RK11

BASIC LOGIC TEST 2

MD-11-DZRKK-D

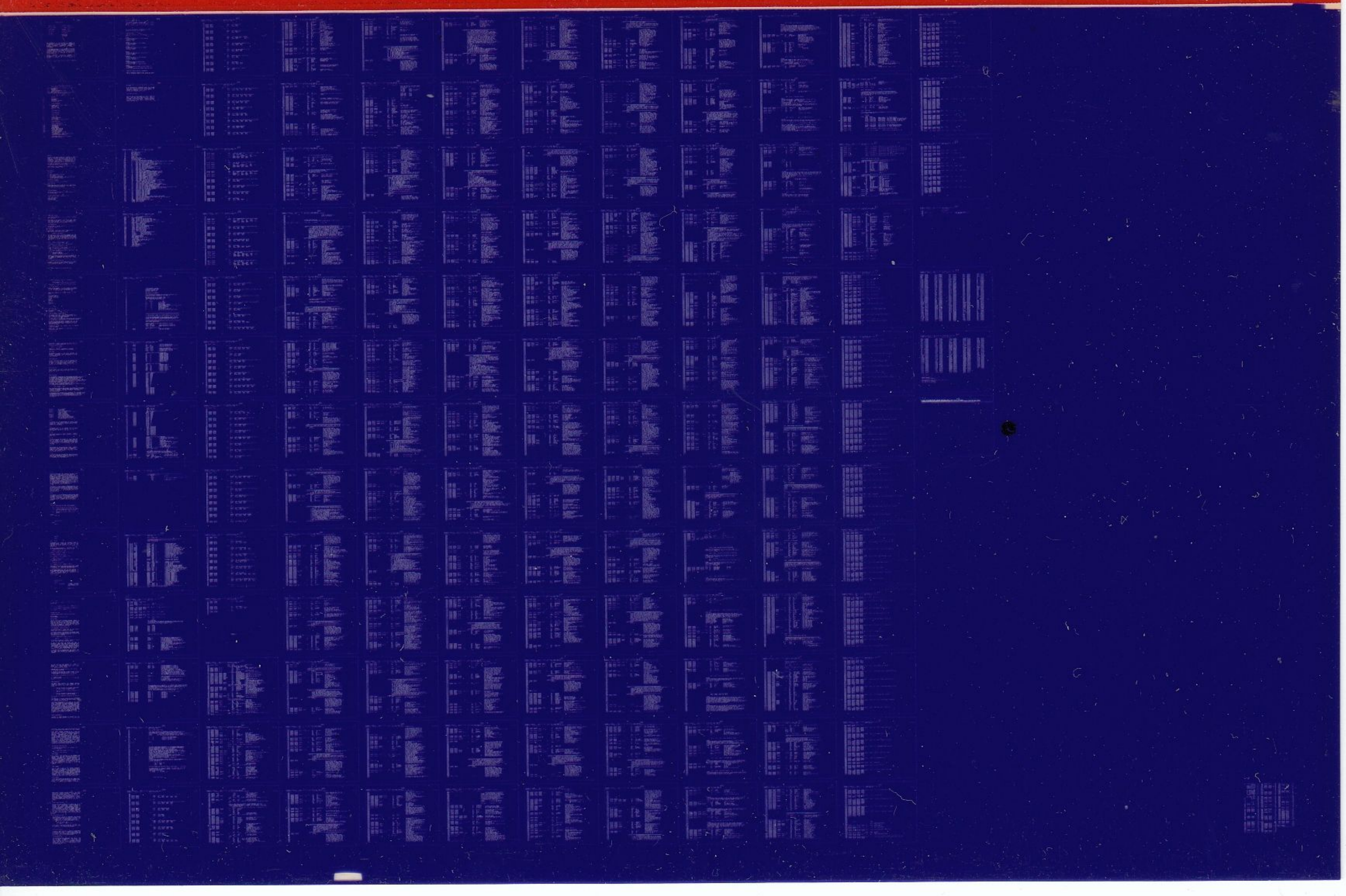
EP-DZRKK-D-DL-B

FICHE 1 OF 1

DEC 1976

digital

MADE IN U.S.A



IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZRKK-D-D
PRODUCT NAME:	RK11 BASIC LOGIC TEST II
DATE CREATED:	DECEMBER, 1976
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 (MC-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 LOGIC TEST II
MAINDEC-11-DZRKK-D
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: SEQ 0005

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 LOGIC TEST II
MAINDEC-11-DZRKA-D
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- TIONVES 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 PREVIOUS '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 S.R<07:00>
SW<06>=1 DROP THE DRIVE AFTER MAXIMUM ALLOWABLE 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 ITERATIONS, 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 PCINTERS 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 1777777 (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.

$TYPE:
TYPE ROUTINE FOR TYPING OUT ASCII STRINGS.
LOCATED AT "$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 RX11 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  MAINDEC-11-DZRKK-D
                        ;*COPYRIGHT (C) 1974,1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY JIM KAPADIA
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
                        ;*
                        ;*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 ***
001100                  STACK=  1100
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE FOR LINE FEED
```

```
                        CR=     15              ;;CODE FOR CARRIAGE RETURN
                        CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
                        PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
                        STKLMT= 177774          ;;STACK LIMIT REGISTER
                        PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
                        DSWR=   177570          ;;HARDWARE SWITCH REGISTER
                        DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
                        R0=     %0              ;;GENERAL REGISTER
                        R1=     %1              ;;GENERAL REGISTER
                        R2=     %2              ;;GENERAL REGISTER
                        R3=     %3              ;;GENERAL REGISTER
                        R4=     %4              ;;GENERAL REGISTER
                        R5=     %5              ;;GENERAL REGISTER
                        R6=     %6              ;;GENERAL REGISTER
                        R7=     %7              ;;GENERAL REGISTER
                        SP=     %6              ;;STACK POINTER
                        PC=     %7              ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
                        PR0=    0               ;;PRIORITY LEVEL 0
                        PR1=    40              ;;PRIORITY LEVEL 1
                        PR2=    100             ;;PRIORITY LEVEL 2
                        PR3=    140             ;;PRIORITY LEVEL 3
                        PR4=    200             ;;PRIORITY LEVEL 4
                        PR5=    240             ;;PRIORITY LEVEL 5
                        PR6=    300             ;;PRIORITY LEVEL 6
                        PR7=    340             ;;PRIORITY LEVEL 7


                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
                        SW15=   100000
                        SW14=   40000
                        SW13=   20000
                        SW12=   10000
                        SW11=   4000
                        SW10=   2000
                        SW09=   1000
                        SW08=   400
                        SW07=   200
                        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)
            100000              BIT15=  100000
            040000              BIT14=  40000
            020000              BIT13=  20000
            010000              BIT12=  10000
            004000              BIT11=  4000
            002000              BIT10=  2000
            001000              BIT09=  1000
            000400              BIT08=  400
            000200              BIT07=  200
            000100              BIT06=  100
            000040              BIT05=  40
            000020              BIT04=  20
            000010              BIT03=  10
            000004              BIT02=  4
            000002              BIT01=  2
            000001              BIT00=  1
                                .EQUIV  BIT09,BIT9
                                .EQUIV  BIT08,BIT8
                                .EQUIV  BIT07,BIT7
                                .EQUIV  BIT06,BIT6
                                .EQUIV  BIT05,BIT5
                                .EQUIV  BIT04,BIT4
                                .EQUIV  BIT03,BIT3
                                .EQUIV  BIT02,BIT2
                                .EQUIV  BIT01,BIT1
                                .EQUIV  BIT00,BIT0

                                ;*BASIC "CPU" TRAP VECTOR ADDRESSES
            000004              ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
            000010              RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
            000014              TBITVEC=14              ;;"T" BIT
            000014              TRTVEC= 14              ;;TRACE TRAP
            000014              BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
            000020              IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
            000024              PWRVEC= 24              ;;POWER FAIL
            000030              EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
            000034              TRAPVEC=34              ;;"TRAP" TRAP
            000060              TKVEC=  60              ;;TTY KEYBOARD VECTOR
            000064              TPVEC=  64              ;;TTY PRINTER VECTOR
            000240              PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
                                .SBTTL  TRAP CATCHER

            000000                      .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
            000174              .=174
000174  000000  DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000  SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                .SBTTL  STARTING ADDRESS(ES)
000200  000137  002636          JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
                                .SBTTL  ACT11 HOOKS
```

```
                                ;;*****************************************************************
                                ;HOOKS REQUIRED BY ACT11
000204                                  $SVPC=.                 ;SAVE PC
000046                                  .=46
000046  020646                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
000052                                  .=52
000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
000204                                  .=$SVPC                 ;; RESTORE PC
```

```
                                .SBTTL  COMMON TAGS

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

                        001100                  .=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          $WR:    .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 REQUIRE.
001156     012          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157     000          $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REGO) 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/
```

```
001224  020105  051120  051505
001232  052116     000
                001236                          .EVEN
001236  005015  047516  042516  MSG2:   .ASCIZ  <15><12>/NONE/
001244     000
001245     015  041412  052116  MSG3:   .ASCIZ  <15><12>/CNT RDY DIDN'T SET/
001252  051040  054504  042040
001260  042111  023516  020124
001266  042523  000124
001272  005015  051104  053111  MSG4:   .ASCIZ  <15><12>/DRIVE /
001300  020105     000
001303     015  040412  046114  MSG5:   .ASCII  <15><12>/ALL DRVS/
001310  042040  053122     123
001315     040  051104  050117  MSG6:   .ASCIZ  / DROPD/<15><12>
001322  006504  000012
                                        .EVEN

                                ;RK11 REGISTERS
                                ;IF FOR ANY REASON THE REGISTER ADDRESSES ARE DIFFERENT FROM THESE
                                ;(GIVEN BELOW), THE CONTENTS OF THE APPROPRIATE POINTERS SHOULD BE
                                ;MODIFIED SO THAT THE CORRECT ADDRESS IS USED.
                                ;
                                        .EVEN
001326  177400                  RKDS:   177400
001330  177402                  RKER:   177402
001332  177404                  RKCS:   177404
001334  177406                  RKWC:   177406
001336  177410                  RKBA:   177410
001340  177412                  RKDA:   177412
001342  177416                  RKDB:   177416


                                ;TAGS AND GENERAL DATA AREA
                                ;
                                ;

001344  000000                  SIMUL:  0               ;FLAG TO BE SET TO 1 WHEN ON SIMULATOR
001346  000000                  FTITLE: 0               ;FLAG FOR PRINTING PROGRAM TITLE
001350  000000                  DRIVAD: 0               ;CONTAINS ADDRESS OF THE DRIVE UNDER TEST
001352  000000                  DRVDON: 0               ;CONTAINS THE NUMBER OF DRIVES CHECKED.
                                                        ;IT IS INCREMENTED EACH TIME THE TESTS FOR
                                                        ;A DRIVE IS COMPLETED.
001354  000000                  DRVPTR: 0               ;CONTAINS THE POINTER TO THE DRIVE FLAG (DRIVE0
                                                        ;-DRIVE7) OF THE DRIVE TO BE CHECKED NEXT.
001356  000000                  INDX1:  0               ;GENERAL INDEX FOR KEEPING COUNT
001360  000000                  INDX2:  0               ; GENERAL INDEX
001362  000000                  COUNT:  0               ;GENERAL COUNT REGISTER
001364  000000                  COUNT1: 0               ;COUNT REGISTER USED FOR 'DRESET' SUBROUTINE
001366  000000                  TIMER:  0               ;TIMER REGISTER
001370  000000                  EFLG1:  0               ;SET, TO INDICATE A PARTICULAR
                                                        ;ERROR CONDITION
```

```
                                SEEK0:  100             ;CONTAINS ADDRESS OF CYLINDER 2
                                SEEK1:  1000            ;CONTAINS ADDRESS OF CYLINDER 20
                                SEEK2:  14500           ;CONTAINS ADDRESS OF CYLINDER 312
                                RKPRI:  200             ;CONTAINS THE CPU LEVEL AT WHICH
                                                        ;RK11 NORMALLY INTERRUPTS. THIS WORD
                                                        ;SHOULD BE CHANGED IF RK11 IS DESINGATED
                                                        ;A BR LEVEL OTHER THAN 5. E.G. IF IT IS CHANGED
                                                        ;TO 6, THIS WORD SHOULD BE CHANGED TO 240.
                                RKVEC:  220             ;CONTAINS THE NORMAL VECTOR ADDRESS TO WHICH
                                                        ;RK11 INTERRUPTS. IF THIS IS NOT SO, CHANGE
                                                        ;THIS WORD TO CONTAIN MODIFIED VECTOR ADDRESS.
001404  000000                  FFLAG:  0
001406  000000                  ODDEVN: 0               ;USED TO DETERMINE WHICH OF RK-05F DRIVES ACTIVE
                                                        ;0 IF EVEN DRIVE
                                                        ;-1 IF ODD DRIVE
001410  000000                  DDPCH:  0               ;IF PROGRAM LOADED FROM RK05, CONTAINS
                                                        ;ADDRESS OF DRIVE WITH RKDP PACK
001412  000000                  DRIVS:  0               ;CONTAINS THE NUMBER OF DRIVES PRESENT



                        ;THE FLAGS BELOW (BIT 0) ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE
                        ;IS PRESENT AND IS TO BE TESTED. BIT 12, IF SET, INDICATES THAT THE DRIVE
                        ;WAS DROPPED AFTER MAXIMUM ALLOWABLE NUMBER OF ERRORS OCCURED ON THAT
                        ;DRIVE (SW 6 SET).
                        ;IF MORE THAN 5 ERRORS OCCUR IN THE HARDWARE POLLING TEST (LAST)
                        ;THEN ALL DRIVES ARE DROPPED. BUT BIT 12 IS NOT SET.

001414  000000          DRIV0:  0               ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
001416  000000          DRIV1:  0               ;FOR DRIVE 1
001420  000000          DRIV2:  0               ;FOR DRIVE 2
001422  000000          DRIV3:  0               ;FOR DRIVE 3
001424  000000          DRIV4:  0               ;FOR DRIVE 4
001426  000000          DRIV5:  0               ;FOR DRIVE 5
001430  000000          DRIV6:  0               ;FOR DRIVE 6
001432  000000          DRIV7:  0               ;FOR DRIVE 7

001434  000000          T56FLG: 0
001436  000000          PHYDRV: 0
001440  000000          SIZYET: 0
```

```
                        .SBTTL  ERROR POINTER TABLE

                        ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                        ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                        ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                        ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                        ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                        ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                        ;*      DH              ;;POINTS TO THE DATA HEADER
                        ;*      DT              ;;POINTS TO THE DATA
                        ;*      DF              ;;POINTS TO THE DATA FORMAT

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
```

```
                                ;ITEM   1
                                
001442  025300                          EM12    ;'SIN' IS SET
001444  032225                          DH44    ;PC     RKCS    RKER    RKDS    RKDA
001446  031640                          DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
001450  000000                          0

                                ;ITEM   2

001452  027151                          EM70    ;CNTRL RDY DIDN'T SET ON READ/FMT FROM DISK ADDRESS
001454  032011                          DH14    ;PC     RKCS    RKER    RKWC
001456  031660                          DT26    ;$ERRPC $REG0   $REG1   $REG2
001460  000000                          0

                                ;ITEM   3

001462  025325                          EM16    ;RKDA WRONG AFTER SSE
001464  031745                          DH4     ;PC     EXPCT   RECVD
001466  031630                          DT2     ;$ERRPC $REG0   $REG1
001470  000000                          0

                                ;ITEM   4

001472  025354                          EM21    ;RKDS ERROR
001474  032111                          DH34    ;PC     RKDS
001476  031622                          DT1     ;$ERRPC $REG0
001500  000000                          0

                                ;ITEM   5

001502  025366                          EM30    ;'DPL' BIT SET, CHECK DRIVE POWER
001504  032053                          DH30    ;PC     RKCS    RKER    RKDS
001506  031660                          DT26    ;$ERRPC $REG0   $REG1   $REG2
001510  000000                          0

                                ;ITEM   6

001512  025376                          EM31    ;'DRU' BIT SET, CHECK DRIVE
001514  032053                          DH30    ;PC     RKCS    RKER    RKDS
001516  031660                          DT26    ;$ERRPC $REG0   $REG1   $REG2
001520  000000                          0

                                ;ITEM   7

001522  025406                          EM32    ;'RKDS' BIT NOT SET
001524  032111                          DH34    ;PC     RKDS
001526  031622                          DT1     ;$ERRPC $REG0
001530  000000                          0

                                ;ITEM   10

001532  025427                          EM33    ;'DRY' NOT SET
001534  032225                          DH44    ;PC     RKCS    RKER    RKDS    RKDA
001536  031640                          DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
001540  000000                          0
```

```
                                ;ITEM   11
001542  025447                          EM34    ;'SOK' DID NOT SET
001544  032111                          DH34    ;PC     RKDS
001546  031622                          DT1     ;SERRPC $REGO
001550  000000                          0

                                ;ITEM   12
001552  025466                          EM35    ;'SEC COUNTR' DIDN'T COUNT TO 0
001554  032127                          DH35    ;PC     SEC-CNTR
001556  031622                          DT1     ;SERRPC $REGO
001560  000000                          0

                                ;ITEM   13
001562  025521                          EM36    ;'SEC COUNTR' DIDN'T INCREMENT
001564  032147                          DH36    ;PC     PRSNT-COUNT     NXT-COUNT
001566  031630                          DT2     ;SERRPC $REGO           $REG1
001570  000000                          0

                                ;ITEM   14
001572  025551                          EM37    ;'SECTOR COUNTER' INCREMENTED WRONG
001574  031745                          DH4     ;PC     EXPCTD  RECVD
001576  031630                          DT2     ;SERRPC $REGO   $REG1
001600  000000                          0

                                ;ITEM   15
001602  025605                          EM40    ;DIDN'T GET SC=SA FOR THIS SECTOR
001604  032177                          DH40    ;PC     SECTOR  RKDS
001606  031630                          DT2     ;SERRPC $REGO   $REG1
001610  000000                          0

                                ;ITEM   16
001612  025645                          EM41    ;ERROR-'R/W/S RDY' SHOULD BE SET
001614  032111                          DH34    ;PC     RKDS
001616  031622                          DT1     ;SERRPC $REGO
001620  000000                          0

                                ;ITEM   17
001622  025313                          EM13    ;RKBA ERROR
001624  031745                          DH4     ;PC     EXPCT   RECVD
001626  031630                          DT2     ;SERRPC $REGO   $REG1
001630  000000                          0

                                ;ITEM   20
001632  025702                          EM43    ;UNEXPECTED RK11 INTERRUPT
001634  032046                          DH21    ;PC
001636  031654                          DT21    ;SERRPC
001640  000000                          0
```

```
                                ;ITEM   21

001642  025734                          EM44    ;'CNTRL RDY' DIDN'T SET AFTER SEEK OR DRIVE RESET
001644  032225                          DH44    ;PC     RKCS    RKER    RKDS    RKDA
001646  031640                          DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3.
001650  000000                          0

                                ;ITEM   22

001652  026010                          EM45    ;'ERR' OR 'HE' SET ON SEEK OR DRIVE RESET
001654  032225                          DH44    ;PC     RKCS    RKER    RKDS    RKDA
001656  031640                          DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
001660  000000                          0

                                ;ITEM   23

001662  026056                          EM46    ;RKER BIT, ON SEEK OR DRIVE RESET
001664  032053                          DH30    ;PC     RKCS    RKER    RKDS
001666  031660                          DT26    ;$ERRPC $REG0   $REG1   $REG2
001670  000000                          0

                                ;ITEM   24

001672  026114                          EM47    ;RKCS CHANGED AFTER FUNCTION WAS DONE
001674  031745                          DH4     ;PC     EXPCT   RECVD
001676  031630                          DT2     ;$ERRPC $REG0   $REG1
001700  000000                          0

                                ;ITEM   25

001702  026156                          EM50    ;'R/W/S RDY' DID NOT CLEAR
001704  032053                          DH30    ;PC     RKCS    RKER    RKDS
001706  031660                          DT26    ;$ERRPC $REG0   $REG1   $REG2
001710  000000                          0

                                ;ITEM   26

001712  026205                          EM51    ;'R/W/S RDY' DIDN'T SET AFTER SEEK OR DRIVE RESET
001714  032225                          DH44    ;PC     RKCS    RKER    RKDS    RKDA
001716  031640                          DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
001720  000000                          0

                                ;ITEM   27

001722  026260                          EM52    ;RKDA CHANGED AFTER SEEK
001724  031745                          DH4     ;PC     EXPCTD  REGVD
001726  031630                          DT2     ;$ERRPC $REG0   $REG1
001730  000000                          0

                                ;ITEM   30

001732  026305                          EM53    ;'CNTRL RDY' DIDN'T CLEAR AS GO WAS SET
001734  032053                          DH30    ;PC     RKCS    RKER    RKDS
001736  031660                          DT26    ;$ERRPC $REG0   $REG1   $REG2
001740  000000                          0
```

```
                                ;ITEM   31
                                
001742  026350                  EM54    ;'CNTRL RDY' DIDN'T SET ON DOING WRITE/FMT STARTING
                                        ;FROM <DSK-ADRES>
001744  032272                  DH54    ;PC     RKCS    RKER    RKDS    RKDA
                                        ;DRV#   CYL   <DSK-ADRES>         SUR     SECTR
001746  031672                  DT54    ;$ERRPC $REG0   $REG1   $REG2   $REG3
                                        ;$REG4  $REG5   $REG6   $REG>
001750  000000                  0

                                ;ITEM   32

001752  026442                  EM55    ;'ME' OR 'ERR'  ON WRITE/FMT STARTING FROM
                                        ;<DSK-ADRES>
001754  032272                  DH54    ;PC     RKCS    RKER    RKDS    RKDA
                                        ;DRV#   CYL   <DSK-ADRES>         SUR     SECTR
001756  031672                  DT54    ;$ERRPC $REG0   $REG1   $REG2   $REG3
                                        ;$REG4  $REG5   $REG6   $REG7
001760  000000                  0

                                ;ITEM   33

001762  026521                  EM56    ;RKDA INCREMENTED WRONG ON WRITE OR WRITE FORMAT
001764  032401                  DH56    ;PC     EXPCT:  DRV#    CYL     SUR     SECTR
                                        ;RECVD: DRV#    CYL     SUR     SECTR
001766  031672                  DT54    ;$ERRPC         $REG0   $REG1   $REG2   $REG3
                                        ;$REG4  $REG5   $REG6   $REG7
001770  000000                  0

                                ;ITEM   34

001772  026560                  EM57    ;RKWC DIDN'T OVERFLOW ON WRITE OR WRITE FORMAT
001774  031773                  DH5     ;PC     RECVD
001776  031622                  DT1     ;$ERRPC $REG0
002000  000000                  0

                                ;ITEM   35

002002  026616                  EM60    ;RKBA INCREMENTED WRONG ON WRITE OR WRITE FORMAT
002004  031745                  DH4     ;PC     EXPCT   RECVD
002006  031630                  DT2     ;$ERRPC $REG0   $REG1
002010  000000                  0

                                ;ITEM   36

002012  026655                  EM61    ;RKER SET, ON WRITE/READ/FORMAT
002014  032053                  DH30    ;PC     RKCS    RKER    RKDS
002016  031660                  DT26    ;$ERRPC $REG0   $REG1   $REG2
002020  000000                  0

                                ;ITEM   37

002022  026712                  EM62    ;RKDB ERROR
002024  031745                  DH4     ;PC     EXPCT   RECVD
002026  031630                  DT2     ;$ERRPC $REG0   $REG1
002030  000000                  0
```

```
                                :ITEM   40

002032  026724                          EM63    ;RKDA INCREMENTED WRONG ON READ OR READ FORMAT
002034  032401                          DH56    ;PC     EXPCT:  DRV#    CYL     SUR     SECTR
                                                ;RECVD: DRV#    CYL     SUR     SECTR
002036  031672                          DT54    ;SERRPC         $REG0   $REG1   $REG2   $REG3
                                                ;$REG4  $REG5   $REG6   $REG7
002040  000000                          0

                                :ITEM   41

002042  026770                          EM64    ;RKWC DID NOT OVERFLOW ON READ OR READ FORMAT
002044  032536                          DH64    ;PC     RKWC    RKDA
002046  031630                          DT2     ;SERRPC $REG0   $REG1
002050  000000                          0

                                :ITEM   42

002052  027033                          EM65    ;RKBA INCREMENTED WRONG ON READ OR READ FORMAT
002054  031745                          DH4     ;PC     EXPCT   RECVD
002056  031630                          DT2     ;SERRPC $REG0   $REG1
002060  000000                          0

                                :ITEM   43

002062  027077                          EM66    ;INCORRECT HEADER FROM 'SECTOR'
002064  032532                          DH66    ;PC     SECTR   EXPCT   RECVD
002066  031660                          DT26    ;SERRPC $REG0   $REG1   $REG2.
002070  000000                          0

                                ;ITEM   44

002072  027136                          EM67    ;DATA ERROR
002074  032570                          DH67    ;PC     EXPCT   RECVD   DSK-ADRES
002076  031660                          DT26    ;SERRPC $REG0   $REG1   $REG2
002100  000000                          0

                                ;ITEM   45

002102  027151                          EM70    ;'CNTRL RDY' DIDN'T SET ON DOING READ/FMT STARTING
                                                ; FROM <DSK-ADRES>
002104  032272                          DH54    ;PC     RKCS    RKER    RKDS    RKDA
                                                ;DRV#   CYL     <DSK-ADRES>     SUR     SECTR
002106  031672                          DT54    ;SERRPC $REG0   $REG1   $REG2   $REG3
                                                ;$REG4  $REG5   $REG6   $REG7
002110  000000                          0

                                ;ITEM   46

002112  027242                          EM71    ;'HE' OR 'ERR' BIT SET ON READ/FMT STARTING
                                                ; FROM <DSK-ADRES>
002114  032272                          DH54    ;PC     RKCS    RKER    RKDS    RKDA
                                                ;DRV#   CYL     <DSK-ADRES>     SUR     SECTR
002116  031672                          DT54    ;SERRPC $REG0   $REG1   $REG2   $REG3
                                                ;$REG4  $REG5   $REG6   $REG7
```

```
   667  002120  000000                        0
   668
   669                                ;ITEM   47
   670
   671  002122  027320                        EM72    ;WRONG DRIVE ID IN RKDS AFTER SEEK
   672  002124  031745                        DH4     ;PC     EXPCT   RECVD
   673  002126  031630                        DT2     ;SERRPC SREG0   SREG1
   674  002130  000000                        0
   675
   676                                ;ITEM   50
   677
   678  002132  027362                        EM73    ;HARDWARE POLL, DRIVE ID BITS(13-15) SHOULD BE CLEAR
   679  002134  032111                        DH34    ;PC     RKDS
   680  002136  031630                        DT2     ;SERRPC SREG0
   681  002140  000000                        0
   682
   683                                ;ITEM   51
   684
   685  002142  027434                        EM74    ;HARDWARE POLL, INTERRUPTING DRIVE # NOT PRESENT
   686  002144  032630                        DH74    ;PC     DRIVE #
   687  002146  031622                        DT1     ;SERRPC SREG0
   688  002150  000000                        0
   689
   690                                ;ITEM   52
   691
   692  002152  027504                        EM75    ;'DRIVE #' DID NOT INTERRUPT DURING HARDWARE POLL
   693  002154  032630                        DH74    ;PC     DRIVE #
   694  002156  031622                        DT1     ;SERRPC SREG0
   695  002160  000000                        0
   696
   697                                ;ITEM   53
   698
   699  002162  027554                        EM76    ;SCP DID NOT SET AFTER WAS DONE
   700  002164  033004                        DH117   ;PC     RKCS
   701  002166  031622                        DT1     ;SERRPC SREG0
   702  002170  000000                        0
   703
   704                                ;ITEM   54
   705
   706  002172  027617                        EM77    ;RKDA CHANGED AFTER 'DRIVE RESET'
   707  002174  031745                        DH4     ;PC     EXPCT   RECVD
   708  002176  031630                        DT2     ;SERRPC SREG0   SREG1
   709  002200  000000                        0
   710
   711                                ;ITEM   55
   712
   713  002202  027654                        EM100   ;DATA ERROR AT WORD#
   714  002204  032651                        DH100   ;PC     WORD#   EXPCT   RECVD
   715  002206  031660                        DT26    ;SERRPC SREG0   SREG1   SREG2
   716  002210  000000                        0
   717
   718                                ;ITEM   56
   719
   720  002212  027677                        EM101   ;CNTRL RDY DID NOT SET AFTER READ CHECK
   721  002214  032225                        DH44    ;PC     RKCS    RKER    RKDS    RKDA
   722  002216  031640                        DT20    ;SERRPC SREG0   SREG1   SREG2   SREG3
```

```
   723  002220  000000                          0
   724
   725                                  ;ITEM   57
   726
   727  002222  027741                          EM102   ;'ERR' OF 'HE' SET ON READ CHECK
   728  002224  032053                          DH30    ;PC     RKCS    RKER    RKDS
   729  002226  031660                          DT26    ;$ERRPC $REGO   $REG1   $REG2
   730  002230  000000                          0
   731
   732                                  ;ITEM   60
   733
   734  002232  027765                          EM103   ;'CSE' ON READ CHECK
   735  002234  032706                          DH103   ;PC     RKER
   736  002236  031622                          DT1     ;$ERRPC $REGO
   737  002240  000000                          0
   738
   739                                  ;ITEM   61
   740
   741  002242  030003                          EM104   ;RKWC DID NOT OVERFLOW ON READ CHECK OR WRITE CHECK
   742  002244  032722                          DH104   ;PC     RECVD   RKCS
   743  002246  031630                          DT2     ;$ERRPC $REGO   $REG1
   744  002250  000000                          0
   745
   746                                  ;ITEM   62
   747
   748  002252  030054                          EM105   ;RKDA INCREMENTED WRONG ON READ CHECK
   749  002254  031745                          DH4     ;PC     EXPCT   RECVD
   750  002256  031630                          DT2     ;$ERRPC $REGO   $REG1
   751  002260  000000                          0
   752
   753                                  ;ITEM   63
   754
   755  002262  030112                          EM106   ;RKBA CHANGED AFTER READ CHECK
   756  002264  031745                          DH4     ;PC     EXPCT   RECVD
   757  002266  031630                          DT2     ;$ERRPC $REGO   $REG1
   758  002270  000000                          0
   759
   760                                  ;ITEM   64
   761
   762  002272  030143                          EM107   ;MEMORY WORD CHANGED AFTER READ CHECK
   763  002274  032746                          DH107   ;PC     LOC     EXPCT   RECVD
   764  002276  031660                          DT26    ;$ERRPC $REGO   $REG1   $REG2
   765  002300  000000                          0
   766
   767                                  ;ITEM   65
   768
   769  002302  030204                          EM110   ;CNTRL RDY DID NOT SET AFTER WRITE CHECK
   770  002304  032225                          DH44    ;PC     RKCS    RKER    RKDS    RKDA
   771  002306  031640                          DT20    ;$ERRPC $REGO   $REG1   $REG2   $REG3
   772  002310  000000                          0
   773
   774                                  ;ITEM   66
   775
   776  002312  030247                          EM111   ;HE OR ERR BIT SET AFTER DOING WRITE CHECK
   777  002314  032053                          DH30    ;PC     RKCS    RKER    RKDS
   778  002316  031660                          DT26    ;$ERRPC $REGO   $REG1   $REG2
```

