RK11/05F/J

RK11 BASIC LOGIC TEST 2

MD-11-DZRKK-E

EP-DZRKK-E-DL-A

FICHE 1 OF 1

AUG 1977

digital

MADE IN USA

B01

.REM %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZRKK-E-D
PRODUCT NAME:	RK11 BASIC LOGIC TEST 2
DATE CREATED:	APRIL, 1977
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	JIM KAPADIA
REVISED BY:	PERVEZ ZAKI TOM SAWYER CHUCK HESS

QUICK LOOK-UP OPERATING INSTRUCTIONS

FOR A QUICK REFERENCE, LOOK UP THE FOLLOWING SECTIONS:

1.0 ABSTRACT
2.0 REQUIREMENTS
4.1 LOADING AND OPERATOR ACTION
7.0 SWITCH OPTIONS

FOR A MORE COMPLETE EXPLANATION REFER TO THE TABLE OF CONTENTS BELOW AND THE FOLLOWING DOCUMENT.

TABLE OF CONTENTS

1.0 ABSTRACT

THE RK11 LOGIC TESTS CONSIST OF A SERIES OF TESTS AIMED AT CHECKING THE BASIC LOGIC OF THE RK11 CONTROLLER. THIS PROGRAM IS THE SECOND PART OF THE TWO-PART RK11 LOGIC TESTS. IT SHOULD BE NOTED THAT LOGIC TEST I AND LOGIC TEST II TOGETHER CONSTITUTE A COMPLETE PROGRAM AND BOTH OF THEM SHOULD BE RUN.

WHEN USED IN CONJUNCTION WITH A DRIVE IT IS CAPABLE OF DETECTING FAULTS IN THE DRIVE ALSO.

USED CORRECTLY THIS PROGRAM CAN BE AN EFFECTIVE ANALYTIC AND DIAGNOSTIC TOOL.

2.0 REQUIREMENTS

2.1 EQUIPMENT

A. PDP11 WITH CONSOLE TELETYPE.
B. 8K OF MEMORY
C. RK11 OR RKV11 CONTROLLER
D. 1-8 RK05 OR RK05F DRIVES OR THE RK05 SIMULATOR (DRIVE TYPES MAY BE MIXED)

2.2 PRELIMINARY PROGRAMS

RK11 BASIC LOGIC TEST I (MD-11-DZRKJ)

2.3 EXECUTION TIME

ERROR FREE FIRST PASS ON PDP11/20 WITH CORE MEMORY TAKES APPROXIMATELY TWO MINUTES. CONSIDERABLY LESS FOR FASTER MACHINES OR MEMORIES.

3.0 STARTING ADDRESS

200 FOR ANY MODE OF OPERATION. NORMAL START UP WITH ALL SWITCHES DOWN.

4.0 PROGRAM CONTROL MODES & OPERATOR ACTION

PAPER TAPE LOADING
RKDP DUMP MODE
RKDP CHAIN MODE
ACT11

4.1 PAPER TAPE LOADING

4.1.1 LOAD PROGRAM INTO MEMORY USING STANDARD PROCEDURE FOR .ABS TAPES.

4.1.2 MAKE SURE THAT THE DRIVES TO BE CHECKED ARE LOADED WITH DISKS AND ARE IN 'RUN'. 'WRT ENABLE' THEM. CHECK THAT 'WRT PROT' LIGHT ON THESE DRIVES IS OFF. PUT DRIVES THAT ARE NOT TO BE TESTED ON 'LOAD'.

4.1.3 LOAD ADDRESS 200

4.1.4 SET SWITCHES IF DESIRED (SEE SEC 7.0) IF TESTING ON SIMULATOR PUT SW<10> UP.

PRESS START.

4.1.5 THE PROGRAM IDENTIFIES ITSELF (NAME,MAINDEC NO), THEN THE FOLLOWING QUESTION IS ASKED:

DRIVES TO BE TESTED?

THE USER SHOULD TYPE IN THE DRIVE NUMBERS THAT ARE IN 'RUN' AND TO BE TESTED. CARRIAGE RETURN SHOULD TERMINATE THE STRING. IF AN RK-05F IS TO BE TESTED, TYPE THE SUFFIX 'F' WITH THE FIRST DRIVE OF THE PAIR. FOR EXAMPLE, IF DRIVES 2 AND 3 ARE ON AN RK-05F, TYPE ONLY 2F.

EXMP: DRIVES TO BE TESTED? 0,1,2<CR>

THE DRIVES DO NOT HAVE TO BE IN LOGICAL ORDER.

EXMP: DRIVES TO BE TESTED? 2,4<CR>

IF ANY ONE DRIVE IS TO BE TESTED, TYPE IN THAT NUMBER. IT DOES NOT HAVE TO BE DRIVE 0.

THUS A NORMAL SEQUENCE WITH DRIVES 0,1 WOULD BE:

RK11 BASIC LOGIC TEST 2
MAINDEC-11-DZRKK-E
DRIVES TO BE TESTED? 0,1<CR>

4.1.6 THERE IS A "RUBOUT" FEATURE WHICH ALLOWS RUBBING OUT ANY NUMBER OF CHARACTERS THAT WERE TYPED IN WRONG. THE RUBBED OUT CHARACTERS ARE ECHOED BACK WITHIN SLASHES.

"↑U" DELETES THE ENTIRE LINE

4.1.7 IF REPLY TO ANY OF THE ABOVE QUESTION IS IN A WRONG FORMAT (EX: 012<CR>;0,8<CR>; 0,A<CR>; M<CR> ETC), IT IS AUTOMATICALLY REJECTED, A "?" IS PRINTED OUT;

THE CORRECT ANSWER CAN NOW BE RETYPED AGAIN.

4.1.8 THE DRIVE NUMBER BEING TESTED OUT IS PRINTED:

```
DRIVE N ;N=0,1...7
IF THE DRIVE IS AN RK-05F, AN F IS APPENDED
```

AT THE END OF A PASS THE FOLLOWING TYPE-OUT OCCURS

```
END PASS # X
```

WHERE X= PASS NUMBER (1,2,3---), CONTROL IS PASSED TO THE BEGINNING OF THE PROGRAM AND RE-EXECUTION BEGINS. NO QUESTIONS ARE TO BE ANSWERED AGAIN.

4.1.9 ERROR FREE PASSES OF THE PROGRAM APPEAR AS SHOWN BELOW.

```
RK11 BASIC LOGIC TEST 2
MAINDEC-11-DZRKK-E
DRIVES TO BE TESTED?
0,1<CR>
DRIVE 0
DRIVE 1
END PASS # 1
 0
DRIVE 1
END PASS # 2
...
...
```

4.2 RKDP DUMP MODE

4.2.1 THE PROGRAM IS LOADED INTO THE MEMORY BY THE RKDP MONITOR

4.2.2 START AS NORMALLY USING SA 200

4.2.3 THE PROGRAM IDENTIFIES ITSELF (NAME,MAINDEC NO.). ON FINDING OUT THAT THE LOADING WAS BY RKDP (DUMP MODE), THE FOLLOWING MESSAGE APPEARS:

'TO TEST DRIVE 'N' HALT PROGRAM, REMOVE RKDP PACK AND REPLACE IT WITH A WORK PACK, CLEAR LOCATION 40, AND RESTART PROGRAM'

IF DRIVE 'N' IS TO BE TESTED, THE RKDP PACK ON THAT

DRIVE SHOULD BE REPLACED BY ANOTHER PACK, THE DRIVE SHOULD BE PUT ON 'WRT ENABL' (BECAUSE RKDP WRITE PROTECTS THE DRIVE).

IF DRIVE 'N' IS NOT TO BE CHECKED, THEN THE MESSAGE SHOULD BE IGNORED.

AFTER THIS, THE SEQUENCE OF QUESTIONING IS AS EXPLAINED IN SEC 4.1.5.

4.3 RKDP CHAIN MODE

THE PROGRAM IS CHAIN-LOADED FROM THE RKDP PACK ON DRIVE 'N'. AFTER THE PROGRAM IDENTIFIES ITSELF THE FOLLOWING PRINTOUT OCCURS.

'DRIVE 'N' NOT TESTED'

THERE IS NO OPERATOR INTERVENTION REQUIRED. THE PROGRAM FINDS OUT THE NUMBER OF DRIVES PRESENT.

4.4 ACT11 MODE

THE PROGRAM IS LOADED BY THE ACT11 MONITOR. ON STARTING, IDENTIFIES ITSELF, ASCERTAINS THE NUMBER OF DRIVES AND PROCEEDS WITH THE EXECUTION OF THE TESTS AS BEFORE.

5.0 DRIVE SELECTION

IF ANY PARTICULAR DRIVE IS TO BE SELECTED FOR TESTING, PUT THAT DRIVE ON 'RUN', 'WRITE ENABLE'; PUT REST OF THE DRIVES ON 'LOAD', 'WRITE LOCK' AND IN REPLY TO THE QUES- TIONYES TO BE TESTED?) TYPE IN THE DRIVE NUMBER FOLLOWED BY CR. SEE SEC 4.1.5.

6.0 DRIVE-LESS TEST

USE RK11 BASIC LOGIC TEST I, WHICH IS ACTUALLY THE FIRST PART OF THE TWO-PART RK11 BASIC LOGIC TESTS. SEE SEC 1.0, 2.2.

7.0 SWITCH OPTIONS

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

THE SCOPE ROUTINE OR BEGINS A NEW TEST. THE
'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE
TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

'SWR = NNNNNN NEW ='

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

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

SW<15>=1	HALT ON ERROR
SW<14>=1	LOOP ON TEST
SW<13>=1	INHIBIT ERROR PRINTOUTS
SW<12>=1	CYCLE ON ERROR TO THE PREVIOLS 'SCOPE' STATEMENT
SW<11>=1	INHIBIT ITERATIONS
SW<10>=1	TESTING ON SIMULATOR
SW<09>=1	LOOP ON SPECIFIC ERROR
SW<08>=1	LOOP ON TEST AS PER SW<07:00>
SW<06>=1	DROP THE DRIVE AFTER MAXIMUM ALLOWARLE NUMBER OF ERRORS OCCUR

7.1 SW<15>

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

7.2 SW<14>

THE PROGRAM LOOPS ON THE SUBTEST THAT IS BEING EXECUTED WHEN THE SWITCH IS PUT ON. THIS SWITCH IS USED NORMALLY ALONG SW 15. SEE SEC 8.0.

7.3 SW <13>

THIS SWITCH INHIBITS ALL ERROR MESSAGES. NORMALLY USED WHEN LOOPING ON TEST (SW 14) OR LOOPING ON ERROR (SW 9).

7.4 SW <12>

THIS SWITCH ALLOWS THE PORGRAM TO CYCLE FROM THE POINT OF ERROR TO THE PREVIOUS SCOPE STATEMENT. NOTE THAT IN DOING SO ANY INITIALIZATION BEING DONE AT THE BEGINING OF THE SUBTEST WILL BE DONE AGAIN AND AGAIN. SEE SEC 8.0 FOR DIFFERENT SCOPE LOOPS

AVAILABLE.

7.5 SW <11>

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

7.6 SW <10>

THIS SWITCH WHEN SET INDICATES THAT TESTING IS BEING DONE ON A SIMULATOR. THE SWITCH SHOULD BE PUT UP BEFORE START- ING THE PROGRAM. NOTE THAT RK11C IS NOT COMPATIBLE WITH THE SIMULATOR.

7.7 SW <09>

THIS SWITCH PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP. NOTE THATKE SW12 THE INITIALIZATION OF PARAMETERS AT THE BEGINNING OF THE SUBTEST MAY NOT BE DONE IN THIS CASE. THIS SWITCH IS HELPFUL WHEN A PARTICULAR PART OF A SUBTEST IS BEING REPEATED USING DIFFERENT PARAMETERS AND YOU WANT TO SCOPE ON THE PARAMETER IN ERROR. 'EXAMPLE: RKDA IS BEING WRITTEN AND READ BACK WITH COUNT PATTERNS FROM 1 TO 177777. PATTERN 561 IS GIVING ERROR, YOU MIGHT NOT WANT TO GO THROUGH THE 560 PATTERNS BEFORE HITTING ERROR ON THE 561TH PATTERN. IN THIS CASE SW 9 WILL GIVE YOU A SCOPE LOOP ON THE 561TH PATTERN ONLY

7.8 SW <08>

THIS SWITCH IS USED TO SELECT A PARTICULAR TEST (AS PER SW<00-07>) FOR EXECUTION AND SUBSEQUENT LOOPING. THUS IF TEST 15 IS TO BE SELECTED THE SWITCH SETTING WOULD BE 000415. IT SHOULD BE NOTED THAT BEFORE SELECTING TEST 15, ALL THE PREVIOUS TESTS (1-14) WILL BE EXECUTED.

7.9 SW<06>

THIS SWITCH ALLOWS THE PROGRAM TO DROP A DRIVE FROM THE SELECTION LIST AND TESTING AFTER MAXIMUM ALLOWABLE ERROR COUNT (TOTAL NUMBER OF ERRORS) ON THAT DRIVE IS EXCEEDED. THE MAXIMUM ALLOWABLE ERROR COUNT IS 5, AFTER 5 ERRRORS HAVE OCCURED DRIVE IS DROPPED AND A MESSAGE (DRIVE # XXX DROPPED) IS PRINTED.

8.0 SCOPE LOOPS

THERE ARE THREE KINDS OF SCOPE LOOPS AVAILABLE

1. SW14: LOOPING IS DONE FOR THE ENTIRE SUB-TEST

2. SW12: LOOPING IS DONE FROM THE POINT OF ERROR BACK TO THE PREVIOUS 'SCOPE' STATEMENT.

3. SW09: PROVIDE THE TIGHTEST POSSIBLE SCOPE LOOP SEE SEC. 7.7

EXAMPLE:

```
TST1:   SCOPE
        :


        INITIALIZATION
        :
  ERROR 1
        :
        ERROR 2
        :
        ERROR 3
        :
        ERROR 4
        :
        :
TST2:   SCOPE
```

THE SEQUENCE OF LOOPING FOR DIFFERENT CASES IS EXPLAINED BELOW. NOTE THAT 'TST1' AND 'TST2' ARE TAGS WHICH DEFINE THE BOUNDARY OF A TEST, (IN THIS CASE TEST 1). TEST 1 STARTS AT 'TST1' AND ENDS JUST BEFORE 'TST2'.

IN THE ILLUSTRATION BELOW --> INDICATES THE POINT FROM WHERE RETURN IS MADE AND LOOPING IS DONE.

1. ERROR 2 OCCURS, SW 14 SET.

TST1..ERROR 2..TST2-->TST1..ERROR 2..TST2-->TST1...

2. ERROR 2 OCCURS, SW 12 SET.

TST1...ERROR 2-->TST1...ERROR2-->TST1...

3. ERROR 2,3; SW 14 SET.

TST1..ERROR 2..ERROR 3..TST2-->TST1..ERROR 2..ERROR 3..TST2-->TST1...

4. ERROR 2,3; SW 12 SET.

TST1...ERROR 2-->TST1...ERROR 2-->TST1....

NOTE THAT LOOPING IS DONE FROM THE VERY FIRST ERROR ENCOUNTERED. THE MORE BASIC AND ERROR THE EARLIER IT OCCURS AND IS DETECTED AND SHOULD BE FIXED.

IN THE ABOVE EXAMPLE NO PART OF THE SUB-TEST IS BEING REPEASING DIFFERENT PARAMETERS, HENCE IT SO HAPPENS THAT SW 9 AND 12 GIVE THE SAME KIND OF LOOPS. THE EXAMPLE BELOW WILL DEMONSTRATE THE DIFFERENCE BETWEEN SW 9 AND 12.

```
TST1:   SCOPE
        :



        INITIALIZATION
        :
        ERROR 1
        :
        MOV     #1$,$LPERR      ;'$LPERR'   CONTAINS
                                ;THE ADDRESS TO LOOP
                                ;BACK ON ERROR- SW 9
1$:     :                       ----
        :                       I
        ER                  I   N REPETITIONS
        :                       I
TST2:   SCOPE                   ----
```

1. SW 12 SET, ERROR 2 OCCURS DURING K.TH REPETITIONS

TST1..1,2...K.ERROR 2-->TST1..1,2...K.ERROR 2-->TST1..

2. SW 9 SET, ERROR 2 OCCURS DURING K.TH REPETITION

1$..K..ERROR 2-->1$..K..ERROR 2-->1$...

9.0 PROGRAM STRUCTURE

THERE ARE THREE DISTINCT PARTS OF THE PROGRAM.

SET-UP PHASE
DRIVE-DEPENDENT CONTROLLER TESTS

9.1 SET-UP PHASE

SETTING UP OF INITIAL POINTERS, VECTORS, TABLES IS DONE IN THIS PART. IN THIS SECTION THE DECISION IS MADE ABOUT THE PROGRAM MODE-PAPER TAPE, RKDP DUMP, CHAIN OR ACT11. IF IN A NON-INTERVENTION MODE (CHAIN, ACT11) NUMBER OF DRIVES AND THE TYPE OF CONTROLLER IS FOUND OUT. FLAGS ARE SET TO INDICATE

WHICH DRIVES ARE TO BE TESTED, ETC.

9.2 DRIVE DEPENDENT CONTROLLER TESTS

THIS SECTION FORMS A MAJOR PART OF THE PROGRAM WHEREIN MOST OF THE CONTROLLER IS CHECKED.

JUST BEFORE ENTERING THIS SECTION THE PROGRAM FINDS OUT WHICH DRIVE IS TO BE CHECKED. IF IN RKDP CHAIN MODE, DRIVE 'N' IF PRESENT, IS SKIPPED AND THE NEXT AVAILABLE DRIVE IS SELECTED.

THE DRIVE NUMBER BEING TESTED IS PRINTED OUT:

```
DRIVE N          ;N=0,1,2...7
```

THE TESTING IS DONE IN A LOGICAL HIERCHY, SIMPLER THINGS FIRST, THEN MORE COMPLEX AND SO ON.

IN ONE OF THE TESTS THE ENTIRE DISK PACK IS FORMATTED, CHECKS ARE MADE FOR ERROR CONDITIONS. THE FIRST WORD OF EVERY SECTOR IS WRITTEN AS A PSUEDO-HEADER, REFLECTING THE ABSOLUTE ADDRESS OF THAT SECTOR (DRIVE #, CYLINDER #, SURFACE #, SECTOR #). EXAMPLE: THE PSUEDO-HEADER FOR SECTOR 5, SURFACE 0, CYLINDER 20, DRIVE 0 WOULD BE 001005.

IN THE NEXT TEST THE HEADERS FROM THE ENTIRE PACK ARE READ AND CHECKED FOR CORRECTNESS. IN A SUBSEQUENT TEST ALL THE PSUEDO-HEADERS ARE READ AND VERIFIED.

ALL THE FUNCTIONS ARE CHECKED OUT. 'SEEK' IS CHECKED IN THE THREE DIFFERENT VELOCITY MODES (HIGH, MEDIUM, LOW). VARIOUS ERRORS LIKE 'NXD', 'NXC', ETC. ARE SIMULATED AND CHECKED.

HARDWARE POGIC IS CHECKED USING ALL THE DRIVES THAT HAVE BEEN INDICATED.

AT THE END OF THIS SECTION, A CHECK IS MADE IF ALL INDICATED DRIVES HAVE BEEN TESTED. IF NOT, CONTROL IS TRANSFERRED TO THE BEGINNING OF THIS SECTION.

THUS ONE PASS OF THE PROGRAM INVOLVES DOING

1. SUBTEST #1 ONCE
2. DRIVE-DEPENDENT TESTS FOR ALL THE SELECTED DRIVES.

10.0 ERROR REPORTING

THE ERROR TABLE STARTING AT $ERRTB CONTAINS INFORMATION PERTAINING TO EVERY ERROR THAT CAN OCCUR. EACH ITEM IN THE TABLE CONSISTS OF FOUR

ENTRIES.

A. EM - THIS IS A POINTER TO THE ERROR MESSAGE TO BE TYPED OUT WHEN THE ERROR OCCURS.

B. DH - THIS IS A POINTER TO THE DATA HEADER TO BE TYPED OUT.

C. DT - THIS IS A POINTER TO THE DATA WHICH IS TO BE TYPED TYPED OUT UNDER THE HEADERS.

D. 0 - THIS IS A TERMINATOR SIGNIFYING THE END OF THE ITEM.

THE ERROR CALL IS AN EMT INSTRUCTION WITH ITS LOWER BYTE ENCODED TO INDICATE THE ERROR NUMBER. THUS OR 1" WOULD BE (EMT+1) IE 104001.

EVERY ERROR CORRESPONDS TO AN ITEM IN THE ERROR TABLE. THUS "ERROR 14" WOULD CORRESPOND TO ITEM 14. AS FAR AS POSSIBLE, THE ERROR MESSAGES HAVE BEEN KEPT SHORT, BUT CLARITY IS NOT SACRIFICED FOR BREVITY. INSPITE OF THIS, IF THE USER FINDS A NEED, HE CAN LOOK UP THE ENTIRE ERROR MESSAGE IN THE ERROR ITEMS TABLE FOUND IN THE BEGINNING OF THE LISTINGS. THUS FOR "ERROR 14", "ITEM 14" IN THE ITEM TABLE CAN BE LOOKED UP. WHEN THE ERROR INSTRUCTION IS EXECUTED A TRAP OCCURS TO THE ERROR HA LOCATED AT $ERROR WHICH PROCESSES THE ERROR CALL. SEE SEC 12.3

11.0 ERROR INTERPRETATION

WHENEVER AN ERROR MESSAGE IS PRINTED OUT, ALL REGISTERS AND OTHER DATA PERTAINING TO THE ERROR ARE ALSO GIVEN. RKDS, RKER...RKBA INDICATE THE CONTENTS OF THE CORRESPONDING REGISTERS AT THE TIME OF ERROR.

EVERY ERROR MESSAGE CONTAINS A PC. THIS PC INDICATES THE POSITION IN PROGRAM WHERE THE ERROR CALL IS LOCATED. THE ERROR MESSAGE, BECAUSE OF PRACTICAL CONSIDERATIONS IS MADE SHORT AND MEANINGFUL. THE USER IS ADVTO LOOK UP THE PC IN THE PROGRAM LISTING, WHERE HE WILL FIND MORE INFORMATION ABOUT THE ERROR. IN MANY INSTANCES, A SINGLE FAULT WILL GIVE RISE TO MORE THAN ONE ERROR REPORT. A LITTLE DELIBERATION AND CAREFUL EXAMINATION OF THE DATA GIVEN WILL BE CERTAINLY VERY HELPFUL IN PINPOINTING THE FAULT. A BRIEF EXPLANATION OF WHAT IS BEING CHECKED IN THE SUBTEST IS GIVEN AT THE BEGINNING OF EVERY SUBTEST. ALL THE NUMBERS GIVEN WITH ERROR MESSAGES ARE IN OCTAL.

12.0 HANDLERS AND COMMON ROUTINES

THE COMMOSED ROUTINES USED IN THE PROGRAM ARE CALLED IN TWO WAYS.

A. AS A SUBROUTINE THROUGH 'JSR' CALL

B. THROUGH A 'TRAP' HANDLER

12.1 TRAP HANDLER

MANY COMMONLY USED ROUTINES IN THE PROGRAM ARE CALLED USING THE TRAP INSTRUCTION AND THE 'TRAP' HANDLER. THE LOWER BYTE OF THE TRAP INSTRUCTION IS ENCODED DIFFERENTLY FOR DIFFERENCT ROUTINES. THE TRAP HANDLER IS LOCATED AT '$TRAP'. WHEN A CALL FOR A ROUTINE IS EXECUTED, A TRAP OCCURS TO THE HANDLER ARAP'. THE HANDLER PICKS UP THE LOWER BYTE OF THE "CALL INSTRUCTION" AND USES IT TO FORM THE STARTING ADDRESS OF THE ROUTINE TO GO TO FOR SERVICE.

12.2 SCOPE HANDLER

THE 'IOT' TRAP IS USED BY THE 'SCOPE' STATEMENT. WHEN 'SCOPE' IS EXECUTED, AN IOT TRAP OCCURS TO MEMORY LOCATION '$SCOPE'. THE SCOPE HANDLER STARTS AT $SCOPE. DEPENDING ON THE SWITCH SETTINGS THE HANDLER DECIDES TO LOOP ON TEXT, INHIBIT ITERATIONS ETC. THERE ARE CERTAIN POINTERS AND FLAGS WHICH ARE ADJUSTED. THUS, IT IS NOT ADVISABLE START THE PROGRAM AT ANY GIVEN LOCATION SINCE THE VARIOUS POINTERS AND FLAGS MAY NOT BE CORRECTLY ADJUSTED.

12.3 ERROR HANDLER

AN EMT TRAP INSTRUCTION IS USED BY THE ERROR CALL. THE LOWER BYTE IS ENCODED TO GIVE DIFFERENT ERROR CALLS. (EX: ERROR 1 = 104000+1; ERROR 16 = 104000+16). WHEN THE ERROR STATEMENT IS EXECUTED, A TRAP OCCURS TO MEMORY LOCATION '$ERROR'. THE ERROR HANDLER IS LOCATED AT '$ERROR'. THE HANDLER FORMS THE POINTER TO ERROR TABLE, WHICH IS USED IF AN ERROR MESSAGE IS TO BE TYPED OUEPENDING ON THE SWITCH SETTINGS, A DECISION ABOUT HALTING ON ERROR, INHIBITING TYPEOUT, LOOPING ON ERROR ETC. IS MADE. IF AN ERROR MESSAGE IS TO BE TYPED OUT AN EXIT IS MADE TO THE ERROR MESSAGE TYPEOUT ROUTINE LOCATED AT '$ERRTYP'.

12.4 CONTROL RESET ROUTINE

THE CALL FOR THIS ROUTINE IS "CNT.RESET" AND IS AN ENCODED 'TRAP' INSTRUCTION. WHEN "CNT.RESET" IS EXECUTED THE CONTROL RESET ROUTINE STARTING AT

"CN.RST" IS ENTERED. A CONTROL RESET IS ISSUED THE PROGRAM WAITS TILL THE CONTROL READY SETS, ON WHICH THE ROUTINE IS EXITED. IF CONTROL READY DOES NOT SET WITHIN A CERTAIN TIME AN ERROR IS REPORTED. THE PC TYPED OUT IS THE LOCATION WHERE THE "CNT.RESET" CALL IS LOCATED. THE WAITING TIME IS 2.8 MS FOR 11/20 AND 560 US FOR 11/45 WITH BIPOLAR MEMORY.

12.5 CONTROL READY ROUTINE

THIS ROUTINE IS CALLED BY "CNT.RDY" (AN ENCODED 'TRAP' INSTRUCTION) AND IS LOCATED AT "CN.RDY". THE ROUTINE WAITS FOR THE CONTROL READY TO SET AND WHEN IT DOES, EXITS IF CONTROL READY DOES NOT SET WITHIN A SPECIFIED TIME AN ERROR MESSAGE IS GIVEN

```
CNTRL RDY DIDN'T SET
PC = XXXXXX     RKCS = YYYYYY
```

THE PC IS THE LOCATION AT WHICH THE "CNT.RDY" CALL IS LOCATED. THE WAITING TIME IS 949 MS FOR 11/20 AND 189 MS FOR 11/45 WITH BIPOLAR MEMORY.

12.6 DRIVE RESET ROUTINE

THE DRIVE - RESET ROUTINE IS LOCATED AT "DRESET" AND IS CALLED BY A "JSR". IT ISSUES A DRIVE RESET AND WAITS FOR THE R/W/S RDY TO SET, ON WHICH THE ROUTINE IS EXITED. THE WAITING TIME IS 4959 MS FOR 11/20 AND 991 MS FOR 11/45 WITH BIPOLAR MEMORY.

12.7 TIME DELAY ROUTINE

THIS ROUTINE PROVIDES A VARIABLE TIME DELAY. THE CALL IS DELAY ,N WHERE N=1 TO 177777 (OCTAL) TIME DELAY PROVIDED= 7.5 TIMES(X) N MICRO SECS FOR 11/20, 1.5N US FOR 11/45 (N CONVERTED TO DECIMAL BEFORE COMPUTING DELAY) IF THE USER WANTS TO CHANGE THE DELAY AT ANY POINT IT CAN BE DONE BY SIMPLY CHANGING VARIABLE 'N'.

12.8 WAIT FOR INTERRUPT ROUTINE

THIS ROUTINE PROVIDES A VARIABLE TIME LIMIT DURING WHICH RK11 INTERRUPT MAY OCCUR. THE IS
WAT.INT .N N=1 TO 177777 (OCTAL)
WAITING TIME=7.5 TIMES(X) N US FOR 11/20, 1.5N US

FOR 11/45 UPON ENTERING THE ROUTINE CPU PRIORITY IS DROPPED SO THAT RK11 CAN INTERRUPT.

12.9 OTHER ROUTINES

THERE ARE OTHER COMMONLY USED ROUTINES AS LISTED BELOW.

$TYPE:
TYPE ROUTINE FOR TYPING OLT ASCII STRINGS.
LOCATED AT "$TYPE"
CALLED BY "TYPE"

$TYPOC:
ROUTINE FOR TYPING OUT OCTAL NUMBERS.
LOCATED AT "$TYPOC"
CALLED BY "TYPOC"

$TYPDS:
ROUTINE FOR TYPING OUT DECIMAL NUMBERS.
LOCATED AT "$TYPDS"
CALLED BY "TYPDS"

$RDLIN:
ROUTINE FOR INPUTTING ASCII STRINGS FROM TTY.
LOCATED AT "$RDLIN"
CALLED BY "RDLIN"

$ERRTYP:
ROUTINE FOR TYPING OUT ERROR MESSAGES.
LOCATED AT $ERRTYP
CALLED BY "JSR $ERRTYP"

$PWRDN:
ROUTINE FOR HANDLING POWER FAILURE.
LOCATED AT $PWRDN
CALLED WHEN THERE IS A POWER FAILURE.

$PWRUP:
ROUTINE FOR HANDLING POWER UP AFTER A POWER FAIL.
LOCATE$PWRUP
CALLED WHEN POWER RETURNS AFTER HAVING GONE DOWN.

13.0 UNEXPECTED TIMEOUTS AND RK11 INTERRUPTS

WHEN AN UNEXPECTED TIMEOUT OCCURS, THE PC AT WHICH TIME OUT OCCURED IS TYPED OUT AND THE PROGRAM HALTS. IF IT IS INTACT, IT CAN BE RESTARTED BY PRESSING CONTINUE.

IF AN UNEXPECTED RK11 INTERRUPT OCCURS THE PROGRAM TYPES OUT THE PC AT WHICH THE INTERRUPT CAME IN AND THEN HALTS. PRESSING CONTINUE WOULD RESTART THE PROGRAM FROM BEGINING. SW 9- LOOPING CAITY IS PROVIDED AS A TROUBLE SHOOTING AID.

14.0 QUICK VERIFYING MODE

THE FIRST PASS OF THE PROGRAM IS A QUICK VERIFYING MODE. ALL THE TESTS ARE DONE ONLY ONCE, ON SUBSEQUENT PASSES THE TESTS ARE ITERATED (NORMALLY 50 TIMES, 5 IN SOME CASES). THUS THE FIRST PASS TAKES A SHORTER TIME TO COMPLETE, WHEREAS SUBSEQUENT PASSES TAKE MORE TIME.

```
                        .TITLE  MD-11-DZRKK-E, RK11 BASIC LOGIC TEST 2
                        ;*COPYRIGHT (C) 1974,1977
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY JIM KAPADIA
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*
                        ;*PROGRAM REVISED BY TOM SAWYER, MARCH, 1976
                        ;*REVISED BY CHUCK HESS, AUGUST, 1976
                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        12            CYCLE ON ERROR TO PREVIOUS 'SCOPE' STATEMENT
                        ;*        11            INHIBIT ITERATIONS
                        ;*        10            TESTING ON SIMULATOR
                        ;*         9            LOOP ON ERROR
                        ;*         8            LOOP ON TEST IN SWR<7:0>
                        ;*         6            DROP THE DRIVE IF MORE THAN 5 ERRORS


                        ;*****************************************************************
                        ;YOU ARE ADVISED TO READ THE DOCUMENT BEFORE USING THIS PROGRAM.
                        ;ON GETTING AN ERROR REFER TO THE LISTINGS AT THE PC POINTED
```

```
                                ;OUT IN THE ERROR MESSAGE. ADJACENT ERROR MESSAGES IF FOLLOWED
                                ;CAREFULLY COULD LEAD TO AN EASY PINPOINTING OF THE FAULT
                                
                                ;*****************************************************************
                                .SBTTL  BASIC DEFINITIONS
                                
                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
003100                          STACK=  1100
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
                                
                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE FOR LINE FEED
000015                          CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
                                
                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
000004                          R4=     %4              ;;GENERAL REGISTER
000005                          R5=     %5              ;;GENERAL REGISTER
000006                          R6=     %6              ;;GENERAL REGISTER
000007                          R7=     %7              ;;GENERAL REGISTER
000006                          SP=     %6              ;;STACK POINTER
000007                          PC=     %7              ;;PROGRAM COUNTER
                                
                                ;*PRIORITY LEVEL DEFINITIONS
000000                          PR0=    0               ;;PRIORITY LEVEL 0
000040                          PR1=    40              ;;PRIORITY LEVEL 1
000100                          PR2=    100             ;;PRIORITY LEVEL 2
000140                          PR3=    140             ;;PRIORITY LEVEL 3
000200                          PR4=    200             ;;PRIORITY LEVEL 4
000240                          PR5=    240             ;;PRIORITY LEVEL 5
000300                          PR6=    300             ;;PRIORITY LEVEL 6
000340                          PR7=    340             ;;PRIORITY LEVEL 7
                                
                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                          SW15=   100000
040000                          SW14=   40000
020000                          SW13=   20000
010000                          SW12=   10000
004000                          SW11=   4000
002000                          SW10=   2000
001000                          SW09=   1000
000400                          SW08=   400
000200                          SW07=   200
000100                          SW06=   100
```

```
                        
                        SW05=   40
                        SW04=   20
                        SW03=   10
                        SW02=   4
                        SW01=   2
                        SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0
                        
                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                        BIT15=  100000
                        BIT14=  40000
                        BIT13=  20000
                        BIT12=  10000
                        BIT11=  4000
                        BIT10=  2000
                        BIT09=  1000
                        BIT08=  400
                        BIT07=  200
                        BIT06=  100
                        BIT05=  40
                        BIT04=  20
                        BIT03=  10
                        BIT02=  4
                        BIT01=  2
                        BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0
                        
                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
```

```
1006            000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
1007            000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
1008                                    .SBTTL  TRAP CATCHER
1009
1010            000000                          .=0
1011                                    ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
1012                                    ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1013                                    ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1014            000174                          .=174
1015  000174    000000                DISPREG:  .WORD  0              ;;SOFTWARE DISPLAY REGISTER
1016  000176    000000                SWREG:    .WORD  0              ;;SOFTWARE SWITCH REGISTER
1017                                    .SBTTL  STARTING ADDRESS(ES)
1018  000200    000137  002636                JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
1019                                    .SBTTL  ACT11 HOOKS
1020
1021                                    ;;*****************************************************************
1022                                    ;HOOKS REQUIRED BY ACT11
1023            000204                          $SVPC=.                 ;SAVE PC
1024            000046                          .=46
1025  000046    020734                        $ENDAD                    ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SECP
1026            000052                          .=52
1027  000052    000000                        .WORD   0                 ;;2)SET LOC.52 TO ZERO
1028            000204                          .=$SVPC                 ;; RESTORE PC
```

```
                                .SBTTL  COMMON TAGS

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

                                .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000          $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
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          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
001162                                                  ;;WHICH  ($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          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001210  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001212  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001213  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001214  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                        ;;**********************************************************************
001216  005015  051104  053111  MSG1:   .ASCIZ  <15><12>/DRIVE PRESNT/
```

```
1085  001224  020105  051120  051505
1086  001232  052116     000
1087          001236                          .EVEN
1088  001236  005015  047516  042516  MSG2:   .ASCIZ  <15><12>/NONE/
1089  001244     000
1090
1091  001245     015  041412  052116  MSG3:   .ASCIZ  <15><12>/CNT RDY DIDN'T SET/
1092  001252  051040  054504  042040
1093  001260  042111  023516  020124
1094  001266  042523  000124
1095
1096  001272  005015  051104  053111  MSG4:   .ASCIZ  <15><12>/DRIVE /
1097  001300  020105     000
1098
1099  001303     015  040412  046114  MSG5:   .ASCII  <15><12>/ALL DRVS/
1100  001310  042040  053122     123
1101
1102  001315     040  051104  050117  MSG6:   .ASCIZ  / DROPD/<15><12>
1103  001322  006504  000012
1104                                          .EVEN
1105
1106                                  ;RK11 REGISTERS
1107                                  ;IF FOR ANY REASON THE REGISTER ADDRESSES ARE DIFFERENT FROM THESE
1108                                  ;(GIVEN BELOW), THE CONTENTS OF THE APPROPRIATE POINTERS SHOULD BE
1109                                  ;MODIFIED SO THAT THE CORRECT ADDRESS IS USED.
1110                                  ;
1111                                          .EVEN
1112  001326  177400                  RKDS:   177400
1113  001330  177402                  RKER:   177402
1114  001332  177404                  RKCS:   177404
1115  001334  177406                  RKWC:   177406
1116  001336  177410                  RKBA:   177410
1117  001340  177412                  RKDA:   177412
1118  001342  177416                  RKDB:   177416
1119
1120
1121                                  ;TAGS AND GENERAL DATA AREA
1122                                  ;
1123                                  ;
1124
1125  001344  000000                  SIMUL:  0               ;FLAG TO BE SET TO 1 WHEN ON SIMULATOR
1126  001346  000000                  FTITLE: 0               ;FLAG FOR PRINTING PROGRAM TITLE
1127  001350  000000                  DRIVAD: 0               ;CONTAINS ADDRESS OF THE DRIVE UNDER TEST
1128  001352  000000                  DRVDON: 0               ;CONTAINS THE NUMBER OF DRIVES CHECKED.
1129                                                          ;IT IS INCREMENTED EACH TIME THE TESTS FOR
1130                                                          ;A DRIVE IS COMPLETED.
1131  001354  000000                  DRVPTR: 0               ;CONTAINS THE POINTER TO THE DRIVE FLAG (DRIVEC
1132                                                          ;-DRIVE7) OF THE DRIVE TO BE CHECKED NEXT.
1133  001356  000000                  INDX1:  0               ;GENERAL INDEX FOR KEEPING COUNT
1134  001360  000000                  INDX2:  0               ; GENERAL INDEX
1135  001362  000000                  COUNT:  0               ;GENERAL COUNT REGISTER
1136  001364  000000                  COUNT1: 0               ;COUNT REGISTER USED FOR 'DRESET' SUBROUTINE
1137  001366  000000                  TIMER:  0               ;TIMER REGISTER
1138  001370  000000                  EFLG1:  0               ;SET, TO INDICATE A PARTICULAR
1139                                                          ;ERROR CONDITION
1140
```

```
1141  001372  000100                  SEEK0:  100             ;CONTAINS ADDRESS OF CYLINDER 2
1142  001374  001000                  SEEK1:  1000            ;CONTAINS ADDRESS OF CYLINDER 20
1143  001376  014500                  SEEK2:  14500           ;CONTAINS ADDRESS OF CYLINDER 312
1144  001400  000200                  RKPRI:  200             ;CONTAINS THE CPU LEVEL AT WHICH
1145                                                          ;RK11 NORMALLY INTERRUPTS. THIS WORD
1146                                                          ;SHOULD BE CHANGED IF RK11 IS DESINGATED
1147                                                          ;A BR LEVEL OTHER THAN 5, E.G. IF IT IS CHANGED
1148                                                          ;TO 5, THIS WORD SHOULD BE CHANGED TO 240.
1149  001402  000220                  RKVEC:  220             ;CONTAINS THE NORMAL VECTOR ADDRESS TO WHICH
1150                                                          ;RK11 INTERRUPTS. IF THIS IS NOT SO, CHANGE
1151                                                          ;THIS WORD TO CONTAIN MODIFIED VECTOR ADDRESS.
1152  001404  000000                  FFLAG:  0
1153  001406  000000                  ODDEVN: 0               ;USED TO DETERMINE WHICH OF RK-05F DRIVES ACTIVE
1154                                                          ;0 IF EVEN DRIVE
1155                                                          ;-1 IF ODD DRIVE
1156  001410  000000                  DDPCH:  0               ;IF PROGRAM LOADED FROM RK05, CONTAINS
1157                                                          ;ADDRESS OF DRIVE WITH RKDP PACK
1158  001412  000000                  DRIVS:  0               ;CONTAINS THE NUMBER OF DRIVES PRESENT
1159
1160
1161
1162
1163                                  ;THE FLAGS BELOW (BIT 0) ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE
1164                                  ;IS PRESENT AND IS TO BE TESTED. BIT 12, IF SET, INDICATES THAT THE DRIVE
1165                                  ;WAS DROPPED AFTER MAXIMUM ALLOWABLE NUMBER OF ERRORS OCCURED ON THAT
1166                                  ;DRIVE (SW 6 SET).
1167                                  ;IF MORE THAN 5 ERRORS OCCUR IN THE HARDWARE POLLING TEST (LAST)
1168                                  ;THEN ALL DRIVES ARE DROPPED. BUT BIT 12 IS NOT SET.
1169
1170  001414  000000                  DRIV0:  0               ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
1171  001416  000000                  DRIV1:  0               ;FOR DRIVE 1
1172  001420  000000                  DRIV2:  0               ;FOR DRIVE 2
1173  001422  000000                  DRIV3:  0               ;FOR DRIVE 3
1174  001424  000000                  DRIV4:  0               ;FOR DRIVE 4
1175  001426  000000                  DRIV5:  0               ;FOR DRIVE 5
1176  001430  000000                  DRIV6:  0               ;FOR DRIVE 6
1177  001432  000000                  DRIV7:  0               ;FOR DRIVE 7
1178
1179  001434  000000                  T56FLG: 0
1180  001436  000000                  PHYDRV: 0
1181  001440  000000                  SIZYET: 0
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TAPLE 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


        001442                  $ERRTB:




                                ;THE ERROR ITEMS TABLE CONSISTS OF ALL THE POSSIBLE ERROR MESSAGES
                                ;USED IN THIS PROGRAM. AN ERROR CALL IN THE PROGRAM CORRESPONDS TO
                                ;THE ITEM NUMBER IN THE ERROR TABLE. THUS 'ERROR       1' IN THE
                                ;PROGRAM CORRESPONDS TO 'ITEM   1' IN THE ERROR TABLE.
                                ;'EM###' IS THE  POINTER TO THE ERROR MESSAGE WHICH WILL BE TYPED
                                ;OUT IN CASE THAT ERROR WERE TO OCCUR. THUS FOR 'ERROR  1' THE ERROR
                                ;MESSAGE TYPE OUT WILL BE 'TIME OUT ON RK11 REG'.
                                ;'DH###' IS THE POINTER TO THE HEADER BLOCK WHICH WILL BE TYPED OUT
                                ;IMMEDIATELY AFTER THE ERROR MESSAGE.
                                ;'DT###'  SERVES AS A POINTER TO THE MEMORY LOCATIONS WHERE
                                ;THE INFORMATION RELEVANT TO THE ERROR TYPE OUTS (LIKE PC, CONTENTS
                                ;OF RKCS ETC.) WILL BE PICKED UP FROM.
                                ;THE LAST ROW CONTAINING '0' SERVES AS A TERMINATOR.
                                ;EXAMPLE:
                                ;IF ON RUNNING THIS PROGRAM A TIMEOUT WERE TO OCCUR ON ADDRESSING RKDS
                                ;(177400), BECAUSE OF SOME FAULT, THE FOLOWING TYPEOUT WOULD
                                ;OCCUR ON THE TELETYPE.
                                ;
                                ;         TIME OUT ON RK11 REG
                                ;           PC        REG
                                ;         ######    177400
                                ;
                                ;NOTE THAT ###### WOULD BE THE ACTUAL PC WHERE 'ERROR   1' IS LOCATED.

                                ;THE ERROR HANDLER IS LOCATED AT '$ERROR'. THE ERROR CALL IS AN 'EMT'
                                ;INSTRUCTION WITH ITS LOWER BYTE ENCODED TO PROVIDE INDEXING TO THE
                                ;ITEMS IN THE ERROR TABLE.
                                ;THUS 'ERROR    1'   IS 104001
                                ;     'ERROR    103' IS 104126   ETC.





                                        ;ERROR ITEMS TABLE

```

```
1238                                    ;ITEM   1
1239
1240  001442  025376                            EM12    ;'SIN' IS SET
1241  001444  032323                            DH44    ;PC     RKCS    RKER    RKDS    RKDA
1242  001446  031736                            DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
1243  001450  000000                            0
1244
1245                                    ;ITEM   2
1246
1247  001452  027247                            EM70    ;CNTRL RDY DIDN'T SET ON READ/FMT FROM DISK ADDRESS
1248  001454  032107                            DH14    ;PC     RKCS    RKER    RKWC
1249  001456  031756                            DT26    ;$ERRPC $REG0   $REG1   $REG2
1250  001460  000000                            0
1251
1252                                    ;ITEM   3
1253
1254  001462  025423                            EM16    ;RKDA WRONG AFTER SSE
1255  001464  032043                            DH4     ;PC     EXPCT   RECVD
1256  001466  031726                            DT2     ;$ERRPC $REG0   $REG1
1257  001470  000000                            0
1258
1259                                    ;ITEM   4
1260
1261  001472  025452                            EM21    ;RKDS ERROR
1262  001474  032207                            DH34    ;PC     RKDS
1263  001476  031720                            DT1     ;$ERRPC $REG0
1264  001500  000000                            0
1265
1266                                    ;ITEM   5
1267
1268  001502  025464                            EM30    ;'DPL' BIT SET, CHECK DRIVE POWER
1269  001504  032151                            DH30    ;PC     RKCS    RKER    RKDS
1270  001506  031756                            DT26    ;$ERRPC $REG0   $REG1   $REG2
1271  001510  000000                            0
1272
1273                                    ;ITEM   6
1274
1275  001512  025474                            EM31    ;'DRU' BIT SET, CHECK DRIVE
1276  001514  032151                            DH30    ;PC     RKCS    RKER    RKDS
1277  001516  031756                            DT26    ;$ERRPC $REG0   $REG1   $REG2
1278  001520  000000                            0
1279
1280                                    ;ITEM   7
1281
1282  001522  025504                            EM32    ;'RKOS' BIT NOT SET
1283  001524  032207                            DH34    ;PC     RKDS
1284  001526  031720                            DT1     ;$ERRPC $REG0
1285  001530  000000                            0
1286
1287                                    ;ITEM   10
1288
1289  001532  025525                            EM33    ;'DRY' NOT SET
1290  001534  032323                            DH44    ;PC     RKCS    RkER    RKDS    RKDA
1291  001536  031736                            DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
1292  001540  000000                            0
1293
```

```
1294                                    ;ITEM   11
1295
1296  001542  025545                            EM34    ;'SOK' DID NOT SET
1297  001544  032207                            DH34    ;PC     RKDS
1298  001546  031720                            DT1     ;$ERRPC $REGO
1299  001550  000000                            0
1300
1301                                    ;ITEM   12
1302
1303  001552  025564                            EM35    ;'SEC COUNTR' DIDN'T COUNT TO 0
1304  001554  032225                            DH35    ;PC     SEC-CNTR
1305  001556  031720                            DT1     ;$ERRPC $REGO
1306  001560  000000                            0
1307
1308                                    ;ITEM   13
1309
1310  001562  025617                            EM36    ;'SEC COUNTR' DIDN'T INCREMENT
1311  001564  032245                            DH36    ;PC     PRSNT-COUNT     NXT-COUNT
1312  001566  031726                            DT2     ;$ERRPC $REGO           $REG1
1313  001570  000000                            0
1314
1315                                    ;ITEM   14
1316
1317  001572  025647                            EM37    ;'SECTOR COUNTER' INCREMENTED WRONG
1318  001574  032043                            DH4     ;PC     EXPCTD  RECVD
1319  001576  031726                            DT2     ;$ERRPC $REGO   $REG1
1320  001600  000000                            0
1321
1322                                    ;ITEM   15
1323
1324  001602  025703                            EM40    ;DIDN'T GET SC=SA FOR THIS SECTOR
1325  001604  032275                            DH40    ;PC     SECTOR  RKDS
1326  001606  031726                            DT2     ;$ERRPC $REGO   $REG1
1327  001610  000000                            0
1328
1329                                    ;ITEM   16
1330
1331  001612  025743                            EM41    ;ERROR-'R/W/S RDY' SHOULD BE SET.
1332  001614  032207                            DH34    ;PC     RKDS
1333  001616  031720                            DT1     ;$ERRPC $REGO
1334  001620  000000                            0
1335
1336                                    ;ITEM   17
1337
1338  001622  025411                            EM13    ;RKBA ERROR
1339  001624  032043                            DH4     ;PC     EXPCT   RECVD
1340  001626  031726                            DT2     ;$ERRPC $REGO   $REG1
1341  001630  000000                            0
1342
1343                                    ;ITEM   20
1344
1345  001632  026000                            EM43    ;UNEXPECTED RK11 INTERRUPT
1346  001634  032144                            DH21    ;PC
1347  001636  031752                            DT21    ;$ERRPC
1348  001640  000000                            0
1349
```

```
1350                                    ;ITEM   21
1351
1352  001642  026032                            EM44    ;'CNTRL RDY' DIDN'T SET AFTER SEEK OR DRIVE RESET
1353  001644  032323                            DH44    ;PC     RKCS    RKER    RKDS    RKDA
1354  001646  031736                            DT20    ;$ERRPC $REGO   $REG1   $REG2   $REG3.
1355  001650  000000                            0
1356
1357                                    ;ITEM   22
1358
1359  001652  026106                            EM45    ;'ERR' OR 'HE' SET ON SEEK OR DRIVE RESET
1360  001654  032323                            DH44    ;PC     RKCS    RKER    RKDS    RKDA
1361  001656  031736                            DT20    ;$ERRPC $REGO   $REG1   $REG2   $REG3
1362  001660  000000                            0
1363
1364                                    ;ITEM   23
1365
1366  001662  026154                            EM46    ;RKER BIT, ON SEEK OR DRIVE RESET
1367  001664  032151                            DH30    ;PC     RKCS    RKER    RKDS
1368  001666  031756                            DT26    ;$ERRPC $REGO   $REG1   $REG2
1369  001670  000000                            0
1370
1371                                    ;ITEM   24
1372
1373  001672  026212                            EM47    ;RKCS CHANGED AFTER FUNCTION WAS DONE
1374  001674  032043                            DH4     ;PC     EXPCT   RECVD
1375  001676  031726                            DT2     ;$ERRPC $REGO   $REG1
1376  001700  000000                            0
1377
1378                                    ;ITEM   25
1379
1380  001702  026254                            EM50    ;'R/W/S RDY' DID NOT CLEAR
1381  001704  032151                            DH30    ;PC     RKCS    RKER    RKDS
1382  001706  031756                            DT26    ;$ERRPC $REGO   $REG1   $REG2
1383  001710  000000                            0
1384
1385                                    ;ITEM   26
1386
1387  001712  026303                            EM51    ;'R/W/S RDY' DIDN'T SET AFTER SEEK OR DRIVE RESET
1388  001714  032323                            DH44    ;PC     RKCS    RKER    RKDS    RKDA
1389  001716  031736                            DT20    ;$ERRPC $REGO   $REG1   $REG2   $REG3
1390  001720  000000                            0
1391
1392                                    ;ITEM   27
1393
1394  001722  026356                            EM52    ;RKDA CHANGED AFTER SEEK
1395  001724  032043                            DH4     ;PC     EXPCTD  REGVD
1396  001726  031726                            DT2     ;$ERRPC $REGO   $REG1
1397  001730  000000                            0
1398
1399                                    ;ITEM   30
1400
1401  001732  026403                            EM53    ;'CNTRL RDY' DIDN'T CLEAR AS GO WAS SET
1402  001734  032151                            DH30    ;PC     RKCS    RKER    RKDS
1403  001736  031756                            DT26    ;$ERRPC $REGO   $REG1   $REG2
1404  001740  000000                            0
1405
```

```
1406                                    ;ITEM   31
1407
1408  001742  026446                              EM54    ;'CNTRL RDY' DIDN'T SET ON DOING WRITE/FMT STARTING
1409                                                    ;FROM <DSK-ADRES>
1410  001744  032370                              DH54    ;PC     RKCS    RKER    RKDS    RKDA
1411                                                    ;DRV#   CYL   <DSK-ADRES>          SUR     SECTR
1412  001746  031770                              DT54    ;$ERRPC $REG0   $REG1   $REG2   $REG3
1413                                                    ;$REG4  $REG5   $REG6   $REG7
1414  001750  000000                              0
1415
1416                                    ;ITEM   32
1417
1418  001752  026540                              EM55    ;'HE' OR 'ERR'  ON WRITE/FMT STARTING FROM
1419                                                    ;<DSK-ADRES>
1420  001754  032370                              DH54    ;PC     RKCS    RKER    RKDS    RKDA
1421                                                    ;DRV#   CYL   <DSK-ADRES>          SUR     SECTR
1422  001756  031770                              DT54    ;$ERRPC $REG0   $REG1   $REG2   $REG3
1423                                                    ;$REG4  $REG5   $REG6   $REG7
1424  001760  000000                              0
1425
1426                                    ;ITEM   33
1427
1428  001762  026617                              EM56    ;RKDA INCREMENTED WRONG ON WRITE OR WRITE FORMAT
1429  001764  032477                              DH56    ;PC     EXPCT:  DRV#    CYL     SUR     SECTR
1430                                                    ;RECVD: DRV#    CYL     SUR     SECTR
1431  001766  031770                              DT54    ;$ERRPC         $REG0   $REG1   $REG2   $REG3
1432                                                    ;$REG4  $REG5   $REG6   $REG7
1433  001770  000000                              0
1434
1435                                    ;ITEM   34
1436
1437  001772  026656                              EM57    ;RKWC DIDN'T OVERFLOW ON WRITE OR WRITE FORMAT
1438  001774  032071                              DH5     ;PC     RECVD
1439  001776  031720                              DT1     ;$ERRPC $REG0
1440  002000  000000                              0
1441
1442                                    ;ITEM   35
1443
1444  002002  026714                              EM60    ;RKBA INCREMENTED WRONG ON WRITE OR WRITE FORMAT
1445  002004  032043                              DH4     ;PC     EXPCT   RECVD
1446  002006  031726                              DT2     ;$ERRPC $REG0   $REG1
1447  002010  000000                              0
1448
1449                                    ;ITEM   36
1450
1451  002012  026753                              EM61    ;RKER SET, ON WRITE/READ/FORMAT
1452  002014  032151                              DH30    ;PC     RKCS    RKER    RKDS
1453  002016  031756                              DT26    ;$ERRPC $REG0   $REG1   $REG2
1454  002020  000000                              0
1455
1456                                    ;ITEM   37
1457
1458  002022  027010                              EM62    ;RKDB ERROR
1459  002024  032043                              DH4     ;PC     EXPCT   RECVD
1460  002026  031726                              DT2     ;$ERRPC $REG0   $REG1
1461  002030  000000                              0
```

```
1462
1463                                    ;ITEM   40
1464
1465  002032  027022                            EM63    ;RKDA INCREMENTED WRONG ON READ OR READ FORMAT
1466  002034  032477                            DH56    ;PC     EXPCT:  DRV#    CYL     SUR     SECTR
1467                                                    ;RECVD: DRV#    CYL     SUR     SECTR
1468  002036  031770                            DT54    ;$ERRPC         $REG0   $REG1   $REG2   $REG3
1469                                                    ;$REG4  $REG5   $REG6   $REG7
1470  002040  000000                            0
1471
1472                                    ;ITEM   41
1473
1474  002042  027066                            EM64    ;RKWC DID NOT OVERFLOW ON READ OR READ FORMAT
1475  002044  032604                            DH64    ;PC     RKWC    RKDA
1476  002046  031726                            DT2     ;$ERRPC $REG0   $REG1
1477  002050  000000                            0
1478
1479                                    ;ITEM   42
1480
1481  002052  027131                            EM65    ;RKBA INCREMENTED WRONG ON READ OR READ FORMAT
1482  002054  032043                            DH4     ;PC     EXPCT   RECVD
1483  002056  031726                            DT2     ;$ERRPC $REG0   $REG1
1484  002060  000000                            0
1485
1486                                    ;ITEM   43
1487
1488  002062  027175                            EM66    ;INCORRECT HEADER FROM 'SECTOR'
1489  002064  032630                            DH66    ;PC     SECTR   EXPCT   RECVD
1490  002066  031756                            DT26    ;$ERRPC $REG0   $REG1   $REG2.
1491  002070  000000                            0
1492
1493                                    ;ITEM   44
1494
1495  002072  027234                            EM67    ;DATA ERROR
1496  002074  032666                            DH67    ;PC     EXPCT   RECVD   DSK-ADRES
1497  002076  031756                            DT26    ;$ERRPC $REG0   $REG1   $REG2
1498  002100  000000                            0
1499
1500                                    ;ITEM   45
1501
1502  002102  027247                            EM70    ;'CNTRL RDY' DIDN'T SET ON DOING READ/FMT STARTING
1503                                                    ; FROM <DSK-ADRES>
1504  002104  032370                            DH54    ;PC     RKCS    RKER    RKDS    RKDA
1505                                                    ;DRV#   CYL     <DSK-ADRES>     SUR     SECTR
1506  002106  031770                            DT54    ;$ERRPC $REG0   $REG1   $REG2   $REG3
1507                                                    ;$REG4  $REG5   $REG6   $REG7
1508  002110  000000                            0
1509
1510                                    ;ITEM   46
1511
1512  002112  027340                            EM71    ;'HE' OR 'ERR' BIT SET ON READ/FMT STARTING
1513                                                    ; FROM <DSK-ADRES>
1514  002114  032370                            DH54    ;PC     RKCS    RKER    RKDS    RKDA
1515                                                    ;DRV#   CYL     <DSK-ADRES>     SUR     SECTR
1516  002116  031770                            DT54    ;$ERRPC $REG0   $REG1   $REG2   $REG3
1517                                                    ;$REG4  $REG5   $REG6   $REG7
```

