                    BR      28$             ;GO CONTINUE COMPARE
11139
11140  043060  012705  072414    29$:    MOV     #OBUFF,R5       ;GET DESTINATION
11141  043064  012703  070414            MOV     #IBUFF,R3       ;GET SOURCE
11142  043070  032700  030000            BIT     #BIT12!BIT13,R0 ;USE MEM MANAGE?
11143  043074  001412                    BEQ     21$             ;NO - SKIP
11144
11145  043076  012437  172354            MOV     (R4)+,@#KIPAR6  ;LOAD PAR FOR RELOCATION
11146  043102  032700  010000            BIT     #BIT12,R0       ;RELOCATE SOURCE?
11147  043106  001403                    BEQ     20$             ;NO - SKIP
11148  043110  012705  140070            MOV     #140070,R5      ;SET DESTINATION TO USE PAR6 + OFFSET
11149  043114  000402                    BR      21$             ;SKIP
11150  043116  012703  140070    20$:    MOV     #140070,R3      ;SET SOURCE TO USE PAR6 + OFFSET
11151
11152  043122  012401              21$:  MOV     (R4)+,R1        ;STORE COUNT
11153  043124  010446                    MOV     R4,-(SP)        ;STORE RETURN
11154  043126  010304                    MOV     R3,R4           ;PUT IN IBUFF POINTER
11155  043130  005003                    CLR     R3              ;CLEAR R3 FOR WORD NUMBER COUNTER
11156  043132  032700  000077            BIT     #77,R0          ;ANY DATA PATTERN SPECIFIED?
11157  043136  001666                    BEQ     13$             ;NO - GO DO COMPARE
11158
11159                          ;         START OF GENERATION ROUTINE
11160
11161  043140  010537  001700            MOV     R5,DESHLD       ;STORE PARAMETERS FOR COMPARE
11162  043144  010437  001702            MOV     R4,SRCHLD
11163  043150  010337  001704            MOV     R3,WRDNUM
11164  043154  010137  001706            MOV     R1,WRDCNT
11165
11166                          ;         CODE TO GENERATE DATA PATTERN IN AREA POINTED TO BY R5.
11167                          ;         MEMORY MANAGEMENT WILL BE TURNED ON BUT RELOCATION
11168                          ;         WILL NOT OCCUR UNLESS REQUIRED BY SWITCHES
11169
11170  043160  032700  030000            BIT     #BIT12!BIT13,R0 ;MEMORY MANAGEMENT REQUIRED?
11171  043164  001402                    BEQ     33$             ;NO - SKIP
11172  043166  005237  177572            INC     @#SR0           ;TURN IT ON
11173  043172  032700  002000    33$:    BIT     #BIT10,R0       ;GENERATE PATTERN IN IBUFF?
11174  043176  001403                    BEQ     32$             ;NO - SKIP
11175  043200  010446                    MOV     R4,-(SP)        ;ELSE SWAP R4 AND R5
11176  043202  010504                    MOV     R5,R4
11177  043204  012605                    MOV     (SP)+,R5
11178
11179  043206  122700  000001    32$:    CMPB    #1,R0           ;PATTERN 1 (ALL ZEROS)?
11180  043212  001004                    BNE     55$             ;NO - SKIP
11181  043214  005025            30$:    CLR     (R5)+           ;CLEAR WORD IN BUFF
11182  043216  005301                    DEC     R1              ;DEC WORD COUNT
```

```
11183  043220  001375                  BNE     30$             ;LOOP UNTIL WORD COUNT ZERO
11184  043222  000550                  BR      22$             ;EXIT BUILD
11185
11186  043224  122700  000007  55$:    CMPB    #7,R0           ;PATTERN 7 (ALL ONES)?
11187  043230  001005                  BNE     56$             ;NO - SKIP
11188  043232  012725  177777  31$:    MOV     #-1,(R5)+       ;LOAD WORD IN BUFF
11189  043236  005301                  DEC     R1              ;DEC WORD COUNT
11190  043240  001374                  BNE     31$             ;LOOP UN  L WORD COUNT ZERO
11191  043242  000540                  BR      22$             ;EXIT BUILD
11192
11193  043244  122700  000002  56$:    CMPB    #2,R0           ;PATTERN 2 SET UP
11194  043250  001003                  BNE     57$
11195  043252  012703  047144          MOV     #PAT02,R3
11196  043256  000504                  BR      70$
11197
11198  043260  122700  000003  57$:    CMPB    #3,R0           ;PATTERN 3 SET UP
11199  043264  001003                  BNE     58$
11200  043266  012703  047204          MOV     #PAT03,R3
11201  043272  000476                  BR      70$
11202
11203  043274  122700  000004  58$:    CMPB    #4,R0           ;PATTERN 4 SET UP
11204  043300  001003                  BNE     59$
11205  043302  012703  047244          MOV     #PAT04,R3
11206  043306  000470                  BR      70$
11207
11208  043310  122700  000005  59$:    CMPB    #5,R0           ;PATTERN 5 SET UP
11209  043314  001003                  BNE     60$
11210  043316  012703  047304          MOV     #PAT05,R3
11211  043322  000462                  BR      70$
11212
11213  043324  122700  000006  60$:    CMPB    #6,R0           ;PATTERN 6 SET UP
11214  043330  001003                  BNE     61$
11215  043332  012703  047344          MOV     #PAT06,R3
11216  043336  000454                  BR      70$
11217
11218  043340  122700  000010  61$.    CMPB    #10,R0          ;PATTERN 10 SET UP
11219  043344  001003                  BNE     62$
11220  043346  012703  047404          MOV     #PAT10,R3
11221  043352  000446                  BR      70$
11222
11223  043354  122700  000011  62$:    CMPB    #11,R0          ;PATTERN 11 SET UP
11224  043360  001003                  BNE     63$
11225  043362  012703  047444          MOV     #PAT11,R3
11226  043366  000440                  BR      70$
11227
11228  043370  122700  000012  63$:    CMPB    #12,R0          ;PATTERN 12 SET UP
11229  043374  001003                  BNE     64$
11230  043376  012703  047504          MOV     #PAT12,R3
11231  043402  000432                  BR      70$
11232
11233  043404  122700  000013  64$:    CMPB    #13,R0          ;PATTERN 13 SET UP
11234  043410  001003                  BNE     65$
11235  043412  012703  047544          MOV     #PAT13,R3
11236  043416  000424                  BR      70$
11237
11238  043420  122700  000014  65$:    CMPB    #14,R0          ;PATTERN 14 SET UP
```

```
11239  043424  001003                    BNE     66$
11240  043426  012703  047604            MOV     #PAT14,R3
11241  043432  000416                    BR      70$
11242
11243  043434  122700  000015    66$:    CMPB    #15,R0          ;PATTERN 15 SET UP
11244  043440  001003                    BNE     67$
11245  043442  012703  047644            MOV     #PAT15,R3
11246  043446  000410                    BR      70$
11247
11248  043450  122700  000016    67$:    CMPB    #16,R0          ;PATTERN 16 SET UP
11249  043454  001003                    BNE     68$
11250  043456  012703  047704            MOV     #PAT16,R3
11251  043462  000402                    BR      70$
11252
11253  043464  012703  047704    68$:    MOV     #PAT16,R3       ;SET UP FOR 16
11254
11255  043470  032700  004000    70$:    BIT     #BIT11,R0       ;FIRST WORD REPEAT?
11256  043474  001020                    BNE     73$             ;YES - SKIP
11257  043476  010446                    MOV     R4,-(SP)        ;STORE R4
11258  043500  010046                    MOV     R0,-(SP)        ;STORE R0
11259  043502  012700  000020            MOV     #16.,R0         ;PRESET COUNT FOR PATTERN LENGTH
11260  043506  010504                    MOV     R5,R4           ;STORE START OF BUFF
11261
11262  043510  012325            71$:    MOV     (R3)+,(R5)+     ;MOV WORD TO BUFF
11263  043512  005301                    DEC     R1              ;DEC WORD COUNT
11264  043514  001405                    BEQ     74$             ;EXIT IF ZERO
11265  043516  005300                    DEC     R0              ;DEC PAT LENGTH COUNT
11266  043520  001373                    BNE     71$             ;LOOP IF NOT ZERO
11267
11268  043522  012425            72$:    MOV     (R4)+,(R5)+     ;REPEAT PATTERN IN BUFFER
11269  043524  005301                    DEC     R1              ;DEC WORD COUNT
11270  043526  001375                    BNE     72$             ;LOOP UNTIL WORD COUNT ZERO
11271
11272  043530  012600            74$:    MOV     (SP)+,R0        ;RESTORE R0
11273  043532  012604                    MOV     (SP)+,R4        ;RESTORE R4
11274  043534  000403                    BR      22$             ;EXIT BUILD
11275
11276  043536  011325            73$:    MOV     (R3),(R5)+      ;MOV THE SAME WORD INTO BUFFER
11277  043540  005301                    DEC     R1              ;DEC WORD COUNT
11278  043542  001375                    BNE     73$             ;LOOP UNTIL ZERO
11279
11280  043544  032700  030000    22$:    BIT     #BIT12!BIT13,R0 ;MEMORY MANAGEMENT REQUIRED?
11281  043550  001402                    BEQ     34$             ;NO - SKIP
11282  043552  005337  177572            DEC     @#SR0           ;TURN OFF MEM MANAGEMENT
11283  043556  005700            34$:    TST     R0              ;IS COMPARE REQUIRED?
11284  043560  100012                    BPL     23$             ;NO - SKIP
11285  043562  013705  001700            MOV     DESHLD,R5       ;RESTORE COMPARE PARAMETERS
11286  043566  013704  001702            MOV     SRCHLD,R4
11287  043572  013703  001704            MOV     WRDNUM,R3
11288  043576  013701  001706            MOV     WRDCNT,R1
11289  043602  000137  042714            JMP     13$             ;GO START COMPARE
11290  043606                    23$:
11291  043606  012604                    MOV     (SP)+,R4        ;;POP STACK INTO R4
11292  043610  000137  043034            JMP     16$             ;GO TO EXIT
11293
11294                            ;;**************************************************************
```

```
                                        .SBTTL  PHASE LOCK LOOP CLOCK ADJUSTMENT ROUNTINE
                                        ;*      THIS ROUTINE IS ENTERED VIA A START AT LOCATION 220(8). THE
                                        ;*      PROGRAM FIRST RUNS TEST 1, 2, AND 3 TO SET UP THE INTERNAL
                                        ;*      PROGRAM VARIABLES AND THEN JUMPS TO THE CLOCK ADJUST ROUTINE.
                                        ;*      THE ROUTINE SELECTS THE FIRST AVAILABLE DRIVE AND SETS AND
                                        ;*      RESETS DIAGNOSTIC MODE BIT IN MR1. INSTRUCTIONS ON WHERE TO
                                        ;*      SCOPE AND WHAT TO ADJUST ARE TYPED ON THE CONSOLE.
                                        ;*
                                        ;*      THIS ROUTINE WILL LOOP UNTIL THE PROCESSOR IS HALTED.

043614  104401  052662          ADJCLK: TYPE    ,OPR019         ;TYPE ADJUSTMENT INSTRUCTIONS

043620  012762  000040  000010          MOV     #SCLR,RKCS2(R2) ;CLEAR SUBSYSTEM
043626  013762  001626  000010          MOV     DRVNUM,RKCS2(R2)        ;SELECT DRIVE
043634  012762  000001  000000          MOV     #1,RKCS1(R2)
043642  032762  000200  000000  5$:     BIT     #RDY,RKCS1(R2)  ;WAIT FOR READY
043650  001774                          BEQ     5$
043652  032737  100000  001664          BIT     #LCLKPR,OPTFLG  ;TEST IF CLOCK PRESENT
043660  001402                          BEQ     1$              ;NO - SKIP
043662  005077  136022                  CLR     @KWLADD         ;CLEAR INTERRUPT ENABLE

043666  012762  000040  000026  1$:     MOV     #DMD,RKMR1(R2)  ;SET DIAG MODE
043674  012701  000014                  MOV     #12.,R1         ;SET A COUNT
043700  005301                  2$:     DEC     R1              ;DEC COUNT
043702  001376                          BNE     2$              ;LOOP UNTIL ZERO
043704  012762  000000  000026          MOV     #0,RKMR1(R2)    ;CLEAR MR1
043712  012701  000014                  MOV     #12.,R1         ;SET COUNT
043716  005301                  3$:     DEC     R1              ;DEC COUNT
043720  001376                          BNE     3$              ;LOOP UNTIL ZERO
043722  000761                          BR      1$              ;RESTART LOOP

                                        .SBTTL  CONTROLLED HALT SUBROUTINE
                                        ;*      THIS ROUTINE IS ENTERED WHEN A CONTROL C IS TYPED. THE
                                        ;*      SUBSYSTEM IS CLEARED, THE DRIVE IS RECALIBRATED, AND, IF
                                        ;*      NECESSARY, CERTAIN CYLINDERS ARE REFORMATED. THE REFORMATTING
                                        ;*      IS CONTROLLED BY THE LOCATION REFMT WHICH CONTAINS THE ADDRESS
                                        ;*      OF THE CYLINDER TO BE REFORMATTED.

043724  012737  000111  001302  CTRHLT: MOV     #$TN,$TESTN             ;SET UP FOR HALT FAIL

043732  104416                          TSSINIT                 ;CLEAR SUBSYSTEM
043734  104003                          ERROR   3               ;BAD INIT ERROR

043736  113700  001102                  MOVB    $TSTNM,R0       ;GET CURRENT TEST NUMBER
043742  042700  177400                  BIC     #177400,R0      ;CLEAR UNUSED BITS
043746  022700  000003                  CMP     #3,R0           ;TEST IF TEST NUMBER 3
043752  001464                          BEQ     PROGEND         ;GO TO HALT PROG
043754  004437  035674                  JSR     R4,LRLOAD       ;LOAD "L" REGS
043760  000113                          RECAL                   ;RECAL
043762  000000                          0                       ;0 WORDS
043764  000000                          0                       ;0 IS BUFF ADDRESS
043766     000                          .BYTE   0               ;SECTOR 0
043767     000                          .BYTE   0               ;TRACK 0
043770  000000                          0                       ;CYLINDER 0

043772  104417                          TLOADRK                 ;LOAD RK REGS
```

```
11351  043774  104423                        TWAT16                  ;WAIT FOR INTERRUPT
11352  043776  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
11353
11354  044000  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
11355  044002  104004                        ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
11356
11357  044004  005037  001662                CLR     INTSET          ;CLEAR INTERRUPT FLAG
11358  044010  104437                        TWAT8S                  ;WAIT FOR SECOND INTERRUPT
11359  044012  104002                        ERROR   2
11360
11361  044014  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
11362  044016  104004                        ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
11363
11364  044020  104416                        TSSINIT                 ;CLEAR SUBSYSTEM
11365  044022  104003                        ERROR   3               ;BAD INIT ERROR
11366
11367  044024  005737  001676                TST     REFMT           ;TEST IF REFORMAT REQUIRED
11368  044030  001435                        BEQ     PROGEND         ;NO - GO TO HALT
11369  044032  104401  053025                TYPE    ,OPR020         ;TYPE MESSAGE
11370
11371  044036  004437  035674                JSR     R4,LRLOAD       ;LOAD "L" REGS
11372  044042  000127                        WRHEAD                  ;WRHEAD
11373  044044  177676                        -102                    ;-102 WORDS
11374  044046  072414                        OBUFF                   ;OBUFF IS BUFF ADDRESS
11375  044050     000                        .BYTE   0               ;SECTOR 0
11376  044051     000                        .BYTE   0               ;TRACK 0
11377  044052  000312                        312                     ;CYLINDER 312
11378
11379  044054  005737  001676                TST     REFMT           ;TEST IF CYL 0
11380  044060  100002                        BPL     5$              ;NO - SKIP
11381  044062  005037  001614                CLR     L.DCYL          ;ELSE LOAD FOR CYL 0
11382  044066  004437  042260        5$:     JSR     R4,GENCOM       ;GENERATE HEADERS
11383  044072  001200                        1200
11384
11385  044074  104417                        TLOADRK                 ;LOAD RK REGS
11386  044076  104434                        TWAT159                 ;WAIT FOR INTERRUPT
11387  044100  104002                        ERROR   2               ;TO SLOW/NOT COMPLETE ERROR
11388
11389  044102  104421                        TCHKOP                  ;CHECK OPERATION FOR ANY ERRORS
11390  044104  104004                        ERROR   4  ;OR 5, 6, 7  ;REPORT ALL ERRORS
11391
11392  044106  122737  000002  001607        CMPB    #2,L.DT         ;TEST IF TRACK 2 FORMATTED
11393  044114  001403                        BEQ     PROGEND         ;YES - SKIP
11394  044116  105237  001607                INCB    L.DT            ;ELSE BUMP TRACK
11395  044122  000761                        BR      5$              ;DO NEXT TRACK
11396
11397  044124  104401  053121        PROGEND:        TYPE    ,OPR021 ;TYPE HALT MESSAGE
11398  044130  ^12706  001100                MOV     #STACK,SP       ;CLEAR STACK
11399  044134  105037  001103                CLRB    $ERFLG          ;CLEAR ERROR FLAG
11400  044140  005037  001264                CLR     $ESCAPE         ;CLEAR ESCAPE
11401  044144  005737  000042                TST     @#42            ;TEST IF MONITOR PRESENT
11402  044150  001404                        BEQ     10$             ;NO - SKIP
11403  044152  005037  032020                CLR     $EOPCT          ;SET FOR END OF PROGRAM
11404  044156  000137  031772                JMP     $EOP            ;GO TO END OF PASS
11405
11406  044162  000000        10$:            HALT                    ;HALT PROGRAM
```

```
11407  044164  000137  001776                  JMP     START1          ;GO TO RESTART IF CONTINUE
11408
11409                                  .SBTTL  HALT FAIL ROUTINE
11410                                  ;*      THIS ROUTINE IS ENTERED IF A HARDWARE ERROP IS DETECTED WHEN
11411                                  ;*      THE CARTRIDGE IS BEING REFORMATTED PRIOR TO HALT.
11412  044170  000240                  ABTFAIL:        NOP
11413  044172  104401  053162                  TYPE    ,OPR022         ;TYPE HALT FAIL MESSAGE
11414  044176  000137  044124                  JMP     PROGEND         ;GO STOP PROGRAM
11415                                  .SBTTL  TYPE ROUTINE
11416
11417                                  ;;*****************************************************************
11418                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
11419                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
11420                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
11421                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
11422                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
11423                                  ;*
11424                                  ;*CALL:
11425                                  ;*1) USING A TRAP INSTRUCTION
11426                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
11427                                  ;*OR
11428                                  ;*      TYPE
11429                                  ;*      MESADR
11430                                  ;*
11431
11432  044202  105737  001157  $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
11433  044206  100002                  BPL     1$              ;;BR IF YES
11434  044210  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
11435  044212  000430                  BR      3$              ;;LEAVE
11436  044214  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
11437  044216  017600  000002          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
11438  044222  122737  000001  001316  CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
11439  044230  001011                  BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
11440  044232  132737  000100  001317  BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
11441  044240  001405                  BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
11442  044242  010037  044252          MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
11443  044246  004737  033354          JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
11444  044252  000000          61$:    .WORD   0               ;;MESSAGE ADDRESS
11445  044254  132737  000040  001317  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
11446  044262  001003                  BNE     60$             ;;YES,SKIP TYPE OUT
11447  044264  112046          2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
11448  044266  001005                  BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
11449  044270  005726                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
11450  044272  012600          60$:    MOV     (SP)+,R0        ;;RESTORE R0
11451  044274  062716  000002  3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
11452  044300  000002                  RTI                     ;;RETURN
11453  044302  122716  000011  4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
11454  044306  001430                  BEQ     8$
11455  044310  122716  000200          CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
11456  044314  001006                  BNE     5$
11457  044316  005726                  TST     (SP)+           ;;POP  <CR><LF> EQUIV
11458  044320  104401                  TYPE                    ;;TYPE A CR AND LF
11459  044322  001273                  $CRLF
11460  044324  105037  044460          CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
11461  044330  000755                  BR      2$              ;;GET NEXT CHARACTER
11462  044332  004737  044414  5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
```

```
11463  044336  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
11464  044342  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
11465  044344  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
11466                                                                  ;;AND THE NULL CHAR.
11467  044350  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
11468  044354  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
11469  044356  004737  044414                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
11470  044362  105337  044460                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
11471  044366  000770                          BR      7$              ;;LOOP
11472
11473                                          ;HORIZONTAL TAB PROCESSOR
11474
11475  044370  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
11476  044374  004737  044414          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
11477  044400  132737  000007  044460          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
11478  044406  001372                          BNE     9$              ;;TAB STOP
11479  044410  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
11480  044412  000724                          BR      2$              ;;GET NEXT CHARACTER
11481  044414  105777  134530          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
11482  044420  100375                          BPL     $TYPEC
11483  044422  116677  000002  134522          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
11484  044430  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
11485  044436  001003                          BNE     1$              ;;BRANCH IF NO
11486  044440  105037  044460                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
11487  044444  000406                          BR      $TYPEX          ;;EXIT
11488  044446  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
11489  044454  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
11490  044456  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
11491  044460  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
11492  044462  000207          $TYPEX: RTS     PC
11493
11494                                          .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
11495
11496                                          ;;**********************************************************************
11497                                          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
11498                                          ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
11499                                          ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
11500                                          ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
11501                                          ;*REPLACED WITH SPACES.
11502                                          ;*CALL:
11503                                          ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
11504                                          ;*      TYPDS                   ;;GO TO THE ROUTINE
11505
11506  044464                          $TYPDS:
11507  044464  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11508  044466  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11509  044470  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11510  044472  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11511  044474  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
11512  044476  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
11513  044502  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
11514  044506  100004                          BPL     1$              ;;BR IF INPUT IS POS.
11515  044510  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
11516  044512  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
11517  044520  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
11518  044522  012703  044700                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
```

```
11519  044526  112723  000040            MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
11520  044532  005002                2$: CLR     R2              ;;CLEAR THE BCD NUMBER
11521  044534  016001  044670            MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
11522  044540  160105                3$: SUB     R1,R5           ;;FORM THIS BCD DIGIT
11523  044542  002402                    BLT     4$              ;;BR IF DONE
11524  044544  005202                    INC     R2              ;;INCREASE THE BCD DIGIT BY 1
11525  044546  000774                    BR      3$
11526  044550  060105                4$: ADD     R1,R5           ;;ADD BACK THE CONSTANT
11527  044552  005702                    TST     R2              ;;CHECK IF BCD DIGIT=0
11528  044554  001002                    BNE     5$              ;;FALL THROUGH IF 0
11529  044556  105716                    TSTB    (SP)            ;;STILL DOING LEADING 0'S?
11530  044560  100407                    BMI     7$              ;;BR IF YES
11531  044562  106316                5$: ASLB    (SP)            ;;MSD?
11532  044564  103003                    BCC     6$              ;;BR IF NO
11533  044566  116663  000001  177777    MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
11534  044574  052702  000060        6$: BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
11535  044600  052702  000040        7$: BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
11536  044604  110223                    MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
11537  044606  005720                    TST     (R0)+           ;;JUST INCREMENTING
11538  044610  020027  000010            CMP     R0,#10          ;;CHECK THE TABLE INDEX
11539  044614  002746                    BLT     2$              ;;GO DO THE NEXT DIGIT
11540  044616  003002                    BGT     8$              ;;GO TO EXIT
11541  044620  010502                    MOV     R5,R2           ;;GET THE LSD
11542  044622  000764                    BR      6$              ;;GO CHANGE TO ASCII
11543  044624  105726                8$: TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
11544  044626  100003                    BPL     9$              ;;BR IF NO
11545  044630  116663  177777  177776    MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
11546  044636  105013                9$: CLRB    (R3)            ;;SET THE TERMINATOR
11547  044640  012605                    MOV     (SP)+,R5        ;;POP STACK INTO R5
11548  044642  012603                    MOV     (SP)+,R3        ;;POP STACK INTO R3
11549  044644  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
11550  044646  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
11551  044650  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
11552  044652  104401  044700            TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
11553  044656  016666  000002  000004    MOV     2(SP),4(SP)     ;;ADJUST THE STACK
11554  044664  012616                    MOV     (SP)+,(SP)
11555  044666  000002                    RTI                     ;;RETURN TO USER
11556  044670  023420            $DTBL:  10000.
11557  044672  001750                    1000.
11558  044674  000144                    100.
11559  044676  000012                    10.
11560  044700  000004            $DBLK:  .BLKW   4
11561                             .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
11562
11563                             ;;**************************************************************
11564                             ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
11565                             ;*OCTAL (ASCII) NUMBER AND TYPE IT.
11566                             ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
11567                             ;*CALL:
11568                             ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
11569                             ;*      TYPOS                   ;;CALL FOR TYPEOUT
11570                             ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
11571                             ;*      .BYTE   M               ;;M=1 OR 0
11572                             ;*                                      ;;1=TYPE LEADING ZEROS
11573                             ;*                                      ;;0=SUPPRESS LEADING ZEROS
11574                             ;*
```