```
000779  002320  000000                        0
000780
000781                                ;ITEM   67
000782
000783  002322  030274                        EM112   ;WRITE CHECK ERROR
000784  002324  032053                        DH30    ;PC     RKCS    RkER    RKDS
000785  002326  031660                        DT26    ;$ERRPC $REG0   $REG1   $REG2
000786  002330  000000                        0
000787
000788                                ;ITEM   70
000789
000790  002332  030315                        EM113   ;RKDA INCREMENTED WRONG ON WRITE CHECK
000791  002334  031745                        DH4     ;PC     EXPCT   RECVD
000792  002336  031630                        DT2     ;$ERRPC $REG0   $REG1
000793  002340  000000                        0
000794
000795                                ;ITEM   71
000796
000797  002342  030354                        EM114   ;RKBA INCREMENTED WRONG ON WRITE CHECK
000798  002344  031745                        DH4     ;PC     EXPCT   RECVD
000799  002346  031630                        DT2     ;$ERRPC $REG0   $REG1
000800  002350  000000                        0
000801
000802                                ;ITEM   72
000803
000804  002352  030413                        EM115   ;RKBA INCREMENTED WITH IBA SET
000805  002354  031745                        DH4     ;PC     EXPCT   RECVD
000806  002356  031630                        DT2     ;$ERRPC $REG0   $REG1
000807  002360  000000                        0
000808
000809                                ;ITEM   73
000810
000811  002362  030447                        EM116   ;WRONG MEMORY LOCATION CHANGED WITH IBA SET
000812  002364  032651                        DH100   ;PC     WORD#   EXPCT   RECVD
000813  002366  031660                        DT26    ;$ERRPC $REG0   $REG1   $REG2
000814  002370  000000                        0
000815
000816                                ;ITEM   74
000817
000818  002372  030522                        EM117   ;RK11 DID NOT INTERRUPT WHEN IDE WAS SET
000819  002374  033004                        DH117   ;PC     RKCS
000820  002376  031622                        DT1     ;$ERRPC $REG0
000821  002400  000000                        0
000822
000823                                ;ITEM   75
000824
000825  002402  030567                        EM120   ;RK11 DID NOT INTERRUPT AFTER SEEK WAS INITIATED
000826  002404  033004                        DH117   ;PC     RKCS
000827  002406  031622                        DT1     ;$ERRPC $REG0
000828  002410  000000                        0
000829
000830                                ;ITEM   76
000831
000832  002412  030642                        EM121   ;SCP SET BEFORE SEEK COMPLETED
000833  002414  033004                        DH117   ;PC     RKCS
000834  002416  031622                        DT1     ;$ERRPC $REG0
```

```
000000                          0

                        ;ITEM   77

002422  030700                  EM122   ;RK11 DID NOT INTERRUPT AFTER SEEK COMPLETED
002424  032053                  DH30    ;PC     RKCS    RKER    RKDS
002426  031660                  DT26    ;$ERRPC $REG0   $REG1   $REG2
002430  000000                  0

                        ;ITEM   100

002432  030747                  EM123   ;CNTRL RESET DID NOT CLEAR 'SCP' BIT
002434  033004                  DH117   ;PC     RKCS
002436  031622                  DT1     ;$ERRPC $REG0
002440  000000                  0

                        ;ITEM   101

002442  031006                  EM124   ;RK11 DID NOT INTERRUPT AFTER READ WAS DONE
002444  033004                  DH117   ;PC     RKCS
002446  031622                  DT1     ;$ERRPC $REG0
002450  000000                  0

                        ;ITEM   102

002452  031050                  EM125   ;CNTRL RESET DID NOT CLEAR REGISTER
002454  031716                  DH2     ;PC     REGADD  RECVD
002456  031630                  DT2     ;$ERRPC $REG0   $REG1
002460  000000                  0

                        ;ITEM   103

002462  031107                  EM126   ;RK11 DID NOT INTERRUPT AT CPU LEVEL
002464  033020                  DH126   ;PC     LEVEL   RKCS
002466  031630                  DT2     ;$ERRPC $REG0   $REG1
002470  000000                  0

                        ;ITEM   104

002472  031150                  EM127   ;RK11 INTERRUPTED AT WRONG CPU LEVEL
002474  033020                  DH126   ;PC     LEVEL   RKCS
002476  031630                  DT2     ;$ERRPC $REG0   $REG1
002500  000000                  0

                        ;ITEM   105

002502  031212                  EM130   ;'ERR BIT' DID NOT SET IN RKER
002504  033046                  DH130   ;PC     RKCS    RKER    ERR BIT
002506  031660                  DT26    ;$ERRPC $REG0   $REG1   $REG2
002510  000000                  0

                        ;ITEM   106

002512  031247                  EM131   ;HE OR ERR DID NOT SET
002514  033105                  DH131   ;PC     RKCS    RKER
```

```
                                        DT2     ;$ERRPC $REGO   $REG1
                                        0

                                ;ITEM   107

                                        EM132   ;RKER ERROR
                                        DH4     ;PC     EXPCT   RECVD
                                        DT2     ;$ERRPC $REGO   $REG1
                                        0

                                ;ITEM   110

                                        EM133   ;NXC BIT DID NOT SET
                                        DH133   ;PC     RKCS    RKER    RKDA
                                        DT26    ;PC     $REGO   $REG1   $REG2
                                        0

                                ;ITEM   111

                                        EM134   ;RK11 DIDN'T INTERRUPT ON SOFT ERROR
                                        DH131   ;PC     RKCS    RKER
                                        DT2     ;$ERRPC $REGO   $REG1
                                        0

                                ;ITEM   112

                                        EM135   ;MEX BITS INCREMENTED WRONG IN RKCS
                                        DH4     ;PC     EXPCTD  RECVD
                                        DT2     ;$ERRPC $REGO   $REG1
                                        0

                                ;ITEM   113

                                        EM110   ;CNTRL RDY DID NOT SET AFTER WRT CHK
                                        DH14    ;PC     RKCS    RKER    RKWC
                                        DT26    ;$ERRPC $REGO   $REG1   $REG2
                                        0

                                ;ITEM   114

                                        EM137   ;'WPS' NOT CLEAR
                                        DH44    ;PC     RKCS    RKER    RKDS    RKDA
                                        DT20    ;$ERRPC $REGO   $REG1   $REG2   $REG3
                                        0

                                ;ITEM   115

                                        EM140   ;DATA ERROR ON TRANSFER FROM DISK TO TTY
                                        DH140   ;PC     EXPCT   RECVD   RKBA    RKCS
                                        DT20    ;$ERRPC $REGO   $REG1   $REG2   $REG3
                                        0

                                ;ITEM   116

                                        EM141   ;'DRIVE #' PRESENT, BUT NOT SPECIFIED
```

```
947  002614  032630                    DH74    ;PC     DRIVE #
948  002616  031622                    DT1     ;SERRPC $REGO
949  002620  000000                    0
950
951                              ;ITEM   117
952
953  002622  025266                    EM11    ;RKWC ERROR
954  002624  031745                    DH4     ;PC     EXPCT   RECVD
955  002626  031630                    DT2     ;SERRPC $REGO   $REG1
956  002630  000000                    0
957                              ;ITEM   120
958  002632  031560                    EM142
959  002634  000000                    0
960
961
962
```

```
                        000005                  START:  RESET                   ;CLEAR THE BUS
                                                .SBTTL  INITIALIZE THE COMMON TAGS
                                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
002640  012706  001100                                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
002644  005026                                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
002646  022706  001140                                  CMP     #SWR,R6 ;;DONE?
002652  001374                                          BNE     .-6             ;;LOOP BACK IF NO
002654  012706  001100                                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                                ;;INITIALIZE A FEW VECTORS
002660  012737  022046  000020                          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
002666  012737  000340  000022                          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
002674  012737  022320  000030                          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
002702  012737  000340  000032                          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002710  012737  024574  000034                          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002716  012737  000340  000036                          MOV     #340,@#TRAPVEC+2;LEVEL 7
002724  012737  024674  000024                          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
002732  012737  000340  000026                          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
002740  005037  001206                                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
002744  005037  001210                                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
002750  112737  000001  001115                          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
002756  012737  002756  001106                          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002764  012737  002764  001110                          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
002772  013746  000004                                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002776  012737  003032  000004                          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
003004  012737  177570  001140                          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
003012  012737  177570  001142                          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
003020  022777  177777  176112                          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
003026  001012                                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
003030  000403                                          BR      65$             ;;BRANCH IF NO TIMEOUT
003032  012716  003040                          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
003036  000002                                          RTI
003040  012737  000176  001140                  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
003046  012737  000174  001142                          MOV     #DISPREG,DISPLAY
003054  012637  000004                          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

                                                .SBTTL  TYPE PROGRAM NAME
                                                ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
003060  005227  177777                                  INC     #-1             ;;FIRST TIME?
003064  001044                                          BNE     67$             ;;BRANCH IF NO
003066  104401  003124                                  TYPE    ,68$            ;;TYPE ASCIZ STRING
                                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
003072  005737  000042                                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
003076  001006                                          BNE     69$             ;;BRANCH IF YES
003100  023727  001140  000176                          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
003106  001005                                          BNE     70$             ;;BRANCH IF NO
003110  104406                                          GTSWR                   ;;GET SOFT-SWR SETTINGS
003112  000403                                          BR      70$
003114  112737  000001  001134          69$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
003122                                          70$:
003122  000425                                          BR      67$             ;;GET OVER THE ASCIZ
                                                ;;68$:  .ASCIZ  <CRLF>/RK11 LOGIC TEST II/<15><12>/MAINDEC-11-DZRKK-D/<CRLF>
003176                                          67$:
003176  012700  001410                                  MOV     #DDPCH,R0
```

```
1019  003202  012701  177765          MOV     #-13,R1
1020  003206  005020          1$:     CLR     (R0)+
1021  003210  005201                  INC     R1
1022  003212  001375                  BNE     1$
1023  003214  005227  177777          INC     #-1             ;FIRST START ?
1024  003220  001020                  BNE     START1          ;BR IF NOT
1025  003222  013746  000004          MOV     ERRVEC,-(SP)    ;SAVE ERROR VECTOR ADDRESS
1026  003226  012737  003242  000004  MOV     #2$,ERRVEC      ;NEW VECTOR ADDRESS
1027  003234  005737  177776          TST     PS              ;SEE IF PROGRAM CAN REFERENCE THE
1028                                                          ;PROCESSOR STATUS WORD
1029  003240  000406                  BR      3$              ;BR IF REFERENCE DIDN'T CAUSE TRAP
1030  003242  012737  000140  001400  2$:     MOV     #140,RKPRI      ;SETUP INTERRUPTING PRIORITY TO VALUE
1031                                                          ;WHICH WILL ALLOW INTERRUPT ON AN LSI-11
1032  003250  012716  003256          MOV     #3$,(SP)        ;SETUP RETURN ADDRESS
1033  003254  000002                  RTI                     ;RETURN
1034  003256  012637  000004          3$:     MOV     (SP)+,ERRVEC    ;RESTORE THE ERROR VECTOR
1035                                  ;
1036                                  ;FIND OUT IF ACT11, 'XXDP' CHAIN OR DUMP MODE
1037                                  ;
1038  003262  012700  001410          START1: MOV     #DDPCH,R0
1039  003266  012701  177766                  MOV     #-12,R1         ;CLEAR OUT DRIVE TABLE AREA
1040  003272  005020          1$:     CLR     (R0)+
1041  003274  005201                  INC     R1
1042  003276  001375                  BNE     1$
1043  003300  122737  000002  000041          CMPB    #2,41           ;LOADED FROM AN RK05 ?
1044  003306  001166                  BNE     ST2             ;BR IF NOT
1045  003310  013737  000040  001410          MOV     40,DDPCH        ;GET DEVICE INDICATOR AND DRIVE ADDRESS OF
1046                                                          ;LOADING RK05
1047  003316  122737  000010  001410          CMPB    #10,DDPCH       ;VALID DRIVE NUMBER IN BYTE 40 ?
1048  003324  101002                  BHI     2$              ;BR IF YES
1049  003326  105037  001410          CLRB    DDPCH           ;MUST BE DRIVE ZERO WHICH LOADED
1050                                                          ;THIS PROGRAM
1051  003332  005737  000042          2$:     TST     42              ;CHAIN MODE OR ACT11 AUTO ACCEPT ?
1052  003336  001432                  BEQ     4$              ;BR IF NEITHER
1053  003340  005737  001410          TST     DDPCH           ;RUNNING FROM AN RK05 ?
1054  003344  001002                  BNE     3$              ;BR IF YES
1055  003346  000137  004210          JMP     ST3             ;FIND OUT NUMBER OF DRIVES
1056  003352                          3$:
1057  003352  104401  003360          TYPE    ,65$            ;;TYPE ASCIZ STRING
1058  003356  000413                  BR      64$             ;;GET OVER THE ASCIZ
1059                                  ;;65$:  .ASCIZ  <15><12>/NOT TESTING DRIVE /
1060  003406                          64$:
1061  003406  005046                  CLR     -(SP)           ;CLEAR WORD ON STACK
1062  003410  113716  001410          MOVB    DDPCH,(SP)      ;GET DRIVE ADDRESS
1063  003414  104403                  TYPOS                   ;TYPE THE ADDRESS
1064  003416     001                  .BYTE   1               ;ONLY 1 CHARACTER
1065  003417     000                  .BYTE   0               ;SUPRESS LEADING ZEROS
1066  003420  000137  004210          JMP     ST3             ;GET NUMBER OF DRIVES
1067  003424  005227  177777          4$:     INC     #-1             ;FIRST TIME THROUGH HERE ?
1068  003430  001115                  BNE     ST2             ;BR IF NOT
1069  003432  104401  003440          TYPE    ,67$            ;;TYPE ASCIZ STRING
1070  003436  000411                  BR      66$             ;;GET OVER THE ASCIZ
1071                                  ;;67$:  .ASCIZ  <15><12>/TO TEST DRIVE /
1072  003462                          66$:
1073  003462  005046                  CLR     -(SP)           ;CLEAR WORD ON THE STACK
1074  003464  113716  001410          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.
1092  003664  012700  001412   ST2:    MOV     #DRIVS,R0
1093  003670  012701  177767           MOV     #-11,R1
1094  003674  005020           13$:    CLR     (R0)+
1095  003676  005201                   INC     R1
1096  003700  001375                   BNE     13$
1097  003702  104401  003710           TYPE    ,65$            ;;TYPE ASCIZ STRING
1098  003706  000415                   BR      64$             ;;GET OVER THE ASCIZ
1099                          ;;65$:  .ASCIZ  <15><12>/DRIVES TO BE TESTED ?/<15><12>
1100  003742                  64$:
1101  003742  104411                   RDLIN
1102  003744  012600                   MOV     (SP)+,R0        ;GET STARTING ADRES OF ASCII STRING
1103  003746  012701  177770           MOV     #-10,R1         ;SET UP COUNT
1104  003752  112002           1$:     MOVB    (R0)+,R2        ;GET ASCII CHARACTER
1105  003754  042702  177400           BIC     #177400,R2      ;MASK UNWANTED BITS
1106  003760  012703  001414           MOV     #DRIV0,R3
1107  003764  012704  177770           MOV     #-10,R4
1108  003770  012705  000060           MOV     #60,R5
1109  003774  020502           2$:     CMP     R5,R2           ;WAS THE TYPED IN CHARACTER
1110                                                           ;A NUMBER BETWEEN 0-7?
1111  003776  001414                   BEQ     3$              ;YES, BRANCH
1112  004000  005205                   INC     R5              ;NO, INCREMENT
1113  004002  005723                   TST     (R3)+           ;INCREMENT POINTER TO DRV FLAG
1114  004004  005204                   INC     R4              ;CHARACTER THAT WAS INPUT
1115  004006  001372                   BNE     2$              ;SHOULD BE 0-7, IF ANY OTHER
1116                                                           ;TYPE ?? & AGAIN ASK FOR
1117                                                           ;DRIVS TO BE TSTD?
1118  004010  005702                   TST     R2              ;IS IT A TERMINATOR?
1119  004012  001461                   BEQ     6$              ;YES, EXIT. NO DRIVES INDICATED.
1120  004014                  4$:
1121  004014  104401  004022           TYPE    ,67$            ;;TYPE ASCIZ STRING
1122  004020  000402                   BR      66$             ;;GET OVER THE ASCIZ
1123                          ;;67$:  .ASCIZ  /??/
1124  004026                  66$:
1125  004026  000716                   BR      ST2             ;GO, AGAIN ASK QUESTION
1126  004030  005713           3$:     TST     @R3     ;SEE IF ALL READY SELECTED
1127  004032  001370                   BNE     4$      ;ERROR IF SELECTED  ALL READY
1128  004034  005213                   INC     @R3             ;SET UP FLAG FOR THE DRIVE
1129  004036  005237  001412           INC     DRIVS           ;INCREMENT TOTAL NO OF DRIVES PRESENT
1130  004042  111002           11$:    MOVB    @R0,R2          ;GET NEXT CHAR
```

```
1131  004144  042702  177400          BIC     #177400,R2      ;CHARACTER ONLY
1132  004150  022702  000106          CMP     #'F,R2          ;IS IT F?
1133  004154  001026                  BNE     8$              ;NO, GO ON
1134  004156  052713  100000          BIS     #BIT15,@R3      ;SET BIT 15 TO SHOW RK05F
1135  004162  032705  000001          BIT     #BIT0,R5        ;EVEN DRIVE?
1136  004166  001407                  BEQ     9$              ;EVEN DRIVE SO BRANCH
1137  004170  005763  177776          TST     -2(R3)          ;CHECK EVEN DRIVE
1138  004174  001347                  BNE     4$              ;EVEN ALL READY SELECTED
1139  004176  012763  100001  177776  MOV     #BIT15!BIT0,-2(R3)      ;SELECT EVEN DRIVE
1140  004204  000406                  BR      10$             ;CONTINUE
1141  004206  005763  000002          9$:     TST     2(R3)           ;CHECK ODD DRIVE
1142  004212  001340                  BNE     4$              ;ERROR IF SELECTED BE
1143  004214  012763  100001  000002  MOV     #BIT15!BIT0,2(R3)       ;SELECT ODD DRIVE
1144  004222  005237  001412          10$:    INC     DRIVS           ;COUNT DRIVES SELECTED
1145  004226  105720                  TSTB    (R0)+           ;POINT TO NEXT CHAR
1146  004230  000744                  BR      11$             ;CHECK FOR COMMA
1147  004232  022702  000054          8$:     CMP     #54,R2          ;IS IT A 'COMMA'?
1148  004236  001403                  BEQ     5$              ;YES, GO PROCESS NXT WORD
1149  004240  005702                  TST     R2              ;NO, IS IT A TERMINATOR?
1150  004242  001324                  BNE     4$              ;IF NOT, SOMETHING WRONG
1151                                                          ;GO ASK QUESTION AGAIN
1152  004244  000404                  BR      6$              ;EXIT, IF A TERMINATOR
1153  004246  105720          5$:     TSTB    (R0)+           ;INCREMENT PTR TO NXT BYTE
1154                                                          ;IN INPUT BUFFER
1155  004250  005201                  INC     R1              ;THERE SHOULD BE NO MORE THAN
1156  004252  001277                  BNE     1$              ;8 DRIVES, HENCE IF MORE
1157  004254  000717                  BR      4$              ;THAN 8 DIFFERENT NOS. TYPED IN, ERROR!
1158                                                          ;GO AGAIN ASK THE QUESTION
1159
1160  004256  005037  001440          6$:     CLR     SIZYET          ;NO SIZING NEEDED
1161  004262  032777  002000  174750  BIT     #SW10,@SWR      ;TESTING ON SIMULATOR?
1162  004270  001003                  BNE     7$              ;YES, BRANCH
1163  004272  005037  001344          CLR     SIMUL           ;NO, CLR FLAG
1164  004276  000502                  BR      ST4
1165
1166  004300  012737  000001  001344  7$:     MOV     #1,SIMUL        ;SET FLAG TO INDICATE SIMULATOR
1167  004306  000476                  BR      ST4
1168
1169
1170
1171                                  ;CHECK NUMBER OF DRIVES
1172  004310  012737  177777  001440  ST3:    MOV     #-1,SIZYET      ;CHECK FOR RK05F LATER
1173  004316  012737  004370  000004          MOV     #5$,@#4         ;SET UP ADRES FOR TIME-OUT VECTOR
1174  004324  005777  175076          TST     @RKDS           ;REFERENCE RKDS
1175  004330  005777  175104          TST     @RKDA           ;REFERENCE RKDA
1176  004334  012737  004462  000004  MOV     #BADTMO,@#4
1177  004342  104401                  TYPE
1178  004344  001216                  MSG1
1179  004346  012700  177770          MOV     #-10,R0         ;INITIALIZE COUNT FOR THE 8 DRIVES
1180  004352  005037  001412          CLR     DRIVS           ;INITIALIZE # OF DRIVES PRESENT TO 0
1181  004356  005001                  CLR     R1              ;INITIALIZE ADDRESS TO DRIVE 0
1182  004360  005004                  CLR     R4
1183  004362  012702  001414          MOV     #DRIVG,R2
1184  004366  010177  175046          1$:     MOV     R1,@RKDA        ;ADDRESS THE DRIVE
1185  004372  020177  175042          CMP     R1,@RKDA        ;CHECK, WAS IT ADDRESSED?
1186  004376  001405                  BEQ     3$              ;YES
```

```
1187  004300  012703  004304            MOV     #2$,R3
1188  004304  004737  020734      2$:   JSR     PC,$TYERM       ;WHILE CHECKING NUMBER OF DRIVE
1189                                                            ;UNDER NON-MANUAL MODE :-
1190                                                            ;RKDA HAD TO BE ADRESED BUT
1191                                                            ;IT WAS FOUND THAT THE DRIVE NO
1192                                                            ;THAT WAS WRITTEN COULD NOT BE READ BACK
1193                                                            ;CORRECTLY.
1194
1195  004310  000413                BR      4$
1196  004312  032777  000200  175006  3$:   BIT     #200,@RKDS      ;CHECK IF 'DRY' BIT IS SET, IF SET DRIVE IS
1197                                                            ;PRESENT
1198  004320  001407                BEQ     4$              ;
1199  004322  104401                TYPE
1200  004324  001213                $CRLF
1201  004326  005237  001412        INC     DRIVS           ;IF PRESENT, INCREMENT # OF DRIVES
1202  004332  005212                INC     (R2)            ;SET UP FLAG INDICATING THIS DRIVE PRESENT
1203  004334  010446                MOV     R4,-(SP)
1204  004336  104402                TYPOC
1205  004340  005722            4$: TST     (R2)+           ;SHIFT POINTER TO NXT DRIVE INDICATOR
1206  004342  062701  020000        ADD     #20000,R1       ;SET UP ADDRESS FOR THE NEXT DRIVE
1207  004346  005204                INC     R4              ;HAVE U CHECKED FOR ALL 8 DRIVES
1208  004350  005200                INC     R0
1209  004352  001345                BNE     1$
1210  004354  005737  001412        TST     DRIVS
1211  004360  001011                BNE     ST4
1212  004362  104401                TYPE
1213  004364  001236                MSG2
1214  004366  000406                BR      ST4             ;GO CHECK THE DRIVE INDEPENDENT
1215                                                            ;CONTROLLER LOGIC
1216  004370  011603            5$: MOV     (SP),R3         ;GET PC WHERE TIMEOUT OCCURED
1217  004372  022626                CMP     (SP)+,(SP)+     ;RESTORE STACK
1218  004374  062703  177776        ADD     #-2,R3
1219  004400  004737  020734        JSR     PC,$TYERM       ;GO TYPE ERROR MESSAGE
1220                                                            ;WHILE CHECKING FOR THE NUMBER OF
1221                                                            ;DRIVES IN NON-MANUAL MODE:-
1222                                                            ;RKDS AND RKDA HAD TO BE REFERENCED, TIMEOUT
1223                                                            ;OCCURED ON REFERENCING.PC IN THE ERROR
1224                                                            ;MESSAGE INDICATES WHERE THE TIMEOUT OCCURED.
1225
1226                            ;
1227
1228
1229  004404  005037  001434    ST4:  CLR     TS6FLG
1230  004410  005737  001412        TST     DRIVS
1231  004414  001004                BNE     1$
1232  004416  004737  021650        JSR     PC,WATIME
1233  004422  000137  020560        JMP     $EOP
1234  004426  012737  001414  001354  1$:   MOV     #DRIVD,DRVPTR
1235  004434  005037  001352        CLR     DRVDON          ;INITIALIZE THE NO. OF DRIVES
1236                                                            ;THAT HAVE BEEN CHECKED
1237  004440  005037  001350        CLR     DRIVAD          ;INITIALIZE DRIVE ADDRESS TO
1238                                                            ;THE FIRST DRIVE
1239  004444  012737  004462  000004  MOV     #BADTMO,@#4     ;SET TIME OUT VECTOR FOR UNEXPECTED
1240                                                            ;TIME OUTS
1241  004452  012777  004526  174722  MOV     #BADINT,@RKVEC  ;SET UP RK11 INTERRUPT VECTOR FOR
1242                                                            ;UNEXPECTED INTERRUPTS FROM RK11
```

```
                                        BR      TST1            ;GO TO TEST 1


                                        ;THIS ROUTINE HANDLES UNEXPECTED TIME OUTS

004462  011600            BADTMO: MOV     (SP),R0 ;SAVE PC WHERE TIME OUT OCCURED
004464  005740                    TST     -(R0)
004466  022626                    CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
004470  104401  004476            TYPE    ,65$            ;;TYPE ASCIZ STRING
004474  000407                    BR      64$             ;;GET OVER THE ASCIZ
                            ;;65$:  .ASCIZ  <:5><:2 >'TIMOUT,PC=/
004514                      64$:
004514  010046                    MOV     R0,-(SP)        ;SET UP FOR TYPING OUT PC
004516  104402                    TYPOC                   ;GO TYPE OUT OCTAL PC
004520  000000                    HALT
004522  000137  002636            JMP     @#START


                                        ;THIS ROUTINE HANDLES UNEXPECTED INTERRUPTS FROM RK11
                                        ;SW 9 AND 10 FOR LOOPING ON ERROR
                                        ;AND LOOPING ON TEST IN WHICH TIMEOUT
                                        ;OCCURRED, ARE PROVIDED.

004526  011600            BADINT: MOV     (SP),R0         ;SAVE PC WHERE INTERRUPT OCCURED
004530  005740                    TST     -(R0)
004532  032777  020000  174400            BIT     #20000,@SWR     ;INHIBIT ERROR TYPEOUT?
004540  001014                    BNE     1$              ;YES, DON'T TYPE OUT
004542  104401                    TYPE
004544  001213                    $CRLF
004546  104401                    TYPE
004550  025702                    EMT3                    ;TYPE 'UNEXPEXTED RK11 INTERRUPT'
                                                          ;TYPE ' AT PC='
004552  104401  004560            TYPE    ,65$            ;;TYPE ASCIZ STRING
004556  000403                    BR      64$             ;;GET OVER THE ASCIZ
                            ;;65$:  .ASCIZ  /,PC=/
004566                      64$:
004566  010046                    MOV     R0,-(SP)        ;SET UP FOR TYPING OUT PC
004570  104402                    TYPOC                   ;GO TYPE OCTAL PC WHERE BAD
                                                          ;INTERUPT OCCURED
004572  032777  001000  174340  1$:       BIT     #1000,@SWR      ;LOOP ON ERROR?
004600  001403                    BEQ     2$              ;NO, BRANCH
004602  022626                    CMP     (SP)+,(SP)+     ;YES, REPOSITION STACK
004604  000177  174276            JMP     @$LPADR         ;GO TO THE STARTING ADDRESS OF
                                                          ;THE TEST THAT GAVE UNEXPECTED INTERRUPT
004610  032777  040000  174322  2$:       BIT     #40000,@SWR     ;LOOP ON TEST?
004616  001401                    BEQ     3$              ;NO, BRANCH
004620  000002                    RTI                     ;YES, LOOP. GO BACK WHER U INTERRUPTED FROM.
004622  000000            3$:     HALT                    ;UNEXPECTED INTERRUPT OCCURED AS
                                                          ;INDICATED IN THE TYPE OUT.U CAN LOOP
                                                          ;ON ERROR, TEST,OR INHIBIT TYPEOUT BY
                                                          ;SETTING APPROPRIATE SWITCHES.
004624  000137  002636            JMP     @#START         ;GO BACK TO THE START OF THE
                                                          ;PROGRAM. THUS PRESSING CONTINUE
```

```
                                                ;AFTER THE ABOVE HALT WILL
                                                ;RESTART THE PROGRAM



                        ;RESTART AFTER POWER FAIL
                        ;THE PROGRAM WOULD RESTART HERE IF POWER CAME BACK AFTER A FALIURE.
004630  004737  021650  PFSTRT: JSR     PC,WATIME       ;KILL TIME



                        ;;*****************************************************************
                        ;*TEST 1        CHECK THAT THE DRIVES THAT ARE NOT SPECIFIED ARE NOT FOUND TO BE PRESENT
                                ;*THIS TEST CHECKS THAT THE DRIVES THAT ARE NOT SPECIFIED
                                ;*(IN RESPONSE TO "DRIVS TO BE TSTD?") ARE NOT FOUND TO BE PRESENT.
                                ;*EVERY DRIVE  FROM 0 TO 7 IS ADDRESSED. IF A PARTICULAR DRIVE
                                ;*GIVES 'DRY' (IN RKDS), IT IS CHECKED THAT  THIS DRIVE
                                ;*WAS SPECIFIED BY THE USER, IF IT WAS NOT AN ERROR IS
                                ;*REPORTED, GIVING THE DRIVE NUMBER. IT IS LIKELY THAT THE USER
                                ;*MAY HAVE FORGOTTEN TO PUT THE DRIVE (THAT IS NOT SPECIFIED) ON
                                ;*'LOAD'. IF THIS IS THE CASE THEN PUT THIS DRIVE ON 'LOAD'
                                ;*IF THIS IS NOT THE CASE, THERE IS A GENUINE ERROR. (TWO DIFFERENT
                                ;*DRIVE ADDRESSES MAY BE RESULTING IN THE SELECTION OF THE SAME
                                ;*PHYSICAL DRIVE.)
                        ;;*****************************************************************
004634  000004         TST1:   SCOPE

004636  012700  001414         MOV     #DRIVD,R0       ;INITIALIZE POINTER
004642  005001                 CLR     R1              ;INITIALIZE DRIVE ADRES 0
004644  005002                 CLR     R2              ;INITIALIZE DRIVE # 0
004646  005737  001410  1$:    TST     DDPCH           ;LOADED FROM AN RK05 ?
004652  001403                 BEQ     2$              ;B IF NOT
004654  120237  001410         CMPB    R2,DDPCH        ;LOADED FROM THIS DRIVE ?
004660  001435                 BEQ     4$              ;BR IF YES
004662  010177  174452  2$:    MOV     R1,@RKDA        ;ADRES THE DRIVE
004666  105777  174434         TSTB    @RKDS           ;DRIVE READY?
004672  100005                 BPL     3$              ;NO, THIS DRIVE NOT PRESENT
                                                        ;YES, THIS DRIVE SELECTED
004674  005710                 TST     @R0             ;WAS THIS DRIVE SPECIFIED BY
                                                        ;THE USER?
004676  001026                 BNE     4$              ;YES, OK
                                                        ;NO, THIS DRIVE # WAS NOT SPECIFIED
                                                        ;BY THE USER, BUT STILL IS GIVING
                                                        ;'DRY' WHEN ADRESED. REPORT EROR.
004700  010237  001162         MOV     R2,$REGD        ;GET DRIVE #
004704  104116                 ERROR   116             ;THIS DRIVE # WAS NOT SPECIFIED BY
                                                        ;THE USER, BUT WHEN ADRESED GAVE
                                                        ;'DRY'. CHECK THAT THIS DRIVE # IF
                                                        ;PHYSICALLY PRESENT IS ON 'LOAD'. IF
                                                        ;THIS IS NOT THE CASE, THEN ONE DRIVE
                                                        ;MAY BE GETTING SELECTED BY TWO DIFFERENT
                                                        ;LOGICAL ADDRESSES.
004706  005710         3$:     TST     @R0             ;CHECK THAT THIS DRIVE WAS NOT INDICATED
004710  001421                 BEQ     4$              ;IF IT WAS, & IT IS NOT FOUND TO BE
                                                        ;PRESENT (DRY CLEAR), REPORT ERROR.
```