ERROR POINTER TABLE

```
1518  002120  000000                        0
1519
1520                                ;ITEM   47
1521
1522  002122  027416                        EM72    ;WRONG DRIVE ID IN RKDS AFTER SEEK
1523  002124  032043                        DH4     ;PC     EXPCT   RECVD
1524  002126  031726                        DT2     ;$ERRPC $REGO   $REG1
1525  002130  000000                        0
1526
1527                                ;ITEM   50
1528
1529  002132  027460                        EM73    ;HARDWARE POLL, DRIVE ID BITS (13-15) SHOULD BE CLEAR
1530  002134  032207                        DH34    ;PC     RKDS
1531  002136  031726                        DT2     ;$ERRPC $REGO
1532  002140  000000                        0
1533
1534                                ;ITEM   51
1535
1536  002142  027532                        EM74    ;HARDWARE POLL, INTERRUPTING DRIVE # NOT PRESENT
1537  002144  032726                        DH74    ;PC     DRIVE #
1538  002146  031720                        DT1     ;$ERRPC $REGO
1539  002150  000000                        0
1540
1541                                ;ITEM   52
1542
1543  002152  027602                        EM75    ;'DRIVE #' DID NOT INTERRUPT DURING HARDWARE POLL
1544  002154  032726                        DH74    ;PC     DRIVE #
1545  002156  031720                        DT1     ;$ERRPC $REGO
1546  002160  000000                        0
1547
1548                                ;ITEM   53
1549
1550  002162  027652                        EM76    ;SCP DID NOT SET AFTER WAS DONE
1551  002164  033102                        DH117   ;PC     RKCS
1552  002166  031720                        DT1     ;$ERRPC $REGO
1553  002170  000000                        0
1554
1555                                ;ITEM   54
1556
1557  002172  027715                        EM77    ;RKDA CHANGED AFTER 'DRIVE RESET'
1558  002174  032043                        DH4     ;PC     EXPCT   RECVD
1559  002176  031726                        DT2     ;$ERRPC $REGO   $REG1
1560  002200  000000                        0
1561
1562                                ;ITEM   55
1563
1564  002202  027752                        EM100   ;DATA ERROR AT WORD#
1565  002204  032747                        DH100   ;PC     WORD#   EXPCT   RECVD
1566  002206  031756                        DT26    ;$ERRPC $REGO   $REG1   $REG2
1567  002210  000000                        0
1568
1569                                ;ITEM   56
1570
1571  002212  027775                        EM101   ;CNTRL RDY DID NOT SET AFTER READ CHECK
1572  002214  032323                        DH44    ;PC     RKCS    RKER    RKDS    RKDA
1573  002216  031736                        DT20    ;$ERRPC $REGO   $REG1   $REG2   $REG3
```

```
1574  002220  000000                          0
1575
1576                                  ;ITEM   57
1577
1578  002222  030037                          EM102   ;'ERR' OF 'HE' SET ON READ CHECK
1579  002224  032151                          DH30    ;PC     RkCS    RKER    RKDS
1580  002226  031756                          DT26    ;$ERRPC $REG0   $REG1   $REG2
1581  002230  000000                          0
1582
1583                                  ;ITEM   60
1584
1585  002232  030063                          EM103   ;'CSE' ON READ CHECK
1586  002234  033004                          DH103   ;PC     RKER
1587  002236  031720                          DT1     ;$ERRPC $REG0
1588  002240  000000                          0
1589
1590                                  ;ITEM   61
1591
1592  002242  030101                          EM104   ;RKWC DID NOT OVERFLOW ON READ CHECK OR WRITE CHECK
1593  002244  033020                          DH104   ;PC     RECVD   RkCS
1594  002246  031726                          DT2     ;$ERRPC $REG0   $REG1
1595  002250  000000                          0
1596
1597                                  ;ITEM   62
1598
1599  002252  030152                          EM105   ;RKDA INCREMENTED WRONG ON READ CHECK
1600  002254  032043                          DH4     ;PC     EXPCT   RECVD
1601  002256  031726                          DT2     ;$ERRPC $REG0   $REG1
1602  002260  000000                          0
1603
1604                                  ;ITEM   63
1605
1606  002262  030210                          EM106   ;RKBA CHANGED AFTER READ CHECK
1607  002264  032043                          DH4     ;PC     EXPCT   RECVD
1608  002266  031726                          DT2     ;$ERRPC $REG0   $REG1
1609  002270  000000                          0
1610
1611                                  ;ITEM   64
1612
1613  002272  030241                          EM107   ;MEMORY WORD CHANGED AFTER READ CHECK
1614  002274  033044                          DH107   ;PC     LOC     EXPCT   RECVD
1615  002276  031756                          DT26    ;$ERRPC $REG0   $REG1   $REG2
1616  002300  000000                          0
1617
1618                                  ;ITEM   65
1619
1620  002302  030302                          EM110   ;CNTRL RDY DID NOT SET AFTER WRITE CHECK
1621  002304  032323                          DH44    ;PC     RKCS    RKER    RKDS    RKDA
1622  002306  031736                          DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
1623  002310  000000                          0
1624
1625                                  ;ITEM   66
1626
1627  002312  030345                          EM111   ;HE OR ERR BIT SET AFTER DOING WRITE CHECK
1628  002314  032151                          DH30    ;PC     RKCS    RKER    RKDS
1629  002316  031756                          DT26    ;$ERRPC $REG0   $REG1   $REG2
```

```
1630  002320  000000                        0
1631
1632                                ;ITEM   67
1633
1634  002322  030372                        EM112   ;WRITE CHECK ERROR
1635  002324  032151                        DH30    ;PC     RKCS    RKER    RKDS
1636  002326  031756                        DT26    ;$ERRPC $REG0   $REG1   $REG2
1637  002330  000000                        0
1638
1639                                ;ITEM   70
1640
1641  002332  030413                        EM113   ;RKDA INCREMENTED WRONG ON WRITE CHECK
1642  002334  032043                        DH4     ;PC     EXPCT   RECVD
1643  002336  031726                        DT2     ;$ERRPC $REG0   $REG1
1644  002340  000000                        0
1645
1646                                ;ITEM   71
1647
1648  002342  030452                        EM114   ;RKBA INCREMENTED WRONG ON WRITE CHECK
1649  002344  032043                        DH4     ;PC     EXPCT   RECVD
1650  002346  031726                        DT2     ;$ERRPC $REG0   $REG1
1651  002350  000000                        0
1652
1653                                ;ITEM   72
1654
1655  002352  030511                        EM115   ;RKBA INCREMENTED WITH IBA SET
1656  002354  032043                        DH4     ;PC     EXPCT   RECVD
1657  002356  031726                        DT2     ;$ERRPC $REG0   $REG1
1658  002360  000000                        0
1659
1660                                ;ITEM   73
1661
1662  002362  030545                        EM116   ;WRONG MEMORY LOCATION CHANGED WITH IBA SET
1663  002364  032747                        DH100   ;PC     WORD#   EXPCT   RECVD
1664  002366  031756                        DT26    ;$ERRPC $REG0   $REG1   $REG2
1665  002370  000000                        0
1666
1667                                ;ITEM   74
1668
1669  002372  030620                        EM117   ;RK11 DID NOT INTERRUPT WHEN IDE WAS SET
1670  002374  033102                        DH117   ;PC     RKCS
1671  002376  031720                        DT1     ;$ERRPC $REG0
1672  002400  000000                        0
1673
1674                                ;ITEM   75
1675
1676  002402  030665                        EM120   ;RK11 DID NOT INTERRUPT AFTER SEEK WAS INITIATED
1677  002404  033102                        DH117   ;PC     RKCS
1678  002406  031720                        DT1     ;$ERRPC $REG0
1679  002410  000000                        0
1680
1681                                ;ITEM   76
1682
1683  002412  030740                        EM121   ;SCP SET BEFORE SEEK COMPLETED
1684  002414  033102                        DH117   ;PC     RKCS
1685  002416  031720                        DT1     ;$ERRPC $REG0
```

```
1686  002420  000000                          0
1687
1688                                  ;ITEM   77
1689
1690  002422  030776                          EM122   ;RK11 DID NOT INTERRUPT AFTER SEEK COMPLETED
1691  002424  032151                          DH30    ;PC     RKCS    RKER    RKDS
1692  002426  031756                          DT26    ;$ERRPC $REG0   $REG1   $REG2
1693  002430  000000                          0
1694
1695                                  ;ITEM   100
1696
1697  002432  031045                          EM123   ;CNTRL RESET DID NOT CLEAR 'SCP' BIT
1698  002434  033102                          DH117   ;PC     RKCS
1699  002436  031720                          DT1     ;$ERRPC $REG0
1700  002440  000000                          0
1701
1702                                  ;ITEM   101
1703
1704  002442  031104                          EM124   ;RK11 DID NOT INTERRUPT AFTER READ WAS DONE
1705  002444  033102                          DH117   ;PC     RKCS
1706  002446  031720                          DT1     ;$ERRPC $REG0
1707  002450  000000                          0
1708
1709                                  ;ITEM   102
1710
1711  002452  031146                          EM125   ;CNTRL RESET DID NOT CLEAR REGISTER
1712  002454  032014                          DH2     ;PC     REGADD  RECVD
1713  002456  031726                          DT2     ;$ERRPC $REG0   $REG1
1714  002460  000000                          0
1715
1716                                  ;ITEM   103
1717
1718  002462  031205                          EM126   ;RK11 DID NOT INTERRUPT AT CPU LEVEL
1719  002464  033116                          DH126   ;PC     LEVEL   RKCS
1720  002466  031726                          DT2     ;$ERRPC $REG0   $REG1
1721  002470  000000                          0
1722
1723                                  ;ITEM   104
1724
1725  002472  031246                          EM127   ;RK11 INTERRUPTED AT WRONG CPU LEVEL
1726  002474  033116                          DH126   ;PC     LEVEL   RKCS
1727  002476  031726                          DT2     ;$ERRPC $REG0   $REG1
1728  002500  000000                          0
1729
1730                                  ;ITEM   105
1731
1732  002502  031310                          EM130   ;'ERR BIT' DID NOT SET IN RKER
1733  002504  033144                          DH130   ;PC     RKCS    RKER    ERR BIT
1734  002506  031756                          DT26    ;$ERRPC $REG0   $REG1   $REG2
1735  002510  000000                          0
1736
1737
1738                                  ;ITEM   106
1739
1740  002512  031345                          EM131   ;HE OR ERR DID NOT SET
1741  002514  033203                          DH131   ;PC     RKCS    RKER
```

```
1742  002516  031726                        DT2     ;$ERRPC $REG0   $REG1
1743  002520  000000                        0
1744
1745                                ;ITEM   107
1746
1747  002522  031372                        EM132   ;RKER ERROR
1748  002524  032043                        DH4     ;PC     EXPCT   RECVD
1749  002526  031726                        DT2     ;$ERRPC $REG0   $REG1
1750  002530  000000                        0
1751
1752                                ;ITEM   110
1753
1754  002532  031404                        EM133   ;NXC BIT DID NOT SET
1755  002534  033231                        DH133   ;PC     RKCS    RKER    RKDA
1756  002536  031756                        DT26    ;PC     $REG0   $REG1   $REG2
1757  002540  000000                        0
1758
1759                                ;ITEM   111
1760
1761  002542  031427                        EM134   ;RK11 DIDN'T INTERRUPT ON SOFT ERROR
1762  002544  033203                        DH131   ;PC     RKCS    RKER
1763  002546  031726                        DT2     ;$ERRPC $REG0   $REG1
1764  002550  000000                        0
1765
1766                                ;ITEM   112
1767
1768  002552  031470                        EM135   ;MEX BITS INCREMENTED WRONG IN RKCS
1769  002554  032043                        DH4     ;PC     EXPCTD  RECVD
1770  002556  031726                        DT2     ;$ERRPC $REG0   $REG1
1771  002560  000000                        0
1772
1773                                ;ITEM   113
1774
1775  002562  030302                        EM110   ;CNTRL RDY DID NOT SET AFTER WRT CHK
1776  002564  032107                        DH14    ;PC     RKCS    RKER    RKWC
1777  002566  031756                        DT26    ;$ERRPC $REG0   $REG1   $REG2
1778  002570  000000                        0
1779
1780                                ;ITEM   114
1781
1782  002572  031525                        EM137   ;'WPS' NOT CLEAR
1783  002574  032323                        DH44    ;PC     RKCS    RKER    RKDS    RKDA
1784  002576  031736                        DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
1785  002600  000000                        0
1786
1787                                ;ITEM   115
1788
1789  002602  031543                        EM140   ;DATA ERROR ON TRANSFER FROM DISK TO TTY
1790  002604  033267                        DH140   ;PC     EXPCT   RECVD   RKBA    RKCS
1791  002606  031736                        DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
1792  002610  000000                        0
1793
1794
1795                                ;ITEM   116
1796
1797  002612  031612                        EM141   ;'DRIVE #' PRESENT, BUT NOT SPECIFIED
```

```
1798  002614  032726                    DH74    ;PC      DRIVE #
1799  002616  031720                    DT1     ;$ERRPC  $REGO
1800  002620  000000                    0
1801
1802                              ;ITEM  117
1803
1804  002622  025364                    EM11    ;RKWC ERROR
1805  002624  032043                    DH4     ;PC      EXPCT    RECVD
1806  002626  031726                    DT2     ;$ERRPC  $REGO    $REG1
1807  002630  000000                    0
1808                              ;ITEM  120
1809  002632  031656                    EM142
1810  002634  000000                    0
1811
1812
1813
```

```
1814  002636  000005                          START:  RESET                   ;CLEAR THE BUS
1815                                          ;;GIVE DRIVES TIME TO LOAD HEADS IN CASE OF AN APT START.
1816  002640  023737  000042  000046                  CMP     @#42,@#46       ;ARE WE IN ACT11 AUTOMATIC MODE?
1817  002646  001016                                  BNE     STARTA          ;NO, SKIP DELAY
1818  002650  005077  176464                          CLR     @RKDA           ;SELECT UNIT 0
1819  002654  012700  000250                          MOV     #250,R0         ;WAIT FOR..
1820  002660  032777  000200  176440  20$:           BIT     #200,@RKDS      ;DRIVE READY..
1821  002666  001006                                  BNE     STARTA          ;IN CASE..
1822  002670  005001                                  CLR     R1              ;OF APT..
1823  002672  005301                                  DEC     R1              ;START, BUT..
1824  002674  001376                                  BNE     .-2             ;DON'T WAIT..
1825  002676  005300                                  DEC     R0              ;FOREVER.
1826  002700  001367                                  BNE     20$
1827  002702  000000                                  HALT                    ;RKDS BIT 7 (DRIVE READY, NEVER SET
1828  002704                                  STARTA:
1829                                          .SBTTL  INITIALIZE THE COMMON TAGS
1830                                          ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1831  002704  012706  001100                          MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1832  002710  005026                                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1833  002712  022706  001140                          CMP     #SWR,R6 ;;DONE?
1834  002716  001374                                  BNE     .-6             ;;LOOP BACK IF NO
1835  002720  012706  001100                          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1836                                          ;;INITIALIZE A FEW VECTORS
1837  002724  012737  022134  000020                  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1838  002732  012737  000340  000022                  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1839  002740  012737  022406  000030                  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1840  002746  012737  000340  000032                  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1841  002754  012737  024672  000034                  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1842  002762  012737  000340  000036                  MOV     #340,@#TRAPVEC+2;LEVEL 7
1843  002770  012737  024772  000024                  MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1844  002776  012737  000340  000026                  MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1845  003004  005037  001206                          CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1846  003010  005037  001210                          CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1847  003014  112737  000001  001115                  MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1848  003022  012737  003022  001106                  MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1849  003030  012737  003030  001110                  MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1850                                          ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1851                                          ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1852  003036  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1853  003042  012737  003076  000004                  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1854  003050  012737  177570  001140                  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1855  003056  012737  177570  001142                  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1856  003064  022777  177777  176046                  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1857  003072  001012                                  BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1858                                                                          ;;AND  THE HARDWARE SWR IS NOT = -1
1859  003074  000403                                  BR      65$             ;;BRANCH IF NO TIMEOUT
1660  003076  012716  003104          64$:           MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1861  003102  000002                                  RTI
1862  003104  012737  000176  001140  65$:           MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1863  003112  012737  000174  001142                  MOV     #DISPREG,DISPLAY
1864  003120  012637  000004          66$:           MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1865
1866  003124  023737  000042  000046                  CMP     @#42,@#46       ;ARE WE IN ACT11 AUTOMATIC MODE?
1867  003132  001416                                  BEQ     69$             ;YES, SKIP TITLE
1868                                          .SBTTL  TYPE PROGRAM NAME
1869                                          ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
```

```
1870  003134  005227  177777                  INC     #-1             ;;FIRST TIME?
1871  003140  001043                          BNE     67$             ;;BRANCH IF NO
1872  003142  104401  003200                  TYPE    ,68$            ;;TYPE ASCIZ STRING
1873                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1874  003146  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1875  003152  001006                          BNE     69$             ;;BRANCH IF YES
1876  003154  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1877  003162  001005                          BNE     70$             ;;BRANCH IF NO
1878  003164  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
1879  003166  000403                          BR      70$
1880  003170  112737  000001  001134  69$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
1881  003176                        70$:
1882  003176  000424                          BR      67$             ;;GET OVER THE ASCIZ
1883                                ;;68$:    .ASCIZ  <CRLF>/RK11 LOGIC TEST 2/<15><12>/MAINDEC-11-DZRKK-E/<CRLF>
1884  003250                        67$:
1885  003250  012700  001410                  MOV     #DDPCH,R0
1886  003254  012701  177765                  MOV     #-13,R1
1887  003260  005020                1$:       CLR     (R0)+
1888  003262  005201                          INC     R1
1889  003264  001375                          BNE     1$
1890  003266  005227  177777                  INC     #-1             ;FIRST START ?
1891  003272  001020                          BNE     START1          ;BR IF NOT
1892  003274  013746  000004                  MOV     ERRVEC,-(SP)    ;SAVE ERROR VECTOR ADDRESS
1893  003300  012737  003314  000004          MOV     #2$,ERRVEC      ;NEW VECTOR ADDRESS
1894  003306  005737  177776                  TST     PS              ;SEE IF PROGRAM CAN REFERENCE THE
1895                                                                  ;PROCESSOR STATUS WORD
1896  003312  000406                          BR      3$              ;BR IF REFERENCE DIDN'T CAUSE TRAP
1897  003314  012737  000140  001400  2$:     MOV     #140,RKPRI      ;SETUP INTERRUPTING PRIORITY TO VALUE
1898                                                                  ;WHICH WILL ALLOW INTERRUPT ON AN LSI-11
1899  003322  012716  003330                  MOV     #3$,(SP)        ;SETUP RETURN ADDRESS
1900  003326  000002                          RTI                     ;RETURN
1901  003330  012637  000004          3$:     MOV     (SP)+,ERRVEC    ;RESTORE THE ERROR VECTOR
1902                                ;
1903                                ;FIND OUT IF ACT11, 'XXDP' CHAIN OR DUMP MODE
1904                                ;
1905  003334  012700  001410        START1: MOV     #DDPCH,R0
1906  003340  012701  177766                  MOV     #-12,R1         ;CLEAR OUT DRIVE TABLE AREA
1907  003344  005020                1$:       CLR     (R0)+
1908  003346  005201                          INC     R1
1909  003350  001375                          BNE     1$
1910  003352  122737  000002  000041          CMPB    #2,41           ;LOADED FROM AN RK05 ?
1911  003360  001166                          BNE     ST2             ;BR IF NOT
1912  003362  013737  000040  001410          MOV     40,DDPCH        ;GET DEVICE INDICATOR AND DRIVE ADDRESS OF
1913                                                                  ;LOADING RK05
1914  003370  122737  000010  001410          CMPB    #10,DDPCH       ;VALID DRIVE NUMBER IN BYTE 40 ?
1915  003376  101002                          BHI     2$              ;BR IF YES
1916  003400  105037  001410                  CLRB    DDPCH           ;MUST BE DRIVE ZERO WHICH LOADED
1917                                                                  ;THIS PROGRAM
1918  003404  005737  000042          2$:     TST     42              ;CHAIN MODE OR ACT11 AUTO ACCEPT ?
1919  003410  001432                          BEQ     4$              ;BR IF NEITHER
1920  003412  005737  001410                  TST     DDPCH           ;RUNNING FROM AN RK05 ?
1921  003416  001002                          BNE     3$              ;BR IF YES
1922  003420  000137  004262                  JMP     ST3             ;FIND OUT NUMBER OF DRIVES
1923  003424                          3$:
1924  003424  104401  003432                  TYPE    ,65$            ;;TYPE ASCIZ STRING
1925  003430  000413                          BR      64$             ;;GET OVER THE ASCIZ
```

```
1926
1927  003460
1928  003460  005046
1929  003462  113716  001410
1930  003466  104403
1931  003470     001
1932  003471     000
1933  003472  000137  004262
1934  003476  005227  177777
1935  003502  001115
1936  003504  104401  003512
1937  003510  000411
1938
1939  003534
1940  003534  005046
1941  003536  113716  001410
1942  003542  104403
1943  003544     001
1944  003545     000
1945  003546  104401  003554
1946  003552  000431
1947
1948  003636
1949  003636  104401  003644
1950  003642  000435
1951
1952  003736
1953
1954
1955
1956
1957
1958
1959  003736  012700  001412
1960  003742  012701  177767
1961  003746  005020
1962  003750  005201
1963  003752  001375
1964  003754  104401  003762
1965  003760  000415
1966
1967  004014
1968  004014  104411
1969  004016  012600
1970  004020  012701  177770
1971  004024  112002
1972  004026  042702  177400
1973  004032  012703  001414
1974  004036  012704  177770
1975  004042  012705  000060
1976  004046  020502
1977
1978  004050  001414
1979  004052  005205
1980  004054  005723
1981  004056  005204
```

```
;;65$:  .ASCIZ  <15><12>/NOT TESTING DRIVE /
64$:
        CLR     -(SP)           ;CLEAR WORD ON STACK
        MOVB    DDPCH,(SP)      ;GET DRIVE ADDRESS
        TYPOS                   ;TYPE THE ADDRESS
        .BYTE   1               ;ONLY 1 CHARACTER
        .BYTE   0               ;SUPRESS LEADING ZEROS
        JMP     ST3             ;GET NUMBER OF DRIVES
4$:     INC     #-1             ;FIRST TIME THROUGH HERE ?
        BNE     ST2             ;BR IF NOT
        TYPE    ,67$            ;;TYPE ASCIZ STRING
        BR      66$             ;;GET OVER THE ASCIZ
;;67$:  .ASCIZ  <15><12>/TO TEST DRIVE /
66$:
        CLR     -(SP)           ;CLEAR WORD ON THE STACK
        MOVB    DDPCH,(SP)      ;GET DRIVE ADDRESS
        TYPOS                   ;TYPE THE DRIVE ADDRESS
        .BYTE   1               ;ONLY 1 CHARACTER
        .BYTE   0               ;SUPRESS LEADING ZEROS
        TYPE    ,69$            ;;TYPE ASCIZ STRING
        BR      68$             ;;GET OVER THE ASCIZ
;;69$:  .ASCIZ  / HALT PROGRAM, REMOVE RKDP PACK AND REPLACE IT/<15><12>
68$:
        TYPE    ,71$            ;;TYPE ASCIZ STRING
        BR      70$             ;;GET OVER THE ASCIZ
;;71$:  .ASCIZ  /WITH A WORK PACK, CLEAR LOCATION 40, AND RESTART PROGRAM/
70$:

;FIND OUT FROM USER WHICH DRIVES (LOGICAL ADDRESSES) ARE TO BE
;TESTED (DRIVES TO BE TESTED ?). IN REPLY THE USER SHOULD TYPE IN THE
;LOGICAL ADDRESSES SEPERATED BY COMMAS. THUS IF 2 DRIVES 0,1 ARE PRESENT:
;       'DRIVS TO B TSTD?'
;       '0,1<CR>'           A CAR. RET. SHOULD BE TYPED TO TERMINATE THE LIST.
ST2:    MOV     #DRIVS,R0
        MOV     #-11,R1
13$:    CLR     (R0)+
        INC     R1
        BNE     13$
        TYPE    ,65$            ;;TYPE ASCIZ STRING
        BR      64$             ;;GET OVER THE ASCIZ
;;65$:  .ASCIZ  <15><12>/DRIVES TO BE TESTED ?/<15><12>
64$:
        RDLIN
        MOV     (SP)+,R0        ;GET STARTING ADRES OF ASCII STRING
        MOV     #-10,R1         ;SET UP COUNT
1$:     MOVB    (R0)+,R2        ;GET ASCII CHARACTER
        BIC     #177400,R2      ;MASK UNWANTED BITS
        MOV     #DRIV0,R3
        MOV     #-10,R4
        MOV     #60,R5
2$:     CMP     R5,R2           ;WAS THE TYPED IN CHARACTER
                                ;A NUMBER BETWEEN 0-7?
        BEQ     3$              ;YES, BRANCH
        INC     R5              ;NO, INCREMENT
        TST     (R3)+           ;INCREMENT POINTER TO DRV FLAG
        INC     R4              ;CHARACTER THAT WAS INPUT
```

```
1982  004060  001372                          BNE     2$              ;SHOULD BE 0-7, IF ANY OTHER
1983                                                                  ;TYPE '?' & AGAIN ASK FOR
1984                                                                  ;DRIVS TO BE TSTD?
1985  004062  005702                          TST     R2              ;IS IT A TERMINATOR?
1986  004064  001461                          BEQ     6$              ;YES, EXIT. NO DRIVES INDICATED.
1987  004066                          4$:
1988  004066  104401  004074                  TYPE    ,67$            ;;TYPE ASCIZ STRING
1989  004072  000402                          BR      66$             ;;GET OVER THE ASCIZ
1990                                  ;;67$:  .ASCIZ  /???/
1991  004100                          66$:
1992  004100  000716                          BR      ST2             ;GO, AGAIN ASK QUESTION
1993  004102  005713                  3$:     TST     @R3     ;SEE IF ALL READY SELECTED
1994  004104  001370                          BNE     4$      ;ERROR IF SELECTED  ALL READY
1995  004106  005213                          INC     @R3             ;SET UP FLAG FOR THE DRIVE
1996  004110  005237  001412                  INC     DRIVS           ;INCREMENT TOTAL NO OF DRIVES PRESENT
1997  004114  111002                  11$:    MOVB    @R0,R2          ;GET NEXT CHAR
1998  004116  042702  177400                  BIC     #177400,R2      ;CHARACTER ONLY
1999  004122  022702  000106                  CMP     #'F,R2          ;IS IT F?
2000  004126  001026                          BNE     8$              ;NO, GO ON
2001  004130  052713  100000                  BIS     #BIT15,@R3      ;SET BIT 15 TO SHOW RK05F
2002  004134  032705  000001                  BIT     #BIT0,R5        ;EVEN DRIVE?
2003  004140  001407                          BEQ     9$              ;EVEN DRIVE SO BRANCH
2004  004142  005763  177776                  TST     -2(R3)          ;CHECK EVEN DRIVE
2005  004146  001347                          BNE     4$              ;EVEN ALL READY SELECTED
2006  004150  012763  100001  177776          MOV     #BIT15!BIT0,-2(R3)      ;SELECT EVEN DRIVE
2007  004156  000406                          BR      10$             ;CONTINUE
2008  004160  005763  000002          9$:     TST     2(R3)           ;CHECK ODD DRIVE
2009  004164  001340                          BNE     4$              ;ERROR IF SELECTED BEFORE
2010  004166  012763  100001  000002          MOV     #BIT15!BIT0,2(R3)       ;SELECT ODD DRIVE
2011  004174  005237  001412          10$:    INC     DRIVS           ;COUNT DRIVES SELECTED
2012  004200  105720                          TSTB    (R0)+           ;POINT TO NEXT CHAR
2013  004202  000744                          BR      11$             ;CHECK FOR COMMA
2014  004204  022702  000054          8$:     CMP     #54,R2          ;IS IT A 'COMMA'?
2015  004210  001403                          BEQ     5$              ;YES, GO PROCESS NXT WORD
2016  004212  005702                          TST     R2              ;NO, IS IT A TERMINATOR?
2017  004214  001324                          BNE     4$              ;IF NOT, SOMETHING WRONG
2018                                                                  ;GO ASK QUESTION AGAIN
2019  004216  000404                          BR      6$              ;EXIT, IF A TERMINATOR
2020  004220  105720                  5$:     TSTB    (R0)+           ;INCREMENT PTR TO NXT BYTE
2021                                                                  ;IN INPUT BUFFER
2022  004222  005201                          INC     R1              ;THERE SHOULD BE NO MORE THAN
2023  004224  001277                          BNE     1$              ;8 DRIVES, HENCE IF MORE
2024  004226  000717                          BR      4$              ;THAN 8 DIFFERENT NOS. TYPED IN, ERROR!
2025                                                                  ;GO AGAIN ASK THE QUESTION
2026
2027  004230  005037  001440          6$:     CLR     SIZYET          ;NO SIZING NEEDED
2028  004234  032777  002000  174676          BIT     #SW10,@SWR      ;TESTING ON SIMULATOR?
2029  004242  001003                          BNE     7$              ;YES, BRANCH
2030  004244  005037  001344                  CLR     SIMUL           ;NO, CLR FLAG
2031  004250  000502                          BR      ST4
2032
2033  004252  012737  000001  001344  7$:     MOV     #1,SIMUL        ;SET FLAG TO INDICATE SIMULATOR
2034  004260  000476                          BR      ST4
2035
2036
2037
```

```
2038                                            ;CHECK NUMBER OF DRIVES
2039  004262  012737  177777  001440  ST3:      MOV     #-1,SIZYET      ;CHECK FOR RKDSF LATER
2040  004270  012737  004442  000004            MOV     #5$,@#4         ;SET UP ADRES FOR TIME-OUT VECTOR
2041  004276  005777  175024                    TST     @RKDS           ;REFERENCE RKDS
2042  004302  005777  175032                    TST     @RKDA           ;REFERENCE RKDA
2043  004306  012737  004534  000004            MOV     #BADTM0,@#4
2044  004314  104401                            TYPE
2045  004316  001216                            MSG1
2046  004320  012700  177770                    MOV     #-10,R0         ;INITIALIZE COUNT FOR THE 8 DRIVES
2047  004324  005037  001412                    CLR     DRIVS           ;INITIALIZE # OF DRIVES PRESENT TO 0
2048  004330  005001                            CLR     R1              ;INITIALIZE ADDRESS TO DRIVE 0
2049  004332  005004                            CLR     R4
2050  004334  012702  001414                    MOV     #DRIV0,R2
2051  004340  010177  174774          1$:       MOV     R1,@RKDA        ;ADDRESS THE DRIVE
2052  004344  020177  174770                    CMP     R1,@RKDA        ;CHECK, WAS IT ADDRESSED?
2053  004350  001405                            BEQ     3$              ;YES
2054  004352  012703  004356                    MOV     #2$,R3
2055  004356  004737  021022          2$:       JSR     PC,TYERM        ;WHILE CHECKING NUMBER OF DRIVE
2056                                            ;UNDER NON-MANUAL MODE :-
2057                                            ;RKDA HAD TO BE ADRESED BUT
2058                                            ;IT WAS FOUND THAT THE DRIVE NO
2059                                            ;THAT WAS WRITTEN COULD NOT BE READ BACK
2060                                            ;CORRECTLY.
2061
2062  004362  000413                            BR      4$
2063  004364  032777  000200  174734  3$:       BIT     #200,@RKDS      ;CHECK IF 'DRY' BIT IS SET, IF SET DRIVE IS
2064                                            ;PRESENT
2065  004372  001407                            BEQ     4$              ;
2066  004374  104401                            TYPE
2067  004376  001213                            $CRLF
2068  004400  005237  001412                    INC     DRIVS           ;IF PRESENT, INCREMENT # OF DRIVES
2069  004404  005212                            INC     (R2)            ;SET UP FLAG INDICATING THIS DRIVE PRESENT
2070  004406  010446                            MOV     R4,-(SP)
2071  004410  104402                            TYPOC
2072  004412  005722                    4$:     TST     (R2)+           ;SHIFT POINTER TO NXT DRIVE INDICATOR
2073  004414  062701  020000                    ADD     #20000,R1       ;SET UP ADDRESS FOR THE NEXT DRIVE
2074  004420  005204                            INC     R4              ;HAVE U CHECKED FOR ALL 8 DRIVES
2075  004422  005200                            INC     R0
2076  004424  001345                            BNE     1$
2077  004426  005737  001412                    TST     DRIVS
2078  004432  001011                            BNE     ST4
2079  004434  104401                            TYPE
2080  004436  001236                            MSG2
2081  004440  000406                            BR      ST4             ;GO CHECK THE DRIVE INDEPENDENT
2082                                            ;CONTROLLER LOGIC
2083  004442  011603                    5$:     MOV     (SP),R3         ;GET PC WHERE TIMEOUT OCCURED
2084  004444  022626                            CMP     (SP)+,(SP)+     ;RESTORE STACK
2085  004446  062703  177776                    ADD     #-2,R3
2086  004452  004737  021022                    JSR     PC,TYERM        ;GO TYPE ERROR MESSAGE
2087                                            ;WHILE CHECKING FOR THE NUMBER OF
2088                                            ;DRIVES IN NON-MANUAL MODE:-
2089                                            ;RKDS AND RKDA HAD TO BE REFERENCED, TIMEOUT
2090                                            ;OCCURED ON REFERENCING.PC IN THE ERROR
2091                                            ;MESSAGE INDICATES WHERE THE TIMEOUT OCCURED:
2092
2093                                    ;
```

```
2094
2095
2096  004456  005037  001434          ST4:    CLR     TS6FLG
2097  004462  005737  001412                  TST     DRIVS
2098  004466  001004                          BNE     1$
2099  004470  004737  021736                  JSR     PC,WATIME
2100  004474  000137  020646                  JMP     $EOP
2101  004500  012737  001414  001354  1$:     MOV     #DRIVD,DRVPTR
2102  004506  005037  001352                  CLR     DRVDON          ;INITIALIZE THE NO. OF DRIVES
                                                                      ;THAT HAVE BEEN CHECKED
2103
2104  004512  005037  001350                  CLR     DRIVAD          ;INITIALIZE DRIVE ADDRESS TO
                                                                      ;THE FIRST DRIVE
2105
2106  004516  012737  004534  000004          MOV     #BADTMO,@#4     ;SET TIME OUT VECTOR FOR UNEXPECTED
2107                                                                  ;TIME OUTS
2108  004524  012777  004600  174650          MOV     #BADINT,@RKVEC  ;SET UP RK11 INTERRUPT VECTOR FOR
2109                                                                  ;UNEXPECTED INTERRUPTS FROM RK11
2110  004532  000465                          BR      TST1            ;GO TO TEST 1
2111
2112
2113
2114
2115                                  ;THIS ROUTINE HANDLES UNEXPECTED TIME OUTS
2116
2117  004534  011600          BADTMO: MOV     (SP),R0 ;SAVE PC WHERE TIME OUT OCCURED
2118  004536  005740                  TST     -(R0)
2119  004540  022626                  CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
2120  004542  104401  004550          TYPE    ,65$            ;;TYPE ASCIZ STRING
2121  004546  000407                  BR      64$             ;;GET OVER THE ASCIZ
2122                          ;;65$:  .ASCIZ  <15><12>/TIMOUT,PC=/
2123  004566                  64$:
2124  004566  010046                  MOV     R0,-(SP)        ;SET UP FOR TYPING OUT PC
2125  004570  104402                  TYPOC                   ;GO TYPE OUT OCTAL PC
2126  004572  000000                  HALT
2127  004574  000137  002636          JMP     @#START
2128
2129
2130
2131                                  ;THIS ROUTINE HANDLES UNEXPECTED INTERRUPTS FROM RK11
2132                                  ;SW 9 AND 10 FOR LOOPING ON ERROR
2133                                  ;AND LOOPING ON TEST IN WHICH TIMEOUT
2134                                  ;OCCURRED, ARE PROVIDED.
2135
2136  004600  011600          BADINT: MOV     (SP),R0         ;SAVE PC WHERE INTERRUPT OCCURED
2137  004602  005740                  TST     -(R0)
2138  004604  032777  020000  174326  BIT     #20000,@SWR     ;INHIBIT ERROR TYPEOUT?
2139  004612  001014                  BNE     1$              ;YES, DON'T TYPE OUT
2140  004614  104401                  TYPE
2141  004616  001213                  $CRLF
2142  004620  104401                  TYPE
2143  004622  026000                  EM43                    ;TYPE 'UNEXPEXTED RK11 INTERRUPT'
2144                                                          ;TYPE ' AT PC='
2145  004624  104401  004632          TYPE    ,65$            ;;TYPE ASCIZ STRING
2146  004630  000403                  BR      64$             ;;GET OVER THE ASCIZ
2147                          ;;65$:  .ASCIZ  /,PC=/
2148  004640                  64$:
2149  004640  010046                  MOV     R0,-(SP)        ;SET UP FOR TYPING OUT PC
```

```
2150  004642  104402                          TYPOC                   ;GO TYPE OCTAL PC WHERE BAD
2151                                                                  ;INTERUPT OCCURED
2152  004644  032777  001000  174266  1$:     BIT     #1000,@SWR      ;LOOP ON ERROR?
2153  004652  001403                          BEQ     2$              ;NO, BRANCH
2154  004654  022626                          CMP     (SP)+,(SP)+     ;YES, REPOSITION STACK
2155  004656  000177  174224                  JMP     @$LPADR         ;GO TO THE STARTING ADDRESS OF
2156                                                                  ;THE TEST THAT GAVE UNEXPECTED INTERRUPT
2157  004662  032777  040000  174250  2$:     BIT     #40000,@SWR     ;LOOP ON TEST?
2158  004670  001401                          BEQ     3$              ;NO, BRANCH
2159  004672  000002                          RTI                     ;YES, LOOP. GO BACK WHER U INTERRUPTED FROM.
2160  004674  000000                  3$:     HALT                    ;UNEXPECTED INTERRUPT OCCURED AS
2161                                                                  ;INDICATED IN THE TYPE OUT.U CAN LOOP
2162                                                                  ;ON ERROR, TEST,OR INHIBIT TYPEOUT BY
2163                                                                  ;SETTING APPROPRIATE SWITCHES.
2164  004676  000137  002636                  JMP     @#START         ;GO BACK TO THE START OF THE
2165                                                                  ;PROGRAM. THUS PRESSING CONTINUE
2166                                                                  ;AFTER THE ABOVE HALT WILL
2167                                                                  ;RESTART THE PROGRAM
2168
2169
2170
2171                                  ;RESTART AFTER POWER FAIL
2172                                  ;THE PROGRAM WOULD RESTART HERE IF POWER CAME BACK AFTER A FALIURE.
2173
2174  004702  004727  021736          PFSTRT: JSR     PC,WATIME       ;KILL TIME
2175
2176
2177
2178                                  ;;*****************************************************************
2179                                  ;*TEST 1        CHECK THAT THE DRIVES THAT ARE NOT SPECIFIED ARE NOT FOUND TO BE PRESENT
2180                                          ;*THIS TEST CHECKS THAT THE DRIVES THAT ARE NOT SPECIFIED
2181                                          ;*(IN RESPONSE TO "DRIVS TO BE TSTD?") ARE NOT FOUND TO BE PRESENT.
2182                                          ;*EVERY DRIVE  FROM 0 TO 7 IS ADDRESSED. IF A PARTICULAR DRIVE
2183                                          ;*GIVES 'DRY' (IN RKDS), IT IS CHECKED THAT  THIS DRIVE
2184                                          ;*WAS SPECIFIED BY THE USER, IF IT WAS NOT AN ERROR IS
2185                                          ;*REPORTED, GIVING THE DRIVE NUMBER. IT IS LIKELY THAT THE USER
2186                                          ;*MAY HAVE FORGOTTEN TO PUT THE DRIVE (THAT IS NOT SPECIFIED) ON
2187                                          ;*'LOAD'. IF THIS IS THE CASE THEN PUT THIS DRIVE ON 'LOAD'.
2188                                          ;*IF THIS IS NOT THE CASE, THERE IS A GENUINE ERROR. (TWO DIFFERENT
2189                                          ;*DRIVE ADDRESSES MAY BE RESULTING IN THE SELECTION OF THE SAME
2190                                          ;*PHYSICAL DRIVE.)
2191                                  ;;*****************************************************************
2192  004706  000004                  TST1:   SCOPE
2193
2194  004710  012700  001414                  MOV     #DRIV0,R0       ;INITIALIZE POINTER
2195  004714  005001                          CLR     R1              ;INITIALIZE DRIVE ADRES 0
2196  004716  005002                          CLR     R2              ;INITIALIZE DRIVE # 0
2197  004720  005737  001410          1$:     TST     DDPCH           ;LOADED FROM AN RKOS ?
2198  004724  001403                          BEQ     2$              ;B IF NOT
2199  004726  120237  001410                  CMPB    R2,DDPCH        ;LOADED FROM THIS DRIVE ?
2200  004732  001435                          BEQ     4$              ;BR IF YES
2201  004734  010177  174400          2$:     MOV     R1,@RKDA        ;ADRES THE DRIVE
2202  004740  105777  174362                  TSTB    @RKDS           ;DRIVE READY?
2203  004744  100005                          BPL     3$              ;NO, THIS DRIVE NOT PRESENT
2204                                                                  ;YES, THIS DRIVE SELECTED
2205  004746  005710                          TST     @R0             ;WAS THIS DRIVE SPECIFIED BY
```

```
2206
2207  004750  001026                        BNE     4$              ;THE USER?
2208                                                                  ;YES, OK
2209                                                                  ;NO, THIS DRIVE # WAS NOT SPECIFIED
2210                                                                  ;BY THE USER, BUT STILL IS GIVING
2211  004752  010237  001162                MOV     R2,$REGO        ;'DRY' WHEN ADRESED. REPORT ERROR.
2212  004756  104116                        ERROR   116             ;GET DRIVE #
2213                                                                  ;THIS DRIVE # WAS NOT SPECIFIED BY
2214                                                                  ;THE USER, BUT WHEN ADRESED GAVE
2215                                                                  ;'DRY'. CHECK THAT THIS DRIVE # IF
2216                                                                  ;PHYSICALLY PRESENT IS ON 'LOAD'. IF
2217                                                                  ;THIS IS NOT THE CASE, THEN ONE DRIVE
2218                                                                  ;MAY BE GETTING SELECTED BY TWO DIFFERENT
2219  004760  005710                3$:     TST     @R0             ;LOGICAL ADDRESSES.
2220  004762  001421                        BEQ     4$              ;CHECK THAT THIS DRIVE WAS NOT INDICATED
2221                                                                  ;IF IT WAS, & IT IS NOT FOUND TO BE
2222  004764  004737  020770                JSR     PC,GT4RG        ;PRESENT (DRY CLEAR), REPORT ERROR.
2223  004770  104010                        ERROR   10              ;GET RKCS, ER, DS, DA
2224                                                                  ;DRIVE # (AS IN RKDA) WAS INDICATED BY
2225                                                                  ;THE USER, BUT WAS NOT FOUND TO BE PRESENT.
2226                                                                  ;CHECK THAT THE  ROTARY DRIVE SELECTION
2227                                                                  ;SWITCH ON THE MODULE IS SET TO THE RIGHT
2228                                                                  ;DRIVE #.
2229  004772  005010                        CLR     @R0             ;THIS DRIVE IS NOT FOUND TO BE PRESENT
2230                                                                  ;HENCE DROP IT FROM THE SELECTION TABLE.
2231  004774  010003                        MOV     R0,R3           ;DRIVE ADDR
2232  004776  162703  001414                SUB     #DRIV0,R3       ;MINUS OFFSET FOR TABLE
2233  005002  042703  000003                BIC     #3,R3           ;EVEN DRIVE OF PAIR
2234  005006  062703  001414                ADD     #DRIV0,R3       ;POINT TO EVEN OF PAIR IF RK05 F
2235  005012  042723  100000                BIC     #100000,(R3)+   ;NOT SPECIFIED AS F MODEL
2236  005016  042713  100000                BIC     #100000,(R3)    ;SAME
2237  005022  005337  001412                DEC     DRIVS           ;DECREMENT DRIVE COUNT
2238  005026  005202                4$:     INC     R2              ;INCRMNT DRIVE #
2239  005030  005720                        TST     (R0)+           ;INCRMNT POINTER
2240  005032  062701  020000                ADD     #20000,R1       ;INCRMNT ADRES TO NXT DRIVE
2241  005036  001330                        BNE     1$              ;LUP BAK IF NOT DONE
2242
2243
2244                                                                  ;THIS PART OF THE PROGRAM IS GOING TO BE REPEATED FOR
2245                                                                  ;EACH DRIVE PRESENT
2246                                                                  ;
2247                                                                  ;'DRIVAD' CONTAINS IN BITS 15,14,13 THE ADDRESS OF THE
2248                                                                  ;DRIVE BEING CURRENTLY CHECKED.
2249                                                                  ;
2250  00504C                        NUDRV:
2251
2252
2253                                                                  ;;*****************************************************************
2254                                                                  ;*TEST 2        FIND OUT NEXT DRIVE TO BE CHECKED
2255                                                                  ;THIS CODE FINDS OUT THE NEXT DRIVE THAT IS PRESENT AND THEN  SETS UP
2256                                                                  ;THE ADDDRESS IN DRIVAD (BITS 13,14,15). THUS THROUGHOUT THE FOLLOWING TESTS
2257                                                                  ;THE DRIVE TESTED IS THE DRIVE WHOOSE ADDRESS IS IN 'DRIVAD'.
2258                                                                  ;;*****************************************************************
2259  005040  000004                TST2:   SCOPE
2260  005042  012737  000001  001206        MOV     #1,$TIMES       ;;DO 1 ITERATION
2261  005050  012737  000002  001102        MOV     #2,$TSTNM       ;RESET POINTER TO THIS TEST
```

```
2262                                                                    ;NO. CHANGE THIS (2) IN CASE THE
2263                                                                    ;TEST NO. CHANGES
2264  005056  005037  001112                    CLR     $ERTTL        ;CLEAR TOTAL ERROR COUNT
2265  005062  005737  001412                    TST     DRIVS         ;R THERE ANY DRIVES PRESENT?
2266  005066  001002                            BNE     .+6           ;YES, BRANCH
2267  005070  000137  020646            4$:     JMP     $EOP          ;NO, JMP TO THE END
2268  005074  013701  001354                    MOV     DRVPTR,R1     ;GET THAT POINTER TO THE NEXT
2269                                                                    ;DRIVE FLAG
2270  005100  032721  000001            2$:     BIT     #BIT0,(R1)+   ;IS THIS DRIVE PRESENT?
2271  005104  001005                            BNE     1$            ;YES
2272  005106  062737  020000  001350    6$:     ADD     #20000,DRIVAD ;FORM NXT DRIVE ADRES
2273  005114  001371                            BNE     2$
2274  005116  000764                            BR      4$
2275  005120  005737  001410            1$:     TST     DDPCH         ;PROGRAM LOADED FROM AN RK05 ?
2276  005124  001413                            BEQ     3$            ;NO, BRANCH
2277  005126  013746  001350                    MOV     DRIVAD,-(SP)  ;PUT TEST DRIVE ADDRESS ON THE STACK
2278  005132  000316                            SWAB    (SP)          ;SETUP TO RIGHT JUSTIFY THE ADDRESS
2279  005134  006216                            ASR     (SP)          ;RIGHT JUSTIFY THE ADDRESS
2280  005136  006216                            ASR     (SP)          ;RIGHT JUSTIFY THE ADDRESS
2281  005140  006216                            ASR     (SP)          ;RIGHT JUSTIFY THE ADDRESS
2282  005142  006216                            ASR     (SP)          ;RIGHT JUSTIFY THE ADDRESS
2283  005144  006216                            ASR     (SP)          ;RIGHT JUSTIFY THE ADDRESS
2284  005146  122637  001410                    CMPB    (SP)+,DDPCH   ;PROGRAM LOADED FROM THIS DRIVE ?
2285  005152  001755                            BEQ     6$            ;BR IF YES, DON'T TEST THE DRIVE
2286  005154  010137  001354            3$:     MOV     R1,DRVPTR     ;STORE POINTER TO THE NEXT
2287                                                                    ;DRIVE FLAG
2288  005160  104401  001272                    TYPE    ,MSG4
2289  005164  013746  001350                    MOV     DRIVAD,-(R6)  ;GET THE DRIVE ADDRESS
2290  005170  004737  021174                    JSR     PC,SHFTRT     ;GO SHIFT IT TO THE RIGHT
2291  005174  005037  001404                    CLR     FFLAG
2292  005200  011600                            MOV     (R6),R0       ;DRIVE NUMBER
2293  005202  104403                            TYPOS                 ;GO TYPE THE OCTAL # FOR THE
2294                                                                    ;DRIVE THAT IS BEING CHECKED
2295  005204     001     000                    .BYTE   1,0
2296  005206  006300                            ASL     R0            ;INDEX TO TABLE
2297  005210  005760  001414                    TST     DRIV0(R0)     ;SEE IF F
2298  005214  100006                            BPL     5$            ;NO
2299  005216  104401  005224                    TYPE    ,65$          ;;TYPE ASCIZ STRING
2300  005222  000401                            BR      64$           ;;GET OVER THE ASCIZ
2301                                    ;;65$:  .ASCIZ  /F/
2302  005226                            64$:
2303  005226  005237  001404                    INC     FFLAG         ;SET F FLAG
2304  005232  104401                    5$:     TYPE
2305  005234  001213                            $CRLF                 ;TYPE CR, LF
2306                                    ;;******************************************************************
2307                                    ;*TEST 3        CHECK THAT DRIVE IS SUPPLIED WITH POWER-DPL BIT
2308                                    ;;******************************************************************
2309  005236  000004                    TST3:   SCOPE
2310  005240  104413                            CNT.RESET             ;GO, DO CONTROL RESET
2311                                                                    ;THIS IS A CALL FOR THE 'CNTRL-
2312                                                                    ;RESET' ROUTINE. A CONTROL RESET IS
2313                                                                    ;ISSUED AND AFTER A CERTAIN TIME
2314                                                                    ;IF THE 'CNTRL RDY' DOES NOT SET
2315                                                                    ;AN ERROR IS REPORTED. NOTE THAT
2316                                                                    ;THE PC IN ERROR MESSAGE IS THE
2317                                                                    ;PC WHERE 'CNT.RESET' IS LOCATED.
```

```
2318                                                                    ;THIS IS A VERY BASIC ERR& IF IT
2319                                                                    ;OCCURS GO BACK TO TEST 10
2320  005242  013700  001326                      MOV     RKDS,R0
2321  005246  013777  001350  174064              MOV     DRIVAD,@RKDA    ;ADDRESS THE DRIVE UNDER TEST
2322  005254  005710                              TST     @R0             ;CHECK IF ANY BIT OF RKDS IS SET?
2323  005256  001003                              BNE     1$              ;IF SET, BRANCH
2324  005260  011037  001162                      MOV     @R0,$REG0       ;GET RKDS
2325  005264  104004                              ERROR   4               ;RKDS ERROR! RKDS IF ADDRESSED
2326                                                                    ;CORRECTLY SHOULD BE NON-ZERO
2327  005266  012777  000015  174036  1$:         MOV     #15,@RKCS       ;ISSUE A DRV RESET, IF DRVE
2328                                                                    ;POWER IS LO, DPL WILL SET
2329  005274  005001                              CLR     R1
2330  005276  032710  010000          2$:         BIT     #10000,@R0      ;IS 'DPL' BIT SET?
2331  005302  001003                              BNE     3$              ;DPL IS SET, BRANCH
2332  005304  005201                              INC     R1              ;WAIT FOR SOME TIME TO
2333  005306  001373                              BNE     2$              ;SEE IF DPL WOULD SET
2334  005310  000403                              BR      4$-2            ;OK, DPL NOT SET
2335  005312  004737  020776          3$:         JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
2336  005316  104005                              ERROR   5               ;DPL BIT OF RKDS IS SET, CHECK DRIVE POWER
2337
2338
2339  005320  005001                              CLR     R1
2340  005322  032710  000100          4$:         BIT     #100,@R0        ;DID R/W/S RDY BIT SET?
2341  005326  001010                              BNE     TST4            ;;YES, EXIT
2342  005330  104417  000011                      DELAY   11              ;TIME DELAY
2343  005334  005201                              INC     R1              ;WAIT FOR R/W/S RDY
2344  005336  001371                              BNE     4$
2345  005340  017737  173762  001162              MOV     @RKDS,$REG0     ;GET RKDS
2346  005346  104016                              ERROR   16              ;R/W/S RDY DID NOT SET AFTER
2347                                                                    ;DRIVE RESET. DRIVE RESET WAS DONE
2348                                                                    ;TO CHECK 'DPL'BIT . THIS TEST
2349                                                                    ;IS NOT FOR CHECKING DRIVE RESET.
2350                                                                    ;U MIGHT WANT TO USE THE TEST PROVIDED
2351                                                                    ;FOR CHECKING DRIVE RESET.
2352
2353                                    ;;*****************************************************************
2354                                    ;*TEST 4        CHECK THAT 'DRIVE UNSAFE' IS CLEAR, 'HDEN' IS SET, 'WPS' IS CLEAR
2355                                    ;;*****************************************************************
2356  005350  000004                    TST4:   SCOPE
2357  005352  104413                            CNT.RESET                ;GO, DO CONTROL RESET
2358                                                                    ;THIS IS A CALL FOR THE 'CNTRL-
2359                                                                    ;RESET' ROUTINE. A CONTROL RESET IS
2360                                                                    ;ISSUED AND AFTER A CERTAIN TIME
2361                                                                    ;IF THE 'CNTRL RDY' DOES NOT SET
2362                                                                    ;AN ERROR IS REPORTED. NOTE THAT
2363                                                                    ;THE PC IN ERROR MESSAGE IS THE
2364                                                                    ;PC WHERE 'CNT.RESET' IS LOCATED.
2365                                                                    ;THIS IS A VERY BASIC ERR 3 IF IT
2366                                                                    ;OCCURS GO BACK TO TEST 10
2367  005354  013777  001350  173756            MOV     DRIVAD,@RKDA    ;SET DRIVE ADDRESS
2368  005362  017700  173740                    MOV     @RKDS,R0                ;GET RKDS
2369  005366  032700  002000                    BIT     #2000,R0        ;IS 'DRU' BIT OF RKDS SET?
2370  005372  001403                            BEQ     1$              ;NO
2371  005374  004737  020776                    JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
2372  005400  104006                            ERROR   6               ;'DRU' BIT OF RKDS IS SET, CHECK
2373                                                                    ;DRIV BY PUTTING RUN-LOAD SW TO LOAD
```

```
2374                                                            ;THEN BACK TO RUN
2375  005402  032700  004000          1$:     BIT     #4000,R0  ;IS 'HDEN' BIT SET?
2376  005406  001004                         BNE     2$        ;YES, BRANCH
2377  005410  017737  173712  001162         MOV     @RKDS,$REG0 ;GET RKDS
2378  005416  104007                         ERROR   7         ;ERROR, 'RKDS' BIT IS NOT SET
2379
2380  005420  032777  000040  173700  2$:     BIT     #40,@RKDS ;IS 'WPS' CLEAR?
2381  005426  001403                         BEQ     .+5       ;YES, EXIT
2382  005430  004737  020770                 JSR     PC,GT4RG  ;GET RKCS, ER, DS, DA
2383  005434  104114                         ERROR   114       ;'WPS'-WRITE PROTECT STATUS- BIT OF
                                                                ;OF RKDS SHOULD BE CLEAR, IF THIS DRIVE
                                                                ;IS WRITE ENABLED. CHECK & SEE IF THIS
                                                                ;DRIVE IS WRITE ENABLED, IF IT IS NOT,
                                                                ;WRITE ENABLE IT.
2384
2385
2386
2387
2388
2389                                                            ;
2390                                                    ;;*******************************************************
2391                                                    ;*TEST 5        CHECK THAT 'DRIVE READY' IS SET IN RKDS
2392                                                    ;;*******************************************************
2393  005436  000004                 TST5:   SCOPE
2394  005440  104413                         CNT.RESET         ;GO, DO CONTROL RESET
                                                                ;THIS IS A CALL FOR THE 'CNTRL-
                                                                ;RESET' ROUTINE. A CONTROL RESET IS
                                                                ;ISSUED AND AFTER A CERTAIN TIME
                                                                ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                ;AN ERROR IS REPORTED. NOTE THAT
                                                                ;THE PC IN ERROR MESSAGE IS THE
                                                                ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                ;THIS IS A VERY BASIC ERR & IF IT
                                                                ;OCCURS GO BACK TO TEST 10
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404  005442  013777  001350  173670          MOV     DRIVAD,@RKDA ;ADDRS THE DRIVE
2405  005450  105777  173652                  TSTB    @RKDS     ;IS 'DRY' SET?
2406  005454  100403                         BMI     TST6      ;YES, OK
2407  005456  004737  020770                 JSR     PC,GT4RG  ;GO, GET RKCS, ER, DS, DA
2408  005462  104010                         ERROR   10        ;'DRY' NOT SET
2409                                                          ;
2410                                                  ;;*******************************************************
2411                                                  ;*TEST 6        CHECK THAT 'SOK' BIT CAN SET
2412                                                          ;* THIS TEST CHECKS THAT WITHIN A CERTAIN TIME
2413                                                          ;* 'SOK' BIT CAN SET, IF IT DOES NOT AN ERROR IS REPORTED
2414                                                  ;;*******************************************************
2415  005464  000004                 TST6:   SCOPE
2416  005466  013777  001350  173644          MOV     DRIVAD,@RKDA ;ADDRS THE DRIVE
2417  005474  005001                         CLR     R1        ;INITIALIZE COUNT FOR TIMING WAIT LOOP
2418  005476  032777  000400  173622  1$:     BIT     #400,@RKDS ;IS SOK SET?
2419  005504  001006                         BNE     TST7      ;;EXIT
2420  005506  005201                         INC     R1        ;NO, WAIT
2421  005510  001372                         BNE     1$        ;WAITED LONG?
2422  005512  017737  173610  001162         MOV     @RKDS,$REG0 ;GET RKDS
2423  005520  104011                         ERROR   11        ;WAITED LONG BUT 'SEC OK' BIT DID NOT
2424                                                          ;SET
2425                                                          ;
2426
2427
2428                                                  ;;*******************************************************
2429                                                  ;*TEST 7        CHECK THAT 'SECTOR COUNTER' CAN COUNT FROM 0-13
```

```
                                                ;* THIS TEST CHECKS THAT THE SECTOR COUNTER CAN COUNT FROM
                                                ;* 0-13
                                                ;* 1) FIRST, FOR INITIALIZING PURPOSES THERE IS A TIMED LOOP
                                                ;* DURING WHICH SECTOR COUNTER SHOULD COUNT DOWN TO 0. IF THIS
                                                ;* IS NOT DONE AN ERROR IS REPORTED
                                                ;* 2) AFTER A COUNT OF 0 IS REACHED, THE PROGRAM WAITS
                                                ;* FOR A CERTAIN TIME, DURING WHICH THE SEC COUNTER
                                                ;* IS SAMPLED.  IF THE COUNTER DOES NOT CHANGE WITHIN THIS
                                                ;* TIME PERIOD AN ERROR IS REPORTED.
                                                ;* 3) UPON FINDING THAT THE COUNTER HAS CHANGED, IT IS CHECKED
                                                ;* IF IT INCREMENTED CORRECTLY.  IF IT DID NOT AN ERROR IS REPORTED
                                                ;* 4) IF IT INCREMENTED CORRECTLY, THE PROGRAM AGAIN WAITS IN A
                                                ;* LOOP TILL THE COUNTER CHANGES.  (STEPS 2,3,4 ARE REPEATED
                                                ;* TILL THE COUNTER COUNTS UP TO 13)
                                                ;;*****************************************************************
005522  000004                         TST7:   SCOPE
005524  104413                                 CNT.RESET                  ;GO, DO CONTROL RESET
                                                                           ;THIS IS A CALL FOR THE 'CNTRL-
                                                                           ;RESET' ROUTINE. A CONTROL RESET IS
                                                                           ;ISSUED AND AFTER A CERTAIN TIME
                                                                           ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                           ;AN ERROR IS REPORTED. NOTE THAT
                                                                           ;THE PC IN ERROR MESSAGE IS THE
                                                                           ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                           ;THIS IS A VERY BASIC ERR & IF IT
                                                                           ;OCCURS GO BACK TO TEST 10
005526  013777  001350  173604                 MOV     DRIVAD,@RKDA
005534  013700  001326                         MOV     RKDS,R0            ;INITIALIZE
005540  005037  001356                         CLR     INDX1              ;'COUNT' - TO TIME 'ERROR 35'
005544  005005                                 CLR     R5                 ;INITIALIZE 'COUNT' - TO TIME
                                                                           ;'ERROR 36' (WAIT LOOP)
005546  012704  177764                         MOV     #-14,R4            ;INITIALIZE 'COUNT' - FOR THE 12 SECTORS.
005552  012703  000001                         MOV     #1,R3              ;R3 CONTAINS THE 'NEXT' COUNT OF SEC-CNTR
                                                                           ;R1 CONTAINS THE 'PREVIOUS' COUNT OF SEC-CNTR
                                                                           ;R2 CONTAINS THE 'PRESENT' COUNT OF SEC-CNTR
005556  005037  001360                 1$:     CLR     INDX2              ;INITIALIZE 'COUNT' - TO TIME
                                                                           ;(WAIT LOOP) 'ERROR 34'
005562  005237  001356                         INC     INDX1              ;KEEP TIMING FOR 'ERROR 35'
005566  001440                                 BEQ     6$                 ;BRANCH & REPORT ERROR IF WAITED LONG?
005570  005237  001360                 2$:     INC     INDX2              ;KEEP TIMING FOR 'ERROR 34'
005574  001441                                 BEQ     7$                 ;BRANCH & REPORT ERROR IF WAITED LONG?

005576  011001                                 MOV     @R0,R1             ;GET RKDS
005600  032701  000400                         BIT     #400,R1            ;IS 'SOK' SET?
005604  001771                                 BEQ     2$                 ;NO, WAIT FOR IT TO SET
005606  021001                                 CMP     @R0,R1             ;MAKE SURE THAT 2 CONSECUTIVE
005610  001362                                 BNE     1$                 ;READINGS OF SEC-CNTR ARE SAME
005612  042701  177760                         BIC     #177760,R1         ;YES, MASK OUT NON-SEC CNTR BITS
005616  001357                                 BNE     1$                 ;IS IT SECTOR 0, IF NOT LOOP BACK &
                                                                           ;WAIT FOR SECTOR 0
005620  005204                 3$:     INC     R4                 ;KEEP TRACK OF SECTORS CHECKED
005622  001447                         BEQ     TST10              ;;EXIT IF ALL SECTORS CHKD
005624  005205                 4$:     INC     R5                 ;KEEP TIMING FOR 'ERROR 36'
005626  001431                         BEQ     8$                 ;BR & REPORT ERROR IF WAITED LONG
005630  011002                         MOV     @R0,R2             ;GET RKDS
005632  032702  000400                 BIT     #400,R2            ;IS SOK SET?
```

```
2486  005636  001772                        BEQ     4$              ;NO, WAIT FOR SOK
2487  005640  021002                        CMP     @R0,R2          ;MAKE SURE THAT 2 CONSECUTIVE
2488  005642  001370                        BNE     4$              ;READINGS OF SEC-CNTR ARE SAME
2489  005644  042702  177760                BIC     #177760,R2      ;MASK NON-SEC-CNTR BITS
2490  005650  020201                        CMP     R2,R1           ;HAS SEC CNTR INCREMENTED?
2491  005652  001764                        BEQ     4$              ;NO, WAIT FOR IT TO CHANGE
2492  005654  020203                        CMP     R2,R3           ;YES, DID IT INCREMENT CORRECTLY?
2493  005656  001023                        BNE     9$              ;NO - REPORT ERROR
2494
2495  005660  005203                5$:     INC     R3              ;INCREMENT "NEXT COUNT"
2496  005662  005201                        INC     R1              ;INCREMENT "PREVIOUS COUNT"
2497  005664  005005                        CLR     R5              ;INITIALIZE AGAIN FOR TIMING 'ERROR 36'
2498  005666  000754                        BR      3$              ;GO & CHECK THE NEXT SECTOR COUNT
2499
2500  005670  010137  001162        6$:     MOV     R1,$REG0        ;GET 'SEC CNTR'
2501  005674  104012                        ERROR   12              ;WAITED LONG, BUT SECTOR COUNTER
2502                                                                ;DID NOT COUNT TO 0
2503  005676  000421                        BR      TST10           ;;EXIT
2504
2505  005700  017737  173422  001162  7$:     MOV     @RKDS,$REG0     ;GET RKDS
2506  005706  104011                        ERROR   11              ;WAITED LONG, BUT 'SOK' BIT DID
2507                                                                ;NOT SET
2508  005710  000414                        BR      TST10           ;;EXIT
2509
2510  005712  010237  001162        8$:     MOV     R2,$REG0        ;GET SEC CNTR (PRESENT COUNT)
2511  005716  010337  001164                MOV     R3,$REG1        ;GET "NEXT COUNT"
2512  005722  104013                        ERROR   13              ;WAITED LONG, BUT THE SECTOR
2513                                                                ;COUNTER DID NOT INCREMENT FROM
2514                                                                ;THE PRESENT COUNT TO THE NEXT COUNT
2515  005724  000406                        BR      TST10           ;;EXIT
2516
2517  005726  010337  001162        9$:     MOV     R3,$REG0        ;GET 'NEXT COUNT' (SEC CNTR SHOULD BE THIS)
2518  005732  010237  001164                MOV     R2,$REG1        ;GET PRESENT COUNT (WHAT SEC CNTR WAS)
2519  005736  104014                        ERROR   14              ;SEC CNTR INCREMENTED WRONG, DID
2520                                                                ;NOT INCREMENT FROM PRESENT COUNT
2521                                                                ;TO NEXT COUNT
2522  005740  000747                        BR      5$
2523                                          ;
2524
2525                                  ;;**********************************************************************
2526                                  ;*TEST 10       CHECK THAT SC=SA CAN BE GENERATED
2527                                          ;* THIS TEST CHECKS THAT SC=SA CAN BE GENERATED FOR
2528                                          ;* EVERY SECTOR
2529                                  ;;**********************************************************************
2530  005742  000004                TST10:  SCOPE
2531  005744  104413                        CNT.RESET               ;GO, DO CONTROL RESET
2532                                                                ;THIS IS A CALL FOR THE 'CNTRL-
2533                                                                ;RESET' ROUTINE. A CONTROL RESET IS
2534                                                                ;ISSUED AND AFTER A CERTAIN TIME
2535                                                                ;IF THE 'CNTRL RDY' DOES NOT SET
2536                                                                ;AN ERROR IS REPORTED. NOTE THAT
2537                                                                ;THE PC IN ERROR MESSAGE IS THE
2538                                                                ;PC WHERE 'CNT.RESET' IS LOCATED.
2539                                                                ;THIS IS A VERY BASIC ERR 3 IF IT
2540                                                                ;OCCURS GO BACK TO TEST 10
2541  005746  013704  001350                MOV     DRIVAD,R4
```

```
2542  005752  013700  001326          MOV     RKDS,R0
2543  005756  012703  177764          MOV     #-14,R3         ;INITIALIZE COUNT FOR # OF SECTORS
2544  005762  010477  173352    1$:   MOV     R4,@RKDA        ;ADDRESS THE DRIVE
2545  005766  005005                  CLR     R5              ;INITIALIZE COUNT - FOR TIMING ERROR
2546  005770  005205                2$:   INC     R5              ;KEEP TIMING FOR ERROR
2547  005772  001410                  BEQ     3$              ;REPORT ERROR IF WAITED LONG
2548  005774  011001                  MOV     @R0,R1          ;GET RKDS
2549  005776  032701  000020          BIT     #20,R1          ;IS SC=SA SET?
2550  006002  001772                  BEQ     2$              ;NO, WAIT FOR IT
2551  006004  005204                4$:   INC     R4              ;ADDRS THE NEXT SECTOR
2552  006006  005203                  INC     R3              ;ARE ALL SECTORS CHECKED FOR SC=SA
2553  006010  001364                  BNE     1$              ;NO, GO & CHECK NEXT
2554  006012  000406                  BR      TST11           ;;YES, EXIT
2555
2556  006014  110437  001162    3$:   MOVB    R4,$REG0         GET SECTOR ADDRESS
2557  006020  010137  001164          MOV     R1,$REG1         GET RKDS
2558  006024  104015                  ERROR   15              ;COULD NOT GET SC=SA FOR THIS
2559                                                          ;'SECTOR ADDRESS'
2560  006026  000766                  BR      4$              ;GO CHK FOR THE REST
2561                                   ;
2562                            ;;****************************************************************
2563                            ;*TEST 11        CHECK THAT 'R/W/S RDY' IS SET & 'SIN' IS CLEAR
2564                            ;;****************************************************************
2565  006030  000004          TST11:  SCOPE
2566  006032  104413                  CNT.RESET               ;GO, DO CONTROL RESET
2567  006034  013777  001350  173276  MOV     DRIVAD,@RkDA    ;ADDRESS THE DRIVE
2568  006042  005001                  CLR     R1
2569  006044  017700  173256    1$:   MOV     @RKDS,R0        ;GET RKDS
2570  006050  032700  000100          BIT     #100,R0         ;IS R/W/S RDY SET?
2571  006054  001005                  BNE     2$              ;YES, BRANCH
2572  006056  005201                  INC     R1              ;WAITED LONG ENOUGH?
2573  006060  001371                  BNE     1$              ;IF NOT LUP BAK & WAIT
2574  006062  010037  001162          MOV     R0,$REG0        ;GET RKDS
2575  006066  104016                  ERROR   16              ;R/W/S RDY SHOULD BE SET
2576  006070  032700  001000    2$:   BIT     #1000,R0        ;IS SIN CLEAR?
2577  006074  001403                  BEQ     TST12           ;;YES, EXIT
2578  006076  004737  020770          JSR     PC,GT4RG        ;GET RKCS,ER,DS,DA
2579  006102  104001                  ERROR   1               ;'SIN' SHOULD HAVE BEEN CLEAR
2580                                                          ;IT WAS NOT CLEAR
2581                                                          ;NEXT TEST IS GOING TO CHECK
2582                                                          ;DRIVE RESET, SIN SHOULD BE
2583                                                          ;CLEARED THEN. IT WILL BE CHECKED
2584                                                          ;THERE.
2585
2586                            ;;****************************************************************
2587                            ;*TEST 12        CHECK 'DRIVE RESET'
2588                                    ;*THIS TEST CHECKS THE VERY BASIC DRIVE RESET LOGIC.
2589                                    ;*SINCE THE HEADS ARE AT CYLINDER 0 (GOING INTO THIS
2590                                    ;*TEST) DRIVE RESET RETRACTS THEM BACK BEYOND CYLINDER 0,
2591                                    ;*AFTER WHICH THEY ARE PUSHED FORWARD TO CYLINDER 0 AGAIN.
2592                                    ;*IN THE LATER PART OF THIS PROGRAM THERE IS A DRIVE RESET
2593                                    ;*TEST WHICH DOES THE RESET FROM LAST CYLINDER.
2594                            ;;****************************************************************
2595  006104  000004          TST12:  SCOPE
2596  006106  104413                  CNT.RESET               ;GO, DO CONTROL RESET
2597                                                          ;THIS IS A CALL FOR THE 'CNTRL-
```

```
2598                                                                    ;RESET' ROUTINE. A CONTROL RESET IS
2599                                                                    ;ISSUED AND AFTER A CERTAIN TIME
2600                                                                    ;IF THE 'CNTRL RDY' DOES NOT SET
2601                                                                    ;AN ERROR IS REPORTED. NOTE THAT
2602                                                                    ;THE PC IN ERROR MESSAGE IS THE
2603                                                                    ;PC WHERE 'CNT.RESET' IS LOCATED.
2604                                                                    ;THIS IS A VERY BASIC ERR & IF IT
2605                                                                    ;OCCURS GO BACK TO TEST 10
2606  006110  013700  001332                MOV     RKCS,R0
2607  006114  005004                        CLR     R4                 ;INITIALIZ COUNT - TO TIME ERROR
2608  006116  013777  001350  173214        MOV     DRIVAD,@RKDA       ;ADDRESS THE DRIVE
2609  006124  012710  000015                MOV     #15,@R0            ;'DRIVE RESET', GO
2610  006130  104412                        CHKCRDY                    ;GO CHECK IF CONTROL RDY IS SET
2611                                                                    ;IF SO, SKIP THE EROR MESSAGE.
2612  006132  104021                        ERROR   21                 ;CNTRL RDY DID NOT SET AFTER
2613                                                                    ;SENDING CYL ADDR TO THE DRIV.
2614                                                                    ;'ADD ACK' SHOULD HAVE COME BACK
2615                                                                    ;FROM DRIVE, THEREUPON SETTING 'CN RDY'
2616  006134  012705  177776        2$:     MOV     #-2,R5             ;SET UP DELAY COUNTER
2617  006140  032777  000100  173160  6$:   BIT     #100,@RKDS         ;CHECK FOR R/W/S READY
2618  006146  001402                        BEQ     .+6
2619  006150  000137  006172                JMP     3$
2620  006154  005204                        INC     R4
2621  006156  001370                        BNE     6$
2622  006160  005205                        INC     R5
2623  006162  001366                        BNE     6$
2624  006154  004737  020770                JSR     PC,GT4RG           ;GO, GET RKCS, ER, DS, DA
2625  006170  104026                        ERROR   26                 ;R/W/S RDY DID NOT SET AFTER
2626                                                                    ;DRIVE RESET
2627
2628  006172  032777  001000  173126  3$:   BIT     #1000,@RKDS        ;DID SIN SET?
2629  006200  001403                        BEQ     5$                 ;NO, BRANCH
2630  006202  004737  020770                JSR     PC,GT4RG           ;GO, GET RKCS,ER,DS,DA
2631  006206  104001                        ERROR   1                  ;SIN SET, AFTER A
2632                                                                    ;DRIVE RESET.
2633  006210  032710  140000        5$:     BIT     #140000,@R0        ;WAS 'ERR' BIT OR 'HE' BIT SET?
2634  006214  001403                        BEQ     4$                 ;NO
2635  006216  004737  020770                JSR     PC,GT4RG           ;GO, GET RKCS, ER, DS, DA
2636  006222  104022                        ERROR   22                 ;'ERR' OR 'HE' BIT SET WHILE DOING
2637                                                                    ;DRIVE RESET
2638  006224  022710  000214        4$:     CMP     #214,@R0           ;DOES RKCS STILL CONTAIN THE
2639                                                                    ;'DRIV RES' BITS
2640  006230  001406                        BEQ     TST13              ;YES, EXIT
2641  006232  012737  000214  001162        MOV     #214,$REG0         ;GET EXPCTD RKCS
2642  006240  011037  001164                MOV     @R0,$REG1          ;GET RKCS, RECVD
2643  006244  104024                        ERROR   24                 ;NO - RKCS SHOULD CONTAIN THE 'DRIV RES'
2644                                                                    ;FUNCTION, ERROR IF DIFFERENT.
2645
2646                                ;;*****************************************************************
2647                                ;*TEST 13       CHECK 'SEEK' TO CYLINDER 0
2648                                        ;* THIS TEST CHECKS THE SEEK LOGIC DOING SEEK TO CYLINDER 0.
2649                                        ;* NOTE THAT SINCE THE HEADS ARE ALREADY ON CYLINDER 0, NO
2650                                        ;* HEAD MOVEMENT IS INVOLVEDN AND THE STRESS IS ON THE BASIC SEEk
2651                                        ;* LOGIC.
2652                                ;;*****************************************************************
2653  006246  000004                TST13:  SCOPE
```

```
2654  006250  104413                          CNT.RESET                ;GO, DO CONTROL RESET
2655                                                                   ;THIS IS A CALL FOR THE 'CNTRL-
2656                                                                   ;RESET' ROUTINE. A CONTROL RESET IS
2657                                                                   ;ISSUED AND AFTER A CERTAIN TIME
2658                                                                   ;IF THE 'CNTRL RDY' DOES NOT SET
2659                                                                   ;AN ERROR IS REPORTED. NOTE THAT
2660                                                                   ;THE PC IN ERROR MESSAGE IS THE
2661                                                                   ;PC WHERE 'CNT.RESET' IS LOCATED.
2662                                                                   ;THIS IS A VERY BASIC ERR & IF IT
2663                                                                   ;OCCURS GO BACK TO TEST 10
2664  006252  104421                          TST.SIN                  ;GO CHECK IF SIN SET. IF SET
2665                                                                   ;A DO DRIVE RESET TO CLEAR IT
2666  006254  013700  001332                  MOV     RKCS,R0
2667  006260  013777  001350  173052          MOV     DRIVAD,@RKDA     ;ADDRESS THE DRIVE
2668
2669  006266  012710  000011                  MOV     #11,@R0          ;'SEEK' GO
2670  006272  104412                          CHKCRDY                  ;GO CHECK IF CONTROL RDY IS SET
2671                                                                   ;IF SO, SKIP THE EROR MESSAGE.
2672  006274  104021                          ERROR   21               ;'CNTRL RDY' DID NOT SET AFTER SENDING
2673                                                                   ;CYL ADDR TO THE DRIVE, 'ADD ACK'
2674                                                                   ;SHOULD HAVE COME BACK FROM THE
2675                                                                   ;DRIVE, THEREUPON SETTING 'CNTRL RDY'
2676  006276  005005                  2$:     CLR     R5
2677  006300  032777  000100  173020          BIT     #100,@RKDS       ;DID R/W/S RDY BIT SET?
2678  006306  001005                          BNE     3$               ;YES, BRANCH
2679  006310  005205                          INC     R5               ;WAITED LONG ENOUGH?
2680  006312  001372                          BNE     2$+2             ;IF NOT, LUP BAK & WAIT
2681  006314  004737  020770                  JSR     PC,GT4RG         ;GO, GET RKCS, ER, DS, DA
2682  006320  104026                          ERROR   26               ;R/W/S RDY DID NOT SET AFTER SEEK
2683  006322  032777  001000  172776  3$:     BIT     #1000,@RKDS      ;DID SIN SET?
2684  006330  001403                          BEQ     6$               ;NO, BRANCH
2685  006332  004737  020770                  JSR     PC,GT4RG         ;GO, GET RKCS,ER,DS,DA
2686  006336  104001                          ERROR   1                ;SIN SET ON DOING SEEK
2687                                                                   ;TO CYL 0 NOTE THIS IS THE
2688                                                                   ;FIRST TIME THE HEADS HAVE
2689                                                                   ;BEEN MOVED
2690
2691  006340  032710  140000          6$:     BIT     #140000,@R0      ;WAS 'ERR' OR 'HE' BIT SET?
2692  006344  001403                          BEQ     4$
2693
2694  006346  004737  020770                  JSR     PC,GT4RG         ;GO, GET RKCS, ER, DS, DA
2695  006352  104022                          ERROR   22               ;'ERR' OR 'HE' BIT SET WHILE DOING 'SEEK'
2696
2697  006354  005777  172750          4$:     TST     @RKER            ;WAS ANY BIT IN RKER SET?
2698  006360  001403                          BEQ     5$               ;NO
2699  006362  004737  020776                  JSR     PC,GT3RG         ;GO, GET RKCS, ER, DS
2700  006366  104023                          ERROR   23               ;RKER SHOWS AN ERROR BIT, CHECK
2701
2702  006370  022710  000210          5$:     CMP     #210,@R0         ;DOES RKCS STILL CONTAIN 'SEEK' FUNCTION
2703  006374  001406                          BEQ     TST14            ;;YES, EXIT
2704  006376  012737  000210  001162          MOV     #210,$REG0       ;GET EXPCTD RKCS
2705  006404  011037  001164                  MOV     @R0,$REG1        ;GET RKCS RECVD
2706  006410  104024                          ERROR   24               ;NO, RKCS SHOULD BE STILL CONTAINING
2707                                                                   ;'SEEK' FUNCTION ERROR - IF IT CHANGED
2708                                           ;
2709                                   ;;*****************************************************************
```

```
                                        T14     CHECK R/W/S RDY IS CLEAR WHEN HEADS ARE IN MOTION

                                        ;*TEST 14       CHECK R/W/S RDY IS CLEAR WHEN HEADS ARE IN MOTION
                                                ;*THIS TEST CHECKS THAT R/W/S DOES GET CLEARED
                                                ;*WHEN THE HEADS ARE IN MOTION. SINCE 'MOVE L' ON
                                                ;*M7700 (RKOS) GENERATES THIS SIGNAL, ABSENCE OF
                                                ;*R/W/S RDY-CLEAR COULD MEAN A FAULT ON M7702
                                                ;*WHERE 'MOVE L' IS GENERATED.
                                                ;*NOTE THIS IS THE FIRST TIME HEADS ARE MADE TO MOVE BY SEEKING
                                                ;*TO CYLINDER 2.
                                        ;;*****************************************************************
006412  000004                        TST14:  SCOPE
006414  104413                                CNT.RESET                       ;GO, DO CONTROL RESET
                                                                            ;THIS IS A CALL FOR THE 'CNTRL-
                                                                            ;RESET' ROUTINE. A CONTROL RESET IS
                                                                            ;ISSUED AND AFTER A CERTAIN TIME
                                                                            ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                            ;AN ERROR IS REPORTED. NOTE THAT
                                                                            ;THE PC IN ERROR MESSAGE IS THE
                                                                            ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                            ;THIS IS A VERY BASIC ERR & IF IT
                                                                            ;OCCURS GO BACK TO TEST 10
006416  104421                                TST.SIN                         ;GO CHECK IF SIN IS SET
                                                                            ;IF SET DO DRV-RESET TO CLR IT
006420  004737  021500                        JSR     PC,DRESET               ;MAKE SURE HEADS R ON CYL 0
006424  104026                                ERROR   26                      ;R/W/S RDY DIDN'T SET
                                                                            ;AFTER THE ABOVE DRV RESET
006426  005005                                CLR     R5
006430  013777  001350  172702                MOV     DRIVAD,@RKDA
006436  052777  000100  172674                BIS     #100,@RKDA              ;SEEK CYLINDER 2
006444  013701  001326                        MOV     RKDS,R1
006450  012777  000011  172654                MOV     #11,@RKCS               ;SEEK, GO
006456  032711  000100              1$:       BIT     #100,@R1                ;DID R/W/S RDY CLR?
006462  001405                                BEQ     2$                      ;YES, BRANCH
006464  005205                                INC     R5
006466  100373                                BPL     1$
006470  004737  020776                        JSR     PC,GT3RG
006474  104025                                ERROR   25                      ;R/W/S RDY WAS NOT CLEAR WHEN HEADS
                                                                            ;WERE SEEKING TO CYLINDER 2

006476  004737  021432              2$:       JSR     PC,TSTRWS               ;GO, WAIT FOR R/W/S RDY TO SET
006502  104016                                ERROR   16                      ;R/W/S RDY DID NOT SET AFTER SEEK
                                                                            ;WAS TRIED TO CYLINDER 2 (ABOVE).
                                                                            ;NOTE THIS WAS THE FIRST TIME A SEEK
                                                                            ;WAS TRIED TO A CYLINDER OTHER THAN
                                                                            ;0.


                                        ;;*****************************************************************
                                        ;*TEST 15       CHECK 'WRITE' FORMAT FUNCTION-CYLINDER 0, SECTOR 0
                                                ;*THIS TEST CHECKS THE LOGIC INVOLVED IN THE WRITE FMT
                                                ;*FUNCTION. ON ISSUING A WRT FMT, THE FOLLOWING IS CHECKED
                                                ;*1)  CNTRL RDY WAS CLEARED AS GO WAS SET.
                                                ;*2)  CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION OF FUNCTION
                                                ;*3)  IF 'HE' OR 'ERR' BIT SET?
                                                ;*4)  IF RKDA INCREMENTED CORRECTLY FROM 0 TO 1?
```

```
                                T15     CHECK 'WRITE' FORMAT FUNCTION-CYLINDER 0, SECTOR 0

                                        ;*5)  IF RKWC OVERFLOWED CORRECTLY TO 0?
                                        ;*6)  IF RKBA INCREMENTED CORRECTLY BY 2?
                                        ;*7)  IF ANY BIT IN RKER SET?
                                        ;*8)  IF THE 'WRT FMT' FUNCTION BITS ARE STILL IN THE RKCS?
                                        ;*NOTE THAT ONE WORD '125252' WAS WRITTEN ON SECTOR
                                        ;*0 & IT WILL BE CHECKED IN THE NEXT TESTS.
                                ;;*******************************************************
006504  000004                 TST15:  SCOPE
006506  104413                         CNT.RESET                      ;GO, DO CONTROL RESET
                                                                       ;THIS IS A CALL FOR THE 'CNTRL-
                                                                       ;RESET' ROUTINE. A CONTROL RESET IS
                                                                       ;ISSUED AND AFTER A CERTAIN TIME
                                                                       ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                       ;AN ERROR IS REPORTED. NOTE THAT
                                                                       ;THE PC IN ERROR MESSAGE IS THE
                                                                       ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                       ;THIS IS A VERY BASIC ERR & IF IT
                                                                       ;OCCURS GO BACK TO TEST 10
006510  104421                         TST.SIN                        ;GO CHECK IF SIN IS SET
                                                                       ;IF SET, DO DRIVE RESET TO CLR IT
006512  012703  033336                 MOV     #OUTBUF,R3
                                                                       ;THIS CODE SETS UP A 256 WORD BUFFER
                                                                       ;WHICH WILL BE USED TO WRITE 1 SECTOR
                                                                       ;ON THE DISK
                                                                       ;1ST WORD       000001
                                                                       ;2ND WORD       177777  2'S COMPLEMENT
                                                                       ;3RD WORD       000002  OF ABOVE
                                                                       ;4TH WORD       177776
                                                                       ;
                                                                       ;253RD WORD     000177
                                                                       ;254TH WORD     177601
                                                                       ;255TH WORD     000000
                                                                       ;256TH WORD     125252
006516  012700  000001                 MOV     #1,R0                  ;SET COUNT

006522  010023                 9$:     MOV     R0,(R3)+               ;SET UP DATA WORDS
006524  010013                         MOV     R0,(R3)
006526  005423                         NEG     (R3)+
006530  005200                         INC     R0
006532  022700  000200                 CMP     #200,R0                ;DONE?
006536  001371                         BNE     9$
006540  005023                         CLR     (R3)+                  ;SET 255TH WORD TO 0
006542  012713  125252                 MOV     #125252,@R3            ;SET 256TH WORD

006546  012703  033336                 MOV     #OUTBUF,R3             ;RESET POINTER TO OUTBUF
006552  013701  001332                 MOV     RKCS,R1
006556  013702  001336                 MOV     RKBA,R2
006562  010312                         MOV     R3,@R2                 ;FROM HERE-SET UP CURRENT ADDRESS
006564  012777  177400  172542         MOV     #-400,@RKWC            ;SET UP WORD COUNT 400 WORDS
006572  013777  001350  172540         MOV     DRIVA0,@RKDA           ;SET UP DISK ADDRS, SECTOR 0, CYLINDER 0
006600  012711  002003                 MOV     #2003,@R1              ;WRITE FORMAT, GO

006604  105711                 1$:     TSTB    @R1                    ;WAS 'CNTRL RDY' CLEARED AS GO WAS SET?
006606  100003                         BPL     2$                     ;YES, BRANCH
006610  004737  020776                 JSR     PC,GT3RG               ;GO, GET RKCS, ER, DS
```

```
2822  006614  104030                          ERROR   30              ;'CNTRL RDY' DIDN'T CLEAR AS GO
2823                                                                  ;WAS SET TO 'WRITE FORMAT'
2824  006616  005000                  2$:     CLR     R0
2825  006620  105711                          TSTB    @R1             ;WAS 'CNTRL RDY' SET ON COMPLETION OF WRITE?
2826  006622  100411                          BMI     3$              ;YES, BRANCH
2827  006624  005200                          INC     R0              ;NO, HAVE U WAITED LONG ENOUGH?
2828  006626  001374                          BNE     2$+2            ;IF NOT, LOOP BACK & WAIT
2829                                                                  ;IF YES, REPORT ERROR
2830  006630  004737  020770                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
2831  006634  013737  001350  001202          MOV     DRIVAD,$REG10
2832  006642  104416                          BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                                      ;$REG10 INTO DR #,CYL,SUR,SEC BITS
2833
2834  006644  104031                          ERROR   31              ;'CNTRL RDY' DIDN'T SET ON COMPLETION
2835                                                                  ;OF WRITE FORMAT
2836                                                                  ;WRT FMT WAS DONE STARTING AT <DSK-ADRES>
2837                                                                  ;INDICATED IN EROR MSGE.
2838  006646  004737  021230          3$:     JSR     PC,CHKHE        ;GO CHECK IF 'HE' OR 'ERR' BIT SET,
2839                                                                  ;IF YES, SAVE RKCS, ER, DS, DA.
2840                                                                  ;RETURN HERE IF ERROR.
2841  006652  104032                          ERROR   32              ;'HE' OR 'ERR' BIT SET WHILE DOING
2842                                                                  ;A WRITE FORMAT
2843                                                                  ;WRT FMT WAS DONE STARTING AT <DSK-ADRES
2844                                                                  ;INDICATED IN EROR MSGE.
2845  006654  004737  021256          4$:     JSR     PC,CHKDA        ;GO CHECK IF RKDA INCREMENTED CORRECTLY
2846                                                                  ;IF NOT, RETURN HERE.
2847  006660  104033                          ERROR   33              ;RKDA SHOULD HAVE INCREMENTED BY
2848                                                                  ;1 SECTOR, IT DID NOT
2849  006662  004737  021312          5$:     JSR     PC,CHKWC        ;CHECK IF WORD COUNT OVERFLOWED, IF
2850                                                                  ;NOT RETURN HERE.
2851  006666  104034                          ERROR   34              ;RKWC DID NOT OVERFLOW TO 0, AFTER
2852                                                                  ;XFER ON WRITE FORMAT
2853  006670  022712  034336          6$:     CMP     #OUTBUF+1000,@R2        ;DID RKBA INCREMENT CORRECTLY?
2854  006674  001406                          BEQ     7$              ;YES, BRANCH
2855  006676  012737  034336  001162          MOV     #OUTBUF+1000,$REG0 ;GET EXPCTD RKBA
2856  006704  011237  001164                  MOV     @R2,$REG1       ;GET ACTUAL RKBA
2857  006710  104035                          ERROR   35              ;RKBA DIDN'T INCREMENT BY 1000 AFTER
                                                                      ;WRITE FORMAT OF 400 WORDS
2858
2859  006712  004737  021336          7$:     JSR     PC,CHKER        ;CHECK IOF ANY BIT IN RKER SET.
2860                                                                  ;IF YES RETURN HERE.
2861  006716  104036                          ERROR   36              ;RKER BIT SET ON DOING 1 WORD
2862                                                                  ;WRITE FORMAT
2863  006720  022711  002202          8$:     CMP     #2202,@R1       ;DOES RKCS STILL HAVE 'WRT FMT' BITS?
2864  006724  001406                          BEQ     TST16           ;YES, EXIT
2865  006726  012737  002202  001162          MOV     #2202,$REG0     ;GET EXPCTD RKCS
2866  006734  011137  001164                  MOV     @R1,$REG1       ;GET ACTUAL RKCS
2867  006740  104024                          ERROR   24              ;RKCS DIDN'T CONTAIN 'WRT FMT' BITS
                                                                      ;AFTER THE FUNCTION WAS COMPLETED
2868
2869                                                  ;
2870                                          ;;*******************************************************
2871                                          ;*TEST 16        CHECK 'READ FORMAT' FUNCTION-CYLINDER 0, SECTOR 0
2872                                                  ;*THIS TEST CHECKS THE LOGIC INVOLVED IN THE WRITE FMT
2873                                                  ;*FUNCTION. ON ISSUING A WRT FMT, THE FOLLOWING IS CHECKED
2874                                                  ;*1)  CNTRL RDY WAS CLEARED AS GO WAS SET.
2875                                                  ;*2)  CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION OF FUNCTION
2876                                                  ;*3)  IF 'HE' OR 'ERR' BIT SET?
2877                                                  ;*4)  IF RKDA INCREMENTED CORRECTLY FROM 0 TO 1?
```

```
                        T16     CHECK 'READ FORMAT' FUNCTION-CYLINDER 0, SECTOR 0

                                ;*5)  IF RKWC OVERFLOWED CORRECTLY TO 0?
                                ;*6)  IF RKBA INCREMENTED CORRECTLY BY 2?
                                ;*7)  IF ANY BIT IN RKER SET?
                                ;*8)  IF THE CORRECT HEADER WAS RECEIVED?
                                ;*9)  FOR RK11C, AFTER RD FMT RKDB CONTAINS THE CHECkSUM
                                ;*FOR THAT SECTOR. (125252 IN THIS CASE, BECAUSE THE
                                ;*FIRST WORD IN SEC 0 WAS WRITTEN AS 125252 IN
                                ;*THE PREVIOUS TEST)
                                ;*10)  FOR RK11D, AFTER RD FMT RKDB SHOULD CONTAIN
                                ;*A ZERO
                                ;*11)  IF THE RD FMT FUNCTION BITS ARE STILL IN
                                ;*THE RKCS?
                        ;;*****************************************************************
006742  000004         TST16:  SCOPE
006744  005000                 CLR     R0
006746  104413                 CNT.RESET               ;GO, DO CONTROL RESET
                                                        ;THIS IS A CALL FOR THE 'CNTRL-
                                                        ;RESET' ROUTINE. A CONTROL RESET IS
                                                        ;ISSUED AND AFTER A CERTAIN TIME
                                                        ;IF THE 'CNTRL RDY' DOES NOT SET
                                                        ;AN ERROR IS REPORTED. NOTE THAT
                                                        ;THE PC IN ERROR MESSAGE IS THE
                                                        ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                        ;THIS IS A VERY BASIC ERR & IF IT
                                                        ;OCCURS GO BACK TO TEST 10
006750  104421                 TST.SIN                ;GO CHECK IF SIN IS SET
                                                        ;IF SET, DO DRIVE RESET TO CLR IT
006752  013701  001332         MOV     RKCS,R1
006756  013702  001336         MOV     RKBA,R2
006762  012703  033336         MOV     #OUTBUF,R3
006766  010312                 MOV     R3,@R2         ;SETUP ADRS WHERE HEADER WORC IS TO BE
                                                        ;X-FERRED
006770  012777  177777  172336 MOV     #-1,@RKWC      ;SET UP WORD COUNT
006776  013777  001350  172334 MOV     DRIVAD,@RKDA   ;SET UP DISK ADRS, SECTOR 0, CYLINDER 0
007004  012711  002005         MOV     #2005,@R1      ;READ FORMAT, GO

007010  105711          1$:    TSTB    @R1            ;WAS 'CNTRL RDY' CLEARED AS GO WAS SET?
007012  100003                 BPL     2$             ;YES, BRANCH
007014  004737  020776         JSR     PC,GT3RG       ;GO, GET RKCS, RKER
007020  104030                 ERROR   30             ;CNTRL RDY DIDN'T CLEAR AS GO WAS
                                                        ;SET TO 'READ FORMAT'
007022  005000          2$:    CLR     R0
007024  105711                 TSTB    @R1            ;WAS 'CNTRL RDY' SET ON COMPLETION OF
                                                        ;TRANSFER
007026  100411                 BMI     3$             ;YES, BRANCH
007030  005200                 INC     R0             ;NO, HAVE U WAITED LONG ENOUGH?
007032  001374                 BNE     2$+2           ;IF NOT, LOOP BACK & WAIT
                                                        ;IF YES, REPORT ERROR
007034  004737  020770         JSR     PC,GT4RG       ;GO, GET RKCS, ER, DS,DA
007040  013737  001350  001202 MOV     DRIVAD,$REG1D
007046  104416                 BRKDA4                 ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                        ;$REG1D INTO DR #,CYL,SUR,SEC BITS
007050  104045                 ERROR   45             ;'CNTRL RDY' DIDN'T SET ON COMPLETION
                                                        ;OF READ FORMAT
                                                        ;READ FMT WAS DONE STARTING AT <DSK-ADRES>
                                                        ;INDICATED IN EROR MESGE
```

```
2934  007052  004737  021230          3$:     JSR     PC,CHKHE        ;CHECK IF 'ERR' OR 'HE' BIT SET, IF
2935                                                                    ;YES RETURN HERE.
2936  007056  104046                          ERROR   46              ;'HE' OR 'ERR' BIT SET WHILE
2937                                                                    ;DOING A 'READ FORMAT'
2938                                                                    ;READ FMT WAS DONE STARTING AT <DSK-ADRES>
2939                                                                    ;INDICATED IN EROR MESGE
2940  007060  004737  021256          4$:     JSR     PC,CHKDA        ;CHECK IF RKDA INCREMENTED CORRECTLY
2941                                                                    ;IF NOT, RETURN HERE.
2942  007064  104040                          ERROR   40              ;RKDA SHOULD HAVE INCREMENTED
2943                                                                    ;BY 1 SECTOR, IT DID NOT
2944
2945  007066  004737  021312          5$:     JSR     PC,CHKWC        ;CHECK IF RKWC OVERFLOWED TO 0, IF
2946                                                                    ;NOT RETURN HERE.
2947  007072  104041                          ERROR   41              ;RKWC DID NOT OVERFLOW TO 0
2948                                                                    ;AFTER XFER ON READ FORMAT
2949  007074  022712  033340          6$:     CMP     #OUTBUF+2,@R2   ;DID RKBA INCREMENT TO NXT WORD ADDRS?
2950  007100  001406                          BEQ     7$              ;YES, BRANCH
2951  007102  012737  033340  001162          MOV     #OUTBUF+2,$REG0 ;GET EXPCTD RKBA
2952  007110  011237  001164                  MOV     @R2,$REG1       ;GET ACTUAL RKBA
2953  007114  104042                          ERROR   42              ;RKBA DIDN'T INCREMENT BY 2 AFTER
2954                                                                    ;'READ FORMAT' OF 1 WORD
2955  007116  004737  021336          7$:     JSR     PC,CHKER        ;CHECK IF ANY BIT IN RKER SET, IF
2956                                                                    ;YES RETURN HERE.
2957  007122  104036                          ERROR   36              ;RKER BIT SET ON DOING
2958                                                                    ;1 WORD READ FORMAT
2959  007124  005713                  8$:     TST     @R3             ;DOES OUTBUF CONTAIN THE HEADER
2960                                                                    ;WORD-0
2961  007126  001407                          BEQ     9$              ;YES, BRANCH
2962  007130  005037  001162                  CLR     $REG0           ;GET SECTOR NO.
2963  007134  005037  001164                  CLR     $REG1           ;EXPCTD HEADER
2964  007140  011337  001166                  MOV     @R3,$REG2       ;GET HEADER RECVD
2965  007144  104043                          ERROR   43              ;CORRECT HEADER WORD-0-WAS
2966                                                                    ;NOT RECEIVED ON READ FORMAT
2967  007146  022711  002204          9$:     CMP     #2204,@R1       ;DOES RKCS HAVE THE 'RDFMT' BITS?
2968  007152  001406                          BEQ     TST17           ;;YES, BRANCH
2969  007154  012737  002204  001162          MOV     #2204,$REG0     ;GET EXPCTD RKCS
2970  007162  011137  001164                  MOV     @R1,$REG1       ;GET ACTUAL RKCS
2971  007166  104024                          ERROR   24              ;RKCS DIDN'T CONTAIN 'RD FMT'
2972                                                                    ;BITS AFTER FUNCTION WAS
2973                                                                    ;COMPLETED
2974
2975
2976
2977                                  ;;**********************************************************************
2978                                  ;*TEST 17        CHECK 'READ' FUNCTION-CYLINDER 0,SECTOR 0
2979                                          ;*THIS IS THE FIRST TIME A PURE READ IS PREFORMED IN THIS
2980                                          ;*TEST SEQUENCE.  THE FOLLOWING IS CHECKED
2981                                          ;*1) CNTRL RDY CLEARS AS GO IS SET
2982                                          ;*2) CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
2983                                          ;*OF FUNCTION
2984                                          ;*3) IF 'HE' OR 'ERR' BIT SET?
2985                                          ;*4) IF RKDA INCREMENTED CORRECTLY?
2986                                          ;*5) IF RKWC OVERFLOWED TO 0?
2987                                          ;*6) IF RKBA INCREMENTED CORRECTLY?
2988                                          ;*7) IF ANY RKER BIT SET?
2989                                          ;*8) IF THE CORRECT PSUEDO-HEADER (FIRST WORD) WAS
```

```
                                        ;*READ FROM SECTOR 0
                                        ;*9) IF THE 'READ' FUNCTION BITS ARE STILL IN RKCS
                                        ;;*****************************************************************
007170  000004                  TST17:  SCOPE
007172  104413                          CNT.RESET               ;GO, DO CONTROL RESET
                                                                ;THIS IS A CALL FOR THE 'CNTRL-
                                                                ;RESET' ROUTINE. A CONTROL RESET IS
                                                                ;ISSUED AND AFTER A CERTAIN TIME
                                                                ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                ;AN ERROR IS REPORTED. NOTE THAT
                                                                ;THE PC IN ERROR MESSAGE IS THE
                                                                ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                ;THIS IS A VERY BASIC ERR & IF IT
                                                                ;OCCURS GO BACK TO TEST 10
007174  104421                          TST.SIN                 ;GO CHECK IF SIN IS SET
                                                                ;IF SET, DO DRIVE RESET TO CLR IT
007176  013701  001332                  MOV     RKCS,R1
007202  005000                          CLR     R0
007204  013702  001336                  MOV     RKBA,R2
007210  012703  033336                  MOV     #OUTBUF,R3
007214  010312                          MOV     R3,@R2          ;SET UP ADDRS WHERE DATA WORD IS
                                                                ;TO BE X-FERRED
007216  012777  177400  172110          MOV     #-400,@RKWC     ;SET UP WORD COUNT
007224  013777  001350  172106          MOV     DRIVAD,@RKDA    ;SET UP DISK ADRS, SECTOR 0, CYLINDER 0
007232  012711  000005                  MOV     #5,@R1          ;READ, GO

007236  105711          1$:     TSTB    @R1             ;WAS 'CNTRL RDY' CLEARED AS GO WAS SET?
007240  100003                  BPL     2$              ;YES, BRANCH
007242  004737  020776          JSR     PC,GT3RG        ;GO, GET RKCS, ER
007246  104030                  ERROR   30              ;CNTRL RDY DID NOT CLEAR AS GO
                                                        ;WAS SET TO 'READ'
007250  005000          2$:     CLR     R0
007252  105711                  TSTB    @R1             ;WAS CNTRL RDY SET ON COMPLETION
                                                        ;OF TRANSFER?
007254  100411                  BMI     3$              ;YES, BRANCH
007256  005200                  INC     R0              ;NO, HAVE U WAITED LONG ENOUGH?
007260  001374                  BNE     2$+2            ;IF NOT, LOOP BACK & WAIT
                                                        ;IF YES, REPORT ERROR
007262  004737  020770          JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
007266  013737  001350  001202  MOV     DRIVAD,$REG1O
007274  104416                  BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                        ;$REG1O INTO DR #,CYL,SUR,SEC BITS
007276  104045                  ERROR   45              ;CNTRL RDY DID NOT SET ON
                                                        ;COMPLETION OF READ
                                                        ;READ WAS DONE STARTING AT  DSK-ADRES
                                                        ;INDICATED IN EROR MESGE

007300  004737  021230  3$:     JSR     PC,CHKHE        ;CHECK IF 'ERR' OR 'HE' BIT IS SET
                                                        ;IF YES, RETURN HERE.
007304  104046                  ERROR   46              ;'HE' OR 'ERR' BIT SET WHILE
                                                        ;DOING A READ.
                                                        ;READ WAS DONE STARTING AT <DSK-ADRES>
                                                        ;INDICATED IN EROR MESGE
007306  004737  021256  4$:     JSR     PC,CHKDA        ;CHECK IF RKDA INCREMENTED CORRECTLY.
                                                        ;IF NOT RETURN HERE.
007312  104040                  ERROR   40              ;RKDA DID NOT INCREMENT
```

```
3046
3047  007314  004737  021312          5$:   JSR     PC,CHKWC          ;BY 1 (SECTOR)
3048                                        ;CHECK IF RKWC OVERFLOWED TO 0.
3049  007320  104041                  ;IF NOT RETURN HERE.
3050                                  ;RKWC DID NOT OVERFLOW TO 0,
3051  007322  022712  034336          6$:   CMP     #OUTBUF+1000,@R2  ;AFTER X-FER ON READ
3052  007326  001406                  ;DID RKBA INCREMENT CORRECTLY?
3053  007330  012737  034336  001162        BEQ     7$                ;YES, BRANCH
3054  007336  011237  001164              MOV     #OUTBUF+1000,$REG0  ;GET EXPCTD RKBA
3055  007342  104042                  ;GET ACTUAL RKBA
3056                                  ;RKBA DID NOT INCREMENT BY 2
3057  007344  004737  021336          7$:   MOV     @R2,$REG1
3058                                  ;AFTER 'READ' OF 1 WORD
3059  007350  104036                  ;CHECK IF ANY BIT IN RKER SET.
3060                                  ;IF YES RETURN HERE.
3061  007352  022713  000001          8$:   ERROR   36                ;RKER BIT SET ON DOING 1
3062                                  ;WORD 'READ'
3063  007356  001411                  ;DOES OUTBUF CONTAIN THE RIGHT
3064  007360  012737  000001  001162        CMP     #1,@R3            ;DATA WORD
3065  007366  011337  001164              BEQ     9$                ;YES BRANCH
3066  007372  013737  001350  001166        MOV     #1,$REG0          ;GET EXPCTD DATA WORD
3067  007400  104044                      MOV     (R3),$REG1        ;GET RECVD DATA WORD
3068                                      MOV     DRIVAD,$REG2      ;GET DISK ADRS FROM WHICH READ WAS DONE
3069                                      ERROR   44                ;DID NOT READ THE CORRECT
3070                                                                ;DATA WORD--FROM DISK ADRES.
3071
3072                                                                ;SEC 0, CYL 0, SUR 0
3073
3074                                                                ;AFTER 1 SECTOR READ RKDB CONTAINS
3075                                                                ;FOR RK11C
3076                                                                ;THE CHECKSUM FOR THAT SECTOR
3077                                                                ;FOR RK11D
3078                                                                ;THE LAST WORD TRANSFERRED TO MEMORY
3079                                                                ;
3080                                                                ;IT SO HAPPENS THAT WITH THE SECTOR
3081                                                                ;THAT WAS READ, RKDB CONTAINS THE
3082  007402  022777  125252  171732  9$:   CMP     #125252,@RKDB     ;SAME INFORMATION FOR BOTH RK11C
3083  007410  001407                      BEQ     10$               ;AND RK11D
3084  007412  012737  125252  001162        MOV     #125252,$REG0     ;DOES RKDB CONTAIN THE EXPCTD WORD?
3085  007420  017737  171716  001164        MOV     @RKDB,$REG1       ;YES, BRANCH
3086  007426  104037                      ERROR   37                ;GET EXPCTD RKDB
3087                                                                ;GET RECVD RKDB
3088                                                                ;RKDB DOESD NOT CONTAIN THE
3089  007430  022711  000204          10$:  CMP     #204,@R1          ;EXPCTD WORD AFTER A READ OF SEC 0
3090  007434  001406                      BEQ     11$               ;CYL 0
3091  007436  012737  000204  001162        MOV     #204,$REG0        ;DOES RKCS HAVE THE 'READ' BITS?
3092  007444  011137  001164              MOV     @R1,$REG1         ;YES , BRANCH
3093  007450  104024                      ERROR   24                ;GET EXPCTD RKCS
3094                                                                ;GET RECVD RKCS
3095                                                                ;RKCS DID NOT CONTAIN 'READ'
3096  007452  104413          11$:  CNT.RESET                       ;FUNCTION BITS AFTER OPERATION
3097  007454  005777  171662              TST     @RKDB             ;WAS COMPLETED
3098  007460  001407                      BEQ     TST20             ;GO DO CONTROL RESET
3099  007462  013737  001342  001164        MOV     RKDB,$REG1        ;DID CONTROL RESET CLEAR RKDB?
3100  007470  017737  171646  001164        MOV     @RKDB,$REG1       ;;YES, EXIT
3101  007476  104102                      ERROR   102               ;GET ADRES OF RKDB
                                                                    ;GET CONTENTS OF RKDB
                                                                    ;CONTROL RESET DIDN'T CLR RKCS
```

```
3102
3103                                    ;;*****************************************************************
3104                                    ;*TEST 20       CHECK 'WRITE FORMAT' -CYLINDER 0, SECTOR 0-13
3105                                            ;*THIS TEST GOES ONE STEP FURTHER & PERFORMS A WRT
3106                                            ;*FMT ON CYLINDER 0 & CHECKS THE FOLLOWING
3107                                            ;*1)  IF CNTRL RDY SET WITHIN A CERTAIN TIME ON COMPLETION
3108                                            ;*OF THE FUNCTION
3109                                            ;*2)  IF 'HE' OR 'ERR' BIT SET?
3110                                            ;*3)  IF THE RKDA INCREMENTS CORRECTLY?
3111                                            ;*4)  IF THE RKDB IS CLEAR?
3112                                            ;*WRT FMT IS DONE ONE SECTOR AT A TIME
3113                                            ;*THE FIRST WORD OF EVERY SECTOR IS WRITTEN AS A
3114                                            ;*PSUEDO-HEADER CONSISTING OF DRIVE #, CYLINDER #, SURFACE
3115                                            ;*& SECTOR #. THIS WILL BE READ & CHECKED IN THE FOLLOWING TEST.
3116                                    ;;*****************************************************************
3117  007500  000004                   TST20:  SCOPE
3118  007502  013703  001332                   MOV     RKCS,R3
3119  007506  012702  177764                   MOV     #-14.,R2        ;SET UP COUNT FOR 12 SECTORS
3120  007512  013704  001340                   MOV     RKDA,R4
3121  007516  013701  001350                   MOV     DRIVAD,R1       ;GET DRIVE ADDRESS
3122  007522  010105                           MOV     R1,R5           ;STORE IT
3123  007524  005205                           INC     R5
3124  007526  012737  007534  001110           MOV     #1$,$_PERR      ;SET RETURN ADRES FOR LUPING
3125                                                                    ;ON     ERROR (SW 9)
3126  007534  104413                   1$:     CNT.RESET               ;GO, DO CONTROL RESET
3127                                                                    ;THIS IS A CALL FOR THE 'CNTRL-
3128                                                                    ;RESET' ROUTINE. A CONTROL RESET IS
3129                                                                    ;ISSUED AND AFTER A CERTAIN TIME
3130                                                                    ;IF THE 'CNTRL RDY' DOES NOT SET
3131                                                                    ;AN ERROR IS REPORTED. NOTE THAT
3132                                                                    ;THE PC IN ERROR MESSAGE IS THE
3133                                                                    ;PC WHERE 'CNT.RESET' IS LOCATED.
3134                                                                    ;THIS IS A VERY BASIC ERR & IF IT
3135                                                                    ;OCCURS GO BACK TO TEST 10
3136  007536  104421                           TST.SIN                 ;GO CHECK IF SIN IS SET
3137                                                                    ;IF SET, DO DRIVE RESET TO CLR IT
3138  007540  005000                           CLR     R0
3139  007542  010137  033336                   MOV     R1,OUTBUF       ;THIS WORD TO BE X-FERRED. FIRST
3140                                                                    ;WORD OF EACH SECTOR WILL BE THE
3141                                                                    ;ACTUAL DRIVE-ADDRS CONSISTING OF
3142                                                                    ;DRIVE NO, CYL ADDRS, SURFACE
3143                                                                    ;SECTOR NO.
3144  007546  012777  033336  171562           MOV     #OUTBUF,@RKBA   ;ADRS FROM WHICH DATA WORD IS TO
3145                                                                    ;X-FERRED
3146  007554  012777  177777  171552           MOV     #-1,@RKWC       ;SET UP WORD COUNT
3147  007562  010114                           MOV     R1,@R4  ;ADDRS THE DRIVE, CYL 0, & CORRECT SECTOR
3148  007564  012713  002003                   MOV     #2003,@R3       ;WRITE FORMAT, GO
3149
3150  007570  105777  171536           2$:     TSTB    @RKCS           ;DID 'CNTRL RDY' SET?
3151  007574  100410                           BMI     3$              ;YES, BRANCH
3152  007576  005200                           INC     R0              ;NO, HAVE U WAITED LONG?
3153  007600  001373                           BNE     2$              ;IF NOT, LOOP BACK & WAIT
3154                                                                    ;IF YES, REPORT ERROR
3155  007602  004737  020770                   JSR     PC,GTHRG        ;GO, GET RKCS, ER, DS,DA
3156  007606  013137  001202                   MOV     R1,$REG10       ;GET DISK ADRS (UNIT,CYL,SUR,SEC) TO WHICH
3157                                                                    ;WRITE FORMAT WAS DONE
```

```
3158  007612  104416                          BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
3159                                                                  ;$REGIO INTO DR #,CYL,SUR,SEC BITS
3160  007614  104031                          ERROR   31              ;'CNTRL RDY' DID NOT SET ON COMPLETION
3161                                                                  ;OF 'WRITE FORMAT'
3162                                                                  ;WRT FMT WAS DONE STARTING AT <DSK-ADRES>
3163                                                                  ;INDICATED IN EROR MSGE.
3164  007616  004737  021222          3$:     JSR     PC,CHKHE1       ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
3165                                                                  ;IF YES RETURN HERE.
3166  007622  104032                          ERROR   32              ;'HE' OR 'ERR' BIT SET WHILE DOING
3167                                                                  ;WRITE FORMAT ON CYLINDER 0,
3168                                                                  ;SECTOR IN ERROR IS AS SHOWN IN
3169                                                                  ;DISK-ADRES BITS 0-3
3170                                                                  ;WRT FMT WAS DONE STARTING AT <DSK-ADRES>
3171                                                                  ;INDICATED IN EROR MSGE.
3172
3173  007624  004737  021264          4$:     JSR     PC,CHKDA1       ;CHECK IF RKDA INCREMENTED CORRECTLY?
3174
3175  007630  104033                          ERROR   33              ;RKDA DID NOT INCREMENT CORRECT
3176                                                                  ;AFTER 1 WORD 'WRITE FORMAT' ON
3177                                                                  ;CYLINDER 0, SECTOR IN ERROR IS 1
3178                                                                  ;LESS THAN THAT SHOWN IN EXPCTD RKDA
3179  007632  005777  171504          5$:     TST     @RKDB           ;CHECK THAT RKDB DOES CONTAIN A 0
3180                                                                  ;AFTER WRT BECAUSE LAST WORD WRITTEN
3181                                                                  ;WAS SERIALLY SHIFTED OUT TO THE DISK
3182  007636  001406                          BEQ     6$              ;YES, BRANCH
3183  007640  005037  001162                  CLR     $REG0           ;THIS IS WHAT RKDB SHOULD CONTAIN
3184  007644  017737  171472  001164          MOV     @RKDB,$REG1     ;GET RKDB
3185  007652  104037                          ERROR   37              ;RKDB SHOULD BE 0 AFTER WRT SINCE THE
3186                                                                  ;LAST WORD WRITTEN WAS SERIALLY SHIFTED
3187                                                                  ;OUT OF RKDB
3188  007654  005201                  6$:     INC     R1              ;INCREMENT DRIVE ADDRS TO NXT SECTOR
3189  007656  005205                          INC     R5              ;
3190  007660  122705  000014                  CMPB    #14,R5          ;R U GOING TO CHECK THE LAST SECTOR?
3191  007664  001002                          BNE     .+6             ;IF NOT,BRANCH
3192  007666  062705  000004                  ADD     #4,R5           ;IF YES,INCREMENT R5 CORRECTLY TO 'EXPCTD RKDA'
3193                                                                  ;AFTER HAVING CHECKED THE LAST SECTOR
3194  007672  005202                          INC     R2              ;HAVE U FORMATTED ALL 12 SECTORS?
3195  007674  001317                          BNE     1$              ;IF NOT, BRANCH BACK & LOOP
3196                                                                  ;IF YES, EXIT
3197
3198                                  ;;*****************************************************************
3199                                  ;*TEST 21       CHECK 'READ FORMAT'-CYLINDER 0, SECTOR 0-13
3200                                          ;*THIS TEST PERFORMS A RD FMT ON THE 12 SECTORS OF CYLINDER 0
3201                                          ;*THE FOLLOWING IS CHECKED
3202                                          ;*1)  IF CNTRL RDY SET WITHIN A CERTAIN TIME ON COMPLETION
3203                                          ;*OF THE FUNCTION
3204                                          ;*2)  IF 'HE' OR 'ERR' BIT SET?
3205                                          ;*3)  IF THE RKDA INCREMENTS CORRECTLY?
3206                                          ;*4)  RKBA INCREMENTED CORRECTLY BY 30 (OCTAL)
3207                                          ;*5)  RKWC OVERFLOWED TO 0 FROM -14 (OCTAL)
3208                                          ;:6)  CORRECT HEADER WAS RECEIVED FROM ALL 12 SECTORS.
3209                                          ;*7)  RKCS STILL CONTAINS THE 'RD FMT' FUNCTION BITS.
3210                                          ;*IF THERE IS A READ ERROR IN THIS TEST OR ANY
3211                                          ;*OTHER TESTS THE USER SHOULD MAKE SURE THAT
3212                                          ;*IT IS AN IRRECOVERABLE ERROR AND NOT A TRANSIENT
3213                                          ;*ONE. THIS CAN BE DONE BY LOOPING ON THE TEST
```

```
                                        ;*IN QUESTION. USUALLY A TRANSIENT ERROR
                                        ;*DISAPPEARS ON RETRIES, WHEREAS A LOGIC ERROR DOES NOT.
                                ;;*****************************************************************
007676  000004                TST21:  SCOPE
007700  005005                        CLR     R5
007702  104413                        CNT.RESET                       ;GO, DO CONTROL RESET
                                                                        ;THIS IS A CALL FOR THE 'CNTRL-
                                                                        ;RESET' ROUTINE. A CONTROL RESET IS
                                                                        ;ISSUED AND AFTER A CERTAIN TIME
                                                                        ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                        ;AN ERROR IS REPORTED. NOTE THAT
                                                                        ;THE PC IN ERROR MESSAGE IS THE
                                                                        ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                        ;THIS IS A VERY BASIC ERR & IF IT
                                                                        ;OCCURS GO BACK TO TEST 10
007704  104421                        TST.SIN                         ;GO CHECK IF SIN IS SET
                                                                        ;IS SET, DO DRIVE RESET TO CLR IT
007706  013700  001332                MOV     RKCS,R1
007712  012700  177764                MOV     #-14,R0                 ;SET UP COUNT FOR 12 SECTORS
007716  013702  001340                MOV     RKDA,R2
007722  013712  001350                MOV     DRIVAD,@R2              ;ADDRESS THE DRIVE
007726  012704  033336                MOV     #OUTBUF,R4
007732  010477  171400                MOV     R4,@RKBA                ;ADRS TO WHICH X-FER DATA FROM DSK
007736  012777  177764  171370        MOV     #-14,@RKWC              ;SET UP WORD COUNT FOR 12 HEADERS TO BREAD
007744  012777  002005  171360        MOV     #2005,@RKCS             ;READ FORMAT, GO

007752  105777  171354        1$:     TSTB    @RKCS                   ;DID CNTRL RDY SET ON COMPLETION?
007756  100411                        BMI     2$                      ;YES, BRANCH
007760  005205                        INC     R5                      ;NO, WAIT FOR IT TO SET
007762  001373                        BNE     1$                      ;IF WAITED LONG ENOUGH REPORT
                                                                        ;ERROR, OTHERWISE LOOP BACK & WAIT
007764  004737  020770                JSR     PC,GT4RG                ;GO, GET RKCS, ER, DS,DA
007770  013737  001350  001202        MOV     DRIVAD,$REG10
007776  104416                        BRKDA4                          ;GO TO 'BDA4' & BREAK CONTENTS OF
                                                                        ;$REG10 INTO DR#,CYL,SUR,SEC BITS
010000  104045                        ERROR   45                      ;CNTRL RDY DID NOT SET ON COMPLETION
                                                                        ;OF READ FORMAT-OF CYLINDER 0,
                                                                        ;SECTORS 0-13
                                                                        ;READ FMT WAS DONE STARTING AT <DSK-ADRES
                                                                        ;INDICATED IN EROR MESGE
010002  004737  021230        2$:     JSR     PC,CHKHE                ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
                                                                        ;IF YES RETURN HERE.
010006  104046                        ERROR   46                      ;'ERR' OR 'HE' BIT SET ON DOING
                                                                        ;READ FMT-OF CYLINDER 0, SEC 0-13
                                                                        ;READ FMT WAS DONE STARTING AT <DSK-ADRES
                                                                        ;INDICATED IN EROR MESGE
010010  013705  001350        3$:     MOV     DRIVAD,R5
010014  062705  000020                ADD     #20,R5                  ;RKDA SHOULD HAVE INCREMENTD TO (R2)

010020  004737  021264                JSR     PC,CHKDA1               ;CHECK IF RKDA INCREMENTED CORRECTLY,
                                                                        ;IF NOT, RETURN HERE.
010024  104040                        ERROR   40                      ;RKDA DID NOT INCREMENT BY 12
                                                                        ;AFTER A 'RD FMT' OF 12 HEADERS OF
                                                                        ;CYLINDER 0, SECTORS 0-13
                                                                        ;RKBA SHOULD INCREMENT BY 24 BYTES
                                                                        ;AT THE END OF X-FER
```

```
3270  010026  022777  033366  171302  4$:   CMP   #OUTBUF+30,@RKBA ;DID RKBA INCREMENT CORRECTLY?
3271  010034  001407                        BEQ   5$              ;YES, BRANCH
3272  010036  012737  033366  001162        MOV   #OUTBUF+30,$REGO        ;GET EXPCTD RKBA
3273  010044  017737  171266  001164        MOV   @RKBA,$REG1     ;GET ACTUAL RKBA
3274  010052  104042                        ERROR 42              ;RKBA DID NOT INCREMENT CORRECTLY
3275                                                              ;AFTER READ FORMAT OF 12 HEADERS
3276  010054  004737  021312          5$:   JSR   PC,CHKWC        ;GO CHECK IF RKWC OVERFLOWED TO 0
3277                                                              ;IF NOT RETURN HERE.
3278  010060  104041                        ERROR 41              ;RKWC DID NOT OVERFLOW TO 0
3279                                                              ;AFTER 'RD FMT' OF 12 HEADERS
3280                                                              ;OF CYLINDER 0
3281  010062  005724                  6$:   TST   (R4)+           ;WAS THE CORRECT HEADER RECIEVED?
3282  010064  001413                        BEQ   7$              ;YES, BRANCH
3283  010066  010037  001162                MOV   R0,$REGO        ;GET SECTOR FOR WHICH THE HEADER
3284  010072  062737  000014  001162        ADD   #14,$REGO       ;COULD NOT BE READ CORRECT
3285  010100  005037  001164                CLR   $REG1           ;EXPCTD HEADER-0, FOR CYL 0
3286  010104  014437  001166                MOV   -(R4),$REG2     ;GET WRONG HEADER RECVD
3287  010110  104043                        ERROR 43              ;HEADER WAS NOT READ RIGHT FOR
3288                                                              ;SECTOR (AS IN ER MSGE), & CYL 0
3289  010112  005724                        TST   (R4)+           ;WAS THE CORRECT HEADER RECVD?
3290  010114  005200                  7$:   INC   R0              ;YES, HAVE U CHECKED FOR ALL 12 SECTORS?
3291  010116  001361                        BNE   6$              ;IF NOT, LOOP BACK & CHK HDR FRM NXT SECTR
3292
3293  010120  004737  021336                JSR   PC,CHKER        ;CHECK IF ANY BIT IN RKER IS SET,
3294                                                              ;IF YES, RETURN HERE.
3295  010124  104036                        ERROR 36              ;RKER BIT SET ON DOING RD FMT
3296                                                              ;OF CYL 0, SECTORS 0-13
3297  010126  022711  002204          8$:   CMP   #2204,@R1       ;DOES RKCS STILL CONTAIN FUNCTION BITS?
3298  010132  001406                        BEQ   TST22           ;YES, EXIT
3299  010134  012737  002204  001162        MOV   #2204,$REGO     ;GET EXPCTD RKCS
3300  010142  011137  001164                MOV   @R1,$REG1       ;GET ACTUAL RKCS
3301  010146  104024                        ERROR 24              ;RKCS DID NOT CONTAIN 'RD FMT'
3302                                                              ;FUNCTION BITS ON COMPETION OF
3303                                                              ;THE FUNCTION
3304
3305
3306
3307                                  ;;******************************************************************
3308                                  ;*TEST 22       CHECK 'READ',CYLINDER 0, SECTORS 0 TO 13
3309                                          ;*THIS TEST PERFORMS A READ OF ALL THE SECTORS OF CYLINDER 0
3310                                          ;*& CHECKS THE FOLLOWING
3311                                          ;*1) CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
3312                                          ;*OF THE FUNCTION
3313                                          ;*2) IF 'NE' OR 'ERR' BIT SET?
3314                                          ;*3) IF THE CORRECT PSUEDO-HEADER (FIRST WORD OF EVERY)
3315                                          ;*SECTOR, WRITTEN IN A PREVIOUS TEST) WAS RECEIVED.
3316                                          ;*4) IF RKDS CONTAINS THE CORRECT WORD.
3317                                          ;*4) IF RKDA INCREMENTED CORRECTLY.
3318                                          ;*5) IF REST OF THE (377) WORDS IN EACH SECTOR ARE '0' . NOTE
3319                                          ;*PREVIOUSLY ONE WORD WAS WRITTEN PER SECTOR.
3320                                          ;*6) IF RKCS STILL CONTAINS THE 'READ' FUNCTION BITS
3321                                          ;*7) IF CONTROL RESET CLEARS RKDB.
3322                                          ;*  IF TESTING IS BEING DONE ON A SIMULATOR ONLY LAST SECTOR(13)
3323                                          ;*IS READ BECAUSE THE SIMULATOR CAN STORE ONLY 1 SECTOR (256 WORDS).
3324                                          ;*HENCE ONLY THE DATA WRITTEN LAST CAN BE READ BACK.
3325                                  ;;******************************************************************
```

```
3326  010150  000004                    TST22:  SCOPE
3327  010152  012737  010224  001110            MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
3328                                                                    ;ON      ERROR (SW 9)
3329  010160  013703  001332                    MOV     RKCS,R3
3330  010164  013701  001350                    MOV     DRIVAD,R1
3331  010170  010105                            MOV     R1,R5
3332  010172  012704  033336                    MOV     #OUTBUF,R4
3333  010176  005737  001344                    TST     SIMUL           ;TESTING ON SIMULATOR?
3334  010202  001405                            BEQ     9$              ;NO, BRANCH
3335                                                                    ;IF TESTING ON SIMULATOR READ
3336                                                                    ;SECTOR  13  ONLY
3337  010204  052701  000013                    BIS     #13,R1          ;SET BITS FOR SEC 13
3338  010210  052705  000020                    BIS     #20,R5          ;RKDA SHOULD INCRMNT TO THIS AFTER READ
3339  010214  000403                            BR      1$
3340  010216  012702  177764            9$:     MOV     #-14,R2         ;SET COUNT FOR 12 SECTORS
3341  010222  005205                            INC     R5              ;RKDA SHOULD INCREMENT TO
3342                                                                    ;THIS AFTER 1 SECTOR  READ
3343  010224  104413                    1$:     CNT.RESET               ;GO, DO CONTROL RESET
3344                                                                    ;THIS IS A CALL FOR THE 'CNTRL-
3345                                                                    ;RESET' ROUTINE. A CONTROL RESET IS
3346                                                                    ;ISSUED AND AFTER A CERTAIN TIME
3347                                                                    ;IF THE 'CNTRL RDY' DOES NOT SET
3348                                                                    ;AN ERROR IS REPORTED. NOTE THAT
3349                                                                    ;THE PC IN ERROR MESSAGE IS THE
3350                                                                    ;PC WHEPE 'CNT.RESET' IS LOCATED.
3351                                                                    ;THIS IS A VERY BASIC ERR & IF IT
3352                                                                    ;OCCURS GO BACK TO TEST 10
3353  010226  104421                            TST.SIN                 ;GO CHECK IF SIN IS SET
3354                                                                    ;IF SET, DO DRIVE RESET TO CLR IT
3355  010230  010177  171104                    MOV     R1,@RKDA        ;ADDRESS THE DRIVE
3356  010234  010477  171076                    MOV     R4,@RKBA        ;ADRS TO WHICH X-FER DATA FROM DISK
3357  010240  012777  177400  171066            MOV     #-400,@RKWC     ;SETUP WORD COUNT
3358  010246  012713  000005                    MOV     #5,@R3          ;READ,GO
3359
3360  010252  005000                            CLR     R0
3361  010254  105713                    2$:     TSTB    @R3             ;DID CNTRL RDY SET ON COMPETION?
3362  010256  100410                            BMI     3$              ;YES, BRANCH
3363  010260  005200                            INC     R0              ;NO, WAIT FOR IT TO SET
3364  010262  001374                            BNE     2$              ;IF WAITED LONG ENOUGH, REPORT
3365                                                                    ;ERROR, OTHERWISE LOOP BAK & WAIT
3366  010264  004737  020770                    JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
3367  010270  010137  001202                    MOV     R1,$REG10       ;GET SECTOR ADDRES WHERE ERROR OCCURED
3368  010274  104416                            BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
3369                                                                    ;$REG10 INTO DR #,CYL,SUR,SEC BITS
3370  010276  104045                            ERROR   45              ;CNTRL RDY DID NOT SET ON COMPLETION
3371                                                                    ;OF READ OF CYLINDER 0, SECTOR
3372                                                                    ;AS SHOWN IN <DSK-ADRES>
3373                                                                    ;READ WAS DONE STARTING AT <DSK-ADRES
3374                                                                    ;INDICATED IN EROR MESGE
3375  010300  004737  021222            3$:     JSR     PC,CHKHE1       ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
3376                                                                    ;IF YES RETURN HERE.
3377  010304  104046                            ERROR   46              ;HE OR ERR BIT SET
3378                                                                    ;ON 'READ' OF CYLINDER 0, SECTOR
3379                                                                    ;AS SHOWN IN <DSK-ADRES>
3380                                                                    ;READ WAS DONE STARTING AT <DSK-ADRES>
3381                                                                    ;INDICATED IN EROR MESGE
```

```
3382  010306  020114                  4$:     CMP     R1,(R4)         ;WAS THE DATA WORD RECVD, CORRECT?
3383                                                                    ;THE FIRST DATA WORD OF EACH SECTOR
3384                                                                    ;IS AN ADRS WORD COMRISING OF DRIVE NO,
3385                                                                    ;CYLINDER ADRS, SUR, SECTOR ADRS
3386  010310  001407                          BEQ     5$
3387  010312  010137  001162                  MOV     R1,$REG0        ;GET EXPCTD DATA WORD FROM DISK
3388  010316  011437  001164                  MOV     (R4),$REG1      ;GET THE DATA WORD RECVD
3389  010322  010137  001166                  MOV     R1,$REG2        ;GET DISK ADRES
3390  010326  104044                          ERROR   44              ;DID NOT RECIEVE CORRECT DATA WORD ON
3391                                                                    ;READ, OF CYLINDER 0, SECTOR AS SHOWN IN 'DSK
3392                                                                    ;ADRES' OF EXPCTD DATA WORD
3393  010330  004737  021264          5$:     JSR     PC,CHKDA1       ;CHECK IF RKDA INCREMENTED CORRECTLY,
3394                                                                    ;IF NOT RETURN HERE.
3395  010334  104040                          ERROR   40              ;RKDA DID NOT INCREMENT CORRECTLY
3396                                                                    ;AFTER READ OF 1 WORD, FROM CYL 0
3397                                                                    ;SEC IN ERROR IS 1 LESS THAN THAT
3398                                                                    ;SHOWN IN EXPCTD RKDA
3399                                                                    ;
3400                                                                    ;AS A RESULT OF 'WRT FMT' IN A PREVIOUS TEST
3401                                                                    ;FIRST WORD OF EVERY SECTOR IS NON-
3402                                                                    ;ZERO (PSUEDO-HDR), REST 377 WORDS
3403                                                                    ;ARE ALL 0'S.
3404                                                                    ;CHECK IF THE REST OF THE 377
3405                                                                    ;WORDS ARE ALL 0'S
3406  010336  012737  177775  001370          MOV     #-3,EFLG1       ;ALLOW ONLY 3 ERRORS
3407  010344  012700  033340                  MOV     #OUTBUF+2,R0    ;INITIALIZE PTR TO 2ND WRD IN BUFR
3408  010350  012737  177401  001362          MOV     #-377,COUNT     ;CHECK 377 WORDS IN THE BUFFER
3409  010356  005710                  11$:    TST     @R0             ;IS THIS WRD 0?
3410  010360  001005                          BNE     12$             ;NO, ERROR
3411  010362  005720                          TST     (R0)+           ;INCRMNT PTR TO NXT WRD
3412  010364  005237  001362                  INC     COUNT           ;CHKD ALL 377 WRDS?
3413  010370  001372                          BNE     11$
3414  010372  000412                          BR      7$              ;YES, BRANCH
3415  010374  005037  001162          12$:    CLR     $REG0           ;GET EXPCTD WORD
3416  010400  012037  001164                  MOV     (R0)+,$REG1     ;GET WORD RECVD
3417  010404  010137  001166                  MOV     R1,$REG2        ;GET DISK ADRES, ERROR IN THIS
3418                                                                    ;SECTOR
3419  010410  104044                          ERROR   44              ;DATA ERROR, THE LAST 377 WORDS
3420                                                                    ;READ FROM EACH SECTOR SHOULD BE 0
3421                                                                    ;IN A PREVIOUS TEST, FIRST WORD OF
3422                                                                    ;EVERY SEC (CYL 0) WAS WRITTEN AS A
3423                                                                    ;PSUEDO-HDR, REST OF THE WORDS IN THE
3424                                                                    ;SECTR ARE AUTOMATICALLY WRITTEN AS
3425                                                                    ;0'S. THIS ERROR MAY MEAN THAT IT
3426                                                                    ;DIDN'T HAPPEN SO
3427  010412  005237  001370                  INC     EFLG1           ;ALLOW ONLY 3 DATA ERORS OF THIS KIND
3428  010416  001357                          BNE     11$
3429
3430
3431  010420  005737  001344          7$:     TST     SIMUL           ;TESTING ON SIMULATOR?
3432  010424  001011                          BNE     10$             ;YES BRANCH
3433                                                                    ;IF NOT TESTING ON SIMULATOR GO AHEAD
3434                                                                    ; & READ ALL 12 SECTORS ON CYL 0
3435  010426  005201                          INC     R1              ;INCREMENT DRIV-ADRES TO NXT SECTOR
3436  010430  005205                          INC     R5              ;INCREMENT 'EXPCTD DRIV-ADRES'
3437  010432  122705  000014                  CMPB    #14,R5          ;R U GOING TO READ THE LAST SECTOR?
```

```
3438  010436  001002                        BNE     .+6             ;IF NOT, BRANCH
3439  010440  062705  000004                ADD     #4,R5           ;IF YES, INCREMENT 'EXPCTD RKDA'
3440                                                                  ;CORRECTLY
3441  010444  005202                        INC     R2              ;HAVE U READ ALL 12 SECTORS?
3442  010446  001266                        BNE     1$              ;IF NOT LOOP BACK & READ THE
3443                                                                  ;NXT SECTOR
3444  010450  022713  000204        10$:    CMP     #204,@R3        ;DOES RKCS, STILL HAVE THE 'READ' FUNCTION
3445  010454  001406                        BEQ     8$              ;YES, BRANCH
3446  010456  012737  000204  001162        MOV     #204,$REG0      ;GET EXPCTD RKCS
3447  010464  011337  001164                MOV     @R3,$REG1       ;GET RKCS RECVD
3448  010470  104024                        ERROR   24              ;RKCS SHOULD STILL CONTAIN THE 'READ'
3449                                                                  ;FUNCTION BITS
3450  010472  104413                8$:     CNT.RESET               ;GO ,DO CONTROL RESET
3451                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
3452                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
3453                                                                  ;ISSUED AND AFTER A CERTAIN TIME
3454                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
3455                                                                  ;AN ERROR IS REPORTED. NOTE THAT
3456                                                                  ;THE PC IN ERROR MESSAGE IS THE
3457                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
3458                                                                  ;THIS IS A VERY BASIC ERR & IF IT
3459                                                                  ;OCCURS GO BACK TO TEST 10
3460  010474  005777  170642                TST     @RKDB           ;DID CNTRL RESET CLEAR RKDB?
3461  010500  001407                        BEQ     TST23           ;;YES, EXIT
3462  010502  013737  001342  001162        MOV     RKDB,$REG0      ;GET ADRES OF RKDB
3463  010510  017737  170626  001164        MOV     @RKDB,$REG1     ;GET CONTENTS OF RKDB
3464  010516  104102                        ERROR   102             ;CONTROL RESET DID NOT
3465                                                                  ;CLEAR RKDB
3466
3467
3468                          ;;*****************************************************************
3469                          ;*TEST 23       CHECK 'WRITE FORMAT' OF THE DISK
3470                                  ;*THIS TEST WRITE FORMATS THE ENTIRE DISK.  THE FIRST
3471                                  ;*WORD OF EVERY SECTOR IS WRITTEN TO BE A PSUEDO-HEADER
3472                                  ;*CONSISTING OF THE DRIVE #, CYLINDER #, SURFACE & SECTOR #.
3473                                  ;*1 SECTOR IS WRITTEN AT A TIME. THE WRITING IS DONE
3474                                  ;*IN THIS ORDER: CYL 0-SUR 0; CYL 0-SUR 1;  CYL 1-SUR 0
3475                                  ;*CYL 1-SUR 1; CYL 2-SUR 0; CYL 2-SUR 1----- CYL 312-SUR 1.
3476                                  ;*IMPORTANCE OF THIS TEST SHOULD BE REALIZED, THIS IS
3477                                  ;*THE FIRST TIME EACH & EVERY SECTOR ON THE DISK IS
3478                                  ;*ACCESSED & WRITTEN ON.  THIS IS THE FIRST TIME RKDA
3479                                  ;*IS BEING MADE TO INCREMENT OVER THE ENTIRE DISK (FROM
3480                                  ;*000000 TO 014520) IF A 'SIN' OCCURS AT ANY POINT
3481                                  ;*A DRIVE RESET IS DONE BEFORE DOING WRT FMT FOR THE NEXT
3482                                  ;*SECTOR.  ANY OTHER ERROR IS CLEARED THROUGH A CONTROL RESET.
3483                                  ;*THE FOLLOWING CHECKING IS DONE AFTER WRITING EACH
3484                                  ;*CYLINDER.
3485                                  ;*1. CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
3486                                  ;*OF THE FUNCTION.
3487                                  ;*2. IF 'SIN' OCCURRED?
3488                                  ;*3. IF 'HE' OR 'ERR' BIT SET?
3489                                  ;*4. IF RKDA INCREMENTED CORRECTLY, INCLUDING BOUNDARY
3490                                  ;*CONDITIONS (SECTOR COUNTER BITS OVERFLOWING INTO SURFACE.
3491                                  ;*SURFACE BIT OVERFLOWING INTO CYLINDER BITS) AT THE END
3492                                  ;*OF THIS POINTERS ARE INCREMENTED ADJUSTED, ETC.
3493                                  ;*& 'WRT FMT' ON THE NEXT SECTOR IS DONE.
```

```
3494                                    ;;*****************************************************************
3495  010520  000004                    $TST23:  SCOPE
3496  010522  012737  000001  001206             MOV     #1,$TIMES       ;;DO 1 ITERATION
3497  010530  012737  010560  001110             MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
3498                                              ;ON  ERROR (SW 9)
3499  010536  005003                             CLR     R3              ;(R3)=0, SURFACE 0 BEING WRITTEN
3500                                              ;(R3)-1, SURFACE 1 BEING WRITTEN
3501  010540  012704  177465                     MOV     #-313,R4        ;SET UP COUNT FOR 203 CYLINDERS
3502  010544  012702  177764                     MOV     #-14,R2         ;SET UP COUNT FOR 12 SECTORS
3503  010550  013701  001350                     MOV     DRIVAD,R1       ;GET DRIVE ADRES
3504  010554  010105                             MOV     R1,R5           ;STORE IT
3505  010556  005205                             INC     R5
3506  010560  104413                    1$:      CNT.RESET               ;GO, DO CONTROL RESET
3507                                              ;THIS IS A CALL FOR THE 'CNTRL-
3508                                              ;RESET' ROUTINE. A CONTROL RESET IS
3509                                              ;ISSUED AND AFTER A CERTAIN TIME
3510                                              ;IF THE 'CNTRL RDY' DOES NOT SET
3511                                              ;AN ERROR IS REPORTED. NOTE THAT
3512                                              ;THE PC IN ERROR MESSAGE IS THE
3513                                              ;PC WHERE 'CNT.RESET' IS LOCATED.
3514                                              ;THIS IS A VERY BASIC ERR & IF IT
3515                                              ;OCCURS GO BACK TO TEST 10
3516  010562  104421                             TST.SIN                 ;GO CHECK IF SIN IS SET
3517                                              ;IF SET, DO DRIVE RESET TO CLR IT
3518  010564  005037  001362            7$:      CLR     COUNT
3519  010570  010137  033336                     MOV     R1,OUTBUF       ;THIS WORD TO BE WRITTEN.  THE FIRST
3520                                              ;WORD OF EACH SECTOR WILL BE THE ACTUA-
3521                                              ;DISK-ADRES, CONSISTING OF THE DRIVE NO.
3522                                              ;CYL ADRES, SURFACE BIT SECTOR ADRES
3523  010574  012777  033336  170534             MOV     #OUTBUF,@RKBA   ;ADRES FROM WHICH WORD IS TO B X-FERRED
3524  010602  012777  177777  170524             MOV     #-1,@RKWC       ;SET UP WORD COUNT
3525  010610  010177  170524                     MOV     R1,@RKDA        ;ADRES THE DRIVE, WITH CORRECT CY-
3526                                              ;& SECTOR ADRES
3527  010614  012777  002003  170510             MOV     #2003,@RKCS     ;WRITE FORMAT, GO
3528
3529  010622  105777  170504           2$:      TSTB    @RKCS           ;DID CNTRL RDY SET
3530  010626  100411                             BMI     3$              ;YES, BRANCH
3531  010630  005237  001362                     INC     COUNT           ;NO, HAVE U WAITED LONG ENOUGH?
3532  010634  001372                             BNE     2$              ;IF NOT, LOOP BACK & WAIT
3533                                              ;IF YES, REPORT ERROR
3534  010636  004737  020770                     JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
3535  010642  010137  001202                     MOV     R1,$REG10       ;GET DISK ADRES, WHERE ERROR OCCURED
3536  010646  104416                             BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
3537                                              ;$REG10 INTO DR #,CYL,SUR,SEC BITS
3538  010650  104031                             ERROR   31              ;CNTRL RDY DID NOT SET ON COMPLETION
3539                                              ;OF 'WRITE FORMAT'  ON SECTOR AS
3540                                              ;SHOWN IN <DSK-ADRES>
3541                                              ;WRT FMT WAS DONE STARTING AT <DSK-ADRES>
3542                                              ;INDICATED IN EROR MSGE.
3543  010652  032777  001000  170446     3$:      BIT     #1000,@RKDS     ;DID SIN BIT SET?
3544  010660  001405                             BEQ     4$              ;NO, BRANCH
3545  010662  004737  020776                     JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
3546  010666  010137  001170                     MOV     R1,$REG3        ;GET, DISK-ADRES WHERE ERROR OCCURED
3547  010672  104001                             ERROR   1               ;SIN SET WHILE DOING WRT FMT
3548                                              ;TO DISK-ADRES (AS IN $REG3)
3549
```

```
3550  010674  004737  021222          4$:     JSR     PC,CHKHE1       ;CHECK IF 'ERR' OR 'HE' BIT IS SET
3551                                                                    ;IF YES, RETURN HERE.
3552  010700  104032                          ERROR   32              ;HE OR ERR SET WHILE DOING WRITE
3553                                                                    ;FORMAT ON SECTOR AS INDICATED IN
3554                                                                    ;<DSK-ADRES>
3555                                                                    ;WRT FMT WAS DONE STARTING AT <DSK-ADRES>
3556                                                                    ;INDICATED IN EROR MSGE.
3557  010702  004737  021264          5$:     JSR     PC,CHKDA1       ;CHECK IF RKDA INCREMENTED CORRECTLY,
3558                                                                    ;IF NOT, RETURN HERE.
3559  010706  104033                          ERROR   33              ;RKDA DID NOT INCREMENT CORRECTLY
3560                                                                    ;AFTER 'WRITE FORMAT' WAS DONE
3561                                                                    ;TO THE SECTOR PREVIOUS TO THAT
3562                                                                    ;INDICATED IN 'EXPCTD' RKDA
3563  010710  005201          6$:     INC     R1              ;INCREMENT TO THE NXT SECTOR
3564  010712  005205                  INC     R5              ;INCREMENT R5, TO WHAT RKDA WILL INCREMENT
3565  010714  022702  177776          CMP     #-2,R2          ;R U GOING TO FORMAT THE LAST SECTOR
3566                                                          ;IN THE CYLINDER ?
3567  010720  001002                  BNE     .+6             ;IF NOT, BRANCH
3568  010722  062705  000004          ADD     #4,R5           ;INCREMENT R5 CORRECTLY TO 'EXPCTD RKDA'
3569  010726  005202                  INC     R2              ;HAVE U FORMATTED ALL 12 SECTORS
3570                                                          ;ON THIS CYLINDER
3571  010730  001313                  BNE     1$              ;IF NOT, LOOP BACK & FORMAT THE
3572                                                          ;NEXT SECTOR
3573                                                          ;YES
3574  010732  012702  177764          MOV     #-14,R2         ;RESET THE COUNT FOR 12 SECTORS
3575  010736  042701  000037          BIC     #37,R1          ;CLEAR THE SEC ADRES BITS
3576  010742  005703                  TST     R3              ;SURFACE 1?
3577  010744  001006                  BNE     8$              ;YES, BRANCH
3578  010746  005203                  INC     R3              ;NO, SET FLAG
3579  010750  062701  000020          ADD     #20,R1          ;INCREMENT TO THE NXT SURFACE
3580  010754  010105                  MOV     R1,R5           ;THIS IS WHAT RKDA SHOULD
3581  010756  005205                  INC     R5              ;INCREMENT TO.
3582  010760  000677                  BR      1$              ;GO, DO NXT SURFACE
3583  010762  062701  000040          8$:     ADD     #40,R1          ;INCREMENT TO NXT CYL
3584  010766  010105                  MOV     R1,R5           ;POSITION FOR
3585  010770  005205                  INC     R5              ;EXPCTD RKDA
3586  010772  005003                  CLR     R3
3587  010774  005204                  INC     R4              ;HAVE U FORMATTED ALL 203 CYLINDERS
3588  010776  001270                  BNE     1$              ;IF NOT, LOOP BACK & FORMAT THE
3589                                                          ;NEXT CYLINDER
3590
3591
3592
3593                          ;;*****************************************************************
3594                          ;*TEST 24       CHECK 'READ FORMAT' FOR THE ENTIRE DISK
3595                                  ;*THIS TEST READ FORMATS THE ENTRE DISK, WHICH WAS WRT
3596                                  ;*FORMATTED IN THE PREVIOUS TEST.  THE FOLLOWING CHECKING
3597                                  ;*IS DONE
3598                                  ;*1. CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
3599                                  ;*OF FUNCTION
3600                                  ;*2. IF 'SIN' OCCURRED?
3601                                  ;*3. IF 'HE' OR 'ERR' OCCURRED?
3602                                  ;*4. RKDA INCREMENTED CORRECTLY.
3603                                  ;*5. IF THE CORRECT HEADER WAS READ.
3604                                  ;*6. IF RKWC OVERFLOWED CORRECTLY.
3605                                  ;*12 SECTORS (1 CYLINDER) ARE READ AT A TIME. IF 'SIN'
```

```
                                                ;*OCCURS A DRIVE RESET IS DONE BEFORE READING THE NEXT
                                                ;*SECTOR.  READING IS DONE IN THIS ORDER CYL 0-SUR 0;
                                                ;*CYL 0-SUR 1; CYL 1-SUR 0; CYL 1-SUR 1; CYL 2-SUR 0;
                                                ;*CYL 2-SUR 1;-----CYL 312-SUR 1. IF TESTING ON SIMULATOR, ONLY
                                                ;*THE LAST CYLINDER (312), LAST SECTOR (13), SURFACE 1 IS READ.
                                        ;;**************************************************************
011000  000004                        TST24:  SCOPE
011002  012737  000001  001206                MOV     #1,$TIMES       ;;DO 1 ITERATION
011010  012737  011074  001110                MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
                                                                        ;ON      ERROR (SW 9)
011016  005037  001356                        CLR     INDX1           ;INDX1=0, SURFACE 0 BEING READ
                                                                        ;INDX1=1, SURFACE 1 BEING READ
011022  013701  001350                        MOV     DRIVAD,R1       ;GET DRIVE ADRES
011026  010102                                MOV     R1,R2
011030  005737  001344                        TST     SIMUL           ;TESTING ON SIMULATOR?
011034  001410                                BEQ     12$             ;NO, BRANCH
011036  052701  014533                        BIS     #14533,R1       ;SET BITS FOR CYL 312, SEC 13, SUR 1
                                                                        ;ON SIMULATOR, CHECK ONLY CYL 312,
                                                                        ;SECTOR 13, SURFACE 1
011042  052702  014540                        BIS     #14540,R2       ;RKDA SHOULD INCRMNT TO THIS AFTR
                                                                        ;RD FMT OF 1 SECTOR
011046  012737  177777  001370                MOV     #-1,EFLG1       ;SET COUNT FOR READING HDR
                                                                        ;FROM 1 SECTOR ONLY
011054  000407                                BR      1$
011056  012705  177465                12$:    MOV     #-313,R5        ;SET UP COUNT FOR 203 CYLINDERS
011062  012737  177764  001370                MOV     #-14.EFLG1      ;SET COUNT FOR 12 HDRS TO BE
                                                                        ;READ FROM EACH CYLINDER
011070  062702  000020                        ADD     #20,R2          ;THIS IS WHAT RKDA SHOULD INCREMENT
                                                                        ;BY, AFTER 'RD FMT' OF EACH CYLINDER
011074  104413                        1$:     CNT.RESET               ;GO, DO CONTROL RESET
                                                                        ;THIS IS A CALL FOR THE 'CNTRL-
                                                                        ;RESET' ROUTINE. A CONTROL RESET IS
                                                                        ;ISSUED AND AFTER A CERTAIN TIME
                                                                        ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                        ;AN ERROR IS REPORTED. NOTE THAT
                                                                        ;THE PC IN ERROR MESSAGE IS THE
                                                                        ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                        ;THIS IS A VERY BASIC ERR & IF IT
                                                                        ;OCCURS GO BACK TO TEST 10
011076  104421                                TST.SIN                 ;CHECK IF SIN IS SET
                                                                        ;IF SET DO DRV-RESET TO CLR IT
011100  012703  033336                        MOV     #OUTBUF,R3      ;STORE ADRES OF BUFFER
011104  005037  001360                11$:    CLR     INDX2
011110  010377  170222                        MOV     R3,@RKBA        ;ADRES TO WHICH DATA IS TO BE X-FERRED
                                                                        ;FROM THE DISK
011114  013777  001370  170212                MOV     EFLG1,@RKWC     ;SET UP WORD COUNT FOR 12 HEADERS
                                                                        ;TO BE READ OFF EACH CYLINDER
                                                                        ;(ONLY 1 FOR SIMULATOR)
011122  010177  170212                        MOV     R1,@RKDA        ;ADRES THE DRIVE WITH CORRECT
                                                                        ;CYLINDER & SECTOR ADRES
011126  012777  002005  170176                MOV     #2005,@RKCS     ;READ FORMAT, GO
011134  105777  170172                2$:     TSTB    @RKCS           ;DID CNTRL RDY SET?
011140  100411                                BMI     3$              ;YES, BRANCH
```

```
                                                INC     INDX2           ;NO, HAVE U WAITED LONG ENOUGH?
                                                BNE     2$              ;IF NOT, LOOP BACK & WAIT FOR IT
                                                                        ;IF YES, REPORT ERROR
                                                JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
                                                MOV     R1,$REG10       ;GET DRIV-ADRES STARTING WHICH
                                                                        ;'READ FORMAT' WAS DONE
                                                BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                                        ;$REG10 INTO DR #,CYL,SUR,SEC BITS
                                                ERROR   45              ;CNTRL RDY DID NOT SET AFTER
                                                                        ;READ FORMAT. 'RKDA' IN EROR MSGE
                                                                        ;GIVES THE CONTENTS OF RKDA AT THE
                                                                        ;TIME OF ERROR.
                                                                        ;READ FMT WAS DONE STARTING AT <DSK-ADRES
                                                                        ;INDICATED IN EROR MSGE.

                                        170134  3$:     BIT     #1000,@RKDS     ;DID 'SIN' SET?
                                                        BEQ     4$              ;NO, BRANCH
                                                        JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
                                                        MOV     R1,$REG3        ;GET DISK-ADRES WHERE 'SIN'
                                                                                ;OCCURED
                                                ERROR   1               ;SIN ERROR ON DOING RD FMT
                                                                        ;TO CYL INDICATED IN $REG3

                                        021222  4$:     JSR     PC,CHKHE1       ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
                                                                                ;IF YES, RETURN HERE.
                                                ERROR   46              ;HE OR ERR WHILE DOING A READ
                                                                        ;FORMAT. 'RKDA' IN EROR MSGE GIVES
                                                                        ;THE CONTENTS OF RKDA AT THE TIME OF ERROR
                                                                        ;READ FMT WAS DONE STARTING AT <DSK-ADRES
                                                                        ;INDICATED IN EROR MESGE

                                        170120  5$:     CMP     R2,@RKDA        ;DID RKDA INCREMENT CORRECTLY BY 12 SEC
                                                        BEQ     6$
                                                        MOV     R2,$REG10       ;GET EXPCTD RKDA
                                                        BRKDA0                  ;GO TO 'BDA0' & BREAK  CONTENTS OF
                                                                                ;$REG10 INTO DR #,CYL,SUR,SEC BITS

                                        170104  001202          MOV     @RKDA,$REG10    ;GET RECVD RDKA
                                                        BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                                                ;$REG10 INTO DR #,CYL,SUR,SEC BITS
                                                ERROR   40              ;RKDA DID NOT INCREMENT BY 12 SECTORS
                                                                        ;AFTER RD FMT WAS DONE. ADRES
                                                                        ;OF CYLINDER IN ERROR CAN BE OBTAINED
                                                                        ;FROM 'EXPCTD' RDDA

                                        001370  6$:     MOV     EFLG1,R0        ;SET UP COUNT FOR 12 HEADERS TO B CHKD
                                                                                ;(ONLY 1, IF SIMULATOR)
                                                        MOV     R1,R4           ;GET DRIV-ADRES FROM WHERE RDFMT WAS DONE
                                                        BIC     #160037,R4      ;GET THE CYLINDER ADRES ONLY. (HEADER)
                                        7$:             CMP     R4,(R3)         ;IS THE RECVD HEADER SAME AS EXPCTD?
                                                        BEQ     8$
                                                        MOV     R4,$REG1        ;GET EXPCTD HEADER WORD
                                                        MOV     (R3),$REG2      ;GET HEADER WORD RECVD
                                                        MOV     R0,$REG0
                                                        ADD     #14,$REG0       ;GET THE SECTOR (OCTAL NO) WHICH DID
                                                                                ;NOT GIVE THE CORRECT HEADER
                                                ERROR   43              ;DID NOT RECIEVE THE CORRECT HEADER
                                                                        ;WORD FROM 'SECTOR' AS INDICATED
                                                                        ;(NOTE SECTOR # IS OCTAL)
```

```
3718  011304  005723                  8$:     TST     (R3)+           ;INCREMENT POINTER TO THE NXT WORD
3719                                                                  ;IN MEMORY WHERE THE RECVD HDR IS STORED
3720  011306  005200                          INC     R0              ;HAVE U CHECKED ALL 12 HEADERS?
3721  011310  001361                          BNE     7$              ;IF NOT, LOOP BACK & CHK THE NXT.
3722                                                                  ;YES, ALL HEADERS FOR THIS CYLINDER
3723                                                                  ;CHECKED.
3724  011312  004737  021312                  JSR     PC,CHKWC        ;CHECK IF RKWC OVERFLOWED TO 0, IF
3725                                                                  ;NOT RETURN HERE.
3726  011316  104041                          ERROR   41              ;RKWC DID NOT OVERFLOW AFTER DOING
3727                                                                  ;RDFMT OF 12 SECTORS ON THE CYLINDER
3728                                                                  ;NOTE THAT 'RKDA' IS THE INCREMENTED
3729                                                                  ;RKDA AFTER THE RDFMT
3730  011320  005737  001344          9$:     TST     SIMUL           ;TSTING ON SIMULATOR?
3731  011324  001031                          BNE     TST25           ;;IF YES, EXIT
3732                                                                  ;NO
3733  011326  005737  001356                  TST     INDX1           ;DOING SURFACE 1
3734  011332  001011                          BNE     10$             ;YES, BRANCH
3735  011334  005237  001356                  INC     INDX1           ;NO
3736  011340  062701  000020                  ADD     #20,R1          ;INCREMENT DRIV ADRES TO THE NXT SURFACE
3737  011344  010102                          MOV     R1,R2
3738  011346  062702  000020                  ADD     #20,R2          ;THIS IS WHAT RKDA SHOULD INCREMENT
3739                                                                  ;TO, AFTER READ FMT OF THE CYLINDER
3740  011352  000137  011074                  JMP     1$              ;GO RD FMT THE NXT SURFACE
3741  011356  005037  001356          10$:    CLR     INDX1           ;
3742  011362  042701  000037                  BIC     #37,R1          ;CLR SEC, SURFACE BITS
3743  011366  062701  000040                  ADD     #40,R1          ;INCREMENT TO NXT CYL
3744  011372  010102                          MOV     R1,R2           ;THIS IS WHAT RKDA SHOULD BE
3745  011374  062702  000020                  ADD     #20,R2          ;AFTER RD FMT OF CYLINDER
3746  011400  005205                          INC     R5              ;HAVE U DONE ALL CYLINDERS?
3747  011402  001402                          BEQ     TST25           ;;EXIT
3748  011404  000137  011074                  JMP     1$              ;IF NOT, LOOP BACK & READ FMT FROM
3749                                                                  ;THE NXT CYLINDER
3750
3751
3752                                  ;;**************************************************************
3753                                  ;*TEST 25       CHECK 'READ' OF THE ENTIRE DISK
3754                                          ;*READ OF THE ENTIRE DISK (ONE WORD PER SECTOR) IS DONE
3755                                          ;*IN THIS TEST.  IN A PREVIOUS TEST THE FIRST WORD OF
3756                                          ;*EVERY SECTOR WAS WRITTEN LIKE A PSUEDO-HEADER (DRIVE #,
3757                                          ;*CYLINDER #, SURFACE & SECTOR #).  THESE PSUEDO HEADERS
3758                                          ;*WILL BE READ & CHECKED IN THIS TEST, PROVING THAT ANY
3759                                          ;*SECTOR CAN BE ACCESSED AND READ.
3760                                          ;*THE FOLLOWING CHECKING IS DONE
3761                                          ;*1. CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
3762                                          ;*OF FUNCTION.
3763                                          ;*2. IF 'SIN' OCCURRED?
3764                                          ;*3. IF 'HE' OR 'ERR' OCCURRED?
3765                                          ;*4. THE CORRECT FIRST WORD FROM EVERY SECTOR
3766                                          ;*WAS RECEIVED.  THIS WORD REFLECTS THE ABSOLUTE
3767                                          ;*DISK ADDRESS (DRV #, CYL #, SUR, SEC#) OF THAT SECTOR.
3768                                          ;*5. IF RKDB CONTAINED THE CORRECT WORD.
3769                                          ;*IF 'SIN' OCCURS DRIVE RESET IS DONE BEFORE READING
3770                                          ;*THE NEXT SECTOR.  READ IS DONE IN THIS ORDER SEC 0-11
3771                                          ;*CYL 0 SUR 0 -> SEC 0-11 CYL 0 SUR 1 -> SEC 0-11 CYL 1,....
3772                                          ;*IF TESTING ON SIMULATOR ONLY LAST CYLINDER (312), LAST
3773                                          ;*SECTOR (13), SURFACE 1 IS READ.
```

```
                                ;;*****************************************************************
011410  000004                 TST25:  SCOPE
011412  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
011420  012737  011464  001110          MOV     #1$,$LPERR      ;SET RETURN ADRES FOR
                                                                ;LOOPING ON ERROR (SW9)
011426  012703  033336                  MOV     #OUTBUF,R3
011432  005004                          CLR     R4              ;FLAG, CLEAR WHEN READING SURFACE 0
                                                                ;SET WHEN READING SURFACE 1
011434  013701  001350                  MOV     DRIVAD,R1       ;GET DRIVE ADDRESS
011440  005737  001344                  TST     SIMUL           ;TSTING ON SIMULATOR?
011444  001403                          BEQ     10$             ;IF NOT BRANCH
011446  052701  014533                  BIS     #14533,R1       ;SET ADRES BITS FOR LAST CYL (312)
011452  000404                          BR      1$              ;LAST SECTOR (13), SURFACE 1
011454  012700  177764          10$:    MOV     #-14,R0         ;SET COUNT FOR 12 SECTORS
011460  012705  177465                  MOV     #-313,R5        ;SET UP COUNT FOR 203 CYLINDERS

011464  104413                  1$:     CNT.RESET               ;GO, DO CONTROL RESET
                                                                ;THIS IS A CALL FOR THE 'CNTRL-
                                                                ;RESET' ROUTINE. A CONTROL RESET IS
                                                                ;ISSUED AND AFTER A CERTAIN TIME
                                                                ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                ;AN ERROR IS REPORTED. NOTE THAT
                                                                ;THE PC IN ERROR MESSAGE IS THE
                                                                ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                ;THIS IS A VERY BASIC ERR & IF IT
                                                                ;OCCURS GO BACK TO TEST 10

011466  104421                          TST.SIN                 ;GO CHECK SIN, IF SET DO
                                                                ;DRIVE RESET TO CLR IT

011470  005037  001356          8$:     CLR     INDX1
011474  010377  167636                  MOV     R3,@RKBA        ;ADRES TO WHICH DATA IS TO B X-FERRED
                                                                ;FROM THE DISK
011500  012777  177777  167626          MOV     #-1,@RKWC       ;SET UP WORD COUNT
011506  010177  167626                  MOV     R1,@RKDA        ;ADRES THE DRIVE WITH CORRECT
                                                                ;CYLINDER & SECTOR ADRES
011512  012777  000005  167612          MOV     #5,@RkCS        ;READ, GO

011520  105777  167606          2$:     TSTB    @RkCS           ;DID CNTRL RDY SET?
011524  100411                          BMI     3$              ;YES, BRANCH
011526  005237  001356                  INC     INDX1           ;NO, HAVE U WAITED LONG ENOUGH
011532  001372                          BNE     2$              ;IF NOT, LOOP BACK & WAIT FOR IT
                                                                ;IF YES, REPORT ERROR
011534  004737  020770                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
011540  010137  001202                  MOV     R1,$REG10       ;GET DISK-ADRES WHERE ERROR OCCURED
011544  104416                          BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                                ;$REG10 INTO DR #,CYL,SUR,SEC BITS
011546  104045                          ERROR   45              ;CNTRL RDY DID NOT SET AFTER DOING
                                                                ;A 1 WORD READ FROM ADRES AS
                                                                ;INDICATED IN <DISK-ADRES>
                                                                ;'RKDA' IN EROR MSGE GIVES THE
                                                                ;CONTENTS OF RKDA AT THE TIME OF ERROR

011550  032777  001000  167550  3$:     BIT     #1000,@RkDS     ;DID 'SIN' SET?
011556  001405                          BEQ     4$              ;NO, BRANCH
011560  004737  020776                  JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
011564  010137  001170                  MOV     R1,$REG3        ;GET DISK-ADRES WHERE SIN OCCURED3
011570  104001                          ERROR   1               ;'SIN' ERROR ON DOING READ FROM
```

```
                                                        ;DISK-ADRES INDICATED IN $REG3
                                                        ;CHECK IF 'ERR' OR 'HE' BIT IS SET.
                                                        ;IF YES, RETURN HERE.
011576  104046                  ERROR   46             ;'HE' OR 'ERR' ON DOING A READ OF
                                                        ;1 WORD FROM ADRES AS INDICATED
                                                        ;IN (DISK-ADRES)
                                                        ;'RKDA' IN ERCR MSGE GIVES THE
                                                        ;CONTENTS OF RKDA AT THE TIME OF ERCR
011600  020113          5$:     CMP     R1,(R3)        ;WAS THE CORRECT DATA WORD RECVD?
011602  001407                  BEQ     6$
011604  010137  001162          MOV     R1,$REG0       ;GET EXPCTD DATA WORD
011610  011337  001164          MOV     (R3),$REG1     ;GET DATA WORD RECVD
011614  010137  001166          MOV     R1,$REG2       ;GET DISK-ADRES
011620  104044                  ERROR   44             ;DID NOT RECIEVE THE CORRECT
                                                        ;DATA WORD FROM DISK ON DOING
                                                        ;1 WORD READ FROM 'DISK-ADRES'
                                                        ;AS INDICATED BY 'EXPCTD' DATA WORD
                                                        ;NOTE THAT IN A PREVIOUS TEST THE
                                                        ;FIRST WORD OF EACH SECTOR IS UNIQUELY
                                                        ;WRITTEN WITH A WORD GIVING THE
                                                        ;ABSOLUTE ADDRESS OF THAT SECTOR IN
                                                        ;TERMS OF, DRIV #, CYL ADRES, SUR, SEC ADRS.
011622  020177  167514          6$:     CMP     R1,@RKDB       ;DOES RKDB CONTAIN CORRECT WORD
011626  001406                  BEQ     7$             ;YES, BRANCH
011630  010137  001162          MOV     R1,$REG0       ;NO, GET EXPCTD RKDB
011634  017737  167502  001164  MOV     @RKDB,$REG1    ;GET RKDB RECVD
011642  104037                  ERROR   37             ;RKDB ERROR ON READ.
                                                        ;FOR RK11C, AFTER A READ RKDB
                                                        ;CONTAINS CHECKSUM FOR THE SECTOR
                                                        ;READ.
                                                        ;WHEREAS FOR RK11D, AFTER READ
                                                        ;RKDB CONTAINS THE LAST WORD
                                                        ;READ FROM THAT SECTOR $
                                                        ;X-FERRED TO MEMORY
011644  005737  001344          7$:     TST     SIMUL          ;TESTING ON SIMULATOR?
011650  001022                  BNE     TST26          ;;IF YES, EXIT
011652  005201                  INC     R1             ;INCREMENT  TO ADRES NEXT SECTOR
011654  005200                  INC     R0             ;HAVE U CHKD ALL 12 SECTORS?
011656  001302                  BNE     1$             ;IF NOT, LUP BAK & CHK THE NXT
                                                        ;IF YES...
011660  012700  177764          MOV     #-14,R0        ;RESET THE COUNT FOR 12 SECTORS
011664  042701  000037          BIC     #37,R1         ;CLEAR SECTOR, SURFACE BITS
011670  005704                  TST     R4             ;DOING SURFACE 1?
011672  001004                  BNE     9$             ;YES, BRANCH
011674  005204                  INC     R4             ;NO
011676  062701  000020          ADD     #20,R1         ;INCREMENT THE ADRES TO NXT SURFACE
011702  000670                  BR      1$             ;GO READ SURFACE 1
011704  005004          9$:     CLR     R4
011706  062701  000040          ADD     #40,R1         ;INCREMENT TO NXT CYL
011712  005205                  INC     R5             ;HAVE U CHKD ALL 203 CYLINDERS
011714  001263                  BNE     1$             ;IF NOT, LOOP BACK & CHK THE NXT CYLINDER
                                                        ;YES


                        ;;*****************************************************************
                        ;*TEST 26        CHECK 'SEEK' FUNCTION, WITH DIFFERENT VELOCITY MODES
```

```
                                        ;* THIS TEST CHECKS SEEK IN DIFFERENT VELOCITY MODES (DIFF < 3
                                        ;* 3 < DIFF < 31, DIFF > 31).  FOR THESE 3 BASIC VELOCITIES SEEK IS DONE BOT-
                                        ;* IN FWD AND REV DIRECTION TO CHECK THE ADDER & DIFFERENCE LOGIC.  IF
                                        ;* WHILE DOING A SEEK 'SIN' OCCURS, A DRIVE RESET IS DONE TO INITIALIZE
                                        ;* THE POSITIONING LOGIC
                                    ;;**********************************************************************
011716  000004                    T$T26:  SCOPE
011720  012737  000005  001206            MOV     #5,$TIMES       ;;DO 5 ITERATIONS
011726  012703  001372                    MOV     #SEEK0,R3       ;INITIALIZE POINTER TO THE FIRST
                                                                  ;SEEK ADDRESS
011732  005037  001356                    CLR     INDX1           ;INDX1, WHEN 0 INDICATES SEEK IN FWD DIRECTION
                                                                  ;       WHEN 1 INICATES SEEK IN REV DIRECTION
011736  013700  001332                    MOV     RKCS,R0
011742  013701  001326                    MOV     RKDS,R1
011746  013702  001330                    MOV     RKER,R2
011752  012737  011760  001110            MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING ON
                                                                  ;EROR (SW 9)
011760  000240                    1$:     NOP
011762  104413                    2$:     CNT.RESET               ;GO, DO CONTROL RESET
                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
                                                                  ;ISSUED AND AFTER A CERTAIN TIME
                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                  ;AN ERROR IS REPORTED. NOTE THAT
                                                                  ;THE PC IN ERROR MESSAGE IS THE
                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                  ;THIS IS A VERY BASIC ERR & IF IT
                                                                  ;OCCURS GO BACK TO TEST 10
011764  104421                            TST.SIN                 ;GO, CHECK IF SIN IS SET, IF SET
                                                                  ;DO DRV-RESET TO CLEAR IT


011766  013704  001350                    MOV     DRIVAD,R4       ;GET DRIV-ADRES
011772  051304                            BIS     (R3),R4         ;SET CYLINDER BITS
011774  010477  167340                    MOV     R4,@RKDA        ;ADDRS THE DRIVE
012000  012710  000011                    MOV     #11,@R0         ;SET 'SEEK', 'GO'

012004  104412                            CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
                                                                  ;IF SO, SKIP THE EROR MESSAGE.
012006  104021                            ERROR   21              ;'CNTRL RDY' DID NOT SET AFTER
                                                                  ;SENDING CYL ADD TO THE DRIV, 'ADD ACK'
                                                                  ;FROM DRIVE SHLD HAVE COME BACK
                                                                  ;THEREUPON SETTING 'CNTRL RDY'
012010  005005                    4$:     CLR     R5
012012  032711  000100            5$:     BIT     #100,@R1        ;DID R/W/S RDY SET?
012016  001005                            BNE     6$              ;YES, BRANCH
012020  005205                            INC     R5              ;NO, WAIT
012022  001373                            BNE     5$              ;WAITED LONG?
012024  004737  020770                    JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
012030  104026                            ERROR   26              ;R/W/S RDY DID NOT SET ON
                                                                  ;COMPLETION OF SEEK
012032  032711  001000            6$:     BIT     #1000,@R1       ;DID SIN SET?
012036  001403                            BEQ     7$              ;NO, BRANCH
012040  004737  020770                    JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
012044  104001                            ERROR   1               ;SIN SET ON DOING SEEK
012046  032710  140000            7$:     BIT     #140000,@R0     ;DID 'HE' OR 'ERR' SET?
```

```
3942  012052  001403                        BEQ     8$              ;YES
3943  012054  004737  020770                JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
3944  012060  104022                        ERROR   22              ;'ERR OF 'HE' BIT SET WHEN
3945                                                                 ;SEEKING TO CYL AS INDICATED
3946                                                                 ;IN RKDA
3947
3948  012062  022710  000210          8$:   CMP     #210,@R0        ;DOES RKCS STILL CONTAIN THE 'SEEK' FNCTION
3949  012066  001406                        BEQ     9$              ;YES - EXIT
3950  012070  011037  001164                MOV     @R0,$REG1       ;NO, GET RKCS RECVD
3951  012074  012737  000210  001162        MOV     #210,$REG0      ;GET EXPCTD RKCS
3952  012102  104024                        ERROR   24              ;RKCS SHOULD CONTAIN THE 'SEEK' BITS
3953                                                                 ;IF NOT, ERROR
3954
3955  012104  020477  167230          9$:   CMP     R4,@RKDA        ;DID RKDA CHANGE?
3956  012110  001406                        BEQ     10$             ;NO
3957  012112  010437  001162                MOV     R4,$REG0        ;YES, GET EXPCTD?
3958  012116  017737  167216  001164        MOV     @RKDA,$REG1     ;GET RKDA
3959  012124  104027                        ERROR   27              ;RKDA CHANGED AFTER DOING SEEK
3960
3961  012126  010477  167206          10$:  MOV     R4,@RKDA        ;ADRES THE DRIVE,SEC 0
3962  012132  012777  033336  167176        MOV     #OUTBUF,@RKBA   ;READ ONE HEADER INTO THIS
3963  012140  012777  177777  167166        MOV     #-1,@RKWC       ;BUS ADRES
3964  012146  012710  002005                MOV     #2005,@R0       ;GO,READ FORMAT
3965  012152  104414                        CNT.RDY                 ;WAIT FOR CNTRL RDY
3966  012154  021337  033336                CMP     (R3),OUTBUF     ;WAS THE CORRECT READER READ (FROM
3967  012160  001410                        BEQ     11$             ;CYLINDER TO WHICH SEEK WAS DONE BEFORE,
3968  012162  005037  001162                CLR     $REG0           ;STORE SEC # FROME  WHERE HDR WAS RD .0)
3969  012166  011337  001164                MOV     (R3),$REG1      ;GET EXPCTD HEADER
3970  012172  013737  033336  001166        MOV     OUTBUF,$REG2    ;GET HDR RECVD
3971  012200  104043                        ERROR   43              ;WRONG HDR WAS RECVD FROM CYLINDER (ADRES
3972                                                                 ;IN ER MSGE). NOTE THAT A PURE SEEK WAS
3973                                                                 ;DONE TO THIS CYL BEFORE  READING HDR
3974                                                                 ;USING READ FORMAT
3975  012202  005737  001356          11$:  TST     INDX1           ;SEEK IN REVRSE DIRECTION?
3976  012206  001007                        BNE     12$             ;YES, BRANCH
3977  012210  005723                        TST     (R3)+           ;NO, INCREMENT PTR TO NXT SEEK ADRES
3978  012212  022703  001400                CMP     #SEEK2+2,R3     ;DONE WITH ALL SKS IN FWD DIR?
3979  012216  001260                        BNE     1$              ;NO, GO & DO NXT ONE
3980  012220  005237  001356                INC     INDX1           ;SET FLAG INDICATING SK IN REVRSE
3981  012224  005743                        TST     -(R3)
3982  012226  005743                  12$:  TST     -(R3)           ;POSITION PTR TO NXT SK IN REV
3983  012230  022703  001370                CMP     #SEEK0-2,R3     ;DONE WITH ALL?
3984  012234  001251                        BNE     1$              ;IF NOT, DO NXT ONE
3985
3986
3987
3988                                  ;;*****************************************************************
3989                                  ;*TEST 27       CHECK DRIVE RESET FROM LAST CYLINDER
3990                                          ;*THE HEADS ARE POSITIONED ON THE LAST CYLINDER (DOING
3991                                          ;*AN IMPLIED SEEK-READ).  THEN A DRIVE RESET IS ISSUED.
3992                                          ;*IT'S CHECKED IF THE HEADS WERE BROUGHT BACK TO 0 BY
3993                                          ;*DOING A 1 WORD READ & CHECKING THAT THE CORRECT WORD
3994                                          ;*WAS RECEIVED. IF TESTING ON SIMULATOR THIS TEST IS SKIPPED.
3995                                  ;;*****************************************************************
3996  012236  000004                  TST27:  SCOPE
3997  012240  012737  000005  001206          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
```

```
                                                ISI     SIMUL           ;R U ON A SIMULATOR?
                                                BNE     TST30           ;;YES, EXIT
                                                MOV     RKCS,R1
                                                CNT.RESET               ;GO, DO CONTROL RESET
                                                                        ;THIS IS A CALL FOR THE 'CNTRL-
                                                                        ;RESET' ROUTINE. A CONTROL RESET IS
                                                                        ;ISSUED AND AFTER A CERTAIN TIME
                                                                        ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                        ;AN ERROR IS REPORTED. NOTE THAT
                                                                        ;THE PC IN ERROR MESSAGE IS THE
                                                                        ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                        ;THIS IS A VERY BASIC ERR & IF IT
                                                                        ;OCCURS GO BACK TO TEST 10
                                                CLR     R0
                                                MOV     #OUTBUF,R3      ;ADRES WHERE DATA WILL BE READ INTO
                                                MOV     DRIVAD,R4
                                                MOV     R4,R5
                                                BIS     #14500,R5       ;SET CYL ADRES=312 (OCTAL)
                                                MOV     R5,@RKDA        ;ADRES THE DRIVE, LAST CYLINDER
                                                MOV     #-1,@RKWC       ;READ 1 WORD
                                                MOV     R3,@RKBA        ;INTO THIS MEMORY ADRES

                                                MOV     #5,@R1          ;READ, GO

                                                CLR     R0
                                        1$:     CNT.RDY                 ;THIS IS A CALL FOR CN.RDY ROUTINE
                                                                        ;WHICH WAITS FOR CNTRL RDY TO SET.
                                                                        ;A RETURN IS MADE AFTER CNTRL RDY
                                                                        ;SETS. IF WITHIN A CERTAIN TIME
                                                                        ;CNTRL RDY DOESN'T SET AN ERROR
                                                                        ;MESSAGE IS GIVEN. WAITING TIME
                                                                        ;883 MS FOR 11/20, 175 MS FOR 11/45
                                        2$:     CMP     R5,@R3          ;WAS THE CORRECT WORD READ?
                                                BEQ     3$              ;YES, SEEK TO 312 WAS DONE CORRECTLY.S
                                                MOV     R5,$REG0        ;GET EXPCTD WORD
                                                MOV     @R3,$REG1       ;GET WORD RECVD
                                                MOV     R5,$REG2        ;GET DSK-ADRES FROM WHERE WORD WAS READ
                                                ERROR   44              ;DID NOT READ BACK CORRECT WORD FROM
                                                                        ;LAST CYL, SEC 0. IF TEST 45 & 46
                                                                        ;WERE SUCCESSFULLY DONE THIS
                                                                        ;ERROR MEANS THAT IMPLIED SEEK
                                                                        ;TO CYL 312 COULD NOT B DONE
                                        3$:     MOV     #15,@R1         ;DRIVE RESET, GO
                                                CNT.RDY                 ;THIS IS A CALL FOR CN.RDY ROUTINE
                                                                        ;WHICH WAITS FOR CNTRL RDY TO SET.
                                                                        ;A RETURN IS MADE AFTER CNTRL RDY
                                                                        ;SETS. IF WITHIN A CERTAIN TIME
                                                                        ;CNTRL RDY DOESN'T SET AN ERROR
                                                                        ;MESSAGE IS GIVEN. WAITING TIME
                                                                        ;883 MS FOR 11/20, 175 MS FOR 11/45
                                                CLR     R0
                                        4$:     BIT     #100,@RKDS      ;DID R/W/S RDY SET?
                                                BNE     5$              ;YES, BRANCH
                                                MOV     #-15,R2         ;IF U R ON A SLOWER MACHINE
                                                INC     R2              ;& DO NOT NEED SUCH A LARGE MACHINE
                                                BNE     .-2             ;TIME LOOP, CHANGE THESE 3
```

```
4054                                                                    ;INSTRUCTIONS TO 'NOP' THE
4055                                                                    ;LOOP TIME WILL BE REDUCED
4056                                                                    ;TO 1100 MS
4057                                                                    .
4058                                                                    ;THE TOTAL TIME FOR THE ABOVE
4059                                                                    ;LOOPS (W/O PUTTING 'NOP'S) IS
4060                                                                    ;5304 MS FOR 11/20 AND
4061                                                                    ;1061 MS FOR 11/45 WITH MOS
4062                                                                    ;OR BIPOLAR MEMORY
4063  012402  005200                        INC     R0                 ;WAITED LONG?
4064  012404  001366                        BNE     4$                 ;IF NOT, LUP BAK & WAIT
4065                                                                    ;IF YES, ERROR
4066  012406  004737  020770                JSR     PC,GT4RG           ;GET RKCS,ER,DS,DA
4067  012412  104026                        ERROR   26                 ;R/W/S RDY DID NOT SET AFTER
4068                                                                    ;DOING DRIVE RESET
4069  012414  032711  140000        5$:     BIT     #140000,@R1        ;DID HE OR ERR BIT SET?
4070  012420  001403                        BEQ     6$                 ;IF NOT, BRANCH
4071
4072  012422  004737  020770                JSR     PC,GT4RG           ;GET RKCS,ER,DS,DA FOR ERROR MESSAGE
4073  012426  104022                        ERROR   22                 ;HE OR ERR BIT SET ON DOING DRIVE
4074                                                                    ;RESET FROM LAST CYLINDER
4075  012430  005205                6$:     INC     R5                 ;POSITION R5 TO EXPCTD RKDA
4076  012432  020577  166702                CMP     R5,@RKDA           ;DID THE CYL ADRES BITS IN RKDA GET CHANGED?
4077  012436  001406                        BEQ     7$                 ;NO, BRANCH
4078  012440  010537  001162                MOV     R5,$REG0           ;GET EXPCTD RKDA
4079  012444  017737  166670  001164        MOV     @RKDA,$REG1        ;GET RKDA RECVD
4080  012452  104054                        ERROR   54                 ;CYLINDER ADRES BITS IN RKDA
4081                                                                    ;GOT CHANGED AFTER
4082                                                                    ;DRIVE RESET, FROM LAST CYLINDER
4083  012454  012777  177777  166652  7$:     MOV     #-1,@RKWC          ;READ 1 WORD
4084  012462  010377  166650                MOV     R3,@RKBA           ;INTO THIS ADRES
4085  012466  010477  166646                MOV     R4,@RKDA           ;FROM THIS DSK ADRES-CYL 0, SEC 0
4086
4087  012472  012711  000005                MOV     #5,@R1             ;READ, GO
4088
4089  012476  005000                        CLR     R0
4090  012500  104414                8$:     CNT.RDY                    ;THIS IS A CALL FOR CN.RDY ROUTINE
4091                                                                    ;WHICH WAITS FOR CNTRL RDY TO SET.
4092                                                                    ;A RETURN IS MADE AFTER CNTRL RDY
4093                                                                    ;SETS. IF WITHIN A CERTAIN TIME
4094                                                                    ;CNTRL RDY DOESN'T SET AN ERROR
4095                                                                    ;MESSAGE IS GIVEN. WAITING TIME
4096                                                                    ;883 MS FOR 11/20, 175 MS FOR 11/45
4097  012502  020413                9$:     CMP     R4,@R3             ;WAS THE CORRECT WORD READ?
4098  012504  001407                        BEQ     TST30              ;YES, EXIT
4099  012506  010437  001162                MOV     R4,$REG0           ;GET EXPCTD WORD
4100  012512  011337  001164                MOV     @R3,$REG1          ;GET WORD RECVD
4101  012516  010437  001166                MOV     R4,$REG2           ;GET DISK ADRES WHERE ERROR OCCURED
4102  012522  104044                        ERROR   44                 ;DID NOT READ CORRECT WORD FROM
4103                                                                    ;CYL 0, SEC 0. IF TEST 45 & 46
4104                                                                    ;WERE SUCCESSFULLY DONE THIS
4105                                                                    ;ERROR COULD MEAN THAT DRIVE-RESET
4106                                                                    ;DID NOT BRING HEADS BACK TO 0.
4107
4108
4109                                ;;******************************************************************
```

```
                                ;*TEST 30       'WRITE' - 256 WORD BLOCK ON SECTOR 0, CYLINDER 0
                                        ;THE TEST BELOW SHOULD BE CONSIDERED AS A SET UP PHASE FOR
                                        ;THE FOLLOWING TEST.  IT WRITES A BLOCK OF 256 WORDS IN
                                        ;SECTOR 0, CYLINDER 0 WITH A SPECIFIC PATTERN AND THIS WRITTEN
                                        ;BLOCK WILL BE MADE USE OF IN THE NEXT TEST TO CHECK
                                        ;OUT 'WRITE-CHECK' AND 'READ CHECK' FUNCTIONS.
                                ;;*****************************************************************
012524  000004                TST30:  SCOPE
012526  104413                        CNT.RESET                       ;GO, DO CONTROL RESET
                                                                        ;THIS IS A CALL FOR THE 'CNTRL-
                                                                        ;RESET' ROUTINE. A CONTROL RESET IS
                                                                        ;ISSUED AND AFTER A CERTAIN TIME
                                                                        ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                        ;AN ERROR IS REPORTED. NOTE THAT
                                                                        ;THE PC IN ERROR MESSAGE IS THE
                                                                        ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                        ;THIS IS A VERY BASIC ERR& IF IT
                                                                        ;OCCURS GO BACK TO TEST 10
012530  104421                        TST.SIN                         ;CHECK IF SIN IS SET, IF SET
                                                                        ;DO DRIVE RESET TO CLEAR IT
012532  013704  001332                MOV     RKCS,R4
                                                                        ;THE FOLLOWING CODE IS FOR SETTING
                                                                ;UP THE I/O BUFFER IN MEMORY (STARTING AT
                                                                ;OUTBUF), WITH A PARTICULAR 256 WORD PATTERN.
                                                                ;STARTING FROM THE FIRST WORD IN THE BUFFER
                                                                ;THE LO BYTE WILL BE A COUNT PATTERN
                                                                ;FROM 0 TO 255 (DECIMAL), WHEREAS THE
                                                                ;HI-BYTE WILL BE THE COMPLEMENT OF LO BYTE,
                                                                ;A DECREASING COUNT PATTERN FROM 255 TO 0.
                                                                ;I.E.THE BUFFER WILL LOOK LIKE:
                                                                ;OUTBUF         (1 111 111 1 00 000 000)
                                                                ;OUTBUF+2       (1 111 111 0 00 000 001)
                                                                ;
                                                                ;LAST WORD      (0 000 000 0 11 111 111)
012536  012700  033336                        MOV     #OUTBUF,R0
012542  012701  177401                        MOV     #177401,R1      ;PATTERN GENERATING NUMBER
012546  012702  177400                        MOV     #-400,R2        ;SET UP COUNT FOR 256 WORDS
012552  012703  177400                        MOV     #177400,R3      ;SET UP THE FIRST PATTERN TO B WRITTEN
012556  010320                        MOV     R3,(R0)+        ;SET UP FIRST WORD IN I/O BUFFER
012560  005202                        INC     R2              ;INCREMENT COUNT
012562  060103                1$:     ADD     R1,R3           ;SET UP NEXT WORD PATTERN
012564  010320                        MOV     R3,(R0)+        ;WRITE IT IN NXT I/O BUFFER WORD
012566  005202                        INC     R2              ;HAVE U WRITTEN ALL 256 WORDS
012570  001374                        BNE     1$              ;IF NOT GO & WRITE NEXT PATTERN
012572  012777  177400  166534        MOV     #-400,@RKWC     ;WRITE 256 WORDS
012600  012777  033336  166530        MOV     #OUTBUF,@RKBA   ;STARTING FROM THIS BUS ADRES
012606  013777  001350  166524        MOV     DRIVAD,@RKDA    ;TO THIS DISK ADRES, CYL 0, SEC 0
012614  012714  000003                MOV     #3,@R4          ;WRITE, GO
012620  105714                2$:     TSTB    @R4             ;WAS CNTRL RDY CLEARED AS GO WAS SET?
012622  100003                        BPL     3$-2            ;YES. BRANCH
012624  004737  020776                JSR     PC,GT3RG        ;GET RKCS, ER, DS
```

```
4166  012630  104030                          ERROR   30              ;CNTRL RDY DID NOT CLEAR AS GO WAS SET
4167                                                                  ;TO 'WRITE'
4168
4169  012632  005002                          CLR     R2
4170  012634  105777  166472          3$:     TSTB    @RKCS           ;DID CNTRL RDY SET?
4171  012640  100411                          BMI     4$              ;YES, BRANCH
4172  012642  005202                          INC     R2              ;WAITED LONG ENOUGH?
4173  012644  001373                          BNE     3$              ;IF NOT, LUP BAK & WAIT
4174                                                                  ;IF YES, ERROR
4175  012646  004737  020770                  JSR     PC,GT4RG        ;GO, GET RKCD, ER, DS, DA
4176  012652  013737  001350  001202          MOV     DRIVAD,$REG10   ;GET THE STARING ADRES
4177  012660  104416                          BRKDA4                  ;BREAK CONTENTS OF $REG10 INTO
4178                                                                  ;DRV #, CYL, SUR, SEC #
4179  012662  104031                          ERROR   31              ;CNTRL RDY DID NOT SET ON COMPLETION
4180                                                                  ;OF WRITE OF 256 WORDS ON CYL 0, SEC 0
4181                                                                  ;'RKDA' IN EROR MSGE GIVES THE
4182                                                                  ;CONTENTS OF RKDA AT THE TIME OF ERCR
4183                                                                  ;WRITE WAS DONE STARTING AT <DSK-ADRES>
4184                                                                  ;INDICATED IN EROR MSGE
4185  012664  004737  021230          4$:     JSR     PC,CHKHE        ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
4186                                                                  ;IF YES, RETURN HERE
4187  012670  104032                          ERROR   32              ;HE OR ERR BIT SET ON DOING WRITE OF
4188                                                                  ;256 WORDS ON CYL 0, SEC 0
4189                                                                  ;WRITE WAS DONE STARTING AT <DSK-ADRES>
4190                                                                  ;INDICATED IN EROR MSGE
4191                                                                  ;'RKDA' IN EROR MSGE GIVES THE
4192                                                                  ;CONTENTS OF RKDA AT THE TIME OF EROR
4193  012672  020077  166440          5$:     CMP     R0,@RKBA        ;DID RKBA INCREMENT CORRECTLY?
4194  012676  001406                          BEQ     6$              ;YES, BRANCH
4195  012700  010037  001162                  MOV     R0,$REG0        ;GET EXPCTD RKBA
4196  012704  017737  166426  001164          MOV     @RKBA,$REG1     ;GET RKBA RECVD
4197  012712  104035                          ERROR   35              ;RKBA DID NOT INCREMENT CORRECTLY
4198                                                                  ;(BY 1000 OCTAL BYTES) AFTER WRITE
4199                                                                  ;OF 400 (OCTAL) WORDS ON SEC 0, CYL 0
4200  012714  004737  021312          6$:     JSR     PC,CHKWC        ;CHECK IF RKWC OVERFLOWED TO 0,
4201                                                                  ;IF NOT RETURN HERE.
4202  012720  104034                          ERROR   34              ;RKWC DID NOT OVERFLOW, AFTER A
4203                                                                  ;WRITE OF 256 WORDS ON CYL 0, SEC 0
4204  012722  004737  021256          7$:     JSR     PC,CHKDA        ;CHECK IF RKDA INCREMENTED CORRECTLY.
4205                                                                  ;IF NOT RETURN HERE
4206  012726  104033                          ERROR   33              ;RKDA DID NOT INCREMENT BY 1 AFTER
4207                                                                  ;A WRITE OF 256 WORDS IN CYL 0, SEC 0
4208  012730  004737  021336          8$:     JSR     PC,CHKER        ;CHECK IF ANY BIT RKER IS SET
4209                                                                  ;IF YES RETURN HERE.
4210  012734  104036                          ERROR   36              ;RKER BIT SET ON DOING WRITE ON
4211                                                                  ;CYLINDER 0, SECTOR 0
4212  012736  022714  000202          9$:     CMP     #202,@R4        ;DOES RKCS STILL CONTAIN THE WRITE BITS?
4213  012742  001406                          BEQ     TST31           ;YES, EXIT
4214  012744  012737  000202  001162          MOV     #202,$REG0      ;GET EXPECTED RKCS
4215  012752  011437  001164                  MOV     @R4,$REG1       ;GET RKCS RECVD
4216  012756  104024                          ERROR   24              ;RKCS DID NOT CONTAIN THE 'WRITE'
4217                                                                  ;BITS AFTER THE FUNCTION WAS DONE.
4218
4219
4220                                  ;;*****************************************************************
4221                                  ;*TEST 31        CHECK THAT WRITE WAS DONE CORRECTLY
```

```
                                        ;*THIS TEST CHECKS IF THE 'WRITE' OF 256 WORDS DONE IN PPEVIOUS
                                        ;*TEST IS GOOD.  THE SEQUENCE OF OPERATIONS IS AS FOLLOWING:
                                        ;*1) DO A READ OF 256 WORDS FROM SECTOR 0, CYLINDER 0
                                        ;*   INTO A BUFFER STARTING AT 'OUTBUF'.
                                        ;*2) COMPARE & CHECK THE DATA THAT IS READ (STARTING AT 'OUTBUF',
                                        ;*   WITH THE DATA THAT WAS GENERATED PREVIOUSLY
                                        ;*3) REPORT AN ERROR IF THE DATA READ BACK FROM DISK DOES
                                        ;*   NOT COMPARE WITH DATA THAT WAS SUPPOSE TO HAVE BEEN WRITTEN
                                    ;;*********************************************************************
012760  000004                    T31$:  SCOPE
012762  104413                           CNT.RESET                ;GO, DO CONTROL RESET
                                                                 ;THIS IS A CALL FOR THE 'CNTRL-
                                                                 ;RESET' ROUTINE. A CONTROL RESET IS
                                                                 ;ISSUED AND AFTER A CERTAIN TIME
                                                                 ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                 ;AN ERROR IS REPORTED. NOTE THAT
                                                                 ;THE PC IN ERROR MESSAGE IS THE
                                                                 ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                 ;THIS IS A VERY BASIC ERR& IF IT
                                                                 ;OCCURS GO BACK TO TEST 10
012764  104421                           TST.SIN                  ;CHECK IF SIN IS SET, IF SET
                                                                 ;DO DRIVE RESET TO CLEAR IT
012766  012700  177400                   MOV     #-400,R0         ;SET COUNT FOR 400 WORDS
012772  012701  033336                   MOV     #OUTBUF,R1       ;TO BE CLEARED IN THE BUFFER
012776  005021                    8$:    CLR     (R1)+            ;CLR THE 400 WORD BUFFER
013000  005200                           INC     R0               ;STARTING AT 'OUTBUF'
013002  001375                           BNE     8$
013004  005000                           CLR     R0
013006  012777  177400  166320           MOV     #-400,@RKWC      ;READ 256 WORDS
013014  012777  033336  166314           MOV     #OUTBUF,@RKBA    ;INTO THIS ADRES
013022  013777  001350  166310           MOV     DRIVAD,@RKDA     ;STARTING FROM THIS DISK ADRES
013030  012777  000005  166274           MOV     #5,@RKCS         ;READ, GO
013036  105777  166270            1$:    TSTB    @RKCS            ;DID CNTRL RDY SET?
013042  100411                           BMI     2$               ;YES, BRANCH
013044  005200                           INC     R0               ;WAITED LONG ENOUGH?
013046  001373                           BNE     1$               ;IF NOT, LUP BAK & WAIT
                                                                 ;ERROR, IF YES
013050  004737  020770                   JSR     PC,GT4RG         ;GO, GET RKCD, ER, DS, DA
013054  013737  001350  001202           MOV     DRIVAD,$REG10    ;GET THE STARTING ADRES
013062  104416                           BRKDA4                   ;GO TO 'BDA4' & BREAK CONTENTS OF
                                                                 ;$REG10 INTO DRV #, CYL, SUR, SEC BITS
013064  104045                           ERROR   45               ;CNTRL RDY DID NOT SET AFTER READ
                                                                 ;OF 400 WORDS FROM CYL 0, SEC 0
                                                                 ;'RKDA' IN EROR MSGE GIVES THE
                                                                 ;CONTENTS OF RKDA AT THE TIME OF EROR
                                                                 ;READ WAS DONE STARTING AT <DSK-ADRES>
                                                                 ;INDICATED IN EROR MESGE
013066  032777  001000  166232    2$:    BIT     #1000,@RKDS      ;IS SIN SET?
013074  001033                           BNE     TST32            ;;IF YES, EXIT
013076  012701  177400            5$:    MOV     #-400,R1
013102  012702  177777                   MOV     #177777,R2
013106  012703  033336                   MOV     #OUTBUF,R3
013112  012705  177773                   MOV     #-5,R5
013116  062702  177401            6$:    ADD     #177401,R2
```

```
                                T31     CHECK THAT WRITE WAS DONE CORRECTLY

4278  013122  020213                            CMP     R2,(R3) ;WAS THE READ WORD SAME AS THE WORD
4279                                                              ;THAT WAS SUPPOSE TO BE WRITTEN
4280  013124  001414                            BEQ     7$      ;YES, BRANCH
4281                                                              ;NO, ERROR
4282  013126  010137  001162                    MOV     R1,$REG0        ;GET THE # OF WORD
4283  013132  062737  000401  001162            ADD     #401,$REG0      ;THAT IS IN ERROR (EXAMPLE=1,2--376,377,400,
4284  013140  010237  001164                    MOV     R2,$REG1        ;GET EXPCTD WORD (THAT WAS SUPPOSED TO
4285                                                              ;BE WRITTEN)
4286  013144  011337  001166                    MOV     (R3),$REG2      ;GET WORD RECVD (THAT WAS READ BAK)
4287  013150  104055                            ERROR   55              ;DID NOT READ BACK WORD THAT WAS SUPPOSED
4288                                                              ;TO HAVE BEEN WRITTEN PREVIOUSLY. POSITION
4289                                                              ;OF WORD IN ERROR IS AS INDICATED BY
4290                                                              ;WORD # ($REG0), SEC 0, CYL 0
4291  013152  005205                            INC     R5
4292  013154  001403                            BEQ     TST32           ;;EXIT
4293  013156  005723                    7$:     TST     (R3)+           ;INCREMENT POINTER TO NXT WORD (THAT
4294                                                              ;WAS READ BACK)
4295  013160  005201                            INC     R1              ;HAVE U CHKD ALL 256 WORDS?
4296  013162  001355                            BNE     6$              ;IF NOT, LUP BAK & CHK THE NXT WORD
4297                                                              ;IF YES, EXIT
4298
4299                            ;;*****************************************************************
4300                            ;*TEST 32       CHECK 'READ CHECK' FUNCTION - CYLINDER 0, SECTOR 0
4301                                    ;*THIS TEST CHECKS OUT THE BASIC :READ CHECK' LOGIC, USING THE DATA BLOCK
4302                                    ;*'CYLINDER, SECTOR 0) WRITTEN IN A PREVIOUS TEST.  HENCE THE TEST WHICH
4303                                    ;*WRITES THE DATA BLOCK SHOULD BE DONE PRIOR TO THIS TEST.
4304                            ;;*****************************************************************
4305  013164  000004            TST32:  SCOPE
4306  013166  104413                    CNT.RESET                       ;GO, DO CONTROL RESET
4307                                                              ;THIS IS A CALL FOR THE 'CNTRL-
4308                                                              ;RESET' ROUTINE. A CONTROL RESET IS
4309                                                              ;ISSUED AND AFTER A CERTAIN TIME
4310                                                              ;IF THE 'CNTRL RDY' DOES NOT SET
4311                                                              ;AN ERROR IS REPORTED. NOTE THAT
4312                                                              ;THE PC IN ERROR MESSAGE IS THE
4313                                                              ;PC WHERE 'CNT.RESET' IS LOCATED.
4314                                                              ;THIS IS A VERY BASIC ERR& IF IT
4315                                                              ;OCCURS GO BACK TO TEST 10
4316  013170  104421                    TST.SIN                         ;CHECK IF SIN IS SET, IF SET
4317                                                              ;DO DRIVE RESET TO CLEAR IT
4318  013172  013701  001332                    MOV     RKCS,R1
4319  013176  013702  001334                    MOV     RKWC,R2
4320  013202  013703  001340                    MOV     RKDA,R3
4321  013206  013704  001336                    MOV     RKBA,R4
4322  013212  012737  052525  033336            MOV     #52525,OUTBUF
4323  013220  012712  177400                    MOV     #-400,@R2       ;READ CHECK 256 WORDS
4324  013224  013713  001350                    MOV     DRIVAD,@R3      ;STARTING FROM CYL 0, SECTOR 0
4325  013230  012714  033336                    MOV     #OUTBUF,@R4
4326  013234  012711  000013            .       MOV     #13,@R1         ;READ CHECK, GO
4327
4328  013240  105711                    1$:     TSTB    @R1             ;DID CNTRL RDY GET CLEARED AS GO WAS SET?
4329  013242  100003                            BPL     2$              ;YES, BRANCH
4330  013244  004737  020776                    JSR     PC,GT3RG        ;GET RKCS, ER, DS
4331  013250  104030                            ERROR   30              ;CNTRL RDY DID NOT CLEAR AS GO
4332  013252  104412                    2$:     CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
4333                                                              ;IF SO, SKIP THE EROR MESSAGE.
```

```
4334
4335  013254  104056                          ERROR   56              ;WAS SET TO 'READ CHECK'
4336                                                                  ;CNTRL RDY DID NOT SET ON DOING
                                                                      ;'READ CHECK' FROM CYL 0, SEC 0
4337  013256  032711  140000          3$:     BIT     #140000,@R1     ;DID 'ERR' OR 'HE' BIT SET?
4338  013262  001403                          BEQ     4$              ;NO, BRANCH
4339  013264  004737  020776                  JSR     PC,GT3RG        ;GO, GET RKCS,ER,DS FOR ERROR MESSAGE
4340  013270  104057                          ERROR   57              ;'ERR' OR 'HE' BIT SET ON DOING
4341                                                                  ;'READ CHECK' ON CYLINDER 0, SEC 0
4342  013272  032777  000002  166030  4$:     BIT     #2,@RKER        ;DID 'CSE' BIT SET IN RKER?
4343  013300  001404                          BEQ     5$              ;NO, BRANCH
4344  013302  017737  166022  001162          MOV     @RKER,$REG0     ;GET RKER
4345  013310  104060                          ERROR   60              ;SOFT ERROR - CSE - ON DOING 'READ
4346                                                                  ;CHECK' ON CYLINDER 0, SECTOR 0
4347                                                                  ;U SHOULD HAVE GOT ERRROR 102 ALSO
4348  013312  005712                  5$:     TST     @R2             ;DID WORD COUNT OVERFLOW TO 0?
4349  013314  001405                          BEQ     6$              ;YES, BRANCH
4350  013316  011237  001162                  MOV     @R2,$REG0       ;GET RKWC
4351  013322  011137  001164                  MOV     @R1,$REG1       ;GET RKCS
4352  013326  104061                          ERROR   61              ;WORD COUNT DID NOT OVERFLOW
4353                                                                  ;ON DOING 'READ CHK' ON CYL 0, SEC 0
4354  013330  013702  001350          6$:     MOV     DRIVAD,R2       ;RKDA SHOULD INCREMENT
4355  013334  005202                          INC     R2              ;TO THIS AFTER 'RD CHK' IS DONE
4356  013336  020213                          CMP     R2,@R3          ;DID RKDA INCREMENT CORRECTLY?
4357  013340  001405                          BEQ     7$
4358  013342  010237  001162                  MOV     R2,$REG0        ;GET EXPCTD RKDA
4359  013346  011337  001164                  MOV     @R3,$REG1       ;GET RKDA RECVD
4360  013352  104062                          ERROR   62              ;RKDA DID NOT INCREMENT CORRECTLY
4361                                                                  ;(BY 1) ON DOING 'READ CHK' ON
4362                                                                  ;CYL 0, SEC 0
4363  013354  022714  033336          7$:     CMP     #OUTBUF,@R4     ;DID RKBA GET CHANGED?
4364  013360  001406                          BEQ     8$              ;NO, BRANCH (RKBA WON'T CHANGE, NO NPR'S)
4365  013362  012737  033336  001162          MOV     #OUTBUF,$REG0   ;GET EXPCTD RKBA
4366  013370  011437  001164                  MOV     @R4,$REG1       ;GET RKBA RECVD
4367  013374  104063                          ERROR   63              ;RKBA CHANGED AFTER DOING 'READ CHK'
4368                                                                  ;ON CYLINDER 0, SECTOR 0. SHOULD
4369                                                                  ;NOT CHANGE, FOR, NO NPR'S.
4370  013376  022737  052525  033336  8$:     CMP     #52525,OUTBUF   ;'OUTBUF' SHOULD STILL CONTAIN THE
4371                                                                  ;SAME WORD AS IT DID BEFORE 'RD CHK'
4372                                                                  ;NOTE THAT AT THE BEGINING OF THIS TEST
4373                                                                  ;52525 WAS WRITTEN INTO 'OUTBUF'
4374  013404  001412                          BEQ     TST33           ;YES, EXIT
4375                                                                  ;REPORT ERROR IF 'OUTBUF' CHANGED
4376  013406  012737  033336  001162          MOV     #OUTBUF,$REG0   ;GET ADRES OF OUTBUF
4377  013414  012737  052525  001164          MOV     #52525,$REG1    ;GET EXPCTD WORD IN 'OUTBUF'
4378  013422  013737  033336  001166          MOV     OUTBUF,$REG2    ;GET WORD FOUND IN 'OUTBUF'
4379  013430  104064                          ERROR   64              ;AS MENTIONED ABOVE, IF 'WRITE' OF
4380                                                                  ;256 WORD DATA BLOCK WAS DONE
4381                                                                  ;CORRECTLY BEFORE, THEN THIS ERROR
4382                                                                  ;COULD MEAN THAT AN NPR WAS DONE
4383                                                                  ;ON 'READ CHECK'.
4384
4385                          ;;*****************************************************************
4386                          ;*TEST 33       CHECK THE 'WRITE CHECK' FUNCTION - ON CYLINDER 0, SECTOR 0
4387                                  ;*THIS TEST CHECKS OUT THE BASIC 'WRITE CHECK' LOGIC, USING THE 256
4388                                  ;*WORD DATA BLOCK (SECTOR 0, CYLINDER 0) WRITTEN IN A PREVIOUS
4389                                  ;*TEST. THE BUFFER IN MEMORY, USED FOR COMPARISON OF DATA, IS THE
```

```
                        T33     CHECK THE 'WRITE CHECK' FUNCTION - ON CYLINDER 0, SECTOR 0

                                ;*ONE STARTING AT 'OUTBUF'. HENCE THE TEST WHICH WRITES THE
                                ;*256 WORD BLOCK ON THE DISK (AS WELL AS CREATING THE 256
                                ;*256 WORD MEMORY BUFFER) SHOULD BE DONE BEFORE THIS TEST.
                                ;;*****************************************************************
013432  000004                TST33:  SCOPE
013434  104413                        CNT.RESET               ;GO, DO CONTROL RESET
                                                            ;THIS IS A CALL FOR THE 'CNTRL-
                                                            ;RESET' ROUTINE. A CONTROL RESET IS
                                                            ;ISSUED AND AFTER A CERTAIN TIME
                                                            ;IF THE 'CNTRL RDY' DOES NOT SET
                                                            ;AN ERROR IS REPORTED. NOTE THAT
                                                            ;THE PC IN ERROR MESSAGE IS THE
                                                            ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                            ;THIS IS A VERY BASIC ERR& IF I~
                                                            ;OCCURS GO BACK TO TEST 10
013436  104421                        TST.SIN                 ;CHECK IF SIN IS SET, IF SE~
                                                            ;DO DRIVE RESET TO CLEAR IT
013440  013701  001332                MOV     RKCS,R1
013444  012700  177400                MOV     #-400,R0
013450  012702  033336                MOV     #OUTBUF,R2
013454  012703  177777                MOV     #177777,R3
013460  062703  177401        1$:     ADD     #177401,R3
013464  010322                MOV     R3,(R2)+
013466  005200                INC     R0
013470  001373                BNE     1$
013472  012777  177400  165634      MOV     #-400,@RkWC     ;WRITE CHECK 256 WORDS
013500  012777  033336  165630      MOV     #OUTBUF,@RKBA   ;STARTING AT THIS BUS PDRES
013506  013777  001350  165624      MOV     DRIVAD,@RKDA    ;WITH THIS DISK DATA BLOCK (CYL 0, SEC 0,
013514  012711  000007              MOV     #7,@R1          ;WRITE CHECK, GO

013520  005000                2$:     CLR     R0              ;GIVE SOME TIME
013522  105711                        TSTB    @R1             ;DID CNTRL RDY CLEAR AS GO WAS SET?
013524  100003                        BPL     3$              ;YES BRANCH
013526  004737  020776                JSR     PC,GT3RG        ;GET RKCS, ER, DS
013532  104030                        ERROR   30              ;CNTRL RDY DID NOT CLEAR AS GO WAS
                                                            ;SET TO DO WRITE CHECK
013534  104412                3$:     CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
                                                            ;IF SO, SKIP THE EROR MESSAGE.
013536  104065                        ERROR   65              ;CNTRL RDY DID NOT SET AFTER
                                                            ;COMPLETING WRITE CHECK ON
                                                            ;CYLINDER 0, SECTOR 0
013540  032711  140000        4$:     BIT     #140000,@R1     ;DID HE OR ERR BIT SET
013544  001403                        BEQ     5$              ;NO, BRANCH
013546  004737  020776                JSR     PC,GT3RG        ;GO GET RKCS ER DS FOR ERROR MESSAGE
013552  104066                        ERROR   66              ;HE OR ERR BIT SET ON DOING WRITE
                                                            ;CHK ON CYLINDER 0, SEC 0
013554  032777  000001  165546  5$:     BIT     #1,@RKER        ;DID WCE SET IN RKER?
013562  001403                        BEQ     6$              ;NO, BRANCH
013564  004737  020776                JSR     PC,GT3RG        ;YES GET RKCS, ER, DS
013570  104067                        ERROR   67              ;WCE ON WRITE CHECK OF CYL 0, SEC 0
                                                            ;NOTE THAT IF A PREVIOUS TEST
                                                            ;& THEN COMPARED WITH MEMORY BUFFER
                                                            ;TO SEE IF IT WAS WRITTEN CORRECT WAS
                                                            ;DONE RIGHT BEFORE, THIS ERROR SHOULD NOT
                                                            ;HAPPEN UNLESS THERE IS A FAULT IN THE
                                                            ;COMPARING LOGIC OF 'WRT CHK'
```

```
                                T33     CHECK THE 'WRITE CHECK' FUNCTION - ON CYLINDER 0, SECTOR 0

4446  013572  005777  165536          6$:     TST     @RKWC           ;DID RKWC OVERFLOW?
4447  013576  001406                          BEQ     7$              ;YES, BRANCH
4448  013600  017737  165530  001162          MOV     @RKWC,$REG0     ;NO, GET RKWC
4449  013606  011137  001164                  MOV     @R1,$REG1       ;GET RKCS
4450  013612  104061                          ERROR   61              ;RKWC DID NOT OVERFLOW AFTER
4451                                                                  ;WRITE CHECK ON CYL 0, SEC 0
4452  013614  013704  001350          7$:     MOV     DRIVAD, R4      ;RKDA SHOULD INCREMENT
4453  013620  005204                          INC     R4              ;TO THIS AFTER WRT CHK
4454  013622  020477  165512                  CMP     R4,@RKDA        ;DID RKDA INCREMENT CORRECTLY?
4455  013626  001406                          BEQ     8$              ;YES, BRANCH
4456  013630  010437  001162                  MOV     R4,$REG0        ;NO, GET EXPCTD RKDA
4457  013634  017737  165500  001164          MOV     @RKDA,$REG1     ;GET RKDA RECVD
4458  013642  104070                          ERROR   70              ;RKDA DID NOT INCREMENT CORRECTLY
4459                                                                  ;(BY 1 SECTOR) AFTER WAT CHK ON SEC 0, CYL 0
4460  013644  022777  034336  165464  8$:     CMP     #OUTBUF+1000,@RKBA ;DID RKBA INCREMENT CORRECTLY?
4461  013652  001407                          BEQ     9$              ;YES, EXIT
4462  013654  012737  034336  001162          MOV     #OUTBUF+1000,$REG0 ;GET EPCTD RKBA
4463  013662  017737  165450  001164          MOV     @RKBA,$REG1     ;GET RKBA RECVD
4464  013670  104071                          ERROR   71              ;RKBA DID NOT INCREMENT CORRECTLY
4465                                                                  ;(BY 1000 BYTES) AFTER A WRT CHK
4466                                                                  ;OF 256 WORDS ON CYL 0, SEC 0
4467  013672  022711  000206          9$:     CMP     #206,@R1        ;DOES RKCS STILL CONTAIN THE WRT CHK BITS?
4468  013676  001406                          BEQ     TST34           ;;YES, BRANCH
4469  013700  012737  000206  001162          MOV     #206,$REG0      ;NO, GET EXPCTD RKCS
4470  013706  011137  001164                  MOV     @R1,$REG1       ;GET RKCS RECVD
4471  013712  104024                          ERROR   24              ;RKCS BITS CHANGED AFTER WRT CHK
4472                                                                  ;WAS DONE
4473                                ;;********************************************************************
4474                                ;*TEST 34       CHECK THAT IBA INHIBITS INCREMENTING OF RKBA
4475                                        ;*THIS TEST CHECKS THAT THE BUS ADDRESS DOES NOT INCREMENT WHEN
4476                                        ;*THE IBA BIT IS SET. SEQUENCE OF OPERATIONS:
4477                                        ;*1) CLEAR OUT 256 WORD BUFFER IN MEMORY (OUTBUF,
4478                                        ;*2) READ FROM SECTOR 0, CYLINDER 0 THE 256 WORD BLOCK THAT WAS
4479                                        ;*WRITTEN IN A PREVIOUS TEST (NOTE: THAT TEST SHOULD HAVE BEEN
4480                                        ;*DONE BEFORE THIS). IBA BIT IS SET DURING READ BACK.
4481                                        ;*3) CHECK THAT RKBA DID NOT INCREMENT
4482                                        ;*4) CHECK THAT THE ENTIRE BLOCK WAS READ INTO THE SAME MEMORY
4483                                        ;*WORD (OUTBUF) & THE REST OF THE WORDS IN THAT BUFFER ARE C
4484                                        ;*AS PREVIOUSLY CLEARED OUT.
4485                                ;;********************************************************************
4486  013714  000004                TST34:  SCOPE
4487  013716  104413                          CNT.RESET               ;GO, DO CONTROL RESET
4488                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
4489                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
4490                                                                  ;ISSUED AND AFTER A CERTAIN TIME
4491                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
4492                                                                  ;AN ERROR IS REPORTED. NOTE THAT
4493                                                                  ;THE PC IN ERROR MESSAGE IS THE
4494                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
4495                                                                  ;THIS IS A VERY BASIC ERR8 IF IT
4496                                                                  ;OCCURS GO BACK TO TEST 10
4497  013720  104421                          TST.SIN                 ;CHECK IF SIN IS SET, IF SET
4498                                                                  ;DO DRIVE RESET TO CLEAR IT
4499  013722  013701  001332                  MOV     RKCS,R1
4500  013726  012700  177400                  MOV     #-400,R0        ;SET UP COUNT FOR 256 WORDS
4501  013732  012702  033336                  MOV     #OUTBUF,R2
```

```
4502  013736  010203                          MOV     R2,R3
4503
4504  013740  005023                  1$:     CLR     (R3)+           ;CLEAR OUT THE 256
4505  013742  005200                          INC     R0              ;WORD MEMORY BUFFER STARTING
4506  013744  001375                          BNE     1$              ;AT 'OUTBUF'
4507  013746  012777  177400  165360          MOV     #-400,@RKWC     ;READ BACK 256 WORDS
4508  013754  010277  165356                  MOV     R2,@RKBA        ;INTO THIS BUS ADRES (IBA WILL B SET.
4509  013760  013777  001350  165352          MOV     DRIVAD,@RKDA    ;FROM THIS DSK ADRES (SEC 0, CYL 0.
4510                                                                  ;NOTE: SEC 0 HAS BEEN WRITTEN IN A
4511                                                                  ;PREVIOUS TEST WITH A UNIQUE PATTERN
4512  013766  012711  004005                  MOV     #4005,@R1       ;READ, GO, IBA SET
4513
4514  013772  005037  001362                  CLR     COUNT
4515  013776  105711                  2$:     TSTB    @R1             ;DID CNTRL RDY SET?
4516  014000  100412                          BMI     3$              ;YES, BRANCH
4517  014002  005237  001362                  INC     COUNT           ;WAITED LONG ENOUGH?
4518  014006  001373                          BNE     2$              ;IF NOT, LUP BAK & WAIT
4519  014010  004737  020770                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
4520  014014  013737  001350  001202          MOV     DRIVAD,$REG1D   ;GET THE STARTING ADRES
4521  014022  104416                          BRKDA4                  ;BREAK CONTENTS OF $REG1D
4522                                                                  ;INTO DR #, CYL, SUR, SEC
4523  014024  104045                          ERROR   45              ;CNTRL RDY DID NOT SET AFTER DOING
4524                                                                  ;READ
4525  014026  004737  021230          3$:     JSR     PC,CHKHE        ;CHECK IF 'ERR' OR 'HE' BIT IS SET.
4526                                                                  ;IF YES, RETURN HERE.
4527  014032  104046                          ERROR   46              ;ERR BIT SET ON DOING READ FROM SEC 0.
4528                                                                  ;CYL 0 (INDICATED  IN <DSK-ADRES >
4529                                                                  ;'RKDA' IN EROR MSGE GIVES THE
4530                                                                  ;CONTENTS OF RKDA AT THE TIME OF ERCR
4531
4532  014034  020277  165276          4$:     CMP     R2,@RKBA        ;DID RKBA INCREMENT?
4533  014040  001406                          BEQ     5$              ;OK IF NOT, BRANCH
4534  014042  010237  001162                  MOV     R2,$REG0        ;GET EXPCTD RKBA
4535  014046  017737  165264  001164          MOV     @RKBA,$REG1     ;GET RKBA RECVD
4536  014054  104072                          ERROR   72              ;RKBA INCREMNTED WHEN IBA BIT WAS
4537                                                                  ;SET, SHOULD NOT HAVE
4538  014056  032777  001000  165242  5$:     BIT     #1000,@RKDS     ;IS SIN SET?
4539  014064  001042                          BNE     TST35           ;;IF YES, EXIT
4540  014066  012700  177400                  MOV     #-400,R0
4541  014072  022712  000377                  CMP     #377,@R2        ;CHECK THAT THE FIRST WORD IN
4542                                                                  ;'OUTBUF' IS 377 (LAST WORD OF SEC 0,
4543                                                                  ;CYL 0). NOTE THAT READ WAS DONE
4544  014076  001411                          BEQ     6$              ;INTO THIS SAME WRD WITH IBA SET
4545  014100  012737  000377  001162          MOV     #377,$REG0      ;GET EXPCTD WORD (LAST WORD OF THE BUFFER
4546  014106  011237  001164                  MOV     (R2),$REG1      ;GET WORD RECVD (LAST WRD FROM SEC 0)
4547  014112  013737  001350  001166          MOV     DRIVAD,$REG2    ;DISK ADRES WHERE ERROR OCCURED
4548                                                                  ;(SEC 0, CYL 0 LAST WORD)
4549                                                                  ;DATA ERROR
4550  014120  104044                          ERROR   44              ;THE FIRST WORD IN MEM BUFFER (OUTBUF)
4551                                                                  ;SHOULD BE NON-ZERO & SHOULD CONTAIN
4552                                                                  ;THE LAST WORD READ BACK FROM SEC 0
4553                                                                  ;CYL 0,THIS DID NOT HAPPEN IF THE ERROR OCCURS
4554  014122  005722                  6$:     TST     (R2)+           ;INCREMENT POINTER TO THE NXT WORD
4555  014124  012705  177773                  MOV     #-5,R5          ;ALLOW ONLY 5 MESAGES FOR ERR 116
4556  014130  005200                  7$:     INC     R0              ;CHKD ALL 256 WORDS IN THE BUFFER?
4557  014132  001417                          BEQ     TST35           ;;YES, EXIT
```

```
4559  014134  005722                          TST     (R2)+           ;IS THIS WORD 0?
4559  014136  001774                          BEQ     7$              ;YES, LUP BAK & CHK THE NXT WORD?
4560  014140  005037  001164                  CLR     $REG1           ;ERROR. GET EXPCTED WORD - 0
4561  014144  014237  001166                  MOV     -(R2),$REG2     ;GET WORD THAT WAS FOUND IN THE BUFFER
4562  014150  010004                          MOV     R0,R4
4563  014152  062704  000401                  ADD     #401,R4
4564  014156  010437  001162                  MOV     R4,$REG0        ;THIS 'WORD #' IN MEMORY BUFFER
4565                                                                  ;SHOULD HAVE BEEN ZERO
4566  014162  104403                          ERROR   73              ;THE 256 WORD BUFER (STARTING AT
4567                                                                  ;OUTBUF) WAS CLEARED BEFORE READING
4568                                                                  ;BAK SEC 0 INTO IT. SINCE THE IBA
4569                                                                  ;BIT WAS SET DURING THE READ, ONLY
4570                                                                  ;THE FIRST WORD OF (OUTBUF) SHOULD
4571                                                                  ;HAVE CHANGED. THE REST OF THE WORDS
4572                                                                  ;SHOULD BE STILL 0. IF THIS ERROR
4573                                                                  ;OCCURS, 'WORD #' (OF THE BUFFER) AS
4574                                                                  ;INDICATED IN THE EROR MESAGE) GOT
4575                                                                  ;CHANGED WHEN READ WAS DONE FROM
4576                                                                  ;THE DISK, INDICATING THAT WITH IBA
4577                                                                  ;SET X-FER WAS NOT DONE INTO THE
4578                                                                  ;SAME MEMORY LOCATION. 'WORD #'
4579                                                                  ;IS OCTAL & SPECIFIES THE POSITION
4580                                                                  ;IN THE BUFFER (FIRST WORD IS 'WORD #' 1)
4581  014164  005205                          INC     R5
4582  014166  001401                          BEQ     TST35           ;;EXIT
4583  014170  000757                          BR      7$
4584                                          ;
4585                                          ;;*****************************************************************
4586                                          ;*TEST 35       CHECK THAT RK11 INTERRUPTS WHEN IDE IS SET
4587                                                  ;*THIS TEST CHECKS IF RK11 INTERRUPTS TO ITS DESIGNATED VECTOR
4588                                                  ;*ADDRESS WHEN IDE BIT IS SET, WITH CONTROL READY SET & GO CLEAR.
4589                                                  ;* IT IS NORMALLY 220, UNLESS IT HAS BEEN CHANGED. IF IT HAS BEEN
4590                                                  ;*CHANGED RK11 WILL INTERRUPT TO 'RKVEC'. NOTE 'RKVEC' HAS
4591                                                  ;*TO BE SET UP BY THE USER.
4592                                          ;;*****************************************************************
4593  014172  000004                          TST35:  SCOPE
4594  014174  104413                                  CNT.RESET               ;GO, DO CONTROL RESET
4595                                                                          ;THIS IS A CALL FOR THE 'CNTRL-
4596                                                                          ;RESET' ROUTINE. A CONTROL RESET IS
4597                                                                          ;ISSUED AND AFTER A CERTAIN TIME
4598                                                                          ;IF THE 'CNTRL RDY' DOES NOT SET
4599                                                                          ;AN ERROR IS REPORTED. NOTE THAT
4600                                                                          ;THE PC IN ERROR MESSAGE IS THE
4601                                                                          ;PC WHERE 'CNT.RESET' IS LOCATED.
4602                                                                          ;THIS IS A VERY BASIC ERR& IF IT
4603                                                                          ;OCCURS GO BACK TO TEST 10
4604  014176  104421                                  TST.SIN                 ;CHECK IF SIN IS SET, IF SET
4605                                                                          ;DO DRIVE RESET TO CLEAR IT
4606  014200  012746  000340                          MOV     #340,-(SP)
4607  014204  012746  014212                          MOV     #64$,-(SP)
4608  014210  000002                                  RTI
4609  014212                                  64$:
4610  014212  013701  001332                          MOV     RKCS,R1
4611  014216  013700  001402                          MOV     RKVEC,R0        ;GET POINTER TO RK VECTOR ADRES
4612  014222  012720  014256                          MOV     #1$,(R0)+       ;SET UP INTERRUPT VECTOR FOR RK11
4613  014226  012710  000340                          MOV     #340,(R0)       ;SET PSW ON INTERRUPT
```

```
4614  014232  105711                          TSTB    @R1             ;WAIT FOR CNTRL RDY TO SET
4615  014234  100376                          BPL     .-2             ;
4616  014236  012711  000100                 MOV     #100,@R1        ;SET IDE BIT IN RKCS
4617  014242  104420  000005                 WAT.INT .5              ;WAIT FOR INTERRUPT, ATLEAST
4618                                                                  ;37 US FOR 11/20, 7 US FOR 11/45
4619  014246  011137  001162                 MOV     @R1,$REGO       ;GET RKCS
4620  014252  104074                          ERROR   74              ;RK11 DID NOT INTERRUPT WHEN IDE
4621                                                                  ;WAS SET, WITH CNTRLE RDY SET & GO
4622                                                                  ;CLEAR
4623  014254  000400                          BR      1$
4624  014256  022626                  1$:     CMP     (SP)+,(SP)+     ;RK11 INTERRUPTED CORRECTLY TO
4625                                                                  ;THIS. RESTORE STACK POINTER
4626                                                                  ;(FROM RK11 INTERRUPT)
4627  014260  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
4628                                                                  ;(FROM  WAT.INT)
4629  014262  012777  014276  165112          MOV     #2$,@RKVEC      ;IF THERE IS FAULTY POLLING OR INTERUPT
4630                                                                  ; LOGIC SECOND INTERRUPT MIGHT OCCUR
4631  014270  104420  000005                 WAT.INT .5              ;WAIT FOR INTERRUPT,IF ANY
4632                                                                  ;DUE TO FAULTY LOGIC
4633
4634  014274  000403                          BR      3$
4635
4636  014276  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE STACK PTR (FROM RK11 INTRUPT)
4637  014300  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK PTR (FROM WAT.INT)
4638  014302  104020                          ERROR   20              ;AN UNEXPECTED RK11 INTERRUPT
4639                                                                  ;OCCURED. THERE SHOULD HAVE BEEN
4640                                                                  ;ONLY 1 INTERRUPT (TO 1$ ABOVE)
4641  014304  012777  004600  165070  3$:     MOV     #BADINT,@RKVEC  ;RESTORE VECTOR ADRES FOR
4642                                                                  ;UNEXPECTED RK11 INTERRUPT.
4643  014312  012746  000340                 MOV     #340,-(SP)
4644  014316  012746  014324                  MOV     #65$,-(SP)
4645  014322  000002                          RTI
4646  014324                          65$:
4647
4648
4649                                  ;;*****************************************************************
4650                                  ;*TEST 36       CHECK THAT WITH IDE SET RK11 INTERRUPTS AFTER INTIATION & COMPLETION OF
4651                                          ;*THIS TEST CHECKS THAT AN INTERRUPT FROM RK11 OCCURS AFTER
4652                                          ;*A SEEK IS INITIATED WITH 'IDE' BIT SET, AND THEN A SECOND
4653                                          ;*INTERRUPT OCCURS AFTER THE SEEK IS DONE. IT ALSO CHECKS THAT
4654                                          ;*AFTER THE FIRST INTERRUPT 'SCP' BIT IS NOT SET, WHEREAS AFTER
4655                                          ;*THE SECOND INTERRUPT 'SCP' IS SET.
4656                                          ;*THIS TEST ALSO CHECKS A PART OF THE POLLING LOGIC.
4657                                  ;;*****************************************************************
4658  014324  000004                  T3T36:  SCOPE
4659  014326  012737  000005  001206          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
4660  014334  104413                          CNT.RESET               ;GO, DO CONTROL RESET
4661                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
4662                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
4663                                                                  ;ISSUED AND AFTER A CERTAIN TIME
4664                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
4665                                                                  ;AN ERROR IS REPORTED. NOTE THAT
4666                                                                  ;THE PC IN ERROR MESSAGE IS THE
4667                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
4668                                                                  ;THIS IS A VERY BASIC ERR3 IF IT
4669                                                                  ;OCCURS GO BACK TO TEST 1C
```

```
4670  014336  013700  001332                MOV     @RKCS,R0
4671  014342  013777  001350  164770        MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE
4672  014350  004737  021500                JSR     PC,DRESET       ;GO, DO DRIVE RESET
4673  014354  104026                        ERROR   26              ;R/W/S RDY DIDN'T SET AFTER DOING
4674                                                                ;ABOVE DRIVE RESET
4675  014356  013701  001402        2$:     MOV     RKVEC,R1
4676  014362  012721  014426                MOV     #3$,(R1)+       ;SET UP VECTOR ADRES FOR RK11 INTERUPT
4677  014366  012711  000340                MOV     #340,(R1)       ;SET UP PSW ON INTERRUPT
4678  014372  052777  000040  164740        BIS     #40,@RKDA       ;ADRES  CYLINDER #1
4679  014400  012710  000111                MOV     #111,@R0        ;SEEK, GO WITH IDE SET
4680  014404  104420  000300                WAT.INT ,300            ;WAIT FOR THE DRIVE TO
4681                                                                ;INTERRUPT AFTER ADRES WAS RECVD
4682                                                                ;WAITING TIME= 1.4 MS FOR 11/20
4683                                                                ;280 US FOR 11/45
4684                                                                ;ERROR, IF INTERUPT DID NOT OCCUR
4685                                                                ;BY NOW
4686  014410  012777  004600  164764        MOV     #BADINT,@RKVEC  ;RESTORE UNEXPECTED RK11 INTERRUPT
4687  014416  011037  001162                MOV     @R0,$REG0       ;GET RKCS
4688  014422  104075                        ERROR   75              ;INTERRUPT DID NOT OCCUR AFTER
4689                                                                ;SEEK WAS INITIATED WITH IDE SET
4690  014424  000402                        BR      3$+4
4691  014426  022626          3$:           CMP     (SP)+,(SP)+     ;OK, IF RK11 INTERRUPTED TO THIS
4692                                                                ;RESTORE STACK POINTER (FROM RK11 INTERRUPT)
4693  014430  022626                        CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER (FROM
4694                                                                ;WAT.INT)
4695  014432  012777  014476  164742        MOV     #5$,@RKVEC      ;SET UP NEW VECTOR ADRES FOR RK11
4696  014440  032710  020000                BIT     #20000,@R0      ;IS SCP CLEAR
4697  014444  001403                        BEQ     4$              ;YES, BRANCH
4698  014446  011037  001162                MOV     @R0,$REG0       ;GET RKCS
4699  014452  104076                        ERROR   76              ;SCP SET BEFORE SEEK TO LAST
4700                                                                ;CYLINDER WAS DONE
4701  014454  104420  056700          4$:   WAT.INT ,56700          ;WAIT FOR DRIVE TO INTERRUPT
4702                                                                ;AFTER SEEK WAS COMPLETED
4703                                                                ;WAITING TIME=180 MS FOR 11/20
4704                                                                ;36 MS FOR 11/45
4705  014460  012777  004600  164714        MOV     #BADINT,@RKVEC  ;IT'S AN ERROR IF BY THIS TIME
4706                                                                ;INTERRUPT HAS NOT OCCURERED
4707  014466  004737  020776                JSR     PC,GT3RG                ;GO GET RKCS, ER, DS
4708  014472  104077                        ERROR   77              ;RK11 DID NOT INTERRUPT AFTER SEEK (TO
4709                                                                ;LAST CYLINDER) WAS DONE WITH IDE SET
4710  014474  000401                        BR      5$+2
4711  014476  022626          5$:           CMP     (SP)+,(SP)+     ;OK, IF RK11 INTERUPTED TO THIS AFTER
4712                                                                ;SEEK WAS COMPLETED. RESTORE
4713                                                                ;STACK POINTER (FROM RK11 INTERRUPT)
4714  014500  022626                        CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER (FROM
4715                                                                ;WAT.INT)
4716  014502  012777  004600  164672        MOV     #BADINT,@RKVEC  ;RESTORE RK11 INTERRUPT VECTOR ADRES
4717                                                                ;FOR UNEXPECTED INTERUTS
4718  014510  032710  020000                BIT     #20000,@R0      ;DID SCP BIT SET?
4719  014514  001003                        BNE     6$              ;YES, BRANCH
4720  014516  011037  001162                MOV     @R0,$REG0       ;GET RKCS
4721  014522  104053                        ERROR   53              ;SCP DID NOT SET AFTER RK11 INTERRUPTED
4722                                                                ;INDICATING SEEK WAS DONE
4723  014524  017701  164576          6$:   MOV     @RKDS,R1        ;GET RKDS
4724  014530  042701  017777                BIC     #17777,R1       ;MASK NON-ID BITS IN RKDS
4725  014534  020137  001350                CMP     R1,DRIVAD       ;CORRECT ID BITS IN RKDS?
```