```
11575                                   ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
11576                                   ;*$TYPOS OR $TYPOC
11577                                   ;*CALL:
11578                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
11579                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
11580                                   ;*
11581                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
11582                                   ;*CALL:
11583                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
11584                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
11585
11586  044710  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
11587  044714  116637  000001  045133          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
11588  044722  112637  045135                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
11589  044726  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
11590  044732  000406                          BR      $TYPON
11591  044734  112737  000001  045133  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
11592  044742  112737  000006  045135          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
11593  044750  112737  000005  045132  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
11594  044756  010346                          MOV     R3,-(SP)        ;;SAVE R3
11595  044760  010446                          MOV     R4,-(SP)        ;;SAVE R4
11596  044762  010546                          MOV     R5,-(SP)        ;;SAVE R5
11597  044764  113704  045135                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
11598  044770  005404                          NEG     R4
11599  044772  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
11600  044776  110437  045134                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
11601  045002  113704  045133                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
11602  045006  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
11603  045012  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
11604  045014  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
11605  045016  000404                          BR      3$              ;;GO DO MSB
11606  045020  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
11607  045022  006105                          ROL     R5
11608  045024  006105                          ROL     R5
11609  045026  010503                          MOV     R5,R3
11610  045030  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
11611  045032  105337  045134                  DECB    $OMODE          ;;TYPE THIS DIGIT?
11612  045036  100016                          BPL     7$              ;;BR IF NO
11613  045040  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
11614  045044  001002                          BNE     4$              ;;TEST FOR 0
11615  045046  005704                          TST     R4              ;;SUPPRESS THIS 0?
11616  045050  001403                          BEQ     5$              ;;BR IF YES
11617  045052  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
11618  045054  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
11619  045060  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
11620  045064  110337  045130                  MOVB    R3,8$           ;;SAVE FOR TYPING
11621  045070  104401  045130                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
11622  045074  105337  045132          7$:     DECB    $OCNT           ;;COUNT BY 1
11623  045100  003347                          BGT     2$              ;;BR IF MORE TO DO
11624  045102  002402                          BLT     6$              ;;BR IF DONE
11625  045104  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
11626  045106  000744                          BR      2$              ;;GO DO THE LAST DIGIT
11627  045110  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
11628  045112  012604                          MOV     (SP)+,R4        ;;RESTORE R4
11629  045114  012603                          MOV     (SP)+,R3        ;;RESTORE R3
11630  045116  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
```

```
11631  045124  012616                  MOV     (SP)+,(SP)
11632  045126  000002                  RTI                     ;;RETURN
11633  045130     000          8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
11634  045131     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
11635  045132     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
11636  045133     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
11637  045134  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
11638                           .SBTTL  TTY INPUT ROUTINE
11639
11640                           ;;*****************************************************************
11641                           .ENABL  LSB
11642  045136  000000          $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
11643  045140  000000          $TKQIN: .WORD   0               ;;INPUT POINTER
11644  045142  000000          $TKQOUT: .WORD  0               ;;OUTPUT POINTER
11645  045144  000001          $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
11646          045145          $TKQEND=.
11647          045146          .EVEN
11648
11649                           ;*TK INITIALIZE ROUTINE
11650                           ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
11651                           ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
11652                           ;
11653                           ;*CALL:
11654                           ;*      JSR     PC,$TKINT
11655                           ;*      RETURN
11656                           ;
11657  045146  005037  045136  $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
11658  045152  012737  045144  045140  MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
11659  045160  013737  045140  045142  MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
11660  045166  012737  045216  000060  MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
11661  045174  012737  000200  000062  MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
11662  045202  005777  133740          TST     @$TKB           ;;CLEAR DONE FLAG
11663  045206  012777  000100  133730  MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
11664  045214  000207                  RTS     PC              ;;RETURN TO CALLER
11665
11666                           ;*TK SERVICE ROUTINE
11667                           ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
11668                           ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
11669                           ;*IT IN THE QUEUE.
11670                           ;*IF THE CHARACTER IS A "CONTROL-C" (^C) $TKINT IS CALLED AND
11671                           ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (CTRHLT)
11672                           ;
11673  045216  117746  133724  $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
11674  045222  042716  177600          BIC     #^C177,(SP)     ;;STRIP THE JUNK
11675  045226  021627  000003          CMP     (SP),#3         ;;IS IT A CONTROL C?
11676  045232  001007                  BNE     1$              ;;BRANCH IF NO
11677  045234  104401  046332          TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
11678  045240  004737  045146          JSR     PC,$TKINT       ;;INIT THE KEYBOARD
11679  045244  005726                  TST     (SP)+           ;;CLEAN UP STACK
11680  045246  000137  043724          JMP     CTRHLT          ;;CONTROL C RESTART
11681  045252  021627  000007  1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
11682  045256  001004                  BNE     2$              ;;BRANCH IF NO
11683  045260  022737  000176  001140  CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
11684  045266  001500                  BEQ     6$              ;;GO TO SWR CHANGE
11685
11686  045270          2$:
```

```
11687  045270  022737  000001  045136          CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
11688  045276  001004                          BNE     3$              ;;BRANCH IF NO
11689  045300  104401  001266                  TYPE    ,$BELL          ;;RING THE TTY BELL
11690  045304  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
11691  045306  000451                          BR      5$              ;;EXIT
11692  045310  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
11693  045314  001021                          BNE     32$             ;;BRANCH IF NO
11694  045316  005077  133622                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
11695  045322  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
11696  045324  105777  133614          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
11697  045330  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
11698  045332  117746  133610                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
11699  045336  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
11700  045342  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
11701  045346  001366                          BNE     31$             ;;BRANCH IF NO
11702  045350  012777  000100  133566          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERPUPTS
11703  045356  000002                          RTI                     ;;RETURN
11704  045360  005237  045136          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
11705  045364  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
11706  045370  002405                          BLT     4$              ;;BRANCH IF YES
11707  045372  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
11708  045376  003002                          BGT     4$              ;;BRANCH IF YES
11709  045400  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
11710  045404  112677  177530          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
11711  045410  005237  045140                  INC     $TKQIN          ;;UPDATE THE POINTER
11712  045414  023727  045140  045145          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
11713  045422  001003                          BNE     5$              ;;BRANCH IF NO
11714  045424  012737  045144  045140          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
11715  045432  000002                  5$:     RTI                     ;;RETURN
11716
11717                                          ;;**********************************************************************
11718                                          ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
11719                                          ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
11720                                          ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
11721                                          ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
11722  045434  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
11723  045442  001124                          BNE     15$             ;;EXIT IF NOT
11724  045444  105777  133474                  TSTB    @$TKS           ;;IS A CHAR WAITING?
11725  045450  100121                          BPL     15$             ;;IF NOT, EXIT
11726  045452  117746  133470                  MOVB    @$TKB,-(SP)     ;;YES
11727  045456  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
11728  045462  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
11729  045466  001300                          BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
11730                                                                  ;;AND EXIT
11731
11732                                          ;;**********************************************************************
11733                                          ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
11734                                          ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
11735                                          ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
11736  045470  123727  001134  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
11737  045476  001674                          BEQ     2$              ;;BRANCH IF YES
11738  045500  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
11739  045502  004737  045146                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
11740  045506  005077  133432                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
11741  045512  112737  000001  001135          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
11742
```

```
11743  045520  104401  046344                 TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
11744  045524  104401  046351          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
11745  045530  013746  000176                 MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
11746  045534  104402                         TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
11747  045536  104401  046362                 TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
11748  045542  005046                 19$:    CLR     -(SP)           ;;CLEAR COUNTER
11749  045544  005046                         CLR     -(SP)           ;;THE NEW SWR
11750  045546  105777  133372          7$:    TSTB    @$TKS           ;;CHAR THERE?
11751  045552  100375                         BPL     7$              ;;IF NOT TRY AGAIN
11752
11753  045554  117746  133366                 MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
11754  045560  042716  177600                 BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
11755
11756  045564  021627  000003                 CMP     (SP),#3         ;;IS IT A CONTROL-C?
11757  045570  001015                         BNE     9$              ;;BRANCH IF NOT
11758  045572  104401  046332                 TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
11759  045576  062706  000006                 ADD     #6,SP           ;;CLEAN UP STACK
11760  045602  123727  001135  000001         CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
11761  045610  001003                         BNE     8$              ;;BRANCH IF NO
11762  045612  012777  000100  133324         MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
11763  045620  000137  043724          8$:    JMP     CTRHLT          ;;CONTROL-C RESTART
11764
11765
11766  045624  021627  000025          9$:    CMP     (SP),#25        ;;IS IT A CONTROL-U?
11767  045630  001005                         BNE     10$             ;;BRANCH IF NOT
11768  045632  104401  046337                 TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
11769  045636  062706  000006          20$:   ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
11770  045642  000737                         BR      19$             ;;LET'S TRY IT AGAIN
11771
11772
11773  045644  021627  000015          10$:   CMP     (SP),#15        ;;IS IT A <CR>?
11774  045650  001022                         BNE     16$             ;;BRANCH IF NO
11775  045652  005766  000004                 TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
11776  045656  001403                         BEQ     11$             ;;BRANCH IF YES
11777  045660  016677  000002  133252         MOV     2(SP),@SWR      ;;SAVE NEW SWR
11778  045666  062706  000006          11$:   ADD     #6,SP           ;;CLEAR UP STACK
11779  045672  104401  001273          14$:   TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
11780  045676  123727  001135  000001         CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
11781  045704  001003                         BNE     15$             ;;BRANCH IF NOT
11782  045706  012777  000100  133230         MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
11783  045714  000002          15$:           RTI                     ;;RETURN
11784  045716  004737  044414          16$:   JSR     PC,$TYPEC       ;;ECHO CHAR
11785  045722  021627  000060                 CMP     (SP),#60        ;;CHAR < 0?
11786  045726  002420                         BLT     18$             ;;BRANCH IF YES
11787  045730  021627  000067                 CMP     (SP),#67        ;;CHAR > 7?
11788  045734  003015                         BGT     18$             ;;BRANCH IF YES
11789  045736  042726  000060                 BIC     #60,(SP)+       ;;STRIP-OFF ASCII
11790  045742  005766  000002                 TST     2(SP)           ;;IS THIS THE FIRST CHAR
11791  045746  001403                         BEQ     17$             ;;BRANCH IF YES
11792  045750  006316                         ASL     (SP)            ;;NO, SHIFT PRESENT
11793  045752  006316                         ASL     (SP)            ;;    CHAR OVER TO MAKE
11794  045754  006316                         ASL     (SP)            ;;    ROOM FOR NEW ONE.
11795  045756  005266  000002          17$:   INC     2(SP)           ;;KEEP COUNT OF CHAR
11796  045762  056616  177776                 BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
11797  045766  000667                         BR      7$              ;;GET THE NEXT ONE
11798  045770  104401  001272          18$:   TYPE    ,$QUES          ;;TYPE ?<CR><LF>
```

```
11799  045774  000720                          BR      20$             ;;SIMULATE CONTROL-U
11800                                  .DSABL  LSB
11801
11802
11803                                  ;;*****************************************************************
11804                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
11805                                  ;*CALL:
11806                                  ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
11807                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
11808                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
11809                                  ;
11810
11811  045776  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
11812  046000  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
11813  046006  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
11814  046012  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
11815  046014  012746  046022                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
11816  046020  000002                          RTI                     ;;POP NEW PC AND PS
11817  046022                          64$:
11818  046022  005737  045136          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
11819  046026  001775                          BEQ     1$
11820  046030  005337  045136                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
11821  046034  117766  177102  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
11822  046042  005237  045142                  INC     $TKQOUT         ;;UPDATE THE POINTER
11823  046046  023727  045142  045145          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
11824  046054  001003                          BNE     2$              ;;BRANCH IF NO
11825  046056  012737  045144  045142          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
11826  046064  000002                  2$:     RTI                     ;;RETURN
11827                                  ;;*****************************************************************
11828                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
11829                                  ;*CALL:
11830                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
11831                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
11832                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
11833
11834  046066  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
11835  046070  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
11836  046072  012703  046322          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
11837  046076  022703  046332          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
11838  046102  101456                          BLOS    4$              ;;BR IF YES
11839  046104  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
11840  046106  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
11841  046110  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
11842  046114  001022                          BNE     5$              ;;BR IF NO
11843  046116  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
11844  046120  001007                          BNE     6$              ;;BR IF NO
11845  046122  112737  000134  046320          MOVB    #'\,9$          ;;TYPE A BACK SLASH
11846  046130  104401  046320                  TYPE    ,9$
11847  046134  012716  177777                  MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
11848  046140  005303                  6$:     DEC     R3              ;;BACKUP BY ONE
11849  046142  020327  046322                  CMP     R3,#$TTYIN      ;;STACK EMPTY?
11850  046146  103434                          BLO     4$              ;;BR IF YES
11851  046150  111337  046320                  MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
11852  046154  104401  046320                  TYPE    ,9$             ;;GO TYPE
11853  046160  000746                          BR      2$              ;;GO READ ANOTHER CHAR.
11854  046162  005716                  5$:     TST     (SP)            ;;RUBOUT KEY SET?
```

```
11855  046164  001406                        BEQ     7$              ;;BR IF NO
11856  046166  112737  000134  046320        MOVB    #'\,9$          ;;TYPE A BACK SLASH
11857  046174  104401  046320                TYPE    ,9$
11858  046200  005016                        CLR     (SP)            ;;CLEAR THE RUBOUT KEY
11859  046202  122713  000025        7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
11860  046206  001003                        BNE     8$              ;;BR IF NO
11861  046210  104401  046337                TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
11862  046214  000726                        BR      1$              ;;GO START OVER
11863  046216  122713  000022        8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "^R"?
11864  046222  001011                        BNE     3$              ;;BRANCH IF NO
11865  046224  105013                        CLRB    (R3)            ;;CLEAR THE CHARACTER
11866  046226  104401  001273                TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
11867  046232  104401  046322                TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
11868  046236  000717                        BR      2$              ;;GO PICKUP ANOTHER CHACTER
11869  046240  104401  001272        4$:     TYPE    ,$QUES          ;;TYPE A '?'
11870  046244  000712                        BR      1$              ;;CLEAR THE BUFFER AND LOOP
11871  046246  111337  046320        3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
11872  046252  104401  046320                TYPE    ,9$
11873  046256  122723  000015                CMPB    #15,(R3)+       ;;CHECK FOR RETURN
11874  046262  001305                        BNE     2$              ;;LOOP IF NOT RETURN
11875  046264  105063  177777                CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
11876  046270  104401  001274                TYPE    ,$LF            ;;TYPE A LINE FEED
11877  046274  005726                        TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
11878  046276  012603                        MOV     (SP)+,R3        ;;RESTORE R3
11879  046300  011646                        MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
11880  046302  016666  000004  000002        MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
11881  046310  012766  046322  000004        MOV     #$TTYIN,4(SP)
11882  046316  000002                        RTI                     ;;RETURN
11883  046320     000              9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
11884  046321     000                      .BYTE   0               ;;TERMINATOR
11885  046322  000010             $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
11886  046332  041536  005015     000  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
11887  046337     136  006525  000012  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
11888  046344  043536  005015     000  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
11889  046351     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
11890  046356  036440  000040
11891  046362  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
11892  046370  020075     000
11893          046374                  .EVEN
11894                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
11895
11896                                  ;;**************************************************************
11897                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
11898                                  ;*CHANGE IT TO BINARY.
11899                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
11900                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
11901                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
11902                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
11903                                  ;*CALL:
11904                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
11905                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
11906                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
11907
11908  046374  011646             $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
11909  046376  016666  000004  000002        MOV     4(SP),2(SP)     ;;INPUT NUMBER
11910  046404  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
```

```
11911  046406  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11912  046410  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11913  046412  104411          1$:     RDLIN                   ;;READ AN ASCIZ LINE
11914  046414  012600                  MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
11915  046416  010037  046522          MOV     R0,5$           ;;AND SAVE IT
11916  046422  005001                  CLR     R1              ;;CLEAR DATA WORD
11917  046424  005002                  CLR     R2
11918  046426  112046          2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
11919  046430  001420                  BEQ     3$              ;;IF ZERO GET OUT
11920  046432  122716  000060          CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
11921  046436  003026                  BGT     4$              ;;IS AN OCTAL DIGIT
11922  046440  122716  000067          CMPB    #'7,(SP)
11923  046444  002423                  BLT     4$
11924  046446  006301                  ASL     R1              ;;*2
11925  046450  006102                  ROL     R2
11926  046452  006301                  ASL     R1              ;;*4
11927  046454  006102                  ROL     R2
11928  046456  006301                  ASL     R1              ;;*8
11929  046460  006102                  ROL     R2
11930  046462  042716  177770          BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
11931  046466  062601                  ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
11932  046470  000756                  BR      2$              ;;LOOP
11933  046472  005726          3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
11934  046474  010166  000012          MOV     R1,12(SP)       ;;SAVE THE RESULT
11935  046500  010237  046532          MOV     R2,$HIOCT
11936  046504  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
11937  046506  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
11938  046510  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
11939  046512  000002                  RTI                     ;;RETURN
11940  046514  005726          4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
11941  046516  105010                  CLRB    (R0)            ;;SET A TERMINATOR
11942  046520  104401                  TYPE                    ;;TYPE UP THRU THE BAD CHAR.
11943  046522  000000          5$:     .WORD   0
11944  046524  104401  001272          TYPE    ,$QUES          ;;''?'' ''CR'' & ''LF''
11945  046530  000730                  BR      1$              ;;TRY AGAIN
11946  046532  000000          $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
11947                          .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
11948
11949                          ;;**************************************************************
11950                          ;*SAVE R0-R5
11951                          ;*CALL:
11952                          ;*      SAVREG
11953                          ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
11954                          ;*
11955                          ;*TOP---(+16)
11956                          ;* +2---(+18)
11957                          ;* +4---R5
11958                          ;* +6---R4
11959                          ;* +8---R3
11960                          ;*+10---R2
11961                          ;*+12---R1
11962                          ;*+14---R0
11963
11964  046534          $SAVREG:
11965  046534  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11966  046536  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
```

```
11967  046540  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11968  046542  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11969  046544  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
11970  046546  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
11971  046550  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
11972  046554  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
11973  046560  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
11974  046564  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
11975  046570  000002                  RTI
11976
11977                                ;*RESTORE R0-R5
11978                                ;*CALL:
11979                                ;*      RESREG
11980  046572                        $RESREG:
11981  046572  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
11982  046576  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
11983  046602  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
11984  046606  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
11985  046612  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
11986  046614  012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
11987  046616  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
11988  046620  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
11989  046622  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
11990  046624  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
11991  046626  000002                  RTI
11992                                .SBTTL  POWER DOWN AND UP ROUTINES
11993
11994                                ;;**************************************************************************
11995                                ;POWER DOWN ROUTINE
11996  046630  012737  046770  000024 $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
11997  046636  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
11998  046644  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11999  046646  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
12000  046650  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
12001  046652  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
12002  046654  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
12003  046656  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
12004  046660  017746  132254          MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
12005  046664  010637  046774          MOV     SP,$SAVR6       ;;SAVE SP
12006  046670  012737  046702  000024  MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
12007  046676  000000                  HALT
12008  046700  000776                  BR      .-2             ;;HANG UP
12009
12010                                ;;**************************************************************************
12011                                ;POWER UP ROUTINE
12012  046702  012737  046770  000024 $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
12013  046710  013706  046774          MOV     $SAVR6,SP       ;;GET SP
12014  046714  005037  046774          CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
12015  046720  005237  046774          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
12016  046724  001375                  BNE     1$              ;;OF  WORD
12017  046726  012677  132206          MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
12018  046732  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
12019  046734  012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
12020  046736  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
12021  046740  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
12022  046742  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
```

```
12023  046744  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
12024  046746  012737  046630  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
12025  046754  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
12026  046762  104401                          TYPE                    ;REPORT THE POWER FAILURE
12027  046764  046776                $PWRMG:   .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
12028  046766  000002                          RTI
12029  046770  000000                $ILLUP:   HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
12030  046772  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
12031  046774  000000                $SAVR6:   0                       ;;PUT THE SP HERE
12032  046776  005015  047520  042527  $POWER:   .ASCIZ  <15><12>"POWER"
12033  047004  000122
12034                                          .EVEN
12035                                .SBTTL    TRAP DECODER
12036
12037                                ;;*********************************************************************
12038                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
12039                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
12040                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
12041                                ;*GO TO THAT ROUTINE.
12042
12043  047006  010046                $TRAP:    MOV     R0,-(SP)        ;;SAVE R0
12044  047010  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
12045  047014  005740                          TST     -(R0)           ;;BACKUP BY 2
12046  047016  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
12047  047020  006300                          ASL     R0              ;;POSITION FOR INDEXING
12048  047022  016000  047042                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
12049  047026  000200                          RTS     R0              ;;GO TO ROUTINE
12050
12051
12052                                ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
12053
12054  047030  011646                $TRAP2:   MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
12055  047032  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
12056  047040  000002                          RTI                     ;;RESTORE THE PSW
12057
12058                                .SBTTL    TRAP TABLE
12059
12060                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
12061                                ;*BY THE "TRAP" INSTRUCTION.
12062
12063                                ;       ROUTINE
12064                                ;       -------
12065  047042  047030                $TRPAD:   .WORD   $TRAP2
12066  047044  044202                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
12067  047046  044734                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
12068  047050  044710                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
12069  047052  044750                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
12070  047054  044464                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
12071
12072  047056  045524                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
12073
12074  047060  045434                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
12075  047062  045776                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
12076  047064  046066                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
12077  047066  046374                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
12078  047070  046534                          $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
```

```
12079  047072  046572                  $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
12080  047074  035222                  SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERROR
12081  047076  037176                  SSINIT  ;;CALL=TSSINIT  TRAP+16(104416) INITIALIZE SUBSYSTEM
12082  047100  035734                  LOADRK  ;;CALL=TLOADRK  TRAP+17(104417) LOAD RK611 FOR OPERATION
12083  047102  036164                  GETRK   ;;CALL=TGETRK   TRAP+20(104420) GET RK611 REGISTERS
12084  047104  040162                  CHKOP   ;;CALL=TCHKOP   TRAP+21(104421) CHECK OPREATION FOR ANY ERRORS
12085  047106  040130                  CHKWE   ;;CALL=TCHKWE   TRAP+22(104422) CHECK OPERATION FOR EXPECTED ERRORS
12086  047110  035530                  IWAT16  ;;CALL=TWAT16   TRAP+23(104423) WAIT 16 MS
12087  047112  035520                  IWAT32  ;;CALL=TWAT32   TRAP+24(104424) WAIT 32 MS
12088  047114  035510                  IWAT48  ;;CALL=TWAT48   TRAP+25(104425) WAIT 48 MS
12089  047116  035500                  IWAT64  ;;CALL=TWAT64   TRAP+26(104426) WAIT 64 MS
12090  047120  035470                  IWAT80  ;;CALL=TWAT80   TRAP+27(104427) WAIT 80 MS
12091  047122  035460                  IWAT96  ;;CALL=TWAT96   TRAP+30(104430) WAIT 96 MS
12092  047124  035450                  ! AT112 ;;CALL=TWAT112  TRAP+31(104431) WAIT 112 MS
12093  047126  035440                  IWAT128 ;;CALL=TWAT128  TRAP+32(104432) WAIT 128 MS
12094  047130  035430                  IWAT144 ;;CALL=TWAT144  TRAP+33(104433) WAIT 144 MS
12095  047132  035420                  IWAT159 ;;CALL=TWAT159  TRAP+34(104434) WAIT 160 MS
12096  047134  035410                  IWAT1S  ;;CALL=TWAT1S   TRAP+35(104435) WAIT FOR 1 SECOND
12097  047136  035400                  IWAT2S  ;;CALL=TWAT2S   TRAP+36(104436) WAIT FOR 2 SECONDS
12098  047140  035360                  IWAT8S  ;;CALL=TWAT8S   TRAP+37(104437) WAIT FOR 8 SECONDS
12099  047142  035370                  IWAT1M  ;;CALL=TWAT1M   TRAP+40(104440) WAIT FOR 1 MIN
12100          000102          $TERM=.-$TRPAD
```