```
1355  004712  004737  020702                  JSR     PC,GT4RG        ;GET RKCS, ER, DS, DA
1356  004716  104010                          ERROR   10              ;DRIVE # (AS IN RKDA) WAS INDICATED BY
1357                                                                  ;THE USER, BUT WAS NOT FOUND TO BE PRESENT.
1358                                                                  ;CHECK THAT THE  ROTARY DRIVE SELECTION
1359                                                                  ;SWITCH ON THE MODULE IS SET TO THE RIGHT
1360                                                                  ;DRIVE #.
1361
1362  004720  005010                          CLR     @R0             ;THIS DRIVE IS NOT FOUND TO BE PRESENT
1363                                                                  ;HENCE DROP IT FROM THE SELECTION TABLE.
1364  004722  010003                          MOV     R0,R3           ;DRIVE ADDR
1365  004724  162703  001414                  SUB     #DRIVD,R3       ;MINUS OFFSET FOR TABLE
1366  004730  042703  000003                  BIC     #3,R3           ;EVEN DRIVE OF PAIR
1367  004734  062703  001414                  ADD     #DRIVD,R3       ;POINT TO EVEN OF PAIR IF RK05 F
1368  004740  042723  100000                  BIC     #100000,(R3)+   ;NOT SPECIFIED AS F MODEL
1369  004744  042713  100000                  BIC     #100000,(R3)    ;SAME
1370  004750  005337  001412                  DEC     DRIVS           ;DECREMENT DRIVE COUNT
1371  004754  005202                  4$:     INC     R2              ;INCRMNT DRIVE #
1372  004756  005720                          TST     (R0)+           ;INCRMNT POINTER
1373  004760  062701  020000                  ADD     #20000,R1       ;INCRMNT ADRES TO NXT DRIVE
1374  004764  001330                          BNE     1$              ;LUP BAK IF NOT DONE
1375
1376
1377                                          ;THIS PART OF THE PROGRAM IS GOING TO BE REPEATED FOR
1378                                          ;EACH DRIVE PRESENT
1379                                          ;
1380                                          ;'DRIVAD' CONTAINS IN BITS 15,14,13 THE ADDRESS OF THE
1381                                          ;DRIVE BEING CURRENTLY CHECKED.
1382                                          ;
1383  004766                          NUDRV:
1384
1385
1386                                          ;;*****************************************************************
1387                                          ;TEST 2        FIND OUT NEXT DRIVE TO BE CHECKED
1388                                          ;THIS CODE FINDS OUT THE NEXT DRIVE THAT IS PRESENT AND THEN  SETS UP
1389                                          ;THE ADDDRESS IN DRIVAD (BITS 13,14,15). THUS THROUGHOUT THE FOLLOWING TESTS
1390                                          ;THE DRIVE TESTED IS THE DRIVE WHOOSE ADDRESS IS IN 'DRIVAD'.
1391                                          ;;*****************************************************************
1392  004766  000004                  TST2:   SCOPE
1393  004770  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
1394  004776  012737  000002  001102          MOV     #2,$TSTNM       ;RESET POINTER TO THIS TEST
1395                                                                  ;NO. CHANGE THIS (2) IN CASE THE
1396                                                                  ;TEST NO. CHANGES
1397  005004  005037  001112                  CLR     $ERTTL          ;CLEAR TOTAL ERROR COUNT
1398  005010  005737  001412                  TST     DRIVS           ;R THERE ANY DRIVES PRESENT?
1399  005014  001002                          BNE     .+6             ;YES, BRANCH
1400  005016  000137  020560          4$:     JMP     $EOP            ;NO, JMP TO THE END
1401  005022  013701  001354                  MOV     DRVPTR,R1       ;GET THAT POINTER TO THE NEXT
1402                                                                  ;DRIVE FLAG
1403  005026  032721  000001          2$:     BIT     #BIT0,(R1)+     ;IS THIS DRIVE PRESENT?
1404  005032  001005                          BNE     1$              ;YES
1405  005034  062737  020000  001350  6$:     ADD     #20000,DRIVAD   ;FORM NXT DRIVE ADRES
1406  005042  001371                          BNE     2$
1407  005044  000764                          BR      4$
1408  005046  005737  001410          1$:     TST     DDPCH           ;PROGRAM LOADED FROM AN RK05 ?
1409  005052  001413                          BEQ     3$              ;NO, BRANCH
1410  005054  013746  001350                  MOV     DRIVAD,-(SP)    ;PUT TEST DRIVE ADDRESS ON THE STACK
```

```
1411  005060  000316                    SWAB    (SP)            ;SETUP TO RIGHT JUSTIFY THE ADDRESS
1412  005062  006216                    ASR     (SP)            ;RIGHT JUSTIFY THE ADDRESS
1413  005064  006216                    ASR     (SP)            ;RIGHT JUSTIFY THE ADDRESS
1414  005066  006216                    ASR     (SP)            ;RIGHT JUSTIFY THE ADDRESS
1415  005070  006216                    ASR     (SP)            ;RIGHT JUSTIFY THE ADDRESS
1416  005072  006216                    ASR     (SP)            ;RIGHT JUSTIFY THE ADDRESS
1417  005074  122637  001410            CMPB    (SP)+,DRVCH     ;PROGRAM LOADED FROM THIS DRIVE ?
1418  005100  001755                    BEQ     6$              ;BR IF YES, DON'T TEST THE DRIVE
1419  005102  010137  001354    3$:     MOV     R1,DRVPTR       ;STORE POINTER TO THE NEXT
1420                                                            ;DRIVE FLAG
1421  005106  104401  001272            TYPE    ,MSG4
1422  005112  013746  001350            MOV     DRIVAD,-(R6)    ;GET THE DRIVE ADDRESS
1423  005116  004737  021106            JSR     PC,SHFRT        ;GO SHIFT IT TO THE RIGHT
1424  005122  005037  001404            CLR     FFLAG
1425  005126  011600                    MOV     (R6),R0         ;DRIVE NUMBER
1426  005130  104403                    TYPOS                   ;GO TYPE THE OCTAL # FOR THE
1427                                                            ;DRIVE THAT IS BEING CHECKED
1428  005132     001     000            .BYTE   1,0
1429  005134  006300                    ASL     R0              ;INDEX TO TABLE
1430  005136  005760  001414            TST     DRIV0(R0)       ;SEE IF F
1431  005142  100006                    BPL     5$              ;NO
1432  005144  104401  005152            TYPE    ,65$            ;;TYPE ASCIZ STRING
1433  005150  000401                    BR      64$             ;;GET OVER THE ASCIZ
1434                            ;;65$:  .ASCIZ  /F/
1435  005154                    64$:
1436  005154  005237  001404            INC     FFLAG           ;SET F FLAG
1437  005160  104401            5$:     TYPE
1438  005162  001213                    $CRLF                   ;TYPE CR, LF
1439                            ;;**************************************************************
1440                            ;*TEST 3        CHECK THAT DRIVE IS SUPPLIED WITH POWER-DPL BIT
1441                            ;;**************************************************************
1442  005164  000004            TST3:   SCOPE
1443  005166  104413                    CNT.RESET               ;GO, DO CONTROL RESET
1444                                                            ;THIS IS A CALL FOR THE 'CNTRL-
1445                                                            ;RESET' ROUTINE. A CONTROL RESET IS
1446                                                            ;ISSUED AND AFTER A CERTAIN TIME
1447                                                            ;IF THE 'CNTRL RDY' DOES NOT SET
1448                                                            ;AN ERROR IS REPORTED. NOTE THAT
1449                                                            ;THE PC IN ERROR MESSAGE IS THE
1450                                                            ;PC WHERE 'CNT.RESET' IS LOCATED.
1451                                                            ;THIS IS A VERY BASIC ERR& IF IT
1452                                                            ;OCCURS GO BACK TO TEST 10
1453  005170  013700  001326            MOV     RKDS,R0
1454  005174  013777  001350  174136    MOV     DRIVAD,@RKDA    ;ADDRESS THE DRIVE UNDER TEST
1455  005202  005710                    TST     @R0             ;CHECK IF ANY BIT OF RKDS IS SET?
1456  005204  001003                    BNE     1$              ;IF SET, BRANCH
1457  005206  011037  001162            MOV     @R0,$REG0       ;GET RKDS
1458  005212  104004                    ERROR   4               ;RKDS ERROR! RKDS IF ADDRESSED
1459                                                            ;CORRECTLY SHOULD BE NON-ZERO
1460  005214  012777  000015  174110    1$:     MOV     #15,@RkCS       ;ISSUE A DRV RESET, IF DRVE
1461                                                            ;POWER IS LO, DPL WILL SET
1462  005222  005001                    CLR     R1
1463  005224  032710  010000            2$:     BIT     #10000,@R0      ;IS 'DPL' BIT SET?
1464  005230  001003                    BNE     3$              ;DPL IS SET, BRANCH
1465  005232  005201                    INC     R1              ;WAIT FOR SOME TIME TO
1466  005234  001373                    BNE     2$              ;SEE IF DPL WOULD SET
```

```
 1467  005236  000403                        BR      4$-2            ;OK, DPL NOT SET
 1468  005240  004737  020710          3$:   JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
 1469  005244  104005                        ERROR   5               ;DPL BIT OF RKDS IS SET. CHECK DRIVE POWER
 1470
 1471
 1472  005246  005001                        CLR     R1
 1473  005250  032710  000100          4$:   BIT     #100,@R0        ;DID R/W/S RDY BIT SET?
 1474  005254  001010                        BNE     TST4            ;;YES, EXIT
 1475  005256  104417  000011                DELAY   .11             ;TIME DELAY
 1476  005262  005201                        INC     R1              ;WAIT FOR R/W/S RDY
 1477  005264  001371                        BNE     4$
 1478  005266  017737  174034  001162        MOV     @RKDS,$REG0     ;GET RKDS
 1479  005274  104016                        ERROR   16              ;R/W/S RDY DID NOT SET AFTER
 1480                                                                ;DRIVE RESET. DRIVE RESET WAS DONE
 1481                                                                ;TO CHECK 'DPL'BIT . THIS TEST
 1482                                                                ;IS NOT FOR CHECKING DRIVE RESET.
 1483                                                                ;U MIGHT WANT TO USE THE TEST PROVIDED
 1484                                                                ;FOR CHECKING DRIVE RESET.
 1485
 1486                                ;;**************************************************************
 1487                                ;*TEST 4        CHECK THAT 'DRIVE UNSAFE' IS CLEAR, 'HDEN' IS SET, 'WPS' IS CLEAR
 1488                                ;;**************************************************************
 1489  005276  000004                TST4:   SCOPE
 1490  005300  104413                        CNT.RESET               ;GO, DO CONTROL RESET
 1491                                                                ;THIS IS A CALL FOR THE 'CNTRL-
 1492                                                                ;RESET' ROUTINE. A CONTROL RESET IS
 1493                                                                ;ISSUED AND AFTER A CERTAIN TIME
 1494                                                                ;IF THE 'CNTRL RDY' DOES NOT SET
 1495                                                                ;AN ERROR IS REPORTED. NOTE THAT
 1496                                                                ;THE PC IN ERROR MESSAGE IS THE
 1497                                                                ;PC WHERE 'CNT.RESET' IS LOCATED.
 1498                                                                ;THIS IS A VERY BASIC ERR & IF IT
 1499                                                                ;OCCURS GO BACK TO TEST 10
 1500  005302  013777  001350  174030        MOV     DRIVAD,@RKDA    ;SET DRIVE ADDRESS
 1501  005310  017700  174012                MOV     @RKDS,R0                ;GET RKDS
 1502  005314  032700  002000                BIT     #2000,R0        ;IS 'DRU' BIT OF RKDS SET?
 1503  005320  001403                        BEQ     1$              ;NO
 1504  005322  004737  020710                JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
 1505  005326  104006                        ERROR   6               ;'DRU' BIT OF RKDS IS SET. CHECK
 1506                                                                ;DRIV BY PUTTING RUN/LOAD SW TO LOAD
 1507                                                                ;THEN BACK TO RUN
 1508  005330  032700  004000          1$:   BIT     #4000,R0        ;IS 'HDEN' BIT SET?
 1509  005334  001004                        BNE     2$              ;YES, BRANCH
 1510  005336  017737  173764  001162        MOV     @RKDS,$REG0     ;GET RKDS
 1511  005344  104007                        ERROR   7               ;ERROR, 'RKDS' BIT IS NOT SET
 1512
 1513  005346  032777  000040  173752  2$:   BIT     #40,@RKDS       ;IS 'WPS' CLEAR?
 1514  005354  001403                        BEQ     TST5            ;;YES, EXIT
 1515  005356  004737  020702                JSR     PC,GT4RG        ;GET RKCS, ER, DS, DA
 1516  005362  104114                        ERROR   114             ;'WPS'-WRITE PROTECT STATUS- BIT OF
 1517                                                                ;OF RKDS SHOULD BE CLEAR. IF THIS DRIVE
 1518                                                                ;IS WRITE ENABLED. CHECK & SEE IF THIS
 1519                                                                ;DRIVE IS WRITE ENABLED, IF IT IS NOT.
 1520                                                                ;WRITE ENABLE IT.
 1521
 1522                                ;
```

```
                                ;;*****************************************************************
                                ;*TEST 5         CHECK THAT 'DRIVE READY' IS SET IN RKDS
                                ;;*****************************************************************
005364  000004                  T5T5:   SCOPE
005366  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
005370  013777  001350  173742          MOV     DRIVAD,@RKDA            ;ADDRS THE DRIVE
005376  105777  173724                  TSTB    @RKDS                   ;IS 'DRY' SET?
005402  100403                          BMI     T5T6                    ;;YES, OK
005404  004737  020702                  JSR     PC,GT4RG                ;GO, GET RKCS, ER, DS, DA
005410  104010                          ERROR   10                      ;'DRY' NOT SET
                                        ;
                                ;;*****************************************************************
                                ;*TEST 6         CHECK THAT 'SOK' BIT CAN SET
                                        ;* THIS TEST CHECKS THAT WITHIN A CERTAIN TIME
                                        ;* 'SOK' BIT CAN SET, IF IT DOES NOT AN ERROR IS REPORTED
                                ;;*****************************************************************
005412  000004                  T5T6:   SCOPE
005414  013777  001350  173716          MOV     DRIVAD,@RKDA            ;ADDRS THE DRIVE
005422  005001                          CLR     R1                      ;INITIALIZE COUNT FOR TIMING WAIT LOOP
005424  032777  000400  173674  1$:     BIT     #400,@RKDS              ;IS SOK SET?
005432  001006                          BNE     T5T7                    ;;EXIT
005434  005201                          INC     R1                      ;NO, WAIT
005436  001372                          BNE     1$                      ;WAITED LONG?
005440  017737  173662  001162          MOV     @RKDS,$REGO             ;GET RKDS
005446  104011                          ERROR   11                      ;WAITED LONG BUT 'SEC OK' BIT DID NOT
                                                                        ;SET
                                        ;
                                ;;*****************************************************************
                                ;*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)
                                ;;*****************************************************************
005450  000004                  T5T7:   SCOPE
```

```
1579  005452  104413                  CNT.RESET               ;GO, DO CONTROL RESET
1580                                                          ;THIS IS A CALL FOR THE 'CNTRL-
1581                                                          ;RESET' ROUTINE. A CONTROL RESET IS
1582                                                          ;ISSUED AND AFTER A CERTAIN TIME
1583                                                          ;IF THE 'CNTRL RDY' DOES NOT SET
1584                                                          ;AN ERROR IS REPORTED. NOTE THAT
1585                                                          ;THE PC IN ERROR MESSAGE IS THE
1586                                                          ;PC WHERE 'CNT.RESET' IS LOCATED.
1587                                                          ;THIS IS A VERY BASIC ERR & IF IT
1588                                                          ;OCCURS GO BACK TO TEST 10
1589  005454  013777  001350  173656      MOV     DRIVAD,@RKDA
1590  005462  013700  001326              MOV     RKDS,R0     ;INITIALIZE
1591  005466  005037  001356              CLR     INDX1       ;'COUNT' - TO TIME 'ERROR 35'
1592  005472  005005                      CLR     R5          ;INITIALIZE 'COUNT' - TO TIME
1593                                                          ;'ERROR 36' (WAIT LOOP)
1594  005474  012704  177764              MOV     #-14,R4     ;INITIALIZE 'COUNT' - FOR THE 12 SECTORS.
1595  005500  012703  000001              MOV     #1,R3       ;R3 CONTAINS THE 'NEXT' COUNT OF SEC-CNTR
1596                                                          ;R1 CONTAINS THE 'PREVIOUS' COUNT OF SEC-CNTR
1597                                                          ;R2 CONTAINS THE 'PRESENT' COUNT OF SEC-CNTR
1598  005504  005037  001360      1$:     CLR     INDX2       ;INITIALIZE 'COUNT' - TO TIME
1599                                                          ;(WAIT LOOP) 'ERROR 34'
1600  005510  005237  001356              INC     INDX1       ;KEEP TIMING FOR 'ERROR 35'
1601  005514  001440                      BEQ     6$          ;BRANCH & REPORT ERROR IF WAITED LONG?
1602  005516  005237  001360      2$:     INC     INDX2       ;KEEP TIMING FOR 'ERROR 34'
1603  005522  001441                      BEQ     7$          ;BRANCH & REPORT ERROR IF WAITED LONG?
1604
1605  005524  011001                      MOV     @R0,R1      ;GET RKDS
1606  005526  032701  000400              BIT     #400,R1     ;IS 'SOK' SET?
1607  005532  001771                      BEQ     2$          ;NO, WAIT FOR IT TO SET
1608  005534  021001                      CMP     @R0,R1      ;MAKE SURE THAT 2 CONSECUTIVE
1609  005536  001362                      BNE     1$          ;READINGS OF SEC-CNTR ARE SAME
1610  005540  042701  177760              BIC     #177760,R1  ;YES, MASK OUT NON-SEC CNTR BITS
1611  005544  001357                      BNE     1$          ;IS IT SECTOR 0, IF NOT LOOP BACK &
1612                                                          ;WAIT FOR SECTOR 0
1613  005546  005204      3$:     INC     R4          ;KEEP TRACK OF SECTORS CHECKED
1614  005550  001447              BEQ     TST10       ;;EXIT IF ALL SECTORS CHKD
1615  005552  005205      4$:     INC     R5          ;KEEP TIMING FOR 'ERROR 36'
1616  005554  001431              BEQ     8$          ;BR & REPORT ERROR IF WAITED LONG
1617  005556  011002              MOV     @R0,R2      ;GET RKDS
1618  005560  032702  000400      BIT     #400,R2     ;IS SOK SET?
1619  005564  001772              BEQ     4$          ;NO, WAIT FOR SOK
1620  005566  021002              CMP     @R0,R2      ;MAKE SURE THAT 2 CONSECUTIVE
1621  005570  001370              BNE     4$          ;READINGS OF SEC-CNTR ARE SAME
1622  005572  042702  177760      BIC     #177760,R2  ;MASK NON-SEC-CNTR BITS
1623  005576  020201              CMP     R2,R1       ;HAS SEC CNTR INCREMENTED?
1624  005600  001764              BEQ     4$          ;NO, WAIT FOR IT TO CHANGE
1625  005602  020203              CMP     R2,R3       ;YES, DID IT INCREMENT CORRECTLY?
1626  005604  001023              BNE     9$          ;NO - REPORT ERROR
1627
1628  005606  005203      5$:     INC     R3          ;INCREMENT "NEXT COUNT"
1629  005610  005201              INC     R1          ;INCREMENT "PREVIOUS COUNT"
1630  005612  005005              CLR     R5          ;INITIALIZE AGAIN FOR TIMING 'ERROR 36'
1631  005614  000754              BR      3$          ;GO & CHECK THE NEXT SECTOR COUNT
1632
1633  005616  010137  001162  6$: MOV     R1,$REGO    ;GET 'SEC CNTR'
1634  005622  104012              ERROR   12          ;WAITED LONG, BUT SECTOR COUNTER
```

```
                                                        ;DID NOT COUNT TO 0
005624  000421                  BR      TST10           ;;EXIT

005626  017737  173474  001162  7$:     MOV     @RKDS,$REG0     ;GET RKDS
005634  104011                  ERROR   11              ;WAITED LONG, BUT 'SOK' BIT DID
                                                        ;NOT SET
005636  000414                  BR      TST10           ;;EXIT

005640  010237  001162          8$:     MOV     R2,$REG0        ;GET SEC CNTR (PRESENT COUNT)
005644  010337  001164          MOV     R3,$REG1        ;GET "NEXT COUNT"
005650  104013                  ERROR   13              ;WAITED LONG, BUT THE SECTOR
                                                        ;COUNTER DID NOT INCREMENT FROM
                                                        ;THE PRESENT COUNT TO THE NEXT COUNT
005652  000406                  BR      TST10           ;;EXIT

005654  010337  001162          9$:     MOV     R3,$REG0        ;GET 'NEXT COUNT' (SEC CNTR SHOULD BE THIS)
005660  010237  001164          MOV     R2,$REG1        ;GET PRESENT COUNT (WHAT SEC CNTR WAS)
005664  104014                  ERROR   14              ;SEC CNTR INCREMENTED WRONG, DID
                                                        ;NOT INCREMENT FROM PRESENT COUNT
                                                        ;TO NEXT COUNT
005666  000747                  BR      5$
                                ;

                        ;;**********************************************************************
                        ;*TEST 10       CHECK THAT SC=SA CAN BE GENERATED
                                ;* THIS TEST CHECKS THAT SC=SA CAN BE GENERATED FOR
                                ;* EVERY SECTOR
                        ;;**********************************************************************
005670  000004          TST10:  SCOPE
005672  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
005674  013704  001350                  MOV     DRIVAD,R4
005700  013700  001326                  MOV     RKDS,R0
005704  012703  177764                  MOV     #-14,R3         ;INITIALIZE COUNT FOR # OF SECTORS
005710  010477  173424          1$:     MOV     R4,@RKDA        ;ADDRESS THE DRIVE
005714  005005                  CLR     R5              ;INITIALIZE COUNT - FOR TIMING ERROR
005716  005205          2$:     INC     R5              ;KEEP TIMING FOR ERROR
005720  001410                  BEQ     3$              ;REPORT ERROR IF WAITED LONG
005722  011001                  MOV     @R0,R1          ;GET RKDS
005724  032701  000020          BIT     #20,R1          ;IS SC=SA SET?
005730  001772                  BEQ     2$              ;NO, WAIT FOR IT
005732  005204          4$:     INC     R4              ;ADDRS THE NEXT SECTOR
005734  005203                  INC     R3              ;ARE ALL SECTORS CHECKED FOR SC=SA
005736  001364                  BNE     1$              ;NO, GO & CHECK NEXT
005740  000406                  BR      TST11           ;;YES, EXIT

005742  110437  001162          3$:     MOVB    R4,$REG0        ;GET SECTOR ADDRESS
005746  010137  001164          MOV     R1,$REG1        ;GET RKDS
```

```
1691  005752  104015                          ERROR   15              ;COULD NOT GET SC=SA FOR THIS
1692                                                                  ;'SECTOR ADDRESS'
1693  005754  000766                          BR      4$              ;GO CHK FOR THE REST
1694                                          ;
1695                                  ;;**************************************************************
1696                                  ;*TEST 11       CHECK THAT 'R/W/S RDY' IS SET & 'SIN' IS CLEAR
1697                                  ;;**************************************************************
1698  005756  000004                  TST11:  SCOPE
1699  005760  104413                          CNT.RESET               ;GO, DO CONTROL RESET
1700  005762  013777  001350  173350          MOV     DRIVAD,@RKDA    ;ADDRESS THE DRIVE
1701  005770  005001                          CLR     R1
1702  005772  017700  173330          1$:     MOV     @RKDS,R0        ;GET RKDS
1703  005776  032700  000100                  BIT     #100,R0         ;IS R/W/S RDY SET?
1704  006002  001005                          BNE     2$              ;YES, BRANCH
1705  006004  005201                          INC     R1              ;WAITED LONG ENOUGH?
1706  006006  001371                          BNE     1$              ;IF NOT LUP BAK & WAIT
1707  006010  010037  001162                  MOV     R0,$REG0        ;GET RKDS
1708  006014  104016                          ERROR   16              ;R/W/S RDY SHOULD BE SET
1709  006016  032700  001000          2$:     BIT     #1000,R0        ;IS SIN CLEAR?
1710  006022  001403                          BEQ     TST12           ;;YES, EXIT
1711  006024  004737  020702                  JSR     PC,GT4RG        ;GET RKCS,ER,DS,DA
1712  006030  104001                          ERROR   1               ;'SIN' SHOULD HAVE BEEN CLEAR
1713                                                                  ;IT WAS NOT CLEAR
1714                                                                  ;NEXT TEST IS GOING TO CHECK
1715                                                                  ;DRIVE RESET, SIN SHOULD BE
1716                                                                  ;CLEARED THEN. IT WILL BE CHECKED
1717                                                                  ;THERE.
1718
1719                                  ;;**************************************************************
1720                                  ;*TEST 12       CHECK 'DRIVE RESET'
1721                                          ;*THIS TEST CHECKS THE VERY BASIC DRIVE RESET LOGIC.
1722                                          ;*SINCE THE HEADS ARE AT CYLINDER 0 (GOING INTO THIS
1723                                          ;*TEST) DRIVE RESET RETRACTS THEM BACK BEYOND CYLINDER 0,
1724                                          ;*AFTER WHICH THEY ARE PUSHED FORWARD TO CYLINDER 0 AGAIN.
1725                                          ;*IN THE LATER PART OF THIS PROGRAM THERE IS A DRIVE RESET
1726                                          ;*TEST WHICH DOES THE RESET FROM LAST CYLINDER.
1727                                  ;;**************************************************************
1728  006032  000004                  TST12:  SCOPE
1729  006034  104413                          CNT.RESET               ;GO, DO CONTROL RESET
1730                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
1731                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
1732                                                                  ;ISSUED AND AFTER A CERTAIN TIME
1733                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
1734                                                                  ;AN ERROR IS REPORTED. NOTE THAT
1735                                                                  ;THE PC IN ERROR MESSAGE IS THE
1736                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
1737                                                                  ;THIS IS A VERY BASIC ERR & IF IT
1738                                                                  ;OCCURS GO BACK TO TEST 10
1739  006036  013700  001332                  MOV     RKCS,R0
1740  006042  005004                          CLR     R4              ;INITIALIZ COUNT - TO TIME ERROR
1741  006044  013777  001350  173266          MOV     DRIVAD,@RKDA    ;ADDRESS THE DRIVE
1742  006052  012710  000015                  MOV     #15,@R0         ;'DRIVE RESET', GO
1743  006056  104412                          CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
1744                                                                  ;IF SO, SKIP THE EROR MESSAGE.
1745  006060  104021                          ERROR   21              ;CNTRL RDY DID NOT SET AFTER
1746                                                                  ;SENDING CYL ADDR TO THE DRIV.
```

```
                                                        ;'ADD ACK' SHOULD HAVE COME BACK
                                                        ;FROM DRIVE, THEREUPON SETTING 'CN RDY'
006062  032777  000100  173236  2$:     BIT     #100,@RKDS      ;DID R/W/S RDY SET?
006070  001005                          BNE     3$              ;YES, BRANCH
006072  005204                          INC     R4              ;WAITED LONG?
006074  001372                          BNE     2$              ;IF NOT LUP BAK & WAIT
006076  004737  020702                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
006102  104026                          ERROR   26              ;R/W/S RDY DID NOT SET AFTER
                                                        ;DRIVE RESET

006104  032777  001000  173214  3$:     BIT     #1000,@RKDS     ;DID SIN SET?
006112  001403                          BEQ     5$              ;NO, BRANCH
006114  004737  020702                  JSR     PC,GT4RG        ;GO, GET RKCS,ER,DS,DA
006120  104001                          ERROR   1               ;SIN SET, AFTER A
                                                        ;DRIVE RESET.
006122  032710  140000          5$:     BIT     #140000,@R0     ;WAS 'ERR' BIT OR 'HE' BIT SET?
006126  001403                          BEQ     4$              ;NO
006130  004737  020702                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
006134  104022                          ERROR   22              ;'ERR' OR 'HE' BIT SET WHILE DOING
                                                        ;DRIVE RESET
006136  022710  000214          4$:     CMP     #214,@R0        ;DOES RKCS STILL CONTAIN THE
                                                        ;'DRIV RES' BITS
006142  001406                          BEQ     TST13           ;;YES, EXIT
006144  012737  000214  001162          MOV     #214,$REG0      ;GET EXPCTD RKCS
006152  011037  001164                  MOV     @R0,$REG1       ;GET RKCS, RECVD
006156  104024                          ERROR   24              ;NO - RKCS SHOULD CONTAIN THE 'DRIV RES'
                                                        ;FUNCTION, ERROR IF DIFFERENT.