T36 CHECK THAT WITH IDE SET RK11 INTERRUPTS AFTER INTIATION & COMPLETION OF SEEK

```
4726  014540  001414                          BEQ     7$              ;YES, BRANCH
4727
4728  014542  013746  001350                  MOV     DRIVAD,-(SP)    ;PUSH DRV ADRES ON THE STACK
4729  014546  004737  021174                  JSR     PC,SHFTRT       ;GO, SHIFT RIGHT DRV #
4730  014552  012637  001162                  MOV     (SP)+,$REGO     ;GET EXPCTD DRV #
4731  014556  010146                          MOV     R1,-(SP)        ;PUSH ID BITS ON THE STACK
4732  014560  004737  021174                  JSR     PC,SHFTRT       ;GO SHIFT THEM RIGHT
4733  014564  012637  001164                  MOV     (SP)+,$REG1     ;POP THE RECVD ID BITS
4734  014570  104047                          ERROR   47              ;WRONG ID BITS WERE RECVD IN
4735                                                                  ;RKDS AFTER SEEK WAS DONE (INTRUPT
4736                                                                  ;MODE). 'EXPCT' INDICATES THE DRIVE
4737                                                                  ;# THAT SHOULD HAVE BEEN IN THE
4738                                                                  ;ID BITS, 'RECVD' INDICATES THE
4739                                                                  ;DRIVE # THAT WAS RECVD IN THE ID BITS
4740
4741  014572                          7$:
4742  014572  012746  000340                  MOV     #340,-(SP)
4743  014576  012746  014604                  MOV     #64$,-(SP)
4744  014602  000002                          RTI
4745  014604                          64$:
4746  014604  104413                          CNT.RESET               ;GO DO CONTROL RESET
4747  014606  013777  001350  164524          MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE
4748  014614  032777  160000  164504          BIT     #160000,@RKDS   ;DID CNTRL RESET CLEAR DRIVE ID BITS?
4749  014622  001404                          BEQ     8$              ;YES, BRANCH
4750  014624  017737  164476  001162          MOV     @RKDS,$REGO     ;GET RKDS
4751  014632  104050                          ERROR   50              ;CONTROL RESET DIDN'T CLEAR THE
4752
4753                                                                  ;DRIVE ID BITS (13-15) IN RKDS
4754
4755  014634  022710  000200          8$:     CMP     #200,@RO        ;WAS SCP BIT CLEARED BY CNTRL RESET?
4756  014640  001403                          BEQ     TST37           ;YES, EXIT
4757  014642  011037  001162                  MOV     @RO,$REGO       ;GET RKCS
4758  014646  104100                          ERROR   100             ;CNTRL RESET DID NOT CLEAR SCP BIT
4759
4760                                  ;;**************************************************************
4761                                  ;*TEST 37       CHECK THAT WITH IDE SET RK11 INTERRUPTS WHEN READ IS DONE
4762                                          ;*THIS TEST CHECKS THAT WHEN A DATA TRANSFER FUNCTION IS DONE
4763                                          ;*WITH IDE BIT SET, RK11 INTERRUPTS WHEN THE FUNCTION IS COMPLETED
4764                                          ;*FUNCTION USED IN THIS TEST IS READ.
4765                                  ;;**************************************************************
4766  014650  000004                  TST37:  SCOPE
4767  014652  104413                          CNT.RESET               ;GO, DO CONTROL RESET
4768                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
4769                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
4770                                                                  ;ISSUED AND AFTER A CERTAIN TIME
4771                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
4772                                                                  ;AN ERROR IS REPORTED. NOTE THAT
4773                                                                  ;THE PC IN ERROR MESSAGE IS THE
4774                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
4775                                                                  ;THIS IS A VERY BASIC ERR& IF IT
4776                                                                  ;OCCURS GO BACK TO TEST 10
4777  014654  104421                          TST.SIN                 ;CHECK IF SIN IS SET, IF SET
4778                                                                  ;DO DRIVE RESET TO CLEAR IT
4779
4780  014656  013700  001332                  MOV     RKCS,R0
4781  014662  013702  001340                  MOV     RKDA,R2
```

```
4782  014666  013704  001336                    MOV     RKBA,R4
4783  014672  013701  001350                    MOV     DRIVAD,R1
4784  014676  052701  000013                    BIS     #13,R1          ;SET BITS FOR SEC 13
4785  014702  012777  177600  164424            MOV     #-200,@RKWC     ;READ 200 (OCTAL WORDS)
4786  014710  010112                            MOV     R1,@R2          ;FROM THIS DISK ADRES (CYL 0, SEC 13)
4787  014712  012714  033336                    MOV     #OUTBUF,@R4     ;INTO THIS BUS ADRES
4788  014716  013705  001402                    MOV     RKVEC,R5
4789  014722  012725  014760                    MOV     #1$,(R5)+       ;SET UP VECTOR ADRES FOR RK11 TO INTRUPT
4790  014726  012715  000340                    MOV     #340,(R5)       ;SET PSW ON INTERUPT
4791  014732  012710  000105                    MOV     #105,@R0        ;READ, GO, IDE SET
4792  014736  104420  127710                    WAT.INT ,127710         ;WAIT FOR RK11 TO INTERRUPT ON
4793                                                                    ;COMPLETION OF READ
4794                                                                    ;WAITING TIME= 337 MS FOR 11/20
4795                                                                    ;67 MS FOR 11/45
4796  014742  012777  004600  164432            MOV     #BADINT,@RKVEC  ;RESTORE UNEXPCTED INTERRUPT VECTOR ADRES
4797  014750  011037  001162                    MOV     @R0,$REG0       ;GET RKCS
4798  014754  104101                            ERROR   101             ;RK11 DID NOT INTERRUPT AFTER READ
4799                                                                    ;WAS DONE, IDE BIT SET.
4800  014756  000404                            BR      1$+10
4801  014760  022626                    1$:     CMP     (SP)+,(SP)+     ;OK, IF RK11 INTERRUPTED TO THIS
4802                                                                    ;RESTORE STACK POINTER (FROM RK11 INTERRUPT)
4803  014762  022626                            CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER (FROM WAT.INT)
4804  014764  012777  004600  164410            MOV     #BADINT,@RKVEC  ;RESTORE UNEXPECTED RK11 INTERRUPT
4805                                                                    ;VECTOR ADRES
4806  014772  004737  021336                    JSR     PC,CHKER        ;CHECK IF ANY BIT IN RKER IS SET,
4807                                                                    ;IF YES, RETURN HERE.
4808  014776  104036                            ERROR   36              ;RKER SET ON DOING READ FROM SEC 0,
4809                                                                    ; CYL 13 IN INTERRUPT MODE
4810  015000  062701  000005            4$:     ADD     #5,R1           ;RKDA SHOULD HAVE INCREMENTED TO THIS
4811  015004  020112                            CMP     R1,@R2          ;DID RKDA INCREMENT CORRECTLY?
4812  015006  001405                            BEQ     2$              ;YES BRANCH
4813  015010  010137  001162                    MOV     R1,$REG0        ;GET EXPCTD RTDA
4814  015014  011237  001164                    MOV     @R2,$REG1       ;GET RKDA RECVD
4815  015020  104040                            ERROR   40              ;RKDA INCREMENTED WRONG ON DOING
4816                                                                    ;A READ ON CYL 0, SEC 13
4817  015022  004737  021312            2$:     JSR     PC,CHKWC        ;CHECK THAT RKWC OVERFLOWED TO 0,
4818                                                                    ;IF NOT RETURN HERE.
4819  015026  104041                            ERROR   41              ;RKWC DIDN'T OURFLO AFTER
4820                                                                    ;A READ OF 200 WORDS
4821
4222  015030                            3$:
4823  015030  012746  000340                    MOV     #340,-(SP)
4824  015034  012746  015042                    MOV     #64$,-(SP)
4825  015040  000002                            RTI
4826  015042                            64$:
4827  015042  022714  033736                    CMP     #OUTBUF+400,@R4 ;DID RKBA INCREMENT CORRECTLY?
4828  015046  001406                            BEQ     TST40           ;;YES, EXIT
4829  015050  012737  033736  001162            MOV     #OUTBUF+400,$REG0 ;GET EXPCT RKBA
4830  015056  011437  001164                    MOV     @R4,$REG1       ;GET RKBA RECVD
4831  015062  104042                            ERROR   42              ;RKBA DID NOT INCREMENT CORRECTLY
4832                                                                    ;AFTER A READ OF 200 WORDS
4833
4834                                    ;;**********************************************************************
4835                                    ;*TEST 40        CHECK THAT RK11 INTERRUPTS AT BR5 ONLY
4836                                            ;*THIS TEST CHECKS THAT RK11 CAN ITERRUPT AT BR5 ONLY.  IF IT
4837                                            ;*INTERRUPTS AT A LEVEL HIGHER THAN BR5 AN ERROR IS INDICATED.
```

```
                                        ;*IF IT DOES NOT INTERRUPT AT BR5 ORLOWER THEN ALSO AN
                                        ;*ERROR IS INDICATED.  IF FOR SOME REASON THE INTERRUPT
                                        ;*LEVEL IS CHANGED FROM BR5, THEN CONTENTS OF RKPRI WILL
                                        ;*HAVE TO BE CHANGED ACCORDINGLY AND STILL TEXT WILL
                                        ;*CHECK FOR THIS BR LEVEL.
                                ;;**************************************************************
015064  000004                TST40:  SCOPE
015066  104413                        CNT.RESET               ;GO, DO CONTROL RESET
                                                                ;THIS IS A CALL FOR THE 'CNTRL-
                                                                ;RESET' ROUTINE. A CONTROL RESET IS
                                                                ;ISSUED AND AFTER A CERTAIN TIME
                                                                ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                ;AN ERROR IS REPORTED. NOTE THAT
                                                                ;THE PC IN ERROR MESSAGE IS THE
                                                                ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                ;THIS IS A VERY BASIC ERR& IF IT
                                                                ;OCCURS GO BACK TO TEST 10
015070  104421                        TST.SIN                 ;CHECK IF SIN IS SET, IF SET
                                                                ;DO DRIVE RESET TO CLEAR IT
015072  012737  015126  001110        MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
                                                                ;ON ERROR (SW 9)
015100  013700  001332                MOV     RKCS,R0
015104  013777  001350  164226        MOV     DRIVAD,@RKDA
015112  012701  000007                MOV     #7,R1           ;PRIORITY LEVEL 7
015116  012702  000340                MOV     #340,R2         ;BR LEVEL 7 FOR PSW
015122  013703  001400                MOV     RKPRI,R3        ;NOTE, IF RK11 INTERRUPT LEVEL IS
                                                                ;CHANGED FROM 5 TO ANY OTHER LEVEL
                                                                ;THEN CHANGE CONTENTS OF 'RKPRI'
                                                                ; ACCORDINGLY
015126  013704  001402        1$:     MOV     RKVEC,R4
015132  012724  015240                MOV     #3$,(R4)+       ;SET UP ADRES FOR RK11 TO INTERUPT
015136  012714  000340                MOV     #340,(R4)       ;SET UP PSW ON INTERUPT
015142  010246                        MOV     R2,-(SP)                ;SET PROCESSOR PRIORITY LEVEL AS
015144  012746  015152                MOV     #4$,-(SP)
015150  000002                        RTI
015152                          4$:
                                                                ;INDICATED BY R2
015152  012710  000100                MOV     #100,@R0        ;SET THE IDE BIT
015156  012705  177760                MOV     #-20,R5         ;WAIT FOR THE RK11 INTERRUPT
015162  005205                        INC     R5              ;WAITING TIME=78 US FOR 11/20
015164  001376                        BNE     .-2             ;13 US FOR 11/45
015166  020203                        CMP     R2,R3           ;WAS THE CPU PRIORITY LEVEL LESS THAN
015170  003005                        BGT     2$              ;THE RK11 LEVEL?  IF YES, RK11
                                                                ;SHOULD HAVE INTERRUPTED. ERROR,
                                                                ;IF IT DID NOT
015172  010137  001162                MOV     R1,$REG0        ;GET CPU BR LEVEL
015176  011037  001164                MOV     @R0,$REG1       ;GET RKCS
015202  104103                        ERROR   103             ;THOUGH CPU LEVEL WAS LESS THAN
                                                                ;THE RK11 LEVEL (5), RK11 DID NOT
                                                                ;INTERRUPT
015204  005010                2$:     CLR     @R0             ;CLEAR RKCS
015206  062702  177740                ADD     #-40,R2         ;DECREASE THE PRIORITY LEVEL (FOR
                                                                ;CPU) BY 1
015212  005301                        DEC     R1              ;CPU WILL B AT THIS LEVEL
015214  001344                        BNE     1$              ;LUP BAK & CHK FOR THIS BR LEVEL.
                                                                ;DONE WITH CHKING FOR ALL LEVELS.
015216  012777  004600  164156        MOV     #BADINT,@RKVEC  ;RESTORE UNEXPECTED RK11 INTERRJPT
```

```
                                                        ;VECTOR
                        MOV     #340,-(SP)
                        MOV     #64$,-(SP)
                        RTI
                64$:
                        BR      TST41           ;;EXIT,TO NXT TST

                3$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
                        MOV     #BADINT,@RKVEC  ;RESTORE UNEXPECTED RK11 INTERRUPT
                                                ;VECTOR
                        CMP     R2,R3           ;IF THIS INTERRUPT OCCURED WHEN
                        BLE     2$              ;CPU LEVEL WAS LESS THAN THE
                                                ;RK11 PRIORITY LEVEL (5) THEN IT IS
                                                ;OK.  IF NOT SO, ERROR
                        MOV     R1,$REG0        ;GET CPU BR LEVEL
                        MOV     @R0,$REG1       ;GET RKCS
                        ERROR   104             ;RK11 INTERRUPTED WHEN THE CPU
                                                ;LEVEL (AS POINTED BY R1) WAS
                                                ;HIGHER OR SAME AS THE RK11
                                                ;LEVEL (5)
                        BR      2$              ;GO BACK & CHK THE NXT LEVEL

;;**************************************************************
;*TEST 41        SIMULATE & CHECK 'OVR' ERROR
                ;*THIS TEST SIMULATES OVERRUN ERROR AND CHECKS IF THE OVR
                ;*BIT IN RKER GETS SET.  THEN IT IS CLEARED USING CNTRL RESET
                ;*& CHECKED THAT IT WAS CLEARED.  OVR CONDITION IS SIMULATED
                ;*BY TRYING TO READ 401(OCTAL) WORDS FROM LAST CYLINDER(312).
                ;*LAST SECTOR (13), SURFACE 1.
;;**************************************************************
TST41:  SCOPE
                CNT.RESET                       ;GO, DO CONTROL RESET
                                                ;THIS IS A CALL FOR THE 'CNTRL-
                                                ;RESET' ROUTINE. A CONTROL RESET IS
                                                ;ISSUED AND AFTER A CERTAIN TIME
                                                ;IF THE 'CNTRL RDY' DOES NOT SET
                                                ;AN ERROR IS REPORTED. NOTE THAT
                                                ;THE PC IN ERROR MESSAGE IS THE
                                                ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                ;THIS IS A VERY BASIC ERR& IF IT
                                                ;OCCURS GO BACK TO TEST 10
                TST.SIN                         ;CHECK IF SIN IS SET. IF
                                                ;SET, DO DRIVE RESET TO CLR IT
                        MOV     DRIVAD,R1       ;GET ADRES OF DRIVE
                        BIS     #14533,R1       ;SET BITS FOR LAST CYLINDER (312.
                                                ;SUR 1, LAST SECTOR (13)
                        MOV     #-401,@RKWC     ;READ 401 WORDS
                        MOV     #OUTBUF,@RKBA   ;INTO THIS MEMORY BUFFER
                        MOV     R1,@RKDA        ;FROM THIS DSK ADRES. LAST CYL.
                                                ;LAST SEC, SURFACE 1
                        MOV     #5,@RKCS                ;READ, GO

                        CLR     R2
                1$:     TSTB    @RKCS           ;DID CNTRL RDY SET?
                        BMI     2$              ;YES, BRANCH
                        INC     R2              ;NO, WAIT FOR IT
```

```
4950  015346  001373                          BNE     1$              ;IF WAITED LONG, REPORT ERROR MESSAGE BECAUSE
4951                                                                  ;OVR SHOULD HAVE SET HE CAUSING
4952                                                                  ;CNTRL RDY TO SET BY NOW
4953  015350  017737  163760  001166          MOV     @RKWC,$REG2
4954  015356  004737  021004                  JSR     PC,GT2RG        ;GO, GET RKCS, ER
4955  015362  104002                          ERROR   2               ;CNTRL RDY DID NOT SET AFTER DOING
4956                                                                  ;AN OVR READ.  HE SHOULD HAVE OCCURED
4957                                                                  ;SETTING CNTRL RDY (HE BECAUSE OF
4958                                                                  ;OVR CONDITIONS)
4959  015364  032777  040000  163736  2$:     BIT     #40000,@RKER    ;DID OVR BIT SET IN RKER?
4960  015372  001006                          BNE     3$
4961  015374  004737  021004                  JSR     PC,GT2RG        ;GET RKCS, ER
4962  015400  012737  040000  001166          MOV     #40000,$REG2    ;THIS BIT (OVR) DID NOT SET.
4963  015406  104105                          ERROR   105             ;OVR ERROR BIT DID NOT SET IN RKER
4964                                                                  ;ON SIMULATING OVR CONDITIONS
4965  015410  022777  140204  163714  3$:     CMP     #140204,@RKCS   ;DID HE & ERR SET WHEN OVR SET IN RKER?
4966  015416  001403                          BEQ     4$              ;YES, BRANCH
4967  015420  004737  021004                  JSR     PC,GT2RG        ;GET RKCS, ER
4968  015424  104106                          ERROR   106             ;HE OR ERR BIT DID NOT SET IN RKCS WHEN
4969                                                                  ;AN OVR ERROR WAS SIMULATED
4970                                                                  ;CLEAR OVER, ERR, HE BITS
4971  015426  104413                  4$:     CNT.RESET               ;GO, DO CONTROL RESET
4972                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
4973                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
4974                                                                  ;ISSUED AND AFTER A CERTAIN TIME
4975                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
4976                                                                  ;AN ERROR IS REPORTED. NOTE THAT
4977                                                                  ;THE PC IN ERROR MESSAGE IS THE
4978                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
4979                                                                  ;THIS IS A VERY BASIC ERR& IF IT
4980                                                                  ;OCCURS GO BACK TO TEST 10
4981  015430  004737  021352                  JSR     PC,CHKECLR      ;CHECK IF 'OVR' BIT WAS CLEARED BY
4982                                                                  ;CON.RESET, IF NOT RETURN HERE.
4983  015434  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR OVR
4984                                                                  ;BIT IN RKER
4985  015436  004737  021376          5$:     JSR     PC,CHKCCLR      ;CHECK IF 'ERR' & 'HE' BIT GOT CLEARED BY
4986                                                                  ;CON.RESET, IF NOT RETURN HERE.
4987  015442  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
4988                                                                  ;HE OR ERR BIT IN RKCS.
4989  015444  004737  021500          6$:     JSR     PC,DRESET       ;GO DO DRIVE RESET
4990  015450  104026                          ERROR   26              ;R/W/S RDY DIDN'T SET
4991                                                                  ;AFTER THE ABOVE DRIVE RESET
4992
4993                          ;;*****************************************************************
4994                          ;*TEST 42       SIMULATE & CHECK PGE ERROR
4995                                  ;*THIS TEST SIMULATES 'PROGRAMMING ERROR' & CHECKS IF IT IS
4996                                  ;*DETECTED BY PGE BIT IN RKER.  THEN A CNTRL RESET IS DONE &
4997                                  ;*IT IS CHECKED IF PGE BIT WAS CLEARED.  IT IS ALSO CHECKED IF
4998                                  ;*THE SETTING & CLEARING OF PGE BIT SETS & CLEARS HE, ERR
4999                                  ;*BITS IN RKCS.
5000                          ;;*****************************************************************
5001  015452  000004          *ST42:  SCOPE
5002  015454  104413                  CNT.RESET               ;GO, DO CONTROL RESET
5003                                                          ;THIS IS A CALL FOR THE 'CNTRL-
5004                                                          ;RESET' ROUTINE. A CONTROL RESET IS
5005                                                          ;ISSUED AND AFTER A CERTAIN TIME
```

```
                                                                ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                ;AN ERROR IS REPORTED. NOTE THAT
                                                                ;THE PC IN ERROR MESSAGE IS THE
                                                                ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                ;THIS IS A VERY BASIC ERR& IF IT
                                                                ;OCCURS GO BACK TO TEST 10
015456  104421                  TST.SIN                       ;GO CHECK IF SIN IS SET, IF
                                                              ;SET DO DRIVE RESET TO CLR IT
015460  013701  001330          MOV     RKER,R1
015464  013777  001350  163646  MOV     DRIVAD,@RKDA          ;ADRES THE DRIVE, CYLINDER 0

015472  012777  002011  163632  MOV     #2011,@RKCS           ;SEEK, GO WITH FMT SET
                                                              ;THIS IS A PGE SIMULATION
015500  104414                  CNT.RDY                       ;THIS IS A CALL FOR 'CN.RDY'
                                                              ;ROUTINE WHICH WAITS FOR CNT
                                                              ;RDY TO SET. IF CNTRL RDY DOES
                                                              ;NOT SET WITHIN 883 MS/ 11-20
                                                              ;(176 MS FOR 11-45 WITH BIPOLAR,
                                                              ;AN ERROR IS REPORTED
015502  032711  004000          BIT     #4000,@R1             ;DID PGE BIT IN RKER SET?
015506  001006                  BNE     1$                    ;YES, BRANCH
015510  012737  004000  001166  MOV     #4000,$REG2           ;THIS BIT IN RKER (PGE) DID NOT SET
015516  004737  021004          JSR     PC,GT2RG              ;GO GET RKCS, ER FOR MESAGE
015522  104105                  ERROR   105                   ;PGE BIT DID NOT SET IN RKER
                                                              ;ON SIMULATION OF PGE CONDITION
                                                              ;$REG2 CONTAINS THE RKER BIT .PGE.
                                                              ;THAT SHOULD HAVE SET.
015524  022777  142210  163600  1$:     CMP     #!142210,@RKCS ;DID HE & ERR BITS SET?
015532  001403                  BEQ     2$                    ;YES, BRANCH
015534  004737  021004          JSR     PC,GT2RG              ;GO, GET RKCS, ER
015540  104106                  ERROR   106                   ;HE OR ERR BIT DID NOT SET WHEN
                                                              ;PGE SET IN RKER.
                                                              ;CLEAR PGE, HE, ERR BITS
015542  104413                  2$:     CNT.RESET             ;GO, DO CONTROL RESET
                                                              ;THIS IS A CALL FOR THE 'CNTRL-
                                                              ;RESET' ROUTINE. A CONTROL RESET IS
                                                              ;ISSUED AND AFTER A CERTAIN TIME
                                                              ;IF THE 'CNTRL RDY' DOES NOT SET
                                                              ;AN ERROR IS REPORTED. NOTE THAT
                                                              ;THE PC IN ERROR MESSAGE IS THE
                                                              ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                              ;THIS IS A VERY BASIC ERR& IF IT
                                                              ;OCCURS GO BACK TO TEST 10
015544  004737  021352          JSR     PC,CHKECLR            ;CHECK IF 'PGE' BIT GOT CLEARED BY
                                                              ;CONTROL RESET, IF NOT RETURN HERE.
015550  104102                  ERROR   102                   ;CNTRL RESET DID NOT CLEAR
                                                              ;PGE BIT IN RKER
015552  004737  021376          3$:     JSR     PC,CHKCCLR    ;CHECK IF 'ERR' BITGOT CLEARED BY
                                                              ;CON.RESET, IF NOT RETURN HERE.
015556  104102                  ERROR   102                   ;RKCS BITS HE OR ERR DID NOT
                                                              ;GET CLEARED BY CNTRL RESET

                        ;;*****************************************************************
                        ;*TEST 43        SIMULATE & CHECK NXM ERROR
                                ;*THIS TEST SIMULATES A NON-EXISTENT MEMORY ERROR (NXM) AND
                                ;*CHECKS IF IT IS DETECTED BY NXM BIT OR RKER.LOCATION 760000
```

```
5062                                    ;*IS REFERENCED & IT HAPPENS TO BE A NON EXISTENT LOCATION
5063                                    ;*(FOR DIAGNOSTIC PURPOSES LIKE THIS). IT IS ALSO CHECKED
5064                                    ;*IF HE & ERR BITS ALSO SET AND ALL 3 BITS CAN BE CLEARED
5065                                    ;* BY CONTROL RESET.
5066                                    ;*****************************************************************
5067  015560  000004                TST43:  SCOPE
5068  015562  104413                        CNT.RESET                       ;GO, DO CONTROL RESET
5069                                                                        ;THIS IS A CALL FOR THE 'CNTRL-
5070                                                                        ;RESET' ROUTINE. A CONTROL RESET IS
5071                                                                        ;ISSUED AND AFTER A CERTAIN TIME
5072                                                                        ;IF THE 'CNTRL RDY' DOES NOT SET
5073                                                                        ;AN ERROR IS REPORTED. NOTE THAT
5074                                                                        ;THE PC IN ERROR MESSAGE IS THE
5075                                                                        ;PC WHERE 'CNT.RESET' IS LOCATED.
5076                                                                        ;THIS IS A VERY BASIC ERR& IF IT
5077                                                                        ;OCCURS GO BACK TO TEST 10
5078  015564  104421                        TST.SIN                         ;GO CHECK IF SIN IS SET
5079                                                                        ;IF SET DO DRIVE RESET TO CLR IT
5080  015566  005002                        CLR     R2
5081  015570  013700  001332                MOV     RKCS,R0
5082  015574  012777  177777  163532        MOV     #-1,@RKWC               ;WRITE CHECK 1 WORD
5083  015602  012777  160000  163526        MOV     #160000,@RKBA           ;AT THIS BUS ADRES
5084  015610  013777  001350  163522        MOV     DRIVAD,@RKDA            ;WITH THIS DISK ADRES (CYL 0, SEC 0)
5085  015616  012710  000067                MOV     #67,@R0                 ;WRT CHK, GO, MEX BITS SET
5086  015622  105777  163504          1$:   TSTB    @RKCS                   ;DID CNTRL RDY SET AS A RESULT OF HE?
5087  015626  100410                        BMI     2$                      ;YES, BRANCH
5088  015630  005202                        INC     R2                      ;WAITED LONG ENOUGH?
5089  015632  001373                        BNE     1$                      ;IF NOT LUP BAK & WAIT
5090  015634  004737  021004                JSR     PC,GT2RG                ;GET RKCS, ER
5091  015640  017737  163470  001166        MOV     @RKWC,$REG2             ;GET RKWC
5092  015646  104113                        ERROR   113                     ;CNTRL RDY DID NOT SET ON DOING
5093                                                                        ;A WRT CHK WITH A NXM LOCATION.
5094                                                                        ;THIS HE SHOULD HAVE SET THE
5095                                                                        ;CNTRL RDY BIT IN RKCS
5096  015650  032777  002000  163452  2$:   BIT     #2000,@RKER             ;DID NXM BIT IN RKER SET?
5097  015656  001006                        BNE     3$                      ;YES, BRANCH
5098  015660  004737  021004                JSR     PC,GT2RG                ;GO GET RKCS, RKER
5099  015664  012737  002000  001166        MOV     #2000,$REG2             ;THIS BIT (NXM) DID NOT SET IN RKER
5100  015672  104105                        ERROR   105                     ;NXM BIT DID NOT SET IN RKER ON
5101                                                                        ;SIMULATING NXM CONDITION.
5102  015674  022710  140266          3$:   CMP     #140266,@R0             ;DID HE & ERR BIT SET?
5103  015700  001403                        BEQ     4$                      ;YES, BRANCH
5104  015702  004737  021004                JSR     PC,GT2RG                        ;GO, GET RKCS, RKER
5105  015706  104106                        ERROR   106                     ;HE OR ERR BIT DID NOT SET WHEN
5106                                                                        ;NXM ERROR WAS SIMULATED
5107                                                                        ;CLEAR NXM, HE, ERR BITS
5108  015710  104413                4$:   CNT.RESET                       ;GO, DO CONTROL RESET
5109                                                                        ;THIS IS A CALL FOR THE 'CNTRL-
5110                                                                        ;RESET' ROUTINE. A CONTROL RESET IS
5111                                                                        ;ISSUED AND AFTER A CERTAIN TIME
5112                                                                        ;IF THE 'CNTRL RDY' DOES NOT SET
5113                                                                        ;AN ERROR IS REPORTED. NOTE THAT
5114                                                                        ;THE PC IN ERROR MESSAGE IS THE
5115                                                                        ;PC WHERE 'CNT.RESET' IS LOCATED.
5116                                                                        ;THIS IS A VERY BASIC ERR& IF IT
5117                                                                        ;OCCURS GO BACK TO TEST 10
```

```
5118  015712  004737  021352                  JSR     PC,CHKECLR      ;CHECK IF 'NXM' BIT GOT CLEARED BY
5119                                                                    ;CON.RESET, IF NOT RETURN HERE.
5120  015716  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
5121                                                                    ;NXM BIT IN RKER
5122  015720  004737  021376          5$:     JSR     PC,CHKCCLR      ;CHECK IF 'HE' & 'ERR' BITS GOT CLEARED
5123                                                                    ;BY CON.RESET, IF NOT RETURN HERE.
5124  015724  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
5125                                                                    ;HE OR ERR BIT IN RKCS.
5126  015726  004737  021432          6$:     JSR     PC,TSTRWS       ;GO CHECK IF R/W/S RDY IS SET &
5127                                                                    ;WAIT FOR IT.  SKIP ERROR IF IT IS SET
5128  015732  104016                          ERROR   16              ;R/W/S RDY IS NOT SET
5129
5130                                  ;;*****************************************************************
5131                                  ;*TEST 44       SIMULATE & CHECK NXD ERROR
5132                                          ;*THIS TEST SIMULATES NON-EXISTENT DISK ERROR & CHECKS IF
5133                                          ;*IT IS DETECTED BY NXD BIT OF RKER.  IF ALL EIGHT ARE PRESENT
5134                                          ;*THEN THIS TEST IS ABORTED FOR SIMULATION CANNOT BE DONE.
5135                                  ;;*****************************************************************
5136  015734  000004                  TST44:  SCOPE
5137  015736  104413                          CNT.RESET                       ;GO, DO CONTROL RESET
5138                                                                          ;THIS IS A CALL FOR THE 'CNTRL-
5139                                                                          ;RESET' ROUTINE. A CONTROL RESET IS
5140                                                                          ;ISSUED AND AFTER A CERTAIN TIME
5141                                                                          ;IF THE 'CNTRL RDY' DOES NOT SET
5142                                                                          ;AN ERROR IS REPORTED. NOTE THAT
5143                                                                          ;THE PC IN ERROR MESSAGE IS THE
5144                                                                          ;PC WHERE 'CNT.RESET' IS LOCATED.
5145                                                                          ;THIS IS A VERY BASIC ERR8 IF IT
5146                                                                          ;OCCURS GO BACK TO TEST 10
5147  015740  104421                          TST.SIN                         ;CHECK IF SIN IS SET, IF SET
5148                                                                          ;DO DRV RESET TO CLR IT
5149  015742  013700  001332                  MOV     RKCS,R0
5150  015746  012702  160000                  MOV     #160000,R2      ;ADRES DRIVE 7 TO FIND
5151                                                                  ;IF IT IS PRESENT
5152  015752  010277  163362          1$:     MOV     R2,@RKDA        ;ADRES DRIVE # POINTED TO BY R2
5153  015756  104417  000001                  DELAY   ,1              ;TIME DELAY, 7.5 US ON 11/20,
5154                                                                  ;1.5 US ON 11/45
5155  015762  105777  163340                  TSTB    @RKDS           ;IS IT PRESENT?
5156  015766  100004                          BPL     2$              ;NO, BRANCH
5157  015770  062702  160000                  ADD     #-20000,R2      ;ADRES THE NXT DRIVE IN THE
5158                                                                  ;REVERSE ORDER. I.E. 7,6,...
5159  015774  001366                          BNE     1$              ;LUP BAK & TRY TO FIND A DRIVE
5160                                                                  ;THAT'S NOT PRESENT
5161  015776  000435                          BR      TST45           ;;EXIT TO THE NXT TST
5162
5163  016000  012710  000015          2$:     MOV     #15,@R0         ;DRIVE RESET, ON A NX DRIVE
5164  016004  104417  000106                  DELAY   .106            ;TIME DELAY, 525 US ON 11/20
5165                                                                  ;105 US ON 11/45
5166  016010  105777  163314                  TSTB    @RKER           ;DID NXD BIT IN RKER SET?
5167  016014  001006                          BNE     3$              ;YES, BRANCH
5168  016016  004737  021004                  JSR     PC,GT2RG        ;GET RKCS, RKER
5169  016022  012737  000200  001166          MOV     #200,$REG2      ;THIS BIT (NXD) IN RKER DID NOT SET
5170  016030  104105                          ERROR   105             ;NXD BIT DID NOT SET ON TRYING
5171                                                                  ;TO PERFORM A FUNCTION ON A
5172                                                                  ;NON-EXISTENT DRIVE
5173                                                                  ;CHECK THAT THE JUMPER CARD CONTAINING
```

```
                                                        ;JUMPERS FOR DRIVES PRESENT IS PROPERLY
                                                        ;CONNECTED
                                                        ;NOTE THAT ON RK11C IF A DRIVE
                                                        ;IS OFFLINE BUT PHYSICALLY PRESENT
                                                        ;(IE. DRY IS CLR FOR THAT DRIVE)
                                                        ;& A FUNCTION IS INITIATED ON THAT
                                                        ;DRIVE NXD WON'T SET, BUT U WILL
                                                        ;GET ONLY A DRE,HE & ERR.
016032  022710  140214          3$:     CMP     #140214,@R0     ;DID HE & ERR SET WHEN NXD SET?
016036  001403                          BEQ     4$              ;YES BRANCH
016040  004737  021004                  JSR     PC,GT2RG        ;HE OR ERR BIT DID NOT SET
016044  104106                          ERROR   106             ;WHEN NXD WAS SIMULATED
                                                        ;CLEAR NXD, HE, ERR BITS
016046  104413                  4$:     CNT.RESET               ;GO, DO CONTROL RESET
                                                        ;THIS IS A CALL FOR THE 'CNTRL-
                                                        ;RESET' ROUTINE. A CONTROL RESET IS
                                                        ;ISSUED AND AFTER A CERTAIN TIME
                                                        ;IF THE 'CNTRL RDY' DOES NOT SET
                                                        ;AN ERROR IS REPORTED. NOTE THAT
                                                        ;THE PC IN ERROR MESSAGE IS THE
                                                        ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                        ;THIS IS A VERY BASIC ERR& IF IT
                                                        ;OCCURS GO BACK TO TEST 10
016050  004737  021352                  JSR     PC,CHKECLR      ;CHECK IF 'NXD' BIT WAS CLEARED BY
                                                        ;CON.RESET. IF NOT, RETURN HERE.
016054  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
                                                        ;NXD BIT IN RKER
016056  004737  021376          5$:     JSR     PC,CHKCCLR      ;CHECK IF 'HE' & 'ERR' BITS WERE CLEARED
                                                        ;BY CON.RESET. IF NOT RETURN HERE.
016062  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
                                                        ;HE OR ERR BIT IN RKCS
016064  004737  021432                  JSR     PC,TSTRWS       ;GO CHECK & WAIT FOR R/W/S RDY
                                                        ;TO SET.  IF SET SKIP ERROR
016070  104016                          ERROR   16              ;R/W/S SHOULD BE SET, IT'S
                                                        ;NOT

                                ;;*****************************************************************
                                ;*TEST 45        SIMULATE & CHECK NXC ERROR
                                        ;*THIS TEST SIMULATES THE NON-EXISTENT CYLINDER ERROR & CHECKS
                                        ;*IF IT IS DETECTED BY THE NXC BIT OF RKER, HE & ERR BITS
                                        ;*OF RKCS.  IT IS CHECKED IF THEY CAN BE CLEARED BY CONTROL
                                        ;*RESET
                                ;;*****************************************************************
016072  000004                  T45:    SCOPE
016074  013700  001332                  MOV     RKCS,R0
016100  012737  177773  001362  2$:     MOV     #-5,COUNT       ;ALLOW 'ERROR 133' ONLY 5 TIMES
016106  013702  001350                  MOV     DRIVAD,R2       ;GET ADRES OF DRIVE
016112  052702  014540                  BIS     #14540,R2       ;SET BITS FOR CYL 313
016116  012737  016124  001110          MOV     #3$,$LPERR      ;SET RETURN ADRES FOR
                                                        ;LUPING ON EROR (SW9)
016124  104413                  3$:     CNT.RESET               ;GO, DO CONTROL RESET
                                                        ;THIS IS A CALL FOR THE 'CNTRL-
                                                        ;RESET' ROUTINE. A CONTROL RESET IS
                                                        ;ISSUED AND AFTER A CERTAIN TIME
                                                        ;IF THE 'CNTRL RDY' DOES NOT SET
                                                        ;AN ERROR IS REPORTED. NOTE THAT
```

```
5230                                                                    ;THE PC IN ERROR MESSAGE IS THE
5231                                                                    ;PC WHERE 'CNT.RESET' IS LOCATED.
5232                                                                    ;THIS IS A VERY BASIC ERR& IF IT
5233                                                                    ;OCCURS GO BACK TO TEST 10
5234  016126  004737  021432                  JSR     PC,TSTRWS       ;GO CHECK & WAIT FOR R/W/S RDY
5235                                                                    ;TO SET. IF SET SKIP ERROR BELOW
5236  016132  104016                          ERROR   16              ;R/W/S RDY IS NOT SET
5237  016134  104421                          TST.SIN                 ;CHECK IF SIN IS SET, IF SET
5238                                                                    ;DO DRIVE RESET TO CLR IT
5239  016136  010277  163176                  MOV     R2,@RKDA        ;ADRES DRIVE, NXC CYLINDER
5240  016142  012710  000011                  MOV     #11,@RD         ;SEEK, GO TO NXC CYL
5241  016146  104412                          CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
5242                                                                    ;IF SO, SKIP THE EROR MESSAGE.
5243  016150  104021                          ERROR   21              ;SEEK WAS TRIED TO A NON EXISTENT
5244                                                                    ;CYLINDER, NXC SHOULD HAVE OCCURED
5245                                                                    ;SETTING CNTRL RDY. BUT CNTRL RDY
5246                                                                    ;DID NOT SET.
5247  016152  032777  000100  163150  9$:     BIT     #100,@RKER      ;DID NXC SET?
5248  016160  001020                          BNE     4$              ;YES, BRANCH
5249  016162  004737  021004                  JSR     PC,GT2RG        ;GO GET RKCS, ER
5250  016166  017737  163146  001166          MOV     @RKDA,$REG2     ;GET RKDA
5251  016174  104110                          ERROR   110             ;NXC DID NOT SET WHEN SEEK
5252                                                                    ;WAS TRIED TO CYLINDER AS INDICATED
5253                                                                    ;IN RKDA
5254  016176  004737  021432                  JSR     PC,TSTRWS       ;CHECK & WAIT FOR R/W/S RDY.
5255                                                                    ;IF SET SKIP ERROR
5256  016202  104016                          ERROR   16              ;R/W/S SHOULD BE SET
5257  016204  104413                          CNT.RESET               ;GO DO CONTROL RESET
5258  016206  004737  021500                  JSR     PC,DRESET       ;GO DO DRIVE RESET
5259  016212  104026                          ERROR   26              ;NXC DID NOT SET AND DRIVE MAY
5260                                                                    ;HAVE TRIED TO DO A SEEK, AFTER
5261                                                                    ;WHICH R/W/S RDY DID NOT SET
5262  016214  005237  001362                  INC     COUNT           ;ALLOW ONLY 5 MESSAGES FOR
5263  016220  001405                          BEQ     5$              ;ERROR 133
5264  016222  062702  000040          4$:     ADD     #40,R2          ;ADRES THE NXT CYL(IN NON-EXISTENT ZONE.
5265  016226  032702  017740            .     BIT     #17740,R2       ;CHKD FOR ALL NXC'S?
5266  016232  001334                          BNE     3$              ;IF NOT, LUP BAK & CHK THE NXT NXC
5267
5268  016234  032710  140000          5$:     BIT     #140000,@RD     ;DID HE & ERR BIT SET WHEN NXC BIT SET?
5269  016240  001003                          BNE     6$              ;YES, BRANCH
5270  016242  004737  021004                  JSR     PC,GT2RG        ;GET RKCS, ER
5271  016246  104106                          ERROR   106             ;HE OR ERR BIT DID NOT SET IN RKCS
5272                                                                    ;WHEN NXC ERROR WAS SIMULATED
5273                                                                    ;CLEAR HE, ERR, NXC BITS
5274  016250  104413                  6$:     CNT.RESET               ;GO, DO CONTROL RESET
5275                                                                    ;THIS IS A CALL FOR THE 'CNTRL-
5276                                                                    ;RESET' ROUTINE. A CONTROL RESET IS
5277                                                                    ;ISSUED AND AFTER A CERTAIN TIME
5278                                                                    ;IF THE 'CNTRL RDY' DOES NOT SET
5279                                                                    ;AN ERROR IS REPORTED. NOTE THAT
5280                                                                    ;THE PC IN ERROR MESSAGE IS THE
5281                                                                    ;PC WHERE 'CNT.RESET' IS LOCATED.
5282                                                                    ;THIS IS A VERY BASIC ERR& IF IT
5283                                                                    ;OCCURS GO BACK TO TEST 10.
5284  016252  004737  021352                  JSR     PC,CHKECLR      ;CHECK IF 'NXC' BIT WAS CLEARED BY
5285                                                                    ;CON.RESET. IF NOT, RETURN HERE.
```

```
5286  016256  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
5287                                                                  ;NXC BIT IN RKER
5288  016260  032710  140000          7$:     BIT     #140000,@R0     ;DID HE & ERR BITS GET CLEARED?
5289  016264  001405                          BEQ     TST46           ;;YES, EXIT
5290  016266  010037  001162                  MOV     R0,$REG0        ;GET ADRES OF RKCS
5291  016272  011037  001164                  MOV     @R0,$REG1       ;GET RKCS CONTENTS
5292  016276  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
5293                                                                  ;HE OR ERR BIT IN RKCS
5294
5295                                  ;;**************************************************************
5296                                  ;*TEST 46        SIMULATE & CHECK NXS ERROR
5297                                          ;*THIS TEST SIMULATES NON-EXISTENT SECTOR ERROR & CHECKS THAT
5298                                          ;*IT IS DETECTED BY NXS BIT OF RKER.  IT IS CHECKED THAT
5299                                          ;*WHEN NXS SETS HE & ERR OF RKER ALSO SETS, AND ALL THREE
5300                                          ;*CAN BE CLEARED BY CONTROL RESET.
5301                                  ;;**************************************************************
5302  016300  000004                  TST46:  SCOPE
5303  016302  104413                          CNT.RESET               ;GO, DO CONTROL RESET
5304                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
5305                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
5306                                                                  ;ISSUED AND AFTER A CERTAIN TIME
5307                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
5308                                                                  ;AN ERROR IS REPORTED. NOTE THAT
5309                                                                  ;THE PC IN ERROR MESSAGE IS THE
5310                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
5311                                                                  ;THIS IS A VERY BASIC ERR& IF IT
5312                                                                  ;OCCURS GO BACK TO TEST 10
5313  016304  013700  001332                  MOV     RKCS,R0
5314  016310  013777  001350  163022          MOV     DRIVAD,@RKDA    ;GET ADRES OF DRIVE
5315  016316  052777  000014  163014          BIS     #14,@RKDA       ;SET BITS FOR SECTOR 12 (DECIMAL
5316  016324  012777  177777  163002          MOV     #-1,@RKWC       ;READ 1 WORD
5317  016332  012777  033336  162776          MOV     #OUTBUF,@RKBA   ;INTO THIS BUS ADRES
5318  016340  012710  000005                  MOV     #5,@R0          ;READ, GO (FROM NX SECTOR)
5319  016344  104414                          CNT.RDY                 ;THIS IS A CALL FOR 'CN.RDY'
5320                                                                  ;ROUTINE WHICH WAITS FOR CNT
5321                                                                  ;RDY TO SET. IF CNTRL RDY DOES
5322                                                                  ;NOT SET WITHIN 883 MS/ 11-20
5323                                                                  ;(176 MS FOR 11-45 WITH BIPOLAR)
5324                                                                  ;AN ERROR IS REPORTED
5325                                                                  ;NXS ERROR SHOULD OCCUR NOW
5326  016346  017702  162756                  MOV     @RKER,R2
5327  016352  032702  000040                  BIT     #40,R2          ;DID NXS BIT SET IN RKER?
5328  016356  001006                          BNE     1$              ;YES, BRANCH
5329  016360  004737  021004                  JSR     PC,GT2RG        ;GO GET RKCS, RKER
5330  016364  012737  000040  001166          MOV     #40,$REG2       ;THIS BIT (NXS) IN RKER DID NOT SET
5331  016372  104105                          ERROR   105             ;NXS BIT DID NOT SET ON SIMULATING
5332                                                                  ;NXS ERROR
5333  016374  042702  000040          1$:     BIC     #40,R2          ;MASK NXS BIT
5334  016400  001407                          BEQ     2$              ;CHECK IF ANY OTHER
5335                                                                  ;RKER BIT SET
5336  016402  012737  000040  001162          MOV     #40,$REG0       ;GET EXPCTD RKER
5337  016410  017737  162714  001164          MOV     @RKER,$REG1     ;GET RKER RECVD
5338  016416  104107                          ERROR   107             ;ONLY 'NXS' SHOULD BE SET
5339                                                                  ;IN RKER, ANOTHER RKER BIT
5340                                                                  ;WAS SET. (NOTE 'NXS' WAS
5341                                                                  ;SIMULATED)
```

```
5342  016420  022710  140204          2$:     CMP     #140204,@R0     ;DID HE & ERR BITS SET?
5343  016424  001403                          BEQ     3$              ;YES, BRANCH
5344  016426  004737  021004                  JSR     PC,GT2RG        ;GO GET RKCS, RKER
5345  016432  104106                          ERROR   106             ;HE OR ERR BIT DID NOT SET WHEN
5346                                                                  ;NXS ERROR OCCURED
5347                                                                  ;CLEAR NXS, HE, ERR BITS
5348  016434  104413                  3$:     CNT.RESET               ;GO, DO CONTROL RESET
5349                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
5350                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
5351                                                                  ;ISSUED AND AFTER A CERTAIN TIME
5352                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
5353                                                                  ;AN ERROR IS REPORTED. NOTE THAT
5354                                                                  ;THE PC IN ERROR MESSAGE IS THE
5355                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
5356                                                                  ;THIS IS A VERY BASIC ERR& IF IT
5357                                                                  ;OCCURS GO BACK TO TEST 10
5358  016436  004737  021352                  JSR     PC,CHKECLR      ;CHECK IF 'NXS' BIT WAS CLEARED BY
5359                                                                  ;CON.RESET. IF NOT, RETUEN HERE.
5360  016442  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
5361                                                                  ;NXS BIT IN RKER
5362  016444  004737  021376          4$:     JSR     PC,CHKCCLR      ;CHECL IF 'HE' & 'ERR' BITS WERE CLEARED
5363                                                                  ;BY CON.RESET. IF NOT, RETURN HERE.
5364  016450  104102                          ERROR   102             ;RKCS BITS ERR OR HE WERE NOT
5365                                                                  ;CLEARED BY CNTRL RESET
5366
5367                                  ;;**************************************************************
5368                                  ;*TEST 47        SIMULATE & CHECK WCE
5369                                          ;*THIS TEST SIMULATES A WRITE CH CK ERROR AND CHECKS THAT IT
5370                                          ;*IS DETECTED BY WCE BIT OF RKER   FOR COMPARISON IT USES
5371                                          ;*THE 256 WORDS DATA BLOCK WRITTEN ON SECTOR 0, CYLINDER 0
5372                                          ;*IN A PREVIOUS TEST.  THIS BLOCK IS COMPARED WITH THE 256 WORDS
5373                                          ;*MEMORY BUFFER STARTING AT 'OUTBUF'.  WCE IS SIMULATED BY
5374                                          ;*DROPPING A BIT FROM ONE OF THE WORDS IN THE MEMORY BUFFER.
5375                                  ;;**************************************************************
5376  016452  000004                  TST47:  SCOPE
5377  016454  013700  001332                  MOV     RKCS,R0
5378  016460  104413                          CNT.RESET               ;GO, DO CONTROL RESET
5379                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
5380                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
5381                                                                  ;ISSUED AND AFTER A CERTAIN TIME
5382                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
5383                                                                  ;AN ERROR IS REPORTED. NOTE THAT
5384                                                                  ;THE PC IN ERROR MESSAGE IS THE
5385                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
5386                                                                  ;THIS IS A VERY BASIC ERR& IF IT
5387                                                                  ;OCCURS GO BACK TO TEST 10
5388  016462  104421                          TST.SIN                 ;CHECK IF SIN IS SET, IF
5389                                                                  ;SET DO DRV-RESET TO CLR IT
5390  016464  012701  033336                  MOV     #OUTBUF,R1      ;THIS CODE SETS UP A MEMORY
5391  016470  012702  177400                  MOV     #-400,R2        ;BUFFER OF 256 WORDS STARTING
5392  016474  012703  177777                  MOV     #177777,R3      ;AT OUTBUF
5393                                                                  ;FIRST WORD  177400
5394                                                                  ;SECOND      177001
5395  016500  062703  177401          1$:     ADD     #177401,R3
5396  016504  010321                          MOV     R3,(R1)+        ;LAST WORD   000377
5397  016506  005202                          INC     R2              ;HAVE U GENERATED ALL 256 WORDS?
```

```
5398  016510  001373                          BNE     1$              ;IF NOT, LUP BAK & GENERATE NXT
5399
5400  016512  012737  170007  033354          MOV     #170007,OUTBUF+16 ;WCE WILL B SIMULATED BY DROPPING A
5401                                                                  ;BIT IN THE EIGATH WORD WHICH IS
5402                                                                  ;SUPPOSED TO B 174007
5403  016520  012777  177400  162606          MOV     #-400,@RKWC     ;WRT CHK 400 WORDS
5404  016526  012777  033336  162602          MOV     #OUTBUF,@RKBA   ;STARTING AT THIS BUS ADRES
5405  016534  013777  001350  162576          MOV     DRIVAD,@RKDA    ;WITH THIS DISK ADRES, SEC 0, CYL 0
5406  016542  012710  000007                  MOV     #7,@RD          ;WRT CHK, GO
5407
5408  016546  104412                          CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
5409                                                                  ;IF SO, SKIP THE ERROR MESSAGE.
5410  016550  104065                          ERROR   65              ;CNTRL RDY DID NOT SET
5411                                                                  ;AFTER WRT CHK
5412  016552  032777  000001  162550  3$:     BIT     #1,@RKER        ;DID WCE BIT SET?
5413  016560  001006                          BNE     4$
5414  016562  004737  021004                  JSR     PC,GT2RG        ;GO, GET RKCS, RKER
5415  016566  012737  000001  001166          MOV     #1,$REG2        ;THIS BIT (WCE) DID NOT SET
5416  016574  104105                          ERROR   105             ;WCE DID NOT SET ON SIMULATING
5417                                                                  ;WCE CONDITIONS
5418  016576  022710  100206          4$:     CMP     #100206,@RD     ;IS RKCS CORRECT?
5419  016602  001403                          BEQ     5$              ;YES, BRANCH
5420  016604  004737  021004                  JSR     PC,GT2RG        ;GO, GET RKCS, RKER
5421  016610  104106                          ERROR   106             ;HE OR ERR BIT DID NOT SET WHEN
5422                                                                  ;WCE WAS SIMULATED
5423  016612  104413                  5$:     CNT.RESET               ;CNTRL RESET
5424  016614  004737  021352                  JSR     PC,CHKECLR      ;WAS 'WCE' BIT CLEARED?
5425                                                                  ;IF NOT, RETURN HERE.
5426  016620  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
5427                                                                  ;WCE BIT IN RKER
5428  016622  004737  021376          6$:     JSR     PC,CHKCCLR      ;CHECK IF 'ERR' BIT WAS CLEARED. IF
5429                                                                  ;NOT RETURN HERE.
5430  016626  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
5431                                                                  ;RKCS
5432
5433                                  ;;******************************************************************
5434                                  ;*TEST 50       CHECK THAT SSE STOPS ALL CONTROL ACTION ON SOFT ERROR
5435                                          ;*THIS TEST CHECKS THAT WHEN 'STOP ON SOFT ERROR' BIT IS SET IN
5436                                          ;*RKCS AND A SOFT ERROR IS ENCOUNTERED ALL CONTROL ACTION WILL
5437                                          ;*STOP AT THE END OF THE CURRENT SECTOR IF IDE BIT IS CLEAR.
5438                                          ;*SOFT ERROR IS SIMULATED BY A WCE AS IN THE PREVIOUS
5439                                          ;*TEST. THE PREVIOUS TEST & THE TEST WHICH WRITES DATA
5440                                          ;*BLOCK ON CYLINDER 0, SECTOR 0, SHOULD BE DONE PRIOR
5441                                          ;*TO THIS TEST. A TWO SECTOR 'WRT CHK' WILL BE DONE.
5442                                          ;*CONTROL ACTION SHOULD STOP AFTER THE FIRST SECTOR DURING
5443                                          ;*WHICH A SOFT ERROR IS SIMULATED.
5444                                  ;;******************************************************************
5445  016630  000004                  TST50:  SCOPE
5446  016632  104413                          CNT.RESET               ;GO, DO CONTROL RESET
5447                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
5448                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
5449                                                                  ;ISSUED AND AFTER A CERTAIN TIME
5450                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
5451                                                                  ;AN ERROR IS REPORTED. NOTE THAT
5452                                                                  ;THE PC IN ERROR MESSAGE IS THE
5453                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
```

```
                                                                ;THIS IS A VERY BASIC ERR& IF IT
                                                                ;OCCURS GO BACK TO TEST 10
016634  104421                  TST.SIN                         ;CHECK IF SIN IS SET, IF
                                                                ;SET DO DRIVE RESET TO CLR IT
016636  013700  001332          MOV     RKCS,R0
016642  012737  170007  033354  MOV     #170007,OUTBUF+16 ;WCE IS SIMULATED BY DROPPING A BIT
                                                                ;IN THE EIGTH WORD (WHICH IS ACTUALLY
                                                                ;174007). NOTE THAT 256 WORD MEMORY
                                                                ;BUFFER IS CREATED IN THE PREVIOUS TEST.
016650  013701  001350          MOV     DRIVAD,R1
016654  012777  177000  162452  MOV     #-1000,@RKWC    ;WRT CHK 1000 (OCTAL) WORDS, 2 SECTORS
016662  012777  033336  162446  MOV     #OUTBUF,@RKBA   ;FROM THIS BUS ADRES
016670  010177  162444          MOV     R1,@RKDA        ;WITH THIS DISK ADRES, SEC 0, CYL 0
016674  012710  000407          MOV     #407,@R0        ;WRT CHK, GO, SSE
016700  104412                  CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
                                                        ;IF SO, SKIP THE EROR MESSAGE.
016702  104065                  ERROR   65              ;CNTRL RDY DID NOT SET AFTER WRT
                                                        ;CHK. A SOFT ERROR (WCE) IN
                                                        ;SECTOR 0 SHOULD HAVE STOPPED
                                                        ;ALL CONTROL ACTION.
016704  022777  000001  162416  2$:     CMP     #1,@RKER        ;CHECK ONLY 'WCE' BIT SHOULD
                                                                ;BE SET?
016712  001407                  BEQ     3$              ;YES, BRANCH
016714  012737  000001  001162  MOV     #1,$REG0        ;GET EXPCTD RKER
016722  017737  162402  001164  MOV     @RKER,$REG1     ;GET RKER RECVD
016730  104107                  ERROR   107             ;ONLY BIT 'WCE' OF RKER
                                                        ;SHOULD BE SET (WCE WAS
                                                        ;SIMULATED ABOVE).  ERROR
                                                        ;IF IT'S NOT
016732  005201          3$:     INC     R1              ;CHECK THAT RKDA INCREMENTED BY
016734  020177  162400          CMP     R1,@RKDA        ;1 SECTOR ONLY IMPLYING THAT
                                                        ;CNTRL ACTION DID STOP AFTER
                                                        ;SOFT ERROR IN SECTOR 0
016740  001406                  BEQ     TST51           ;;YES, EXIT
016742  010137  001162          MOV     R1,$REG0        ;GET EXPCTD RKDA
016746  017737  162366  001164  MOV     @RKDA,$REG1     ;GET RKDA RECVD
016754  104070                  ERROR   70              ;RKDA SHOULD HAVE INCRMNTD
                                                        ;BY 1 SECTOR ONLY, IT DIDN'T.
                                                        ;WCE WAS SIMULATED IN THE
                                                        ;FIRST SECTOR & A WRT CHK
                                                        ;OF 2 SECTORS WAS ISSUED.
                                                        ;CONTROLLER SHOULD STOP AFTER
                                                        ;DETECTING WCE IN THE FIRST
                                                        ;SECTOR.  HENCE RKDA SHOULD
                                                        ;INCREMENT BY 1 SECTOR ONLY


;;**************************************************************
;*TEST 51        CHECK THAT RK11 INTERRUPTS ON SOFT ERROR WHEN SSE & IDE ARE SET
;*THIS TEST CHECKS WHEN SSE BIT IS SET WITH IDE SET AND A SOFT
;*ERROR OCCURS, THEN ALL CONTROL ACTION WILL STOP AND A BUS
;*REQUEST (INTERRUPT) WILL OCCUR AT THE END OF THE CURRENT
;*SECTOR.  SOFT ERROR IS SIMULATED BY WCE AS IN PREVIOUS
;*TEST.  PREREQUISITES FOR THIS TEST ARE THE SAME AS THOSE
;*FOR THE PREVIOUS TEST.
;;**************************************************************
```

```
5510  016756  000004                  TST51:  SCOPE
5511  016760  104413                          CNT.RESET                ;GO, DO CONTROL RESET
5512                                                                   ;THIS IS A CALL FOR THE 'CNTRL-
5513                                                                   ;RESET' ROUTINE. A CONTROL RESET IS
5514                                                                   ;ISSUED AND AFTER A CERTAIN TIME
5515                                                                   ;IF THE 'CNTRL RDY' DOES NOT SET
5516                                                                   ;AN ERROR IS REPORTED. NOTE THAT
5517                                                                   ;THE PC IN ERROR MESSAGE IS THE
5518                                                                   ;PC WHERE 'CNT.RESET' IS LOCATED.
5519                                                                   ;THIS IS A VERY BASIC ERR& IF IT
5520                                                                   ;OCCURS GO BACK TO TEST 10
5521  016762  104421                          TST.SIN                  ;CHECK IF SIN IS SET, IF
5522                                                                   ;SET DO DRIVE RESET TO CLR IT
5523  016764  012737  170007  033354          MOV     #170007,OUTBUF+16 ;WCE IS SIMULATED BY DROPPING A BIT
5524                                                                   ;IN THE EIGHTH WORD (WHICH IS 174007.
5525                                                                   ;NOTE THAT THE 256 WORD MEMORY
5526                                                                   ;BUFFER (STARTING AT OUTBUF) IS
5527                                                                   ;CREATED IN A PREVIOUS TEST.
5528  016772  013701  001350                  MOV     DRIVAD,R1
5529  016776  012777  177000  162330          MOV     #-1000,@RKWC     ;WRT CHK 1000 (OCTAL) WORDS, 2 SECTORS
5530  017004  012777  033336  162324          MOV     #OUTBUF,@RKBA    ;FROM THIS BUS ADRES
5531  017012  010177  162322                  MOV     R1,@RKDA         ;WITH THIS DISK ADRES, SEC 0, CYL 0
5532  017016  013700  001402                  MOV     RKVEC,R0
5533  017022  012720  017054                  MOV     #1$,(R0)+        ;SET UP INTERRUPT VECTOR FOR Rk11
5534  017026  012710  000340                  MOV     #340,@R0         ;SET PSW ON INTERRUPT
5535  017032  012777  000507  162272          MOV     #507,@RKCS       ;WRT CHK, GO, SSE, IDE SET
5536  017040  104420  177777                  WAT.INT,177777           ;WAIT FOR INTERRUPT FROM Rk11
5537                                                                   ;TIME=485 MS FOR 11/20,
5538                                                                   ;97 MS FOR 11/45
5539  017044  004737  021004                  JSR     PC,GT2RG                 ;11/05
5540  017050  104111                          ERROR   111              ;RK11 DID NOT INTERRUPT AFTER A SOFT
5541                                                                   ;ERROR (SIMULATED) IN  SECTOR 0
5542  017052  000417                          BR      2$
5543
5544  017054  022626                  1$:     CMP     (SP)+,(SP)+      ;RESTORE STACK POINTER (FROM RK11 INTRLPT)
5545  017056  022626                          CMP     (SP)+,(SP)+      ;POP STACK (FROM WAT.INT)
5546  017060  012777  004600  162314          MOV     #BADINT,@RKVEC   ;RESTORE RK11 INTERRUPT VECTOR
5547                                                                   ;ADRES FOR UNEXPECTED INTERRUPTS
5548  017066  005201                          INC     R1
5549  017070  020177  162244                  CMP     R1,@RKDA         ;CHECK THAT RKDA INCREMENTED
5550                                                                   ;BY ONLY 1 SECTOR BEFORE INTERRUPT
5551                                                                   ;OCCURRED
5552  017074  001406                          BEQ     2$
5553  017076  010137  001162                  MOV     R1,$REG0         ;GET EXPCTD RKDA
5554  017102  017737  162232  001164          MOV     @RKDA,$REG1      ;GET RKDA RECVD
5555  017110  104003                          ERROR   3                ;RKDA SHOULD HAVE INCREMENTED BY
5556                                                                   ;1 SECTOR ONLY, IF ALL CNTRL ACTION
5557                                                                   ;HAD STOPPED AFTER SOFT ERROR
5558                                                                   ;(SIMULATED) IN SECTOR 0. IT DID NOT.
5559  017112                          2$:
5560  017112  012746  000340                  MOV     #340,-(SP)
5561  017116  012746  017124                  MOV     #64$,-(SP)
5562  017122  000002                          RTI
5563  017124                          64$:
5564  017124  005077  162202                  CLR     @RKCS            ;CLEAR THE IDE BIT
5565
```

```
5566
5567
5568                                    ;*****************************************************************
5569                                    ;*TEST 52       CHECK THE MEX BITS IN RKCS
5570                                    ;*THIS TEST CHECKS OUT THE EXTENDED MEMORY BITS OF THE RKCS.
5571                                    ;*THE RKBA IS SET TO 177776 AND A ONE WORD WRITE CHECK IS TRIED.
5572                                    ;*THIS COULD GIVE RISE TO NXM ERROR, BUT EVEN THEN THE RKBA
5573                                    ;*SHOULD OVERFLOW INTO THE MEX BITS. SIMILIARLY IT IS CHECKED
5574                                    ;*THAT THE OVERFLOWING BIT CAN MAKE THE MEX BITS COUNT
5575                                    ;*01,10,11,00.
5576                                    ;*****************************************************************
5577  017130  000004                    TST52   SCOPE
5578  017132  013700  001332                    MOV     RKCS,R0
5579  017136  012701  177774                    MOV     #-4,R1          ;SET UP THE COUNT
5580  017142  005002                            CLR     R2              ;INITIALIZE MEX BITS TO 0 SET IN RKCS
5581  017144  012737  017152  001110            MOV     #1$,$LPERR      ;SET RETURN ADRES FOR
5582                                                                    ;LUPING ON EROR (SW9)
5583  017152  104417  000142          1$:       DELAY   ,142            ;TIME DELAY
5584  017156  004737  021432                    JSR     PC,TSTRWS       ;WAIT FOR R/W/S RDY
5585  017162  104016                            ERROR   16              ;R/W/S RDY IS NOT SET
5586  017164  104413                            CNT.RESET               ;GO, DO CONTROL RESET
5587                                                                    ;THIS IS A CALL FOR THE 'CNTRL-
5588                                                                    ;RESET' ROUTINE. A CONTROL RESET IS
5589                                                                    ;ISSUED AND AFTER A CERTAIN TIME
5590                                                                    ;IF THE 'CNTRL RDY' DOES NOT SET
5591                                                                    ;AN ERROR IS REPORTED. NOTE THAT
5592                                                                    ;THE PC IN ERROR MESSAGE IS THE
5593                                                                    ;PC WHERE 'CNT.RESET' IS LOCATED.
5594                                                                    ;THIS IS A VERY BASIC ERR& IF IT
5595  017166  010210                            MOV     R2,@R0          ;OCCURS GO BACK TO TEST 10
5596  017170  012777  177777  162136            MOV     #-1,@RKWC       ;SET MEX BITS (AS IN R2) IN RKCS
5597  017176  013777  001350  162134            MOV     DRIVAD,@RKDA    ;WRT CHK 1 WORD
5598  017204  012777  177776  162124            MOV     #177776,@RKBA   ;THIS DISK ADRES, SEC 0, CYL 0
5599                                                                    ;THIS BUS ADRES. NOTE THIS BA
5600  017212  052710  000007                    BIS     #7,@R0          ;IN CONJUCTION WITH MEX BITS OF RKCS
5601                                                                    ;WRT CHK, GO
5602                                                                    ;THERE MAY BE A NXM OR WCE BUT
5603                                                                    ;WHATEVER THE CASE RKBA SHOULD
5604  017216  104412                            CHKCRDY                 ;OVERFLOW MAKING THE MEX BITS COUNT
5605                                                                    ;GO CHECK IF CONTROL RDY IS SET
5606  017220  104065                            ERROR   65              ;IF SO, SKIP THE EROR MESSAGE.
5607  017222  010205                    3$:     MOV     R2,R5           ;CNTRL RDY DID NOT SET AFTER WRT CHK
5608  017224  062705  000020                    ADD     #20,R5
5609  017230  042705  000100                    BIC     #100,R5         ;MEX BITS SHOULD INCREMENT BY 1 TO THIS
5610  017234  011004                            MOV     @R0,R4          ;MASK OUT IDE BIT POSITION, IF SET
5611  017236  042704  177717                    BIC     #177717,R4      ;GET RKCS
5612  017242  020504                            CMP     R5,R4           ;MASK OUT ALL BITS EXCEPT MEX
5613  017244  001405                            BEQ     4$              ;DID MEX BITS INCREMENT CORRECTLY?
5614  017246  010537  001162                    MOV     R5,$REG0        ;YES, BRANCH
5615  017252  010437  001164                    MOV     R4,$REG1        ;GET EXPCTD MEX BITS
5616  017256  104112                            ERROR   112             ;GET MEX BITS RECVD
5617                                                                    ;MEX BITS DID NOT INCREMENT AS
5618                                                                    ;'EXPCTD' WHEN RKBA OVERFLOWED.
5619                                                                    ;NOTE THAT BIT POSITION 4 & 5
5620                                                                    ;REFLECT MEX BITS 0 & 1 IN THE
5621  017260  017703  162044           4$:     MOV     @RKER,R3        ;ERROR MESSAGE.
                                                                        ;GET RKER
```

```
5622  017264  010305                          MOV     R3,R5
5623  017266  042703  003001                  BIC     #3001,R3        ;MASK WCE,DLT,NXM BIT, IF SET
5624  017272  001410                          BEQ     5$              ;BRANCH IF REST OF RKER CLR
5625  017274  042705  177776                  BIC     #177776,R5      ;MASK NON-WCE BITS
5626  017300  010537  001162                  MOV     R5,$REG0        ;THIS IS THE EXPCTD RKER
5627  017304  017737  162020  001164          MOV     @RKER,$REG1     ;GET RKER RECVD
5628  017312  104107                          ERROR   107             ;ERROR IN RKER.  IT SHOULD
5629                                                                  ;BE AS EXPECTED IN
5630                                                                  ;ERROR MESSAGE
5631  017314  062702  000020          5$:     ADD     #20,R2          ;INCREMENT TO NXT MEX BIT
5632  017320  005201                          INC     R1              ;HAVE U CHKD THE MEX BITS 4 TIMES?
5633  017322  001313                          BNE     1$              ;IF NOT, LUP BACK
5634
5635                                  ;;**********************************************************************
5636                                  ;*TEST 53        TRANSFER FROM DISK TO TTY
5637                                          ;* THIS TEST CHECKS THE HIGH ORDER BITS OF THE ADDRESS
5638                                          ;* LINES.  FIRST A ONE WORD (100) IS WRITTEN ON SECTOR,
5639                                          ;* 2, CYL 0.  THEN IT IS READ BACK, BUT THE NPR IS DONE
5640                                          ;* NOT TO THE MEMORY, BUT THE TELETYPE BUFFER (TKS 177560)
5641                                          ;* AND IT CHECKED THAT THE WORD WAS RECIEVED CORRECTLY.
5642                                          ;*IF IT IS NOT, AN ERROR IS REPORTED. THIS TEST IS
5643                                          ;*SKIPPED ON AN 11/05.
5644                                  ;;**********************************************************************
5645  017324  000004                  TST53:  SCOPE
5646  017326  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
5647                                                                  ;THIS CODE FINDS OUT IF THE CPU
5648                                                                  ;IS AN 11/05 OR ELSE.
5649                                                                  ;ON AN 11/05, R0 (177700) CAN BE
5650                                                                  ;ADDRESSED AS A MEMORY LOCATION, BUT
5651                                                                  ;ON ANY OTHER CPU IF 177700 IS REFERENCED
5652                                                                  ;A TIME OUT WILL OCCUR.
5653  017334  012737  017356  000004          MOV     #5$,@#4         ;SET UP TIME OUT VECTOR
5654  017342  005737  177700                  TST     @#177700        ;REFERENCE R0
5655  017346  012737  004534  000004          MOV     #BADTMO,@#4     ;R0 WAS REFERENCED W/O TIMEOUT
5656                                                                  ;HENCE 11/05
5657  017354  000520                          BR      TST54           ;;SKIP THIS TEST
5658  017356  022626                  5$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
5659  017360  012737  004534  000004          MOV     #BADTMO,@#4     ;RESTORE TIMEOUT VECTOR
5660  017366  012746  000340                  MOV     #340,-(SP)
5661  017372  012746  017400                  MOV     #64$,-(SP)
5662  017376  000002                          RTI
5663  017400                          64$:
5664  017400  013700  001332                  MOV     RKCS,R0
5665  017404  104413                          CNT.RESET               ;GO, DO CONTROL RESET
                                                                      ;THIS IS A CALL FOR THE 'CNTRL-
                                                                      ;RESET' ROUTINE. A CONTROL RESET IS
                                                                      ;ISSUED AND AFTER A CERTAIN TIME
                                                                      ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                      ;AN ERROR IS REPORTED. NOTE THAT
                                                                      ;THE PC IN ERROR MESSAGE IS THE
                                                                      ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                      ;THIS IS A VERY BASIC ERR& IF IT
                                                                      ;OCCURS GO BACK TO TEST 10
5675  017406  012701  033336                  MOV     #OUTBUF,R1
5676  017412  013704  001336                  MOV     RKBA,R4
5677  017416  012711  000100                  MOV     #100,@R1        ;WRITE THIS WORD
```

```
5678  017422  012777  177777  161704          MOV     #-1,@RKWC       ;WRITE 1 WORD
5679  017430  013702  001350                  MOV     DRIVAD,R2
5680  017434  052702  000002                  BIS     #2,R2           ;ON CYL 0, SEC 2
5681  017440  010277  161674                  MOV     R2,@RKDA
5682  017444  010114                          MOV     R1,@R4          ;FROM THIS MEMORY LOC
5683  017446  012710  000003                  MOV     #3,@R0          ;WRITE, GO
5684  017452  005003                          CLR     R3
5685  017454  105710                  1$:     TSTB    @R0
5686  017456  100410                          BMI     2$
5687  017460  005203                          INC     R3
5688  017462  001374                          BNE     1$
5689  017464  004737  020770                  JSR     PC,GT4RG        ;GET RKCS, ER, DS
5690  017470  010237  001202                  MOV     R2,$REG10       ;GET THE STARTING ADRES
5691  017474  104416                          BRKDA4                  ;BREAK IT INTO DRV #, CYL, SUR, SEC #
5692  017476  104031                          ERROR   31              ;CNTRL RDY DID NOT SET AFTER
5693                                                                  ;WRITE OF 1 WORD ON CYL 0, SEC 2
5694  017500  012777  177777  161626  2$:     MOV     #-1,@RKWC       ;READ 1 WORD
5695  017506  010277  161626                  MOV     R2,@RKDA        ;FROM SEC 2, CYL 0
5696  017512  013714  001144                  MOV     $TKS,@R4        ;INTO TTY STAUS REGISTER
5697  017516  005077  161422                  CLR     @$TKS           ;CLEAR TTY KEY BRD STATUS REG
5698
5699  017522  012710  000065                  MOV     #65,@R0         ;READ, MEX BITS SET
5700  017526  005003                          CLR     R3
5701  017530  105710                  3$:     TSTB    @R0
5702  017532  100410                          BMI     4$
5703  017534  005203                          INC     R3
5704  017536  001374                          BNE     3$
5705  017540  004737  020770                  JSR     PC,GT4RG
5706  017544  010237  001202                  MOV     R2,$REG10       ;GET THE STARTING ADRES
5707  017550  104416                          BRKDA4                  ;BREAK IT INTO DR#, CYL, SUR, SEC#
5708  017552  104045                          ERROR   45              ;CNTRL RDY DIDN'T SET AFTER
5709                                                                  ;READ OF 1 WORD FROM CYL 0, SEC 2.
5710                                                                  ;IN EROR MSGE, <DSK-ADRES> GIVES
5711                                                                  ;ADRES WHERE READ BEGAN. 'RKDA'
5712                                                                  ;GIVES CONTENTS OF RKDA AT TIME OF ERCR
5713  017554  032737  000100  001144  4$:     BIT     #100,$TKS       ;WAS THE CORRECT WORD READ INTO
5714                                                                  ;THE TTY  STATUS REGISTER?
5715  017562  001015                          BNE     TST54           ;;YES, EXIT
5716  017564  017705  161354                  MOV     @$TKS,R5                ;GET THE WORD RECVD FROM DISK
5717  017570  010537  001164                  MOV     R5,$REG1
5718  017574  052705  000100                  BIS     #100,R5         ;THIS WORD WAS EXPCTD
5719  017600  010537  001162                  MOV     R5,$REG0        ;STORE EXPCTD WORD
5720  017604  011437  001166                  MOV     @R4,$REG2       ;GET RKBA
5721  017610  011037  001170                  MOV     @R0,$REG3       ;GET RKCS
5722  017614  104115                          ERROR   115             ;DATA ERROR. A ONE WORD (100)
5723                                                                  ;NPR WAS TRIED FROM DISK TO
5724                                                                  ;TTY KEYBOARD STATUS REGISTER
5725                                                                  ;(17756) . BIT 6 SHOULD HAVE BEEN
5726                                                                  ; SET AS RESULT OF THIS
5727                                                                  ;BUT IT WAS NOT
5728
5729
5730                                  ;;*****************************************************************
5731                                  ;*TEST 54       CHECK THAT RKBA CAN COUNT CORRECTLY
5732                                          ;*THIS TEST CHECKS THAT RKBA CAN COUNT CORRECTLY. IT IS SET
5733                                          ;*TO THE DESIRED INITIAL VALUE.  THEN A ONE WORD WRITE CHECK
```

```
                                                ;*IS TRIED, WITH MEX (MEMORY EXTENSION) BITS SET. IF THEPE IS
                                                ;*NO MEMORY PRESENT (FOR CERTAIN BUS ADDRESSES),  THERE
                                                ;*WILL BE AN NXM ERROR STOPPING CONTROLLER ACTION. BUT RKBA
                                                ;*SHOULD HAVE INCREMENTED BY 1 FROM ITS INITIAL VALUE. IF IT
                                                ;*HAS NOT, AN ERROR IS REPORTED.
                                                ;;*****************************************************************
017616  000004                          T$T54:  SCOPE
017620  012737  000005  001206                  MOV     #5,$TIMES       ;;DO 5 ITERATIONS
017626  104421                                  TST.SIN                 ;CHECK IF SIN SET, IF SET DRV RESET
017630  005001                                  CLR     R1              ;INITIALIZE (VALUE OF RKBA)
017632  012702  000002                          MOV     #2,R2           ;INITIALIZE (INCMNTD VALUE OF RKBA.

017636  012737  017650  001110                  MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
                                                                        ;ON EROR

017644  013705  001336                  1$:     MOV     RKBA,R5
017650  004737  021432                          JSR     PC,TSTRWS       ;WAIT FOR R/W/S RDY
017654  104016                                  ERROR   16              ;R/W/S RDY IS NOT SET
017656  104413                                  CNT.RESET               ;DO CONTROL RESET
017660  012777  177777  161446                  MOV     #-1,@RKWC       ;WRITE CHK 1 WORD
017666  010115                                  MOV     R1,@R5  ;THIS BUS ADRES
017670  013777  001350  161442                  MOV     DRIVAD,@RKDA    ;SET DISK ADRES
017676  012777  000067  161426                  MOV     #67,@RKCS       ;WRITE CHECK, GO, MEX BITS SET
017704  104412                                  CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
                                                                        ;IF SO, SKIP THE EROR MESSAGE.
017706  104065                                  ERROR   65              ;CNTRL RDY DID NOT SET AFTER
                                                                        ;WRT CHK WAS TRIED TO NXM LOC
                                                                        ;U MIGHT WANT TO USE TESTS
                                                                        ;CHECKING MEX BITS & NXM.
017710  005237  001356                          INC     INDX1           ;ALLOW ONLY 5 ERRORS OF ABOVE KIND
017714  001417                                  BEQ     5$

017716  020215                  3$:     CMP     R2,@R5                  ;DID RKBA INCREMENT BY 1 FROM
                                                                        ;ITS INITIAL VALUE?
017720  001410                                  BEQ     4$              ;YES, BRANCH
017722  010137  001162                          MOV     R1,$REG0        ;GET EXPCTD RKBA
017726  011537  001164                          MOV     @R5,$REG1       ;GET RKBA RECVD
017732  104017                                  ERROR   17              ;RKBA DID NOT INCREMENT BY
                                                                        ;1 FROM ITS INITIAL VALUE.
                                                                        ;ONE WORD WRT CHK WAS TRIED
                                                                        ;TO A NXM LOCATION. THERE
                                                                        ;WILL BE AN NXM ERROR,
                                                                        ;BUT STILL RKBA SHOULD
                                                                        ;INCREMENT BY 1 FROM ITS
                                                                        ;INITIAL VALUE.
017734  005237  001360                          INC     INDX2           ;ALLOW ONLY 5 ERRORS OF
017740  001405                                  BEQ     5$              ;THE ABOVE KIND
017742  060201                  4$:     ADD     R2,R1                   ;SET NXT VALUE OF RKBA
017744  010102                          MOV     R1,R2
017746  062702  000002                  ADD     #2,R2           ;SET EXPCTD VALUE OF RkBA
017752  001336                          BNE     1$              ;ALL DONE?

017754                          5$:                             ;DUMMY EXIT POINT


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