```
12101                                   .SBTTL  DATA PATTERNS
12102                                   ;DATA PATTERN 1
12103                                   ;       PATTERN IS ALL ZEROS
12104
12105                                   ;DATA PATTERN 2
12106                                   ;       HI-LO FREQ. MIX
12107  047144                           PAT02:
12108  047144  177777                           177777
12109  047146  177777                           177777
12110  047150  177777                           177777
12111  047152  052525                           052525
12112  047154  052525                           052525
12113  047156  052525                           052525
12114  047160  177777                           177777
12115  047162  177777                           177777
12116  047164  052525                           052525
12117  047166  052525                           052525
12118  047170  177777                           177777
12119  047172  052525                           052525
12120  047174  177252                           177252
12121  047176  177252                           177252
12122  047200  172765                           172765
12123  047202  172765                           172765
12124
12125                                   ;DATA PATERN 3
12126                                   ;       HI FREQ. PHASE MIX
12127  047204                           PAT03:
12128  047204  000000                           000000
12129  047206  000000                           000000
12130  047210  000000                           000000
12131  047212  177777                           177777
12132  047214  177777                           177777
12133  047216  177777                           177777
12134  047220  000000                           000000
12135  047222  000000                           000000
12136  047224  177777                           177777
12137  047226  177777                           177777
12138  047230  000000                           000000
12139  047232  177777                           177777
12140  047234  000000                           000000
12141  047236  177777                           177777
12142  047240  000000                           000000
12143  047242  177777                           177777
12144
12145                                   ;DATA PATTERN 4
12146                                   ;       LO FREQ. PHASE MIX
12147  047244                           PAT04:
12148  047244  052525                           052525
12149  047246  052525                           052525
12150  047250  052525                           052525
12151  047252  125252                           125252
12152  047254  125252                           125252
12153  047256  125252                           125252
12154  047260  052525                           052525
12155  047262  052525                           052525
12156  047264  125252                           125252
```

```
12157  047266  125252                    125252
12158  047270  052525                    052525
12159  047272  125252                    125252
12160  047274  052525                    052525
12161  047276  125252                    125252
12162  047300  052525                    052525
12163  047302  125252                    125252
12164
12165                                  ;DATA PATTERN 5
12166                                  ;       MAX PRECOMP. PHASE MIX
12167  047304                          PAT05:
12168  047304  133333                    133333
12169  047306  066666                    066666
12170  047310  155555                    155555
12171  047312  155555                    155555
12172  047314  133333                    133333
12173  047316  066666                    066666
12174  047320  066666                    066666
12175  047322  155555                    155555
12176  047324  155555                    155555
12177  047326  133333                    133333
12178  047330  133333                    133333
12179  047332  133333                    133333
12180  047334  133333                    133333
12181  047336  133333                    133333
12182  047340  133333                    133333
12183  047342  133333                    133333
12184
12185                                  ;DATA PATTERN 6
12186                                  ;       ROTATING BOUNDARY PULSF PRECOMP.
12187  047344                          PAT06:
12188  047344  121105                    121105
12189  047346  150442                    150442
12190  047350  064221                    064221
12191  047352  132110                    132110
12192  047354  055044                    055044
12193  047356  026422                    026422
12194  047360  013211                    013211
12195  047362  105504                    105504
12196  047364  042642                    042642
12197  047366  021321                    021321
12198  047370  110550                    110550
12199  047372  044264                    044264
12200  047374  022132                    022132
12201  047376  011055                    011055
12202  047400  104426                    104426
12203  047402  042213                    042213
12204
12205                                  ;DATA PATTERN 7
12206                                  ;       FIELD OF ALL ONES
12207
12208                                  ;DATA PATTERN 10
12209                                  ;       ROTATING CELL PULSE PRECOMP.
12210  047404                          PAT10:
12211  047404  026455                    026455
12212  047406  113226                    113226
```

```
12213  047410  045513                    045513
12214  047412  122645                    122645
12215  047414  151322                    151322
12216  047416  064551                    064551
12217  047420  132264                    132264
12218  047422  055132                    055132
12219  047424  026455                    026455
12220  047426  113226                    113226
12221  047430  045513                    045513
12222  047432  122645                    122645
12223  047434  151322                    151322
12224  047436  064551                    064551
12225  047440  132264                    132264
12226  047442  055132                    055132
12227
12228                              ;DATA PATTERN 11
12229                              ;       SHIFTED 1 IN A FIELD OF ZEROS
12230  047444                      PAT11
12231  047444  000001                    000001
12232  047446  000002                    000002
12233  047450  000004                    000004
12234  047452  000010                    000010
12235  047454  000020                    000020
12236  047456  000040                    000040
12237  047460  000100                    000100
12238  047462  000200                    000200
12239  047464  000400                    000400
12240  047466  001000                    001000
12241  047470  002000                    002000
12242  047472  004000                    004000
12243  047474  010000                    010000
12244  047476  020000                    020000
12245  047500  040000                    040000
12246  047502  100000                    100000
12247
12248                              ;DATA PATTERN 12
12249                              ;       SHIFTED 0 IN A FIELD OF ONES
12250  047504                      PAT12:
12251  047504  177776                    177776
12252  047506  177775                    177775
12253  047510  177773                    177773
12254  047512  177767                    177767
12255  047514  177757                    177757
12256  047516  177737                    177737
12257  047520  177677                    177677
12258  047522  177577                    177577
12259  047524  177377                    177377
12260  047526  176777                    176777
12261  047530  175777                    175777
12262  047532  173777                    173777
12263  047534  167777                    167777
12264  047536  157777                    157777
12265  047540  137777                    137777
12266  047542  077777                    077777
12267
12268                              ;DATA PATTERN 13
```

```
12269
12270  047544
12271  047544  052525                ;       ALTERNATING 0-1
12272  047546  052525                PAT13:
12273  047550  052525                        052525
12274  047552  052525                        052525
12275  047554  052525                        052525
12276  047556  052525                        052525
12277  047560  052525                        052525
12278  047562  052525                        052525
12279  047564  052525                        052525
12280  047566  052525                        052525
12281  047570  052525                        052525
12282  047572  052525                        052525
12283  047574  052525                        052525
12284  047576  052525                        052525
12285  047600  052525                        052525
12286  047602  052525                        052525
12287
12288                                ;DATA PATTERN 14
12289                                ;       ALTERNATING 1-0
12290  047604                        PAT14:
12291  047604  125252                        125252
12292  047606  125252                        125252
12293  047610  125252                        125252
12294  047612  125252                        125252
12295  047614  125252                        125252
12296  047616  125252                        125252
12297  047620  125252                        125252
12298  047622  125252                        125252
12299  047624  125252                        125252
12300  047626  125252                        125252
12301  047630  125252                        125252
12302  047632  125252                        125252
12303  047634  125252                        125252
12304  047636  125252                        125252
12305  047640  125252                        125252
12306  047642  125252                        125252
12307
12308                                ;DATA PATTERN 15
12309                                ;       SHIFTING ZEROS AND ONES
12310  047644                        PAT15:
12311  047644  000001                        000001
12312  047646  000003                        000003
12313  047650  000007                        000007
12314  047652  000017                        000017
12315  047654  000037                        000037
12316  047656  000077                        000077
12317  047660  000177                        000177
12318  047662  000377                        000377
12319  047664  000777                        000777
12320  047666  001777                        001777
12321  047670  003777                        003777
12322  047672  007777                        007777
12323  047674  017777                        017777
12324  047676  037777                        037777
```