                        ;;**************************************************************
                        ;*TEST 13       CHECK 'SEEK' TO CYLINDER 0
                                ;* THIS TEST CHECKS THE SEEK LOGIC DOING SEEK TO CYLINDER 0.
                                ;* NOTE THAT SINCE THE HEADS ARE ALREADY ON CYLINDER 0, NO
                                ;* HEAD MOVEMENT IS INVOLVEDN AND THE STRESS IS ON THE BASIC SEEK
                                ;* LOGIC.
                        ;;**************************************************************
006160  000004          TST13:  SCOPE
006162  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
006164  104421                  TST.SIN                         ;GO CHECK IF SIN SET. IF SET
                                                        ;A DO DRIVE RESET TO CLEAR IT
006166  013700  001332          MOV     RKCS,R0
006172  013777  001350  173140  MOV     DRIVAD,@RKDA    ;ADDRESS THE DRIVE

006200  012710  000011          MOV     #11,@R0         ;'SEEK' GO
006204  104412                  CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
                                                        ;IF SO, SKIP THE EROR MESSAGE.
006206  104021                  ERROR   21              ;'CNTRL RDY' DID NOT SET AFTER SENDING
                                                        ;CYL ADDR TO THE DRIVE, 'ADD ACK'
```

```
                                                        ;SHOULD HAVE COME BACK FROM THE
                                                        ;DRIVE, THEREUPON SETTING 'CNTRL RDY'
006210  005005                  2$:     CLR     R5
006212  032777  000100  173106          BIT     #100,@RKDS      ;DID R/W/S RDY BIT SET?
006220  001005                          BNE     3$              ;YES, BRANCH
006222  005205                          INC     R5              ;WAITED LONG ENOUGH?
006224  001372                          BNE     2$+2            ;IF NOT, LUP BAK & WAIT
006226  004737  020702                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
006232  104026                          ERROR   26              ;R/W/S RDY DID NOT SET AFTER SEEK
006234  032777  001000  173064  3$:     BIT     #1000,@RKDS     ;DID SIN SET?
006242  001403                          BEQ     6$              ;NO, BRANCH
006244  004737  020702                  JSR     PC,GT4RG        ;GO, GET RKCS,ER,DS,DA
006250  104001                          ERROR   1               ;SIN SET ON DOING SEEK
                                                        ;TO CYL 0 NOTE THIS IS THE
                                                        ;FIRST TIME THE HEADS HAVE
                                                        ;BEEN MOVED

006252  032710  140000          6$:     BIT     #140000,@R0     ;WAS 'ERR' OR 'HE' BIT SET?
006256  001403                          BEQ     4$

006260  004737  020702                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
006264  104022                          ERROR   22              ;'ERR' OR 'HE' BIT SET WHILE DOING 'SEEK'

006266  005777  173036          4$:     TST     @RKER           ;WAS ANY BIT IN RKER SET?
006272  001403                          BEQ     5$              ;NO
006274  004737  020710                  JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
006300  104023                          ERROR   23              ;RKER SHOWS AN ERROR BIT, CHECK

006302  022710  000210          5$:     CMP     #210,@R0        ;DOES RKCS STILL CONTAIN 'SEEK' FUNCTION
006306  001406                          BEQ     TST14           ;;YES, EXIT
006310  012737  000210  001162          MOV     #210,$REG0      ;GET EXPCTD RKCS
006316  011037  001164                  MOV     @R0,$REG1       ;GET RKCS RECVD
006322  104024                          ERROR   24              ;NO, RKCS SHOULD BE STILL CONTAINING
                                                        ;'SEEK' FUNCTION ERROR - IF IT CHANGED
                                        ;
                                ;;**************************************************************
                                ;*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 (RK05) GENERATES THIS SIGNAL, ABSENCE OF
                                        ;*R/W/S RDY-CLEAR COULD MEAN A FAULT ON M7702
                                        ;*WHERE 'MOVE L' IS GENERATED.
                                        ;*NOTE THIS IC THE FIRST TIME HEADS ARE MADE TO MOVE BY SEEKING
                                        ;*TO CYLINDER 2.
                                ;;**************************************************************
006324  000004                  TST14:  SCOPE
006326  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
```

```
1859  006330  104421                          TST.SIN                 ;GO CHECK IF SIN IS SET
1860                                                                  ;IF SET DO DRV-RESET TO CLR IT
1861  006332  004737  021412                  JSR     PC,DRESET       ;MAKE SURE HEADS R ON CYL 0
1862  006336  104026                          ERROR   26              ;R/W/S RDY DIDN'T SET
1863                                                                  ;AFTER THE ABOVE DRV RESET
1864  006340  005005                          CLR     R5
1865  006342  013777  001350  172770          MOV     DRIVAD,@RKDA
1866  006350  052777  000100  172762          BIS     #100,@RKDA      ;SEEK CYLINDER 2
1867  006356  013701  001326                  MOV     RKDS,R1
1868  006362  012777  000011  172742          MOV     #11,@RKCS       ;SEEK, GO
1869  006370  032711  000100          1$:     BIT     #100,@R1        ;DID R/W/S RDY CLR?
1870  006374  001405                          BEQ     2$              ;YES, BRANCH
1871  006376  005205                          INC     R5
1872  006400  100373                          BPL     1$
1873  006402  004737  020710                  JSR     PC,GT3RG
1874  006406  104025                          ERROR   25              ;R/W/S RDY WAS NOT CLEAR WHEN HEADS
1875                                                                  ;WERE SEEKING TO CYLINDER 2
1876
1877  006410  004737  021344          2$:     JSR     PC,TSTRWS       ;GO, WAIT FOR R/W/S RDY TO SET
1878  006414  104016                          ERROR   16              ;R/W/S RDY DID NOT SET AFTER SEEK
1879                                                                  ;WAS TRIED TO CYLINDER 2 (ABOVE..
1880                                                                  ;NOTE THIS WAS THE FIRST TIME A SEEK
1881                                                                  ;WAS TRIED TO A CYLINDER OTHER THAN
1882                                                                  ;0.
1883
1884
1885
1886
1887                    ;;**************************************************************
1888                    ;*TEST 15       CHECK 'WRITE' FORMAT FUNCTION-CYLINDER 0, SECTOR 0
1889                    ;*THIS TEST CHECKS THE LOGIC INVOLVED IN THE WRITE FMT
1890                    ;*FUNCTION. ON ISSUING A WRT FMT, THE FOLLOWING IS CHECKED
1891                    ;*1)  CNTRL RDY WAS CLEARED AS GO WAS SET.
1892                    ;*2)  CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION OF FUNCTION
1893                    ;*3)  IF 'HE' OR 'ERR' BIT SET?
1894                    ;*4)  IF RKDA INCREMENTED CORRECTLY FROM 0 TO 1?
1895                    ;*5)  IF RKWC OVERFLOWED CORRECTLY TO 0?
1896                    ;*6)  IF RKBA INCREMENTED CORRECTLY BY 2?
1897                    ;*7)  IF ANY BIT IN RKER SET?
1898                    ;*8)  IF THE 'WRT FMT' FUNCTION BITS ARE STILL IN THE RKCS?
1899                    ;*NOTE THAT ONE WORD '125252' WAS WRITTEN ON SECTOR
1900                    ;*0 & IT WILL BE CHECKED IN THE NEXT TESTS.
1901                    ;;**************************************************************
1902  006416  000004  TST15:  SCOPE
1903  006420  104413          CNT.RESET               ;GO, DO CONTROL RESET
1904                                                  ;THIS IS A CALL FOR THE 'CNTRL-
1905                                                  ;RESET' ROUTINE. A CONTROL RESET IS
1906                                                  ;ISSUED AND AFTER A CERTAIN TIME
1907                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
1908                                                  ;AN ERROR IS REPORTED. NOTE THAT
1909                                                  ;THE PC IN ERROR MESSAGE IS THE
1910                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
1911                                                  ;THIS IS A VERY BASIC ERR S IF IT
1912                                                  ;OCCURS GO BACK TO TEST 10
1913  006422  104421          TST.SIN                 ;GO CHECK IF SIN IS SET
1914                                                  ;IF SET, DO DRIVE RESET TO CLR IT
```

```
1915  006424  012703  033240          MOV     #OUTBUF,R3
1916                                                    ;THIS CODE SETS UP A 256 WORD BUFFER
1917                                                    ;WHICH WILL BE USED TO WRITE 1 SECTOR
1918                                                    ;ON THE DISK
1919                                                    ;1ST WORD       000001
1920                                                    ;2ND WORD       177777  2'S COMPLEMENT
1921                                                    ;3RD WORD       000002  OF ABOVE
1922                                                    ;4TH WORD       177776
1923                                                    ;
1924                                                    ;253RD WORD     000177
1925                                                    ;254TH WORD     177601
1926                                                    ;255TH WORD     000000
1927                                                    ;256TH WORD     125252
1928
1929  006430  012700  000001          MOV     #1,R0           ;SET COUNT
1930
1931  006434  010023          95:     MOV     R0,(R3)+        ;SET UP DATA WORDS
1932  006436  010013                  MOV     R0,(R3)
1933  006440  005423                  NEG     (R3)+
1934  006442  005200                  INC     R0
1935  006444  022700  000200          CMP     #200,R0         ;DONE?
1936  006450  001371                  BNE     9$
1937  006452  005023                  CLR     (R3)+           ;SET 255TH WORD TO 0
1938  006454  012713  125252          MOV     #125252,@R3     ;SET 256TH WORD
1939
1940  006460  012703  033240          MOV     #OUTBUF,R3      ;RESET POINTER TO OUTBUF
1941  006464  013701  001332          MOV     RKCS,R1
1942  006470  013702  001336          MOV     RKBA,R2
1943  006474  010312                  MOV     R3,@R2          ;FROM HERE-SET UP CURRENT ADDRESS
1944  006476  012777  177400  172630  MOV     #-400,@RKWC     ;SET UP WORD COUNT 400 WORDS
1945  006504  013777  001350  172626  MOV     DRIVAD,@RKDA    ;SET UP DISK ADDRS, SECTOR 0, CYLINDER 0
1946  006512  012711  002003          MOV     #2003,@R1       ;WRITE FORMAT, GO
1947
1948  006516  105711          1$:     TSTB    @R1             ;WAS 'CNTRL RDY' CLEARED AS GO WAS SET?
1949  006520  100003                  BPL     2$              ;YES, BRANCH
1950  006522  004737  020710          JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
1951  006526  104030                  ERROR   30              ;'CNTRL RDY' DIDN'T CLEAR AS GO
1952                                                    ;WAS SET TO 'WRITE FORMAT'
1953  006530  005000          2$:     CLR     R0
1954  006532  105711                  TSTB    @R1             ;WAS 'CNTRL RDY' SET ON COMPLETION OF WRITE?
1955  006534  100411                  BMI     3$              ;YES, BRANCH
1956  006536  005200                  INC     R0              ;NO, HAVE U WAITED LONG ENOUGH?
1957  006540  001374                  BNE     2$+2            ;IF NOT, LOOP BACK & WAIT
1958                                                    ;IF YES, REPORT ERROR
1959  006542  004737  020702          JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
1960  006546  013737  001350  001202  MOV     DRIVAD,$REG1D
1961  006554  104416                  BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
1962                                                    ;$REG1D INTO DR #,CYL,SUR,SEC BITS
1963  006556  104031                  ERROR   31              ;'CNTRL RDY' DIDN'T SET ON COMPLETION
1964                                                    ;OF WRITE FORMAT
1965                                                    ;WRT FMT WAS DONE STARTING AT <DSK-ADRES>
1966                                                    ;INDICATED IN EROR MSGE.
1967  006560  004737  021142          3$:     JSR     PC,CHKHE        ;GO CHECK IF 'HE' OR 'ERR' BIT SET.
1968                                                    ;IF YES, SAVE RKCS, ER, DS, DA.
1969                                                    ;RETURN HERE IF ERROR.
1970  006564  104032                  ERROR   32              ;'HE' OR 'ERR' BIT SET WHILE DOING
```

```
                                                        ;A WRITE FORMAT
                                                        ;WRT FMT WAS DONE STARTING AT 'DSK-ADRES
                                                        ;INDICATED IN ERROR MSGE.
006566  004737  021170        4$:     JSR     PC,CHKDA    ;GO CHECK IF RKDA INCREMENTED CORRECTLY
                                                        ;IF NOT, RETURN HERE.
006572  104033                        ERROR   33          ;RKDA SHOULD HAVE INCREMENTED BY
                                                        ;1 SECTOR, IT DID NOT
006574  004737  021224        5$:     JSR     PC,CHKWC    ;CHECK IF WORD COUNT OVERFLOWED, IF
                                                        ;NOT RETURN HERE.
006600  104034                        ERROR   34          ;RKWC DID NOT OVERFLOW TO 0, AFTER
                                                        ;XFER ON WRITE FORMAT
006602  022712  034240        6$:     CMP     #OUTBUF+1000,@R2    ;DID RKBA INCREMENT CORRECTLY?
006606  001406                        BEQ     7$          ;YES, BRANCH
006610  012737  034240  001162        MOV     #OUTBUF+1000,$REGO  ;GET EXPCTD RKBA
006616  011237  001164                MOV     @R2,$REG1   ;GET ACTUAL RKBA
006622  104035                        ERROR   35          ;RKBA DIDN'T INCREMENT BY 1000 AFTER
                                                        ;WRITE FORMAT OF 400 WORDS
006624  004737  021250        7$:     JSR     PC,CHKER    ;CHECK IOF ANY BIT IN RKER SET,
                                                        ;IF YES RETURN HERE.
006630  104036                        ERROR   36          ;RKER BIT SET ON DOING 1 WORD
                                                        ;WRITE FORMAT
006632  022711  002202        8$:     CMP     #2202,@R1   ;DOES RKCS STILL HAVE 'WRT FMT' BITS?
006636  001406                        BEQ     TST16       ;YES, EXIT
006640  012737  002202  001162        MOV     #2202,$REGO ;GET EXPCTD RKCS
006646  011137  001164                MOV     @R1,$REG1   ;GET ACTUAL RKCS
006652  104024                        ERROR   24          ;RKCS DIDN'T CONTAIN 'WRT FMT' BITS
                                                        ;AFTER THE FUNCTION WAS COMPLETED
                                ;
                        ;;*****************************************************************
                        ;*TEST 16       CHECK 'READ 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 'NE' OR 'ERR' BIT SET?
                                ;*4)  IF RKDA INCREMENTED CORRECTLY FROM 0 TO 1?
                                ;*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?
                        ;;*****************************************************************
000004  TST16:  SCOPE
005000          CLR     R0
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
006662  104421                      TST.SIN           ;GO CHECK IF SIN IS SET
                                                        ;IF SET, DO DRIVE RESET TO CLR IT
006664  013701  001332              MOV     RKCS,R1
006670  013702  001336              MOV     RKBA,R2
006674  012703  033240              MOV     #OUTBUF,R3
006700  010312                      MOV     R3,@R2    ;SETUP ADRS WHERE HEADER WORD IS TO BE
                                                        ;X-FERRED
006702  012777  177777  172424      MOV     #-1,@RKWC ;SET UP WORD COUNT
006710  013777  001350  172422      MOV     DRIVAD,@RKDA ;SET UP DISK ADRS, SECTOR 0, CYLINDER 0
006716  012711  002005              MOV     #2005,@R1 ;READ FORMAT, GO

006722  105711          1$:         TSTB    @R1       ;WAS 'CNTRL RDY' CLEARED AS GO WAS SET?
006724  100003                      BPL     2$        ;YES, BRANCH
006726  004737  020710              JSR     PC,GT3RG  ;GO, GET RKCS, RKER
006732  104030                      ERROR   30        ;CNTRL RDY DIDN'T CLEAR AS GO WAS
                                                        ;SET TO 'READ FORMAT'
006734  005000          2$:         CLR     R0
006736  105711                      TSTB    @R1       ;WAS 'CNTRL RDY' SET ON COMPLETION OF
                                                        ;TRANSFER
006740  100411                      BMI     3$        ;YES, BRANCH
006742  005200                      INC     R0        ;NO, HAVE U WAITED LONG ENOUGH?
006744  001374                      BNE     2$+2      ;IF NOT, LOOP BACK $ WAIT
                                                        ;IF YES, REPORT ERROR
006746  004737  020702              JSR     PC,GT4RG  ;GO, GET RKCS, ER, DS,DA
006752  013737  001350  001202      MOV     DRIVAD,$REG1D
006760  104416                      BRKDA4            ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                        ;$REG1D INTO DR #,CYL,SUR,SEC BITS
006762  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
006764  004737  021142      3$:     JSR     PC,CHKHE  ;CHECK IF 'ERR' OR 'HE' BIT SET, IF
                                                        ;YES RETURN HERE.
006770  104046                      ERROR   46        ;'HE' OR 'ERR' BIT SET WHILE
                                                        ;DOING A 'READ FORMAT'
                                                        ;READ FMT WAS DONE STARTING AT <DSK-ADRES>
                                                        ;INDICATED IN EROR MESGE
006772  004737  021170      4$:     JSR     PC,CHKDA  ;CHECK IF RKDA INCREMENTED CORRECTLY
                                                        ;IF NOT, RETURN HERE.
006776  104040                      ERROR   40        ;RKDA SHOULD HAVE INCREMENTED
                                                        ;BY 1 SECTOR, IT DID NOT

007000  004737  021224      5$:     JSR     PC,CHKWC  ;CHECK IF RKWC OVERFLOWED TO 0. IF
                                                        ;NOT RETURN HERE.
007004  104041                      ERROR   41        ;RKWC DID NOT OVERFLOW TO 0
                                                        ;AFTER XFER ON READ FORMAT
007006  022712  033242      6$:     CMP     #OUTBUF+2,@R2 ;DID RKBA INCREMENT TO NXT WORD ADDRS?
007012  001406                      BEQ     7$        ;YES, BRANCH
007014  012737  033242  001162      MOV     #OUTBUF+2,$REG0 ;GET EXPCTD RKBA
007022  011237  001164              MOV     @R2,$REG1 ;GET ACTUAL RKBA
007026  104042                      ERROR   42        ;RKBA DIDN'T INCREMENT BY 2 AFTER
```

```
                                                ;'READ FORMAT' OF 1 WORD
007030  004737  021250          7$:   JSR   PC,CHKER      ;CHECK IF ANY BIT IN RKER SET. IF
                                                ;YES RETURN HERE.
007034  104036                      ERROR 36            ;RKER BIT SET ON DOING
                                                ;1 WORD READ FORMAT
007036  005713                8$:   TST   @R3           ;DOES OUTBUF CONTAIN THE HEADER
                                                ;WORD-0
007040  001407                      BEQ   9$            ;YES, BRANCH
007042  005037  001162              CLR   $REG0         ;GET SECTOR NO.
007046  005037  001164              CLR   $REG1         ;EXPCTD HEADER
007052  011337  001166              MOV   @R3,$REG2     ;GET HEADER RECVD
007056  104043                      ERROR 43            ;CORRECT HEADER WORD-0-WAS
                                                ;NOT RECEIVED ON READ FORMAT
007060  022711  002204          9$:   CMP   #2204,@R1     ;DOES RKCS HAVE THE 'RDFMT' BITS?
007064  001406                      BEQ   TST17         ;YES, BRANCH
007066  012737  002204  001162      MOV   #2204,$REG0   ;GET EXPCTD RKCS
007074  011137  001164              MOV   @R1,$REG1     ;GET ACTUAL RKCS
007100  104024                      ERROR 24            ;RKCS DIDN'T CONTAIN 'RD FMT'
                                                ;BITS AFTER FUNCTION WAS
                                                ;COMPLETED


;;**************************************************************
;*TEST 17        CHECK 'READ' FUNCTION-CYLINDER 0,SECTOR 0
        ;*THIS IS THE FIRST TIME A PURE READ IS PREFORMED IN THIS
        ;*TEST SEQUENCE.  THE FOLLOWING IS CHECKED
        ;*1) CNTRL RDY CLEARS AS GO IS SET
        ;*2) CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
        ;*OF FUNCTION
        ;*3) IF 'NE' OR 'ERR' BIT SET?
        ;*4) IF RKDA INCREMENTED CORRECTLY?
        ;*5) IF RKWC OVERFLOWED TO 0?
        ;*6) IF RKBA INCREMENTED CORRECTLY?
        ;*7) IF ANY RKER BIT SET?
        ;*8) IF THE CORRECT PSUEDO-HEADER (FIRST WORD) WAS
        ;*READ FROM SECTOR 0
        ;*9) IF THE 'READ' FUNCTION BITS ARE STILL IN RKCS
;;**************************************************************
007102  000004          TST17:  SCOPE
007104  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
007106  104421                  TST.SIN                 ;GO CHECK IF SIN IS SET
                                                ;IF SET, DO DRIVE RESET TO CLR IT
007110  013701  001332          MOV   RKCS,R1
007114  005000                  CLR   R0
007116  013702  001336          MOV   RKBA,R2
007122  012703  033240          MOV   #OUTBUF,R3
```

```
2139  007126  010312                          MOV     R3,@R2          ;SET UP ADDRS WHERE DATA WORD IS
2140                                                                  ;TO BE X-FERRED
2141  007130  012777  177400  172176          MOV     #-400,@RKWC     ;SET UP WORD COUNT
2142  007136  013777  001350  172174          MOV     DRIVAD,@RKDA    ;SET UP DISK ADRS. SECTOR 0, CYLINDER 5
2143  007144  012711  000005                  MOV     #5,@R1          ;READ, GO
2144
2145  007150  105711                  1$:     TSTB    @R1             ;WAS 'CNTRL RDY' CLEARED AS GO WAS SET?
2146  007152  100003                          BPL     2$              ;YES, BRANCH
2147  007154  004737  020710                  JSR     PC,GT3RG        ;GO, GET RKCS, ER
2148  007160  104030                          ERROR   30              ;CNTRL RDY DID NOT CLEAR AS GO
2149                                                                  ;WAS SET TO 'READ'
2150  007162  005000                  2$:     CLR     R0
2151  007164  105711                          TSTB    @R1             ;WAS CNTRL RDY SET ON COMPLETION
2152                                                                  ;OF TRANSFER?
2153  007166  100411                          BMI     3$              ;YES, BRANCH
2154  007170  005200                          INC     R0              ;NO, HAVE U WAITED LONG ENOUGH?
2155  007172  001374                          BNE     2$+2            ;IF NOT, LOOP BACK & WAIT
2156                                                                  ;IF YES, REPORT ERROR
2157  007174  004737  020702                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
2158  007200  013737  001350  001202          MOV     DRIVAD,$REG10
2159  007206  104416                          BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
2160                                                                  ;$REG10 INTO DR #,CYL,SUR,SEC BITS
2161  007210  104045                          ERROR   45              ;CNTRL RDY DID NOT SET ON
2162                                                                  ;COMPLETION OF READ
2163                                                                  ;READ WAS DONE STARTING AT <DSK-ADRES>
2164                                                                  ;INDICATED IN EROR MESGE
2165
2166  007212  004737  021142          3$:     JSR     PC,CHKHE        ;CHECK IF 'ERR' OR 'HE' BIT IS SET
2167                                                                  ;IF YES, RETURN HERE.
2168  007216  104046                          ERROR   46              ;'HE' OR 'ERR' BIT SET WHILE
2169                                                                  ;DOING A READ.
2170                                                                  ;READ WAS DONE STARTING AT <DSK-ADRES>
2171                                                                  ;INDICATED IN EROR MESGE
2172  007220  004737  021170          4$:     JSR     PC,CHKDA        ;CHECK IF RKDA INCREMENTED CORRECTLY,
2173                                                                  ;IF NOT RETURN HERE.
2174  007224  104040                          ERROR   40              ;RKDA DID NOT INCREMENT
2175                                                                  ;BY 1 (SECTOR)
2176  007226  004737  021224          5$:     JSR     PC,CHKWC        ;CHECK IF RKWC OVERFLOWED TO 0,
2177                                                                  ;IF NOT RETURN HERE.
2178  007232  104041                          ERROR   41              ;RKWC DID NOT OVERFLOW TO 0,
2179                                                                  ;AFTER X-FER ON READ
2180  007234  022712  034240          6$:     CMP     #OUTBUF+1000,@R2        ;DID RKBA INCREMENT CORRECTLY?
2181  007240  001406                          BEQ     7$              ;YES, BRANCH
2182  007242  012737  034240  001162          MOV     #OUTBUF+1000,$REG0      ;GET EXPCTD RKBA
2183  007250  011237  001164                  MOV     @R2,$REG1       ;GET ACTUAL RKBA
2184  007254  104042                          ERROR   42              ;RKBA DID NOT INCREMENT BY 2
2185                                                                  ;AFTER 'READ' OF 1 WORD
2186  007256  004737  021250          7$:     JSR     PC,CHKER        ;CHECK IF ANY BIT IN RKER SET,
2187                                                                  ;IF YES RETURN HERE.
2188  007262  104036                          ERROR   36              ;RKER BIT SET ON DOING 1
2189                                                                  ;WORD 'READ'
2190  007264  022713  000001          8$:     CMP     #1,@R3          ;DOES OUTBUF CONTAIN THE RIGHT
2191                                                                  ;DATA WORD
2192  007270  001411                          BEQ     9$              ;YES BRANCH
2193  007272  012737  000001  001162          MOV     #1,$REG0        ;GET EXPCTD DATA WORD
2194  007300  011337  001164                  MOV     (R3),$REG1      ;GET RECVD DATA WORD
```

```
2195  007304  013737  001350  001166          MOV     DRIVAD,$REG2    ;GET DISK ADRS FROM WHICH READ WAS DONE
2196  007312  104044                          ERROR   44              ;DID NOT READ THE CORRECT
2197                                                                  ;DATA WORD--FROM DISK ADRES,
2198
2199                                                                  ;SEC 0, CYL 0, SUR 0
2200
2201                                                                  ;AFTER 1 SECTOR READ RKDB CONTAINS
2202                                                                  ;FOR RK11C
2203                                                                  ;THE CHECKSUM FOR THAT SECTOR
2204                                                                  ;FOR RK11D
2205                                                                  ;THE LAST WORD TRANSFERRED TO MEMORY
2206                                                                  ;
2207                                                                  ;IT SO HAPPENS THAT WITH THE SECTOR
2208                                                                  ;THAT WAS READ, RKDB CONTAINS THE
2209                                                                  ;SAME INFORMATION FOR BOTH RK11C
2210                                                                  ;AND RK11D
2211  007314  022777  125252  172020  9$:     CMP     #125252,@RKDB   ;DOES RKDB CONTAIN THE EXPCTD WORD?
2212  007322  001407                          BEQ     10$             ;YES, BRANCH
2213  007324  012737  125252  001162          MOV     #125252,$REG0   ;GET EXPCTD RKDB
2214  007332  017737  172004  001164          MOV     @RKDB,$REG1     ;GET RECVD RKDB
2215  007340  104037                          ERROR   37              ;RKDB DOESD NOT CONTAIN THE
2216                                                                  ;EXPCTD WORD AFTER A READ OF SEC 0
2217                                                                  ;CYL 0
2218  007342  022711  000204          10$:    CMP     #204,@R1        ;DOES RKCS HAVE THE 'READ' BITS?
2219  007346  001406                          BEQ     11$             ;YES   BRANCH
2220  007350  012737  000204  001162          MOV     #204,$REG0      ;GET EXPCTD RKCS
2221  007356  011137  001164                  MOV     @R1,$REG1       ;GET RECVD RKCS
2222  007362  104024                          ERROR   24              ;RKCS DID NOT CONTAIN 'READ'
2223                                                                  ;FUNCTION BITS AFTER OPERATION
2224                                                                  ;WAS COMPLETED
2225  007364  104413                  11$:    CNT.RESET               ;GO DO CONTROL RESET
2226  007366  005777  171750                  TST     @RKDB           ;DID CONTROL RESET CLEAR RKDB?
2227  007372  001407                          BEQ     TST20           ;;YES, EXIT
2228  007374  013737  001342  001164          MOV     RKDB,$REG1      ;GET ADRES OF RKDB
2229  007402  017737  171734  001164          MOV     @RKDB,$REG1     ;GET CONTENTS OF RKDB
2230  007410  104102                          ERROR   102             ;CONTROL RESET DIDN'T CLR RKDB
2231
2232                                  ;;**************************************************************
2233                                  ;*TEST 20       CHECK 'WRITE FORMAT' -CYLINDER 0, SECTOR 0-13
2234                                          ;*THIS TEST GOES ONE STEP FURTHER & PERFORMS A WRT
2235                                          ;*FMT ON CYLINDER 0 & CHECKS THE FOLLOWING
2236                                          ;*1)  IF CNTRL RDY SET WITHIN A CERTAIN TIME ON COMPLETION
2237                                          ;*OF THE FUNCTION
2238                                          ;*2)  IF 'NE' OR 'ERR' BIT SET?
2239                                          ;*3)  IF THE RKDA INCREMENTS CORRECTLY?
2240                                          ;*4)  IF THE RKDB IS CLEAR?
2241                                          ;*WRT FMT IS DONE ONE SECTOR AT A TIME
2242                                          ;*THE FIRST WORD OF EVERY SECTOR IS WRITTEN AS A
2243                                          ;*PSUEDO-HEADER CONSISTING OF DRIVE #, CYLINDER #, SURFACE
2244                                          ;*& SECTOR #. THIS WILL BE READ & CHECKED IN THE FOLLOWING TEST.
2245                                  ;;**************************************************************
2246  007412  000004                  TST20:  SCOPE
2247  007414  013703  001332                  MOV     RKCS,R3
2248  007420  012702  177764                  MOV     #-14,R2         ;SET UP COUNT FOR 12 SECTORS
2249  007424  013704  001340                  MOV     RKDA,R4
2250  007430  013701  001350                  MOV     DRIVAD,R1       ;GET DRIVE ADDRESS
```

```
2251  007434  010105                      MOV     R1,R5           ;STORE IT
2252  007436  005205                      INC     R5
2253  007440  012737  007446  001110      MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
2254                                                              ;ON      ERROR (SW 9)
2255  007446  104413              1$:     CNT.RESET               ;GO, DO CONTROL RESET
2256                                                              ;THIS IS A CALL FOR THE 'CNTRL-
2257                                                              ;RESET' ROUTINE. A CONTROL RESET IS
2258                                                              ;ISSUED AND AFTER A CERTAIN TIME
2259                                                              ;IF THE 'CNTRL RDY' DOES NOT SET
2260                                                              ;AN ERROR IS REPORTED. NOTE THAT
2261                                                              ;THE PC IN ERROR MESSAGE IS THE
2262                                                              ;PC WHERE 'CNT.RESET' IS LOCATED.
2263                                                              ;THIS IS A VERY BASIC ERR & IF IT
2264                                                              ;OCCURS GO BACK TO TEST 10
2265  007450  104421                      TST.SIN                 ;GO CHECK IF SIN IS SET
2266                                                              ;IF SET, DO DRIVE RESET TO CLR IT
2267  007452  005000                      CLR     R0
2268  007454  010137  033240              MOV     R1,OUTBJF       ;THIS WORD TO BE X-FERRED. FIRST
2269                                                              ;WORD OF EACH SECTOR WILL BE THE
2270                                                              ;ACTUAL DRIVE-ADDRS CONSISTING OF
2271                                                              ;DRIVE NO, CYL ADDRS, SURFACE
2272                                                              ;SECTOR NO.
2273  007460  012777  033240  171650      MOV     #OUTBUF,@RKBA   ;ADRS FROM WHICH DATA WORD IS TO
2274                                                              ;X-FERRED
2275  007466  012777  177777  171640      MOV     #-1,@RKWC       ;SET UP WORD COUNT
2276  007474  010114                      MOV     R1,@R4  ;ADDRS THE DRIVE, CYL 0, & CORRECT SECTOR
2277  007476  012713  002003              MOV     #2003,@R3       ;WRITE FORMAT, GO
2278
2279  007502  105777  171624      2$:     TSTB    @RKCS           ;DID 'CNTRL RDY' SET?
2280  007506  100410                      BMI     3$              ;YES, BRANCH
2281  007510  005200                      INC     R0              ;NO, HAVE U WAITED LONG?
2282  007512  001373                      BNE     2$              ;IF NOT, LOOP BACK & WAIT
2283                                                              ;IF YES, REPORT ERROR
2284  007514  004737  020702              JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
2285  007520  010137  001202              MOV     R1,$REG10       ;GET DISK ADRS (UNIT,CYL,SUR,SEC) TO WHICH
2286                                                              ;WRITE FORMAT WAS DONE
2287  007524  104416                      BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
2288                                                              ;$REG10 INTO DR #,CYL,SUR,SEC BITS
2289  007526  104031                      ERROR   31              ;'CNTRL RDY' DID NOT SET ON COMPLETION
2290                                                              ;OF 'WRITE FORMAT'
2291                                                              ;WRT FMT WAS DONE STARTING AT <DSK-ADRES>
2292                                                              ;INDICATED IN EROR MSGE.
2293  007530  004737  021134      3$:     JSR     PC,CHKHE1       ;CHECK IF 'ERR' OR 'HE' BIT IS SET.
2294                                                              ;IF YES RETURN HERE.
2295  007534  104032                      ERROR   32              ;'HE' OR 'ERR' BIT SET WHILE DOING
2296                                                              ;WRITE FORMAT ON CYLINDER 0,
2297                                                              ;SECTOR IN ERROR IS AS SHOWN IN
2298                                                              ;DISK-ADRES BITS 0-3
2299                                                              ;WRT FMT WAS DONE STARTING AT <DSK-ADRES>
2300                                                              ;INDICATED IN EROR MSGE.
2301
2302  007536  004737  021176      4$:     JSR     PC,CHKDA1       ;CHECK IF RKDA INCREMENTED CORRECTLY?
2303
2304  007542  104033                      ERROR   33              ;RKDA DID NOT INCREMENT CORRECT
2305                                                              ;AFTER 1 WORD 'WRITE FORMAT' ON
2306                                                              ;CYLINDER 0, SECTOR IN ERROR IS 1
```

```
                                                        ;LESS THAN THAT SHOWN IN EXPCTD RKDA
007544  005777  171572          5$:     TST     @RKDB   ;CHECK THAT RKDB DOES CONTAIN A 0
                                        ;AFTER WRT BECAUSE LAST WORD WRITTEN
                                        ;WAS SERIALLY SHIFTED OUT TO THE DISK
007550  001406                  BEQ     6$              ;YES, BRANCH
007552  005037  001162          CLR     $REG0           ;THIS IS WHAT RKDB SHOULD CONTAIN
007556  017737  171560  001164  MOV     @RKDB,$REG1     ;GET RKDB
007564  104037                  ERROR   37              ;RKDB SHOULD BE 0 AFTER WRT SINCE THE
                                                        ;LAST WORD WRITTEN WAS SERIALLY SHIFTED
                                                        ;OUT OF RKDB
007566  005201          6$:     INC     R1              ;INCREMENT DRIVE ADDRS TO NXT SECTOR
007570  005205                  INC     R5              ;
007572  122705  000014          CMPB    #14,R5          ;R U GOING TO CHECK THE LAST SECTOR?
007576  001002                  BNE     .+6             ;IF NOT, BRANCH
007600  062705  000004          ADD     #4,R5           ;IF YES, INCREMENT R5 CORRECTLY TO 'EXPCTD RKDA'
                                                        ;AFTER HAVING CHECKED THE LAST SECTOR
007604  005202                  INC     R2              ;HAVE U FORMATTED ALL 12 SECTORS?
007606  001317                  BNE     1$              ;IF NOT, BRANCH BACK & LOOP
                                                        ;IF YES, EXIT

;;*****************************************************************
;*TEST 21        CHECK 'READ FORMAT'-CYLINDER 0, SECTOR 0-13
                ;*THIS TEST PERFORMS A RD FMT ON THE 12 SECTORS OF CYLINDER 0
                ;*THE FOLLOWING IS CHECKED
                ;*1)  IF CNTRL RDY SET WITHIN A CERTAIN TIME ON COMPLETION
                ;*OF THE FUNCTION
                ;*2)  IF 'HE' OR 'ERR' BIT SET?
                ;*3)  IF THE RKDA INCREMENTS CORRECTLY?
                ;*4)  RKBA INCREMENTED CORRECTLY BY 30 (OCTAL)
                ;*5)  RKWC OVERFLOWED TO 0 FROM -14 (OCTAL)
                ;:6)  CORRECT HEADER WAS RECEIVED FROM ALL 12 SECTORS.
                ;*7)  RKCS STILL CONTAINS THE 'RD FMT' FUNCTION BITS.
                ;*IF THERE IS A READ ERROR IN THIS TEST OR ANY
                ;*OTHER TESTS THE USER SHOULD MAKE SURE THAT
                ;*IT IS AN IRRECOVERABLE ERROR AND NOT A TRANSIENT
                ;*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.
;;*****************************************************************
007610  000004  TST21:  SCOPE
007612  005005          CLR     R5
007614  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
007616  104421          TST.SIN                         ;GO CHECK IF SIN IS SET
                                                        ;IS SET, DO DRIVE RESET TO CLR IT
007620  013701  001332          MOV     RKCS,R1
007624  012700  177764          MOV     #-14,R0         ;SET UP COUNT FOR 12 SECTORS
007630  013702  001340          MOV     RKDA,R2
```