```
                                        ;*TEST 55       CHECK FOR RK-05F
                                                ;*THIS TEST CHECKS RK-05F TYPE DRIVES
                                                ;*TO INSURE THAT IF SEEKS ARE ISSUED ON ONE
                                                ;*DRIVE, THE OTHER DRIVE BECOMES BUSY
                                        ;;*****************************************************************
017754  000004                        $T55:   SCOPE
017756  012737  000001  001206                MOV     #1,$TIMES       ;;DO 1 ITERATION
017764  005737  001404                        TST     FFLAG           ;SEE IF RK-05F
017770  001403                        BEQ     1$              ;NOT F
017772  004537  025154                        JSR     R5,FCHECK       ;SEE IF OTHER GOES BUSY
017776  104120                        ERROR   120
020000                                  1$:
                                        ;;*****************************************************************
                                        ;*TEST 56       END OF PROGRAM
                                                ;*THIS IS NOT A TEST, BUT A LINKAGE PROVIDED TO PERFORM
                                                ;*THE ABOVE SUB-TESTS FOR ALL DRIVES THAT ARE PRESENT.
                                                ;*NOTE THAT THE NEXT TEST- HARDWARE POLLING LOGIC-
                                                ;*IS DONE USING ALL THE DRIVES THAT ARE INDICATED PRESENT.
                                                ;*DO NOT LOOP ON THIS 'TEST'.
                                        ;;*****************************************************************
020000  000004                        $T56:   SCOPE
020002  012737  000001  001206                MOV     #1,$TIMES       ;;DO 1 ITERATION
020010  005237  001352                        INC     DRVDON          ;INCREMENT THE COUNT FOR THE NUMBER
                                                                        ;OF DRIVES THAT ARE CHECKED
020014  004737  021500                        JSR     PC,DRESET       ;RESET THE DRIVE
020020  104026                        ERROR   26              ;R/W/S DIDN'T SET AFTER DRIVE RESET
020022  023737  001412  001352  BTEOP:  CMP     DRIVS,DRVDON    ;HAVE U TESTED ALL THE DRIVES
                                                                        ;THAT ARE PRESENT?
020030  001405                        BEQ     1$              ;IF YES, EXIT
020032  062737  020000  001350                ADD     #20000,DRIVAD   ;ADRES THE NXT POSSIBLE DRIVE
020040  000137  005040                        JMP     NUDRV           ;GO BACK AND TEST THE NEXT
                                                                        ;DRIVE PRESENT
020044  005037  001112                  1$:   CLR     $ERTTL
                                        ;;*****************************************************************
                                        ;*TEST 57       CHECK HARDWARE POLLING LOGIC
                                                ;*THIS TEST CHECKS THE HARDWARE POLL LOGIC, USING ALL THE DRIVES
                                                ;*PRESENT ON THE RK11.  ATLEAST TWO DRIVES SHOULD BE PRESENT
                                                ;*TO DO A MEANINGFUL HARDWARE POLL.  SEQUENCE OF OPERATIONS IS
                                                ;*AS FOLLOWING:
                                                ;*1) NUMBER OF DRIVES ON THE RK11 IS ASCERTAINED.
                                                ;*2) HAVING LOCKED OUT ALL INTERRUPTS (CPU PR 7), SEEK IS INITIATED
                                                ;*FOR ONE DRIVE AT A TIME, ONLY WHEN 'CNTRL RDY' IS SET.
                                                ;*3) CPU PRIORITY IS DROPPED TO 4 SO THAT RK11 CAN INTERRUPT, THE INCOMING
                                                ;*INTERRUPT IS PROCESSED TO CHECK IF IT WAS DUE TO 'SEEK DONE' BY
                                                ;*ONE OF THE DRIVES.
                                                ;*4) IF BY THE END OF THE SET TIME A DRIVE HAS NOT INTERRUPTED
                                                ;*AN ERROR MESSAGE IS GIVEN INDICATING WHICH DRIVE DID NOT
                                                ;*INTERRUPT AFTER SEEK WAS DONE.
                                        ;;*****************************************************************
020050  000004                        $T57:   SCOPE
020052  012737  000005  001206                MOV     #5,$TIMES       ;;DO 5 ITERATIONS
020060  005237  001440                        INC     SIZYET          ;FOUND RK05F YET?
```