```
12325  047700  077777                  077777
12326  047702  177777                  177777
12327
12328                                  ;DATA PATTERN 16
12329                                  ;       COMPOSITE ROTATING
12330  047704                          PAT16:
12331  047704  072307                  072307
12332  047706  135143                  135143
12333  047710  156461                  156461
12334  047712  167230                  167230
12335  047714  073514                  073514
12336  047716  035646                  035646
12337  047720  016723                  016723
12338  047722  107351                  107351
12339  047724  143564                  143564
12340  047726  061672                  061672
12341  047730  030735                  030735
12342  047732  114356                  114356
12343  047734  046167                  046167
12344  047736  123073                  123073
12345  047740  151453                  151453
12346  047742  164616                  164616
12347
```

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

047744  000000                  EXPWC:  .WORD   0               ;EXPECTED WORD COUNT (GIVEN)
047746  000000                  EXPUBA: .WORD   0               ;EXPECTED UPPER BA (COMPUTED)
047750  000000                  EXPBA:  .WORD   0               ;EXPECTED BUS ADDRESS (COMPUTED)
047752  000000                  EXPCYL: .WORD   0               ;EXPECTED CYLINDER (COMPUTED)
047754  000000                  EXPSEC: .WORD   0               ;EXPECTED SECTOR (COMPUTED)
047756  000000                  EXPTRK: .WORD   0               ;EXPECTED TRACK (COMPUTED)

047760  000000                  REALWC: .WORD   0               ;WORD COUNT AT END OF OPERATION
047762  000000                  REALUB: .WORD   0               ;REAL UPPER BA
047764  000000                  REALBA: .WORD   0               ;BUS ADDRESS
047766  000000                  REALCY: .WORD   0               ;CYLINDER
047770  000000                  REALTRK:.WORD   0               ;TRACK
047772  000000                  REALSEC:.WORD   0               ;SECTOR

047774  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

047776  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

050000  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
```

```
12404           000100                  UPERR=  BIT6            ;UNIBUS PARITY ERROR
12405           000200                  ILFERR= BIT7            ;ILLEGAL FUNCTION ERROR.
12406
12407  050002   000000          GRP4ER: .WORD   0               ;GROUP 4 ERROR FIELD
12408           000001                  WCERR=  BIT0            ;WORD COUNT ERROR FLAG
12409           000002                  UBAERR= BIT1            ;UPPER BA ERROR
12410           000004                  BAERR=  BIT2            ;BUS ADDRESS ERROR FLAG
12411           000010                  CYLERR= BIT3            ;CYL ADDRESS ERROR FLAG
12412           000020                  TRKERR= BIT4            ;TRACK ADDRESS ERROR FLAG
12413           000040                  SECERR= BIT5            ;SECTOR ADDRESS ERROR FLAG
12414
12415  050004   054006          CRP4MS: .WORD   EM10
12416  050006   054061                  .WORD   EM11A
12417  050010   054033                  .WORD   EM11
12418  050012   054135                  .WORD   EM12
12419  050014   054170                  .WORD   EM13
12420  050016   054220                  .WORD   EM14
12421
12422  050020   054252          GRP3MS: .WORD   EM15
12423  050022   054275                  .WORD   EM16
12424  050024   054320                  .WORD   EM17
12425  050026   054341                  .WORD   EM18
12426  050030   054366                  .WORD   EM19
12427  050032   054410                  .WORD   EM20
12428  050034   054434                  .WORD   EM21
12429  050036   054460                  .WORD   EM22
12430
12431  050040   054501          GRP2MS: .WORD   EM23
12432  050042   054512                  .WORD   EM24
12433  050044   054525                  .WORD   EM25
12434  050046   054541                  .WORD   EM26
12435  050050   054553                  .WORD   EM27
12436  050052   054600                  .WORD   EM28
12437  050054   054613                  .WORD   EM29
12438
12439  050056   054634          GRP1MS: .WORD   EM30
12440  050060   054707                  .WORD   EM31
12441  050062   054727                  .WORD   EM32
12442  050064   054765                  .WORD   EM33
12443  050066   055033                  .WORD   EM34
12444  050070   055050                  .WORD   EM35
12445  050072   055071                  .WORD   EM36
12446  050074   055100                  .WORD   EM37
12447  050076   055123                  .WORD   EM38
12448  050100   055143                  .WORD   EM39
12449  050102   055165                  .WORD   EM40
12450  050104   055217                  .WORD   EM41
12451  050106   055240                  .WORD   EM42
12452  050110   055263                  .WORD   EM43
12453  050112   055325                  .WORD   EM44
12454  050114   055342                  .WORD   EM45
12455
12456  050116   000000          GPSUMF: .WORD   0               ;GROUP ERROR SUMMARY FLAGS
12457           000001                  GRP1ST= BIT0            ;GROUP 1 ERROR SET
12458           000002                  GRP2ST= BIT1            ;GROUP 2 ERROR SET
12459           000004                  GRP3ST= BIT2            ;GROUP 3 ERROR SET
```

```
12460           000010                          GP1NR=  BIT3            ;GROUP 1 ERROR NOT RECEIVED
12461           000020                          GP2NR=  BIT4            ;GROUP 2 ERROR NOT RECEIVED
12462           000040                          GP3NR=  BIT5            ;GROUP 3 ERROR NOT RECEIVED
12463           040000                          DRSTER= BIT14           ;ERROR IN GETTING DRIVE STATUS FLAG.
12464           100000                          REPNR=  BIT15           ;REPORTING NOT RECEIVED SWITCH
12465
12466                                   .SBTTL  TABLE OF OPERATION MESSAGE ADDRESS
12467                                   ;*      THIS TABLE CONTAINS THE ADDRESS OF ASCIZ FIELDS THAT ARE
12468                                   ;*      USED IN REPORTING TO IDENTIFY THE OPERATION BEING PERFORMED.
12469
12470  050120   050160                  CMNDLB: .WORD   OPER00          ;ADDRESS OF SELECT MESSAGE
12471  050122   050175                          .WORD   OPER02          ;               PACK ACK
12472  050124   050206                          .WORD   OPER04          ;               DRIVE CLEAR
12473  050126   050222                          .WORD   OPER06          ;               UNLOAD
12474  050130   050231                          .WORD   OPER10          ;               START SPINDLE
12475  050132   050247                          .WORD   OPER12          ;               RECALIBRATE
12476  050134   050263                          .WORD   OPER14          ;               OFFSET
12477  050136   050272                          .WORD   OPER16          ;               SEEK
12478  050140   050277                          .WORD   OPER20          ;               READ DATA
12479  050142   050311                          .WORD   OPER22          ;               WRITE DATA
12480  050144   050324                          .WORD   OPER24          ;               READ HEADER
12481  050146   050341                          .WORD   OPER26          ;               WRITE HEADER
12482  050150   050357                          .WORD   OPER30          ;               WRITE CHECK
12483  050152   050373                          .WORD   OPER32          ;ILLEGAL OPERATION 33
12484  050154   050417                          .WORD   OPER34          ;                  35
12485  050156   050443                          .WORD   OPER36          ;                  37
12486
12487                                   .SBTTL  OPERATION MESSAGES
12488  050160   051104   053111   020105  OPER00: .ASCIZ  /DRIVE SELECT/
12489  050166   042523   042514   052103
12490  050174      000
12491  050175      120   041501   020113  OPER02: .ASCIZ  /PACK ACK/
12492  050202   041501   000113
12493  050206   051104   053111   020105  OPER04: .ASCIZ  /DRIVE CLEAR/
12494  050214   046103   040505   000122
12495  050222   047125   047514   042101  OPER06: .ASCIZ  /UNLOAD/
12496  050230      000
12497  050231      123   040524   052122  OPER10: .ASCIZ  /START SPINDLE/
12498  050236   051440   044520   042116
12499  050244   042514      000
12500  050247      122   041505   046101  OPER12: .ASCIZ  /RECALIBRATE/
12501  050254   041111   040522   042524
12502  050262      000
12503  050263      117   043106   042523  OPER14: .ASCIZ  /OFFSET/
12504  050270   000124
12505  050272   042523   045505      000  OPER16: .ASCIZ  /SEEK/
12506  050277      122   040505   020104  OPER20: .ASCIZ  /READ DATA/
12507  050304   040504   040524      000
12508  050311      127   044522   042524  OPER22: .ASCIZ  /WRITE DATA/
12509  050316   042040   052101   000101
12510  050324   042522   042101   044040  OPER24: .ASCIZ  /READ HEADERS/
12511  050332   040505   042504   051522
12512  050340      000
12513  050341      127   044522   042524  OPER26: .ASCIZ  /WRITE HEADERS/
12514  050346   044040   040505   042504
12515  050354   051522      000
```

```
12516  050357     127  044522  042524  OPER30: .ASCIZ  /WRITE CHECK/
12517  050364  041440  042510  045503
12518  050372     000
12519  050373     111  046114  043505  OPER32: .ASCIZ  /ILLEGAL FUNCTION 33/
12520  050400  046101  043040  047125
12521  050406  052103  047511  020116
12522  050414  031463     000
12523  050417     111  046114  043505  OPER34: .ASCIZ  /ILLEGAL FUNCTION 35/
12524  050424  046101  043040  047125
12525  050432  052103  047511  020116
12526  050440  032463     000
12527  050443     111  046114  043505  OPER36: .ASCIZ  /ILLEGAL FUNCTION 37/
12528  050450  046101  043040  047125
12529  050456  052103  047511  020116
12530  050464  033463     000
12531  050467     127  044522  042524  OPER37: .ASCIZ  /WRITE DATA ABORTED WITH BSE/
12532  050474  042040  052101  020101
12533  050502  041101  051117  042524
12534  050510  020104  044527  044124
12535  050516  041040  042523     000
12536  050523                         OPER40:
12537  050523     127  044522  042524  OPER41: .ASCIZ  /WRITE CHECK ABORTED WITH WCE/
12538  050530  041440  042510  045503
12539  050536  040440  047502  052122
12540  050544  042105  053440  052111
12541  050552  020110  041527  000105
12542  050560  051127  052111  020105  OPER42: .ASCIZ  /WRITE DATA ABORTED WITH NON-EXISTANT MEMORY/
12543  050566  040504  040524  040440
12544  050574  047502  052122  042105
12545  050602  053440  052111  020110
12546  050610  047516  026516  054105
12547  050616  051511  040524  052116
12548  050624  046440  046505  051117
12549  050632  000131
12550  050634  042522  042101  042040  OPER43: .ASCIZ  /READ DATA ABORTED WITH NON-EXISTANT MEMORY/
12551  050642  052101  020101  041101
12552  050650  051117  042524  020104
12553  050656  044527  044124  047040
12554  050664  047117  042455  044530
12555  050672  052123  047101  020124
12556  050700  042515  047515  054522
12557  050706     000
12558  050707     127  044522  042524  OPER44: .ASCIZ  /WRITE DATA ABORTED WITH UNIBUS PARITY ERROR/
12559  050714  042040  052101  020101
12560  050722  041101  051117  042524
12561  050730  020104  044527  044124
12562  050736  052440  044516  052502
12563  050744  020123  040520  044522
12564  050752  054524  042440  051122
12565  050760  051117     000
12566
```

```
12567                                   .SBTTL  ASCII MESSAGES
12568
12569  050763    040  000040          SPACE2: .ASCIZ  /  /
12570  050766  005015  045522  030466  OPR001: .ASCIZ  <15><12>/RK611 BUS ADDRESS ( /
12571  050774  020061  052502  020123
12572  051002  042101  051104  051505
12573  051010  020123  020050     000
12574  051015     040  020051  020075  OPR002: .ASCIZ  / ) = /
12575  051022     000
12576  051023     122  033113  030461  OPR003: .ASCIZ  /RK611 VECTOR ADDRESS ( /
12577  051030  053040  041505  047524
12578  051036  020122  042101  051104
12579  051044  051505  020123  020050
12580  051052     000
12581  051053     122  033113  030461  OPR004: .ASCIZ  /RK611 PRIORITY ( /
12582  051060  050040  044522  051117
12583  051066  052111  020131  020050
12584  051074     000
12585  051075     111  051516  043125  OPR005: .ASCIZ  /INSUFFICIENT MEMORY. PROGRAM ABORTING./<15><12>
12586  051102  044506  044503  047105
12587  051110  020124  042515  047515
12588  051116  054522  020056  051120
12589  051124  043517  040522  020115
12590  051132  041101  051117  044524
12591  051140  043516  006456  000012
12592  051146                          OPR006:
12593                                   ;       .ASCII  <15><12>/TO BYPASS TESTING DRIVE 0, PLACE IT OFF-LINE/
12594                                   ;       .ASCIZ  <15><12>/TO TEST DRIVE 0, REPLACE PROGRAM PACK WITH SCRATCH PACK/
12595  051146                          OPR007:
12596  051146  005015  044103  047101          .ASCII  <CR><LF>/CHANGE XXDP PACK/
12597  051154  042507  054040  042130
12598  051162  020120  040520  045503
12599  051170  005015  046103  040505          .ASCIZ  <CR><LF>/CLEAR LOC 40,RESTART PROGRAM/
12600  051176  020122  047514  020103
12601  051204  030064  051054  051505
12602  051212  040524  052122  050040
12603  051220  047522  051107  046501
12604  051226     000
12605                                   ;       .ASCII  <15><12>/DRIVE 0 WILL NOT BE TESTED. TO TEST DRIVE 0,/<15><12>
12606                                   ;       .ASCIZ  /RESTART AT 214 AND MOUNT SCRATCH PACK AS DIRECTED./
12607  051227     015  047012  020117  OPR008: .ASCIZ  <15><12>/NO DRIVES AVAILABLE FOR TESTING. PROGRAM ABORTED/
12608  051234  051104  053111  051505
12609  051242  040440  040526  046111
12610  051250  041101  042514  043040
12611  051256  051117  052040  051505
12612  051264  044524  043516  020056
12613  051272  051120  043517  040522
12614  051300  020115  041101  051117
12615  051306  042524  000104
12616  051312  005015  044124  020105  OPR009: .ASCIZ  <15><12>/THE FOLLOWING DRIVES WILL BE TESTED/<15><12>
12617  051320  047506  046114  053517
12618  051326  047111  020107  051104
12619  051334  053111  051505  053440
12620  051342  046111  020114  042502
12621  051350  052040  051505  042524
12622  051356  006504  000012
```

```
12623  051362  005015  047516  050040  OPR010: .ASCII  <15><12>/NO PARITY OPTION FOR MEMORY AREA IN USE/<15><12>
12624  051370  051101  052111  020131
12625  051376  050117  044524  047117
12626  051404  043040  051117  046440
12627  051412  046505  051117  020131
12628  051420  051101  040505  044440
12629  051426  020116  051525  006505
12630  051434     012
12631  051435     055  052440  044516          .ASCIZ  /- UNIBUS PARITY ERROR TEST BYPASSED/<15><12>
12632  051442  052502  020123  040520
12633  051450  044522  054524  042440
12634  051456  051122  051117  052040
12635  051464  051505  020124  054502
12636  051472  040520  051523  042105
12637  051500  005015     000
12638  051503     015  046412  046505  OPR011: .ASCII  <15><12>/MEMORY SIZE NOT LARGE ENOUGH FOR BUS ADDRESS BITS 16 & 17 TESTS
12639  051510  051117  020131  044523
12640  051516  042532  047040  052117
12641  051524  046040  051101  042507
12642  051532  042440  047516  043525
12643  051540  020110  047506  020122
12644  051546  052502  020123  042101
12645  051554  051104  051505  020123
12646  051562  044502  051524  030440
12647  051570  020066  020046  033461
12648  051576  052040  051505  051524
12649  051604  005015
12650  051606  046101  020114  040504          .ASCIZ  /ALL DATA XFER TESTS WITH ADDR >/
12651  051614  040524  054040  042506
12652  051622  020122  042524  052123
12653  051630  020123  044527  044124
12654  051636  040440  042104  020122
12655  051644  000076
12656  051646  031063  000113          OPR012: .ASCIZ  /32K/
12657  051652  032066  000113          OPR013: .ASCIZ  /64K/
12658  051656  033071  000113          OPR014:  ASCIZ  /96K/
12659  051662  041040  050131  051501  OPR015: .ASCIZ  / BYPASSED/<15><12>
12660  051670  042523  006504  000012
12661  051676  005015  020012  020040  OPR016: .ASCII  <15><12><12>/       *** CAUTION ***/<15><12><12>
12662  051704  020040  020040  025052
12663  051712  020052  040503  052125
12664  051720  047511  020116  025052
12665  051726  006452  005012
12666  051732  044124  051511  050040          .ASCII  /THIS PROGRAM SHOULD BE HALTED BY TYPING ^C./<15><12>
12667  051740  047522  051107  046501
12668  051746  051440  047510  046125
12669  051754  020104  042502  044040
12670  051762  046101  042524  020104
12671  051770  054502  052040  050131
12672  051776  047111  020107  041536
12673  052004  006456     012
12674  052007     111  020106  040510          .ASCII  /IF HALTED USING THE HALT KEY, THE STATE OF THE DRIVE/<15><12>
12675  052014  052114  042105  052440
12676  052022  044523  043516  052040
12677  052030  042510  044040  046101
12678  052036  020124  042513  026131
```

```
12679  052044  052040  042510  051440
12680  052052  040524  042524  047440
12681  052060  020106  044124  020105
12682  052066  051104  053111  006505
12683  052074     012
12684  052075     117  020122  040503          .ASCII  /OR CARTRIDGE CANNOT BE PREDICTED./<15><12><12>
12685  052102  052122  044522  043504
12686  052110  020105  040503  047116
12687  052116  052117  041040  020105
12688  052124  051120  042105  041511
12689  052132  042524  027104  005015
12690  052140     012
12691  052141     101  046114  042040          .ASCII  /ALL DRIVES TO BE TESTED MUST BE ON-LINE,/<15><12>
12692  052146  044522  042526  020123
12693  052154  047524  041040  020105
12694  052162  042524  052123  042105
12695  052170  046440  051525  020124
12696  052176  042502  047440  026516
12697  052204  044514  042516  006454
12698  052212     012
12699  052213     122  040505  054504          .ASCII  /READY, AND WRITE LOCK RESET./<15><12>
12700  052220  020054  047101  020104
12701  052226  051127  052111  020105
12702  052234  047514  045503  051040
12703  052242  051505  052105  006456
12704  052250     012
12705  052251     101  054516  042040          .ASCII  /ANY DRIVE NOT TO BE TESTED MUST BE OFF-LINE./<15><12><12>
12706  052256  044522  042526  047040
12707  052264  052117  052040  020117
12708  052272  042502  052040  051505
12709  052300  042524  020104  052515
12710  052306  052123  041040  020105
12711  052314  043117  026506  044514
12712  052322  042516  006456  005012
12713  052330  047516  042524  020072          .ASCII  /NOTE: 2ND AND SUBSEQUENT PASS RUN TIME IS/<15><12>
12714  052336  047062  020104  047101
12715  052344  020104  052523  051502
12716  052352  050505  042525  052116
12717  052360  050040  051501  020123
12718  052366  052522  020116  044524
12719  052374  042515  044440  006523
12720  052402     012
12721  052403     040  020040  020040          .ASCIZ  /      APPROX 2 MIN 30 SEC FOR EACH DRIVE./<15><12>
12722  052410  040440  050120  047522
12723  052416  020130  020062  044515
12724  052424  020116  030063  051440
12725  052432  041505  043040  051117
12726  052440  042440  041501  020110
12727  052446  051104  053111  027105
12728  052454  005015     000
12729  052457     015  043012  051111  OPR017: .ASCII  <15><12>/FIRST 256 SECTORS NOT BSE ERROR FREE./
12730  052464  052123  031040  033065
12731  052472  051440  041505  047524
12732  052500  051522  047040  052117
12733  052506  041040  042523  042440
12734  052514  051122  051117  043040
```

```
12735  052522  042522  027105
12736  052526  040515  044530  052515          .ASCIZ  /MAXIMUM DATA TRANSFER TEST BYPASSED/<15><12>
12737  052534  020115  040504  040524
12738  052542  052040  040522  051516
12739  052550  042506  020122  042524
12740  052556  052123  041040  050131
12741  052564  051501  042523  006504
12742  052572  000012
12743  052574  020040  020040  006440  OPR018: .ASCIZ  /     /<15><12>/ONLY 1 DRIVE. OVERLAPPED OPERATIONS BYPASSED/<15><12>
12744  052602  047412  046116  020131
12745  052610  020061  051104  053111
12746  052616  027105  047440  042526
12747  052624  046122  050101  042520
12748  052632  020104  050117  051105
12749  052640  052101  047511  051516
12750  052646  041040  050131  051501
12751  052654  042523  006504  000012
12752  052662  005015  041523  050117  OPR019: .ASCII  <15><12>@SCOPE: CH1 (TRIG), E53-8; CH2, E49-2 (AC COUPLE, .2V/CM)@
12753  052670  035105  041440  030510
12754  052676  024040  051124  043511
12755  052704  026051  042440  031465
12756  052712  034055  020073  044103
12757  052720  026062  042440  034464
12758  052726  031055  024040  041501
12759  052734  041440  052517  046120
12760  052742  026105  027040  053062
12761  052750  041457  024515
12762  052754  005015  042101  052512          .ASCIZ  <15><12>/ADJUST R72 FOR CONSTANT LEVEL ON CH2/<15><12>
12763  052762  052123  051040  031067
12764  052770  043040  051117  041440
12765  052776  047117  052123  047101
12766  053004  020124  042514  042526
12767  053012  020114  047117  041440
12768  053020  031110  005015     000
12769  053025     015  050012  047522  OPR020: .ASCIZ  <15><12>/PROGRAM HALT PENDING - CARTRIDGE FORMAT BEING CORRECTED/<15><12
12770  053032  051107  046501  044040
12771  053040  046101  020124  042520
12772  053046  042116  047111  020107
12773  053054  020055  040503  052122
12774  053062  044522  043504  020105
12775  053070  047506  046522  052101
12776  053076  041040  044505  043516
12777  053104  041440  051117  042522
12778  053112  052103  042105  005015
12779  053120     000
12780  053121     015  025012  025052  OPR021: .ASCIZ  <15><12>/*****  PROGRAM HALTED  *****/<15><12>
12781  053126  025052  020040  051120
12782  053134  043517  040522  020115
12783  053142  040510  052114  042105
12784  053150  020040  025052  025052
12785  053156  006452  000012
12786  053162  040503  052122  044522  OPR022: .ASCIZ  /CARTRIDGE FORMAT CORRECTION FAILED/<15><12>
12787  053170  043504  020105  047506
12788  053176  046522  052101  041440
12789  053204  051117  042522  052103
12790  053212  047511  020116  040506
```

```
12791  053220  046111  042105  005015
12792  053226     000
12793  053227     015  050012  047522  ABORT:   .ASCIZ  <15><12>/PROGRAM ABORTING BECAUSE ERROR THRESHOLD EXCEEDED/<15><12>
12794  053234  051107  046501  040440
12795  053242  047502  052122  047111
12796  053250  020107  042502  040503
12797  053256  051525  020105  051105
12798  053264  047522  020122  044124
12799  053272  042522  044123  046117
12800  053300  020104  054105  042503
12801  053306  042105  042105  005015
12802  053314     000
12803
12804                          .SBTTL ERROR MESSAGES
12805  053315     106  052101  046101  EM1:     .ASCIZ  /FATAL-NON EXISTANT MEMORY AT RK611 BASE ADDRESS/
12806  053322  047055  047117  042440
12807  053330  044530  052123  047101
12808  053336  020124  042515  047515
12809  053344  054522  040440  020124
12810  053352  045522  030466  020061
12811  053360  040502  042523  040440
12812  053366  042104  042522  051523
12813  053374     000
12814  053375     106  052101  046101  EM2:     .ASCIZ  /FATAL-WRITE READY AND IE DID NOT CAUSE INTERRUPT/
12815  053402  053455  044522  042524
12816  053410  051040  040505  054504
12817  053416  040440  042116  044440
12818  053424  020105  044504  020104
12819  053432  047516  020124  040503
12820  053440  051525  020105  047111
12821  053446  042524  051122  050125
12822  053454  000124
12823  053456  040506  040524  026514  EM3:     .ASCIZ  /FATAL-PARITY ERROR TRAP. PC AT ERROR = /
12824  053464  040520  044522  054524
12825  053472  042440  051122  051117
12826  053500  052040  040522  027120
12827  053506  050040  020103  052101
12828  053514  042440  051122  051117
12829  053522  036440  000040
12830  053526  054105  042520  052103  EM4:     .ASCIZ  /EXPECTED INTERRUPT DID NOT OCCUR OR WAS LATE. COMMAND WAS:/
12831  053534  042105  044440  052116
12832  053542  051105  052522  052120
12833  053550  042040  042111  047040
12834  053556  052117  047440  041503
12835  053564  051125  047440  020122
12836  053572  040527  020123  040514
12837  053600  042524  020056  047503
12838  053606  046515  047101  020104
12839  053614  040527  035123     000
12840  053621     123  041125  054523  EM5:     .ASCIZ  /SUBSYSTEM CLEAR DID NOT RESET ERROR/
12841  053626  052123  046505  041440
12842  053634  042514  051101  042040
12843  053642  042111  047040  052117
12844  053650  051040  051505  052105
12845  053656  042440  051122  051117
12846  053664     000
```

```
12847  053665     123  041125  054523  EM6:    .ASCIZ  /SUBSYSTEM CLEAR DID NOT RESET DEVICE INTERRUPT/
12848  053672  052123  046505  041440
12849  053700  042514  051101  042040
12850  053706  042111  047040  052117
12851  053714  051040  051505  052105
12852  053722  042040  053105  041511
12853  053730  020105  047111  042524
12854  053736  051122  050125  000124
12855  053744  052523  051502  051531  EM7:    .ASCIZ  /SUBSYSTEM CLEAR DID NOT SET READY/
12856  053752  042524  020115  046103
12857  053760  040505  020122  044504
12858  053766  020104  047516  020124
12859  053774  042523  020124  042522
12860  054002  042101  000131
12861  054006  047527  042122  041440  EM10:   .ASCIZ  /WORD COUNT INCORRECT/
12862  054014  052517  052116  044440
12863  054022  041516  051117  042522
12864  054030  052103     000
12865  054033     102  051525  040440  EM11:   .ASCIZ  /BUS ADDRESS INCORRECT/
12866  054040  042104  042522  051523
12867  054046  044440  041516  051117
12868  054054  042522  052103     000
12869  054061     125  050120  051105  EM11A:  .ASCIZ  /UPPER BUS ADDRESS BITS INCORRECT (BA16, 17)/
12870  054066  041040  051525  040440
12871  054074  042104  042522  051523
12872  054102  041040  052111  020123
12873  054110  047111  047503  051122
12874  054116  041505  020124  041050
12875  054124  030501  026066  030440
12876  054132  024467     000
12877  054135     103  046131  047111  EM12:   .ASCIZ  /CYLINDER ADDRESS INCORRECT/
12878  054142  042504  020122  042101
12879  054150  051104  051505  020123
12880  054156  047111  047503  051122
12881  054164  041505  000124
12882  054170  051124  041501  020113  EM13:   .ASCIZ  /TRACK ADDRESS INCORRECT/
12883  054176  042101  051104  051505
12884  054204  020123  047111  047503
12885  054212  051122  041505  000124
12886  054220  042523  052103  051117  EM14:   .ASCIZ  /SECTOR ADDRESS INCORRECT./
12887  054226  040440  042104  042522
12888  054234  051523  044440  041516
12889  054242  051117  042522  052103
12890  054250  000056
12891  054252  047516  026516  054105  EM15:   .ASCIZ  /NON-EXISTANT DRIVE/
12892  054260  051511  040524  052116
12893  054266  042040  044522  042526
12894  054274     000
12895  054275     103  047117  051124  EM16:   .ASCIZ  /CONTROLLER TIMEOUT/
12896  054302  046117  042514  020122
12897  054310  044524  042515  052517
12898  054316  000124
12899  054320  047125  052111  043040  EM17:   .ASCIZ  /UNIT FIELD ERROR/
12900  054326  042511  042114  042440
12901  054334  051122  051117     000
12902  054341     115  046125  050111  EM18:   .ASCIZ  /MULIPLE DRIVE SELECT/
```

```
12903  054346  042514  042040  044522
12904  054354  042526  051440  046105
12905  054362  041505  000124
12906  054366  051120  043517  040522  EM19:   .ASCIZ  /PROGRAMMING ERROR/
12907  054374  046515  047111  020107
12908  054402  051105  047522  000122
12909  054410  047516  026516  054105  EM20:   .ASCIZ  /NON-EXISTANT MEMORY/
12910  054416  051511  040524  052116
12911  054424  046440  046505  051117
12912  054432  000131
12913  054434  047125  041111  051525  EM21:   .ASCIZ  /UNIBUS PARITY ERROR/
12914  054442  050040  051101  052111
12915  054450  020131  051105  047522
12916  054456  000122
12917  054460  046111  042514  040507  EM22:   .ASCIZ  /ILLEGAL FUNCTION/
12918  054466  020114  052506  041516
12919  054474  044524  047117     000
12920  054501     105  041503  044040  EM23:   .ASCIZ  /ECC HARD/
12921  054506  051101  000104
12922  054512  040504  040524  041440  EM24:   .ASCIZ  /DATA CHECK/
12923  054520  042510  045503     000
12924  054525     127  044522  042524  EM25:   .ASCIZ  /WRITE CHECK/
12925  054532  041440  042510  045503
12926  054540     000
12927  054541     104  052101  020101  EM26:   .ASCIZ  /DATA LATE/
12928  054546  040514  042524     000
12929  054553     117  042520  040522  EM27:   .ASCIZ  /OPERATION INCOMPLETE/
12930  054560  044524  047117  044440
12931  054566  041516  046517  046120
12932  054574  052105  000105
12933  054600  042510  042101  051105  EM28:   .ASCIZ  /HEADER VRC/
12934  054606  053040  041522     000
12935  054613     102  042101  051440  EM29:   .ASCIZ  /BAD SECTOR ERROR/
12936  054620  041505  047524  020122
12937  054626  051105  047522  000122
12938  054634  047503  052116  047522  EM30:   .ASCIZ  /CONTROLLER DETECTED DRIVE BUS PARITY ERROR/
12939  054642  046114  051105  042040
12940  054650  052105  041505  042524
12941  054656  020104  051104  053111
12942  054664  020105  052502  020123
12943  054672  040520  044522  054524
12944  054700  042440  051122  051117
12945  054706     000
12946  054707     123  042505  020113  EM31:   .ASCIZ  /SEEK INCOMPLETE/
12947  054714  047111  047503  050115
12948  054722  042514  042524     000
12949  054727     116  047117  042455  EM32:   .ASCIZ  /NON-EXECUTABLE DRIVE FUNCTION/
12950  054734  042530  052503  040524
12951  054742  046102  020105  051104
12952  054750  053111  020105  052506
12953  054756  041516  044524  047117
12954  054764     000
12955  054765     104  044522  042526  EM33:   .ASCIZ  /DRIVE DETECTED DRIVE BUS PARITY ERROR/
12956  054772  042040  052105  041505
12957  055000  042524  020104  051104
12958  055006  053111  020105  052502
```

```
12959  055014  020123  040520  044522
12960  055022  054524  042440  051122
12961  055030  051117     000
12962  055033     106  051117  040515  EM34:   .ASCIZ  /FORMAT ERROR/
12963  055040  020124  051105  047522
12964  055046  000122
12965  055050  051104  053111  020105  EM35:   .ASCIZ  /DRIVE TYPE ERROR/
12966  055056  054524  042520  042440
12967  055064  051122  051117     000
12968  055071     101  020103  047514  EM36:   .ASCIZ  /AC LOW/
12969  055076  000127
12970  055100  050123  047111  046104  EM37:   .ASCIZ  /SPINDLE SPEED LOSS/
12971  055106  020105  050123  042505
12972  055114  020104  047514  051523
12973  055122     000
12974  055123     104  044522  042526  EM38:   .ASCIZ  /DRIVE OFF TRACK/
12975  055130  047440  043106  052040
12976  055136  040522  045503     000
12977  055143     103  046131  047111  EM39:   .ASCIZ  /CYLINDER OVERFLOW/
12978  055150  042504  020122  053117
12979  055156  051105  046106  053517
12980  055164     000
12981  055165     111  046114  043505  EM40:   .ASCIZ  /ILLEGAL DISK PACK ADDRESS/
12982  055172  046101  042040  051511
12983  055200  020113  040520  045503
12984  055206  040440  042104  042522
12985  055214  051523     000
12986  055217     127  044522  042524  EM41:   .ASCIZ  /WRITE LOCK ERROR/
12987  055224  046040  041517  020113
12988  055232  051105  047522  000122
12989  055240  051104  053111  020105  EM42:   .ASCIZ  /DRIVE TIMING ERROR/
12990  055246  044524  044515  043516
12991  055254  042440  051122  051117
12992  055262     000
12993  055263     116  020117  042503  EM43:   .ASCIZ  /NO CERR WITH SOME OTHER ERROR SET/
12994  055270  051122  053440  052111
12995  055276  020110  047523  042515
12996  055304  047440  044124  051105
12997  055312  042440  051122  051117
12998  055320  051440  052105     000
12999  055325     104  044522  042526  EM44:   .ASCIZ  /DRIVE UNSAFE/
13000  055332  052440  051516  043101
13001  055340  000105
13002  055342  042503  051122  051440  EM45:   .ASCIZ  /CERR SET BUT NO OTHER ERROR SET/
13003  055350  052105  041040  052125
13004  055356  047040  020117  052117
13005  055364  042510  020122  051105
13006  055372  047522  020122  042523
13007  055400  000124
13008
13009  055402  053126  042040  042111  EM46:   .ASCIZ  /VV DID NOT SET WITH PACK ACK/
13010  055410  047040  052117  051440
13011  055416  052105  053440  052111
13012  055424  020110  040520  045503
13013  055432  040440  045503     000
13014  055437     123  040524  052524  EM47:   .ASCIZ  /STATUS VALID SET ON SELECT TO NON-EXISTANT DRIVE/
```

```
 13015  055444  020123  040526  044514
 13016  055452  020104  042523  020124
 13017  055460  047117  051440  046105
 13018  055466  041505  020124  047524
 13019  055474  047040  047117  042455
 13020  055502  044530  052123  047101
 13021  055510  020124  051104  053111
 13022  055516  000105
 13023  055520  052123  052101  051525  EM48:   .ASCIZ  /STATUS VALID RESET ON SELECT TO EXISTANT DRIVE/
 13024  055526  053040  046101  042111
 13025  055534  051040  051505  052105
 13026  055542  047440  020116  042523
 13027  055550  042514  052103  052040
 13028  055556  020117  054105  051511
 13029  055564  040524  052116  042040
 13030  055572  044522  042526     000
 13031  055577     123  040524  052524  EM49:   .ASCIZ  /STATUS VALID NOT RESET ON DRIVE RELEASE/
 13032  055604  020123  040526  044514
 13033  055612  020104  047516  020124
 13034  055620  042522  042523  020124
 13035  055626  047117  042040  044522
 13036  055634  042526  051040  046105
 13037  055642  040505  042523     000
 13038  055647     105  050130  041505  EM50:   .ASCIZ  /EXPECTED 2ND INTERRUPT DID NOT OCCUR OR WAS LATE. COMMAND WAS:/
 13039  055654  042524  020104  047062
 13040  055662  020104  047111  042524
 13041  055670  051122  050125  020124
 13042  055676  044504  020104  047516
 13043  055704  020124  041517  052503
 13044  055712  020122  051117  053440