```
2363  007634  013712  001350          MOV     DRIVAD,@R2      ;ADDRESS THE DRIVE
2364  007640  012704  033240          MOV     #OUTBUF,R4
2365  007644  010477  171466          MOV     R4,@RkBA        ;ADRS TO WHICH X-FER DATA FROM DSK
2366  007650  012777  177764  171456  MOV     #-14,@RKWC      ;SET UP WORD COUNT FOR 12 HEADERS TO BREAD
2367  007656  012777  002005  171446  MOV     #2005,@RKCS     ;READ FORMAT, GO
2368
2369  007664  105777  171442  1$:     TSTB    @RKCS           ;DID CNTRL RDY SET ON COMPLETION?
2370  007670  100411          BMI     2$              ;YES, BRANCH
2371  007672  005205          INC     R5              ;NO, WAIT FOR IT TO SET
2372  007674  001373          BNE     1$              ;IF WAITED LONG ENOUGH REPORT
2373                                                  ;ERROR, OTHERWISE LOOP BACK & WAIT
2374  007676  004737  020702          JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
2375  007702  013737  001350  001202  MOV     DRIVAD,$REG10
2376  007710  104416          BRKDA4                  ;GO TO 'BDA4' & BREAK CONTENTS OF
2377                                                  ;$REG10 INTO DR#,CYL,SUR,SEC BITS
2378  007712  104045          ERROR   45              ;CNTRL RDY DID NOT SET ON COMPLETION
2379                                                  ;OF READ FORMAT-OF CYLINDER 0.
2380                                                  ;SECTORS 0-13
2381                                                  ;READ FMT WAS DONE STARTING AT <DSK-ADRES>
2382                                                  ;INDICATED IN EROR MESGE
2383  007714  004737  021142  2$:     JSR     PC,CHKHE        ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
2384                                                  ;IF YES RETURN HERE.
2385  007720  104046          ERROR   46              ;'ERR' OR 'HE' BIT SET ON DOING
2386                                                  ;READ FMT-OF CYLINDER 0, SEC 0-13
2387                                                  ;READ FMT WAS DONE STARTING AT <DSK-ADRES>
2388                                                  ;INDICATED IN EROR MESGE
2389  007722  013705  001350  3$:     MOV     DRIVAD,R5
2390  007726  062705  000020          ADD     #20,R5          ;RKDA SHOULD HAVE INCREMENTD TO (R2)
2391
2392  007732  004737  021176          JSR     PC,CHKDA1       ;CHECK IF RKDA INCREMENTED CORRECTLY,
2393                                                  ;IF NOT, RETURN HERE.
2394  007736  104040          ERROR   40              ;RKDA DID NOT INCREMENT BY 12
2395                                                  ;AFTER A 'RD FMT' OF 12 HEADERS OF
2396                                                  ;CYLINDER 0, SECTORS 0-13
2397                                                  ;RKBA SHOULD INCREMENT BY 24 BYTES
2398                                                  ;AT THE END OF X-FER
2399  007740  022777  033270  171370  4$:     CMP     #OUTBUF+30,@RKBA ;DID RKBA INCREMENT CORRECTLY?
2400  007746  001407          BEQ     5$              ;YES, BRANCH
2401  007750  012737  033270  001162          MOV     #OUTBUF+30,$REG0        ;GET EXPCTD RKBA
2402  007756  017737  171354  001164          MOV     @RKBA,$REG1     ;GET ACTUAL RKBA
2403  007764  104042          ERROR   42              ;RKBA DID NOT INCREMENT CORRECTLY
2404                                                  ;AFTER READ FORMAT OF 12 HEADERS
2405  007766  004737  021224  5$:     JSR     PC,CHKWC        ;GO CHECK IF RKWC OVERFLOWED TO 0
2406                                                  ;IF NOT RETURN HERE.
2407  007772  104041          ERROR   41              ;RKWC DID NOT OVERFLOW TO 0
2408                                                  ;AFTER 'RD FMT' OF 12 HEADERS
2409                                                  ;OF CYLINDER 0
2410  007774  005724          6$:     TST     (R4)+           ;WAS THE CORRECT HEADER RECIEVED?
2411  007776  001413          BEQ     7$              ;YES, BRANCH
2412  010000  010037  001162          MOV     R0,$REG0        ;GET SECTOR FOR WHICH THE HEADER
2413  010004  062737  000014  001162  ADD     #14,$REG0       ;COULD NOT BE READ CORRECT
2414  010012  005037  001164          CLR     $REG1           ;EXPCTD HEADER-0, FOR CYL 0
2415  010016  014437  001166          MOV     -(R4),$REG2     ;GET WRONG HEADER RECVD
2416  010022  104043          ERROR   43              ;HEADER WAS NOT READ RIGHT FOR
2417                                                  ;SECTOR (AS IN ER MSGE), & CYL 0
2418  010024  005724          TST     (R4)+           ;WAS THE CORRECT HEADER RECVD?
```

```
2419  010026  005200                  7$:     INC     R0              ;YES, HAVE U CHECKED FOR ALL 12 SECTORS?
2420  010030  001361                          BNE     6$              ;IF NOT, LOOP BACK & CHK HDR FRM NXT SECTR
2421
2422  010032  004737  021250                  JSR     PC,CHKER        ;CHECK IF ANY BIT IN RKER IS SET,
2423                                                                  ;IF YES, RETURN HERE.
2424  010036  104036                          ERROR   36              ;RKER BIT SET ON DOING RD FMT
2425                                                                  ;OF CYL 0, SECTORS 0-13
2426  010040  022711  002204          8$:     CMP     #2204,@R1       ;DOES RKCS STILL CONTAIN FUNCTION BITS?
2427  010044  001406                          BEQ     TST22           ;;YES, EXIT
2428  010046  012737  002204  001162          MOV     #2204,$REG0     ;GET EXPCTD RKCS
2429  010054  011137  001164                  MOV     @R1,$REG1       ;GET ACTUAL RKCS
2430  010060  104024                          ERROR   24              ;RKCS DID NOT CONTAIN 'RD FMT'
2431                                                                  ;FUNCTION BITS ON COMPETION OF
2432                                                                  ;THE FUNCTION
2433
2434
2435
2436                                  ;;*****************************************************************
2437                                  ;*TEST 22       CHECK 'READ',CYLINDER 0, SECTORS 0 TO 13
2438                                          ;*THIS TEST PERFORMS A READ OF ALL THE SECTORS OF CYLINDER 0
2439                                          ;*& CHECKS THE FOLLOWING
2440                                          ;*1) CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
2441                                          ;*OF THE FUNCTION
2442                                          ;*2) IF 'HE' OR 'ERR' BIT SET?
2443                                          ;*3) IF THE CORRECT PSUEDO-HEADER (FIRST WORD OF EVERY)
2444                                          ;*SECTOR, WRITTEN IN A PREVIOUS TEST) WAS RECEIVED.
2445                                          ;*4) IF RKDS CONTAINS THE CORRECT WORD.
2446                                          ;*4) IF RKDA INCREMENTED CORRECTLY.
2447                                          ;*5) IF REST OF THE (377) WORDS IN EACH SECTOR ARE '0' , NOTE
2448                                          ;*PREVIOUSLY ONE WORD WAS WRITTEN PER SECTOR.
2449                                          ;*6) IF RKCS STILL CONTAINS THE 'READ' FUNCTION BITS
2450                                          ;*7) IF CONTROL RESET CLEARS RKDB.
2451                                          ;*  IF TESTING IS BEING DONE ON A SIMULATOR ONLY LAST SECTOR(13)
2452                                          ;*IS READ BECAUSE THE SIMULATOR CAN STORE ONLY 1 SECTOR (256 WORDS).
2453                                          ;*HENCE ONLY THE DATA WRITTEN LAST CAN BE READ BACK.
2454                                  ;;*****************************************************************
2455  010062  000004                  TST22:  SCOPE
2456  010064  012737  010136  001110          MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
2457                                                                  ;ON     ERROR (SW 9)
2458  010072  013703  001332                  MOV     RKCS,R3
2459  010076  013701  001350                  MOV     DRIVAD,R1
2460  010102  010105                          MOV     R1,R5
2461  010104  012704  033240                  MOV     #OUTBUF,R4
2462  010110  005737  001344                  TST     SIMUL           ;TESTING ON SIMULATOR?
2463  010114  001405                          BEQ     9$              ;NO, BRANCH
2464                                                                  ;IF TESTING ON SIMULATOR READ
2465                                                                  ;SECTOR  13  ONLY
2466  010116  052701  000013                  BIS     #13,R1          ;SET BITS FOR SEC 13
2467  010122  052705  000020                  BIS     #20,R5          ;RKDA SHOULD INCRMNT TO THIS AFTER READ
2468  010126  000403                          BR      1$
2469  010130  012702  177764          9$:     MOV     #-14,R2         ;SET COUNT FOR 12 SECTORS
2470  010134  005205                          INC     R5              ;RKDA SHOULD INCREMENT TO
2471                                                                  ;THIS AFTER 1 SECTOR  READ
2472  010136  104413                  1$:     CNT.RESET               ;GO, DO CONTROL RESET
2473                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
2474                                                                  ;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
010140  104421                      TST.SIN           ;GO CHECK IF SIN IS SET
                                                        ;IF SET, DO DRIVE RESET TO CLR IT
010142  010177  171172              MOV     R1,@RKDA  ;ADDRESS THE DRIVE
010146  010477  171164              MOV     R4,@RKBA  ;ADRS TO WHICH X-FER DATA FROM DISK
010152  012777  177400  171154      MOV     #-400,@RKWC ;SETUP WORD COUNT
010160  012713  000005              MOV     #5,@R3    ;READ,GO

010164  005000                      CLR     R0
010166  105713              2$:     TSTB    @R3       ;DID CNTRL RDY SET ON COMPETION?
010170  100410                      BMI     3$        ;YES, BRANCH
010172  005200                      INC     R0        ;NO, WAIT FOR IT TO SET
010174  001374                      BNE     2$        ;IF WAITED LONG ENOUGH, REPORT
                                                        ;ERROR, OTHERWISE LOOP BAK & WAIT
010176  004737  020702              JSR     PC,GT4RG  ;GO, GET RKCS, ER, DS,DA
010202  010137  001202              MOV     R1,$REG10 ;GET SECTOR ADDRES WHERE ERROR OCCURED
010206  104416                      BRKDA4            ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                        ;$REG10 INTO DR # CYL,SUR,SEC BITS
010210  104045                      ERROR   45        ;CNTRL RDY DID NC  SET ON COMPLETION
                                                        ;OF READ OF CYLINDER 0, SECTOR
                                                        ;AS SHOWN IN <DSK-ADRES>
                                                        ;READ WAS DONE STARTING AT <DSK-ADRES>
                                                        ;INDICATED IN EROR MESGE
010212  004737  021134      3$:     JSR     PC,CHKHE1 ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
                                                        ;IF YES RETURN HERE.
010216  104046                      ERROR   46        ;HE OR ERR BIT SET
                                                        ;ON 'READ' OF CYLINDER 0, SECTOR
                                                        ;AS SHOWN IN <DSK-ADRES>
                                                        ;READ WAS DONE STARTING AT <DSK-ADRES>
                                                        ;INDICATED IN EROR MESGE
01C220  020114              4$:     CMP     R1,(R4)   ;WAS THE DATA WORD RECVD, CORRECT?
                                                        ;THE FIRST DATA WORD OF EACH SECTOR
                                                        ;IS AN ADRS WORD COMRISING OF DRIVE NO,
                                                        ;CYLINDER ADRS, SUR, SECTOR ADRS
010222  001407                      BEQ     5$
010224  010137  001162              MOV     R1,$REG0  ;GET EXPCTD DATA WORD FROM DISK
010230  011437  001164              MOV     (R4),$REG1 ;GET THE DATA WORD RECVD
010234  010137  001166              MOV     R1,$REG2  ;GET DISK ADRES
010240  104044                      ERROR   44        ;DID NOT RECIEVE CORRECT DATA WORD ON
                                                        ;READ, OF CYLINDER 0, SECTOR AS SHOWN IN 'DSK
                                                        ;ADRES' OF EXPCTD DATA WORD
010242  004737  021176      5$:     JSR     PC,CHKDA1 ;CHECK IF RKDA INCREMENTED CORRECTLY,
                                                        ;IF NOT RETURN HERE.
010246  104040                      ERROR   40        ;RKDA DID NOT INCREMENT CORRECTLY
                                                        ;AFTER READ OF 1 WORD, FROM CYL 0
                                                        ;SEC IN ERROR IS 1 LESS THAN THAT
                                                        ;SHOWN IN EXPCTD RKDA
                                                        ;
                                                        ;AS A RESULT OF 'WRT FMT' IN A PREVIOUS TEST
                                                        ;FIRST WORD OF EVERY SECTOR IS NON-
```

```
2531                                                                    ;ZERO (PSUEDO-HDR), REST 377 WORDS
2532                                                                    ;ARE ALL 0'S.
2533                                                                    ;CHECK IF THE REST OF THE 377
2534                                                                    ;WORDS ARE ALL 0'S
2535  010250  012737  177775  001370          MOV     #-3,EFLG1       ;ALLOW ONLY 3 ERRORS
2536  010256  012700  033242                  MOV     #OUTBUF+2,R0    ;INITIALIZE PTR TO 2ND WRD IN BUFR
2537  010262  012737  177401  001362          MOV     #-377,COUNT     ;CHECK 377 WORDS IN THE BUFFER
2538  010270  005710                  11$:    TST     @R0             ;IS THIS WRD 0?
2539  010272  001005                          BNE     12$             ;NO, ERROR
2540  010274  005720                          TST     (R0)+           ;INCRMNT PTR TO NXT WRD
2541  010276  005237  001362                  INC     COUNT           ;CHKD ALL 377 WRDS?
2542  010302  001372                          BNE     11$
2543  010304  000412                          BR      7$              ;YES, BRANCH
2544  010306  005037  001162          12$:    CLR     $REG0           ;GET EXPCTD WORD
2545  010312  012037  001164                  MOV     (R0)+,$REG1     ;GET WORD RECVD
2546  010316  010137  001166                  MOV     R1,$REG2        ;GET DISK ADRES, ERROR IN THIS
2547                                                                  ;SECTOR
2548  010322  104044                          ERROR   44              ;DATA ERROR, THE LAST 377 WORDS
2549                                                                  ;READ FROM EACH SECTOR SHOULD BE 0
2550                                                                  ;IN A PREVIOUS TEST, FIRST WORD OF
2551                                                                  ;EVERY SEC (CYL 0) WAS WRITTEN AS A
2552                                                                  ;PSUEDO-HDR, REST OF THE WORDS IN THE
2553                                                                  ;SECTR ARE AUTOMATICALLY WRITTEN AS
2554                                                                  ;0'S. THIS ERROR MAY MEAN THAT IT
2555                                                                  ;DIDN'T HAPPEN SO
2556  010324  005237  001370                  INC     EFLG1           ;ALLOW ONLY 3 DATA ERORS OF THIS KIND
2557  010330  001357                          BNE     11$
2558
2559
2560  010332  005737  001344          7$:     TST     SIMUL           ;TESTING ON SIMULATOR?
2561  010336  001011                          BNE     10$             ;YES BRANCH
2562                                                                  ;IF NOT TESTING ON SIMULATOR GO AHEAD
2563                                                                  ; & READ ALL 12 SECTORS ON CYL 0
2564  010340  005201                          INC     R1              ;INCREMENT DRIV-ADRES TO NXT SECTOR
2565  010342  005205                          INC     R5              ;INCREMENT 'EXPCTD DRIV-ADRES'
2566  010344  122705  000014                  CMPB    #14,R5          ;R U GOING TO READ THE LAST SECTOR?
2567  010350  001002                          BNE     .+6             ;IF NOT, BRANCH
2568  010352  062705  000004                  ADD     #4,R5           ;IF YES, INCREMENT 'EXPCTD RKDA'
2569                                                                  ;CORRECTLY
2570  010356  005202                          INC     R2              ;HAVE U READ ALL 12 SECTORS?
2571  010360  001266                          BNE     1$              ;IF NOT LOOP BACK & READ THE
2572                                                                  ;NXT SECTOR
2573  010362  022713  000204          10$:    CMP     #204,@R3        ;DOES RKCS, STILL HAVE THE 'READ' FUNCTION
2574  010366  001406                          BEQ     8$              ;YES, BRANCH
2575  010370  012737  000204  001162          MOV     #204,$REG0      ;GET EXPCTD RKCS
2576  010376  011337  001164                  MOV     @R3,$REG1       ;GET RKCS RECVD
2577  010402  104024                          ERROR   24              ;RKCS SHOULD STILL CONTAIN THE 'READ'
2578                                                                  ;FUNCTION BITS
2579  010404  104413                  8$:     CNT.RESET               ;GO ,DO CONTROL RESET
2580                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
2581                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
2582                                                                  ;ISSUED AND AFTER A CERTAIN TIME
2583                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
2584                                                                  ;AN ERROR IS REPORTED. NOTE THAT
2585                                                                  ;THE PC IN ERROR MESSAGE IS THE
2586                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
```

```
                                                        ;THIS IS A VERY BASIC ERR & IF IT
                                                        ;OCCURS GO BACK TO TEST 10
010406  005777  170730          TST     @RKDB           ;DID CNTRL RESET CLEAR RKDB?
010412  001407                  BEQ     TST23           ;;YES, EXIT
010414  013737  001342  001162  MOV     RKDB,$REG0      ;GET ADRES OF RKDB
010422  017737  170714  001164  MOV     @RKDB,$REG1     ;GET CONTENTS OF RKDB
010430  104102                  ERROR   102             ;CONTROL RESET DID NOT
                                                        ;CLEAR RKDB


                        ;;**************************************************************
                        ;*TEST 23       CHECK 'WRITE FORMAT' OF THE DISK
                                ;*THIS TEST WRITE FORMATS THE ENTIRE DISK.  THE FIRST
                                ;*WORD OF EVERY SECTOR IS WRITTEN TO BE A PSUEDO-HEADER
                                ;*CONSISTING OF THE DRIVE #, CYLINDER #, SURFACE & SECTOR #.
                                ;*1 SECTOR IS WRITTEN AT A TIME. THE WRITING 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.
                                ;*IMPORTANCE OF THIS TEST SHOULD BE REALIZED, THIS IS
                                ;*THE FIRST TIME EACH & EVERY SECTOR ON THE DISK IS
                                ;*ACCESSED & WRITTEN ON.  THIS IS THE FIRST TIME RKDA
                                ;*IS BEING MADE TO INCREMENT OVER THE ENTIRE DISK (FROM
                                ;*000000 TO 014520) IF A 'SIN' OCCURS AT ANY POINT
                                ;*A DRIVE RESET IS DONE BEFORE DOING WRT FMT FOR THE NEXT
                                ;*SECTOR.  ANY OTHER ERROR IS CLEARED THROUGH A CONTROL RESET.
                                ;*THE FOLLOWING CHECKING IS DONE AFTER WRITING EACH
                                ;*CYLINDER.
                                ;*1. CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
                                ;*OF THE FUNCTION.
                                ;*2. IF 'SIN' OCCURRED?
                                ;*3. IF 'HE' OR 'ERR' BIT SET?
                                ;*4. IF RKDA INCREMENTED CORRECTLY, INCLUDING BOUNDARY
                                ;*CONDITIONS (SECTOR COUNTER BITS OVERFLOWING INTO SURFACE.
                                ;*SURFACE BIT OVERFLOWING INTO CYLINDER BITS) AT THE END
                                ;*OF THIS POINTERS ARE INCREMENTED ADJUSTED, ETC.
                                ;*& 'WRT FMT' ON THE NEXT SECTOR IS DONE.
                        ;;**************************************************************
000004          TST23:  SCOPE
012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
012737  010472  001110          MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
                                                        ;ON  ERROR (SW 9)
005003                          CLR     R3              ;(R3)=0, SURFACE 0 BEING WRITTEN
                                                        ;(R3)-1, SURFACE 1 BEING WRITTEN
012704  177465                  MOV     #-313,R4        ;SET UP COUNT FOR 203 CYLINDERS
012702  177764                  MOV     #-14,R2         ;SET UP COUNT FOR 12 SECTORS
013701  001350                  MOV     DRIVAD,R1       ;GET DRIVE ADRES
010105                          MOV     R1,R5           ;STORE IT
005205                          INC     R5
104413          1$:     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 S IF IT
                                                        ;OCCURS GO BACK TO TEST ID
010474  104421                          TST.SIN       ;GO CHECK IF SIN IS SET
                                                        ;IF SET, DO DRIVE RESET TO CLR IT
010476  005037  001362          7$:     CLR     COUNT
010502  010137  033240                  MOV     R1,OUTBUF     ;THIS WORD TO BE WRITTEN.  THE FIRST
                                                        ;WORD OF EACH SECTOR WILL BE THE ACTUAL
                                                        ;DISK-ADRES, CONSISTING OF THE DRIVE NO,
                                                        ;CYL ADRES, SURFACE BIT SECTOR ADRES
010506  012777  033240  170622          MOV     #OUTBUF,@RKBA ;ADRES FROM WHICH WORD IS TO B X-FERRED
010514  012777  177777  170612          MOV     #-1,@RKWC     ;SET UP WORD COUNT
010522  010177  170612                  MOV     R1,@RKDA      ;ADRES THE DRIVE, WITH CORRECT CYL
                                                        ;& SECTOR ADRES
010526  012777  002003  170576          MOV     #2003,@RKCS   ;WRITE FORMAT, GO
010534  105777  170572          2$:     TSTB    @RKCS         ;DID CNTRL RDY SET
010540  100411                          BMI     3$            ;YES, BRANCH
010542  005237  001362                  INC     COUNT         ;NO, HAVE U WAITED LONG ENOUGH?
010546  001372                          BNE     2$            ;IF NOT, LOOP BACK & WAIT
                                                        ;IF YES, REPORT ERROR
010550  004737  020702                  JSR     PC,GT4RG      ;GO, GET RKCS, ER, DS,DA
010554  010137  001202                  MOV     R1,$REG1D     ;GET DISK ADRES, WHERE ERROR OCCURED
010560  104416                          BRKDA4                ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                        ;$REG1D INTO DR #,CYL,SUR,SEC BITS
010562  104031                          ERROR   31            ;CNTRL RDY DID NOT SET ON COMPLETION
                                                        ;OF 'WRITE FORMAT', ON SECTOR AS
                                                        ;SHOWN IN <DSK-ADRES>
                                                        ;WRT FMT WAS DONE STARTING AT <DSK-ADRES
                                                        ;INDICATED IN EROR MSGE.
010564  032777  001000  170534  3$:     BIT     #1000,@RKDS   ;DID SIN BIT SET?
010572  001405                          BEQ     4$            ;NO, BRANCH
010574  004737  020710                  JSR     PC,GT3RG      ;GO, GET RKCS, ER, DS
010600  010137  001170                  MOV     R1,$REG3      ;GET, DISK-ADRES WHERE ERROR OCCURED
010604  104001                          ERROR   1             ;SIN SET WHILE DOING WRT FMT
                                                        ;TO DISK-ADRES (AS IN $REG3)
010606  004737  021134          4$:     JSR     PC,CHKHE1     ;CHECK IF 'ERR' OR 'HE' BIT IS SET
                                                        ;IF YES, RETURN HERE.
010612  104032                          ERROR   32            ;HE OR ERR SET WHILE DOING WRITE
                                                        ;FORMAT ON SECTOR AS INDICATED IN
                                                        ;<DSK-ADRES>
                                                        ;WRT FMT WAS DONE STARTING AT <DSK-ADRES>
                                                        ;INDICATED IN EROR MSGE.
010614  004737  021176          5$:     JSR     PC,CHKDA1     ;CHECK IF RKDA INCREMENTED CORRECTLY,
                                                        ;IF NOT, RETURN HERE.
010620  104033                          ERROR   33            ;RKDA DID NOT INCREMENT CORRECTLY
                                                        ;AFTER 'WRITE FORMAT' WAS DONE
                                                        ;TO THE SECTOR PREVIOUS TO THAT
                                                        ;INDICATED IN 'EXPCTD' RKDA
010622  005201          6$:     INC     R1            ;INCREMENT TO THE NXT SECTOR
010624  005205                  INC     R5            ;INCREMENT R5, TO WHAT RKDA WILL INCREMENT
010626  022702  177776          CMP     #-2,R2        ;R U GOING TO FORMAT THE LAST SECTOR
                                                        ;IN THE CYLINDER ?
010632  001002                  BNE     .+6           ;IF NOT, BRANCH
010634  062705  000004          ADD     #4,R5         ;INCREMENT R5 CORRECTLY TO 'EXPCTD RKDA'
010640  005202                  INC     R2            ;HAVE U FORMATTED ALL 12 SECTORS
```

```
                                                        ;ON THIS CYLINDER
010642  001313                        BNE     1$      ;IF NOT, LOOP BACK & FORMAT THE
                                                        ;NEXT SECTOR
                                                        ;YES
010644  012702  177764                  MOV     #-14,R2 ;RESET THE COUNT FOR 12 SECTORS
010650  042701  000037                  BIC     #37,R1  ;CLEAR THE SEC ADRES BITS
010654  005703                        TST     R3      ;SURFACE 1?
010656  001006                        BNE     8$      ;YES, BRANCH
010660  005203                        INC     R3      ;NO, SET FLAG
010662  062701  000020                ADD     #20,R1  ;INCREMENT TO THE NXT SURFACE
010666  010105                        MOV     R1,R5   ;THIS IS WHAT RKDA SHOULD
010670  005205                        INC     R5      ;INCREMENT TO.
010672  000677                        BR      1$      ;GO, DO NXT SURFACE
010674  062701  000040          8$:   ADD     #40,R1  ;INCREMENT TO NXT CYL
010700  010105                        MOV     R1,R5   ;POSITION FOR
010702  005205                        INC     R5      ;EXPCTD RKDA
010704  005003                        CLR     R3
010706  005204                        INC     R4      ;HAVE U FORMATTED ALL 203 CYLINDERS
010710  001270                        BNE     1$      ;IF NOT, LOOP BACK & FORMAT THE
                                                        ;NEXT CYLINDER


                        ;;**************************************************************
                        ;*TEST 24       CHECK 'READ FORMAT' FOR THE ENTIRE DISK
                                ;*THIS TEST READ FORMATS THE ENTRE DISK, WHICH WAS WRT
                                ;*FORMATTED IN THE PREVIOUS TEST.  THE FOLLOWING CHECKING
                                ;*IS DONE
                                ;*1. CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
                                ;*OF FUNCTION
                                ;*2. IF 'SIN' OCCURRED?
                                ;*3. IF 'HE' OR 'ERR' OCCURRED?
                                ;*4. RKDA INCREMENTED CORRECTLY.
                                ;*5. IF THE CORRECT HEADER WAS READ.
                                ;*6. IF RKWC OVERFLOWED CORRECTLY.
                                ;*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.
                        ;;**************************************************************
000004                  $T24:   SCOPE
012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
012737  011006  001110          MOV     #$,$LPERR       ;SET RETURN ADRES FOR LUPING
                                                        ;ON     ERROR (SW 9)
005037  001356                  CLR     INDX1           ;INDX1=0, SURFACE 0 BEING READ
                                                        ;INDX1=1, SURFACE 1 BEING READ
013701  001350                  MOV     DRIVAD,R1       ;GET DRIVE ADRES
010102                          MOV     R1,R2
005737  001344                  TST     SIMUL           ;TESTING ON SIMULATOR?
001410                          BEQ     12$             ;NO, BRANCH
052701  014533                  BIS     #14533,R1       ;SET BITS FOR CYL 312, SEC 13, SUR 1
                                                        ;ON SIMULATOR, CHECK ONLY CYL 312,
                                                        ;SECTOR 13, SURFACE 1
052702  014540                  BIS     #14540,R2       ;RKDA SHOULD INCRMNT TO THIS AFTR
```

```
                                                                ;RD FMT OF 1 SECTOR
010760  012737  177777  001370        MOV     #-1,EFLG1       ;SET COUNT FOR READING HDR
                                                                ;FROM 1 SECTOR ONLY
010766  000407                        BR      1$
010770  012705  177465        12$:    MOV     #-313,R5        ;SET UP COUNT FOR 203 CYLINDERS
010774  012737  177764  001370        MOV     #-14.,EFLG1     ;SET COUNT FOR 12 HDRS TO BE
                                                                ;READ FROM EACH CYLINDER
011002  062702  000020                ADD     #20,R2          ;THIS IS WHAT RKDA SHOULD INCREMENT
                                                                ;BY, AFTER 'RD FMT' OF EACH CYLINDER
011006  104413                1$:     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
011010  104421                        TST.SIN                 ;CHECK IF SIN IS SET
                                                                ;IF SET DO DRV-RESET TO CLR IT

011012  012703  033240                MOV     #OUTBUF,R3      ;STORE ADRES OF BUFFER
011016  005037  001360        11$:    CLR     INDX2
011022  010377  170310                MOV     R3,@RKBA        ;ADRES TO WHICH DATA IS TO BE X-FERRED
                                                                ;FROM THE DISK
011026  013777  001370  170300        MOV     EFLG1,@RKWC     ;SET UP WORD COUNT FOR 12 HEADERS
                                                                ;TO BE READ OFF EACH CYLINDER
                                                                ;(ONLY 1 FOR SIMULATOR)
011034  010177  170300                MOV     R1,@RKDA        ;ADRES THE DRIVE WITH CORRECT
                                                                ;CYLINDER & SECTOR ADRES
011040  012777  002005  170264        MOV     #2005,@RKCS     ;READ FORMAT, GO

011046  105777  170260        2$:     TSTB    @RKCS           ;DID CNTRl RDY SET?
011052  100411                        BMI     3$              ;YES, BRANCH
011054  005237  001360                INC     INDX2           ;NO, HAVE U WAITED LONG ENOUGH?
011060  001372                        BNE     2$              ;IF NOT, LOOP BACK & WAIT FOR IT
                                                                ;IF YES, REPORT ERROR
011062  004737  020702                JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
011066  010137  001202                MOV     R1,$REG10       ;GET DRIV-ADRES STARTING WHICH
                                                                ;'READ FORMAT' WAS DONE
011072  104416                        BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                                ;$REG10 INTO DR #,CYL,SUR,SEC BITS
011074  104045                        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.