```
5846  020064  001002                      BNE     25$             ;YES
5847  020066  004737  025300              JSR     PC,SIZEF        ;FIND WHICH ARE RK-05F
5848  020072  005037  001436      25$:    CLR     PHYDRV          ;NUMBER OF ACTUAL DRIVES
5849  020076  012700  001414              MOV     #DRIV0,R0       ;TABLE
5850  020102  005710              23$:    TST     (R0)            ;DRIVE HERE+?
5851  020104  001405                      BEQ     22$             ;NO
5852  020106  005237  001436              INC     PHYDRV          ;COUNT DRIVE
5853  020112  005710                      TST     (R0)            ;RK05F?
5854  020114  100001                      BPL     22$             ;NO
5855  020116  005720                      TST     (R0)+           ;DONT COUNT F TWICE
5856  020120  005720              22$:    TST     (R0)+   ;NEXT DRIVE
5857  020122  020027  001433              CMP     R0,#DRIV7+1     ;ALL YET
5858  020126  002765                      BLT     23$             ;NO
5859  020130  005037  001406              CLR     ODDEVN          ;EVEN DRIVES FIRST IF F
5860  020134  005737  001412      T56:    TST     DRIVS           ;ANY DRIVES PRESENT?
5661  020140  001002                      BNE     20$             ;YES
5862  020142  000137  020646              JMP     $EOP            ;NO
5863  020146  005237  001434      20$:    INC     T56FLG
5864  020152  013700  001332              MOV     RKCS,R0
5865  020156  005037  001356              CLR     INDX1           ;FLAG TO INDICATE:
5866                                                              ;(INDX1)=0 POLLING DONE AFTER ALL
5867                                                                        ;DRIVES SEEK TO CYL 0
5868                                                              ;(INDX1)=1 POLLING DONE AFTER ALL
5869                                                                        ;DRIVES SEEK TO CYL 4
5870  020162  005037  001360      15$:    CLR     INDX2           ;FLAG INDICATING TYPE OF INTERRUPT
5871                                                              ;SET TO NON-ZERO TO INDICATE
5872                                                              ;THAT THE INTERRUPT IS DUE TO
5873                                                              ;SEEK DONE
5874  020166  104413                      CNT.RESET               ;GO, DO CONTROL RESET
5875                                                              ;THIS IS A CALL FOR THE 'CNTRL-
5876                                                              ;RESET' ROUTINE. A CONTROL RESET IS
5877                                                              ;ISSUED AND AFTER A CERTAIN TIME
5878                                                              ;IF THE 'CNTRL RDY' DOES NOT SET
5879                                                              ;AN ERROR IS REPORTED. NOTE THAT
5880                                                              ;THE PC IN ERROR MESSAGE IS THE
5881                                                              ;PC WHERE 'CNT.RESET' IS LOCATED.
5882                                                              ;THIS IS A VERY BASIC ERR3 IF IT
5883                                                              ;OCCURS GO BACK TO TEST 10
5884  020170  005737  001356              TST     INDX1           ;PERFORMING SEEKS TO CYL 4
5885  020174  001002                      BNE     .+6             ;YES, BRANCH
5886  020176  005002                      CLR     R2              ;NO
5887  020200  000402                      BR      .+6
5888  020202  012702  000200              MOV     #200,R2         ;SET ADRES FOR FOURTH CYLINDER
5889  020206  012701  001414              MOV     #DRIV0,R1       ;INITIALIZE POINTER
5890  020212  012703  177770              MOV     #-10,R3         ;SET COUNT FOR 8 DRIVES
5891  020216  012705  033336              MOV     #OUTBUF,R5      ;INITIALIZE POINTER TO INDICATOR AREA
5892  020222  005025                      CLR     (R5)+           ;CLEAR OUT THE 8-WORD INDICATOR
5893  020224  005203                      INC     R3              ;AREA WHICH IS USED FOR DOING
5894  020226  001375                      BNE     .-4             ;SOFTWARE POLLING LATER ON
5895  020230  012703  177770              MOV     #-10,R3         ;SET COUNT FOR 8 POSSIBLE DRIVES
5896  020234  012705  033336              MOV     #OUTBUF,R5      ;INITIALIZE POINTER TO INDICATOR AREA
5897  020240                      1$:
5898  020240  012746  000340              MOV     #340,-(SP)
5899  020244  012746  020252              MOV     #64$,-(SP)
5900  020250  000002                      RTI
5901  020252                      64$:
```