 13045  055720  051501  046040  052101
 13046  055726  027105  041440  046517
 13047  055734  040515  042116  053440
 13048  055742  051501  000072
 13049  055746  040503  047116  052117  EM51:   .ASCII  /CANNOT READ BAD SECTOR FILE/<15><12>
 13050  055754  051040  040505  020104
 13051  055762  040502  020104  042523
 13052  055770  052103  051117  043040
 13053  055776  046111  006505     012
 13054  056003     124  051505  044524          .ASCIZ  /TESTING ABORTED ON THIS DRIVE/
 13055  056010  043516  040440  047502
 13056  056016  052122  042105  047440
 13057  056024  020116  044124  051511
 13058  056032  042040  044522  042526
 13059  056040     000
 13060  056041     101  044514  047107  EM52:   .ASCII  /ALIGNMENT PACK ON THIS DRIVE./<15><12>
 13061  056046  042515  052116  050040
 13062  056054  041501  020113  047117
 13063  056062  052040  044510  020123
 13064  056070  051104  053111  027105
 13065  056076  005015
 13066  056100  042524  052123  047111          .ASCIZ  /TESTING ABORTED ON THIS DRIVE./
 13067  056106  020107  041101  051117
 13068  056114  042524  020104  047117
 13069  056122  052040  044510  020123
 13070  056130  051104  053111  027105
```

```
13071  056136      000
13072  056137      124  020117  040515  EM53:   .ASCIZ  /TO MANY BAD SECTORS. TESTING ABORTED ON THIS DRIVE/
13073  056144  054516  041040  042101
13074  056152  051440  041505  047524
13075  056160  051522  020056  042524
13076  056166  052123  047111  020107
13077  056174  041101  051117  042524
13078  056202  020104  047117  052040
13079  056210  044510  020123  051104
13080  056216  053111  000105
13081  056222  042510  042101  051105  EM54:   .ASCIZ  /HEADER WORD MISCOMPARE/
13082  056230  053440  051117  020104
13083  056236  044515  041523  046517
13084  056244  040520  042522     000
13085  056251      104  052101  020101  EM55:   .ASCIZ  /DATA WORD MISCOMPARE/
13086  056256  047527  042122  046440
13087  056264  051511  047503  050115
13088  056272  051101  000105
13089
13090  056276  040503  047116  052117  EM56:   .ASCIZ  /CANNOT FIND HEADER 0 IN READ AND SORT HEADERS/
13091  056304  043040  047111  020104
13092  056312  042510  042101  051105
13093  056320  030040  044440  020116
13094  056326  042522  042101  040440
13095  056334  042116  051440  051117
13096  056342  020124  042510  042101
13097  056350  051105  000123
13098  056354  052123  052101  051525  EMSVAL: .ASCIZ  /STATUS VALID/
13099  056362  053040  046101  042111
13100  056370      000
13101  056371      123  046105  041505  EMNZPR: .ASCIZ  /SELECT WITH NON-ZERO STATUS PAIR/
13102  056376  020124  044527  044124
13103  056404  047040  047117  055055
13104  056412  051105  020117  052123
13105  056420  052101  051525  050040
13106  056426  044501  000122
13107  056432  051127  052111  047111  EMWCS2: .ASCIZ  /WRITING COMMAND STATUS REGISTER 2/
13108  056440  020107  047503  046515
13109  056446  047101  020104  052123
13110  056454  052101  051525  051040
13111  056462  043505  051511  042524
13112  056470  020122  000062
13113          055050                  EMDTPE= EM35            ;DRIVE TYPE ERROR
13114  056474  051104  053111  020105  EMDI:   .ASCIZ  /DRIVE INTERRUPT/
13115  056502  047111  042524  051122
13116  056510  050125  000124
13117          054765                  EMDPAR= EM33                    ;DRIVE DETECTED DRIVE BUS PARITY ERROR
13118  056514  051104  053111  020105  EMDSC:  .ASCIZ  /DRIVE STATUS CHANGE/
13119  056522  052123  052101  051525
13120  056530  041440  040510  043516
13121  056536  000105
13122  056540  051104  053111  020105  EMDA:   .ASCIZ  /DRIVE ATTENTION/
13123  056546  052101  042524  052116
13124  056554  047511  000116
13125  056560  047503  052116  047522  EMCCLR: .ASCIZ  /CONTROLLER CLEAR/
13126  056566  046114  051105  041440
```

```
13127  056574  042514  051101     000
13128          050160                    EMSELD= OPER00          ;DRIVE SELECT
13129          050206                    EMDCLR= OPER04                  ;DRIVE CLEAR
13130          050247                    EMRCAL= OPER12                  ;RECALIBRATE
13131  056601     123  041505  047117    EM2INT: .ASCIZ  /SECOND INTERRUPT/
13132  056606  020104  047111  042524
13133  056614  051122  050125  000124
13134  056622  042523  045505  052040    EMSKSF: .ASCIZ  /SEEK TO SELF/
13135  056630  020117  042523  043114
13136  056636     000
13137          050272                    EMSK=   OPER16                  ;SEEK
13138  056637     125  042516  050130    EMUXIT: .ASCIZ  /UNEXPECTED INTERRUPT/
13139  056644  041505  042524  020104
13140  056652  047111  042524  051122
13141  056660  050125  000124
13142  056664  047125  041111  051525    EMUR:   .ASCIZ  /UNIBUS RESET/
13143  056672  051040  051505  052105
13144  056700     000
13145  056701     122  051505  052105    EMRSET: .ASCIZ  /RESET/
13146  056706     000
13147          054541                    EMDLT=  EM26                    ;DATA LATE
13148  056707     122  040505  044504    EMRDB:  .ASCIZ  /READING DATA BUFFER/
13149  056714  043516  042040  052101
13150  056722  020101  052502  043106
13151  056730  051105     000
13152  056733     103  047117  051124    EMCERR: .ASCIZ  /CONTROLLER ERROR/
13153  056740  046117  042514  020122
13154  056746  051105  047522  000122
13155  056754  052523  051502  051531    EMSCLR: .ASCIZ  /SUBSYSTEM CLEAR/
13156  056762  042524  020115  046103
13157  056770  040505  000122
13158  056774  052515  052114  050111    EMMI:   .ASCIZ  /MULTIPLE INTERRUPTS/
13159  057002  042514  044440  052116
13160  057010  051105  052522  052120
13161  057016  000123
13162  057020  040502  020104  042523    DRVABT: .ASCII  /BAD SECTORS ON PACK IN AREAS USED BY TEST (CYL 312(8))/<15><12>
13163  057026  052103  051117  020123
13164  057034  047117  050040  041501
13165  057042  020113  047111  040440
13166  057050  042522  051501  052440
13167  057056  042523  020104  054502
13168  057064  052040  051505  020124
13169  057072  041450  046131  031440
13170  057100  031061  034050  024451
13171  057106  005015
13172  057110  042524  052123  047111          .ASCIZ  /TESTING ABORTED ON THIS DRIVE/
13173  057116  020107  041101  051117
13174  057124  042524  020104  047117
13175  057132  052040  044510  020123
13176  057140  051104  053111  000105
13177
13178                                    .SBTTL  DATA HEADERS FOR ERROR REPORTS
13179  057146  051524  020124  052516    DH001:  .ASCIZ  /TST NUM ERR PC  DRIVE/
13180  057154  020115  051105  020122
13181  057162  041520  020040  051104
13182  057170  053111  000105
```

```
13183  057174  051524  020124  052516  DH002A: .ASCIZ  /TST NUM ERR PC  DRIVE/
13184  057202  020115  051105  020122
13185  057210  041520  020040  051104
13186  057216  053111  000105
13187  057222  045522  051503  020061  DH002B: .ASCIZ  /RKCS1   RKCS2   RKDS    RKER    RKASOF  RKDCYL  RKDA/
13188  057230  020040  045522  051503
13189  057236  020062  020040  045522
13190  057244  051504  020040  020040
13191  057252  045522  051105  020040
13192  057260  020040  045522  051501
13193  057266  043117  020040  045522
13194  057274  041504  046131  020040
13195  057302  045522  040504     000
13196  057307     122  041113  020101  DH002C: .ASCIZ  /RKBA    RKWC/
13197  057314  020040  051040  053513
13198  057322  000103
13199  057324  030101  020060  020040  DH002D: .ASCIZ  /A00     B00     A01     B01     A02     B02     A03     B03/
13200  057332  020040  030102  020060
13201  057340  020040  020040  030101
13202  057346  020061  020040  020040
13203  057354  030102  020061  020040
13204  057362  020040  030101  020062
13205  057370  020040  020040  030102
13206  057376  020062  020040  020040
13207  057404  030101  020063  020040
13208  057412  020040  030102  020063
13209  057420  045522  051503  020061  DH003B. .ASCIZ  /RKCS1   RKCS2   RKDS    RKER    RKASOF  RKMR1/
13210  057426  020040  045522  051503
13211  057434  020062  020040  045522
13212  057442  051504  020040  020040
13213  057450  045522  051105  020040
13214  057456  020040  045522  051501
13215  057464  043117  020040  045522
13216  057472  051115  000061
13217  057476  044124  020105  041101  DH005:  .ASCIZ  /THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:/
13218  057504  053117  020105  051101
13219  057512  020105  054105  042520
13220  057520  052103  042105  042440
13221  057526  051122  051117  020123
13222  057534  044124  052101  042040
13223  057542  042111  047040  052117
13224  057550  051440  052105  044440
13225  057556  020116  050117  051105
13226  057564  052101  047511  035116
13227  057572     000
13228  057573     124  042510  040440  DH006:  .ASCIZ  /THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:/
13229  057600  047502  042526  040440
13230  057606  042522  052440  042516
13231  057614  050130  041505  042524
13232  057622  020104  051105  047522
13233  057630  051522  051440  052105
13234  057636  044440  020116  050117
13235  057644  051105  052101  047511
13236  057652  035116     000
13237  057655     124  042510  040440  DH007:  .ASCIZ  /THE ABOVE ARE ERRORS SET IN OPERATION:/
13238  057662  047502  042526  040440
```

```
13239  057670  042522  042440  051122
13240  057676  051117  020123  042523
13241  057704  020124  047111  047440
13242  057712  042520  040522  044524
13243  057720  047117  000072
13244  057724  047101  020131  044506  DH005A: .ASCIZ  /ANY FIELD WITH ALL ONES MUST BE CONSIDERED INVALID/
13245  057732  046105  020104  044527
13246  057740  044124  040440  046114
13247  057746  047440  042516  020123
13248  057754  052515  052123  041040
13249  057762  020105  047503  051516
13250  057770  042111  051105  042105
13251  057776  044440  053116  046101
13252  060004  042111     000
13253  060007     105  051122  051117  DH010:  .ASCIZ  /ERROR AT COMPLETION OF OPERATION/
13254  060014  040440  020124  047503
13255  060022  050115  042514  044524
13256  060030  047117  047440  020106
13257  060036  050117  051105  052101
13258  060044  047511  000116
13259  060050  054105  052120  020104  DH010A: .ASCIZ  /EXPTD   IS/
13260  060056  020040  051511     000
13261  060063     122  040505  020104  DH010B: .ASCIZ  /READ DATA WITH FORCED CYLINDER OVERFLOW/
13262  060070  040504  040524  053440
13263  060076  052111  020110  047506
13264  060104  041522  042105  041440
13265  060112  046131  047111  042504
13266  060120  020122  053117  051105
13267  060126  046106  053517     000
13268  060133     116  052117  051440  DH011:  .ASCIZ  /NOT SET AS A RESULT OF/
13269  060140  052105  040440  020123
13270  060146  020101  042522  052523
13271  060154  052114  047440  000106
13272  060162  047516  020124  042522  DH012:  .ASCIZ  /NOT RESET AS A RESULT OF/
13273  060170  042523  020124  051501
13274  060176  040440  051040  051505
13275  060204  046125  020124  043117
13276  060212     000
13277  060213     123  052105  040440  DH013:  .ASCIZ  /SET AS A RESULT OF/
13278  060220  020123  020101  042522
13279  060226  052523  052114  047440
13280  060234  000106
13281  060236  042522  042523  020124  DH014:  .ASCIZ  /RESET AS A RESULT OF/
13282  060244  051501  040440  051040
13283  060252  051505  046125  020124
13284  060260  043117     000
13285  060263     107  047517  020104  DH015:  .ASCIZ  /GOOD    BAD     WORD NUM/
13286  060270  020040  041040  042101
13287  060276  020040  020040  053440
13288  060304  051117  020104  052516
13289  060312  000115
13290  060314  052523  051502  051531  DH016:  .ASCIZ  /SUBSYSTEM CLEAR TO RESET SEEK LIMIT ALLOWING HEADS TO RELOAD/
13291  060322  042524  020115  046103
13292  060330  040505  020122  047524
13293  060336  051040  051505  052105
13294  060344  051440  042505  020113
```

```
 13295  060352  044514  044515  020124
 13296  060360  046101  047514  044527
 13297  060366  043516  044040  040505
 13298  060374  051504  052040  020117
 13299  060402  042522  047514  042101
 13300  060410     000
 13301  060411     123  046105  041505  DH017:  .ASCIZ  /SELECT AFTER RECAL/
 13302  060416  020124  043101  042524
 13303  060424  020122  042522  040503
 13304  060432  000114
 13305  060434  042522  040503  020114  DH018:  .ASCIZ  /RECAL COMPLETE ATTN AFTER SELECT/
 13306  060442  047503  050115  042514
 13307  060450  042524  040440  052124
 13308  060456  020116  043101  042524
 13309  060464  020122  042523  042514
 13310  060472  052103     000
 13311
 13312                                  .SBTTL  DATA TABLES FOR ERROR REPORTS
 13313          060476                  .EVEN
 13314  060476  001302  001116  001626  DT003:  .WORD   $TESTN,$ERRPC,DRVNUM,T.CS1,T.CS2,T.DS,T.ER,T.ASOF,T.MR1
 13315  060504  001540  001550  001552
 13316  060512  001554  001556  001566
 13317  060520  001302  001116  001626  DT001:  .WORD   $TESTN,$ERRPC,DRVNUM
 13318  060526                          DT002:
 13319  060526                          DT004:
 13320  060526                          DT005:
 13321  060526                          DT006:
 13322  060526                          DT007:
 13323  060526  001302  001116  001626  DT010:  .WORD   $TESTN,$ERRPC DRVNUM
 13324  060534  001540  001550  001552          .WORD   T.CS1,T.CS2,T.DS,T.ER,T.ASOF
 13325  060542  001554  001556
 13326  060546  001560  001546          .WORD   T.DCYL,T.DA
 13327  060552  001544  001542          .WORD   T.BA,T.WC
 13328  060556  001202  001204  001206  .WORD   $REG10,$REG11,$REG12,$REG13,$REG14,$REG15,$REG16,$REG17
 13329  060564  001210  001212  001214
 13330  060572  001216  001220
 13331  060576  001302  001116  001626  DT015:  .WORD   $TESTN,$ERRPC,DRVNUM
 13332  060604  001202  001204  001206  DT015A: .WORD   $REG10,$REG11,$REG12
 13333                                  .SBTTL  DATA FORMATS FOR ERROR REPORTS
 13334  060612  000001                  DF001:  .WORD   1
 13335  060614     003     000                  .BYTE   3,0
 13336
 13337  060616  000004                  DF002:  .WORD   4
 13338  060620     000     000                  .BYTE   0,0
 13339  060622  057174                  .WORD   DH002A
 13340  060624     003     000                  .BYTE   3,0
 13341  060626  057222                  .WORD   DH002B
 13342  060630     007     000                  .BYTE   7,0
 13343  060632  057307                  .WORD   DH002C
 13344  060634     002     000                  .BYTE   2,0
 13345
 13346
 13347  060636  000002                  DF003:  .WORD   2
 13348  060640     003     000                  .BYTE   3,0
 13349  060642  057420                  .WORD   DH003B
 13350  060644     006     000                  .BYTE   6,0
```

```
13351
13352  060646  000006                DF004:  .WORD   6
13353  060650     000      000               .BYTE   0,0
13354  060652  000000                DF004A: .WORD   0
13355  060654     000      000               .BYTE   0,0
13356  060656  057174                        .WORD   DH002A
13357  060660     003      000               .BYTE   3,0
13358  060662  057222                        .WORD   DH002B
13359  060664     007      000               .BYTE   7,0
13360  060666  057307                        .WORD   DH002C
13361  060670     002      000               .BYTE   2,0
13362  060672  057324                        .WORD   DH002D
13363  060674     010      000               .BYTE   10,0
13364
13365  060676  000007                DF005:  .WORD   7
13366  060700     000      000               .BYTE   0,0
13367  060702  000000                DF005A: .WORD   0
13368  060704     000      000               .BYTE   0,0
13369  060706  057174                        .WORD   DH0C2A
13370  060710     003      000               .BYTE   3,0
13371  060712  057222                        .WORD   DH002B
13372  060714     007      000               .BYTE   7,0
13373  060716  057307                        .WORD   DH002C
13374  060720     002      000               .BYTE   2,0
13375  060722  057324                        .WORD   DH002D
13376  060724     010      000               .BYTE   10,0
13377  060726  057724                        .WORD   DH005A
13378  060730     000      000               .BYTE   0,0
13379
13380  060732  000005                DF006:  .WORD   5
13381  060734     000      000               .BYTE   0,0
13382  060736  000000                DF006A: .WORD   0
13383  060740     000      000               .BYTE   0,0
13384  060742  057174                        .WORD   DH002A
13385  060744     003      000               .BYTE   3,0
13386  060746  057222                        .WORD   DH002B
13387  060750     007      000               .BYTE   7,0
13388  060752  057307                        .WORD   DH002C
13389  060754     002      000               .BYTE   2,0
13390
13391  060756  000004                DF007:  .WORD   4
13392  060760     000      000               .BYTE   0,0
13393  060762  000000                DF007A: .WORD   0
13394  060764     000      000               .BYTE   0,0
13395  060766  057174                        .WORD   DH002A
13396  060770     003      000               .BYTE   3,0
13397  060772  057420                        .WORD   DH003B
13398  060774     006      000               .BYTE   6,0
13399
13400  060776  000004                DF010:  .WORD   4
13401  061000     000      000               .BYTE   0,0
13402  061002  000000                DF010A: .WORD   0
13403  061004     000      000               .BYTE   0,0
13404  061006  057174                        .WORD   DH002A
13405  061010     003      000               .BYTE   3,0
13406  061012  060050                        .WORD   DH010A
```

```
13407  061014      002      000                    .BYTE   2,0
13408
13409  061016  000004                    DF011:  .WORD   4
13410  061020      000      000                    .BYTE   0,0
13411  061022  000000                    DF011A: .WORD   0
13412  061024      000      000                    .BYTE   0,0
13413  061026  057174                            .WORD   DH002A
13414  061030      003      000                    .BYTE   3,0
13415  061032  057420                            .WORD   DH003B
13416  061034  000006  000000                    .WORD   6,0
13417
13418  061040  000002                    DF015:  .WORD   2
13419  061042      003      000                    .BYTE   3,0
13420  061044  060263                            .WORD   DH015
13421  061046      003      000                    .BYTE   3,0
13422
13423  061050  000001                    DF016:  .WORD   1
13424  061052      003      000                    .BYTE   3,0
13425
13426  061054  000052                    BS26:   .BLKW   52
13427  061200  000052                    BS24:   .BLKW   52
13428           070376                            .=70376
13429  070376  000240                            NOP
13430  070400  005014                    SPCHLR: CLR     (R4)            ;CLEAR MEM MAINT REG
13431  070402  005237  001662                     INC     INTSET          ;COUNT THE INTERRUPT
13432  070406  000240                            NOP
13433  070410  000002                    SPCPAR: RTI                     ;RETURN
13434  070412  000240                            NOP
13435  070414  001000                    IBUFF:  .BLKW   1000
13436  072414  001000                    OBUFF:  .BLKW   1000
13437           000001                    .END
```

```
ABASE = 177440          1967#   2289    2330
ABORT   053227          9590   12793#
ABTFAI  044170          9385   11412#
ACDW1 = 000000          2289    2332
ACDW2 = 000000          2289
ACLO  = 000010          2062#  10506
ACLOER= 000100         10508   12377#
ACPUOP= 000000          2289    2304
ADDW0 = 000000          2289
ADDW1 = 000000          2289
ADDW10= 000000          2289
ADDW11= 000000          2289
ADDW12= 000000          2289
ADDW13= 000000          2289
ADDW14= 000000          2289
ADDW15= 000000          2289
ADDW2 = 000000          2289
ADDW3 = 000000          2289
ADDW4 = 000000          2289
ADDW5 = 000000          2289
ADDW6 = 000000          2289
ADDW7 = 000000          2289
ADDW8 = 000000          2289
ADDW9 = 000000          2289
ADEVCT= 000000          2289    2295
ADEVM = 000000          2289    2331
ADJCLK  043614          3268   11305#
AENV  = 000000          2289    2300
AENVM = 000000          2289    2301
AFATAL= 000000          2289    2292
AMADR1= 000000          2289    2317
AMADR2= 000000          2289    2321
AMADR3= 000000          2289    2324
AMADR4= 000000          2289    2327
AMAMS1= 000000          2289    2311
AMAMS2= 000000          2289    2319
AMAMS3= 000000          2289    2322
AMAMS4= 000000          2289    2325
AMSGAD= 000000          2289    2297
AMSGLG= 000000          2289    2298
AMSGTY= 000000          2289    2291
AMTYP1= 000000          2289    2312
AMTYP2= 000000          2289    2320
AMTYP3= 000000          2289    2323
AMTYP4= 000000          2289    2326
APASS = 000000          2289    2294
APRIOR= 000240          1966#   2289
APTCSU= 000040          9443#  11445
APTENV= 000001          9142    9399    9441#   9498   11438
APTSIZ= 000200          2803    9440#
APTSPO= 000100          9401    9442#  11440
ASWREG= 000000          2289    2302
ATESTN= 000000          2289    2293
AUNIT = 000000          2289    2296
AUSWR = 000000          2289    2303
AVECT1= 000210          1965#   2289    2328
```

```
AVECT2= 000000  2289  2329
BAERR = 000004  10309  12410#
BAI   = 000020  2025#  4839  7489  7504  7590  7663  7720  10277
BA16  = 000400  2010#  5073  5159  5499  5552
BA17  = 001000  2011#  5219  5310  5499  5552
BDSRCK  042160  10880#  11049
BITCNT  036242  8760  10002#  10651  10661  10669  10681  10691  10702
BIT0  = 000001  1915#  2007  2040  2059  2701  3061  3093  3697  3936  4442  4630  4632  4634
                5649  7045  8872  9056  10161  12371  12389  12398  12408  12457
BIT00 = 000001  1905#  1915
BIT01 = 000002  1904#  1914
BIT02 = 000004  1903#  1913
BIT03 = 000010  1902#  1912
BIT04 = 000020  1901#  1911
BIT05 = 000040  1900#  1910
BIT06 = 000100  1899#  1909
BIT07 = 000200  1898#  1908
BIT08 = 000400  1897#  1907  9267
BIT09 = 001000  1896#  1906  9285  9509
BIT1  = 000002  1914#  2041  2702  3697  3938  4688  8883  9067  9664  9665  12372  12390  12399
                12409  12458
BIT10 = 002000  1895#  2012  2031  2050  2083  2097  2110  2124  2138  2708  9463  11173  12381
BIT11 = 004000  1894#  2013  2032  2051  2068  2084  2098  2111  2125  2139  2709  9292  11255
                12382
BIT12 = 010000  1893#  2015  2033  2052  2085  2099  2112  2126  2140  11092  11100  11118  11142
                11146  11170  11280  12383
BIT13 = 020000  1892#  2016  2034  2053  2069  2086  2100  2113  2127  2141  9470  9621  11092
                11100  11118  11142  11170  11280  12384
BIT14 = 040000  1891#  2017  2035  2054  2070  2087  2101  2114  2128  2142  4370  4371  6774
                6775  9253  11022  11067  11083  11135  12385  12463
BIT15 = 100000  1890#  2018  2019  2036  2055  2071  2088  2147  2710  4367  4368  4520  7155
                7293  11022  11044  11065  12386  12464
BIT2  = 000004  1913#  2042  2061  2703  5647  9675  12373  12391  12400  12410  12459
BIT3  = 000010  1912#  2024  2043  2062  2704  12374  12392  12401  12411  12460
BIT4  = 000020  1911#  2025  2044  2063  2077  2118  2132  3179  12375  12393  12402  12412  12461
BIT5  = 000040  1910#  2026  2045  2064  2078  2092  2105  2119  2133  3179  12376  12394  12403
                12413  12462
BIT6  = 000100  1909#  2008  2027  2046  2065  2079  2093  2106  2120  2134  2705  5680  9724
                10104  12377  12395  12404
BIT7  = 000200  1908#  2009  2028  2047  2066  2080  2094  2107  2121  2135  2706  2828  3048
                3165  8588  9634  10999  12378  12405
BIT8  = 000400  1907#  2010  2029  2048  2067  2081  2095  2108  2122  2136  2707  3141  3219
                3475  8595  8624  8694  8856  8866  8889  8897  9040  9050  9073  9081  11005
                12379
BIT9  = 001000  1906#  2011  2030  2049  2082  2096  2109  2123  2137  4572  4573  4627  4628
                11005  11020  11037  12380
BPTVEC= 000014  1922#
BSE   = 000200  2047#  7441  7600  10500  10547
BSERPT= 000200  2706#  7443  7445  7603  7605
BSERR = 000100  6806  6882  10549  12395#
BSF24P  001636  2683#  4162  11047
BSF26P  001640  2685#  4064  11043
BSS24P  001642  2687#  4168*  4202  11071
BSS26P  001644  2689#  4153*  4194  11068
BS24    061200  2683  4198  13427#
BS24CT  001650  2692#  4173*
```

```
BS26    061054          2685    4176   13426#
BS26CT  001646          2691#   4160*
CCLR  = 100000          2019#   3509    3563    8026    8302   10150   10177
CDT   = 002000          2012#   3082    3151    9948
CERNER= 100000         10445   12386#
CERR  = 100000          2018#   3075    3125    4285   10165   10221   10225   10443   10518
CFMT  = 010000          2015#   6534    6563    6621    6636    6694    6709    8383   10059   10360   11017   11045   11069
                       11076
CHKBA   037430         10274#  10468
CHKCTS  037642          6810   10336#  10469
CHKOP   040162         10426#  12084
CHKST   040170         10424   10429#
CHKWC   037376         10255#  10466
CHKWE   040130         10417#  12085
CKSWR = 104407          9247    9459    9508   12074#
CLEAR = 000105          1992#   3597    7639    7699    7775    8129    8233    8272    8368    8502    8951    8971
CLKCAL  035246          9761#   9856
CMNDLB  050120          9867   10472   10764   12470#
COE   = 001000          2049#
COERR = 001000          6509   12380#
CP1170= 002000          2708#   5631    5645    5661    5684    5691    9656    9712
CR    = 000015          1830#  11484   11494   12596   12599
CRLF  = 000200          '831#   2827   11455   11494
CSRPTR  001672          2714#   5633*   5647*   5649*   5654    5686*   5688*   5693*   5695*   9655*   9696*
CS1ERB= 024000         10435   12349#
CS2ERB= 177400         10437   12350#
CTO   = 004000          2013#  10167   10557
CTOERR= 000002          7816   10559   12399#
CTRHLT  043724          2957    2986    3163   11333#  11680   11763
CYLERR= 000010         10386   12411#
DCK   = 100000          2055#  10500   10528
DCKERR= 000002          8042   10530   12390#
DDISP = 177570          1837#   2236    2791
DDT   = 000400          2067#
DESHLD  001700          2721#  11085   '1109*  11161*  11285
DF001   060612          2383   13334#
DF002   060616          2398   13337#
DF003   060636          2409   13347#
DF004   060646          2490   13352#
DF004A  060652         10696   10707   13354#
DF005   060676          2504   13365#
DF005A  060702         10716   13367#
DF006   060732          2529   13380#
DF006A  060736         10674   10686   13382#
DF007   060756          2555   13391#
DF007A  060762         10648   13393#
DF010   060776          2578   13400#
DF010A  061002          5515*   5568*   5705*   6816*   6823*   7124*   7131*   7142*   7148*   7262*   7269*   7280*   7286*
                       10472*  13402#
DF011   061016          2592    2601    2610    2618   13409#
DF011A  061022          3250*   3337*   3353*   3388*   3399*   3435*   3486*   3491*   3497*   3503*   3519*   3525*   3542*
                        3548*   3554*   3560*   3582*   3588*   361.*   3620*   3750*   3765*   3772*   3789*   3818*   3830*
                        3859*   3879*   3886*   4281*   4288*   8671*   8738*   8810*   8827*   9011*  10128*  13411#
DF015   061040          2630   13418#
DF016   061050          2638   13423#
DH001   057146          2381   13179#
```

```
DH002A  057174          2407   2628  13183#  13339  13356  13369  13384  13395  13404  13413
DH002B  057222         13187#  13341  13358  13371  13386
DH002C  057307         13196#  13343  13360  13373  13388
DH002D  057324         13199#  13362  13375
DH003B  057420         13209#  13349  13397  13415
DH005   057476         10757   13217#
DH005A  057724         13244#  13377
DH006   057573         10760   10817  13228#
DH007   057655         10755   13237#
DH010   060007          2576   13253#
DH010A  060050         13259#  13406
DH010B  060063         13261#
DH011   060133          2590   13268#
DH012   060162          2599   13272#
DH013   060213          2608   13277#
DH014   060236          2616   13281#
DH015   060263         13285#  13420
DH016   060314          7931   13290#
DH017   060411          8819   13301#
DH018   060434          8827   13305#
DH2N    001372          2396#   7931*  8819*  9867* 10086* 10114*
DH4N    001412          2488#  10697  10708
DH5N    001422          2502#  10717
DH6N    001432          2527#  10675  10687
DH7N    001442          2553#  10649
DI    = 040000          2017#   3488   3522   3545   3585   3617   3815   3856   3883   8958   8978  10221
DISPLA  001142          2236#   2791*  2799*  9307*  9462*
DISPRE  000174          2177#   2799
DLT   = 100000          2036#   4278  10535
DLTERR= 000010         10537   12392#
DMD   = 000040          2078#   3650   3658   3697   3706   8103   8185   8193   8293   8416  11316
DRA   = 000001          2059#
DRDY  = 000200          2066#   3097   3102   3127   8557
DROT  = 000040          2064#  10514
DROTER= 000400          8321   8455  10516  12379#
DRPAR = 000010          2043#
DRPARE= 000010          3470  12374#
DRSTER= 040000         10712  10718  10852  12463#
DRVABT  057020          4206  13162#
DRVBIT  001630          2679#   3188*  3254   3256*  4095   4096   4145   4146   8588   8592*  8593   8595   8603*
DRVDRP  001666          2712#   2915*  3061*  3093*  3131*  3190   3221   4096*  4146*
DRVMSK= 000007          2023#
DRVNUM  001626          2678#   3197*  3262*  3263   3284   3324   3374   3391   3417   3461   3477   3639   3687
                        3739   3805   3847   3911   3928   3974   3991   6536   6564   6920   7433   8552   8591*
                        8600   8604*  8783*  8796*  8818*  8938*  8960*  8981*  8991*  9007*  9900  10067  11308
                       13314  13317  13323  13331
DRVRPT= 004000          2709#   2916   3052   3157   3175
DSC   = 040000          2070#   3247   3494   3551   3769
DSERBI= 000070         10439  12353#
DSWR  = 177570          1836#   2235   2790
DTE   = 010000          2052#
DTERR = 010000          8114  12383#
DTYE  = 000040          2045#   3079   3148   9943
DTYERR= 000040         12376#
DTYPE   001720          2730#   3036*  3082*  3109*  3136   3143*  3151*  3993   6535   6922   7435   8555   9929
                        9938   9963*  9968* 10074
```

```
DT001   060520            2382   13317#
DT002   060526            2397   13318#
DT003   060476            2408   13314#
DT004   060526            2489   13319#
DT005   060526            2503   13320#
DT006   060526            2528   13321#
DT007   060526            2554   13322#
DT010   060526            2591    2600    2609    2617   13323#
DT015   060576            2577    2629   13331#
DT015A  060604            2637   13332#
DOTST = 000001            2701#   3062    3092    3104    3111
ECCW  = 020000            2086#
ECH   = 000100            2046#  10500   10524
ECHERR= 000001            8042   10526   12389#
EMCCLR  056560            3519    3525   13125#
EMCERR  056733            4287   13152#
EMDA    056540            3485    3518    3541    3581    3613    8670    8737    9010   13122#
EMDCLR= 050206            3614    3620   13129#
EMDI    056474            3490    3524    3547    3587    3619    3817    3858    3885   13114#
EMDLT = 054541            4280   10127   13147#
EMDPAR= 054765            3486    3491    3497    3503    3542    3548    3554    3560   13117#
EMDSC   056514            3249    3496    3553    3771   13118#
EMDTPE= 055050            3435   13113#
EMMI    056774            3829   13158#
EMNZPR  056371            3353    3582    3588   13101#
EMRCAL= 050247            3750   13130#
EMRDB   056707            4281    4288   10128   13148#
EMRSET  056701            3879    3886   13145#
EMSCLR  056754            3250    8738   13155#
EMSELD= 050160            3337    3388    3789   13128#
EMSK  = 050272            9011   13137#
EMSKSF  056622            3818    3830    3859   13134#
EMSVAL  056354            3336    3352    3387    3398    3434    3502    3559    3749    3764    3788    8809    8826   13098#
EMTVEC= 000030            1925#   2774*   2775*
EMUR    056664            8671   13142#
EMUXIT  056637            3878   13138#
EMWCS2  056432            3399   13107#
EM1     053315            2955    2984   12805#
EM1N    001360            2380#   2955*   2984*   4093*   4104*   4114*   4121*   4130*   4143*   4206*   7446*   7606*
EM10    054006            5532    5586    5720    7130    7147    7268    7285   12415   12861#
EM10N   001450            2575#   5514*   5525*   5532*   5567*   5579*   5586*   5706*   5720*   6813*   6822*   7121*   7130*
                          7139*   7147*   7259*   7268*   7277*   7285*  10479*
EM11    054033            5525    5579    5706    7121    7139    7259    7277   12417   12865#
EM11A   054061            5514    5567   12416   12869#
EM11N   001460            2589#   3336*   3387*   3434*   3485*   3490*   3496*   3502*   3541*   3547*   3553*   3559*   3749*
                          3771*   3788*   3817*   3858*   8809*   9010*
EM12    054135           12418   12877#
EM12N   001470            2598#   3249*   3352*   3398*   3613*   3619*   3764*   3885*   8670*   8737*
EM13    054170            4280*   6813   12419   12882#
EM13N   001500            2607#   3829*   3878*   4287*  10127*
EM14    054220            6822   12420   12886#
EM14N   001510            2615#   3518*   3524*   3587*   8826*
EM15    054252           12422   12891#
EM15N   001520            2627#  10998*  11001*
EM16    054275           12423   12895#
EM17    054320           12424   12899#
```