011076  032777  001000  170222  3$:     BIT     #1000,@RKDS     ;DID 'SIN' SET?
011104  001405                        BEQ     4$              ;NO, BRANCH
011106  004737  020710                JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
011112  010137  001170                MOV     R1,$REG3        ;GET DISK-ADRES WHERE 'SIN'
                                                                ;OCCURED
```

```
                                ERROR   1               ;SIN ERROR ON DOING RD FMT
                                                        ;TO CYL INDICATED IN $REG3
                        4$:     JSR     PC,CHK4E1       ;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
                        5$:     CMP     R2,@RKDA        ;DID RKDA INCREMENT CORRECTLY BY 12 SEC
                                BEQ     6$
                                MOV     R2,$REG1D       ;GET EXPCTD RKDA
                                BRKDA0                  ;GO TO 'BDA0' & BREAK  CONTENTS OF
                                                        ;$REG1D INTO DR #,CYL,SUR,SEC BITS
                                MOV     @RKDA,$REG1D    ;GET RECVD RDKA
                                BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                        ;$REG1D 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
                        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)
                        8$:     TST     (R3)+           ;INCREMENT POINTER TO THE NXT WORD
                                                        ;IN MEMORY WHERE THE RECVD HDR IS STORED
                                INC     R0              ;HAVE U CHECKED ALL 12 HEADERS?
                                BNE     7$              ;IF NOT, LOOP BACK & CHK THE NXT.
                                                        ;YES, ALL HEADERS FOR THIS CYLINDER
                                                        ;CHECKED.
                                JSR     PC,CHKWC        ;CHECK IF RKWC OVERFLOWED TO 0, IF
                                                        ;NOT RETURN HERE.
                                ERROR   41              ;RKWC DID NOT OVERFLOW AFTER DOING
                                                        ;RDFMT OF 12 SECTORS ON THE CYLINDER
                                                        ;NOTE THAT 'RKDA' IS THE INCREMENTED
                                                        ;RKDA AFTER THE RDFMT
                        9$:     TST     SIMUL           ;TSTING ON SIMULATOR?
                                BNE     TST25           ;;IF YES, EXIT
                                                        ;NO
                                TST     INDX1           ;DOING SURFACE 1
                                BNE     10$             ;YES, BRANCH
                                INC     INDX1           ;NO
                                ADD     #20,R1          ;INCREMENT DRIV ADRES TO THE NXT SURFACE
                                MOV     R1,R2
```

```
2867  011260  062702  000020              ADD     #20,R2          ;THIS IS WHAT RKDA SHOULD INCREMENT
2868                                                              ;TO, AFTER READ FMT OF THE CYLINDER
2869  011264  000137  011006              JMP     1$              ;GO RD FMT THE NXT SURFACE
2870  011270  005037  001356    10$:      CLR     INDX1           ;
2871  011274  042701  000037              BIC     #37,R1          ;CLR SEC, SURFACE BITS
2872  011300  062701  000040              ADD     #40,R1          ;INCREMENT TO NXT CYL
2873  011304  010102                      MOV     R1,R2           ;THIS IS WHAT RKDA SHOULD BE
2874  011306  062702  00002C              ADD     #20,R2          ;AFTER RD FMT OF CYLINDER
2875  011312  005205                      INC     R5              ;HAVE U DONE ALL CYLINDERS?
2876  011314  001402                      BEQ     TST25           ;;EXIT
2877  011316  000137  011006              JMP     1$              ;IF NOT, LOOP BACK & READ FMT FROM
2878                                                              ;THE NXT CYLINDER
2879
2880
2881                            ;;*****************************************************************
2882                            ;*TEST 25       CHECK 'READ' OF THE ENTIRE DISK
2883                                    ;*READ OF THE ENTIRE DISK (ONE WORD PER SECTOR) IS DONE
2884                                    ;*IN THIS TEST.  IN A PREVIOUS TEST THE FIRST WORD OF
2885                                    ;*EVERY SECTOR WAS WRITTEN LIKE A PSUEDO-HEADER (DRIVE #,
2886                                    ;*CYLINDER #, SURFACE & SECTOR #).  THESE PSUEDO HEADERS
2887                                    ;*WILL BE READ & CHECKED IN THIS TEST, PROVING THAT ANY
2888                                    ;*SECTOR CAN BE ACCESSED AND READ.
2889                                    ;*THE FOLLOWING CHECKING IS DONE
2890                                    ;*1. CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
2891                                    ;*OF FUNCTION.
2892                                    ;*2. IF 'SIN' OCCURRED?
2893                                    ;*3. IF 'HE' OR 'ERR' OCCURRED?
2894                                    ;*4. THE CORRECT FIRST WORD FROM EVERY SECTOR
2895                                    ;*WAS RECEIVED.  THIS WORD REFLECTS THE ABSOLUTE
2896                                    ;*DISK ADDRESS (DRV #, CYL #, SUR, SEC#) OF THAT SECTOR.
2897                                    ;*5. IF RKDB CONTAINED THE CORRECT WORD.
2898                                    ;*IF 'SIN' OCCURS DRIVE RESET IS DONE BEFORE READING
2899                                    ;*THE NEXT SECTOR.  READ IS DONE IN THIS ORDER SEC 0-11
2900                                    ;*CYL 0 SUR 0 -> SEC 0-11 CYL 0 SUR 1 -> SEC 0-11 CYL 1,....
2901                                    ;*IF TESTING ON SIMULATOR ONLY LAST CYLINDER (312), LAS
2902                                    ;*SECTOR (13), SURFACE 1 IS READ.
2903                            ;;*****************************************************************
2904  011322  000004            TST25:  SCOPE
2905  011324  012737  000001  001206    MOV     #1,$TIMES       ;;DO 1 ITERATION
2906  011332  012737  011376  001110    MOV     #1$,$LPERR      ;SET RETURN ADRES FOR
2907                                                            ;LOOPING ON ERROR (SW9)
2908  011340  012703  033240            MOV     #OUTBUF,R3
2909  011344  005004                    CLR     R4              ;FLAG, CLEAR WHEN READING SURFACE 0
2910                                                            ;SET WHEN READING SURFACE 1
2911  011346  013701  001350            MOV     DRIVAD,R1       ;GET DRIVE ADDRESS
2912  011352  005737  001344            TST     SIMUL           ;TSTING ON SIMULATOR?
2913  011356  001403                    BEQ     10$             ;IF NOT BRANCH
2914  011360  052701  014533            BIS     #14533,R1       ;SET ADRES BITS FOR LAST CYL (312)
2915  011364  000404                    BR      1$              ;LAST SECTOR (13), SURFACE 1
2916  011366  012700  177764    10$:    MOV     #-14,R0         ;SET COUNT FOR 12 SECTORS
2917  011372  012705  177465            MOV     #-313,R5        ;SET UP COUNT FOR 203 CYLINDERS
2918
2919  011376  104413             1$:    CNT.RESET               ;GO, DO CONTROL RESET
2920                                                            ;THIS IS A CALL FOR THE 'CNTRL-
2921                                                            ;RESET' ROUTINE. A CONTROL RESET IS
2922                                                            ;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
011400  104421                  TST.SIN               ;GO CHECK SIN, IF SET DO
                                                        ;DRIVE RESET TO CLR IT
011402  005037  001356      8$: CLR     INDX1
011406  010377  167724          MOV     R3,@RKBA      ;ADRES TO WHICH DATA IS TO B X-FERRED
                                                        ;FROM THE DISK
011412  012777  177777  167714  MOV     #-1,@RKWC     ;SET UP WORD COUNT
011420  010177  167714          MOV     R1,@RkDA      ;ADRES THE DRIVE WITH CORRECT
                                                        ;CYLINDER & SECTOR ADRES
011424  012777  000005  167700  MOV     #5,@RKCS      ;READ, GO

011432  105777  167674      2$: TSTB    @RKCS         ;DID CNTRL RDY SET?
011436  100411                  BMI     3$            ;YES, BRANCH
011440  005237  001356          INC     INDX1         ;NO, HAVE U WAITED LONG ENOUGH
011444  001372                  BNE     2$            ;IF NOT, LOOP BACK & WAIT FOR IT
                                                        ;IF YES, REPORT ERROR
011446  004737  020702          JSR     PC,GT4RG      ;GO, GET RKCS, ER, DS,DA
011452  010137  001202          MOV     R1,$REG10     ;GET DISK-ADRES WHERE ERROR OCCURED
011456  104416                  BRKDA4                ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                        ;$REG10 INTO DR #,CYL,SUR,SEC BITS
011460  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

011462  032777  001000  167636  3$: BIT     #1000,@RKDS   ;DID 'SIN' SET?
011470  001405                  BEQ     4$            ;NO, BRANCH
011472  004737  020710          JSR     PC,GT3RG      ;GO, GET RKCS, ER, DS
011476  010137  001170          MOV     R1,$REG3      ;GET DISK-ADRES WHERE SIN OCCURED3
011502  104001                  ERROR   1             ;'SIN' ERROR ON DOING READ FROM
                                                        ;DISK-ADRES INDICATED IN $REG3
011504  004737  021134      4$: JSR     PC,CHKHE1     ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
                                                        ;IF YES, RETURN HERE.
011510  104046                  ERROR   46            ;'HE' OR 'ERR' ON DOING A READ OF
                                                        ;1 WORD FROM ADRES AS INDICATED
                                                        ;IN <DISK-ADRES>
                                                        ;'RKDA' IN EROR MSGE GIVES THE
                                                        ;CONTENTS OF RKDA AT THE TIME OF EROR
011512  020113              5$: CMP     R1,(R3)       ;WAS THE CORRECT DATA WORD RECVD?
011514  001407                  BEQ     6$
011516  010137  001162          MOV     R1,$REG0      ;GET EXPCTD DATA WORD
011522  011337  001164          MOV     (R3),$REG1    ;GET DATA WORD RECVD
011526  010137  001166          MOV     R1,$REG2      ;GET DISK-ADRES
011532  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.
011534  020177  167602          6$:     CMP     R1,@RKDB        ;DOES RKDB CONTAIN CORRECT WORD
011540  001406                          BEQ     7$              ;YES, BRANCH
011542  010137  001162                  MOV     R1,$REG0        ;NO, GET EXPCTD RKDB
011546  017737  167570  001164          MOV     @RKDB,$REG1     ;GET RKDB RECVD
011554  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
011556  005737  001344          7$:     TST     SIMUL           ;TESTING ON SIMULATOR?
011562  001022                          BNE     TST26           ;IF YES, EXIT
011564  005201                          INC     R1              ;INCREMENT  TO ADRES NEXT SECTOR
011566  005200                          INC     R0              ;HAVE U CHKD ALL 12 SECTORS?
011570  001302                          BNE     1$              ;IF NOT, LUP BAK & CHK THE NXT
                                                        ;IF YES...
011572  012700  177764                  MOV     #-14,R0         ;RESET THE COUNT FOR 12 SECTORS
011576  042701  000037                  BIC     #37,R1          ;CLEAR SECTOR, SURFACE BITS
011602  005704                          TST     R4              ;DOING SURFACE 1?
011604  001004                          BNE     9$              ;YES, BRANCH
011606  005204                          INC     R4              ;NO
011610  062701  000020                  ADD     #20,R1          ;INCREMENT THE ADRES TO NXT SURFACE
011614  000670                          BR      1$              ;GO READ SURFACE 1
011616  005004                  9$:     CLR     R4
011620  062701  000040                  ADD     #40,R1          ;INCREMENT TO NXT CYL
011624  005205                          INC     R5              ;HAVE U CHKD ALL 203 CYLINDERS
011626  001263                          BNE     1$              ;IF NOT, LOOP BACK & CHK THE NXT CYLINDER
                                                        ;YES


                                ;;*****************************************************************
                                ;*TEST 26       CHECK 'SEEK' FUNCTION, WITH DIFFERRENT 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 BOTH
                                        ;* 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
                                ;;*****************************************************************
011630  000004                  TST26:  SCOPE
011632  012737  000005  001206          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
011640  012703  001372                  MOV     #SEEK0,R3       ;INITIALIZE POINTER TO THE FIRST
                                                        ;SEEK ADDRESS
011644  005037  001356                  CLR     INDX1           ;INDX1, WHEN 0 INDICATES SEEK IN FWD DIRECTION
                                                        ;       WHEN 1 INICATES SEEK IN REV DIRECTION
011650  013700  001332                  MOV     RKCS,R0
011654  013701  001326                  MOV     RKDS,R1
011660  013702  001330                  MOV     RKER,R2
011664  012737  011672  001110          MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING ON
                                                        ;EROR (SW 9)
011672  000240                  1$:     NOP
011674  104413                  2$:     CNT.RESET               ;GO, DO CONTROL RESET
                                                        ;THIS IS A CALL FOR THE 'CNTRL-
```

```
3035                                                    ;RESET' ROUTINE. A CONTROL RESET IS
3036                                                    ;ISSUED AND AFTER A CERTAIN TIME
3037                                                    ;IF THE 'CNTRL RDY' DOES NOT SET
3038                                                    ;AN ERROR IS REPORTED. NOTE THAT
3039                                                    ;THE PC IN ERROR MESSAGE IS THE
3040                                                    ;PC WHERE 'CNT.RESET' IS LOCATED.
3041                                                    ;THIS IS A VERY BASIC ERR & IF IT
3042                                                    ;OCCURS GO BACK TO TEST 10
3043  011676  104421                    TST.SIN        ;GO, CHECK IF SIN IS SET, IF SET
3044                                                    ;DO DRV-RESET TO CLEAR IT
3045
3046
3047  011700  013704  001350            MOV     DRIVAD,R4       ;GET DRIV-ADRES
3048  011704  051304                    BIS     (R3),R4         ;SET CYLINDER BITS
3049  011706  010477  167426            MOV     R4,@RKDA        ;ADDRS THE DRIVE
3050  011712  012710  000011            MOV     #11.,@R0        ;SET 'SEEK', 'GO'
3051
3052  011716  104412                    CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
3053                                                            ;IF SO, SKIP THE EROR MESSAGE.
3054  011720  104021                    ERROR   21              ;'CNTRL RDY' DID NOT SET AFTER
3055                                                            ;SENDING CYL ADD TO THE DRIV, 'ADD ACK'
3056                                                            ;FROM DRIVE SHLD HAVE COME BACK
3057                                                            ;THEREUPON SETTING 'CNTRL RDY'
3058  011722  005005            4$:     CLR     R5
3059  011724  032711  000100    5$:     BIT     #100,@R1        ;DID R/W/S RDY SET?
3060  011730  001005                    BNE     6$              ;YES, BRANCH
3061  011732  005205                    INC     R5              ;NO, WAIT
3062  011734  001373                    BNE     5$              ;WAITED LONG?
3063  011736  004737  020702            JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
3064  011742  104026                    ERROR   26              ;R/W/S RDY DID NOT SET ON
3065                                                            ;COMPLETION OF SEEK
3066  011744  032711  001000    6$:     BIT     #1000,@R1       ;DID SIN SET?
3067  011750  001403                    BEQ     7$              ;NO, BRANCH
3068  011752  004737  020702            JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
3069  011756  104001                    ERROR   1               ;SIN SET ON DOING SEEK
3070  011760  032710  140000    7$:     BIT     #140000,@R0     ;DID 'HE' OR 'ERR' SET?
3071  011764  001403                    BEQ     8$              ;YES
3072  011766  004737  020702            JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
3073  011772  104022                    ERROR   22              ;'ERR OF 'HE' BIT SET WHEN
3074                                                            ;SEEKING TO CYL AS INDICATED
3075                                                            ;IN RKDA
3076
3077  011774  022710  000210    8$:     CMP     #210,@R0        ;DOES RKCS STILL CONTAIN THE 'SEEK' FNCTION
3078  012000  001406                    BEQ     9$              ;YES - EXIT
3079  012002  011037  001164            MOV     @R0,$REG1       ;NO, GET RKCS RECVD
3080  012006  012737  000210  001162    MOV     #210,$REG0      ;GET EXPCTD RKCS
3081  012014  104024                    ERROR   24              ;RKCS SHOULD CONTAIN THE 'SEEK' BITS
3082                                                            ;IF NOT, ERROR
3083
3084  012016  020477  167316    9$:     CMP     R4,@RKDA        ;DID RKDA CHANGE?
3085  012022  001406                    BEQ     10$             ;NO
3086  012024  010437  001162            MOV     R4,$REG0        ;YES, GET EXPCTD?
3087  012030  017737  167304  001164    MOV     @RKDA,$REG1     ;GET RKDA
3088  012036  104027                    ERROR   27              ;RKDA CHANGED AFTER DOING SEEK
3089
3090  012040  010477  167274    10$:    MOV     R4,@RKDA        ;ADRES THE DRIVE,SEC 0
```

```
3091  012044  012777  033240  167264          MOV     #OUTBUF,@RKBA   ;READ ONE HEADER INTO THIS
3092  012052  012777  177777  167254          MOV     #-1,@RKWC       ;BUS ADRES
3093  012060  012710  002005                  MOV     #2005,@RD       ;GO,READ FORMAT
3094  012064  104414                          CNT.RDY                 ;WAIT FOR CNTRL RDY
3095  012066  021337  033240                  CMP     (R3),OUTBUF     ;WAS THE CORRECT READER READ (FROM
3096  012072  001410                          BEQ     11$             ;CYLINDER TO WHICH SEEK WAS DONE BEFORE)
3097  012074  005037  001162                  CLR     $REG0           ;STORE SEC # FROME  WHERE HDR WAS RD (0)
3098  012100  011337  001164                  MOV     (R3),$REG1      ;GET EXPCTD HEADER
3099  012104  013737  033240  001166          MOV     OUTBUF,$REG2    ;GET HDR RECVD
3100  012112  104043                          ERROR   43              ;WRONG HDR WAS RECVD FROM CYLINDER (ADRES
3101                                                                  ;IN ER MSGE). NOTE THAT A PURE SEEK WAS
3102                                                                  ;DONE TO THIS CYL BEFORE  READING HDR
3103                                                                  ;USING READ FORMAT
3104  012114  005737  001356          11$:    TST     INDX1           ;SEEK IN REVRSE DIRECTION?
3105  012120  001007                          BNE     12$             ;YES, BRANCH
3106  012122  005723                          TST     (R3)+           ;NO, INCREMENT PTR TO NXT SEEK ADRES
3107  012124  022703  001400                  CMP     #SEEK2+2,R3     ;DONE WITH ALL SKS IN FWD DIR?
3108  012130  001260                          BNE     1$              ;NO, GO & DO NXT ONE
3109  012132  005237  001356                  INC     INDX1           ;SET FLAG INDICATING SK IN REVRSE
3110  012136  005743                          TST     -(R3)
3111  012140  005743          12$:    TST     -(R3)                   ;POSITION PTR TO NXT SK IN REV
3112  012142  022703  001370                  CMP     #SEEK0-2,R3     ;DONE WITH ALL?
3113  012146  001251                          BNE     1$              ;IF NOT, DO NXT ONE
3114
3115
3116
3117                                  ;;*****************************************************************
3118                                  ;*TEST 27       CHECK DRIVE RESET FROM LAST CYLINDER
3119                                          ;*THE HEADS ARE POSITIONED ON THE LAST CYLINDER (DOING
3120                                          ;*AN IMPLIED SEEK-READ).  THEN A DRIVE RESET IS ISSUED.
3121                                          ;*IT'S CHECKED IF THE HEADS WERE BROUGHT BACK TO 0 BY
3122                                          ;*DOING A 1 WORD READ & CHECKING THAT THE CORRECT WORD
3123                                          ;*WAS RECEIVED. IF TESTING ON SIMULATOR THIS TEST IS SKIPPED.
3124                                  ;;*****************************************************************
3125  012150  000004                  TST27:  SCOPE
3126  012152  012737  000005  001206          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
3127  012160  005737  001344                  TST     SIMUL           ;R U ON A SIMULATOR?
3128  012164  001124                          BNE     TST30           ;;YES, EXIT
3129  012166  013701  001332                  MOV     RKCS,R1
3130  012172  104413                          CNT.RESET               ;GO, DO CONTROL RESET
3131                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
3132                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
3133                                                                  ;ISSUED AND AFTER A CERTAIN TIME
3134                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
3135                                                                  ;AN ERROR IS REPORTED. NOTE THAT
3136                                                                  ;THE PC IN ERROR MESSAGE IS THE
3137                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
3138                                                                  ;THIS IS A VERY BASIC ERR 3 IF IT
3139                                                                  ;OCCURS GO BACK TO TEST 10
3140  012174  005000                          CLR     R0
3141  012176  012703  033240                  MOV     #OUTBUF,R3      ;ADRES WHERE DATA WILL BE READ INTO
3142  012202  013704  001350                  MOV     DRIVAD,R4
3143  012206  010405                          MOV     R4,R5
3144  012210  052705  014500                  BIS     #14500,R5       ;SET CYL ADRES=312 (OCTAL)
3145  012214  010577  167120                  MOV     R5,@RKDA        ;ADRES THE DRIVE, LAST CYLINDER
3146  012220  012777  177777  167106          MOV     #-1,@RKWC       ;READ 1 WORD
```

```
3147  012226  010377  167104                MOV     R3,@RKBA        ;INTO THIS MEMORY ADRES
3148
3149  012232  012711  000005                MOV     #5,@R1          ;READ, GO
3150
3151  012236  005000                        CLR     R0
3152  012240  104414            1$:         CNT.RDY                 ;THIS IS A CALL FOR CN.RDY ROUTINE
3153                                                                 ;WHICH WAITS FOR CNTRL RDY TO SET.
3154                                                                 ;A RETURN IS MADE AFTER CNTRL RDY
3155                                                                 ;SETS. IF WITHIN A CERTAIN TIME
3156                                                                 ;CNTRL RDY DOESN'T SET AN ERROR
3157                                                                 ;MESSAGE IS GIVEN. WAITING TIME
3158                                                                 ;883 MS FOR 11/20, 175 MS FOR 11/45
3159  012242  020513            2$:         CMP     R5,@R3          ;WAS THE CORRECT WORD READ?
3160  012244  001407                        BEQ     3$              ;YES, SEEK TO 312 WAS DONE CORRECTLY5,@
3161  012246  010537  001162                MOV     R5,$REG0        ;GET EXPCTD WORD
3162  012252  011337  001164                MOV     @R3,$REG1       ;GET WORD RECVD
3163  012256  010537  001166                MOV     R5,$REG2        ;GET DSK-ADRES FROM WHERE WORD WAS READ
3164  012262  104044                        ERROR   44              ;DID NOT READ BACK CORRECT WOPD FROM
3165                                                                 ;LAST CYL, SEC 0. IF TEST 45 & 46
3166                                                                 ;WERE SUCCESSFULLY DONE THIS
3167                                                                 ;ERROR MEANS THAT IMPLIED SEEK
3168                                                                 ;TO CYL 312 COULD NOT B DONE
3169  012264  012711  000015            3$:         MOV     #15,@R1         ;DRIVE RESET, GO
3170  012270  104414                        CNT.RDY                 ;THIS IS A CALL FOR CN.RDY ROUTINE
3171                                                                 ;WHICH WAITS FOR CNTRL RDY TO SET.
3172                                                                 ;A RETURN IS MADE AFTER CNTRL RDY
3173                                                                 ;SETS. IF WITHIN A CERTAIN TIME
3174                                                                 ;CNTRL RDY DOESN'T SET AN ERFOR
3175                                                                 ;MESSAGE IS GIVEN. WAITING TIME
3176                                                                 ;883 MS FOR 11/20, 175 MS FOR 11/45
3177  012272  005000                        CLR     R0
3178  012274  032777  000100  167024    4$:         BIT     #100,@RKDS      ;DID R/W/S RDY SET?
3179  012302  001011                        BNE     5$              ;YES, BRANCH
3180  012304  012702  177763                MOV     #-15,R2         ;IF U R ON A SLOWER MACHINE
3181  012310  005202                        INC     R2              ;& DO NOT NEED SUCH A LARGE MACHINE
3182  012312  001376                        BNE     .-2             ;TIME LOOP, CHANGE THESE 3
3183                                                                 ;INSTRUCTIONS TO 'NOP' THE
3184                                                                 ;LOOP TIME WILL BE REDUCED
3185                                                                 ;TO 1100 MS
3186                                                                 ;
3187                                                                 ;THE TOTAL TIME FOR THE ABOVE
3188                                                                 ;LOOPS (W/O PUTTING 'NOP'S) IS
3189                                                                 ;5304 MS FOR 11/20 AND
3190                                                                 ;1061 MS FOR 11/45 WITH MOS
3191                                                                 ;OR BIPOLAR MEMORY
3192  012314  005200                        INC     R0              ;WAITED LONG?
3193  012316  001366                        BNE     4$              ;IF NOT, LUP BAK & WAIT
3194                                                                 ;IF YES, ERROR
3195  012320  004737  020702                JSR     PC,GT4RG        ;GET RKCS,ER,DS,DA
3196  012324  104026                        ERROR   26              ;R/W/S RDY DID NOT SET AFTER
3197                                                                 ;DOING DRIVE RESET
3198  012326  032711  140000            5$:         BIT     #140000,@R1     ;DID HE OR ERR BIT SET?
3199  012332  001403                        BEQ     6$              ;IF NOT, BRANCH
3200
3201  012334  004737  020702                JSR     PC,GT4RG        ;GET RKCS,ER,DS,DA FOR ERROR MESSAGE
3202  012340  104022                        ERROR   22              ;HE OR ERR BIT SET ON DOING DRIVE
```

```
3203
3204  012342  005205                  6$:     INC     R5              ;RESET FROM LAST CYLINDER
3205  012344  020577  166770                  CMP     R5,@RKDA        ;POSITION R5 TO EXPCTD RKDA
3206  012350  001406                          BEQ     7$              ;DID THE CYL ADRES BITS IN RKDA GET CHANGED?
3207  012352  010537  001162                  MOV     R5,$REG0        ;NO, BRANCH
3208  012356  017737  166756  001164          MOV     @RKDA,$REG1     ;GET EXPCTD RKDA
3209  012364  104034                          ERROR   54              ;GET RKDA RECVD
3210                                                                  ;CYLINDER ADRES BITS IN RKDA
3211                                                                  ;GOT CHANGED AFTER
3212  012366  012777  177777  166740  7$:     MOV     #-1,@RKWC       ;DRIVE RESET, FROM LAST CYLINDER
3213  012374  010377  166736                  MOV     R3,@RkBA        ;READ 1 WORD
3214  012400  010477  166734                  MOV     R4,@RKDA        ;INTO THIS ADRES
3215                                                                  ;FROM THIS DSK ADRES-CYL 0, SEC 0
3216  012404  012711  000005                  MOV     #5,@R1          ;READ, GO
3217
3218  012410  005000                          CLR     R0
3219  012412  104414                  8$:     CNT.RDY                 ;THIS IS A CALL FOR CN.RDY ROUTINE
3220                                                                  ;WHICH WAITS FOR CNTRL RDY TO SET.
3221                                                                  ;A RETURN IS MADE AFTER CNTRL RDY
3222                                                                  ;SETS. IF WITHIN A CERTAIN TIME
3223                                                                  ;CNTRL RDY DOESN'T SET AN ERROR
3224                                                                  ;MESSAGE IS GIVEN. WAITING TIME
3225                                                                  ;883 MS FOR 11/20, 175 MS FOR 11/45
3226  012414  020413                  9$:     CMP     R4,@R3          ;WAS THE CORRECT WORD READ?
3227  012416  001407                          BEQ     TST30           ;YES, EXIT
3228  012420  010437  001162                  MOV     R4,$REG0        ;GET EXPCTD WORD
3229  012424  011337  001164                  MOV     @R3,$REG1       ;GET WORD RECVD
3230  012430  010437  001166                  MOV     R4,$REG2        ;GET DISK ADRES WHERE ERROR OCCURED
3231  012434  104044                          ERROR   44              ;DID NOT READ CORRECT WORD FROM
3232                                                                  ;CYL 0, SEC 0. IF TEST 45 & 46
3233                                                                  ;WERE SUCCESSFULLY DONE THIS
3234                                                                  ;ERROR COULD MEAN THAT DRIVE-RESET
3235                                                                  ;DID NOT BRING HEADS BACK TO 0.
3236
3237
3238                                  ;;*****************************************************************
3239                                  ;*TEST 30        'WRITE' - 256 WORD BLOCK ON SECTOR 0, CYLINDER 0
3240                                          ;THE TEST BELOW SHOULD BE CONSIDERED AS A SET UP PHASE FOR
3241                                          ;THE FOLLOWING TEST.  IT WRITES A BLOCK OF 256 WORDS IN
3242                                          ;SECTOR 0, CYLINDER 0 WITH A SPECIFIC PATTERN AND THIS WRITTEN
3243                                          ;BLOCK WILL BE MADE USE OF IN THE NEXT TEST TO CHECK
3244                                          ;OUT 'WRITE-CHECK' AND 'READ CHECK' FUNCTIONS.
3245                                  ;;*****************************************************************
3246  012436  000004                  TST30:  SCOPE
3247  012440  104413                          CNT.RESET               ;GO, DO CONTROL RESET
3248                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
3249                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
3250                                                                  ;ISSUED AND AFTER A CERTAIN TIME
3251                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
3252                                                                  ;AN ERROR IS REPORTED. NOTE THAT
3253                                                                  ;THE PC IN ERROR MESSAGE IS THE
3254                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
3255                                                                  ;THIS IS A VERY BASIC ERR& IF IT
3256                                                                  ;OCCURS GO BACK TO TEST 10
3257  012442  104421                          TST.SIN                 ;CHECK IF SIN IS SET, IF SET
3258                                                                  ;DO DRIVE RESET TO CLEAR IT
```

```
3259  012444  013704  001332                  MOV     RKCS,R4
3260                                                    ;THE FOLLOWING CODE IS FOR SETTING
3261                                            ;UP THE I/O BUFFER IN MEMORY (STARTING AT
3262                                            ;OUTBUF), WITH A PARTICULAR 256 WORD PATTERN.
3263                                            ;STARTING FROM THE FIRST WORD IN THE BUFFER
3264                                            ;THE LO BYTE WILL BE A COUNT PATTERN
3265                                            ;FROM 0 TO 255 (DECIMAL), WHEREAS THE
3266                                            ;HI-BYTE WILL BE THE COMPLEMENT OF LO BYTE,
3267                                            ;A DECREASING COUNT PATTERN FROM 255 TO 0.
3268                                            ;I.E.THE BUFFER WILL LOOK LIKE:
3269                                            ;OUTBUF         (1 111 111 1 00 000 000)
3270                                            ;OUTBUF+2       (1 111 111 0 00 000 001)
3271                                            ;
3272                                            ;LAST WORD      (0 000 000 0 11 111 111)
3273
3274  012450  012700  033240                  MOV     #OUTBUF,R0
3275  012454  012701  177401                  MOV     #177401,R1      ;PATTERN GENERATING NUMBER
3276  012460  012702  177400                  MOV     #-400,R2        ;SET UP COUNT FOR 256 WORDS
3277  012464  012703  177400                  MOV     #177400,R3      ;SET UP THE FIRST PATTERN TO B WRITTEN
3278
3279  012470  010320                  1$:     MOV     R3,(R0)+        ;SET UP FIRST WORD IN I/O BUFFER
3280  012472  005202                          INC     R2              ;INCREMENT COUNT
3281  012474  060103                          ADD     R1,R3           ;SET UP NEXT WORD PATTERN
3282  012476  010320                          MOV     R3,(R0)+        ;WRITE IT IN NXT I/O BUFFER WORD
3283  012500  005202                          INC     R2              ;HAVE U WRITTEN ALL 256 WORDS
3284  012502  001374                          BNE     1$              ;IF NOT GO & WRITE NEXT PATTERN
3285
3286  012504  012777  177400  166622          MOV     #-400,@RKWC     ;WRITE 256 WORDS
3287  012512  012777  033240  166616          MOV     #OUTBUF,@RKBA   ;STARTING FROM THIS BUS ADRES
3288  012520  013777  001350  166612          MOV     DRIVAD,@RKDA    ;TO THIS DISK ADRES, CYL 0, SEC 0
3289
3290  012526  012714  000003                  MOV     #3,@R4          ;WRITE, GO
3291
3292  012532  105714                  2$:     TSTB    @R4             ;WAS CNTRL RDY CLEARED AS GO WAS SET?
3293  012534  100003                          BPL     3$-2            ;YES, BRANCH
3294  012536  004737  020710                  JSR     PC,GT3RG        ;GET RKCS, ER, DS
3295  012542  104030                          ERROR   30              ;CNTRL RDY DID NOT CLEAR AS GO WAS SET
3296                                                                  ;TO 'WRITE'
3297
3298  012544  005002                  3$:     CLR     R2
3299  012546  105777  166560                  TSTB    @RKCS           ;DID CNTRL RDY SET?
3300  012552  100411                          BMI     4$              ;YES, BRANCH
3301  012554  005202                          INC     R2              ;WAITED LONG ENOUGH?
3302  012556  001373                          BNE     3$              ;IF NOT, LUP BAK & WAIT
3303                                                                  ;IF YES, ERROR
3304  012560  004737  020702                  JSR     PC,GT4RG        ;GO, GET RKCD, ER, DS, DA
3305  012564  013737  001350  001202          MOV     DRIVAD,$REG10   ;GET THE STARING ADRES
3306  012572  104416                          BRKDA4                  ;BREAK CONTENTS OF $REG10 INTO
3307                                                                  ;DRV #, CYL, SUR, SEC #
3308  012574  104031                          ERROR   31              ;CNTRL RDY DID NOT SET ON COMPLETION
3309                                                                  ;OF WRITE OF 256 WORDS ON CYL 0, SEC 0
3310                                                                  ;'RKDA' IN EROR MSGE GIVES THE
3311                                                                  ;CONTENTS OF RKDA AT THE TIME OF EROR
3312                                                                  ;WRITE WAS DONE STARTING AT <DSK-ADRES>
3313                                                                  ;INDICATED IN EROR MSGE
3314  012576  004737  021142          4$:     JSR     PC,CHKHE        ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
```

```
                                                ;IF YES, RETURN HERE
012602  104032                  ERROR   32      ;HE OR ERR BIT SET ON DOING WRITE OF
                                                ;256 WORDS ON CYL 0, SEC 0
                                                ;WRITE WAS DONE STARTING AT (DSK-ADRES)
                                                ;INDICATED IN EROR MSGE
                                                ;'RKDA' IN EROR MSGE GIVES THE
                                                ;CONTENTS OF RKDA AT THE TIME OF ERROR
012604  020077  166526          5$:     CMP     R0,@RKBA        ;DID RKBA INCREMENT CORRECTLY?
012610  001406                  BEQ     6$              ;YES, BRANCH
012612  010037  001162          MOV     R0,$REG0        ;GET EXPCTD RKBA
012616  017737  166514  001164  MOV     @RKBA,$REG1     ;GET RkBA RECVD
012624  104035                  ERROR   35              ;RKBA DID NOT INCREMENT CORRECTLY
                                                ;(BY 1000 OCTAL BYTES, AFTER WRITE
                                                ;OF 400 (OCTAL) WORDS ON SEC 0, CYL 0
012626  004737  021224          6$:     JSR     PC,CHKWC        ;CHECK IF RKWC OVERFLOWED TO 0,
                                                ;IF NOT RETURN HERE.
012632  104034                  ERROR   34              ;RKWC DID NOT OVERFLOW, AFTER A
                                                ;WRITE OF 256 WORDS ON CYL 0, SEC 0
012634  004737  021170          7$:     JSR     PC,CHKDA        ;CHECK IF RKDA INCREMENTED CORRECTLY,
                                                ;IF NOT RETURN HERE
012640  104033                  ERROR   33              ;RKDA DID NOT INCREMENT BY 1 AFTER
                                                ;A WRITE OF 256 WORDS IN CYL 0, SEC 0
012642  004737  021250          8$:     JSR     PC,CHKER        ;CHECK IF ANY BIT RKER IS SET
                                                ;IF YES RETURN HERE.
012646  104036                  ERROR   36              ;RKER BIT SET ON DOING WRITE ON
                                                ;CYLINDER 0, SECTOR 0
012650  022714  000202          9$:     CMP     #202,@R4        ;DOES RKCS STILL CONTAIN THE WRITE BITS?
012654  001406                  BEQ     TST31           ;YES, EXIT
012656  012737  000202  001162  MOV     #202,$REG0      ;GET EXPECTED RKCS
012664  011437  001164          MOV     @R4,$REG1       ;GET RKCS RECVD
012670  104024                  ERROR   24              ;RKCS DID NOT CONTAIN THE 'WRITE'
                                                ;BITS AFTER THE FUNCTION WAS DONE.