```
5902  020252  032711  000001          BIT     #BIT0,(R1)      ;IS THIS DRIVE PRESENT?
5903  020256  001433                  BEQ     4$              ;IF NOT, BRANCH
5904  020260  005711                  TST     (R1)            ;RK06F?
5905  020262  100012                  BPL     17$             ;NO, CONTINUE
5906  020264  032702  020000          BIT     #BIT13,R2       ;DRIVE EVEN?
5907  020270  001404                  BEQ     16$             ;YES
5908  020272  005737  001406          TST     ODDEVN          ;DO WE WANT ODD?
5909  020276  001423                  BEQ     4$              ;NO, SO DO NOT TEST
5910  020300  000403                  BR      17$             ;ADD THIS DRIVE TO LIST
5911  020302  005737  001406  16$:    TST     ODDEVN          ;DO WE WANT EVEN?
5912  020306  001017                  BNE     4$              ;NO, SO SKIP
5913  020310  010215          17$:    MOV     R2,(R5)         ;SET UP THIS WORD IN THE
5914                                                          ;INDICATOR AREA SHOWING THAT THIS
5915                                                          ;DRIVE (AS IN BITS 13-15 OF R2)
5916                                                          ;IS PRESENT
5917  020312  042725  017777          BIC     #17777,(R5)+    ;MASK OUT UNWANTED BITS (CYL,SUR,SEC BITS
5918  020316  005004                  CLR     R4
5919  020320  105710          2$:     TSTB    @R0             ;IS CNTRL RDY SET?
5920  020322  100405                  BMI     3$              ;YES, BRANCH
5921  020324  005204                  INC     R4              ;NO, WAIT FOR IT
5922  020326  001374                  BNE     2$              ;IF WAITED LONG REPORT ERROR
5923  020330  004737  020770          JSR     PC,GTHRG                ;GO, GET RKCS,ER,DS,DA
5924  020334  104021                  ERROR   21              ;CNTRL RDY DID NOT SET AFTER ACCEPTING
5925                                                          ;ADRES FROM PREVIOUS SEEK
5926  020336  010277  160776  3$:     MOV     R2,@RKDA        ;ADRES THIS DRIVE, CYL 0 OR CYL 4
5927                                                          ;(WHICHEVER THE CASE MAY BE)
5928  020342  012710  000111          MOV     #111,@R0        ;SEEK,GO,IDE SET
5929  020346  005721          4$:     TST     (R1)+           ;NEXT DRIVE DATA
5930  020350  062702  020000          ADD     #20000,R2       ;INCREMENT DRIVE ADRES (BITS 15,14,13
5931  020354  005203                  INC     R3              ;TO NEXT ONE
5932  020356  001330                  BNE     1$              ;BRANCH BACK IF ALL DRIVES ARE
5933                                                          ;NOT CHECKED TO SEE IF THE NEXT
5934                                                          ;DRIVE IS PRESENT (& IF SO ISSUE A
5935                                                          ;SEEK TO IT)
5936                                                          ;BY NOW SEEKS HAVE BEEN ISSUED
5937                                                          ;TO ALL DRIVES PRESENT & POLLING
5938                                                          ;HAS BEGUN
5939  020360  005004                  CLR     R4
5940  020362  013702  001402  5$:     MOV     RKVEC,R2
5941  020366  012722  020420          MOV     #6$,(R2)+       ;SET ADRES FOR RK11 TO INTERUPT
5942  020372  012712  000340          MOV     #340,(R2)       ;SET PSW ON INTERUPT
5943  020376  013746  001400          MOV     RKPRI,-(SP)     ;DROP CPU PRIORITY TO 4 SO THAT
5944  020402  012746  020410          MOV     #18$,-(SP)              ;RK11 CAN INTERUPT
5945  020406  000002                  RTI
5946  020410  000240          18$:    NOP                     ;THIS IS A TIME LOOP DURING
5947  020412  005204                  INC     R4              ;WHICH ALL DRIVES PRESENT SHOULD
5948  020414  001375                  BNE     18$             ;INTERRUPT
5949  020416  000452                  BR      11$             ;BRANCH AND CHECK IF ALL AVAILABLE
5950                                                          ;DRIVES INTERRUPTED CORRECTLY
5951  020420  022626          6$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
5952  020422  005737  001360          TST     INDX2           ;WAS THIS FIRST INTERRUPT
5953                                                          ;DUE TO 'ADRES ACK' AFTER INITIATION
5954                                                          ;OF SEEK?
5955  020426  001021                  BNE     9$              ;IF YES, CHECK THE FOLLOWING
5956
5957  020430  032710  020000          BIT     #20000,@R0      ;CHECK THAT SCP IS NOT SET
```