EM18	054341	12425	12902#									
EM19	054366	12426	12906#									
EM2	053375	12814#										
EM2INT	056601	3765	3772	13131#								
EM2N	001370	2395#	3758*	9868*	10085*	10113*						
EM20	054410	12427	12909#									
EM21	054434	12428	12913#									
EM22	054460	12429	12917#									
EM23	054501	12431	12920#									
EM24	054512	12432	12922#									
EM25	054525	12433	12924#									
EM26	054541	12434	12927#	13147								
EM27	054553	12435	12929#									
EM28	054600	12436	12933#									
EM29	054613	12437	12935#									
EM3	053456	9623	12823#									
EM3N	001400	2406#	3215*	3229*	3243*	3304*	10227*	10229*	10240*			
EM30	054634	12439	12938#									
EM31	054707	12440	12946#									
EM32	054727	12441	12949#									
EM33	054765	12442	12955#	13117								
EM34	055033	12443	12962#									
EM35	055050	12444	12965#	13113								
EM36	055071	12445	12968#									
EM37	055100	12446	12970#									
EM38	055123	12447	12974#									
EM39	055143	12448	12977#									
EM4	053526	9868	10085	12830#								
EM4N	001410	2487#	3581*									
EM40	055165	12449	12981#									
EM41	055217	12450	12986#									
EM42	055240	12451	12989#									
EM43	055263	12452	12993#									
EM44	055325	12453	12999#									
EM45	055342	12454	13002#									
EM46	055402	3243	13009#									
EM47	055437	3215	13014#									
EM48	055520	3229	13023#									
EM49	055577	3304	13031#									
EM5	053621	10227	12840#									
EM5N	001420	2501#										
EM50	055647	3758	13038#									
EM51	055746	4093	4104	4114	4121	13049#						
EM52	056041	4130	13060#									
EM53	056137	4143	13072#									
EM54	056222	10998	13081#									
EM55	056251	11001	13085#									
EM56	056276	10113	13090#									
EM6	053665	10229	12847#									
EM6N	001430	2526#										
EM7	053744	10240	12855#									
EM7N	001440	2552#										
ERRCNT	001632	2680#	9585*	9588								
ERRLMT	001634	2682#	4310	4806	4879	4960	5029	5110	5179	5259	5330	5404	5475	5816
		5904	5987	6072	6157	6216	6273	6331	6387	6443	6581	6656	6729	6854
		6892	7532											

```
ERRVEC= 000004          1918#   2788    2789*   2800*   9173    9174    9178*   9183*   9195*   9200*   9209*   9219*   9220*
                        9258    9259*   9261*   9264*   9643    9644    9645*   9646*   9692*   9719*   9734*   9735*
EXPBA   047750         10279*  10306*  12358#
EXPCYL  047752         10365*  10379*  10384   12359#
EXPSEC  047754          6824   10383*  12360#
EXPTRK  047756          6814   10382*  12361#
EXPUBA  047746         10305*  10313   12357#
EXPWC   047744         10255*  10257   10480   12356#
FMTE  = 000020          2044#
FMTERR= 000020         12375#
FPFMT = 000400          2707#   7373
GENCOM  042260          4237    4305    4313    4363    4438    4517    4569    4623    4684    4745    4774    4800    4809
                        4840    4874    4882    4925    4938    4954    4963    5000    5023    5032    5067    5086    5103
                        5113    5148    5172    5182    5225    5252    5262    5299    5323    5333    5362    5382    5398
                        5407    5436    5454    5469    5478    5635    5765    5776    5810    5819    5849    5872    5898
                        5907    5935    5951    5981    5990    6020    6032    6066    6075    6105    6117    6151    6160
                        6187    6191    6210    6219    6245    6249    6267    6276    6302    6306    6325    6334    6358
                        6362    6381    6390    6414    6418    6437    6446    6542    6576    6584    6612    6616    6650
                        6659    6685    6689    6723    6732    6771    6784    6788    6848    6857    6864    6886    6895
                        6928    7000    7049    7185    7314    7387    7474    7526    7535    7988    8007    8516    8567
                       10992#  11382
GETDRS  036762         10149#  10713
GETRK   036164          9978#  12083
GNS   = ****** U        2176    2826    9116    9123   12066   12067   12068   12069   12070   12072   12074   12075   12076
                       12077   12078   12079   12080   12081   12082   12083   12084   12085   12086   12087   12088   12089
                       12090   12091   12092   12093   12094   12095   12096   12097   12098   12099
GO    = 000001          2007#
GPSUMF  050116         10434*  10647   10818*  12456#
GP1NR = 000010         10705   10710   10758   10812   10849   12460#
GP2NR = 000020         10684   10758   10812   10845   12461#
GP3NR = 000040         10664   10758   10812   12462#
GRP1ER  047774         10430*  10445*  10492   10598   10690   10800   12370#
GRP1MS  050056         10801   10832   12439#
GRP1ST= 000001         10694   10710   10766   10814   10849   12457#
GRP2ER  047776         10431*  10522   10668   10794   12388#
GRP2MS  050040         10795   10826   12431#
GRP2ST= 000002         10672   10766   10814   10845   12458#
GRP3ER  050000         10432*  10551   10650   10771   12397#
GRP3MS  050020         10772   10821   12422#
GRP3ST= 000004         10655   10766   10814   12459#
GRP4ER  050002          6809*   6811    6820   10259*  10309*  10315*  10386*  10389*  10392*  10433*  10470   10473   12407#
GRP4MS  050004         10479   12415#
GTSWR = 104406          2821   12072#
GTYP0   036134          3198    8629    8699    8784    8940    8983    9962#
GTYP1   036150          3264    8601    8797    8962    8993    9967#
HOLD    036124          7434*   7435*   7436    8554*   8555*   8556    9954#
HOLD1   036126          8553*   8559*   9955#
HOLD2   036130          3419    9942*   9948*   9956#
HOLD3   036132          3711    9943*   9949*   9957#
HT    = 000011          1828#   11453   11494
HVRC  = 000400          2048#   10500   10543
HVRCER= 000040         10545   12394#
IBUFF   070414          2837    4074    4128    4134    4256    4788    4854    4945    5093    5240    5377    5450    5783
                        5798    5867    5886    5958    5972    6039    6054    6124    6139    6198    6255    6313    6369
                        6425    6471    6499    6638    6711    6836    6871    6954    7512    7953    8031   10062   10117
                       11080   11141   13435#
```

```
IDAE   = 002000          2050#
IDAERR= 002000          3942    3950    4011   12381#
IE     = 000100          2008#   2974    3872    7821    7924    8122    8494
ILF    = 000001          2040#  10500   10581
ILFERR= 000200         10583   12405#
INTHLR  034442          2921    3045    5735    9613#
INTR   = 000300          2003#
INTSET  001662          2697#   2948*   2953    2973*   2978    3745*   3827    3863*   3918*   3981*   5656    7432*   7469*
                        7627*   7688*   7768*   7820*   7857*   7926*   8021    8080*   8123*   8227*   8263*   8360*   8487*
                        8636*   8705*   8793*   8813*   8946*   8968*   9604*   9614*   9720*   9728    9733*   9766    9789
                        9857    9912*  10072*  10077   11357*  13431*
IOTVEC= 000020          1923#   2772*   2773*
IR     = 000100          2027#
IWAT1M  035370          9813#  12099
IWAT1S  035410          9819#  12096
IWAT11  035450          9831#  12092
IWAT12  035440          9828#  12093
IWAT14  035430          9825#  12094
IWAT15  035420          9822#  12095
IWAT16  035530          9849#  12086
IWAT2S  035400          9816#  12097
IWAT32  035520          9846#  12087
IWAT48  035510          9843#  12088
IWAT64  035500          9840#  12089
IWAT8S  035360          9810#  12098
IWAT80  035470          9837#  12090
IWAT96  035460          9834#  12091
KIPAR0= 172340          1956#   9155    9187
KIPAR1= 172342          1957#
KIPAR2= 172344          1958#
KIPAR3= 172346          1959#
KIPAR4= 172350          1960#
KIPAR5= 172352          1961#
KIPAR6= 172354          1962#  11145*
KIPAR7= 172356          1963#   9203
KIPDR0= 172300          1945#
KIPDR1= 172302          1946#
KIPDR2= 172304          1947#
KIPDR3= 172306          1948#
KIPDR4= 172310          1949#
KIPDR5= 172312          1950#
KIPDR6= 172314          1951#
KIPDR7= 172316          1952#
KWLADD  001710          2725#   5664*   5680*   9634*   9724*  10057*  10104*  11314*
KWLVEC  001712          2726#   9721
LCKHLR  034506          9633#   9722
LCLKPR= 100000          2710#   9731    9762    9859   10055   10102   11312
LCLKTK  001674          2716#   3867*   3869    9633*   9761*   9768    9770*   9795*   9862   10073*  10080
LDCON   036034          3308    9938#
LF     = 000012          1829#  11488   11494   12596   12599
LOADRK  035734          9912#  12082
LRLOAD  035674          4071    4229    4257    4354    4382    4430    4463    4509    4529    4561    4582    4615    4643
                        4676    4702    4737    4766    4785    4831    4851    4918    4942    4993    5010    5072    5090
                        5141    5158    5218    5237    5292    5309    5354    5373    5429    5447    5498    5551    5623
                        5757    5780    5795    5841    5864    5883    5927    5955    5969    6012    6036    6051    6097
                        6121    6136    6195    6252    6310    6366    6422    6468    6496    6620    6635    6693    6708
```

```
                    6763   6792   6833   6868   6951   6992   7011   7053   7068   7088   7189   7204   7226
                    7318   7335   7379   7413   7458   7481   7509   7583   7616   7638   7653   7678   7698
                    7713   7753   7774   7789   7801   7845   7864   7879   7902   7950   7992   8011   8028
                    8069   8084   8096   8128   8160   8178   8216   8232   8247   8271   8286   8307   8345
                    8367   8382   8397   8409   8440   8463   8501   8519   8570   9892#  11342  11371
LSTCYL  036120      4069   6467   6495   7613   7614   7751   9940*  9946*  9952#
L.ASOF  001612      2667#  8254*
L.BA    001604      2662#  6539*  6924*  8996*  10279  10294
L.CS1   001600      2660#  3067*  3118*  3199*  3232*  3285*  3325*  3375*  3418*  3419*  3462*  3509*  3528*
                    3563*  3569*  3597*  3640*  3688*  3740*  3776*  3806*  3848*  3910*  3929*  3973*  3992*
                    6534*  6535*  6563*  6921*  6972*  8630*  8700*  8785*  8798*  8941*  8951*  8963*  8971*
                    8985*  8997*  9895   9929*  9930   10059  10074*  10075  10213  10283  10360  11017  11045
                   11069  11076
L.CS2   001610      2666#  3065*  3117*  3196*  3284*  3294*  3324*  3374*  3417*  3461*  3639*  3649*  3687*
                    3698*  3739*  3805*  3847*  3911*  3928*  3974*  3991*  4839*  6536*  6564*  6920*  7489*
                    7504*  7590*  7663*  7720*  8186*  8628*  8698*  8782*  8795*  8939*  8961*  8982*  8992*
                   10159  10277
L.DA    001606      2663#  10069
L.DCYL  001614      2668#  4003*  6540*  6566*  6926*  8984*  8994*  10068  10365  10886  11009  11381*
L.DS    001606      2664#  6959*  10368
L.DT    001607      2665#  3930*  4321   6538*  6565*  6923*  10366  10889  11008  11392  11394*
L.MR1   001616      2669#  3326*  3340*  3463*  3507*  3571*  3595*  3650*  3697*  7871*  7910*  8103*  8185*
                    8293*  8314*  8416*  8448*  8471*  8499*
L.WC    001602      2661#  6537*  6925*  8995*  9915   10281  10337
MCLK  = 000400      2081#  10033  10034
MCLOCK  036316      3654   3702   8106   8190   8297   8419   8424   8429   10024#
MDS   = 001000      2030#  10565
MDSERR= 000010     10567   12401#
MEMBAS  001714      2727#  9660
MEMPAR  001670      2713#  9647*  9667*  9705*  9710
MEMPYR= 000004      2703#  5616   5619
MEMSZB= 000002      2702#  4905   4910   5053   5059   5204   5210
MERD  = 001000      2082#
MESMSK= 000017      2075#
MEWD  = 002000      2083#
MILCNT  001652      2693#  9788
MIND  = 000200      2080#  8422   8427
MMVEC = 000250      1934#
MMVECA  001716      2728#  5658*  9670   9709
MSP   = 000100      2079#
MYTIME  035314      3868   9787#  9861   10079
M.CADD= 017760      2150#
M.CDIF= 017760      2149#
M.DRV = 000007      2146#
M.HEAD= 007000      2153#
M.ID  = 000003      2148#
M.PAR = 100000      2147#
M.SECT= 000760      2152#
M.SER = 017770      2151#
NCERWE= 020000     10520   12384#
NED   = 010000      2033#  3077   3146   10553
NEDERR= 000001     10555   12398#
NEM   = 004000      2032#  10573
NEMERR= 000040      5512   5565   10575  12403#
NEWDRV  027716      4097   4147   8586#
NEWPAS  003150      2925#  9148
```

```
NEXINT  034434          2945   9604#  9719
NXF   = 000004          2042#
NXFERR= 000004         12373#
OBUFF   072414          4232   4357   4366   4385   4433   4441   4466   4512   4520*  4532   4564   4572*  4573*
                        4585   4618   4626   4646   4679   4687   4705   4740   4769   4834   4860   4861   4996
                        5144   5295   5357   5393   5432   5464   5626   5644*  5648*  5687*  5694*  5707   5709
                        5714   5716   5760   5844   5930   5946   6015   6100   6539   6623   6696   6766   6774*
                        6775*  6795   6924   6995   7014   7056   7071   7085*  7091   7118   7122   7137   7140
                        7192   7207   7222*  7229   7256   7260   7275   7278   7321   7333*  7338   7382   7416
                        7430   7484   7581*  7586   7656   7661*  7716   7882   7995   8014   8289   8412   8522
                        8573   8996   9673  11023  11060*  11061*  11081  11140  11374  13436#
OFFSET= 000115          1996#  8248
OFST  = 000004          2061#
OPER00  050160          8810  12470  12488#  13128
OPER02  050175         12471  12491#
OPER04  050206         12472  12493#  13129
OPER06  050222         12473  12495#
OPER10  050231         12474  12497#
OPER12  050247         12475  12500#  13130
OPER14  050263         12476  12503#
OPER16  050272         12477  12505#  13137
OPER20  050277         12478  12506#
OPER22  050311         12479  12508#
OPER24  050324         10086  12480  12510#
OPER26  050341         12481  12513#
OPER30  050357         12482  12516#
OPER32  050373         12483  12519#
OPER34  050417         12484  12523#
OPER36  050443         12485  12527#
OPER37  050467          6816   6823  12531#
OPER40  050523         12536#
OPER41  050523          7124   7131   7142   7148   7262   7269   7280   7286  12537#
OPER42  050560          5515  12542#
OPER43  050634          5568  12550#
OPER44  050707          5705  12558#
OPI   = 020000          2053#  10500  10539
OPIERR= 000020         10541  12393#
OPR001  050766          2868  12570#
OPR002  051015          2871   2882   2899  12574#
OPR003  051023          2878  12576#
OPR004  051053          2890  12581#
OPR005  051075          2833  12585#
OPR006  051146          3064  12592#
OPR007  051146          2844   3060   3091  12595#
OPR008  051227          3162  12607#
OPR009  051312          3167  12616#
OPR010  051362          5618  12623#
OPR011  051503          4908   5056   5207  12638#
OPR012  051646          4909  12656#
OPR013  051652          5057  12657#
OPR014  051656          5208  12658#
OPR015  051662          4911   5058   5209  12659#
OPR016  051676          2863  12661#
OPR017  052457          7446   7606  12729#
OPR018  052574          8766  12743#
OPR019  052662         11305  12752#
```

```
OPR020  053025          11369   12769#
OPR021  053121          11397   12780#
OPR022  053162          11413   12786#
OPTFLG  001664           2699#   2861    2864*   2916*   3052    3062*   3092*   3104*   3111    3157    3175*   4905    4910*
                         5053    5059*   5204    5210*   5614    5616    5619*   5631    5645    5661    5684    5691    7373
                         7443    7445*   7603    7605*   9648*   9656*   9657*   9679*   9697*   9704*   9712    9731*   9762
                         9859   10055   10102   11312
OPTTST  034522           2917    3873    5736    6681    8663    9642#
OR    = 000200           2028#
PACK  = 000103           1991#   3232
PARM    001752           2183    2749#
PARPRE= 000100           2705#   5614    9648    9657    9679    9697    9704
PAT   = 000020           2077#   3463    7871
PAT02   047144          11195   12107#
PAT03   047204          11200   12127#
PAT04   047244          11205   12147#
PAT05   047304          11210   12167#
PAT06   047344          11215   12187#
PAT10   047404          11220   12210#
PAT11   047444          11225   12230#
PAT12   047504          11230   12250#
PAT13   047544          11235   12270#
PAT14   047604          11240   12290#
PAT15   047644          11245   12310#
PAT16   047704          11250   11253   12330#
PCA   = 004000           2084#
PCD   = 010000           2085#
PERHLR  034452           9621#   9714
PGE   = 002000           2031#  10569
PGERR = 000020           7968   10571   12402#
PIP   = 020000           2069#
PIRQ  = 177772           1835#
PIRQVE= 000240           1929#
PROGEN  044124          11341   11368   11393   11397#  11414
PR0   = 000000           1852#   2925    3040    9736    9851
PR1   = 000040           1853#
PR2   = 000100           1854#
PR3   = 000140           1855#
PR4   = 000200           1856#
PR5   = 000240           1857#   1966
PR6   = 000300           1858#
PR7   = 000340           1859#   2758    2946    3046    5639    9646    9672    9715    9723
PS    = 177776           1832#   1833
PSW   = 177776           1833#
PWRVEC= 000024           1924#   2778*   2779*  11996*  11997*  12006*  12012*  12024*  12025*
RDCHR = 104410          11839   12075#
RDDATA= 000121           1998#   4072    4786    4852    4943    5011    5091    5159    5238    5310    5375    5448    5552
                         5781    5796    5865    5884    5956    5970    6037    6052    6122    6137    6196    6253    6311
                         6367    6423    6469    6497    6636    6709    6834    6869    6952    7510    7951    8029
RDGATE= 100000           2088#
RDHEAD= 000125           2000#   4258    6563    7802    8097    8383
RDLIN = 104411          11913   12076#
RDOCT = 104412           2872    2883    2900   12077#
RDSTHD  036362           4298    6568   10054#
RDY   = 000200           2009#   2974   10238   11310
REALBA  047764          10310*  12365#
```

```
REALCY  047766  12366#
REALSE  047772  6825  12368#
REALTR  047770  6815  12367#
REALUB  047762  10316*  12364#
REALWC  047760  10260*  10393  10481  12363#
RECAL = 000113  1995#  3740  3910  3973  7790  7846  8085  8217  8346  8630  8700  8785  8941
                8963  11343
REFMT   001676  2717#  2763*  4223*  4350*  4426*  4505*  4557*  4611*  4671*  4730*  4814*  6528*  6608*
                6680*  6759*  6975*  8341*  8584*  11367  11379
REPNR = 100000  10818  12464#
RESREG= 104414  9584  9740  9875  10136  10186  10400  10482  10486  10840  10862  12079#
RESTRT  001762  2181  2752#
RESVEC= 000010  1919#
RKASOF= 000016  1978#  8639  8708
RKBA  = 000004  1973#  7430*
RKCS1 = 000000  1971#  2949*  2974*  3872*  7436*  7821*  7924*  8026*  8122*  8302*  8494*  8556*  8638*
                8707*  10075*  10089  10150*  10161*  10165  10167  10177*  10220  11309*  11310
RKCS2 = 000010  1975#  2971*  3391*  7433*  8552*  8727*  10067*  10096  10159*  10169  10218*  11307*  11308*
RKDA  = 000006  1974#  10069*
RKDB  = 000024  1980#  4255  10066
RKDCYL= 000020  1979#  10068*
RKDS  = 000012  1976#  8557
RKECPS= 000030  1984#
RKECPT= 000032  1985#
RKER  = 000014  1977#  7441  7600
RKMR1 = 000026  1981#  3658*  3706*  8109*  8193*  8300*  8422*  8427*  8432*  10033*  10034*  10160*  11316*
                11320*
RKMR2 = 000034  1982#  10175
RKMR3 = 000036  1983#  10176
RKPRI   001622  2673#  2922*  5651
RKSPAR= 000022  1986#  7960*
RKVEC   001620  2672#  2920*  3044  5657*  5735*
RKWC  = 000002  1972#  7431*  8023
RLS   = 000010  2024#  3294  3649
SAVREG= 104413  9531  9642  9810  9813  9816  9819  9822  9825  9828  9831  9834  9837  9840
                9843  9846  9849  10054  10149  10336  10417  10426  12078#
SCLR  = 000040  2026#  8727  10218  11307
SCOP1 = 104415  3946  3954  4084  4251  4283  4290  4319  4403  4484  4936  6552  6575  6591
                6828  6938  6968  7083  7106  7153  7219  7244  7291  7428  7453  7499  7524
                7542  8830  9014  12080#
SCOP1$  035222  9749#  12080
SECERR= 000040  6820  10392  12413#
SEEK  = 000117  1997#  3806  3848  3929  3992  7459  7617  7679  7754  7865  8070  8985
SELDRV= 000101  1990#  3067  3118  3199  3285  3325  3375  3418  3462  3528  3569  3640  3688
                3776  7903  8161  8179  8308  8398  8441  8464  8798
SETCLK  001742  2185  2746#
SPI   = 000002  2041#
SKIERR= 000002  3950  7892  12372#
SPACE2  050763  3182  9567  9579  12569#
SPAR  = 020000  2016#  10167  10502
SPARER= 000001  3470  10504  12371#
SPCHLR  070400  5657  13430#
SPCPAR  070410  5658  13433#
SPDERR= 000200  10512  12378#
SPDLSS= 000020  2063#  10510
SRCHLD  001702  2722#  11086  11110*  11162*  11286
```

```
SRTFLG  001624          2674#  2746*  2749*  2752*  2755*  2866   3058   3266
SRTINS= 000010          2704#  2861   2864
SRTSPL= 000111          1994#
SR0   = 177572          1938#  9184   9199*  9206*  11094* 11102* 11120* 11172* 11282*
SR1   = 177574          1939#
SR2   = 177576          1940#
SR3   = 172516          1941#  9196*
SSINIT  037176         10209#  12081
STACK = 001100          1823#  2757   2770   2940   9594   9626   11398
START   001772          2174   2180   2755#
START1  001776          2747   2750   2753   2756#  11407
STKLMT= 177774          1834#
SVAL  = 100000          2071#  3213   3227   3302   3333   3349   3384   3395   3431   3500   3557   3747   3762
                        3786   8807   8823
SWR     001140          2235#  2768   2790*  2792   2798*  2805*  2819   9253   9267   9270   9285   9292   9463
                        9470   9505   9509   9586   9621   9749   10747  11683  11722  11777* 12004  12017*
SWREG   000176          2178#  2798   2819   11683  11722  11745
SW0   = 000001          1887#
SW00  = 000001          1877#  1887
SW01  = 000002          1876#  1886
SW02  = 000004          1875#  1885
SW03  = 000010          1874#  1884
SW04  = 000020          1873#  1883
SW05  = 000040          1872#  1882
SW06  = 000100          1871#  1881
SW07  = 000200          1870#  1880
SW08  = 000400          1869#  1879
SW09  = 001000          1868#  1878
SW1   = 000002          1886#
SW10  = 002000          1867#
SW11  = 004000          1866#
SW12  = 010000          1865#  9586
SW13  = 020000          1864#  10747
SW14  = 040000          1863#
SW15  = 100000          1862#
SW2   = 000004          1885#
SW3   = 000010          1884#
SW4   = 000020          1883#
SW5   = 000040          1882#
SW6   = 000100          1881#
SW7   = 000200          1880#
SW8   = 000400          1879#
SW9   = 001000          1878#  9749
S.ACLO= 000100          2106#
S.BRHM= 000100          2120#
S.BRKE= 040000          2142#
S.CART= 000400          2122#
S.DIB = 002000          2138#
S.DOOR= 000200          2121#
S.DRA = 000040          2092#
S.DROT= 020000          2113#
S.DRY = 000200          2094#
S.DSC = 040000          2101#
S.FLT = 000200          2107#
S.FORM= 001000          2096#
S.FWD = 002000          2124#
```

```
S.HDFL= 000200           2135#
S.HDHM= 000040           2119#   7918    8481
S.ICYL= 000040           2105#
S.ILF = 000400           2108#
S.LIMD= 020000           2141#
S.LOAD= 010000           2126#
S.MHD = 000400           2136#
S.NMOV= 010000           2140#
S.OFF = 002000           2097#
S.PAR = 001000           2109#
S.PIP = 020000           2100#
S.PLO = 004000           2139#
S.REV = 004000           2125#
S.RTZ = 020000           2127#
S.SECT= 000020           2132#
S.SKI = 002000           2110#
S.SPIN= 010000           2099#
S.SPLS= 010000           2112#
S.SPOK= 001000           2123#
S.TYPE= 000400           2095#
S.UNLD= 040000           2128#
S.UNS = 040000           2114#
S.VV  = 000100           2093#
S.WCLK= 000040           2133#
S.WGAT= 000100           2134#
S.WLE = 004000           2111#
S.WRL = 004000           2098#
S.XDOK= 000020           2118#
S.XERR= 001000           2137#
TBITVE= 000014           1920#
TCHKOP= 104421           3088    3154    3224    3238    3291    3300    3330    3346    3381    3513    3576    3603    3646
                         3694    3782    3833    4082    4248    4273    4299    4377    4410    4456    4490    4526    4579
                         4640    4699    4713    4752    4782    4797    4848    4871    4934    4952    5008    5021    5084
                         5101    5156    5170    5234    5249    5307    5321    5370    5390    5444    5461    5773    5792
                         5807    5857    5880    5895    5943    5967    6028    6048    6063    6114    6133    6148    6207
                         6264    6322    6378    6434    6480    6549    6569    6632    6647    6705    6720    6781    6845
                         6935    6965    7008    7023    7065    7080    7201    7216    7330    7347    7394    7425    7451
                         7470    7496    7521    7611    7629    7635    7650    7675    7690    7695    7710    7733    7765
                         7771    7786    7861    7915    8004    8048    8139    8172    8198    8244    8260    8268    8283
                         8357    8364    8379    8394    8437    8478    8513    8531    8582    8653    8722    8790    8804
                         8956    8976    9004    11354   11361   11389   12084#
TCHKWE= 104422           3207    3425    3469    3534    3663    3713    3941    3949    4010    4397    4478    4541    4594
                         4655    5509    5562    5698    6508    6804    6880    7100    7238    7813    7891    7965    8040
                         8113    8320    8454    12085#
TGETRK= 104420           3074    3124    3393    3610    3746    3761    3813    3854    3882    4277    8665    8732    8822
                         9864    10084   10130   10230   10429   12083#
TIMCNT  001654           2694#   9764*   9793*   9796*   9855*   10071*
TKVEC = 000060           1927#   11660*  11661*
TLOADR= 104417           3069    3119    3201    3233    3287    3296    3327    3342    3377    3421    3465    3510    3530
                         3565    3572    3599    3642    3652    3690    3700    3742    3778    3808    3850    3913    3932
                         3976    4006    4079    4244    4269    4373    4390    4453    4471    4522    4537    4575    4590
                         4636    4651    4695    4709    4748    4778    4793    4844    4867    4930    4940    5004    5018
                         5080    5097    5152    5166    5230    5245    5303    5317    5366    5386    5440    5458    5506
                         5550    5660    5769    5788    5803    5853    5876    5891    5939    5963    5977    6024    6044
                         6059    6109    6129    6144    6203    6260    6318    6374    6430    6476    6504    6545    6628
                         6643    6701    6716    6777    6800    6841    6876    6931    6961    7004    7019    7061    7076
```