                        ;;*****************************************************************
                        ;*TEST 31       CHECK THAT WRITE WAS DONE CORRECTLY
                                ;*THIS TEST CHECKS IF THE 'WRITE' OF 256 WORDS DONE IN PREVIOUS
                                ;*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
                        ;;*****************************************************************
012672  000004          TST31:  SCOPE
012674  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
```

```
                                TST.SIN                 ;CHECK IF SIN IS SET, IF SET
                                                        ;DO DRIVE RESET TO CLEAR IT
                                MOV     #-400,R0        ;SET COUNT FOR 400 WORDS
                                MOV     #OUTBUF,R1      ;TO BE CLEARED IN THE BUFFER
                        9$:     CLR     (R1)+           ;CLR THE 400 WORD BUFFER
                                INC     R0              ;STARTING AT 'OUTBUF'
                                BNE     8$
                                CLR     R0
                                MOV     #-400,@RKWC     ;READ 256 WORDS
                                MOV     #OUTBUF,@RKBA   ;INTO THIS ADRES
                                MOV     DRIVAD,@RKDA    ;STARTING FROM THIS DISK ADRES

                                MOV     #5,@RKCS        ;READ, GO

                        1$:     TSTB    @RKCS           ;DID CNTRL RDY SET?
                                BMI     2$              ;YES, BRANCH
                                INC     R0              ;WAITED LONG ENOUGH?
                                BNE     1$              ;IF NOT, LUP BAK & WAIT
                                                        ;ERROR, IF YES
                                JSR     PC,GT4RG        ;GO, GET RKCD, ER, DS, DA
                                MOV     DRIVAD,$REG10   ;GET THE STARTING ADRES
                                BRKDA4                  ;GO TO 'BDA4' & BREAK CONTENTS OF
                                                        ;$REG10 INTO DRV #, CYL, SUR, SEC BITS
                                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
                        2$:     BIT     #1000,@RKDS     ;IS SIN SET?
                                BNE     TST32           ;;IF YES, EXIT
                        5$:     MOV     #-400,R1
                                MOV     #177777,R2
                                MOV     #OUTBUF,R3
                                MOV     #-5,R5
                        6$:     ADD     #177401,R2
                                CMP     R2,(R3) ;WAS THE READ WORD SAME AS THE WORD
                                                        ;THAT WAS SUPPOSE TO BE WRITTEN
                                BEQ     7$              ;YES, BRANCH
                                                        ;NO, ERROR
                                MOV     R1,$REG0        ;GET THE # OF WORD
                                ADD     #401,$REG0      ;THAT IS IN ERROR (EXAMPLE=1,2--376,377,400)
                                MOV     R2,$REG1        ;GET EXPCTD WORD (THAT WAS SUPPOSED TO
                                                        ;BE WRITTEN)
                                MOV     (R3),$REG2      ;GET WORD RECVD (THAT WAS READ BAK)
                                ERROR   55              ;DID NOT READ BACK WORD THAT WAS SUPPOSED
                                                        ;TO HAVE BEEN WRITTEN PREVIOUSLY. POSITION
                                                        ;OF WORD IN ERROR IS AS INDICATED BY
                                                        ;WORD # ($REG0), SEC 0, CYL 0
                                INC     R5
                                BEQ     TST32           ;;EXIT
                        7$:     TST     (R3)+           ;INCREMENT POINTER TO NXT WORD (THAT
                                                        ;WAS READ BACK)
                                INC     R1              ;HAVE U CHKD ALL 256 WORDS?
                                BNE     6$              ;IF NOT, LUP BAK & CHK THE NXT WORD
                                                        ;IF YES, EXIT
```

```
                                ;;*****************************************************************
                                ;*TEST 32       CHECK 'READ CHECK' FUNCTION - C   ER 0, SECTOR 0
                                ;*THIS TEST CHECKS OUT THE BASIC :READ CHECK' LOGIC, USING THE DATA BLOCK
                                ;*'CYLINDER, SECTOR 0) WRITTEN IN A PREVIOUS TEST.  HENCE THE TEST WHICH
                                ;*WRITES THE DATA BLOCK SHOULD BE DONE PRIOR TO THIS TEST.
                                ;;*****************************************************************
013076  000004                TST32:  SCOPE
013100  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
013102  104421                        TST.SIN                 ;CHECK IF SI'N IS SET, IF SET
                                                                ;DO DRIVE RESET TO CLEAR IT
013104  013701  001332                MOV     RKCS,R1
013110  013702  001334                MOV     RKWC,R2
013114  013703  001340                MOV     RKDA,R3
013120  013704  001336                MOV     RKBA,R4
013124  012737  052525  033240        MOV     #52525,OUTBUF
013132  012712  177400                MOV     #-400,@R2       ;READ CHECK 256 WORDS
013136  013713  001350                MOV     DRIVAD,@R3      ;STARTING FROM CYL 0, SECTOR 0
013142  012714  033240                MOV     #OUTBUF,@R4
013146  012711  000013                MOV     #13.,@R1        ;READ CHECK, GO

013152  105711                1$:     TSTB    @R1             ;DID CNTRL RDY GET CLEARED AS GO WAS SET?
013154  100003                        BPL     2$              ;YES, BRANCH
013156  004737  020710                JSR     PC,GT3RG        ;GET RKCS, ER, DS
013162  104030                        ERROR   30              ;CNTRL RDY DID NOT CLEAR AS GO
013164  104412                2$:     CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
                                                                ;IF SO, SKIP THE EROR MESSAGE.
                                                                ;WAS SET TO 'READ CHECK'
013166  104056                        ERROR   56              ;CNTRL RDY DID NOT SET ON DOING
                                                                ;'READ CHECK' FROM CYL 0, SEC 0
013170  032711  140000        3$:     BIT     #140000,@R1     ;DID 'ERR' OR 'HE' BIT SET?
013174  001403                        BEQ     4$              ;NO, BRANCH
013176  004737  020710                JSR     PC,GT3RG        ;GO, GET RKCS,ER,DS FOR ERROR MESSAGE
013202  104057                        ERROR   57              ;'ERR' OR 'HE' BIT SET ON DOING
                                                                ;'READ CHECK' ON CYLINDER 0, SEC 0
013204  032777  000002  166116 4$:     BIT     #2,@RKER        ;DID 'CSE' BIT SET IN RKER?
013212  001404                        BEQ     5$              ;NO, BRANCH
013214  017737  166110  001162        MOV     @RKER,$REG0     ;GET RKER
013222  104060                        ERROR   60              ;SOFT ERROR - CSE - ON DOING 'READ
                                                                ;CHECK' ON CYLINDER 0, SECTOR 0
                                                                ;U SHOULD HAVE GOT ERRROR 102 ALSO
013224  005712                5$:     TST     @R2             ;DID WORD COUNT OVERFLOW TO 0?
013226  001405                        BEQ     6$              ;YES, BRANCH
013230  011237  001162                MOV     @R2,$REG0       ;GET RKWC
013234  011137  001164                MOV     @R1,$REG1       ;GET RKCS
013240  104061                        ERROR   61              ;WORD COUNT DID NOT OVERFLOW
                                                                ;ON DOING 'READ CHK' ON CYL 0, SEC 0
```

```
3483  013242  013702  001358          65$:    MOV     DRIVAD,R2       ;RKDA SHOULD INCREMENT
3484  013246  005202                          INC     R2              ;TO THIS AFTER 'RD CHK' IS DONE
3485  013250  020213                          CMP     R2,@R3          ;DID RKDA INCREMENT CORRECTLY?
3486  013252  001405                          BEQ     7$
3487  013254  010237  001162                  MOV     R2,$REG0        ;GET EXPCTD RKDA
3488  013260  011337  001164                  MOV     @R3,$REG1       ;GET RKDA RECVD
3489  013264  104062                          ERROR   62              ;RKDA DID NOT INCREMENT CORRECTLY
3490                                                                  ;(BY 1) ON DOING 'READ CHK' ON
3491                                                                  ;CYL 0, SEC 0
3492  013266  022714  033240          7$:     CMP     #OUTBUF,@R4     ;DID RKBA GET CHANGED?
3493  013272  001406                          BEQ     8$              ;NO, BRANCH (RKBA WON'T CHANGE, NO NPR'S,
3494  013274  012737  033240  001162          MOV     #OUTBUF,$REG0   ;GET EXPCTD RKBA
3495  013302  011437  001164                  MOV     @R4,$REG1       ;GET RKBA RECVD
3496  013306  104063                          ERROR   63              ;RKBA CHANGED AFTER DOING 'READ CHK'
3497                                                                  ;ON CYLINDER 0, SECTOR 0. SHOULD
3498                                                                  ;NOT CHANGE, FOR, NO NPR'S.
3499  013310  022737  052525  033240  8$:     CMP     #52525,OUTBUF   ;'OUTBUF' SHOULD STILL CONTAIN THE
3500                                                                  ;SAME WORD AS IT DID BEFORE 'RD CHK'
3501                                                                  ;NOTE THAT AT THE BEGINING OF THIS TEST
3502                                                                  ;52525 WAS WRITTEN INTO 'OUTBUF'
3503  013316  001412                          BEQ     TST33           ;;YES, EXIT
3504                                                                  ;REPORT ERROR IF 'OUTBUF' CHANGED
3505  013320  012737  033240  001162          MOV     #OUTBUF,$REG0   ;GET ADRES OF OUTBUF
3506  013326  012737  052525  001164          MOV     #52525,$REG1    ;GET EXPCTD WORD IN 'OUTBUF'
3507  013334  013737  033240  001166          MOV     OUTBUF,$REG2    ;GET WORD FOUND IN 'OUTBUF'
3508  013342  104064                          ERROR   64              ;AS MENTIONED ABOVE, IF 'WRITE' OF
3509                                                                  ;256 WORD DATA BLOCK WAS DONE
3510                                                                  ;CORRECTLY BEFORE, THEN THIS ERROR
3511                                                                  ;COULD MEAN THAT AN NPR WAS DONE
3512                                                                  ;ON 'READ CHECK'.
3513
3514                                ;;*****************************************************************
3515                                ;*TEST 33       CHECK THE 'WRITE CHECK' FUNCTION - ON CYLINDER 0, SECTOR 0
3516                                        ;*THIS TEST CHECKS OUT THE BASIC 'WRITE CHECK' LOGIC, USING THE 256
3517                                        ;*WORD DATA BLOCK (SECTOR 0, CYLINDER 0) WRITTEN IN A PREVIOUS
3518                                        ;*TEST. THE BUFFER IN MEMORY, USED FOR COMPARISON OF DATA, IS THE
3519                                        ;*ONE STARTING AT 'OUTBUF'. HENCE THE TEST WHICH WRITES THE
3520                                        ;*256 WORD BLOCK ON THE DISK (AS WELL AS CREATING THE 256
3521                                        ;*256 WORD MEMORY BUFFER) SHOULD BE DONE BEFORE THIS TEST.
3522                                ;;*****************************************************************
3523  013344  000004                TST33:  SCOPE
3524  013346  104413                        CNT.RESET               ;GO, DO CONTROL RESET
3525                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
3526                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
3527                                                                  ;ISSUED AND AFTER A CERTAIN TIME
3528                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
3529                                                                  ;AN ERROR IS REPORTED. NOTE THAT
3530                                                                  ;THE PC IN ERROR MESSAGE IS THE
3531                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
3532                                                                  ;THIS IS A VERY BASIC ERR& IF IT
3533                                                                  ;OCCURS GO BACK TO TEST 10
3534  013350  104421                        TST.SIN                 ;CHECK IF SIN IS SET, IF SET
3535                                                                  ;DO DRIVE RESET TO CLEAR IT
3536  013352  013701  001332                MOV     RKCS,R1
3537  013356  012700  177400                MOV     #-400,R0
3538  013362  012702  033240                MOV     #OUTBUF,R2
```

```
3539  013366  012703  177777            MOV     #177777,R3
3540  013372  062703  177401      1$:   ADD     #177401,R3
3541  013376  010322                    MOV     R3,(R2)+
3542  013400  005200                    INC     R0
3543  013402  001373                    BNE     1$
3544  013404  012777  177400  165722    MOV     #-400,@RKWC     ;WRITE CHECK 256 WORDS
3545  013412  012777  033240  165716    MOV     #OUTBUF,@RKBA   ;STARTING AT THIS BUS ADRES
3546  013420  013777  001350  165712    MOV     DRIVAD,@RKDA    ;WITH THIS DISK DATA BLOCK (CYL 0, SEC 0,
3547  013426  012711  000007            MOV     #7,@R1          ;WRITE CHECK, GO
3548
3549  013432  005000                    CLR     R0              ;GIVE SOME TIME
3550  013434  105711            2$:     TSTB    @R1             ;DID CNTRL RDY CLEAR AS GO WAS SET?
3551  013436  100003                    BPL     3$              ;YES BRANCH
3552  013440  004737  020710            JSR     PC,GT3RG        ;GET RKCS, ER, DS
3553  013444  104030                    ERROR   30              ;CNTRL RDY DID NOT CLEAR AS GO WAS
3554                                                            ;SET TO DO WRITE CHECK
3555  013446  104412            3$:     CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
3556                                                            ;IF SO, SKIP THE EROR MESSAGE.
3557  013450  104065                    ERROR   65              ;CNTRL RDY DID NOT SET AFTER
3558                                                            ;COMPLETING WRITE CHECK ON
3559                                                            ;CYLINDER 0, SECTOR 0
3560  013452  032711  140000      4$:   BIT     #140000,@R1     ;DID HE OR ERR BIT SET
3561  013456  001403                    BEQ     5$              ;NO, BRANCH
3562  013460  004737  020710            JSR     PC,GT3RG        ;GO GET RKCS ER DS FOR ERROR MESSAGE
3563  013464  104066                    ERROR   66              ;HE OR ERR BIT SET ON DOING WRITE
3564                                                            ;CHK ON CYLINDER 0, SEC 0
3565  013466  032777  000001  165634  5$:   BIT     #1,@RKER        ;DID WCE SET IN RKER?
3566  013474  001403                    BEQ     6$              ;NO, BRANCH
3567  013476  004737  020710            JSR     PC,GT3RG        ;YES GET RKCS, ER, DS
3568  013502  104067                    ERROR   67              ;WCE ON WRITE CHECK OF CYL 0, SEC 0
3569                                                            ;NOTE THAT IF A PREVIOUS TEST
3570                                                            ;& THEN COMPARED WITH MEMORY BUFFER
3571                                                            ;TO SEE IF IT WAS WRITTEN CORRECT WAS
3572                                                            ;DONE RIGHT BEFORE, THIS ERROR SHOULD NOT
3573                                                            ;HAPPEN UNLESS THERE IS A FAULT IN THE
3574                                                            ;COMPARING LOGIC OF 'WRT CHK'
3575  013504  005777  165624      6$:   TST     @RKWC           ;DID RKWC OVERFLOW?
3576  013510  001406                    BEQ     7$              ;YES, BRANCH
3577  013512  017737  165616  001162    MOV     @RKWC,$REG0     ;NO, GET RKWC
3578  013520  011137  001164            MOV     @R1,$REG1       ;GET RKCS
3579  013524  104061                    ERROR   61              ;RKWC DID NOT OVERFLOW AFTER
3580                                                            ;WRITE CHECK ON CYL 0, SEC 0
3581  013526  013704  001350      7$:   MOV     DRIVAD, R4      ;RKDA SHOULD INCREMENT
3582  013532  005204                    INC     R4              ;TO THIS AFTER WRT CHK
3583  013534  020477  165600            CMP     R4,@RKDA        ;DID RKDA INCREMENT CORRECTLY?
3584  013540  001406                    BEQ     8$              ;YES, BRANCH
3585  013542  010437  001162            MOV     R4,$REG0        ;NO, GET EXPCTD RKDA
3586  013546  017737  165566  001164    MOV     @RKDA,$REG1     ;GET RKDA RECVD
3587  013554  104070                    ERROR   70              ;RKDA DID NOT INCREMENT CORRECTLY
3588                                                            ;(BY 1 SECTOR) AFTER WAT CHK ON SEC 0, CYL 0
3589  013556  022777  034240  165552  8$:   CMP     #OUTBUF+1000,@RKBA ;DID RKBA INCREMENT CORRECTLY?
3590  013564  001407                    BEQ     9$              ;YES, EXIT
3591  013566  012737  034240  001162        MOV     #OUTBUF+1000,$REG0 ;GET EPCTD RKBA
3592  013574  017737  165536  001164        MOV     @RKBA,$REG1     ;GET RKBA RECVD
3593  013602  104071                        ERROR   71              ;RKBA DID NOT INCREMENT CORRECTLY
3594                                                                ;(BY 1000 BYTES) AFTER A WRT CHK
```

```
                                                        ;OF 256 WORDS ON CYL 0, SEC 0
013604  022711  000206          3$:     CMP     #206,@R1        ;DOES RKCS STILL CONTAIN THE WRT CHK BITS?
013610  001406                          BEQ     TST34           ;;YES, BRANCH
013612  012737  000206  001162          MOV     #206,$REG0      ;NO, GET EXPCTD RKCS
013620  011137  001164                  MOV     @R1,$REG1       ;GET RKCS RECVD
013624  104024                          ERROR   24              ;RKCS BITS CHANGED AFTER WRT CHK
                                                        ;WAS DONE
                                ;;**************************************************************
                                ;*TEST 34       CHECK THAT IBA INHIBITS INCREMENTING OF RKBA
                                        ;*THIS TEST CHECKS THAT THE BUS ADDRESS DOES NOT INCREMENT WHEN
                                        ;*THE IBA BIT IS SET. SEQUENCE OF OPERATIONS:
                                        ;*1) CLEAR OUT 256 WORD BUFFER IN MEMORY (OUTBUF)
                                        ;*2) READ FROM SECTOR 0, CYLINDER 0 THE 256 WORD BLOCK THAT WAS
                                        ;*WRITTEN IN A PREVIOUS TEST (NOTE: THAT TEST SHOULD HAVE BEEN
                                        ;*DONE BEFORE THIS). IBA BIT IS SET DURING READ BACK.
                                        ;*3) CHECK THAT RKBA DID NOT INCREMENT
                                        ;*4) CHECK THAT THE ENTIRE BLOCK WAS READ INTO THE SAME MEMORY
                                        ;*WORD (OUTBUF) & THE REST OF THE WORDS IN THAT BUFFER ARE 0
                                        ;*AS PREVIOUSLY CLEARED OUT.
                                ;;**************************************************************
013626  000004                  ^TST34: SCOPE
013630  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
013632  104421                          TST.SIN                 ;CHECK IF SIN IS SET, IF SET
                                                        ;DO DRIVE RESET TO CLEAR IT
013634  013701  001332                  MOV     RKCS,R1
013640  012700  177400                  MOV     #-400,R0        ;SET UP COUNT FOR 256 WORDS
013644  012702  033240                  MOV     #OUTBUF,R2
013650  010203                          MOV     R2,R3

013652  005023          1$:     CLR     (R3)+           ;CLEAR OUT THE 256
013654  005200                  INC     R0              ;WORD MEMORY BUFFER STARTING
013656  001375                  BNE     1$              ;AT 'OUTBUF'
013660  012777  177400  165446          MOV     #-400,@RKWC     ;READ BACK 256 WORDS
013666  010277  165444                  MOV     R2,@RKBA        ;INTO THIS BUS ADRES (IBA WILL B SET)
013672  013777  001350  165440          MOV     DRIVAD,@RKDA    ;FROM THIS DSK ADRES (SEC 0, CYL 0)
                                                        ;NOTE: SEC 0 HAS BEEN WRITTEN IN A
                                                        ;PREVIOUS TEST WITH A UNIQUE PATTERN
013700  012711  004005                  MOV     #4005,@R1       ;READ, GO, IBA SET

013704  005037  001362                  CLR     COUNT
013710  105711          2$:     TSTB    @R1             ;DID CNTRL RDY SET?
013712  100412                  BMI     3$              ;YES, BRANCH
013714  005237  001362                  INC     COUNT           ;WAITED LONG ENOUGH?
013720  001373                  BNE     2$              ;IF NOT, LUP BAK & WAIT
013722  004737  020702                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
013726  013737  001350  001202          MOV     DRIVAD,$REG10   ;GET THE STARTING ADRES
013734  104416                          BRKDA4                  ;BREAK CONTENTS OF $REG10
```

```
                                                                ;INTO DR #, CYL, SUR, SEC
013736  104045                          ERROR   45              ;CNTRL RDY DID NOT SET AFTER DOING
                                                                ;READ
013740  004737  021142          3$:     JSR     PC,CHKHE        ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
                                                                ;IF YES, RETURN HERE.
013744  104046                          ERROR   46              ;ERR BIT SET ON DOING READ FROM SEC 0,
                                                                ;CYL 0 (INDICATED  IN (DSK-ADRES))
                                                                ;'RKDA' IN EROR MSGE GIVES THE
                                                                ;CONTENTS OF RKDA AT THE TIME OF EROR

013746  020277  165364          4$:     CMP     R2,@RKBA        ;DID RKBA INCREMENT?
013752  001406                          BEQ     5$              ;OK IF NOT, BRANCH
013754  010237  001162                  MOV     R2,$REG0        ;GET EXPCTD RKBA
013760  017737  165352  001164          MOV     @RKBA,$REG1     ;GET RKBA RECVD
013766  104072                          ERROR   72              ;RKBA INCREMNTED WHEN IBA BIT WAS
                                                                ;SET, SHOULD NOT HAVE
013770  032777  001000  165330  5$:     BIT     #1000,@RKDS     ;IS SIN SET?
013776  001042                          BNE     TST35           ;;IF YES, EXIT
014000  012700  177400                  MOV     #-400,R0
014004  022712  000377                  CMP     #377,@R2        ;CHECK THAT THE FIRST WORD IN
                                                                ;'OUTBUF' IS 377 (LAST WORD OF SEC 0,
                                                                ;CYL 0). NOTE THAT READ WAS DONE
014010  001411                          BEQ     6$              ;INTO THIS SAME WRD WITH IBA SET
014012  012737  000377  001162          MOV     #377,$REG0       ;GET EXPCTD WORD (LAST WORD OF THE BUFFER
014020  011237  001164                  MOV     (R2),$REG1      ;GET WORD RECVD (LAST WRD FROM SEC 0)
014024  013737  001350  001166          MOV     DRIVAD,$REG2    ;DISK ADRES WHERE ERROR OCCURED
                                                                ;(SEC 0, CYL 0 LAST WORD)
                                                                ;DATA ERROR
014032  104044                          ERROR   44              ;THE FIRST WORD IN MEM BUFFER (OUTBUF)
                                                                ;SHOULD BE NON-ZERO & SHOULD CONTAIN
                                                                ;THE LAST WORD READ BACK FROM SEC 0
                                                                ;CYL 0,THIS DID NOT HAPPEN IF THE ERROR OCCURS
014034  005722                  6$:     TST     (R2)+           ;INCREMENT POINTER TO THE NXT WORD
014036  012705  177773                  MOV     #-5,R5          ;ALLOW ONLY 5 MESAGES FOR ERR 116
014042  005200                  7$:     INC     R0              ;CHKD ALL 256 WORDS IN THE BUFFER?
014044  001417                          BEQ     TST35           ;.YES, EXIT
014046  005722                          TST     (R2)+           ;IS THIS WORD 0?
014050  001774                          BEQ     7$              ;YES, LUP BAK & CHK THE NXT WORD?
014052  005037  001164                  CLR     $REG1           ;ERROR. GET EXPCTED WORD - 0
014056  014237  001166                  MOV     -(R2),$REG2     ;GET WORD THAT WAS FOUND IN THE BUFFER
014062  010004                          MOV     R0,R4
014064  062704  000401                  ADD     #401,R4
014070  010437  001162                  MOV     R4,$REG0

014074  104073                          ERROR   73              ;THIS 'WORD #' IN MEMORY BUFFER
                                                                ;SHOULD HAVE BEEN ZERO
                                                                ;THE 256 WORD BUFER (STARTING AT
                                                                ;OUTBUF) WAS CLEARED BEFORE READING
                                                                ;BAK SEC 0 INTO IT. SINCE THE IBA
                                                                ;BIT WAS SET DURING THE READ, ONLY
                                                                ;THE FIRST WORD OF (OUTBUF) SHOULD
                                                                ;HAVE CHANGED, THE REST OF THE WORDS
                                                                ;SHOULD BE STILL 0. IF THIS ERROR
                                                                ;OCCURS, 'WORD #' (OF THE BUFFER) AS
                                                                ;INDICATED IN THE EROR MESAGE) GOT
                                                                ;CHANGED WHEN READ WAS DONE FROM
                                                                ;THE DISK, INDICATING THAT WITH IBA
                                                                ;SET X-FER WAS NOT DONE INTO THE
```

```
                                                        ;SAME MEMORY LOCATION. 'WORD #'
                                                        ;IS OCTAL & SPECIFIES THE POSITION
                                                        ;IN THE BUFFER (FIRST WORD IS 'WORD #' 1)
014076  005205                        INC     R5
014100  001401                        BEQ     TST35           ;;EXIT
014102  000757                        BR      7$
                                ;
                                ;;*****************************************************************
                                ;*TEST 35        CHECK THAT RK11 INTERRUPTS WHEN IDE IS SET
                                        ;*THIS TEST CHECKS IF RK11 INTERRUPTS TO ITS DESIGNATED VECTOR
                                        ;*ADDRESS WHEN IDE BIT IS SET, WITH CONTROL READY SET & GO CLEAR.
                                        ;* IT IS NORMALLY 220, UNLESS IT HAS BEEN CHANGED. IF IT HAS BEEN
                                        ;*CHANGED RK11 WILL INTERRUPT TO 'RKVEC'. NOTE 'RKVEC' HAS
                                        ;*TO BE SET UP BY THE USER.
                                ;;*****************************************************************
014104  000004                TST35:  SCOPE
014106  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
014110  104421                        TST.SIN                 ;CHECK IF SIN IS SET, IF SET
                                                        ;DO DRIVE RESET TO CLEAR IT
014112  012746  000340                MOV     #340,-(SP)
014116  012746  014124                MOV     #64$,-(SP)
014122  000002                        RTI
014124                          64$:
014124  013701  001332                MOV     RKCS,R1
014130  013700  001402                MOV     RKVEC,R0        ;GET POINTER TO RK VECTOR ADRES
014134  012720  014170                MOV     #1$,(R0)+       ;SET UP INTERRUPT VECTOR FOR RK11
014140  012710  000340                MOV     #340,(R0)       ;SET PSW ON INTERRUPT
014144  105711                        TSTB    @R1             ;WAIT FOR CNTRL RDY TO SET
014146  100376                        BPL     .-2             ;
014150  012711  000100                MOV     #100,@R1        ;SET IDE BIT IN RKCS
014154  104420  000005                WAT.INT .5              ;WAIT FOR INTERRUPT, ATLEAST
                                                        ;37 US FOR 11/20, 7 US FOR 11/45
014160  011137  001162                MOV     @R1,$REG0       ;GET RKCS
014164  104074                        ERROR   74              ;RK11 DID NOT INTERRUPT WHEN IDE
                                                        ;WAS SET, WITH CNTRLE RDY SET & GO
                                                        ;CLEAR
014166  000400                        BR      1$
014170  022626                  1$:   CMP     (SP)+,(SP)+     ;RK11 INTERRUPTED CORRECTLY TO
                                                        ;THIS. RESTORE STACK POINTER
                                                        ;(FROM RK11 INTERRUPT)
014172  022626                        CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
                                                        ;(FROM  WAT.INT)
014174  012777  014210  165200        MOV     #2$,@RKVEC      ;IF THERE IS FAULTY POLLING OR INTERUPT
                                                        ; LOGIC SECOND INTERRUPT MIGHT OCCUR
014202  104420  000005                WAT.INT ,5              ;WAIT FOR INTERRUPT, IF ANY
                                                        ;DUE TO FAULTY LOGIC
```

```
3763  014206  000403                        BR      3$
3764
3765  014210  022626                2$:     CMP     (SP)+,(SP)+     ;RESTORE STACK PTR (FROM RK11 INTRUPT)
3766  014212  022626                        CMP     (SP)+,(SP)+     ;RESTORE STACK PTR (FROM WAT.INT)
3767  014214  104020                        ERROR   20              ;AN UNEXPECTED RK11 INTERRUPT
3768                                                                ;OCCURED. THERE SHOULD HAVE BEEN
3769                                                                ;ONLY 1 INTERRUPT (TO 1$ ABOVE)
3770  014216  012777  004526  165156  3$:     MOV     #BADINT,@RKVEC  ;RESTORE VECTOR ADRES FOR
3771                                                                ;UNEXPECTED RK11 INTERRUPT,
3772  014224  012746  000340                MOV     #340,-(SP)
3773  014230  012746  014236                MOV     #65$,-(SP)
3774  014234  000002                        RTI
3775  014236                        65$:
3776
3777
3778                                ;;**************************************************************
3779                                ;*TEST 36       CHECK THAT WITH IDE SET RK11 INTERRUPTS AFTER INTIATION & COMPLETION OF
3780                                ;*THIS TEST CHECKS THAT AN INTERRUPT FROM RK11 OCCURS AFTER
3781                                ;*A SEEK IS INITIATED WITH 'IDE' BIT SET, AND THEN A SECOND
3782                                ;*INTERRUPT OCCURS AFTER THE SEEK IS DONE. IT ALSO CHECKS THAT
3783                                ;*AFTER THE FIRST INTERRUPT 'SCP' BIT IS NOT SET, WHEREAS AFTER
3784                                ;*THE SECOND INTERRUPT 'SCP' IS SET.
3785                                ;*THIS TEST ALSO CHECKS A PART OF THE POLLING LOGIC.
3786                                ;;**************************************************************
3787  014236  000004                T36:    SCOPE
3788  014240  012737  000005  001206        MOV     #5,$TIMES       ;;DO 5 ITERATIONS
3789  014246  104413                        CNT.RESET               ;GO, DO CONTROL RESET
3790                                                                ;THIS IS A CALL FOR THE 'CNTRL-
3791                                                                ;RESET' ROUTINE. A CONTROL RESET IS
3792                                                                ;ISSUED AND AFTER A CERTAIN TIME
3793                                                                ;IF THE 'CNTRL RDY' DOES NOT SET
3794                                                                ;AN ERROR IS REPORTED. NOTE THAT
3795                                                                ;THE PC IN ERROR MESSAGE IS THE
3796                                                                ;PC WHERE 'CNT.RESET' IS LOCATED.
3797                                                                ;THIS IS A VERY BASIC ERR& IF IT
3798                                                                ;OCCURS GO BACK TO TEST 10
3799  014250  013700  001332                MOV     RKCS,R0
3800  014254  013777  001350  165056        MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE
3801  014262  004737  021412                JSR     PC,DRESET       ;GO, DO DRIVE RESET
3802  014266  104026                        ERROR   26              ;R/W/S RDY DIDN'T SET AFTER DOING
3803                                                                ;ABOVE DRIVE RESET
3804  014270  013701  001402          2$:   MOV     RKVEC,R1
3805  014274  012721  014340                MOV     #3$,(R1)+       ;SET UP VECTOR ADRES FOR RK11 INTERUPT
3806  014300  012711  000340                MOV     #340,(R1)       ;SET UP PSW ON INTERRUPT
3807  014304  052777  000040  165026        BIS     #40,@RKDA       ;ADRES  CYLINDER #1
3808  014312  012710  000111                MOV     #111,@R0        ;SEEK, GO WITH IDE SET
3809  014316  104420  000300                WAT.INT ,300            ;WAIT FOR THE DRIVE TO
3810                                                                ;INTERRUPT AFTER ADRES WAS RECVD
3811                                                                ;WAITING TIME= 1.4 MS FOR 11/20
3812                                                                ;280 US FOR 11/45
3813                                                                ;ERROR, IF INTERUPT DID NOT OCCUR
3814                                                                ;BY NOW
3815  014322  012777  004526  165052        MOV     #BADINT,@RKVEC  ;RESTORE UNEXPECTED RK11 INTERRUPT
3816  014330  011037  001162                MOV     @R0,$REGO       ;GET RKCS
3817  014334  104075                        ERROR   75              ;INTERRUPT DID NOT OCCUR AFTER
3818                                                                ;SEEK WAS INITIATED WITH IDE SET
```