```
5958  020434  001403                    BEQ     7$              ;BRANCH IF SCP CLEAR
5959  020436  011037  001162            MOV     @R0,$REG0       ;GET RKCS
5960  020442  104076                    ERROR   76              ;AFTER THE FIRST INTERRUPT WHICH
5961                                                            ;IS DUE TO INITIATION OF SEEK, SCP
5962                                                            ;SHOULD NOT HAVE SET. IT DID
5963  020444  017701  160656    7$:     MOV     @RKDS,R1
5964  020450  032701  160000            BIT     #160000,R1      ;RKDS BITS 15-13 SHLOULD BE CLR
5965  020454  001403                    BEQ     8$
5966  020456  010137  001162            MOV     R1,$REG0        ;GET RKDS
5967  020462  104050                    ERROR   50              ;SEEK, WITH IDE SET WAS ISSUED TO
5968                                                            ;ALL AVAILABLE DRIVES. THE FIRST
5969                                                            ;INTERUPT IS DUE TO SEEK INITIATED
5970                                                            ;BY FRST DRV. DRV ID BITS 13-15
5971                                                            ;SHOULD BE CLR AFTR THIS FRST INRUPT.
5972                                                            ;THEY WERE NOT IF THIS ERROR OCCURS.
5973  020464  005237  001360    8$:     INC     INDX2           ;SET UP FLAG INDICATING
5974                                                            ;THAT THE FIRST INTERRUPT DUE
5975                                                            ;TO INITIATION OF SEEK WAS
5976                                                            ;PROCESSED
5977  020470  000734                    BR      5$              ;GO BACK TO THE WAIT LOOP & WAIT
5978                                                            ;FOR NEXT INTERRUPT FROM RK11
5979  020472  013703  001436    9$:     MOV     PHYDRV,R3       ;SET COUNT OF # OF DRIVES PRESENT
5980  020476  012705  033336            MOV     #OUTBUF,R5      ;INITIALIZE POINTER
5981  020502  017701  160620            MOV     @RKDS,R1        ;GET RKDS
5982  020506  042701  017777            BIC     #17777,R1       ;MASK BITS 0-12
5983                                                            ;THE FOLLOWING CODE IS A SOFTWARE
5984                                                            ;POLL WHICH FINDS OUT WHICH DRIVE
5985                                                            ;CAUSED THE PRESENT INTERRUPT
5986                                                            ;AND SETS UP A FLAG BIT FOR
5987                                                            ;THE DRIVE #, INDICATING THAT
5988                                                            ;THIS DRIVE # INTERRUPTED
5989  020512  020125                    CMP     R1,(R5)+
5990  020514  001411                    BEQ     10$             ;BRANCH IF INTERRUPTING DRIVE WAS FOUND
5991  020516  005303                    DEC     R3              ;HAVE U CHKD ALL DRIVS PRESENT?
5992  020520  001374                    BNE     .-6             ;IF NOT LUP BAK & CHK
5993                                                            ;REPORT ERROR IF THE INTERRUPTING
5994                                                            ;DRIVE # (AS IN RKDS 13-15) WAS NOT
5995                                                            ;ANY ONE OF THOSE THAT ARE PRESENT
5996  020522  010146                    MOV     R1,-(R6)        ;GET WORD TO B SHFTD RT
5997  020524  004737  021174            JSR     PC,SHFTRT       ;GO SHIFT IT
5998  020530  012637  001162            MOV     (R6)+,$REG0     ;THIS DRIVE # WAS RECVD IN RKDS AS
5999                                                            ;THE INTERRUPTING DRIVE, BUT THIS
6000                                                            ;DRIVE IS NOT PHYSICALY PRESENT
6001  020534  104051                    ERROR   51              ;RKDS INDICATES AN INTERRUPTING
6002                                                            ;DRIVE # (DURING H'WARE POLL) BUT
6003                                                            ;THAT DRIVE IS ACTUALLY NOT PRESENT
6004  020536  000401                    BR      10$+2
6005  020540  005245            10$:    INC     -(R5)           ;SET UP FLAG INDICATING THAT
6006                                                            ;THE INTERRUPT FOR THIS DRIVE
6007                                                            ;(AFTER IT HAD COMPLETED ITS SEEK)
6008                                                            ;WAS PROCESSED
6009  020542  000707                    BR      5$              ;GO BAK & WAIT FOR FURTHER INTRUPTS
6010  020544  013703  001436    11$:    MOV     PHYDRV,R3       ;GET # OF DRIVES
6011  020550  012705  033336            MOV     #OUTBUF,R5      ;INITIALIZE POINTER
6012
6013  020554  105715            14$:    TSTB    (R5)            ;DID THIS DRIVE INTERRUPT?
```

```
                                        BNE     13$             ;YES, BRANCH
                                        MOV     (R5),-(R6)      ;GET THIS DRIVE #
                                        JSR     PC,SHFTRT       ;SHIFT IT TO THE RIGHT
                                        MOV     (R6)+,$REGO     ;THIS DRIVE # DID NOT INTERRUPT
                                                                ;DURING H'WARE POLL
                                        ERROR   52              ;DRIVE # (AS IN $REGO) DID NOT
                                                                ;INTERRUPT DURING HARDWARE POLL
                                13$:    ADD     #2,R5           ;INCREMENT POINTER TO THE NEXT FLAG
                                        DEC     R3              ;CHKD FOR ALL DRIVES?
                                        BNE     14$             ;IF NOT LUP BACK

                                        TST     INDX1           ;DONE POLLING FOR SEEKS TO CYL 312?
                                        BNE     TSTEND          ;IF YES, EXIT
                                        INC     INDX1           ;IF NOT, INCREMENT FLAG
                                        JMP     15$             ;GO DO IT

                                                                ;INDICATOR TABLE
                                                                ;THE 8-WORD INDICATOR TABLE USED IN
                                                                ;THE FORMER PART OF THIS SUB-TEST
                                                                ;IS LOCATED STARTING AT 'OUTBUF'.
                                                                ;WORDS ARE SET UP TO INDICATE
                                                                ;PRESENCE OF A DRIVE EG: IF
                                                                ;DRIVES 0,1,2 ARE PRESENT, IT WILL
                                                                ;LOOK LIKE
                                                                ;OUTBUF:        000000 BITS 13,14,15
                                                                ;               020000 CONTAIN THE
                                                                ;               040000 DRIVE NO.
                                                                ;               000000 REST 0'S
                                                                ;WHEN A DRIVE INTERRUPTS AFTER SEEK
                                                                ;IS DONE BIT 0 OF THE CORRESPONDING
                                                                ;INDICATOR WORD IS SET. THUS FOR THE
                                                                ;ABOVE EXAMPLE IF ALL DRIVES INTERRUPTED
                                                                ;CORRECTLY THEN IT WILL LOOK LIKE:
                                                                ;       12$:    000001 BIT 0 SET
                                                                ;               020001 TO INDICATE
                                                                ;               040001 DR INTERRUPTED
                                                                ;               000000 REST 0'S

                                TSTEND: INC     ODDEVN          ;NOW ODD IF RK05F
                                        CMP     #2,ODDEVN       ;SEE IF DONE
                                        BEQ     21$             ;ALL DONE
                                        JMP     T56             ;TEST AGAIN
                                21$:    CLR     T56FLG

                                .SBTTL  END OF PASS ROUTINE

                                ;;*****************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
                                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO ST4

                                $EOP:
```

```
 6070  020646  000004                          SCOPE
 6071  020650  005037  001102                  CLR     $ISINM          ;;ZERO THE TEST NUMBER
 6072  020654  005037  001206                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
 6073  020660  005237  001100                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
 6074  020664  042737  100000  001100          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 6075  020672  005327                          DEC     (PC)+           ;;LOOP?
 6076  020674  000001                $EOPCT:   .WORD   1
 6077  020676  003022                          BGT     $DOAGN          ;;YES
 6078  020700  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 6079  020702  000001                $ENDCT:   .WORD   1
 6080  020704  020674                          $EOPCT
 6081  020706  104401  020753                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
 6082  020712  013746  001100                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
 6083  020716  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 6084  020720  104401  020750                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
 6085  020724  013700  000042        $GET42:   MOV     @#42,R0         ;;GET MONITOR ADDRESS
 6086  020730  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 6087  020732  000005                          RESET                   ;;CLEAR THE WORLD
 6088  020734  004710                $ENDAD:   JSR     PC,(R0)         ;;GO TO MONITOR
 6089  020736  000240                          NOP                     ;;SAVE ROOM
 6090  020740  000240                          NOP                     ;;FOR
 6091  020742  000240                          NOP                     ;;ACT11
 6092  020744                        $DOAGN:
 6093  020744  000137                          JMP     @(PC)+          ;;RETURN
 6094  020746  004456                $RTNAD:   .WORD   ST4
 6095  020750     377     377      000 $ENULL:  .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 6096  020753     015  042412  042116 $ENDMG:  .ASCIZ  <15><12>/END PASS #/
 6097  020760  050040  051501  020123
 6098  020766  000043
 6099
 6100
 6101
 6102
 6103
 6104
 6105
 6106                          .SBTTL  GT2RG:  ROUTINE FOR GETTING RKCS,RKER
 6107
 6108                  ;SUBROUTINE FOR TRANSFERRING THE CONTENTS OF RKCS, RKER
 6109                  ;TO $REG0, $REG1 RESPECTIVELY BEFORE TYPING OUT AN ERROR MESSAGE.
 6110                  ;CALL:  JSR     PC,GT2RG
 6111
 6112                          .SBTTL  GT3RG:  ROUTINE FOR GETTING RKCS, RKER, RKDS
 6113
 6114                  ;GT3RG
 6115                  ;SUBROUTINE FOR TRANSFERRING THE CONTENTS OF RKCS, RKER, RKDS
 6116                  ;TO  $REG0, $REG1, $REG2 RESPECTIVELY BEFORE TYPING OUT AN
 6117                  ;ERROR MESSAGE.
 6118                  ;CALL:  JSR     PC,GT3RG
 6119
 6120                          .SBTTL  GT4RG:  ROUTINE FOR GETTING RKCS, RKER, RKDS, RKDA
 6121
 6122                  ;GT4RG
 6123                  ;SUBROUTINE FOR TRANSFERRING CONTENTS OF RKCS, RKER, RKDS
 6124                  ;RKDA TO $REG0, $REG1, $REG2, $REG3 RESPECTIVELY BEFORE
 6125                  ;TYPING OUT AN ERROR MESSAGE.
```

```
                                        ;CALL:  JSR     PC,GT4RG

                                        GT4RG:  MOV     @RKDA,$REG3             ;GET RKDA
                                        GT3RG:  MOV     @RKDS,$REG2             ;GET RKDS
                                        GT2RG:  MOV     @RKER,$REG1             ;GET RKER
                                                MOV     @RKCS,$REG0
                                                RTS     PC


                                                .SBTTL  TYERM:  SPECIAL ERROR MESSAGE ROUTINE

                                        ;TYERM
                                        ;THIS ROUTINE TYPES OUT 'EROR AT PC=X'
                                        ;X IS THE PC WHERE THE  EXPLANATION AS TO WHAT HAPPENED IS GIVEN. THIS ROUTINE
                                        ;IS USED ONLY FOR NON-MANUAL MODE OF THE PROGRAM.
                                        ;CALL:          JSR     TYERM

                                        TYERM:
                                                TYPE    ,65$                    ;;TYPE ASCIZ STRING
                                                BR      64$                     ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <15><12>/EROR.PC=/
                                        64$:
                                                MOV     R3,-(SP)
                                                TYPOC
                                                RTS     PC


                                                .SBTTL  BDA0, BDA4:     BREAK DISK ADDRESS INTO SEC, SUR, CYL, DRIVE

                                        ;BDA0, BDA4

                                        ;THIS ROUTINE BREAKS A DISK ADDRESS (BITS 0-15) INTO  DRIVE #,
                                        ;CYLINDER #, SURFACE, SECTOR #. THE ROUTINE IS CALLED BY USING EITHER
                                        ;BRKDA0 OR BRKDA4, BOTH BEING 'TRAP' INSTRUCTIOS WITH THEIR LOWER BYTES
                                        ;ENCODED TO PROVIDE INDEXING TO 'BDA0' OR 'BDA4'. BEFORE CALLING
                                        ;THE ROUTINE THE DISK ADDRESS WHICH IS TO BE BROKEN AS ABOVE
                                        ;IS DEPOSITED IN $REG10.
                                        ;'BRKDA0' PUTS THE                      BRKDA4 PUTS THE
                                        ;DRIVE # INTO $REG0                     DRIVE # INTO $REG4
                                        ;CYLINDER # INTO $REG1                  CYLINDER # INTO $REG5
                                        ;SURFACE # INTO $REG2                   SURFACE # INTO $REG6
                                        ;SECTOR # INTO $REG3                    SECTOR # INTO $REG7
                                        ;CALL:  BRKDA0                          BRKDA4


                                        BDA0:   MOV     R0,-(SP)        ;PUSH R0 ONTO THE STACK
                                                MOV     #$REG3+2,R0     ;SET JP POINTER
                                                BR      BDAR
```

```
6182  021062  010046                  BDA4:   MOV     R0,-(SP)        ;PUSH R0 ONTO THE STACK
6183  021064  012700  001202                  MOV     #$REG7+2,R0     ;SET UP POINTER
6184
6185  021070  032777  020000  160042  BDAR:   BIT     #20000,@SWR     ;INHIBIT TYPEOUT?
6186  021076  001034                          BNE     2$              ;YES, BRANCH TO EXIT POINT
6187
6188  021100  010146                          MOV     R1,-(SP)        ;PUSH R1 ON STACK
6189  021102  010246                          MOV     R2,-(SP)        ;PUSH R2 ON STACK
6190  021104  013701  001202                  MOV     $REG10,R1       ;GET THE ADDRESS WHICH
6191                                                                  ;HAS TO BE BROKEN
6192  021110  042701  177760                  BIC     #177760,R1      ;EXTRACT SECTOR BITS 0-3
6193  021114  010140                          MOV     R1,-(R0)        ;MOVE SECTOR BITS TO $REG3 OR $REG7
6194  021116  013701  001202                  MOV     $REG10,R1       ;GET THE DSK-ADRES TO BE BROKEN
6195  021122  006201                          ASR     R1              ;SHIFT RIGHT 4 TIMES
6196  021124  006201                          ASR     R1
6197  021126  006201                          ASR     R1
6198  021130  006201                          ASR     R1
6199  021132  010102                          MOV     R1,R2           ;STORE THIS
6200  021134  042702  177776                  BIC     #177776,R2      ;EXTRACCT THE SURFACE BIT
6201  021140  010240                          MOV     R2,-(R0)        ;MOVE SURFACE BIT TO $REG3 OR $REG6
6202  021142  006201                          ASR     R1
6203  021144  010102                          MOV     R1,R2           ;STORE IT
6204  021146  042702  177400                  BIC     #177400,R2      ;EXTRACT THE CYLINDER BITS
6205  021152  010240                          MOV     R2,-(R0)        ;MOVE CYLINDER BITS TO $REG1 OR $REG5
6206  021154  000301                          SWAB    R1              ;SWAB HI-LO BYTES
6207  021156  042701  177770                  BIC     #177770,R1      ;EXTRACT THE DRIVE #
6208  021162  010140                          MOV     R1,-(R0)        ;MOVE DRIVE # TO $REG0 OR $REG4
6209
6210  021164  012602                  2$:     MOV     (SP)+,R2        ;RESTORE R2
6211  021166  012601                          MOV     (SP)+,R1        ;RESTORE R1
6212  021170  012600                          MOV     (SP)+,R0        ;RESTORE R0 FROM THE STACK
6213  021172  000002                          RTI                     ;RETURN FROM INTERRUPT, EXIT THIS
6214                                                                  ;ROUTINE
6215
6216
6217
6218                                          .SBTTL  SHFTRT: SHIFT RIGHT ROUTINE
6219
6220                                  ;SHFTRT
6221                                  ;THIS ROUTINE SHIFTS A WORD TO THE RIGHT 13 TIMES. THE WORD TO BE SHIFTED
6222                                  ;IS PUT ON THE STACK BEFORE ENTERING THIS ROUTINE AND IT IS POPPED UP
6223                                  ;FROM THE STACK AFTER THE SHIFT HAS BEEN DONE.
6224                                  ;CALL:  JSR     PC,SHFTRT
6225
6226  021174  012737  177763  021220  SHFTRT: MOV     #-15.,2$        ;SET UP A COUNT OF 13
6227  021202  000241                          CLC                     ;CLEAR THA C BIT
6228  021204  006066  000002          1$:     ROR     2(R6)           ;ROTATE RIGHT THE WORD TO B SHFTD
6229  021210  005237  021220                  INC     2$              ;SHIFTED 13 TIMES?
6230  021214  001373                          BNE     1$              ;IF NOT LUP BAK & SHIFT
6231  021216  000207                          RTS     PC              ;EXIT FROM THIS SUBROUTINE
6232  021220  000000                  2$:     0
6233
6234
6235
6236
6237
```

```
                                        .SBTTL  CHKHE:  CHECK FOR 'ERR'OR
                                        .SBTTL  CHKHE1: CHECK FOR 'ERR'OR

                                ;;CHKHE
                                ;THIS ROUTINE CHECKS IF 'HE' OR 'ERR' BITS IN RKCS ARE SET. IF ANY OF THE
                                ;TWO BITS ARE SET, THE CONTENTS OF RKCS, ER, DS, AND DA ARE SAVED AND A
                                ;RETURN IS MADE TO THE ERROR MESSAGE FOLLOWING THE 'JSR' CALL.
                                ;AT THE TIME OF ENTRY 'DRIVAD' CONTAINS THE DISK ADDRESS WHICH IS TO
                                ;BE BROKEN DOWN INTO DRIVE #, CYLINDER, SURFACE AND SECTOR #. THIS INFORMATION
                                ;IS SAVED TO BE USED LATER FOR ERROR REPORTING. IF THE BITS ARE NOT SET,
                                ;RETURN IS MADE TO SKIP THE ERROR MESSAGE.

                                ;CHKHE1
                                ;THIS ROUTINE CHECKS IF 'HE' OR 'ERR' BITS IN RKCS ARE SET. IF ANY OF THE
                                ;TWO BITS ARE SET, THE CONTENTS OF RKCS, ER, DS, AND DA ARE SAVED AND A
                                ;RETURN IS MADE TO THE ERROR MESSAGE FOLLOWING THE 'JSR' CALL.
                                ;AT THE TIME OF ENTRY R1 CONTAINS THE DISK ADDRESS WHICH IS TO BE BROKEN
                                ;DOWN INTO DRIVE #, CYLINDER, SURFACE AND SECTOR #. THIS INFORMATION IS
                                ;SAVED TO BE USED LATER FOR ERROR REPORTING. IF THE BITS ARE NOT SET,
                                ;RETURN IS MADE TO SKIP THE ERROR MESSAGE.

021222  010137  001202          CHKHE1: MOV     R1,$REG1O       ;SAVE THE DISK ADRES
021226  000403                          BR      CHE1
                                
021230  013737  001350  001202  CHKHE:  MOV     DRIVAD,$REG1O   ;SAVE THE DISK ADRES
021236  032777  140000  160066  CHE1:   BIT     #140000,@RKCS   ;IS 'HE' OR 'ERR' BIT SET?
021244  001467                          BEQ     CRETRN          ;NO
021246  004737  020770                  JSR     PC,G            ;GET RKCS,ER,DS, DA
021252  104416                          BRKDA4                  ;GO TO 'BDA4' & BREAK CONTENTS O
                                                                ;$REG1O INTO DR#, CYL, SUR, SEC BITS
021254  000207                          RTS     PC              ;RETURN TO THE ERROR MESSAGE
                                
                                
                                
                                        .SBTTL  CHKDA:  CHECK IF RKDA INCREMENTED CORRECTLY

                                ;CHKDA
                                ;THIS ROUTINE CHECKS IF RKDA INCREMENTED CORRECTLY. IF RKDA INCREMENTED
                                ;CORRECTLY RETURN IS MADE TO SKIP THE ERROR MESSAGE.
                                ;IF RKDA DID NOT INCREMENT CORRECTLY, THE EXPECTED AND RECIEVED VALUES
                                ;OF RKDA ARE SAVED AND A RETURN IS MADE TO THE ERROR MESSAGE FOLLOWING THE
                                ;'JSR' CALL.
021256  013705  001350          CHKDA:  MOV     DRIVAD,R5       ;RKDA SHOULD INCREMENT TO THIS
021262  005205                          INC     R5              ;AFTER DATA TRANSFER IS DONE
021264  020577  160050          CHKDA1: CMP     R5,@RKDA        ;DID RKDA INCREMENT CORRECTLY?
021270  001455                          BEQ     CRETRN          ;IF YES, BRANCH
                                                                ;IF NOT, REPORT ERROR
021272  010537  001202                  MOV     R5,$REG1O       ;GET EXPCTD RKDA
021276  104415                          BRKDA0                  ;GO TO 'BDA0' & BREAK  CONTENTS OF
                                                                ;$REG1O INTO DR #,CYL,SUR,SEC BITS
021300  017737  160034  001202          MOV     @RKDA,$REG1O    ;GET ACTUAL RKDA
021306  104416                          BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                                ;$REG1O INTO DR #,CYL,SUR,SEC BITS
021310  000207                          RTS     PC              ;RETURN TO THE ERROR MESSAGE
```

```
                                                .SBTTL  CHKWC:  CHECK IF RKWC OVERFLOWED

                                        ;CHKWC
                                        ;THIS ROUTINE CHECKS IF RKWC OVERFLOWED TO 0. IF IT DID A RETURN IS MADE
                                        ;TO SKIP THE ERROR MESSAGE. IF NOT,THE CONTENTS OF RKWC AND RKDA ARE SAVED
                                        ;AND A RETURN IS MADE TO THE ERROR MESSAGE FOLLOWING THE 'JSR' CALL.
021312  005777  160016                  CHKWC:  TST     @RKWC           ;DID WORD COUNT OVERFLOW TO 0?
021316  001442                                  BEQ     CRETRN          ;IF YES, BRANCH
                                                                        ;IF NOT, ERROR
021320  017737  160010  001162                  MOV     @RKWC,$REG0     ;GET RKWC
021326  017737  160006  001164                  MOV     @RKDA,$REG1     ;GET RKDA
021334  000207                                  RTS     PC              ;RETURN TO THE ERROR MESSAGE


                                                .SBTTL  CHKER:  CHECK RKER CONTENTS
                                        ;CHKER
                                        ;THIS ROUTINE CHECKS IF ANY BIT IN RKER SET. IF NOT RETURN IS MADE TO SKIP
                                        ;THE ERROR MESSAGE. IF ANY BIT IS SET THE CONTENTS OF RKCS, RKER, RKDS ARE
                                        ;SAVED AND A RETURN IS MADE TO THE ERROR MESSAGE.

021336  005777  157766                  CHKER:  TST     @RKER           ;DID ANY BIT IN RKER SET?
021342  001430                                  BEQ     CRETRN          ;NO, BRANCH
                                                                        ;YES, ERROR
021344  004737  020776                          JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS

021350  000207                                  RTS     PC              ;RETURN TO THE ERROR MESSAGE


                                        ;CHKECLR
                                        ;THIS ROUTINE CHECKS THAT RKER IS CLEAR. IF NOT, THE CONTENTS OF RKER
                                        ;ARE SAVED AND A RETURN IS MADE TO THE ERROR MESSAGE FOLLOWING THE "JSR"
                                        ;CALL. IF RKER IS CLEAR THE ERROR MESSAGE IS SKIPPED ON RETURN.

021352  005777  157752                  CHKECLR: TST    @RKER           ;ANY BIT IN RKER SET?
021356  001422                                  BEQ     CRETRN          ;NO
021360  013737  001330  001162                  MOV     RKER,$REG0      ;GET ADRES OF RKER
021366  017737  157736  001164                  MOV     @RKER,$REG1     ;GET CONTENTS OF RKER
021374  000207                                  RTS     PC              ;RETURN TO THE ERROR MESSAGE


                                        ;CHKCCLR
                                        ;THIS ROUTINE CHECKS THAT RKCS IS CLEAR. IF NOT, THE CONTENTS OF RKCS ARE
                                        ;SAVED AND A RETURN IS MADE TO THE ERROR MESSAGE. IF RKCS IS CLEAR THE
                                        ;ERROR MESSAGE IS SKIPPED ON RETURN.
021376  022777  000200  157726          CHKCCLR: CMP    #200,@RKCS      ;IS RKCS CLEAR?
021404  001407                                  BEQ     CRETRN          ;YES
021406  013737  001332  001162                  MOV     RKCS,$REG0      ;SAVE ADRES OF RKCS
021414  017737  157712  001164                  MOV     @RKCS,$REG1     ;SAVE THE CONTENT OF RKCS
021422  000207                                  RTS     PC              ;RETURN TO THE ERROR MESSAGE

021424  062716  000002          CRETRN: ADD     #2,(SP)         ;SKIP ERROR MESSAGE ON
021430  000207                          RTS     PC              ;RETURN


                                                .SBTTL  TSTRWS: WAIT FOR R/W/S RDY ROUTINE
```

```
6350                                    ;TSTRWS
6351                                    ;THIS ROUTINE WAITS FOR R/W/S RDY TO SET. WHEN IT SETS, THE RETURN PC
6352                                    ;IS INCREMENTED SO THAT ON RETURN (TO THE MAIN PROGRAM) THE ERROR
6353                                    ;MESSAGE FOLLOWING THE 'JSR' CALL IS SKIPPED. IF R/W/S RDY DOES NOT SET
6354                                    ;THEN A RETURN IS MADE TO THE ERROR MESSAGE (FOLLOWING THE 'JSR' CALL).
6355                                    ;WAITING TIME IS APPROX. 1040 MS FOR 11/20, APPROX. 208 MS FOR 11/45
6356                                    ;CALL:  JSR     TSTRWS
6357
6358  021432  013777  001350  157700  TSTRWS: MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE
6359  021440  005037  001366          TSTRWS: CLR     TIMER           ;INITIALIZE COUNT
6360  021444  032777  000100  157654  1$:     BIT     #100,@RKDS      ;DID R/W/S RDY SET?
6361  021452  001007                          BNE     2$              ;YES, BRANCH
6362  021454  005237  001366                  INC     TIMER           ;WAIT FOR R/W/S RDY
6363  021460  001371                          BNE     1$              ;ERROR IF IT'S NOT SET BY NOW
6364  021462  017737  157640  001162          MOV     @RKDS,$REGO     ;GET RKDS
6365  021470  000207                          RTS     PC              ;EXIT (TO ERROR FOOLOWING 'JSR TSTRWS')
6366
6367  021472  062716  000002          2$:     ADD     #2,(SP)         ;ADJUST RETURN ADRES TO SKIP OVER
6368                                                                  ;ERROR (FOLLOWING 'JSR TSTRWS')
6369  021476  000207                          RTS     PC              ;EXIT
6370
6371
6372
6373
6374
6375                                                    .SBTTL  DRESET: DRIVE RESET ROUTINE
6376
6377                                    ;DRESET
6378                                    ;THIS ROUTINE DOES A DRIVE RESET ON THE DRIVE WHOOSE ADDRESS IS IN
6379                                    ;RKDA. MULTIPLE RETURN ADDRESSES FOR THIS ROUTINE ARE PROVIDED.
6380                                    ;IF THERE IS NO ERROR (R/W/S RDY SETS WITHIN CERTAIN TIME) , THEN BEFORE
6381                                    ;EXITNG FROM THIS ROUTINE THE RETURN ADDRESS IS INCREMENTED BY 2, TO SKIP
6382                                    ;THE ERROR MESSAGE ON RETURN. IF THERE IS AN ERROR, THE 3 REGISTERS (CS,ER,DS)
6383                                    ;ARE STORED AND THEN A NORMAL EXIT IS MADE FROM THIS ROUTINE TO THE
6384                                    ;ERROR MESSAGE FOLLOWING THE CALL FOR THIS ROUTINE.
6385                                    ;CALL:  JSR     PC,DRESET
6386
6387
6388  021500  005037  001364          DRESET: CLR     COUNT1          ;INITIALIZE THE COUNT
6389  021504  013777  001350  157626          MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE
6390  021512  012777  000015  157612          MOV     #15,@RKCS       ;DRIVE RESET, GO
6391  021520  104414                          CNT.RDY                 ;THIS IS A CALL FOR 'CN.RDY'
6392                                                                  ;ROUTINE WHICH WAITS FOR CNT
6393                                                                  ;RDY TO SET. IF CNTRL RDY DOES
6394                                                                  ;NOT SET WITHIN 883 MS/ 11-20
6395                                                                  ;(176 MS FOR 11-45 WITH BIPOLAR)
6396                                                                  ;AN ERROR IS REPORTED
6397  021522  032777  000100  157576  1$:     BIT     #100,@RKDS              ;DID R/W/S RDY SET?
6398  021530  001013                          BNE     2$
6399  021532  012746  177770                  MOV     #-10.,-(SP)     ;PUSH COUNT ON SP
6400  021536  005216                          INC     (SP)            ;COUNT IT DOWN
6401  021540  001376                          BNE     .-2
6402  021542  005726                          TST     (SP)+           ;POP UP SP
6403  021544  005237  001364                  INC     COUNT1                  ;IF NOT WAIT
6404  021550  001364                          BNE     1$                      ;WAITED LONG?
6405  021552  004737  020770                  JSR     PC,GT4RG
```

```
6406  021556  000402                    BR      2$+4
6407  021560  062716  000002    2$:     ADD     #2,@R6
6408  021564  000207                    RTS     PC
6409
6410
6411
6412
6413                                    .SBTTL  TSTSIN: CHECK 'SIN' ROUTINE
6414
6415                                ;TSTSIN
6416                                ;THIS ROUTINE CHECKS IF 'SIN' IS SET, IF IT IS SET A
6417                                ;DRIVE RESET IS DONE TO CLEAR 'SIN' AND INITIALIZE POSITIONER.
6418                                ;CALL:          TST.SIN
6419                                ;IF ON DOING DRIVE RESET R/W/S RDY DOES NOT SET  A MESSAGE
6420                                ;       ERROR PC=XXXXXX  IS GIVEN.
6421                                ;XXXXXX=PC IN THE MAIN PROGRAM WHERE 'TST.SIN' CALL IS LOCATED.
6422
6423  021566  013777  001350  157544  TSTSIN: MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE
6424  021574  032777  001000  157524          BIT     #1000,@RKDS     ;IS SIN SET?
6425  021602  001403                          BEQ     1$
6426  021604  004737  021500                  JSR     PC,DRESET       ;GO DO DRIVE RESET, SIN  SET
6427  021610  000401                          BR      2$              ;REPORT ERROR
6428  021612  000002                  1$:     RTI
6429  021614  032777  020000  157316  2$:     BIT     #SW13,@SWR      ;INHIBIT TYPEOUT?
6430  021622  001373                          BNE     1$              ;IF YES, SKIP TYPEOUT
6431  021624  104401  021632                  TYPE    ,65$            ;;TYPE ASCIZ STRING
6432  021630  000406                          BR      64$             ;;GET OVER THE ASCIZ
6433                                  ;;65$:  .ASCIZ  /ERROR PC= /
6434  021646                          64$:
6435  021646  011646                          MOV     (SP),-(SP)
6436  021650  062716  177776                  ADD     #-2,(SP)        ;GET THE PC WHERE 'TST.SIN' IS LOCATED
6437  021654  104402                          TYPOC                   ;GO TYPE OUT PC
6438  021656  000755                          BR      1$
6439
6440
6441
6442                                    .SBTTL  DELAY:  TIME DELAY ROUTINE
6443
6444                                ;DELAY
6445                                ;THIS ROUTINE PROVIDES  A VARIABLE TIME DELAY. THE CALL FOR THIS
6446                                ;ROUTINE IS AN ENCODED 'TRAP' INSTRUCTION.
6447                                ;CALL:          DELAY   ,N      N IS ANY OCTAL NO. FROM 1 TO 177777
6448                                ;THE DELAY PROVIDED IS 7.5N US (CONVERT N TO DECIMAL) FOR 11/20
6449                                ;1.5N US FOR 11/45
6450                                ;IF THE USER WANTS TO CHANGE THE DELAY TIME (EXMP: SHORTER DELAY TO
6451                                ;GET A TIGHTER SCOPE LOOP) THE VARIABLE 'N' FOLLOWING 'DELAY' SHOULD
6452                                ;BE CHANGED TO SUIT THE INDIVIDUAL NEED.
6453
6454  021660  017637  000000  001366  DELA.Y: MOV     @(SP),TIMER     ;GET 'AMOUNT' (N) FOR WHICH
6455  021666  062716  000002                  ADD     #2,(SP)         ;DELAY IS TO BE PROVIDED
6456                                                                  ;ADJUST STACK POINTER TO SKIP OVER 'N'
6457  021672  005337  001366          1$:     DEC     TIMER           ;COUNT DOWN TO 0
6458  021676  001375                          BNE     1$
6459
6460  021700  000002                          RTI                     ;RETURN TO MAIN PROGRAM
6461
```

```
                                                .SBTTL  WAT.INT:        WAIT FOR INTERRUPT ROUTINE

                                        ;WAT.INT
                                        ;THIS ROUTINE PROVIDES A VARIABLE TIME WAIT LOOP DURING WHICH AN INTERRUPT
                                        ;FROM RK11 CAN OCCUR. THE CALL IS AN ENCODED 'TRAP' INSTRUCTION.

                                        ;CALL:          WAT.INT ,N      N IS ANY OCTAL NO. FROM 1 TO 177777

                                        ;WAIT LOOP TIME= APPROX. 7.5N US (CONVERT N TO DECIMAL) FOR 11/20
                                        ;APPROX. 1.5N US FOR 11/45
                                        ;UPON ENTERING THE ROUTINE THE CPU PRIORITY IS DROPPED SO THAT
                                        ;RK11 CAN INTERRUPT. NOTE THAT WHEN RK11 INTERRUPTS THIS ROUTINE
                                        ;IS EXITED WITHOUT POPPING THE STACK, THIS POPPING IS DONE AFTER GETTING
                                        ;TO RK11 INTERRUPT HANDLER.
                                        ;IF FOR ANY REASON THE WAIT LOOP TIME HAS TO BE CHANGED IT CAN BE DONE
                                        ;BY SIMPLY CHANGING THE VARIABLE 'N' FOLLOWING THE 'WAT.INT'.

021702  017637  000000  001366  WATINT: MOV     @(SP),TIMER     ;GET 'AMOUNT' (N) FOR WHICH
021710  062716  000002                  ADD     #2,(SP)         ;WAITING IS TO BE DONE
                                                                ;ADJUST STACK POINTER FOR CORRECT RETURN
021714  013746  001400                  MOV     RKPRI,-(SP)     ;DROP CPU PRIORITY SO THAT RK11 CAN
021720  012746  021726                  MOV     #1$,-(SP)       ; INTERRUPT
021724  000002                          RTI
021726  005337  001366          1$:     DEC     TIMER           ;WAIT FOR RK11 TO INTERRUPT
021732  001375                          BNE     1$
                                                                ;IF INTERRUPT HAS NOT OCCURED BY NOW
                                                                ;RETURN AND REPORT ERROR
021734  000002                          RTI                     ;EXIT


                                        ;WATIME

021736  005000                  WATIME: CLR     R0
021740  005001                          CLR     R1
021742  005200                  1$:     INC     R0
021744  001376                          BNE     1$
021746  105201                          INCB    R1
021750  001374                          BNE     1$
021752  000207                          RTS     PC


                                                .SBTT_  CHKCRDY:        CHECK CONTROL READY

                                        ;;CH.CRDY
                                        ;THIS ROUTINE WAITS FOR THE CONTROL READY TO SET. IF THE CONTROL READY BIT
                                        ;DOES NOT SET WITHIN A CERTAIN TIME, THEN THE CONTENTS OF RKCS, RKER, RKDS
                                        ;AND RKDA ARE SAVED AND AN EXIT MADE TO THE ERROR MESSAGE FOLLOWING THE
                                        ;'JSR' CALL FOR THIS ROUTINE.
                                        ;IF CONTROL READY SETS THEN THE RETURN ADDRESS IS ADJUSTED TO SKIP THE
                                        ;ERROR MESSAGE ON RETURN.
                                        ;CALL:  CHKCRDY
                                        ;       ERROR           ;RETURN HERE IF ERROR
```

```
                                        CHKCRDY:        CHECK CONTROL READY

                                        ;       ---             ;RETURN HERE IF NO ERROR

                                        CH.CRDY: CLR TIMER
021754  005037  001366                  1$:     TSTB    @RKCS           ;CNTRL RDY SET?
021760  105777  157346                          BMI     2$              ;YES
021764  100406                                  INC     TIMER
021766  005237  001366                          BNE     1$              ;NO, WAIT
021772  001372                                  JSR     PC,GTHRG        ;SAVE RKCS, ER, DS, DA
021774  004737  020770                          RTI
022000  000002

022002  062716  000002                  2$:     ADD     #2,(SP)         ;ADJUST RETURN ADDRESS TO
022006  000002                                  RTI                     ;SKIP ERROR MESSAGE ON RETURN


                                                .SBTTL  CON.RESET:      CONTROL REST ROUTINE

                                        ;CON.RESET
                                        ;THIS ROUTINE ISSUES A CONTROL RESET AND WAITS FOR
                                        ;THE 'CNTRL RDY' FLAG TO SET.  WHEN THE FLAG SETS
                                        ;AN EXIT IS MADE OUT OF THE ROUTINE.  IF 'CNTRL-RDY'
                                        ;DOES NOT SET WITHIN A CERTAIN TIME AN ERROR MESSAGE
                                        ;       CNT RDY DIDN'T SET
                                        ;       PC=XXXXXX RKCS=YYYYYY
                                        ;IS GIVEN. NOTE THAT XXXXXX IS THE PC WHERE 'CNT.RESET' OR 'CNT.RDY'
                                        ;IS CALLED.

                                        ;CALL:  CNT.RESET



                                                .SBTTL  CNT.RDY:        WAIT FOR CONTROL READY ROUTINE

                                        ;CN.RDY
                                        ;THIS ROUTINE WAITS FOR THE CONTROL READY BIT TO SET AND WHEN IT
                                        ;SETS EXITS OUT. IF WITHIN A CERTAIN TIME CNTRL RDY DOES
                                        ;NOT SET AN ERROR IS REPORTED. WAITING TIME IS 883 MS FOR 11/20
                                        ;175 MS FOR 11/45 WITH BIPOLAR MEMORY.
                                        ;CALL:  CNT.RDY
022010  012777  000001  157314          CN.RST: MOV     #1,@RKCS        ;ISSUE A CONTROL RESET
022016  012737  177500  001170                  MOV     #-300,$REG3     ;SET UP COUNT
022024  000402                                  BR      CN.RDY+4                ;SKIP OVER CN.RDY
022026  005037  001170                  CN.RDY: CLR     $REG3
022032  105777  157274                  1$:     TSTB    @RKCS           ;DID CNTRL-RDY SET?
022036  100435                                  BMI     3$              ;YES, EXIT
022040  005237  001170                          INC     $REG3           ;WAITED LONG?
022044  001372                                  BNE     1$              ;IF NOT, GO BAK & WAIT
022046  032777  020000  157064          2$:     BIT     #SW13,@SWR      ;INHIBIT TYPEOUT?
022054  001026                                  BNE     3$              ;IF YES, SKIP TYPEOUT
022056  104401                                  TYPE
022060  001245                                  MSG3
022062  104401  022070                          TYPE    ,65$            ;;TYPE ASCIZ STRING
022066  000403                                  BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <15><12>/PC=/
022076                                  64$:
022076  011646                                  MOV     (SP),-(SP)
```

```
                        CNT.RDY:        WAIT FOR CONTROL READY ROUTINE

                                SUB     #2,(SP)
                                TYPOC                   ;GO TYPE PC IN THE MAIN PROGRAM,
                                                        ; WHERE ERROR OCCURRED
                                TYPE    ,67$            ;;TYPE ASCIZ STRING
                                BR      66$             ;;GET OVER THE ASCIZ
                        ;;67$:  .ASCIZ  / RKCS=/
                        66$:
                                MOV     @RKCS,-(SP)     ;GET RKCS
                                TYPOC                   ;GO TYPE IT

                        3$:     RTI                     ;RETURN FROM THIS
                                                        ;ROUTINE TO THE MAIN
                                                        ;PROGRAM


                        ;THIS PART OF THE PROGRAM CONTAINS THE COMMON ROUTINES CALLED
                        ;FROM THE SYSMAC.SML PACKAGE
                        ;

                        .SBTTL  SCOPE HANDLER ROUTINE

                        ;;*****************************************************************
                        ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                        ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                        ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                        ;*SW14=1         LOOP ON TEST
                        ;*SW11=1         INHIBIT ITERATIONS
                        ;*SW09=1         LOOP ON ERROR
                        ;*SW08=1         LOOP ON TEST IN SWR<7:0>
                        ;*CALL
                        ;*      SCOPE           ;;SCOPE=IOT

                        $SCOPE:
                                CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
                        1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
                                BNE     $OVER           ;;YES IF SW14=1
                        ;#####START OF CODE FOR THE XOR TESTER#####
                        $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
                                MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
                                MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
                                TST     @#177060        ;;TIME OUT ON XOR?
                                MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
                                BR      $SVLAD          ;;GO TO THE NEXT TEST
                        5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
                                MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
                                BR      7$              ;;LOOP ON THE PRESENT TEST
                        6$:;#####END OF CODE FOR THE XOR TESTER#####
                                BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
                                BEQ     2$              ;;BR IF NO
                                CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
                                BEQ     $OVER           ;;BR IF YES
                        2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
                                BEQ     3$              ;;BR IF NO
                                CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
```

```
 6630  022240  101015                          BHI     3$                      ;;BR IF NO
 6631  022242  032777  001000  156670          BIT     #BIT09,@SWR             ;;LOOP ON ERROR?
 6632  022250  001404                          BEQ     4$                      ;;BR IF NO
 6633  022252  013737  001110  001106  7$:     MOV     $LPERR,$LPADR           ;;SET LOOP ADDRESS TO LAST SCOPE
 6634  022260  000443                          BR      $OVER
 6635  022262  105037  001103          4$:     CLRB    $ERFLG                  ;;ZERO THE ERROR FLAG
 6636  022266  005037  001206                  CLR     $TIMES                  ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
 6637  022272  000415                          BR      1$                      ;;ESCAPE TO THE NEXT TEST
 6638  022274  032777  004000  156636  3$:     BIT     #BIT11,@SWR             ;;INHIBIT ITERATIONS?
 6639  022302  001011                          BNE     1$                      ;;BR IF YES
 6640  022304  005737  001100                  TST     $PASS                   ;;IF FIRST PASS OF PROGRAM
 6641  022310  001406                          BEQ     1$                      ;;      INHIBIT ITERATIONS
 6642  022312  005237  001104                  INC     $ICNT                   ;;INCREMENT ITERATION COUNT
 6643  022316  023737  001206  001104          CMP     $TIMES,$ICNT            ;;CHECK THE NUMBER OF ITERATIONS MADE
 6644  022324  002021                          BGE     $OVER                   ;;BR IF MORE ITERATION REQUIRED
 6645  022326  012737  000001  001104  1$:     MOV     #1,$ICNT                ;;REINITIALIZE THE ITERATION COUNTER
 6646  022334  013737  022404  001206          MOV     $MXCNT,$TIMES           ;;SET NUMBER OF ITERATIONS TO DO
 6647  022342  105237  001102          $SVLAD: INCB    $TSTNM                  ;;COUNT TEST NUMBERS
 6648  022346  011637  001106                  MOV     (SP),$LPADR             ;;SAVE SCOPE LOOP ADDRESS
 6649  022352  011637  001110                  MOV     (SP),$LPERR             ;;SAVE ERROR LOOP ADDRESS
 6650  022356  005037  001210                  CLR     $ESCAPE                 ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 6651  022362  112737  000001  001115          MOVB    #1,$ERMAX               ;;ONLY ALLOW ONE(1) ERROR ON NEXT TES
 6652  022370  013777  001102  156544  $OVER:  MOV     $TSTNM,@DISPLAY         ;;DISPLAY TEST NUMBER
 6653  022376  013716  001106                  MOV     $LPADR,(SP)             ;;FUDGE RETURN ADDRESS
 6654  022402  000002                          RTI                             ;;FIXES PS
 6655  022404  000050          $MXCNT: 50                                      ;;MAX. NUMBER OF ITERATIONS
 6656
 6657
 6658                                  ;;**********************************************************************
 6659
 6660                                  .SBTTL  ERROR HANDLER ROUTINE
 6661
 6662                                  ;*SW15=1        HALT ON ERROR
 6663                                  ;*SW13=1        INHIBIT ERROR TYPEOUTS
 6664                                  ;*SW10=1        TESTING ON SIMULATOR
 6665                                  ;*SW09=1        LOOP ON ERROR
 6666                                  ;*SW12=1        CYCLE ON ERROR TO PREVIOUS 'SCOPE'
 6667                                  ;*SW06=1        DROP DRIVE AFTER MAXIMUM (ALLOWABLE) ERRORS ON THE DRIVE
 6668                                  ;*GO TO $ERRTYP ON ERROR
 6669
 6670  022406  104407          $ERROR: CKSWR                                   ;CHECK FOR SOFTWARE SWITCH REGISTER REQUEST
 6671  022410  105237  001103  7$:     INCB    $ERFLG                          ;SET THE ERROR FLAG
 6672  022414  001775                  BEQ     7$                              ;DON'T LET THE FLAG GO TO ZERO
 6673  022416  013777  001102  156516          MOV     $TSTNM,@DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
 6674  022424  005237  001112  1$:     INC     $ERTTL                          ;COUNT THE NUMBER OF ERRORS
 6675
 6676  022430  032777  000100  156502          BIT     #BIT6,@SWR              ;DESELECT DRIVE SW SET?
 6677  022436  001404                          BEQ     6$                      ;NO
 6678  022440  023727  001112  000005          CMP     $ERTTL,#5               ;MORE THAN 5 ERRORS ON THIS DRIVE?
 6679  022446  101053                          BHI     8$                      ;YES, DESELCT THE DRIVE
 6680
 6681  022450  011637  001116  6$:     MOV     (SP),$ERRPC             ;GET ADDRESS OF ERROR INSTRUCTION
 6682  022454  162737  000002  001116          SUB     #2,$ERRPC
 6683  022462  117737  156430  001114          MOVB    @$ERRPC,$ITEMB          ;STRIP AND SAVE THE ERROR ITEM CODE
 6684  022470  032777  020000  156442          BIT     #SW13,@SWR              ;SKIP TYPEOUT IF SET
 6685  022476  001004                          BNE     2$                      ;SKIP TYPEOUTS
```

```
6686  022500  004737  022730                  JSR     PC,@#$ERRTYP            ;GO TO USER ERROR ROUTINE
6687  022504  104401  001213                  TYPE    ,$CRLF
6688  022510  023737  000042  000046  2$:     CMP     @#42,@#46               ;ARE WE IN ACT11 AUTO MODE?
6689  022516  001403                          BEQ     .+10                    ;YES, HALT ON ERROR
6690  022520  005777  156414                  TST     @SWR                    ;HALT ON ERROR?
6691  022524  100002                          BPL     3$                      ;SKIP IF CONTINUE
6692  022526  000000                          HALT                            ;HALT ON ERROR!
6693  022530  104407                          CKSWR                           ;CHECK FOR SOFTWARE SWITCH REGISTER REQUEST
6694  022532  032777  010000  156400  3$:     BIT     #SW12,@SWR              ;SW 12 SET?
6695  022540  001402                          BEQ     .+6                     ;NO, BRANCH
6696  022542  013716  001106                  MOV     $LPADR,(SP)             ;ADJUST RETURN ADRES FOR SW12
6697  022546  032777  001000  156364          BIT     #SW09,@SWR              ;LOOP ON ERROR SWITCH SET?
6698  022554  001402                          BEQ     4$                      ;BR IF NO
6699  022556  013716  001110                  MOV     $LPERR,(SP)             ;FUDGE RETURN FOR LOOPING
6700  022562  005737  001210          4$:     TST     $ESCAPE                 ;CHECK FOR AN ESCAPE ADDRESS
6701  022566  001402                          BEQ     5$                      ;BR IF NONE
6702  022570  013716  001210                  MOV     $ESCAPE,(SP)            ;FUDGE RETURN ADDRESS FOR ESCAPE
6703  022574  000002                  5$:     RTI                             ;RETURN
6704
6705  022576  005737  001434          8$:     TST     TS6FLG                  ;IF EROR WAS IN LAST TEST (POL
6706                                                                          ;DROP ALL THE DRIVES
6707  022602  001407                          BEQ     10$
6708  022604  104401  001303                  TYPE    ,MSG5
6709  022610  005037  001412                  CLR     DRIVS
6710  022614  022626                          CMP     (SP)+,(SP)+
6711  022616  000137  020646                  JMP     $EOP
6712  022622  013746  001354          10$:    MOV     DRVPTR,-(SP)            ;DROP THE DRIVE FROM THE
6713  022626  162716  000002                  SUB     #2,(SP)                 ;SELECTION LIST
6714  022632  013746  001350                  MOV     DRIVAD,-(SP)            ;DRIVE ADDR TO STACK
6715  022636  004737  021174                  JSR     PC,SHFTRT               ;RIGHT JUSTIFY
6716  022642  042716  000001                  BIC     #1,(R6)                 ;MAKE EVEN
6717  022646  062716  001414                  ADD     #DRIV0,(SP)             ;POINTS TO TABLE FOR EVEN DRIVE
6718  022652  042776  100000  000000          BIC     #BIT15,@(R6)            ;TEST REMAINING DRIVE AS RK05E
6719  022660  062716  000002                  ADD     #2,(R6)                 ;POINT TO ODD
6720  022664  042736  100000                  BIC     #BIT15,@(SP)+           ;TEST AS RK-05E
6721  022670  012736  010000                  MOV     #BIT12,@(SP)+           ;INDICATE THIS DRIVE DROPPED
6722  022674  104401  001272                  TYPE    ,MSG4
6723  022700  013746  001350                  MOV     DRIVAD,-(R6)            ;PUSH DRIVE # ON STACK
6724  022704  004737  021174                  JSR     PC,SHFTRT               ;SHIFT IT BEFORE TYPING
6725  022710  104402                          TYPOC                           ;TYPE OUT DRIVE #
6726  022712  104401  001315                  TYPE    ,MSG6
6727  022716  005337  001412                  DEC     DRIVS                   ;DECREMENT # OF DRIVES PRESNT
6728  022722  022626                  9$:     CMP     (SP)+,(SP)+             ;RESTORE STACK
6729  022724  000137  020022                  JMP     BTEOP                   ;GO BACK TO THE END OF PROGRM
6730                                                                          ;LINKAGE.
6731
6732                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
6733
6734                                  ;;**********************************************************************
6735                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
6736                                  ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
6737                                  ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
6738
6739  022730                          $ERRTYP:
6740  022730  104401  001213                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
6741  022734  010046                          MOV     R0,-(SP)        ;;SAVE R0
```

```
6742  022736  005000                    CLR     R0              ;;PICKUP THE ITEM INDEX
6743  022740  153700  001114            BISB    @#$ITEMB,R0
6744  022744  001004                    BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
6745                                                            ;;TYPE THE PC OF THE ERROR
6746  022746  013746  001116            MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
6747                                                            ;;ERROR ADDRESS
6748  022752  104402                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
6749  022754  000426                    BR      6$              ;;GET OUT
6750  022756  005300          1$:       DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
6751  022760  006300                    ASL     R0              ;;      WORK FOR THE ERROR TABLE
6752  022762  006300                    ASL     R0
6753  022764  006300                    ASL     R0
6754  022766  062700  001442            ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
6755  022772  012037  023002            MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
6756  022776  001404                    BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
6757  023000  104401                    TYPE                    ;;TYPE THE "ERROR MESSAGE"
6758  023002  000000          2$:       .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
6759  023004  104401  001213            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
6760  023010  012037  023020  3$:       MOV     @(R0)+,4$       ;;PICKUP "DATA HEADER" POINTER
6761  023014  001404                    BEQ     5$              ;;SKIP TYPEOUT IF 0
6762  023016  104401                    TYPE                    ;;TYPE THE "DATA HEADER"
6763  023020  000000          4$:       .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
6764  023022  104401  001213            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
6765  023026  011000          5$:       MOV     @(R0),R0        ;;PICKUP "DATA TABLE" POINTER
6766  023030  001004                    BNE     7$              ;;GO TYPE THE DATA
6767  023032  012600          6$:       MOV     (SP)+,R0        ;;RESTORE R0
6768  023034  104401  001213            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
6769  023040  000207                    RTS     PC              ;;RETURN
6770  023042                    7$:
6771  023042  013046                    MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
6772  023044  104402                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
6773  023046  005710                    TST     (R0)            ;;IS THERE ANOTHER NUMBER
6774  023050  001770                    BEQ     6$              ;;BR IF NO
6775  023052  104401  023060            TYPE    ,8$             ;;TYPE TWO(2) SPACES
6776  023056  000771                    BR      7$              ;;LOOP
6777  023060  020040      000   8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
6778          023064                    .EVEN
6779
6780                            .SBTTL  TYPE ROUTINE
6781
6782                            ;;**********************************************************************
6783                            ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
6784                            ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
6785                            ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
6786                            ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
6787                            ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
6788                            ;*
6789                            ;*CALL:
6790                            ;*1) USING A TRAP INSTRUCTION
6791                            ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
6792                            ;*OR
6793                            ;*      TYPE
6794                            ;*      MESADR
6795                            ;*
6796
6797  023064  105737  001157    $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL
```