		7096	7197	7212	7234	7326	7343	7390	7421	7466	7491	7517	7591	7624
		7646	7665	7685	7706	7723	7761	7782	7797	7809	7853	7872	7887	7911
		7958	8000	8019	8036	8077	8092	8104	8136	8168	8188	8224	8240	8256
		8279	8295	8316	8353	8375	8390	8405	8418	8450	8474	8509	8527	8578
		8632	8701	8787	8800	8943	8952	8965	8972	8987	8999	11350	11385	12082#
TPVEC = 000064		1928#												
TRAPVE= 000034		1926#	2776*	2777*										
TRKERR= 000020		6811	10389	12412#										
TRTVEC= 000014		1921#												
TSSINI= 104416		3037	3084	3114	3193	3281	3321	3371	3414	3458	3636	3669	3684	3736
		3802	3844	3907	3925	3970	3988	4066	4226	4242	4267	4296	4351	4380
		4427	4460	4506	4558	4612	4673	4731	4828	4915	4991	5065	5139	5216
		5289	5351	5426	5495	5548	5611	5754	5838	5924	6009	6094	6181	6240
		6297	6354	6410	6464	6493	6532	6561	6609	6682	6760	6831	6917	6948
		6989	7047	7086	7183	7223	7311	7411	7479	7507	7578	7749	7818	7842
		7876	7897	7921	7947	7985	8066	8120	8157	8175	8213	8342	8460	8484
		8491	8549	8564	8618	8688	8779	8935	11335	11364	12081#			
TSTAPT 032202		9139	9142#											
TSTCYL 036122		9941*	9947*	9953#										
TSTLUP 004720		3308#	8602											
TST1 003162		2938#	9312											
TST10 005334		3456#	9319											
TST100 026132		8063#	9375											
TST101 026356		8155#	9376											
TST102 026504		8211#	9377											
TST103 027026		8339#	9378											
TST104 027560		8546#	9379											
TST105 030022		8613#	9380											
TST106 030246		8683#	9381											
TST107 030466		8757#	9382											
TST11 006260		3634#	9320											
TST110 031174		8923#	9383											
TST12 006376		3682#	9321											
TST13 006514		3734#	9322											
TST14 006756		3800#	9323											
TST15 007110		3842#	9324											
TST16 007320		3903#	9325											
TST17 007510		3968#	9326											
TST2 003300		2969#	9313											
TST20 007644		3998	4042#	9327										
TST21 010512		4221#	9328											
TST22 011110		4348#	9329											
TST23 011322		4424#	9330											
TST24 011540		4503#	9331											
TST25 011664		4555#	9332											
TST26 012016		4609#	9333											
TST27 012174		4669#	9334											
TST3 003366		3034#	9314											
TST30 012326		4728#	9335											
TST31 012622		4825#	9336											
TST32 013016		4900#	9337											
TST33 013252		4981#	9338											
TST34 013440		5049#	9339											
TST35 013672		5130#	9340											
TST36 014060		5200#	9341											
TST37 014302		5280#	9342											

```
TST4    004616          3279#   9315
TST40   014470          5349#   9343
TST41   014662          5424#   9344
TST42   015056          5493#   9345
TST43   015262          5546#   9346
TST44   015466          5609#   9347
TST45   016326          5752#   9348
TST46   016532          5836#   9349
TST47   016736          5922#   9350
TST5    004724          3319#   9316
TST50   017154          6007#   9351
TST51   017360          6092#   9352
TST52   017564          6179#   9353
TST53   017710          6238#   9354
TST54   020034          6295#   9355
TST55   020160          6352#   9356
TST56   020304          6408#   9357
TST57   020430          6462#   9358
TST6    005102          3369#   9317
TST60   020502          6491#   9359
TST61   020562          6526#   9360
TST62   021046          6606#   9361
TST63   021230          6678#   9362
TST64   021416          6757#   9363
TST65   022122          6911#   9364
TST66   022340          6987#   9365
TST67   022444          7043#   9366
TST7    005230          3412#   9318
TST70   023066          7179#   9367
TST71   023510          7309#   9368
TST72   023620          7371#   9369
TST73   024466          7576#   9370
TST74   025160          7747#   9371
TST75   025402          7839#   9372
TST76   025660          7945#   9373
TST77   025740          7983#   9374
TWAT1M= 104440          12099#
TWAT1S= 104435          12096#
TWAT11= 104431          4080    4245    4374    4454    4523    4576    4637    4696    4749    4779    6778    6932    7391
                        12092#
TWAT12= 104432          12093#
TWAT14= 104433          12094#
TWAT15= 104434          6477    6505    6546    7422    7592    7666    7724    8488    11386   12095#
TWAT16= 104423          3071    3121    3202    3235    3288    3297    3328    3343    3378    3422    3466    3531    3573
                        3600    3643    3660    3691    3708    3743    3779    3809    3851    3875    3915    3933    3978
                        4007    4270    7467    7625    7647    7686    7707    7762    7783    7798    7854    7873    7888
                        7912    7962    8078    8093    8137    8169    8225    8241    8257    8280    8303    8317    8354
                        8376    8406    8451    8475    8510    8633    8645    8702    8714    8788    8301    8944    8953
                        8966    8973    8988    11351   12086#
TWAT2S= 104436          7810    8814    12097#
TWAT32= 104424          4391    4472    4538    4710    4794    4868    4950    5019    5098    5167    5246    5318    5387
                        5459    5789    5804    5877    5892    5964    5978    6644    6801    6842    6877    6962    7020
                        7077    7097    7213    7235    7344    8111    8195    8265    8391    12087#
TWAT48= 104425          4591    4652    6045    6060    6130    6145    7438    7518    8037    12088#
TWAT64= 104426          6717    8301    8434    8528    8579    12089#
TWAT8S= 104437          3754    3920    3983    7492    7822    7858    7927    8125    8229    8361    8496    8948    8969
```

```
                    11358  12098#
TWAT80= 104427       7472   7632   7693   7769   9001  12090#
TWAT96= 104430       4845   4931   5005   5081   5153   5231   5304   5367   5441   5507   5560   5682   5770
                     5854   5940   6025   6110   6204   6261   6319   6375   6431   6629   6702   7005   7062
                     7198   7327   8001   8081  12091#
TYPDS = 104405       9120   9127  12070#
TYPE  = 104401       2813   2833   2844   2863   2868   2871   2878   2882   2890   2899   3060   3064   3091
                     3162   3167   3180   3182   4908   4909   4911   5056   5057   5058   5207   5208   5209
                     5618   8766   9114   9121   9128   9465   9477   9480   9485   9496   9543   9544   9548
                     9549   9556   9567   9569   9579   9580   9582   9590   9623  10783  10784  10816  10817
                    11305  11369  11397  11413  11458  11552  11621  11677  11689  11743  11744  11747  11758
                    11768  11779  11798  11846  11852  11857  11861  11866  11867  11869  11872  11876  11942
                    11944  12026  12066#
TYPERR  034136       9495   9531#
TYPOC = 104402       2870   2881   2898   9479   9484   9564   9624  11746  12067#
TYPON = 104404      12069#
TYPOS = 104403      12068#
TYPTBL  001722       2731#  3136*  9963   9968
T.ASOF  001556       2649#  3483   3516   3539   3579   3611   8639*  8640   8667   8708*  8709   8734   9008
                    13314  13324
T.BA    001544       2644#  5523   5527   5577   5581   5707   5710   5714   5717   7118   7123   7137   7141
                     7256   7261   7275   7279  10307  10310  10395  13327
T.CS1   001540       2642#  3075   3125   3488   3522   3545   3585   3617   3815   3856   3883   4285   5516
                     5569   8958   8978   9865   998'  10220*  10221  10225  10238  10311  10435  10443  10460
                    10493  10761  13314  13324
T.CS2   001550       2646#  3077   3146   4278  10437  10494  13314  13324
T.DA    001546       2645#  10387  10390  10397  1039'  13326
T.DB    001564       2652#
T.DCYL  001560       2650#  10384  10396  13326
T.DS    001552       2647#  3097   3102   3127   3129   3213   3227   3241   3247   3302   3333   3349   3384
                     3395   3431   3494   3500   3551   3557   3747   3762   3769   3786   8807   8823  10439
                    10495  13314  13324
T.ECPS  001570       2654#
T.ECPT  001572       2655#
T.ER    001554       2648#  3079   3148  10441  10496  13314  13324
T.MR1   001566       2653#  13314
T.MR2   001574       2656#  7918   8481
T.MR3   001576       2657#
T.SPAR  001562       2651#
T.WC    001542       2643#  5530   5534   5584   5588   5721   5724   5727   5730   7127   7133   7145   7150
                     7265   7271   7283   7288  10257  10260  10296  10339  10394  13327
UBAERR= 000002      10315  12409#
UFE   = 000400       2029#  10561
UFERR = 000004       3666   3716  10563  12400#
UNLOA   000107       1993#
UNS   = 040000       2054#
UNSERR= 040000       8455  12385#
UPE   = 020000       2034#  10577
UPERR = 000100       5701  10579  124(  .#
VV    = 000100       2065#  3241
WATSRT  035536       9812   9815   9818   9821   9824   9827   9830   9833   9836   9839   9842   9845   9848
                     9851#
WCE   = 040000       2035#  10532
WCERR = 000001      10259  12408#
WCKERR= 000004       7102   7240  10534  12391#
WLF   = 004000       2051#
```

```
WLERR = 004000          12382#
WRDATA= 000123           1999#  4383   4464   4530   4583   4644   4703   4767   4832   4919   4994   5073   5142
                         5219   5293   5355   5430   5499   5624   5758   5842   5928   6013   6098   6621   6694
                         6793   6993   7054   7190   7319   7414   7434   7482   7584   7654   7880   7993   8012
                         8997
WRDCNT  001706           2724#  11088  11116*  11164*  11288
WRDNUM  001704           2723#  11087  11117*  11163*  11287
WRHEAD= 000127           2001#  4230   4355   4431   4510   4562   4616   4677   4738   6534   6764   6921   7380
                         8410   8520   8571   11372
WRL   = 004000           2068#  3129
WRTCHK= 000131           2002#  7012   7069   7089   7205   7227   7336   7714
WRTGAT= 040000           2087#
$APTHD  000224           2195   2201#
$ASTAT= ****** U         9421   9436
$ATYC   033372           9392   9394#
$ATY1   033346           9390#
$ATY3   033354           9391#  11443
$ATY4   033364           9393#  9501
$AUTOB  001134           2232#  2823*  11736  11893
$BASE   001352           2330#  2869   2877*  2923   2947
$BDADR  001122           2227#
$BDDAT  001126           2229#
$BELL   001266           2281#  9465   9521   11689  11886
$CDW1   001356           2332#
$CHARC  044460           11460*  11470*  11477  11486*  11491#
$CKSWR  045434           11722#  12074
$CMTAG  001100           2215#  2765   2766   2774   2780   2781   2782
$CM1  = 000020           2247#  2248#  2249#  2250#  2251#  2252#  2253#  2254#  2255#  2256#  2257#  2258#  2259#
                         2260#  2261#  2262#  2263#
$CM2  = 000040           2247#  2248#  2249#  2250#  2251#  2252#  2253#  2254#  2255#  2256#  2257#  2258#  2259#
                         2260#  2261#  2262#  2263#
$CM3  = 000020           2245#  2247
$CM4  = 000020           2263#  2264#  2265#  2266#  2267#  2268#  2269#  2270#  2271#  2272#  2273#  2274#  2275#
                         2276#  2277#  2278#  2279#
$CNTLC  046332          11677   11758  11886#
$CNTLG  046344          11743   11888#
$CNTLU  046337          11768   11861  11887#
$CORE   032466           9182   9209#
$CPUOP  001324           2304#
$CRLF   001273           2283*  9128   9485   9496   9521   9543   9548   9556   9569   9582   10783  10816  11459
                        11494   11779  11866  11886  11947
$CROUT  032516           9209   9216#
$DBLK   044700          11518   11552  11560#
$DEVCT  001306           2295#
$DEVM   001354           2331#  3047   4095*  4145*  8593   8617   8687   8759   8770   8927
$DOAGN  032172           9110   9131   9137#
$DTBL   044670          11521   11556#
$ENDAD  032162           2811   9133#  9516
$ENDCT  032026           2780   9112#
$ENULL  032176           9140#
$ENV    001316           2300#  2817   9142   9399   9423   9498   11438
$ENVM   001317           2301#  2803   2828   3048   9401   11440  11445
$EOP    031772           2835   8767   9102#  9384   9596   9627   11404
$EOPCT  032020           2780*  2834*  9109#  9113   9595*  9625*  11403*
$ERFLG  001103           2218#  2980   4086   4125*  9237   9281   9283   9289*  9311   9460*  9521   9751   11399*
$ERMAX  001115           2224#  2783*  9283   9306*  9311
```

```
$ERROR  033614          2774    9252    9458#
$ERRPC  001116          2225#   9467*   9468*   9469    9475*   9476*   947.    9482*   9483    9521    13314   13317   13323
                       13331
$ERRTB  001360          2349#   9540
$ERTTL  001112          2222#   9125    9129*   9466*   9521
$ESCAP  001264          2280#   2782*   2924*   3068*   3106*   3189*   3261*   5659*   5704*   5734*   9305*   9512    9514
                        9521    9871*   9872*   10133*  10134*  10233*  10234*  10856   10860*  10861*  11400*
$ETABL  001316          2299#
$ETEND  001360          2207    2333#
$FATAL  001300          2292#   9427*
$FFLG   033612          9390*   9393*   9421    9430*   9438#
$FILLC  001156          2243#   11463   11494
$FILLS  001155          2242#   1 494
$GDADR  001120          2226#
$GDDAT  001124          2228#
$GET42  032152          9130#
$GTSWR  045524          11744#  12072
$HD     =  000000       1808
$HIBTS  000224          2202#
$HIOCT  046532          11935*  11946#
$ICNT   001104          2219#   9296*   9297    9299*   9310
$ILLUP  046770          11996   12012   12029#
$INTAG  001135          2233#   11741*  11760   11780   11893
$ITEMB  001114          2223#   9469*   9500    9521    9534
$KTNEX  032460          9183    9208#
$KTOUT  032450          9200    9205#
$KT11   032316          2851    9181#   9185*   9208*
$LF     001274          2284#   9521    11494   11876   11886   11947
$LFLG   033611          9431*   9437#
$LPADR  001106          2220#   2784*   9278*   9287*   9303*   9308    9310
$LPERR  001110          2221#   2785*   3923*   3986*   4065*   4239*   4253*   4293*   4379*   4459*   4777*   4913*   4988*
                        5062*   5136*   5213*   5286*   6529*   6558*   6829*   6914*   6945*   7084*   7220*   7408*   7503*
                        8775*   8932*   9287    9304*   9310    9511    9753
$LSTAD  032572          2849    9223*   9230#
$LSTBK  032574          2831    2847    9224*   9231#
$MADR1  001330          2317#   2860*
$MADR2  001334          2321#
$MADR3  001340          2324#
$MADR4  001344          2327#
$MAIL   001276          2203    2207    2290#   2802    2817    9302    9498    11438
$MAMS1  001326          2311#   2859*   4902    4983    5051    5132    5202    5282
$MAMS2  001332          2319#
$MAMS3  001336          2322#
$MAMS4  001342          2325#
$MBADR  000226          2203#
$MFLG   033610          9391*   9397    9432*   9436#
$MNEW   046362          11747   11891#
$MSGAD  001312          2297#   9407*   9410
$MSGLG  001314          2298#   9412*
$MSGTY  001276          2291#   9405    9413*   9425    9429*
$MSWR   046351          11744   11889#
$MTYP1  001327          2312#   2858*
$MTYP2  001333          2320#
$MTYP3  001337          2323#
$MTYP4  001343          2326#
$MXCNT  033120          9300    9310#
```

```
$NULL   001154            2241#  11465  11494
$NWTST= 000001            2931#   2933   2962#  2964   2992#  2994   3272#  3274   3309#  3311   3361#  3363   3405#
                          3407   3438#  3440   3624#  3626   3671#  3673   3721#  3723   3792#  3794   3835#  3837
                          3893#  3895   3959#  3961   4018#  4020   4212#  4214   4332#  4334   4414#  4416   4494#
                          4496   4547#  4549   4599#  4601   4660#  4662   4719#  4721   4816#  4818   4890#  4892
                          4971#  4973   5039#  5041   5120#  5122   5190#  5192   5270#  5272   5341#  5343   5414#
                          5416   5486#  5488   5539#  5541   5593#  5595   5744#  5746   5827#  5829   5914#  5916
                          5998#  6000   6083#  6085   6168#  6170   6227#  6229   6284#  6286   6341#  6343   6397#
                          6399   6453#  6455   6482#  6484   6517#  6519   6598#  6600   6669#  6671   6743#  6745
                          6903#  6905   6979#  6981   7025#  7027   7161#  7163   7299#  7301   7352#  7354   7559#
                          7561   7737#  7739   7830#  7832   7937#  7939   7972#  7974   8053#  8055   8144#  8146
                          8200#  8202   9325#  8327   8534#  8536   8606#  8608   8675#  8677   8742#  8744   8908#
                          8910
$OCNT   045132            11593*  11622*  11635#
$OMODE  045134            11588*  11592*  11597  11600*  11611*  11637#
$OVER   033104             9254    9279    9288   9298    9307#
$PASS   001304             2294#   2802*   2914*  3050    8763    9106*  9107*  9118   9140   9144   9294   9311
$PASTM  000232             2205#
$POWER  046776            12027   12032#
$PWRDN  046630             2778   11996#  12024
$PWRMG  046764            12027#
$PWRUP  046702            12006   12012#
$QUES   001272             2282#   9521   11494  11798  11869  11886  11944  11947
$RDCHR  045776            11811#  12075
$RDDEC= ****** U          12078
$RDLIN  046066            11834#  12076
$RDOCT  046374            11908#  12077
$RDSZ = 000010            11827#
$REGAD  001160             2245#
$REG0   001162             2247#
$REG1   001164             2248#
$REG10  001202             2255#   5521*  5526*  5533*  5573*  5580*  5587*  5709*  5716*  5723*  5729*  6814*  6824*
                           7122*   7132*  7140*  7149*  7260*  7270*  7278*  7287*  10151  10184  10480*  11111*  13328
                          13332
$REG11  001204             2256#   5520*  5527*  5534*  5574*  5581*  5588*  5710*  5717*  5724*  5730*  6815*  6825*
                           7123*   7133*  7141*  7150*  7261*  7271*  7279*  7288*  10481*  11112*  13328  13332
$REG12  001206             2257#  10156  11113*  13328  13332
$REG13  001210             2258#  13328
$REG14  001212             2259#  13328
$REG15  001214             2260#  13328
$REG16  001216             2261#  13328
$REG17  001220             2262#  13328
$REG2   001166             2249#
$REG3   001170             2250#
$REG4   001172             2251#
$REG5   001174             2252#
$REG6   001176             2253#
$REG7   001200             2254#
$RESRE  046572            11980#  12079
$RTNAD  032174             9139#
$R2A  = ****** U          12080
$SAVRE  046534            11964#  12078
$SAVR6  046774            12005*  12013  12014*  12015*  12031#
$SCOPE  032576             2772    9246#
$SETUP= 000137             2744#   2771   2772   2774   2776   2778   2780   2781   2782   2784   2811   2814   9104
                           9247    9459   9508   9516   11681  11686  11687  11717  11893
```

```
$SIZE   032234          2830    9165#
$SIZEX  032522          9207    9217#
$STUP = 177777          2744#
$SVLAD  033050          9262    9301#
$SWR  = 167400          1797#   1808    1812    1813    1814    1815    1816    1817    1818    1819    2279    2280    2281
                        2781    2782    2784    2785    2939    2970    3035    3280    3320    3370    3413    3457    3635
                        3683    3735    3801    3843    3904    3969    4043    4222    4349    4425    4504    4556    4610
                        4670    4729    4826    4901    4982    5050    5131    5201    5281    5350    5425    5494    5547
                        5610    5753    5837    5923    6008    6093    6180    6239    6296    6353    6409    6463    6492
                        6527    6607    6679    6758    6912    6988    7044    7180    7310    7372    7577    7748    7840
                        7946    7984    8064    8156    8212    8340    8547    8614    8684    8758    8924    9099    9105
                        9132    9138    9140    9238    9239    9240    9241    9242    9248    9265    9267    9268    9281
                        9282    9283    9290    9291    9292    9304    9307    9310    9450    9451    9452    9453    9454
                        9463    9470    9505    9509    9521    12028
$SWREG  001320          2302#   2805
$SWRMK= 000000          1819    1820    9242    9243    9271
$SWOBT  033122          9277    9311#   9871    10133   10233   10860
$TERM = 000102          12100#
$TESTN  001302          2293#   9302*   9532*   9533*   9869    10131   10231   10858   11333*  13314   13317   13323   13331
$TIMES  001262          2279#   2781*   2939*   2970*   3035*   3280*   3320*   3370*   3413*   3457*   3635*   3683*   3735*
                        3801*   3843*   3904*   3969*   4043*   4222*   4349*   4425*   4504*   4556*   4610*   4670*   4729*
                        4826*   4901*   4982*   5050*   5131*   5201*   5281*   5350*   5425*   5494*   5547*   5610*   5753*
                        5837*   5923*   6008*   6093*   6180*   6239*   6296*   6353*   6409*   6463*   6492*   6527*   6607*
                        6679*   6758*   6912*   6988*   7044*   7180*   7310*   7372*   7577*   7748*   7840*   7946*   7984*
                        8064*   8156*   8212*   8340*   8547*   8587*   8614*   8684*   8758*   8924*   9105*   9290*   9297
                        9300*   9310
$TKB    001146          2238#   11641   11662   11673   11698   11726   11753
$TKCNT  045136          11642#  11657*  11687   11704*  11818   11820*
$TKINT  045146          2762    3865    8658    11657#  11678   11739
$TKQEN= 045145          11646#  11712   11823
$TKQIN  045140          11643#  11658*  11659   11710*  11711*  11712   11714*
$TKQOU  045142          11644#  11659*  11821   11822*  11823   11825*
$TKQSR  045144          11645#  11658   11714   11825
$TKS    001144          2237#   5665*   5681*   5737*   11641   11663*  11694*  11696   11702*  11724   11740*  11750   11762*
                        11782*
$TKSRV  045216          11660   11673#
$TMP0   001222          2263#   2873*   2875    2877    2884*   2886    2889    2901*   2903    2905    2907*   2908*   2909*
                        2910*   2911*   2913    10219*  10223*
$TMP1   001224          2264#   8769*   8847*   8872    8883    8905*   8926*   9031*   9056    9067    9089*   10645   11026*
                        11028*  11030   11044*  11060   11061   11065   11067*
$TMP10  001242          2271#   10419*  10599   10701   10831
$TMP11  001244          2272#   10420*  10680   10825
$TMP12  001246          2273#   10421*  10660   10820
$TMP13  001250          2274#   10423*  10427*  10457   10596   10614   10658   10678   10699   10750   10805
$TMP14  001252          2275#   10773*  10792   10796*  10798   10802*  10823   10827*  10829   10833*
$TMP15  001254          2276#   10774*  10779*
$TMP16  001256          2277#   10002*  10007*
$TMP17  001260          2278#   10004*  10012   10014*
$TMP2   001226          2265#   10646
$TMP3   001230          2266#   10158*  10172*  10187
$TMP4   001232          2267#   3066*   3095    3099*
$TMP5   001234          2268#   10883*  10893*  10900
$TMP6   001236          2269#   10880*  10901
$TMP7   001240          2270#   3187*   3257*   3262    10881*  10902
$TN   = 000111          1798#   1808    2931    2939#   2962    2970#   2992    3035#   3272    3280#   3309    3320#   3361
                        3370#   3405    3413#   3438    3457#   3624    3635#   3671    3683#   3721    3735#   3792    3801#
```

```
                    3835    3843#   3893    3904#   3959    3969#   4018    4043#   4212    4222#   4332    4349#   4414
                    4425#   4494    4504#   4547    4556#   4599    4610#   4660    4670#   4719    4729#   4816    4826#
                    4890    4901#   4971    4982#   5039    5050#   5120    5131#   5190    5201#   5270    5281#   5341
                    5350#   5414    5425#   5486    5494#   5539    5547#   5593    5610#   5744    5753#   5827    5837#
                    5914    5923#   5998    6008#   6083    6093#   6168    6180#   6227    6239#   6284    6296#   6341
                    6353#   6397    6409#   6453    6463#   6482    6492#   6517    6527#   6598    6607#   6669    6679#
                    6743    6758#   6903    6912#   6979    6988#   7025    7044#   7161    7180#   7299    7310#   7352
                    7372#   7559    7577#   7737    7748#   7830    7840#   7937    7946#   7972    7984#   8053    8064#
                    8144    8156#   8200    8212#   8325    8340#   8534    8547#   8606    8614#   8675    8684#   8742
                    8758#   8908    8924#   9272    9312    11333
$TPB      001152    2240#   11483*  11494
$TPFLG    001157    2244#   11432   11494
$TPS      001150    2239#   11481   11494
$TRAP     047006    2776    12043#
$TRAP2    047030   12054#   12065
$TRP    = 000041   12058#   12067#  12068#  12069#  12070#  12071#  12072   12073#  12074   12075#  12076#  12077#  12078#
                   12079#   12080#  12081#  12082#  12083#  12084#  12085#  12086#  12087#  12088#  12089#  12090#  12091#
                   12092#   12093#  12094#  12095#  12096#  12097#  12098#  12099#  12100#
$TRPAD    047042   12048    12065#  12100
$TSTM     000230    2204#
$TSTNM    001102    2217#    8599*   9104*   9146*   9237    9274*   9301*   9302    9307    9311    9462    9521    9532
                    9927    11338
$TTYIN    046322   11836    11837   11849   11867   11881   11885#
$TYPBN=   ****** U 12071
$TYPDS    044464   11506#   12070
$TYPE     044202    9418    11432#  12058   12066
$TYPEC    044414   11462    11469   11476   11481#  11482   11784
$TYPEX    044462   11487    11489   11492#
$TYPOC    044734   11591#   12067
$TYPON    044750   11590    11593#  12069
$TYPOS    044710   11586#   12068
$UNIT     001310    2296#
$UNITM    000234    2206#
$USWR     001322    2303#
$VECT1    001346    2328#    2879    2888*   2889*   2892    2912*   2913*   2918    2922
$VECT2    001350    2329#
$XTSTR    032622    9256#
$$GET4=   000000    9132#
$$SWOB=   000111    9311#    9312    9313#   9314#   9315#   9316#   9317#   9318#   9319#   9320#   9321#   9322#   9323#
                    9324#    9325#   9326#   9327#   9328#   9329#   9330#   9331#   9332#   9333#   9334#   9335#   9336#
                    9337#    9338#   9339#   9340#   9341#   9342#   9343#   9344#   9345#   9346#   9347#   9348#   9349#
                    9350#    9351#   9352#   9353#   9354#   9355#   9356#   9357#   9358#   9359#   9360#   9361#   9362#
                    9363#    9364#   9365#   9366#   9367#   9368#   9369#   9370#   9371#   9372#   9373#   9374#   9375#
                    9376#    9377#   9378#   9379#   9380#   9381#   9382#   9383#   9384#
$OFILL    045133   11587*   11591*  11601   11636#
$40CAT    000000    2171#    9248    9472
.       = 074414    2156#    2172    2176#   2182#   2184#   2191    2192#   2194#   2196#   2214#   2285    2695#   2769
                    2784     2785    9117#   9140    9141#   9310    9311    9439#   9521    11494   11560#  11641   11645#
                   11646    11647#  11885#  11886   11893#  11947   12008   12030   12100   13313#  13426#  13427#  13428#
                   13435#   13436#
.$ASTA=   ****** U  9391    9394
.$X     = 000224    2191#    2196
```

COMMEN 1# 1930#

ENDCOM 1# 1930#

ERROR 1824# 2956 2985 3038 3072 3085 3089 3115 3122 3155 3194 3203 3211 3216 3225
3230 3236 3239 3244 3251 3282 3289 3292 3298 3301 3305 3322 3329 3331 3338
3344 3347 3354 3372 3379 3382 3389 3400 3415 3423 3429 3436 3459 3467 3473
3487 3492 3498 3505 3514 3520 3526 3532 3538 3543 3549 3555 3561 3574 3577
3583 3589 3601 3604 3615 3621 3637 3644 3647 3661 3667 3670 3685 3692 3695
3709 3717 3737 3744 3751 3759 3766 3773 3780 3784 3790 3803 3810 3819 3831
3834 3845 3852 3860 3880 3887 3908 3916 3921 3926 3934 3945 3953 3971 3979
3984 3989 4008 4014 4067 4081 4083 4094 4105 4115 4122 4131 4144 4207 4227
4243 4246 4249 4268 4271 4274 4282 4289 4297 4300 4301 4302 4308 4317 4352
4375 4378 4381 4392 4401 4411 4428 4455 4457 4461 4473 4482 4491 4507 4524
4527 4539 4545 4559 4577 4580 4592 4598 4613 4638 4641 4653 4659 4674 4697
4700 4711 4714 4732 4750 4753 4780 4783 4795 4798 4804 4812 4829 4846 4849
4869 4872 4878 4885 4916 4932 49[illegible] 4951 4953 4959 4966 4992 5006 5009 5020
5022 5028 5035 5066 5082 5085 5099 5102 5108 5116 5140 5154 5157 5168 5171
5177 5185 5217 5232 5235 5247 5250 5257 5265 5290 5305 5308 5319 5322 5328
5336 5352 5368 5371 5388 5391 5402 5410 5427 5442 5445 5460 5462 5473 5481
5496 5508 5513 5522 5528 5535 5549 5561 5566 5575 5582 5589 5612 5696 5702
5711 5718 5725 5731 5755 5771 5774 5790 5793 5805 5808 5814 5822 5839 5855
5858 5878 5881 5893 5896 5902 5910 5925 5941 5944 5965 5968 5979 5985 5993
6010 6026 6029 6046 6049 6061 6064 6070 6078 6095 6111 6115 6131 6134 6146
6149 6155 6163 6182 6205 6208 6214 6222 6241 6262 6265 6271 6279 6298 6320
6323 6329 6337 6355 6376 6379 6385 6393 6411 6432 6435 6441 6449 6465 6478
6481 6494 6506 6512 6533 6547 6550 6562 6570 6571 6572 6579 6587 6610 6630
6633 6645 6648 6654 6662 6683 6703 6706 6718 6721 6727 6735 6761 6779 6782
6802 6808 6817 6826 6832 6843 6846 6852 6860 6878 6884 6890 6898 6918 6933
6936 6949 6963 6966 6990 7006 7009 7021 7024 7048 7063 7066 7078 7081 7087
7098 7104 7125 7134 7143 7151 7184 7199 7202 7214 7217 7224 7236 7242 7263
7272 7281 7289 7312 7328 7331 7345 7348 7392 7395 7412 7423 7426 7439 7447
7452 7468 7471 7473 7480 7493 7497 7508 7519 7522 7530 7538 7579 7598 7607
7612 7626 7630 7633 7636 7648 7651 7672 7676 7687 7691 7694 7696 7708 7711
7730 7734 7750 7763 7766 7770 7772 7784 7787 7799 7811 7817 7819 7823 7843
7855 7859 7862 7874 7877 7889 7895 7898 7913 7916 7922 7933 7948 7963 7969
7986 8002 8005 8038 8044 8049 8067 8079 8082 8094 8112 8117 8121 8126 8138
8140 8158 8170 8173 8176 8196 8199 8214 8226 8230 8242 8245 8258 8261 8266
8269 8281 8284 8318 8324 8343 8355 8358 8362 8365 8377 8380 8392 8395 8407
8435 8438 8452 8458 8461 8476 8479 8485 8492 8497 8511 8514 8529 8532 8550
8561 8565 8580 8583 8619 8634 8650 8654 8672 8689 8703 8719 8723 8739 8780
8789 8791 8802 8805 8811 8820 8828 8936 8945 8949 8954 8957 8967 8970 8974
8977 8989 9002 9005 9012 11336 11352 11355 11359 11362 11365 11387 11390

ESCAPE 1# 1930#

GETPRI 1# 1930# 9177

GETSWR 1# 1930# 2814#

INITSS 2740# 4241 4266 4295 4828 4914 4990 5064 5138 5215 5288 5426 5495 5548 5611
5754 5838 5924 6009 6094 6181 6240 6297 6354 6410 6464 6493 6531 6560 6609
6682 6760 6830 6916 6947 6989 7047 7086 7183 7223 7311 7410 7478 7506 7578
7749 7818 7842 7876 7897 7921 7947 7985 8066 8119 8157 8175 8213 8342 8460
8484 8491 8549 8563 8618 8688 8778 8934 11335 11364

LOADLS 2742# 4831 4851 4942 4993 5090 5141 5237 5292 5429 5447 5622 5757 5780 5795
5841 5864 5883 5927 5955 5969 6012 6036 6051 6097 6121 6136 6195 6252 6310
6366 6422 6620 6635 6693 6708 6763 6792 6833 6868 6950 6992 7011 7053 7068
7088 7189 7204 7226 7318 7335 7413 7458 7481 7583 7638 7653 7698 7713 7774
7789 7801 7845 7864 7879 7901 7950 7992 8011 8028 8069 8084 8096 8128 8160
8178 8216 8232 8247 8271 8307 8345 8367 8382 8397 8409 8440 8463 8501 8519
8570 11342 11371

MSG	2931#	2933	2962#	2964	2991#	2994	3272#	3274	3309#	3311	3361#	3363	3405#	3407	3438#
	3440	3624#	3626	3671#	3673	3721#	3723	3792#	3794	3835#	3837	3893#	3895	3959#	3961
	4018#	4020	4212#	4214	4332#	4334	4414#	4416	4494#	4496	4547#	4549	4599#	4601	4660#
	4662	4719#	4721	4816#	4818	4890#	4892	4971#	4973	5039#	5041	5120#	5122	5190#	5192
	5270#	5272	5341#	5343	5414#	5416	5486#	5488	5539#	5541	5593#	5595	5744#	5746	5827#
	5829	5914#	5916	5998#	6000	6083#	6085	6168#	6170	6227#	6229	6284#	6286	6341#	6343
	6397#	6399	6453#	6455	6482#	6484	6517#	6519	6598#	6600	6669#	6671	6743#	6745	6903#
	6905	6979#	6981	7025#	7027	7161#	7163	7299#	7301	7352#	7354	7559#	7561	7737#	7739
	7829#	7832	7937#	7939	7971#	7974	8052#	8055	8144#	8146	8200#	8202	8325#	8327	8534#
	8536	8606#	8608	8675#	8677	8742#	8744	8908#	8910						
MULT	1#	1930#													
NEWTST	1#	1930#	2931	2962	2992	3272	3309	3361	3405	3438	3624	3671	3721	3792	3835
	3893	3959	4018	4212	4332	4414	4494	4547	4599	4660	4719	4816	4890	4971	5039
	5120	5190	5270	5341	5414	5486	5539	5593	5744	5827	5914	5998	6083	6168	6227
	6284	6341	6397	6453	6482	6517	6598	6669	6743	6903	6979	7025	7161	7299	7352
	7559	7737	7830	7937	7972	8053	8144	8200	8325	8534	8606	8675	8742	8908	
OPCHK	2744#	4848	4871	4934	4952	5008	5021	5084	5101	5156	5170	5234	5249	5307	5321
	5444	5461	5773	5792	5807	5857	5880	5895	5943	5967	6028	6048	6063	6114	6133
	6148	6207	6264	6322	6378	6434	6480	6549	6632	6647	6705	6720	6781	6845	6935
	6965	7008	7023	7065	7080	7201	7216	7330	7347	7394	7425	7450	7470	7495	7521
	7610	7629	7635	7650	7674	7690	7695	7710	7732	7765	7771	7786	7861	7915	8004
	8047	8139	8172	8198	8244	8260	8268	8283	8357	8364	8379	8394	8437	8478	8513
	8531	8582	8652	8721	8790	8804	8955	8975	9004	11354	11361	11389			
POP	1#	1930#	9433	9434	9903	10037	10235	11103	11122	11126	11290	11547	11936	11985	12017
	12018														
PUSH	1#	1930#	9394	9396	9417	9892	10024	10209	10992	11003	11506	11910	11965	11998	12004
REPORT	1#	1930#													
RESDC	2741#	4879	4960	5029	5110	5179	5259	5330	5475	5816	5904	5987	6072	6157	6216
	6273	6331	6387	6443	6581	6656	6729	6854	6892	7532					
RKLOAD	2743#	4844	4867	4930	4949	5004	5018	5080	5097	5152	5166	5230	5245	5303	5317
	5440	5458	5506	5559	5769	5788	5803	5853	5876	5891	5939	5963	5977	6024	6044
	6059	6109	6129	6144	6203	6260	6318	6374	6430	6476	6504	6628	6643	6701	6716
	6777	6800	6841	6876	6931	6961	7004	7019	7061	7076	7096	7197	7212	7234	7326
	7343	7390	7421	7466	7517	7624	7646	7685	7706	7761	7782	7797	7809	7853	7872
	7887	7911	8000	8036	8077	8092	8136	8168	8224	8240	8256	8279	8316	8353	8375
	8390	8405	8450	8473	8509	8527	8578	8632	8701	8787	8800	8943	8952	8965	8972
	8987	11350	11385												
SCOPE	1825#	2938	2969	3034	3279	3319	3369	3412	3456	3634	3682	3734	3800	3842	3903
	3968	4042	4221	4348	4424	4503	4555	4609	4669	4728	4825	4900	4981	5049	5130
	5200	5280	5349	5424	5493	5546	5609	5752	5836	5922	6007	6092	6179	6238	6295
	6352	6408	6462	6491	6526	6606	6678	6757	6911	6987	7043	7179	7309	7371	7576
	7747	7839	7945	7983	8063	8155	8211	8339	8546	8586	8613	8683	8757	8923	9103
SETPRI	1#	1930#	11814												
SETTRA	12058#	12067	12068	12069	12070	12072	12074	12075	12076	12077	12078	12079	12080	12081	12082
	12083	12084	12085	12086	12087	12088	12089	12090	12091	12092	12093	12094	12095	12096	12097
	12098	12099													
SETUP	1#	1930#	2764												
SKIP	1#	1930#													
SLASH	1#	1930#													
SPACE	1930#														
STARS	1#	1930#	2188	2190	2197	2210	2285	2288	2931	2937	2962	2968	2992	3033	3272
	3278	3309	3318	3361	3368	3405	3411	3438	3455	3624	3633	3671	3681	3721	3733
	3792	3799	3835	3841	3893	3902	3959	3967	4018	4041	4212	4220	4332	4347	4414
	4423	4494	4502	4547	4554	4599	4608	4660	4668	4719	4727	4816	4824	4890	4899
	4971	4980	5039	5048	5120	5129	5190	5199	5270	5279	5341	5348	5414	5423	5486
	5492	5539	5545	5593	5608	5744	5751	5827	5835	5914	5921	5998	6006	6083	6091

```
        6168    6178    6227    6237    6284    6294    6341    6351    6397    6407    6453    6461    6482    6490    6517
        6525    6598    6605    6669    6677    6743    6756    6903    6910    6979    6986    7025    7042    7161    7178
        7299    7308    7352    7370    7559    7575    7737    7746    7830    7838    7937    7944    7972    7982    8053
        8062    8144    8154    8200    8210    8325    8338    8534    8545    8606    8612    8675    8682    8742    8756
        8908    8922    9095    9151    9234    9389    9446    9521    9530    9637    9743   10865   10878   10906   11294
       11417   11496   11563   11640   11717   11732   11803   11827   11896   11949   11994   12010   12037
SWRSU      1#   1930#   2786#
TRMTRP 12058#  12100
TYPBIN     1#   1930#
TYPDEC     1#   1930#   9118    9125
TYPNAM     1#   1797#   1930#   2807
TYPNUM     1#   1930#
TYPOCS     1#   1930#
TYPOCT     1#   1930#   2869    9478    9483   11745
TYPTXT     1#   1930#   9114    9121
$$CMRE  2208#   2247    2248    2249    2250    2251    2252    2253    2254    2255    2256    2257    2258    2259    2260
        2261    2262
$$CMTM  2208#   2263    2264    2265    2266    2267    2268    2269    2270    2271    2272    2273    2274    2275    2276
        2277    2278
$$ESCA     1#   1930#
$$NEWT     1#   1930#   2931    2962    2992    3272    3309    3361    3405    3438    3624    3671    3721    3792    3835
        3893    3959    4018    4212    4332    4414    4494    4547    4599    4660    4719    4816    4890    4971    5039
        5120    5190    5270    5341    5414    5486    5539    5593    5744    5827    5914    5998    6083    6168    6227
        6284    6341    6397    6453    6482    6517    6598    6669    6743    6903    6979    7025    7161    7299    7352
        7559    7737    7830    7937    7972    8053    8144    8200    8325    8534    8606    8675    8742    8908
$$SET  12058#  12067   12068   12069   12070   12072   12074   12075   12076   12077   12078   12079   12080   12081   12082
       12083   12084   12085   12086   12087   12088   12089   12090   12091   12092   12093   12094   12095   12096   12097
       12098   12099
$$SETM  2802#
$$SKIP     1#   1930#
.EQUAT     1#   1797#   1820
.HEADE     1#   1797#   1799
.KT11      1#   1797#   1930
.SETUP     1#   1797#   2744
.SWRHI     1#   1797#   1808
.SWRLO  1797#   1820#
.$ACT1     1#   1797#
.$APTB     1#   2286#
.$APTH     1#   1797#   2186
.$APTY     1#   1797#   9387
.$ASTA     1#
.$CATC     1#
.$CMTA     1#   1797#   2208
.$DB2D     1#
.$DB2O     1#
.$DIV      1#
.$EOP      1#   1797#   9093
.$ERRO     1#   1797#   9444
.$ERRT     1#   1797#
.$MULT     1#
.$POWE     1#   1797#  11992
.$RAND     1#
.$RDDE     1#
.$RDOC     1#   1797#  11894
.$READ     1#   1797#  11638
.$R2AZ     1#
```

```
.$SAVE      1#   1797#  11947
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#   1797#   9232
.$SIZE      1#   1797#   9149
.$SUPR      1#
.$TRAP      1#   1797#  12035
.$TYPB      1#
.$TYPD      1#   1797#  11494
.$TYPE      1#   1797#  11415
.$TYPO      1#   1797#  11561
.$40CA      1#   1797#   2154
.1170       1#


. ABS.  0744'4     000


 ERRORS DETECTED:  0

 DSKZ:CZR6KF.BIN,DSKZ:CZR6KF.LST/CRF/SOL/NL.TOC-DSKM:CZR6KF.SML,CZR6KF.P11
 RUN-TIME: 88 133 11 SECONDS
 RUN-TIME RATIO: 454/233=1.9
 CORE USED:  35K  (69 PAGES)
```