```
6798  023070  100002                  BPL     1$              ;;BR IF YES
6799  023072  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
6800  023074  000407                  BR      3$              ;;LEAVE
6801  023076  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
6802  023100  017600  000002          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
6803  023104  112046          2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
6804  023106  001005                  BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
6805  023110  005726                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
6806  023112  012600          60$:    MOV     (SP)+,R0        ;;RESTORE R0
6807  023114  062716  000002  3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
6808  023120  000002                  RTI                     ;;RETURN
6809  023122  122716  000011  4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
6810  023126  001430                  BEQ     8$
6811  023130  122716  000200          CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
6812  023134  001006                  BNE     5$
6813  023136  005726                  TST     (SP)+           ;;POP  <CR><LF> EQUIV
6814  023140  104401                  TYPE                    ;;TYPE A CR AND LF
6815  023142  001213                  $CRLF
6816  023144  105037  023300          CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
6817  023150  000755                  BR      2$              ;;GET NEXT CHARACTER
6818  023152  004737  023234  5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
6819  023156  123726  001156  6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
6820  023162  001350                  BNE     2$              ;;IF NO GO GET NEXT CHAR.
6821  023164  013746  001154          MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
6822                                                          ;;AND THE NULL CHAR.
6823  023170  105366  000001  7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
6824  023174  002770                  BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
6825  023176  004737  023234          JSR     PC,$TYPEC       ;;GO TYPE A NULL
6826  023202  105337  023300          DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
6827  023206  000770                  BR      7$              ;;LOOP
6828
6829                                  ;HORIZONTAL TAB PROCESSOR
6830
6831  023210  112716  000040  8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
6832  023214  004737  023234  9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
6833  023220  132737  000007  023300  BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
6834  023226  001372                  BNE     9$              ;;TAB STOP
6835  023230  005726                  TST     (SP)+           ;;POP SPACE OFF STACK
6836  023232  000724                  BR      2$              ;;GET NEXT CHARACTER
6837  023234  105777  155710  $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
6838  023240  100375                  BPL     $TYPEC
6839  023242  116677  000002  155702  MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
6840  023250  122766  000015  000002  CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
6841  023256  001003                  BNE     1$              ;;BRANCH IF NO
6842  023260  105037  023300          CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
6843  023264  000406                  BR      $TYPEX          ;;EXIT
6844  023266  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
6845  023274  001402                  BEQ     $TYPEX          ;;BRANCH IF YES
6846  023276  105227                  INCB    (PC)+           ;;COUNT THE CHARACTER
6847  023300  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
6848  023302  000207          $TYPEX: RTS     PC
6849
6850
6851                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
6852
6853                                  ;;**************************************************************
```

```
                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                        ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                        ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                        ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                        ;*REPLACED WITH SPACES.
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                        ;*      TYPDS                   ;;GO TO THE ROUTINE
                                        $TYPDS:
023304  010046                          $TYPDS: MOV     R0,-(SP)        ;;PUSH R0 ON STACK
023306  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
023310  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
023312  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
023314  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
023316  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
023322  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
023326  100004                          BPL     1$              ;;BR IF INPUT IS POS.
023330  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
023332  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
023340  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
023342  012703  023520                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
023346  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
023352  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
023354  016001  023510                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
023360  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
023362  002402                          BLT     4$              ;;BR IF DONE
023364  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
023366  000774                          BR      3$
023370  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
023372  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
023374  001002                          BNE     5$              ;;FALL THROUGH IF 0
023376  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
023400  100407                          BMI     7$              ;;BR IF YES
023402  106316                  5$:     ASLB    (SP)            ;;MSD?
023404  103003                          BCC     6$              ;;BR IF NO
023406  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
023414  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
023420  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
023424  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
023426  005720                          TST     (R0)+           ;;JUST INCREMENTING
023430  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
023434  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
023436  003002                          BGT     8$              ;;GO TO EXIT
023440  010502                          MOV     R5,R2           ;;GET THE LSD
023442  000764                          BR      6$              ;;GO CHANGE TO ASCII
023444  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
023446  100003                          BPL     9$              ;;BR IF NO
023450  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
023456  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
023460  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
023462  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
023464  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
023466  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
023470  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
023472  104401  023520                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
```

```
6910  023476  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
6911  023504  012616                          MOV     (SP)+,(SP)
6912  023506  000002                          RTI                     ;;RETURN TO USER
6913  023510  023420                  $DTBL:  10000.
6914  023512  001750                          1000.
6915  023514  000144                          100.
6916  023516  000012                          10.
6917  023520  000004                  $DBLK:  .BLKW   4
6918
6919                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
6920
6921                                  ;;**************************************************************
6922                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
6923                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
6924                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
6925                                  ;*CALL:
6926                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
6927                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
6928                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
6929                                  ;*      .BYTE   M               ;;M=1 OR 0
6930                                  ;*                                      ;;1=TYPE LEADING ZEROS
6931                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
6932                                  ;*
6933                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
6934                                  ;*$TYPOS OR $TYPOC
6935                                  ;*CALL:
6936                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
6937                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
6938                                  ;*
6939                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
6940                                  ;*CALL:
6941                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
6942                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
6943
6944  023530  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
6945  023534  116637  000001  023753          MOVB    1.SP),$OFILL    ;;LOAD ZERO FILL SWITCH
6946  023542  112637  023755                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
6947  023546  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
6948  023552  000406                          BR      $TYPON
6949  023554  112737  000001  023753  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
6950  023562  112737  000006  023755          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
6951  023570  112737  000005  023752  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
6952  023576  010346                          MOV     R3,-(SP)        ;;SAVE R3
6953  023600  010446                          MOV     R4,-(SP)        ;;SAVE R4
6954  023602  010546                          MOV     R5,-(SP)        ;;SAVE R5
6955  023604  113704  023755                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
6956  023610  005404                          NEG     R4
6957  023612  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
6958  023616  110437  023754                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
6959  023622  113704  023753                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
6960  023626  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
6961  023632  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
6962  023634  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
6963  023636  000404                          BR      3$              ;;GO DO MSB
6964  023640  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
6965  023642  006105                          ROL     R5
```

```
 6966  023644  006105                        ROL     R5
 6967  023646  010503                        MOV     R5,R3
 6968  023650  006103                3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 6969  023652  105337  023754                DECB    $OMODE          ;;TYPE THIS DIGIT?
 6970  023656  100016                        BPL     7$              ;;BR IF NO
 6971  023660  042703  177770                BIC     #177770,R3      ;;GET RID OF JUNK
 6972  023664  001002                        BNE     4$              ;;TEST FOR 0
 6973  023666  005704                        TST     R4              ;;SUPPRESS THIS 0?
 6974  023670  001403                        BEQ     5$              ;;BR IF YES
 6975  023672  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 6976  023674  052703  000060                BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 6977  023700  052703  000040          5$:   BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 6978  023704  110337  023750                MOVB    R3,8$           ;;SAVE FOR TYPING
 6979  023710  104401  023750                TYPE    ,8$             ;;GO TYPE THIS DIGIT
 6980  023714  105337  023752          7$:   DECB    $OCNT           ;;COUNT BY 1
 6981  023720  003347                        BGT     2$              ;;BR IF MORE TO DO
 6982  023722  002402                        BLT     6$              ;;BR IF DONE
 6983  023724  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 6984  023726  000744                        BR      2$              ;;GO DO THE LAST DIGIT
 6985  023730  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
 6986  023732  012604                        MOV     (SP)+,R4        ;;RESTORE R4
 6987  023734  012603                        MOV     (SP)+,R3        ;;RESTORE R3
 6988  023736  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 6989  023744  012616                        MOV     (SP)+,(SP)
 6990  023746  000002                        RTI                     ;;RETURN
 6991  023750     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 6992  023751     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 6993  023752     000                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 6994  023753     000                $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 6995  023754  000000                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 6996
 6997                                .SBTTL  TTY INPUT ROUTINE
 6998
 6999                                ;;**************************************************************
 7000                                .ENABL  LSB
 7001
 7002                                ;;**************************************************************
 7003                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
 7004                                ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
 7005                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
 7006                                ;*WHEN OPERATING IN TTY FLAG MODE.
 7007  023756  022737  000176  001140 $CKSWR: CMP     #$SWREG,SWR     ;;IS THE SOFT-SWR SELECTED?
 7008  023764  001074                        BNE     15$             ;;BRANCH IF NO
 7009  023766  105777  155152                TSTB    @$TKS           ;;CHAR THERE?
 7010  023772  100071                        BPL     15$             ;;IF NO, DON'T WAIT AROUND
 7011  023774  117746  155146                MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
 7012  024000  042716  177600                BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
 7013  024004  022726  000007                CMP     #7,(SP)+        ;;IS IT A CONTROL G?
 7014  024010  001062                        BNE     15$             ;;NO, RETURN TO USER
 7015  024012  123727  001134  000001        CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
 7016  024020  001456                        BEQ     15$             ;;BRANCH IF YES
 7017
 7018  024022  104401  024643                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
 7019  024026  104401  024650          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
 7020  024032  013746  000176                MOV     $SWREG,-(SP)    ;;SAVE SWREG FOR TYPEOUT
 7021  024036  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS
```

```
7022  024040  104401  024661          TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
7023  024044  005046                  CLR     -(SP)           ;;CLEAR COUNTER
7024  024046  005046                  CLR     -(SP)           ;;THE NEW SWR
7025  024050  105777  155070  7$:     TSTB    @$TKS           ;;CHAR THERE?
7026  024054  100375                  BPL     7$              ;;IF NOT TRY AGAIN
7027
7028  024056  117746  155064          MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
7029  024062  042716  177600          BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
7030
7031
7032
7033  024066  021627  000025  9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
7034  024072  001005                  BNE     10$             ;;BRANCH IF NOT
7035  024074  104401  024636          TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
7036  024100  062706  000006  20$:     ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
7037  024104  000757                  BR      19$             ;;LET'S TRY IT AGAIN
7038
7039
7040  024106  021627  000015  10$:     CMP     (SP),#15        ;;IS IT A <CR>?
7041  024112  001022                  BNE     16$             ;;BRANCH IF NO
7042  024114  005766  000004          TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
7043  024120  001403                  BEQ     11$             ;;BRANCH IF YES
7044  024122  016677  000002  155010  MOV     2(SP),@SWR      ;;SAVE NEW SWR
7045  024130  062706  000006  11$:     ADD     #6,SP           ;;CLEAR UP STACK
7046  024134  104401  001213  14$:     TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
7047  024140  123727  001135  000001  CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
7048  024146  001003                  BNE     15$             ;;BRANCH IF NOT
7049  024150  012777  000100  154766  MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
7050  024156  000002          15$:     RTI                     ;;RETURN
7051  024160  004737  023234  16$:     JSR     PC,$TYPEC       ;;ECHO CHAR
7052  024164  021627  000060          CMP     (SP),#60        ;;CHAR < 0?
7053  024170  002420                  BLT     18$             ;;BRANCH IF YES
7054  024172  021627  000067          CMP     (SP),#67        ;;CHAR > 7?
7055  024176  003015                  BGT     18$             ;;BRANCH IF YES
7056  024200  042726  000060          BIC     #60,(SP)+       ;;STRIP-OFF ASCII
7057  024204  005766  000002          TST     2(SP)           ;;IS THIS THE FIRST CHAR
7058  024210  001403                  BEQ     17$             ;;BRANCH IF YES
7059  024212  006316                  ASL     (SP)            ;;NO, SHIFT PRESENT
7060  024214  006316                  ASL     (SP)            ;;    CHAR OVER TO MAKE
7061  024216  006316                  ASL     (SP)            ;;    ROOM FOR NEW ONE.
7062  024220  005266  000002  17$:     INC     2(SP)           ;;KEEP COUNT OF CHAR
7063  024224  056616  177776          BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
7064  024230  000707                  BR      7$              ;;GET THE NEXT ONE
7065  024232  104401  001212  18$:     TYPE    ,$QUES          ;;TYPE ?<CR><LF>
7066  024236  000720                  BR      20$             ;;SIMULATE CONTROL-U
7067                        .DSABL  LSB
7068
7069
7070                        ;;*****************************************************************
7071                        ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
7072                        ;*CALL:
7073                        ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
7074                        ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
7075                        ;*                              ;;WITH PARITY BIT STRIPPED OFF
7076                        ;
7077
```

```
                                ;RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
                                        MOV     4(SP),2(SP)     ;;SAVE THE PS
                                1$:     TSTB    @$TKS           ;;WAIT FOR
                                        BPL     1$              ;;A CHARACTER
                                        MOVB    @$TKB,4(SP)     ;;READ THE TTY
                                        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
                                        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
                                        BNE     3$              ;;BRANCH IF NO
                                2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
                                        BPL     2$              ;;LOOP UNTIL ITS THERE
                                        MOVB    @$TKB,-(SP)     ;;GET CHARACTER
                                        BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
                                        CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
                                        BNE     2$              ;;IF NOT DISCARD IT
                                        BR      1$              ;;YES, RESUME
                                3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
                                        BLT     4$              ;;BRANCH IF YES
                                        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
                                        BGT     4$              ;;BRANCH IF YES
                                        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
                                4$:     RTI                     ;;GO BACK TO USER
                                ;;**************************************************************
                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                ;*CALL:
                                ;*      RDLIN                   ;;INPUT A STRIN  FROM THE TTY
                                ;*      RETURN HERE             ;;ADDRESS OF FI   CHARACTER WILL BE ON THE STACK
                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

                                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
                                        CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
                                1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
                                2$:     CMP     #$TTYIN+22,R3   ;;BUFFER FULL?
                                        BLOS    4$              ;;BR IF YES
                                        RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
                                        MOVB    (SP)+,(R3)      ;;GET CHARACTER
                                10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
                                        BNE     5$              ;;BR IF NO
                                        TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
                                        BNE     6$              ;;BR IF NO
                                        MOVB    #'\,9$          ;;TYPE A BACK SLASH
                                        TYPE    ,9$
                                        MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
                                6$:     DEC     R3              ;;BACKUP BY ONE
                                        CMP     R3,#$TTYIN      ;;STACK EMPTY?
                                        BLO     4$              ;;BR IF YES
                                        MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
                                        TYPE    ,9$             ;;GO TYPE
                                        BR      2$              ;;GO READ ANOTHER CHAR.
                                5$:     TST     (SP)            ;;RUBOUT KEY SET?
                                        BEQ     7$              ;;BR IF NO
                                        MOVB    #'\,9$          ;;TYPE A BACK SLASH
                                        TYPE    ,9$
                                        CLR     (SP)            ;;CLEAR THE RUBOUT KEY
                                7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
                                        BNE     8$              ;;BR IF NO
                                        TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
```

```
7134  024506  000726                          BR      1$              ;;GO START OVER
7135  024510  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
7136  024514  001011                          BNE     3$              ;;BRANCH IF NO
7137  024516  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
7138  024520  104401  001213                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
7139  024524  104401  024614                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
7140  024530  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
7141  024532  104401  001212          4$:     TYPE    ,$QUES          ;;TYPE A '?'
7142  024536  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
7143  024540  111337  024612          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
7144  024544  104401  024612                  TYPE    ,9$
7145  024550  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
7146  024554  001305                          BNE     2$              ;;LOOP IF NOT RETURN
7147  024556  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
7148  024562  104401  001214                  TYPE    ,$LF            ;;TYPE A LINE FEED
7149  024566  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
7150  024570  012603                          MOV     (SP)+,R3        ;;RESTORE R3
7151  024572  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
7152  024574  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
7153  024602  012766  024614  000004          MOV     #$TTYIN,4(SP)
7154  024610  000002                          RTI                     ;;RETURN
7155  024612     000          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
7156  024613     000                  .BYTE   0                       ;;TERMINATOR
7157  024614  000022          $TTYIN: .BLKB   22                      ;;RESERVE 22 BYTES FOR TTY INPUT
7158  024636  052536  005015     000  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
7159  024643     136  006507  000012  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
7160  024650  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
7161  024656  020075     000
7162  024661     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
7163  024666  036440  000040
7164                                                  ;CONTROL U, RUBOUT CAPABILITY
7165                                  .SBTTL  TRAP DECODER
7166
7167                                  ;;**************************************************************
7168                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
7169                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
7170                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
7171                                  ;*GO TO THAT ROUTINE.
7172
7173  024672  010046          $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
7174  024674  016600  000002          MOV     2(SP),R0        ;;GET TRAP ADDRESS
7175  024700  005740                  TST     -(R0)           ;;BACKUP BY 2
7176  024702  111000                  MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
7177  024704  006300                  ASL     R0              ;;POSITION FOR INDEXING
7178  024706  016000  024726          MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
7179  024712  000200                  RTS     R0              ;;GO TO ROUTINE
7180
7181
7182                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
7183
7184  024714  011646          $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
7185  024716  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
7186  024724  000002                  RTI                     ;;RESTORE THE PSW
7187
7188                                  .SBTTL  TRAP TABLE
7189
```

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

                                        ;       ROUTINE
                                        ;       -------
024726  024714                  $TRPAD: .WORD   $TRAP2
024730  023064                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
024732  023554                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
024734  023530                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
024736  023570                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
024740  023304                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

024742  024026                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

024744  023756                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
024746  024240                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
024750  024360                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE

024752  021754                          CH.CRDY ;;CALL=CHKCRDY  TRAP+12(104412) CHECK CONTROL READY

024754  022010                          CN.RST  ;;CALL=CNT.RESET        TRAP+13(104413) CONTROL RESET ROUTINE

024756  022026                          CN.RDY  ;;CALL=CNT.RDY  TRAP+14(104414) WAIT FOR CNTRL RDY TO SET

024760  021052                          BDA0    ;;CALL=BRKDA0   TRAP+15(104415) BREAK RKDA INTO DR #,CYL,SUR,SEC BITS

024762  021062                          BDA4    ;;CALL=BRKDA4   TRAP+16(104416) BREAK RKDA INTO DR #,CYL,SUR,SEC BITS

024764  021660                          DELA.Y  ;;CALL=DELAY    TRAP+17(104417) TIME DELAY ROUTINE

024766  021702                          WATINT  ;;CALL=WAT.INT  TRAP+20(104420) WAIT FOR RK11 INTERRUPT ROUTINE

024770  021566                          TSTSIN  ;;CALL=TST.SIN  TRAP+21(104421) TEST SIN ROUTINE


                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;**************************************************************
                                ;POWER DOWN ROUTINE
024772  012737  025136  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
025000  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
025006  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
025010  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
025012  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
025014  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
025016  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
025020  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
025022  017746  154112                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
025026  010637  025142                  MOV     SP,$SAVR6       ;;SAVE SP
025032  012737  025044  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
025040  000000                          HALT
025042  000776                          BR      .-2             ;;HANG UP

                                ;;**************************************************************
                                ;POWER UP ROUTINE
025044  012737  025136  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
```

```
7246  025052  013706  025142            MOV     $SAVR6,SP       ;;GET SP
7247  025056  005037  025142            CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
7248  025062  005237  025142      1$:   INC     $SAVR6          ;;WAIT FOR THE INC
7249  025066  001375                    BNE     1$              ;;OF  WORD
7250  025070  012677  154044            MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
7251  025074  012605                    MOV     (SP)+,R5        ;;POP STACK INTO R5
7252  025076  012604                    MOV     (SP)+,R4        ;;POP STACK INTO R4
7253  025100  012603                    MOV     (SP)+,R3        ;;POP STACK INTO R3
7254  025102  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
7255  025104  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
7256  025106  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
7257  025110  012737  024772  000024    MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
7258  025116  012737  000340  000026    MOV     #340,@#PWRVEC+2 ;;PRIO:7
7259  025124  104401                    TYPE                    ;REPORT THE POWER FAILURE
7260  025126  025144            $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
7261  025130  012716                    MOV     (PC)+,(SP)      ;;RESTART AT PFSTRT
7262  025132  004702            $PWRAD: .WORD   PFSTRT           ;;RESTART ADDRESS
7263  025134  000002                    RTI
7264  025136  000000            $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
7265  025140  000776                    BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
7266  025142  000000            $SAVR6: 0                       ;;PUT THE SP HERE
7267  025144  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
7268  025152  000122
7269                                    .EVEN
7270
7271  025154  004737  021500    FCHECK: JSR     PC,DRESET               ;RESETB DRIVE
7272  025160  104026                    ERROR   26
7273  025162  104413                    CNT.RESET
7274  025164  013737  001350  025276    MOV     DRIVAD,DRHOLD           ;SAVE DRIVE ADRR
7275  025172  032737  020000  001350    BIT     #20000,DRIVAD           ;SEE IF ODD
7276  025200  001404                    BEQ     1$
7277  025202  042737  020000  001350    BIC     #20000,DRIVAD           ;MAKE EVEN
7278  025210  000403                    BR      2$
7279  025212  052737  020000  001350  1$:   BIS     #20000,DRIVAD           ;MAKE ODD
7280  025220  013777  001350  154112  2$:   MOV     DRIVAD,@RKDA    ;DRIVE ADDR
7281  025226  012777  000011  154076        MOV     #11,@RKCS               ;DRIVE SEEK
7282  025234  104414                    CNT.RDY
7283  025236  013777  025276  154074        MOV     DRHOLD,@RkDA            ;OTHER DRIVE
7284  025244  104414                    CNT.RDY
7285  025246  032777  000100  154052        BIT     #100,@RKDS              ;HEAEDS IN MOTIONN?
7286  025254  001001                    BNE     3$                      ;NO SO RK-05J
7287  025256  005725                    TST     (R5)+                   ;YES RK-05F
7288  025260  013737  025276  001350  3$:   MOV     DRHOLD,DRIVAD           ;RESTORE ADDR
7289  025266  004737  021500            JSR     PC,DRESET               ;WAIT FOR RESET
7290  025272  104026                    ERROR   26
7291  025274  000205                    RTS     R5
7292  025276  000000            DRHOLD: 0
7293  025300  005037  001350    SIZEF:  CLR     DRIVAD          ;START AT DR0
7294  025304  012700  001414            MOV     #DRIV0,R0               ;TABLE OF AVAIL DRIVES
7295  025310  005710            4$:     TST     (R0)                    ;THIS DRIVE HERE?
7296  025312  001413                    BEQ     2$                      ;NO
7297  025314  005760  000002            TST     2(R0)                   ;COMPLEMENT HERE?
7298  025320  001410                    BEQ     2$                      ;NO
7299  025322  004537  025154            JSR     R5,FCHECK               ;SEE IF F MODEL
7300  025326  000405                    BR      2$                      ;J MODEL
7301  025330  052710  100000            BIS     #100000,(R0)            ;SET SIGN FOR F
```

```
7302  025334  052760  100000  000002          BIS     #100000,2(R0)           ;BOTH DRIVES
7303  025342  005720                  2$:     TST     (R0)+
7304  025344  005720                          TST     (R0)+                   ;NEXT PAIR OF DRIVES
7305  025346  062737  040000  001350          ADD     #40000,DRIVAD           ;NEXT ACTUL ADDR
7306  025354  022700  001433                  CMP     #DRV7+1,R0              ;CHECKED ALL?
7307  025360  003353                          BGT     4$                      ;NOT YET
7308  025362  000207                          RTS     PC
7309
7310                                          ;ERROR MESSAGES
7311
7312                                          .SBTTL  ERROR MESSAGES
7313
7314  025364  045522  041527  042440  EM11:   .ASCIZ  /RKWC EROR/
7315  025372  047522  000122
7316
7317
7318  025376  044523  020116  051511  EM12:   .ASCIZ  /SIN IS SET/
7319  025404  051440  052105     000
7320
7321  025411     122  041113  020101  EM13:   .ASCIZ  /RKBA EROR/
7322  025416  051105  051117     000
7323
7324  025423     122  042113  020101  EM16:   .ASCIZ  /RKDA WRONG AFTER 'SSE'/
7325  025430  051127  047117  020107
7326  025436  043101  042524  020122
7327  025444  051447  042523  000047
7328
7329  025452  045522  051504  042440  EM21:   .ASCIZ  /RKDS EROR/
7330  025460  047522  000122
7331

7332  025464  050104  020114  042523  EM30:   .ASCIZ  /DPL SET/
7333  025472  000124
7334
7335  025474  051104  020125  042523  EM31:   .ASCIZ  /DRU SET/
7336  025502  000124
7337
7338  025504  045522  032460  041040  EM32:   .ASCIZ  /RKOS BIT NOT SET/
7339  025512  052111  047040  052117
7340  025520  051440  052105     000
7341
7342  025525     104  054522  041040  EM33:   .ASCIZ  /DRY BIT NOT SET/
7343  025532  052111  047040  052117
7344  025540  051440  052105     000
7345
7346  025545     123  045517  042040  EM34:   .ASCIZ  /SOK DIDN'T SET/
7347  025552  042111  023516  020124
7348  025560  042523  000124
7349
7350  025564  042523  026503  047103  EM35:   .ASCIZ  /SEC-CNTR DIDN'T COUNT TO 0/
7351  025572  051124  042040  042111
7352  025600  023516  020124  047503
7353  025606  047125  020124  047524
7354  025614  030040     000
7355
7356  025617     123  041505  041455  EM36:   .ASCIZ  /SEC-CNTR DIDN'T INCRMNT/
7357  025624  052116  020122  044504
```

```
7358  025632  047104  052047  044440
7359  025640  041516  046522  052116
7360  025646     000
7361
7362  025647     123  041505  041455  EM37:   .ASCIZ  /SEC-COUNTR INCRMENTED WRONG/
7363  025654  052517  052116  020122
7364  025662  047111  051103  042515
7365  025670  052116  042105  053440
7366  025676  047522  043516     000
7367
7368  025703     104  042111  023516  EM40:   .ASCIZ  /DIDN'T GET SC=SA FOR THIS SECTR/
7369  025710  020124  042507  020124
7370  025716  041523  051475  020101
7371  025724  047506  020122  044124
7372  025732  051511  051440  041505
7373  025740  051124     000
7374
7375  025743     105  047522  026522  EM41:   .ASCIZ  "ERDR-R/W/S RDY SHOULD BE SET"
7376  025750  027522  027527  020123
7377  025756  042122  020131  044123
7378  025764  052517  042114  041040
7379  025772  020105  042523  000124
7380
7381  026000  047125  054105  042520  EM43:   .ASCIZ  /UNEXPECTED RK11 INTERRUPT/
7382  026006  052103  042105  051040
7383  026014  030513  020061  047111
7384  026022  042524  051122  050125
7385  026030  000124
7386
7387  026032  047103  051124  020114  EM44:   .ASCIZ  /CNTRL RDY DIDN'T SET AFTER SEEK OR DR RESET/
7388  026040  042122  020131  044504
7389  026046  047104  052047  051440
7390  026054  052105  040440  052106
7391  026062  051105  051440  042505
7392  026070  020113  051117  042040
7393  026076  020122  042522  042523
7394  026104  000124
7395
7396  026106  051105  020122  051117  EM45:   .ASCIZ  /ERR OR HE BIT SET ON SEEK OR DR RESET
7397  026114  044040  020105  044502
7398  026122  020124  042523  020124
7399  026130  047117  051440  042505
7400  026136  020113  051117  042040
7401  026144  020122  042522  042523
7402  026152  000124
7403
7404  026154  045522  051105  041040  EM46:   .ASCIZ  /RKER BIT, ON SEEK OR DR RESET/
7405  026162  052111  020054  047117
7406  026170  051440  042505  020113
7407  026176  051117  042040  020122
7408  026204  042522  042523  000124
7409
7410  026212  045522  051503  041440  EM47:   .ASCIZ  /RKCS CHNGD AFTR FUNCTION WAS DONE/
7411  026220  047110  042107  040440
7412  026226  052106  020122  052506
7413  026234  041516  044524  047117
```

```
7414  026242  053440  051501  042040
7415  026250  047117  000105
7416
7417  026254  027522  027527  020123  EM50:   .ASCIZ  "R/W/S RDY DIDN'T CLEAR"
7418  026262  042122  020131  044504
7419  026270  047104  052047  041440
7420  026276  042514  051101     000
7421
7422  026303     122  053457  051457  EM51:   .ASCIZ  "R/W/S RDY DIDN'T SET AFTR SEEK OR DR RESET"
7423  026310  051040  054504  042040
7424  026316  042111  023516  020124
7425  026324  042523  020124  043101
7426  026332  051124  051440  042505
7427  026340  020113  051117  042040
7428  026346  020122  042522  042523
7429  026354  000124
7430
7431  026356  045522  040504  041440  EM52:   .ASCIZ  /RKDA CHNGD AFTR SEEK/
7432  026364  047110  042107  040440
7433  026372  052106  020122  042523
7434  026400  045505     000
7435
7436  026403     103  052116  046122  EM53:   .ASCIZ   CNTRL RDY DIDN'T CLR AS GO WAS SET/
7437  026410  051040  054504  042040
7438  026416  042111  023516  020124
7439  026424  046103  020122  051501
7440  026432  043440  020117  040527
7441  026440  020123  042523  000124
7442
7443  026446  047103  051124  020114  EM54:   .ASCIZ  "CNTRL RDY DIDN'T SET ON WRT/FMT STARTING FROM <DSK-ADRES>"
7444  026454  042122  020131  044504
7445  026462  047104  052047  051440
7446  026470  052105  047440  020116
7447  026476  051127  027524  046506
7448  026504  020124  052123  051101
7449  026512  044524  043516  043040
7450  026520  047522  020115  042074
7451  026526  045523  040455  051104
7452  026534  051505  000076
7453
7454  026540  042510  047440  020122  EM55:   .ASCIZ  "HE OR ERR ON WRT/FMT STARTING FROM <DSK-ADRES>"
7455  026546  051105  020122  047117
7456  026554  053440  052122  043057
7457  026562  052115  051440  040524
7458  026570  052122  047111  020107
7459  026576  051106  046517  036040
7460  026604  051504  026513  042101
7461  026612  042522  037123     000
7462
7463  026617     122  042113  020101  EM56:   .ASCIZ  <RKDA INCRMNTD WRONG ON WRT-FMT>
7464  026624  047111  051103  047115
7465  026632  042124  053440  047522
7466  026640  043516  047440  020116
7467  026646  051127  026524  046506
7468  026654  000124
7469
```

```
7470  026656  045522  041527  042040  EM57:   .ASCIZ  /RKWC DIDN'T OVRFLO ON WRT FMT/
7471  026664  042111  023516  020124
7472  026672  053117  043122  047514
7473  026700  047440  020116  051127
7474  026706  020124  046506  000124
7475
7476  026714  045522  040502  044440  EM60:   .ASCIZ  /RKBA INCRMNTD WRONG ON WRT FMT/
7477  026722  041516  046522  052116
7478  026730  020104  051127  047117
7479  026736  020107  047117  053440
7480  026744  052122  043040  052115
7481  026752     000
7482
7483  026753     122  042513  020122  EM61:   .ASCIZ  /RKER SET,ON WRT OR RD OR FMT/
7484  026760  042523  026124  047117
7485  026766  053440  052122  047440
7486  026774  020122  042122  047440
7487  027002  020122  046506  000124
7488
7489  027010  045522  041104  042440  EM62:   .ASCIZ  /RKDB EROR/
7490  027016  047522  000122
7491
```

```
7492  027022  045522  040504  044440  EM63:   .ASCIZ  /RKDA INCRMNTD WRONG ON RD OR RD FMT/
7493  027030  041516  046522  052116
7494  027036  020104  051127  047117
7495  027044  020107  047117  051040
7496  027052  020104  051117  051040
7497  027060  020104  046506  000124
7498
7499  027066  045522  041527  042040  EM64:   .ASCIZ  /RKWC DIDN'T OVRFLO ON RD OR RD FMT/
7500  027074  042111  023516  020124
7501  027102  053117  043122  047514
7502  027110  047440  020116  042122
7503  027116  047440  020122  042122
7504  027124  043040  052115     000
7505
7506  027131     122  041113  020101  EM65:   .ASCIZ  /RKBA INCRMNTD WRONG ON RD OR RD FMT/
7507  027136  047111  051103  047115
7508  027144  042124  053440  047522
7509  027152  043516  047440  020116
7510  027160  042122  047440  020122
7511  027166  042122  043040  052115
7512  027174     000
7513
7514  027175     111  041516  051117  EM66:   .ASCIZ  /INCORRECT HEADER FROM 'SECTOR'/
7515  027202  042522  052103  044040
7516  027210  040505  042504  020122
7517  027216  051106  046517  023440
7518  027224  042523  052103  051117
7519  027232  000047
7520
7521  027234  040504  040524  042440  EM67:   .ASCIZ  /DATA ERROR/
7522  027242  051122  051117     000
7523
7524  027247     103  052116  046122  EM70:   .ASCIZ  "CNTRL RDY DIDN'T SET ON RD/FMT STARTING FROM <DSk-ADRES>"
7525  027254  051040  054504  042040
7526  027262  042111  023516  020124
7527  027270  042523  020124  047117
7528  027276  051040  027504  046506
7529  027304  020124  052123  051101
7530  027312  044524  043516  043040
7531  027320  047522  020115  042074
7532  027326  045523  040455  051104
7533  027334  051505  000076
7534
7535  027340  042510  047440  020122  EM71:   .ASCIZ  "HE OR ERR ON RD/FMT STARTING FROM <DSK-ADRES "
7536  027346  051105  020122  047117
7537  027354  051040  027504  046506
7538  027362  020124  052123  051101
7539  027370  044524  043516  043040
7540  027376  047522  020115  042074
7541  027404  045523  040455  051104
7542  027412  051505  000076
7543
7544  027416  051127  047117  020107  EM72:   .ASCIZ  /WRONG DRIVE ID IN RKDS AFTER SEEK/
7545  027424  051104  053111  020105
7546  027432  042111  044440  020116
7547  027440  045522  051504  040440
```

```
7548  027446  052106  051105  051440
7549  027454  042505  000113
7550
7551  027460  051110  053504  042522  EM73:   .ASCIZ  /HRDWRE POLL-DRV ID BITS(13-15) SHLDBE CLR/
7552  027466  050040  046117  026514
7553  027474  051104  020126  042111
7554  027502  041040  052111  024123
7555  027510  031461  030455  024465
7556  027516  051440  046110  041104
7557  027524  020105  046103  000122
7558
7559  027532  051110  053504  042522  EM74:   .ASCIZ  /HRDWRE POLL-INTRUPTING DRIV # NOT PRSNT/
7560  027540  050040  046117  026514
7561  027546  047111  051124  050125
7562  027554  044524  043516  042040
7563  027562  044522  020126  020043
7564  027570  047516  020124  051120
7565  027576  047123  000124
7566
7567  027602  051104  053111  021440  EM75:   .ASCIZ  /DRIV # DIDN'T INTRUPT AFTER HRDWRE POLL/
7568  027610  042040  042111  023516
7569  027616  020124  047111  051124
7570  027624  050125  020124  043101
7571  027632  042524  020122  051110
7572  027640  053504  042522  050040
7573  027646  046117  000114
7574
7575  027652  041523  020120  044504  EM76:   .ASCIZ  /SCP DIDN'T SET AFTER SEEK WAS DONE/
7576  027660  047104  052047  051440
7577  027666  052105  040440  052106
7578  027674  051105  051440  042505
7579  027702  020113  040527  020123
7580  027710  047504  042516     000
7581
7582  027715     122  042113  020101  EM77:   .ASCIZ  /RKDA CHANGD AFTER DRIV RESET/
7583  027722  044103  047101  042107
7584  027730  040440  052106  051105
7585  027736  042040  044522  020126
7586  027744  042522  042523  000124
7587
7588  027752  040504  040524  042440  EM100:  .ASCIZ  /DATA EROR AT WORD#/
7589  027760  047522  020122  052101
7590  027766  053440  051117  021504
7591  027774     000
7592
7593  027775     103  052116  046122  EM101:  .ASCIZ  /CNTRL RDY DIDN'T SET AFTER RD CHK/
7594  030002  051040  054504  042040
7595  030010  042111  023516  020124
7596  030016  042523  020124  043101
7597  030024  042524  020122  042122
7598  030032  041440  045510     000
7599
7600  030037     105  051122  047440  EM102:  .ASCIZ  /ERR OR HE ON RD CHK/
7601  030044  020122  042510  047440
7602  030052  020116  042122  041440
7603  030060  045510     000
```

```
7604
7605  030063     103  042523  047440  EM103:  .ASCIZ  /CSE ON RD CHK/
7606  030070  020116  042122  041440
7607  030076  045510     000
7608
7609  030101     122  053513  020103  EM104:  .ASCIZ  /RKWC DIDN'T OVERFLO ON RD CHK OR WRT CHK/
7610  030106  044504  047104  052047
7611  030114  047440  042526  043122
7612  030122  047514  047440  020116
7613  030130  042122  041440  045510
7614  030136  047440  020122  051127
7615  030144  020124  044103  000113
7616
7617  030152  045522  040504  044440  EM105:  .ASCIZ   RKDA INCRMNTD WRONG ON RD CHK/
7618  030160  041516  046522  052116
7619  030166  020104  051127  047117
7620  030174  020107  047117  051040
7621  030202  020104  044103  000113
7622
7623  030210  045522  040502  041440  EM106:  .ASCIZ  /RKBA CHANGD AFTER RD CHK/
7624  030216  040510  043516  020104
7625  030224  043101  042524  020122
7626  030232  042122  041440  045510
7627  030240     000
7628
7629  030241     115  046505  051117  EM107:  .ASCIZ  /MEMORY WORD CHANGED AFTER RD CHK/
7630  030246  020131  047527  042122
7631  030254  041440  040510  043516
7632  030262  042105  040440  052106
7633  030270  051105  051040  020104
7634  030276  044103  000113
7635
7636  030302  047103  051124  020114  EM110:  .ASCIZ  /CNTRL RDY DIDN'T SET AFTER WRT CHK/
7637  030310  042122  020131  044504
7638  030316  047104  052047  051440
7639  030324  052105  040440  052106
7640  030332  051105  053440  052122
7641  030340  041440  045510     000
7642
7643  030345     110  020105  051117  EM111:  .ASCIZ  /HE OR ERR ON WRT CHK/
7644  030352  042440  051122  047440
7645  030360  020116  051127  020124
7646  030366  044103  000113
7647
7648  030372  051127  052111  020105  EM112:  .ASCIZ  /WRITE CHECK EROR/
7649  030400  044103  041505  020113
7650  030406  051105  051117     000
7651
7652  030413     122  042113  020101  EM113:  .ASCIZ  /RKDA INCRMNTD WRONG ON WRT CHK/
7653  030420  047111  051103  047115
7654  030426  042124  053440  047522
7655  030434  043516  047440  020116
7656  030442  051127  020124  044103
7657  030450  000113
7658
7659  030452  045522  040502  044440  EM114:  .ASCIZ  /RKBA INCRMNTD WRONG ON WRT CHK
```

```
7660  030460  041516  046522  052116
7661  030466  020104  051127  047117
7662  030474  020107  047117  053440
7663  030502  052122  041440  045510
7664  030510     000
7665
7666  030511     122  041113  020101  EM115:  .ASCIZ  /RKBA INCRMNTD. WITH IBA SET/
7667  030516  047111  051103  047115
7668  030524  042124  020054  044527
7669  030532  044124  044440  040502
7670  030540  051440  052105     000
7671
7672  030545     127  047522  043516  EM116:  .ASCIZ  /WRONG MEMORY LOCATION CHANGED WITH IBA SET/
7673  030552  046440  046505  051117
7674  030560  020131  047514  040503
7675  030566  044524  047117  041440
7676  030574  040510  043516  042105
7677  030602  053440  052111  020110
7678  030610  041111  020101  042523
7679  030616  000124
7680
7681  030620  045522  030461  042040  EM117:  .ASCIZ  /RK11 DIDN'T INTRUPT WHEN IDE WAS SET/
7682  030626  042111  023516  020124
7683  030634  047111  051124  050125
7684  030642  020124  044127  047105
7685  030650  044440  042504  053440
7686  030656  051501  051440  052105
7687  030664     000
7688
7689  030665     122  030513  020061  EM120:  .ASCIZ  /RK11 DIDN'T INTRUPT AFTER SK WAS INITIATED/
7690  030672  044504  047104  052047
7691  030700  044440  052116  052522
7692  030706  052120  040440  052106
7693  030714  051105  051440  020113
7694  030722  040527  020123  047111
7695  030730  052111  040511  042524
7696  030736  000104
7697
7698  030740  041523  020120  042523  EM121:  .ASCIZ  /SCP SET BEFORE SEEK COMPLETED/
7699  030746  020124  042502  047506
7700  030754  042522  051440  042505
7701  030762  020113  047503  050115
7702  030770  042514  042524  000104
7703
7704  030776  045522  030461  042040  EM122:  .ASCIZ  /RK11 DIDN'T INTRUPT AFTER SK COMPLETED/
7705  031004  042111  023516  020124
7706  031012  047111  051124  050125
7707  031020  020124  043101  042524
7708  031026  020122  045523  041440
7709  031034  046517  046120  052105
7710  031042  042105     000
7711
7712  031045     103  052116  046122  EM123:  .ASCIZ  /CNTRL RESET DIDN'T CLEAR 'SCP'/
7713  031052  051040  051505  052105
7714  031060  042040  042111  023516
7715  031066  020124  046103  040505
```

```
7716  031074  020122  051447  050103
7717  031102  000047
7718
7719  031104  045522  030461  042040  EM124:  .ASCIZ  /Rk11 DIDN'T INTRUPT AFTER RD DONE/
7720  031112  042111  023516  020124
7721  031120  047111  051124  050125
7722  031126  020124  043101  042524
7723  031134  020122  042122  042040
7724  031142  047117  000105
7725
7726  031146  047103  051124  020114  EM125:  .ASCIZ  /CNTRL RESET DIDN'T CLR REGISTR/
7727  031154  042522  042523  020124
7728  031162  044504  047104  052047
7729  031170  041440  051114  051040
7730  031176  043505  051511  051124
7731  031204     000
7732
7733  031205     122  030513  020061  EM126:  .ASCIZ  /RK11 DIDN'T INTRUPT AT CPU LEVEL/
7734  031212  044504  047104  052047
7735  031220  044440  052116  052522
7736  031226  052120  040440  020124
7737  031234  050103  020125  042514
7738  031242  042526  000114
7739
7740  031246  045522  030461  044440  EM127:  .ASCIZ  /RK11 INTRUPTED AT WRONG CPU LEVEL/
7741  031254  052116  052522  052120
7742  031262  042105  040440  020124
7743  031270  051127  047117  020107
7744  031276  050103  020125  042514
7745  031304  042526  000114
7746
7747  031310  042447  051122  041040  EM130:  .ASCIZ  /'ERR BIT' DID   SET IN RKER/
7748  031316  052111  020047  044504
7749  031324  047104  052047  051440
7750  031332  052105  044440  020116
7751  031340  045522  051105     000
7752
7753  031345     110  020105  051117  EM131:  .ASCIZ  /HE OR ERR DIDN'T SET/
7754  031352  042440  051123  042040
7755  031360  042111  023516  020124
7756  031366  042523  000124
7757
7758  031372  045522  051105  042440  EM132:  .ASCIZ  /RKER EROR/
7759  031400  047522  000122
7760
7761  031404  054116  020103  044502  EM133:  .ASCIZ  /NXC BIT DIDN'T SET/
7762  031412  020124  044504  047104
7763  031420  052047  051440  052105
7764  031426     000
7765
7766  031427     122  030513  020061  EM134:  .ASCIZ  /Rk11 DIDN'T INTRUPT ON SOFT EROR/
7767  031434  044504  047104  052047
7768  031442  044440  052116  052522
7769  031450  052120  047440  020116
7770  031456  047523  052106  042440
7771  031464  047522  000122
```

```
7772
7773  031470  042515  020130  044502  EM135:  .ASCIZ  /MEX BITS INCRMNTD WRONG-RKCS/
7774  031476  051524  044440  041516
7775  031504  046522  052116  020104
7776  031512  051127  047117  026507
7777  031520  045522  051503     000
7778
7779  031525     127  051520  047040  EM137:  .ASCIZ  /WPS NOT CLEAR/
7780  031532  052117  041440  042514
7781  031540  051101     000
7782
7783  031543     104  052101  020101  EM140:  .ASCIZ  /DATA EROR ON TRANSFER FROM DISK TO TTY/
7784  031550  051105  051117  047440
7785  031556  020116  051124  047101
7786  031564  043123  051105  043040
7787  031572  047522  020115  044504
7788  031600  045523  052040  020117
7789  031606  052124  000131
7790
7791  031612  042047  044522  020126  EM141:  .ASCIZ  /'DRIV #' PRESENT, BUT NOT INDICATED/
7792  031620  023443  050040  042522
7793  031626  042523  052116  020054
7794  031634  052502  020124  047516
7795  031642  020124  047111  044504
7796  031650  040503  042524  000104
7797  031656  047040  020117  052502  EM142:  .ASCIZ  / NO BUSY ON OTHER HALF OF RK-05F/
7798  031664  054523  047440  020116
7799  031672  052117  042510  020122
7800  031700  040510  043114  047440
7801  031706  020106  045522  030055
7802  031714  043065     000
7803
7804
7805
7806
7807
7808          031720                          .EVEN
7809
781J                                          .SBTTL  ERROR DATA POINTERS
7811
7812  031720  001116  001162  000000  DT1:    .WORD   $ERRPC,$REG0,0
7813
7814  031726  001116  001162  001164  DT2:    .WORD   $ERRPC,$REG0,$REG1,0
7815  031734  000000
7816
7817  031736  001116  001162  001164  DT20:   .WORD   $ERRPC,$REG0,$REG1,$REG2,$REG3,0
7818  031744  001166  001170  000000
7819
7820  031752  001116  000000          DT21:   .WORD   $ERRPC,0
7821
7822  031756  001116  001162  001164  DT26:   .WORD   $ERRPC,$REG0,$REG1,$REG2,0
7823  031764  001166  000000
7824
7825  031770  001116  001162  001164  DT54:   .WORD   $ERRPC,$REG0,$REG1,$REG2,$REG3,$REG4,$REG5,$REG6,$REG7,0
7826  031776  001166  001170  001172
7827  032004  001174  001176  001200
```

```
7828  032012  000000
7829
7830
7831
7832
7833                                          .SBTTL  ERROR HEADERS
7834
7835  032014  020040  041520  020040  DH2:    .ASCIZ  /  PC   REGADD    RECVD/
7836  032022  051040  043505  042101
7837  032030  020104  020040  051040
7838  032036  041505  042126     000
7839
7840  032043     040  050040  020103  DH4:    .ASCIZ  /  PC    EXPCT   RECVD/
7841  032050  020040  042440  050130
7842  032056  052103  020040  051040
7843  032064  041505  042126     000
7844
7845  032071     040  050040  020103  DH5:    .ASCIZ  /  PC    RECVD/
7846  032076  020040  051040  041505
7847  032104  042126     000
7848
7849  032107     040  050040  020103  DH14:   .ASCIZ  /  PC    RKCS    RKER    RKWC/
7850  032114  020040  051040  041513
7851  032122  020123  020040  051040
7852  032130  042513  020122  020040
7853  032136  051040  053513  000103
7854
7855  032144  020040  041520     000  DH21:   .ASCIZ  /  PC/
7856
7857  032151     040  050040  020103  DH30:   .ASCIZ  /  PC     RKCS    RKER    RKDS/
7858  032156  020040  020040  045522
7859  032164  051503  020040  020040
7860  032172  045522  051105  020040
7861  032200  020040  045522  051504
7862  032206     000
7863
7864  032207     040  050040  020103  DH34:   .ASCIZ  /  PC     RKDS/
7865  032214  020040  020040  045522
7866  032222  051504     000
7867
7868  032225     040  050040  020103  DH35:   .ASCIZ  /  PC   SEC-CNTR/
7869  032232  020040  042523  026503
7870  032240  047103  051124     000
7871
7872  032245     040  050040  020103  DH36:   .ASCIZ  /  PC     PRSNT  NXT-CNT/
7873  032252  020040  020040  051120
7874  032260  047123  020124  047040
7875  032266  052130  041455  052116
7876  032274     000
7877
7878  032275     040  050040  020103  DH40:   .ASCIZ  /  PC    SECTOR   RKDS/
7879  032302  020040  051440  041505
7880  032310  047524  020122  020040
7881  032316  045522  051504     000
7882
7883  032323     040  050040  020103  DH44:   .ASCIZ  /  PC    RKCS    RKER    RKDS    RKDA/
```

```
 7884  032330  020040  051040  041513
 7885  032336  020123  020040  051040
 7886  032344  042513  020122  020040
 7887  032352  051040  042113  020123
 7888  032360  020040  051040  042113
 7889  032366  000101
 7890
 7891  032370  020040  041520  020040  DH54:   .ASCIZ  / PC    RKCS    RKER    RKDS    RKDA   DRV#....CYL.<DSK-ADRS>.SUR..SEC/
 7892  032376  020040  045522  051503
 7893  032404  020040  020040  045522
 7894  032412  051105  020040  020040
 7895  032420  045522  051504  020040
 7896  032426  020040  045522  040504
 7897  032434  020040  042040  053122
 7898  032442  027043  027056  041456
 7899  032450  046131  036056  051504
 7900  032456  026513  042101  051522
 7901  032464  027076  052523  027122
 7902  032472  051456  041505     000
 7903
 7904  032477     040  041520  020040  DH56:   .ASCIZ  / PC  EXPC: DR#   CYL     SUR     SEC  RECV: DR#   CYL     SUR     SEC/
 7905  032504  054105  041520  020072
 7906  032512  051104  020043  020040
 7907  032520  054503  020114  020040
 7908  032526  020040  052523  020122
 7909  032534  020040  051440  041505
 7910  032542  020040  042522  053103
 7911  032550  020072  051104  020043
 7912  032556  020040  054503  020114
 7913  032564  020040  020040  052523
 7914  032572  020122  020040  020040
 7915  032600  042523  000103
 7916
 7917  032604  020040  041520  020040  DH64:   .ASCIZ  / PC    RKWC   RKDA/
 7918  032612  020040  045522  041527
 7919  032620  020040  051040  042113
 7920  032626  000101
 7921
 7922  032630  020040  041520  020040  DH66:   .ASCIZ  / PC    SECTR   EXPCT   RECVD
 7923  032636  020040  042523  052103
 7924  032644  020122  020040  054105
 7925  032652  041520  020124  020040
 7926  032660  042522  053103  000104
 7927
 7928  032666  020040  041520  020040  DH67:   .ASCIZ  / PC    EXPCT   RECVD  DSK-ADRS/
 7929  032674  020040  054105  041520
 7930  032702  020124  020040  042522
 7931  032710  053103  020104  042040
 7932  032716  045523  040455  051104
 7933  032724  000123
 7934
 7935  032726  020040  041520  020040  DH74:   .ASCIZ  / PC      DRIV #/
 7936  032734  020040  020040  051104
 7937  032742  053111  021440     000
 7938
 7939  032747     040  050040  020103  DH100:  .ASCIZ  / PC  WORD #   EXPCT   RECVD/
```

```
7940  032754  053440  051117  020104
7941  032762  020043  020040  054105
7942  032770  041520  020124  020040
7943  032776  042522  053103  000104
7944
7945  033004  020040  041520  020040  DH103:  .ASCIZ  /  PC   RKER/
7946  033012  051040  042513  000122
7947
7948  033020  020040  041520  020040  DH104:  .ASCIZ  /  PC   RECVD   RKCS/
7949  033026  051040  041505  042126
7950  033034  020040  051040  041513
7951  033042  000123
7952
7953  033044  020040  041520  020040  DH107:  .ASCIZ  /  PC     LOC    EXPCT   RECVD/
7954  033052  020040  046040  041517
7955  033060  020040  020040  054105
7956  033066  041520  020124  020040
7957  033074  042522  053103  000104
7958
7959  033102  020040  041520  020040  DH117:  .ASCIZ  /  PC   RKCS/
7960  033110  051040  041513  000123
7961
7962  033116  020040  041520  020040  DH126:  .ASCIZ  /  PC    LEVEL    RKCS/
7963  033124  020040  042514  042526
7964  033132  020114  020040  051040
7965  033140  041513  000123
7966
7967  033144  020040  041520  020040  DH130:  .ASCIZ  /  PC     RKCS    RKER  ERR BIT.
7968  033152  020040  051040  041513
7969  033160  020123  020040  051040
7970  033166  042513  020122  042440
7971  033174  051122  041040  052111
7972  033202     000
7973
7974  033203     040  050040  020103  DH131:  .ASCIZ  /  PC     RKCS    RKER.
7975  033210  020040  020040  045522
7976  033216  051503  020040  020040
7977  033224  045522  051105     000
7978
7979  033231     040  050040  020103  DH133:  .ASCIZ  /  PC     RKCS    RKER    RKDA.
7980  033236  020040  020040  045522
7981  033244  051503  020040  020040
7982  033252  045522  051105  020040
7983  033260  020040  045522  040504
7984  033266     000
7985
7986  033267     040  050040  020103  DH140:  .ASCIZ  /  PC    EXPCT   RECVD   RKBA    RKCS.
7987  033274  020040  042440  050130
7988  033302  052103  020040  051040
7989  033310  041505  042126  020040
7990  033316  020040  045522  040502
7991  033324  020040  020040  045522
7992  033332  051503     000
7993
7994
7995          033336                  .EVEN
```

```
                                        ;DATA BUFFER
                                        
033336  000400                  OUTBUF: .BLKW   256.    ;THIS 256 WORD BUFFER IS FOR
                                                       ;DATA TRANSFERS FROM AND
                                                       ;TO THE DISK.
                                        
                                        
                                .END
```

```
BADINT  004600
BADTMO  004534
BDAR    021070
BDR0    021052
BDR4    021062
BIT0  = 000001
BIT00 = 000001
BIT01 = 000002
BIT02 = 000004
BIT03 = 000010
BIT04 = 000020
BIT05 = 000040
BIT06 = 000100
BIT07 = 000200
BIT08 = 000400
BIT09 = 001000
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BPTVEC= 000014
BRKDA0= 104415
BRKDA4= 104416
BTEOP   020022
CHE1    021236
CHKCCL  021376
CHKCRD= 104412
CHKDA   021256
CHKDA1  021264
CHKECL  021352
CHKER   021336
CHKHE   021230
CHKHE1  021222
CHKWC   021312
CH.CRD  021754
CKSWR = 104407
CNT.RD= 104414
CNT.RE= 104413
CN.RDY  022026
CN.RST  022010
COUNT   001362
COUNT1  001364
CR    = 000015
CRETRN  021424
CRLF  = 000200
DDISP = 177570
DDPCH   001410
DELAY = 104417
DELA.Y  021660
DH100   032747
DH103   033004
DH104   033020
DH107   033044
DH117   033102
DH126   033116
DH130   033144
DH131   033203
DH133   033231
DH14    032107
DH140   033267
DH2     032014
DH21    032144
DH30    032151
DH34    032207
DH35    032225
DH36    032245
DH4     032043
DH40    032275
DH44    032323
DH5     032071
DH54    032370
DH56    032477
DH64    032604
DH66    032630
DH67    032666
DH74    032726
DISPLA  001142
DISPRE  000174
DRESET  021500
DRHOLD  025276
DRIVAD  001350
DRIVS   001412
DRIV0   001414
DRIV1   001416
DRIV2   001420
DRIV3   001422
DRIV4   001424
DRIV5   001426
DRIV6   001430
DRIV7   001432
DRVDON  001352
DRVPTR  001354
DSWR  = 177570
DT1     031720
DT2     031726
DT20    031736
DT21    031752
DT26    031756
DT54    031770
EFLG1   001370
EMTVEC= 000030
EM100   027752
EM101   027775
EM102   030037
EM103   030063
EM104   030101
EM105   030152
EM106   030210
EM107   030241
EM11    025364
EM110   030302
EM111   030345
EM112   030372
EM113   030413
EM114   030452
EM115   030511
EM116   030545
EM117   030620
EM12    025376
EM120   030665
EM121   030740
EM122   030776
EM123   031045
EM124   031104
EM125   031146
EM126   031205
EM127   031246
EM13    025411
EM130   031310
EM131   031345
EM132   031372
EM133   031404
EM134   031427
EM135   031470
EM137   031525
EM140   031543
EM141   031612
EM142   031656
EM16    025423
EM21    025452
EM30    025464
EM31    025474
EM32    025504
EM33    025525
EM34    025545
EM35    025564
EM36    025617
EM37    025647
EM40    025703
EM41    025743
EM43    026000
EM44    026032
EM45    026106
EM46    026154
EM47    026212
EM50    026254
EM51    026303
EM52    026356
EM53    026403
EM54    026446
EM55    026540
EM56    026617
EM57    026656
EM60    026714
EM61    026753
EM62    027010
EM63    027022
EM64    027066
EM65    027131
EM66    027175
EM67    027234
EM70    027247
EM71    027340
EM72    027416
EM73    027460
EM74    027532
EM75    027602
EM76    027652
EM77    027715
ERRVEC= 000004
FCHECK  025154
FFLAG   001404
FTITLE  001346
GTSWR = 104406
GT2RG   021004
GT3RG   020776
GT4RG   020770
HT    = 000011
INDX1   001356
INDX2   001360
IOTVEC= 000020
LF    = 000012
MSG1    001216
MSG2    001236
MSG3    001245
MSG4    001272
MSG5    001303
MSG6    001315
NUDRV   005040
ODDEVN  001406
OUTBUF  033336
PFSTRT  004702
PHYDRV  001436
PIRQ  = 177772
PIRQVE= 000240
PR0   = 000000
PR1   = 000040
PR2   = 000100
PR3   = 000140
PR4   = 000200
PR5   = 000240
PR6   = 000300
PR7   = 000340
PS    = 177775
PSW   = 177776
PWRVEC= 000024
RDCHR = 104410
RDLIN = 104411
RESVEC= 000010
RKBA    001336
RKCS    001332
RKDA    001340
RKDB    001342
RKDS    001326
RKER    001330
RKPRI   001400
RKVEC   001402
RKWC    001334
R6    =%000006
R7    =%000007
SEEK0   001372
SEEK1   001374
SEEK2   001376
SHFTRT  021174
SIMUL   001344
SIZEF   025300
SIZYET  001440
STACK = 001100
START   002636
STARTA  002704
STARTI  003334
STKLMT= 177774
ST2     003736
ST3     004262
ST4     004456
SWR     001140
SWREG   000176
SW0   = 000001
SW00  = 000001
SW01  = 000002
SW02  = 000004
SW03  = 000010
SW04  = 000020
SW05  = 000040
```

```
SW06  = 000100          TST20   007500          TYPE  = 104401          $FILLS  001155          $REG5   001174
SW07  = 000200          TST21   007676          TYPOC = 104402          $GDADR  001120          $REG6   001176
SW08  = 000400          TST22   010150          TYPON = 104404          $GDDAT  001124          $REG7   001200
SW09  = 001000          TST23   010520          TYPOS = 104403          $GET42  020724          $RTNAD  020746
SW1   = 000002          TST24   011000          TS6     020134          $GTSWR  024026          $SAVR6  025142
SW10  = 002000          TST25   011410          TS6FLG  001434          $HD   = 000000          $SCOPE  022134
SW11  = 004000          TST26   011716          WATIME  021736          $ICNT   001104          $SETJP= 000117
SW12  = 010000          TST27   012236          WATINT  021702          $ILLUP  025136          $STUP = 177777
SW13  = 020000          TST3    005236          WAT.IN= 104420          $INTAG  001135          $SVLAD  022342
SW14  = 040000          TST30   012524          $AUTOB  001134          $ITEMB  001114          $SVPC = 000204
SW15  = 100000          TST31   012760          $BDADR  001122          $LF     001214          $SWR  = 165400
SW2   = 000004          TST32   013164          $BDDAT  001126          $LPADR  001106          $SWRMK= 000000
SW3   = 000010          TST33   013432          $CHARC  023300          $LPERR  001110          $TIMES  001206
SW4   = 000020          TST34   013714          $CKSWR  023756          $MNEW   024661          $TKB    001146
SW5   = 000040          TST35   014172          $CMTAG  001100          $MSWR   024650          $TKS    001144
SW6   = 000100          TST36   014324          $CM1  = 000012          $MXCNT  022404          $TN   = 000060
SW7   = 000200          TST37   014650          $CM2  = 000024          $NULL   001154          $TPB    001152
SW8   = 000400          TST4    005350          $CM3  = 000012          $NWTST= 000001          $TPFLG  001157
SW9   = 001000          TST40   015064          $CNTLG  024643          $OCNT   023752          $TPS    001150
TBITVE= 000014          TST41   015270          $CNTLU  024636          $OMODE  023754          $TRAP   024672
TIMER   001366          TST42   015452          $CRLF   001213          $OVER   022370          $TRAP2  024714
TKVEC = 000060          TST43   015560          $DBLK   023520          $PASS   001100          $TRP  = 000022
TPVEC = 000064          TST44   015734          $DOAGN  020744          $POWER  025144          $TRPAD  024726
TRAPVE= 000034          TST45   016072          $DTBL   023510          $PWRAD  025132          $TSTNM  001102
TRTVEC= 000014          TST46   016300          $ENDAD  020734          $PWRDN  024772          $TTYIN  024614
TSTEND  020622          TST47   016452          $ENDCT  020702          $PWRMG  025126          $TYPDS  023304
TSTRWS  021432          TST5    005436          $ENDMG  020753          $PWRUP  025044          $TYPE   023064
TSTSIN  021566          TST50   016630          $ENULL  020750          $QUES   001212          $TYPEC  023234
TST.SI= 104421          TST51   016756          $EOP    020646          $RDCHR  024240          $TYPEX  023302
TST1    004706          TST52   017130          $EOPCT  020674          $RDLIN  024360          $TYPOC  023554
TST10   005742          TST53   017324          $ERFLG  001103          $RDSZ = 000022          $TYPON  023570
TST11   006030          TST54   017616          $ERMAX  001115          $REGAD  001160          $TYPOS  023530
TST12   006104          TST55   017754          $ERROR  022406          $REG0   001162          $XTSTR  022146
TST13   006246          TST56   020000          $ERRPC  001116          $REG1   001164          $$GET4= 000000
TST14   006412          TST57   020050          $ERRTB  001442          $REG10  001202          $DFILL  023753
TST15   006504          TST6    005464          $ERRTY  022730          $REG11  001204          .       = 034336
TST16   006742          TST7    005522          $ERTTL  001112          $REG2   001166
TST17   007170          TYERM   021022          $ESCAP  001210          $REG3   001170
TST2    005040          TYPDS = 104405          $FILLC  001156          $REG4   001172

. ABS.  034336      000

 ERRORS DETECTED:  0

 .DSKM:DZRKKE/SOL=DSKZ:SYSMAC.SML,DSKZ:DZRKKE.P11
 RUN-TIME: 19 27 .5 SECONDS
 RUN-TIME RATIO: 1063/47=22.3
 CORE USED:  34K  (67 PAGES)
```