```
3819  014336  000402                        BR      3$+4
3820  014340  022626                3$:     CMP     (SP)+,(SP)+     ;OK, IF RK11 INTERRUPTED TO THIS
3821                                                                ;RESTORE STACK POINTER (FROM RK11 INTERRUPT)
3822  014342  022626                        CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER (FROM
3823                                                                ;WAT.INT)
3824  014344  012777  014410  165030        MOV     #5$,@RKVEC      ;SET UP NEW VECTOR ADRES FOR RK11
3825  014352  032710  020000               BIT     #20000,@RD      ;IS SCP CLEAR
3826  014356  001403                        BEQ     4$              ;YES, BRANCH
3827  014360  011037  001162               MOV     @RD,$REGD       ;GET RKCS
3828  014364  104076                        ERROR   76              ;SCP SET BEFORE SEEK TO LAST
3829                                                                ;CYLINDER WAS DONE
3830  014366  104420  056700         4$:     WAT.INT ,56700          ;WAIT FOR DRIVE TO INTERRUPT
3831                                                                ;AFTER SEEK WAS COMPLETED
3832                                                                ;WAITING TIME=180 MS FOR 11/20
3833                                                                ;36 MS FOR 11/45
3834  014372  012777  004526  165002        MOV     #BADINT,@RKVEC  ;IT'S AN ERROR IF BY THIS TIME
3835                                                                ;INTERRUPT HAS NOT OCCURERED
3836  014400  004737  020710               JSR     PC,GT3RG                ;GO GET RKCS, ER, DS
3837  014404  104077                        ERROR   77              ;RK11 DID NOT INTERRUPT AFTER SEEK (TO
3838                                                                ;LAST CYLINDER) WAS DONE WITH IDE SET
3839  014406  000401                        BR      5$+2
3840  014410  022626                5$:     CMP     (SP)+,(SP)+     ;OK, IF RK11 INTERUPTED TO THIS AFTER
3841                                                                ;SEEK WAS COMPLETED. RESTORE
3842                                                                ;STACK POINTER (FROM RK11 INTERRUPT)
3843  014412  022626                        CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER (FROM
3844                                                                ;WAT.INT)
3845  014414  012777  004526  164760        MOV     #BADINT,@RKVEC  ;RESTORE RK11 INTERRUPT VECTOR ADRES
3846                                                                ;FOR UNEXPECTED INTERUTS
3847  014422  032710  020000               BIT     #20000,@RD      ;DID SCP BIT SET?
3848  014426  001003                        BNE     6$              ;YES, BRANCH
3849  014430  011037  001162               MOV     @RD,$REGD       ;GET RKCS
3850  014434  104053                        ERROR   53              ;SCP DID NOT SET AFTER RK11 INTERRUPTED
3851                                                                ;INDICATING SEEK WAS DONE
3852  014436  017701  164664         6$:     MOV     @RKDS,R1        ;GET RKDS
3853  014442  042701  017777               BIC     #17777,R1       ;MASK NON-ID BITS IN RKDS
3854  014446  020137  001350               CMP     R1,DRIVAD       ;CORRECT ID BITS IN RKDS?
3855  014452  001414                        BEQ     7$              ;YES, BRANCH
3856
3857  014454  013746  001350               MOV     DRIVAD,-(SP)    ;PUSH DRV ADRES ON THE STACK
3858  014460  004737  021106               JSR     PC,SHFTRT       ;GO, SHIFT RIGHT DRV #
3859  014464  012637  001162               MOV     (SP)+,$REGD     ;GET EXPCTD DRV #
3860  014470  010146                        MOV     R1,-(SP)        ;PUSH ID BITS ON THE STACK
3861  014472  004737  021106               JSR     PC,SHFTRT       ;GO SHIFT THEM RIGHT
3862  014476  012637  001164               MOV     (SP)+,$REG1     ;POP THE RECVD ID BITS
3863  014502  104047                        ERROR   47              ;WRONG ID BITS WERE RECVD IN
3864                                                                ;RKDS AFTER SEEK WAS DONE (INTRUPT
3865                                                                ;MODE). 'EXPCT' INDICATES THE DRIVE
3866                                                                ;# THAT SHOULD HAVE BEEN IN THE
3867                                                                ;ID BITS, 'RECVD' INDICATES THE
3868                                                                ;DRIVE #'THAT WAS RECVD IN THE ID BITS
3869
3870  014504                         7$:
3871  014504  012746  000340               MOV     #340,-(SP)
3872  014510  012746  014516               MOV     #64$,-(SP)
3873  014514  000002                        RTI
3874  014516                         64$:
```

```
                                        CNT.RESET                       ;GO DO CONTROL RESET
                                        MOV     DRIVAD,@RKDA            ;ADRES THE DRIVE
                                        BIT     #160000,@RKDS           ;DID CNTRL RESET CLEAR DRIVE ID BITS?
                                        BEQ     8$                      ;YES, BRANCH
                                        MOV     @RKDS,$REGO             ;GET RKDS
                                        ERROR   50                      ;CONTROL RESET DIDN'T CLEAR THE
                                                                        ;DRIVE ID BITS (13-15) IN RKDS
                                8$:     CMP     #200,@R0                ;WAS SCP BIT CLEARED BY CNTRL RESET?
                                        BEQ     TST37                   ;;YES, EXIT
                                        MOV     @R0,$REGO               ;GET RKCS
                                        ERROR   100                     ;CNTRL RESET DID NOT CLEAR SCP BIT

                        ;;**************************************************************
                        ;*TEST 37       CHECK THAT WITH IDE SET RK11 INTERRUPTS WHEN READ IS DONE
                                ;*THIS TEST CHECKS THAT WHEN A DATA TRANSFER FUNCTION IS DONE
                                ;*WITH IDE BIT SET, RK11 INTERRUPTS WHEN THE FUNCTION IS COMPLETED
                                ;*FUNCTION USED IN THIS TEST IS READ.
                        ;;**************************************************************
                        TST37:  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 CLEAR IT

                                MOV     RKCS,R0
                                MOV     RKDA,R2
                                MOV     RKBA,R4
                                MOV     DRIVAD,R1
                                BIS     #13,R1                          ;SET BITS FOR SEC 13
                                MOV     #-200,@RKWC                     ;READ 200 (OCTAL WORDS)
                                MOV     R1,@R2                          ;FROM THIS DISK ADRES (CYL 0, SEC 13)
                                MOV     #OUTBUF,@R4                     ;INTO THIS BUS ADRES
                                MOV     RKVEC,R5
                                MOV     #1$,(R5)+                       ;SET UP VECTOR ADRES FOR RK11 TO INTRUPT
                                MOV     #340,(R5)                       ;SET PSW ON INTERUPT
                                MOV     #105,@R0                        ;READ, GO, IDE SET
                                WAT.INT ,127710                         ;WAIT FOR RK11 TO INTERRUPT ON
                                                                        ;COMPLETION OF READ
                                                                        ;WAITING TIME= 337 MS FOR 11/20
                                                                        ;67 MS FOR 11/45
                                MOV     #BADINT,@RKVEC                  ;RESTORE UNEXPCTED INTERRUPT VECTOR ADRES
                                MOV     @R0,$REGO                       ;GET RKCS
                                ERROR   101                             ;RK11 DID NOT INTERRUPT AFTER READ
                                                                        ;WAS DONE, IDE BIT SET.
                                BR      1$+10
                        1$:     CMP     (SP)+,(SP)+                     ;OK, IF RK11 INTERRUPTED TO THIS
```

```
                                                        ;RESTORE STACK POINTER (FROM Rk11 INTERRUPT)
014674  022626                      CMP     (SP)+,(SP)+ ;RESTORE STACK POINTER (FROM WAT.INT)
014676  012777  004526  164476      MOV     #BADINT,@RKVEC ;RESTORE UNEXPECTED RK11 INTERRUPT
                                                        ;VECTOR ADRES
014704  004737  021250              JSR     PC,CHKER    ;CHECK IF ANY BIT IN RKER IS SET,
                                                        ;IF YES, RETURN HERE.
014710  104036                      ERROR   36          ;RKER SET ON DOING READ FROM SEC 0,
                                                        ; CYL 13 IN INTERRUPT MODE
014712  062701  000005      4$:     ADD     #5,R1       ;RKDA SHOULD HAVE INCREMENTED TO THIS
014716  020112                      CMP     R1,@R2      ;DID RKDA INCREMENT CORRECTLY?
014720  001405                      BEQ     2$          ;YES BRANCH
014722  010137  001162              MOV     R1,$REG0    ;GET EXPCTD RTDA
014726  011237  001164              MOV     @R2,$REG1   ;GET RKDA RECVD
014732  104040                      ERROR   40          ;RKDA INCREMENTED WRONG ON DOING
                                                        ;A READ ON CYL 0, SEC 13
014734  004737  021224      2$:     JSR     PC,CHKWC    ;CHECK THAT RKWC OVERFLOWED TO 0,
                                                        ;IF NOT RETURN HERE.
014740  104041                      ERROR   41          ;RKWC DIDN'T OURFLO AFTER
                                                        ;A READ OF 200 WORDS
014742              3$:
014742  012746  000340              MOV     #340,-(SP)
014746  012746  014754              MOV     #64$,-(SP)
014752  000002                      RTI
014754              64$:
014754  022714  033640              CMP     #OUTBUF+400,@R4 ;DID RKBA INCREMENT CORRECTLY?
014760  001406                      BEQ     TST40           ;;YES, EXIT
014762  012737  033640  001162      MOV     #OUTBUF+400,$REG0 ;GET EXPCT RKBA
014770  011437  001164              MOV     @R4,$REG1       ;GET RKBA RECVD
014774  104042                      ERROR   42              ;RKBA DID NOT INCREMENT CORRECTLY
                                                            ;AFTER A READ OF 200 WORDS

                            ;;**************************************************************
                            ;*TEST 40       CHECK THAT RK11 INTERRUPTS AT BR5 ONLY
                                ;*THIS TEST CHECKS THAT RK11 CAN ITERRUPT AT BR5 ONLY.  IF IT
                                ;*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.
                            ;;**************************************************************
014776  000004            TST40:  SCOPE
015000  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
015002  104421                    TST.SIN                 ;CHECK IF SIN IS SET, IF SET
                                                        ;DO DRIVE RESET TO CLEAR IT
015004  012737  015040  001110      MOV     #1$,$LPERR  ;SET RETURN ADRES FOR LUPING
```

```
                                                        ;ON ERROR (SW 9)
015012  013700  001332          MOV     RKCS,R0
015016  013777  001350  164314  MOV     DRIVAD,@RKDA
015024  012701  000007          MOV     #7,R1           ;PRIORITY LEVEL 7
015030  012702  000340          MOV     #340,R2         ;BR LEVEL 7 FOR PSW
015034  013703  001400          MOV     RKPRI,R3        ;NOTE, IF RK11 INTERRUPT LEVEL IS
                                                        ;CHANGED FROM 5 TO ANY OTHER LEVEL
                                                        ;THEN CHANGE CONTENTS OF 'RKPRI'
                                                        ; ACCORDINGLY
015040  013704  001402  1$:     MOV     RKVEC,R4
015044  012724  015152          MOV     #3$,(R4)+       ;SET UP ADRES FOR RK11 TO INTERUPT
015050  012714  000340          MOV     #340,(R4)       ;SET UP PSW ON INTERUPT
015054  010246                  MOV     R2,-(SP)                ;SET PROCESSOR PRIORITY LEVEL AS
015056  012746  015064          MOV     #4$,-(SP)
015062  000002                  RTI
015064          4$:                                     ;INDICATED BY R2
015064  012710  000100          MOV     #100,@R0        ;SET THE IDE BIT
015070  012705  177760          MOV     #-20,R5         ;WAIT FOR THE RK11 INTERRUPT
015074  005205                  INC     R5              ;WAITING TIME=78 US FOR 11/20
015076  001376                  BNE     .-2             ;13 US FOR 11/45
015100  020203                  CMP     R2,R3           ;WAS THE CPU PRIORITY LEVEL LESS THAN
015102  003005                  BGT     2$              ;THE RK11 LEVEL?  IF YES, RK11
                                                        ;SHOULD HAVE INTERRUPTED. ERROR.
                                                        ;IF IT DID NOT
015104  010137  001162          MOV     R1,$REG0        ;GET CPU BR LEVEL
015110  011037  001164          MOV     @R0,$REG1       ;GET RKCS
015114  104103                  ERROR   103             ;THOUGH CPU LEVEL WAS LESS THAN
                                                        ;THE RK11 LEVEL (5), RK11 DID NOT
                                                        ;INTERRUPT
015116  005010          2$:     CLR     @R0             ;CLEAR RKCS
015120  062702  177740          ADD     #-40,R2         ;DECREASE THE PRIORITY LEVEL (FOR
                                                        ;CPU) BY 1
015124  005301                  DEC     R1              ;CPU WILL B AT THIS LEVEL
015126  001344                  BNE     1$              ;LUP BAK & CHK FOR THIS BR LEVEL.
                                                        ;DONE WITH CHKING FOR ALL LEVELS.
015130  012777  004526  164244  MOV     #BADINT,@RKVEC  ;RESTORE UNEXPECTED RK11 INTERRUPT
                                                        ;VECTOR
015136  012746  000340          MOV     #340,-(SP)
015142  012746  015150          MOV     #64$,-(SP)
015146  000002                  RTI
015150          64$:
015150  000414                  BR      TST41           ;;EXIT,TO NXT TST

015152  022626          3$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
015154  012777  004526  164220  MOV     #BADINT,@RKVEC  ;RESTORE UNEXPECTED RK11 INTERRUPT
                                                        ;VECTOR
015162  020203                  CMP     R2,R3           ;IF THIS INTERRUPT OCCURED WHEN
015164  003754                  BLE     2$              ;CPU LEVEL WAS LESS THAN THE
                                                        ;RK11 PRIORITY LEVEL (5) THEN IT IS
                                                        ;OK.  IF NOT SO, ERROR
015166  010137  001162          MOV     R1,$REG0        ;GET CPU BR LEVEL
015172  011037  001164          MOV     @R0,$REG1       ;GET RKCS
015176  104104                  ERROR   104             ;RK11 INTERRUPTED WHEN THE CPU
                                                        ;LEVEL (AS POINTED BY R1) WAS
                                                        ;HIGHER OR SAME AS THE RK11
                                                        ;LEVEL (5)
```

```
MAINDEC-11-D2RKA-B     MACY11 27(1006)  04-OCT-76  16:06  PAGE 75
D2RKC.P11    22-SEP-76 08:47           T40     CHECK THAT RK11 INTERRUPTS AT BR5 ONLY                    SEQ 0092

 4043  015200  000746                        BR      2$              ;GO BACK & CHK THE NXT LEVEL
 4044
 4045                                  ;;*****************************************************************
 4046                                  ;*TEST 41       SIMULATE & CHECK 'OVR' ERROR
 4047                                  ;*THIS TEST SIMULATES OVERRUN ERROR AND CHECKS IF THE OVR
 4048                                  ;*BIT IN RKER GETS SET.  THEN IT IS CLEARED USING CNTRL RESET
 4049                                  ;*& CHECKED THAT IT WAS CLEARED.  OVR CONDITION IS SIMULATED
 4050                                  ;*BY TRYING TO READ 401(OCTAL) WORDS FROM LAST CYLINDER(312),
 4051                                  ;*LAST SECTOR (13), SURFACE 1.
 4052                                  ;;*****************************************************************
 4053  015202  000004            T$T41:  SCOPE
 4054  015204  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
 4064  015206  104421                    TST.SIN                     ;CHECK IF SIN IS SET, IF
 4065                                                                 ;SET, DO DRIVE RESET TO CLR IT
 4066  015210  013701  001350            MOV     DRIVAD,R1           ;GET ADRES OF DRIVE
 4067  015214  052701  014533            BIS     #14533,R1           ;SET BITS FOR LAST CYLINDER (312),
 4068                                                                 ;SUR 1, LAST SECTOR (13)
 4069  015220  012777  177377  164106    MOV     #-401,@RKWC         ;READ 401 WORDS
 4070  015226  012777  033240  164102    MOV     #OUTBUF,@RKBA       ;INTO THIS MEMORY BUFFER
 4071  015234  010177  164100            MOV     R1,@RKDA            ;FROM THIS DSK ADRES, LAST CYL.
 4072                                                                 ;LAST SEC, SURFACE 1
 4073  015240  012777  000005  164064    MOV     #5,@RKCS                    ;READ, GO
 4074
 4075  015246  005002                    CLR     R2
 4076  015250  105777  164056      1$:   TSTB    @RKCS               ;DID CNTRL RDY SET?
 4077  015254  100410                    BMI     2$                  ;YES, BRANCH
 4078  015256  005202                    INC     R2                  ;NO, WAIT FOR IT
 4079  015260  001373                    BNE     1$                  ;IF WAITED LONG, REPORT ERROR MESSAGE BECAUSE
 4080                                                                 ;OVR SHOULD HAVE SET HE CAUSING
 4081                                                                 ;CNTRL RDY TO SET BY NOW
 4082  015262  017737  164046  001166    MOV     @RKWC,$REG2
 4083  015270  004737  020716            JSR     PC,GT2RG            ;GO, GET RKCS, ER
 4084  015274  104002                    ERROR   2                   ;CNTRL RDY DID NOT SET AFTER DOING
                                                                      ;AN OVR READ.  HE SHOULD HAVE OCCURED
                                                                      ;SETTING CNTRL RDY (HE BECAUSE OF
                                                                      ;OVR CONDITIONS)
 4088  015276  032777  040000  164024  2$:   BIT     #40000,@RKER        ;DID OVR BIT SET IN RKER?
 4089  015304  001006                    BNE     3$
 4090  015306  004737  020716            JSR     PC,GT2RG            ;GET RKCS, ER
 4091  015312  012737  040000  001166    MOV     #40000,$REG2        ;THIS BIT (OVR) DID NOT SET.
 4092  015320  104105                    ERROR   105                 ;OVR ERROR BIT DID NOT SET IN RKER
                                                                      ;ON SIMULATING OVR CONDITIONS
 4094  015322  022777  140204  164002  3$:   CMP     #140204,@RKCS       ;DID HE & ERR SET WHEN OVR SET IN RKER?
 4095  015330  001403                    BEQ     4$                  ;YES, BRANCH
 4096  015332  004737  020716            JSR     PC,GT2RG            ;GET RKCS, ER
 4097  015336  104106                    ERROR   106                 ;HE OR ERR BIT DID NOT SET IN RKCS WHEN
 4098                                                                 ;AN OVR ERROR WAS SIMULATED
```

```
                                                        ;CLEAR OVER, ERR, HE BITS
015340  104413          45:   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
015342  004737  021264        JSR     PC,CHKECLR      ;CHECK IF 'O.R' BIT WAS CLEARED BY
                                                        ;CON.RESET, IF NOT RETURN HERE.
015346  104102                ERROR   102             ;CNTRL RESET DID NOT CLEAR OVR
                                                        ;BIT IN RKER
015350  004737  021310  55:   JSR     PC,CHKCCLR      ;CHECK IF 'ERR' & 'HE' BIT GOT CLEARED BY
                                                        ;CON.RESET, IF NOT RETURN HERE.
015354  104102                ERROR   102             ;CNTRL RESET DID NOT CLEAR
                                                        ;HE OR ERR BIT IN RKCS.
015356  004737  021412  65:   JSR     PC,DRESET       ;GO DO DRIVE RESET
015362  104026                ERROR   26              ;R/W/S RDY DIDN'T SET
                                                        ;AFTER THE ABOVE DRIVE RESET

                        ;;*****************************************************************
                        ;*TEST 42        SIMULATE & CHECK PGE ERROR
                                ;*THIS TEST SIMULATES 'PROGRAMMING ERROR' & CHECKS IF IT IS
                                ;*DETECTED BY PGE BIT IN RKER.  THEN A CNTRL RESET IS DONE &
                                ;*IT IS CHECKED IF PGE BIT WAS CLEARED.  IT IS ALSO CHECKED IF
                                ;*THE SETTING & CLEARING OF PGE BIT SETS & CLEARS HE, ERR
                                ;*BITS IN RKCS.
                        ;;*****************************************************************
000004  +ST42:  SCOPE
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
104421                  TST.SIN                         ;GO CHECK IF SIN IS SET, IF
                                                        ;SET DO DRIVE RESET TO CLR IT
013701  001330          MOV     RKER,R1
013777  001350  163734  MOV     DRIVAD,@RKDA            ;ADRES THE DRIVE, CYLINDER 0

012777  002011  163720  MOV     #2011,@RKCS             ;SEEK, GO WITH FMT SET
                                                        ;THIS IS A PGE SIMULATION
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
032711  004000          BIT     #4000,@R1               ;DID PGE BIT IN RKER SET?
```

```
015420  001006                  BNE     1$                      ;YES, BRANCH
015422  012737  004000  001166  MOV     #4000,$REG2             ;THIS BIT IN RKER (PGE) DID NOT SET
015430  004737  020716          JSR     PC,GT2RG                ;GO GET RKCS, ER FOR MESAGE
015434  104105                  ERROR   105                     ;PGE BIT DID NOT SET IN RKER
                                                                ;ON SIMULATION OF PGE CONDITION
                                                                ;$REG2 CONTAINS THE RKER BIT (PGE)
                                                                ;THAT SHOULD HAVE SET.
015436  022777  142210  163666 1$:     CMP     #142210,@RKCS   ;DID HE & ERR BITS SET?
015444  001403                  BEQ     2$                      ;YES, BRANCH
015446  004737  020716          JSR     PC,GT2RG                ;GO, GET RKCS, ER
015452  104106                  ERROR   106                     ;HE OR ERR BIT DID NOT SET WHEN
                                                                ;PGE SET IN RKER.
                                                                ;CLEAR PGE, HE, ERR BITS
015454  104413          2$:     CNT.RESET                       ;GO, DO CONTROL RESET
                                                                ;THIS IS A CALL FOR THE 'CNTRL-
                                                                ;RESET' ROUTINE. A CONTROL RESET IS
                                                                ;ISSUED AND AFTER A CERTAIN TIME
                                                                ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                ;AN ERROR IS REPORTED. NOTE THAT
                                                                ;THE PC IN ERROR MESSAGE IS THE
                                                                ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                ;THIS IS A VERY BASIC ERR& IF IT
                                                                ;OCCURS GO BACK TO TEST 10
015456  004737  021264          JSR     PC,CHKECLR              ;CHECK IF 'PGE' BIT GOT CLEARED BY
                                                                ;CONTROL RESET, IF NOT RETURN HERE.
015462  104102                  ERROR   102                     ;CNTRL RESET DID NOT CLEAR
                                                                ;PGE BIT IN RKER
015464  004737  021310  3$:     JSR     PC,CHKCCLR              ;CHECK IF 'ERR' BITGOT CLEARED BY
                                                                ;CON.RESET, IF NOT RETURN HERE.
015470  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
                ;*IS REFERENCED & IT HAPPENS TO BE A NON EXISTENT LOCATION
                ;*(FOR DIAGNOSTIC PURPOSES LIKE THIS). IT IS ALSO CHECKED
                ;*IF HE & ERR BITS ALSO SET AND ALL 3 BITS CAN BE CLEARED
                ;* BY CONTROL RESET.
;;*****************************************************************
015472  000004  TST43:  SCOPE
015474  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
015476  104421                  TST.SIN                         ;GO CHECK IF SIN IS SET
                                                                ;IF SET DO DRIVE RESET TO CLR IT
015500  005002                  CLR     R2
015502  013700  001332          MOV     RKCS,R0
```

```
4211  015506  012777  177777  163620          MOV     #-1,@RKWC       ;WRITE CHECK 1 WORD
4212  015514  012777  160000  163614          MOV     #160000,@RKBA   ;AT THIS BUS ADRES
4213  015522  013777  001350  163610          MOV     DRIVAD,@RKDA    ;WITH THIS DISK ADRES (CYL 0, SEC 0,
4214  015530  012710  000067                  MOV     #67,@R0         ;WRT CHK, GO, MEX BITS SET
4215  015534  105777  163572          1$:     TSTB    @RKCS           ;DID CNTRL RDY SET AS A RESULT OF HE?
4216  015540  100410                          BMI     2$              ;YES, BRANCH
4217  015542  005202                          INC     R2              ;WAITED LONG ENOUGH?
4218  015544  001373                          BNE     1$              ;IF NOT LUP BAK & WAIT
4219  015546  004737  020716                  JSR     PC,GT2RG        ;GET RKCS, ER
4220  015552  017737  163556  001166          MOV     @RKWC,$REG2     ;GET RKWC
4221  015560  104113                          ERROR   113             ;CNTRL RDY DID NOT SET ON DOING
4222                                                                  ;A WRT CHK WITH A NXM LOCATION.
4223                                                                  ;THIS HE SHOULD HAVE SET THE
4224                                                                  ;CNTRL RDY BIT IN RKCS
4225  015562  032777  002000  163540  2$:     BIT     #2000,@RKER     ;DID NXM BIT IN RKER SET?
4226  015570  001006                          BNE     3$              ;YES, BRANCH
4227  015572  004737  020716                  JSR     PC,GT2RG        ;GO GET RKCS, RKER
4228  015576  012737  002000  001166          MOV     #2000,$REG2     ;THIS BIT (NXM) DID NOT SET IN RKER
4229  015604  104105                          ERROR   105             ;NXM BIT DID NOT SET IN RKER ON
4230                                                                  ;SIMULATING NXM CONDITION.
4231  015606  022710  140266          3$:     CMP     #140266,@R0     ;DID HE & ERR BIT SET?
4232  015612  001403                          BEQ     4$              ;YES, BRANCH
4233  015614  004737  020716                  JSR     PC,GT2RG        ;GO, GET RKCS, RKER
4234  015620  104106                          ERROR   106             ;HE OR ERR BIT DID NOT SET WHEN
4235                                                                  ;NXM ERROR WAS SIMULATED
4236                                                                  ;CLEAR NXM, HE, ERR BITS
4237  015622  104413          4$:     CNT.RESET                       ;GO, DO CONTROL RESET
4238                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
4239                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
4240                                                                  ;ISSUED AND AFTER A CERTAIN TIME
4241                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
4242                                                                  ;AN ERROR IS REPORTED. NOTE THAT
4243                                                                  ;THE PC IN ERROR MESSAGE IS THE
4244                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
4245                                                                  ;THIS IS A VERY BASIC ERR3 IF IT
4246                                                                  ;OCCURS GO BACK TO TEST 10
4247  015624  004737  021264                  JSR     PC,CHKECLR      ;CHECK IF 'NXM' BIT GOT C:LEARED BY
4248                                                                  ;CON.RESET, IF NOT RETURN HERE.
4249  015630  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
4250                                                                  ;NXM BIT IN RKER
4251  015632  004737  021310          5$:     JSR     PC,CHKCCLR      ;CHECK IF 'HE' & 'ERR' BITS GOT CLEARED
4252                                                                  ;BY CON.RESET, IF NOT RETURN HERE.
4253  015636  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
4254                                                                  ;HE OR ERR BIT IN RKCS.
4255  015640  004737  021344          6$:     JSR     PC,TSTRWS       ;GO CHECK IF R/W/S RDY IS SET &
4256                                                                  ;WAIT FOR IT.  SKIP ERROR IF IT IS SET
4257  015644  104016                          ERROR   16              ;R/W/S RDY IS NOT SET
4258
4259                                  ;;*****************************************************************
4260                                  ;*TEST 44        SIMULATE & CHECK NXD ERROR
4261                                          ;*THIS TEST SIMULATES NON-EXISTENT DISK ERROR & CHECKS IF
4262                                          ;*IT IS DETECTED BY NXD BIT OF RKER.  IF ALL EIGHT ARE PRESENT
4263                                          ;*THEN THIS TEST IS ABORTED FOR SIMULATION CANNOT BE DONE.
4264                                  ;;*****************************************************************
4265  015646  000004                  TST44:  SCOPE
4266  015650  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
015652  104421                  TST.SIN               ;CHECK IF SIN IS SET, IF SET
                                                        ;DO DRV RESET TO CLR IT
015654  013700  001332          MOV     RKCS,R0
015660  012702  160000          MOV     #160000,R2    ;ADRES DRIVE 7 TO FIND
                                                        ;IF IT IS PRESENT
015664  010277  163450  1$:     MOV     R2,@RKDA      ;ADRES DRIVE # POINTED TO BY R2
015670  104417  000001          DELAY   .1            ;TIME DELAY, 7.5 US ON 11/20,
                                                        ;1.5 US ON 11/45
015674  105777  163426          TSTB    @RKDS         ;IS IT PRESENT?
015700  100004                  BPL     2$            ;NO, BRANCH
015702  062702  160000          ADD     #-20000,R2    ;ADRES THE NXT DRIVE IN THE
                                                        ;REVERSE ORDER. I.E. 7,6,...
015706  001366                  BNE     1$            ;LUP BAK & TRY TO FIND A DRIVE
                                                        ;THAT'S NOT PRESENT
015710  000435                  BR      TST45         ;;EXIT TO THE NXT TST
015712  012710  000015  2$:     MOV     #15,@R0       ;DRIVE RESE , ON A NX DRIVE
015716  104417  000106          DELAY   ,106          ;TIME DELAY, 525 US ON 11/20
                                                        ;105 US ON 11/45
015722  105777  163402          TSTB    @RKER         ;DID NXD BIT IN RKER SET?
015726  001006                  BNE     3$            ;YES, BRANCH
015730  004737  020716          JSR     PC,GT2RG      ;GET RKCS, RKER
015734  012737  000200  001166  MOV     #200,$REG2    ;THIS BIT (NXD) IN RKER DID NOT SET
015742  104105                  ERROR   105           ;NXD BIT DID NOT SET ON TRYING
                                                        ;TO PERFORM A FUNCTION ON A
                                                        ;NON-EXISTENT DRIVE
                                                        ;CHECK THAT THE JUMPER CARD CONTAINING
                                                        ;JUMPERS FOR DRIVES PRESENT IS PROPERLY
                                                        ;CONNECTED
                                                        ;NOTE THAT ON RK11C IF A DRIVE
                                                        ;IS OFFLINE BUT PHYSICALLY PRESENT
                                                        ;(IE. DRY IS CLR FOR THAT DRIVE)
                                                        ;& A FUNCTION IS INITIATED ON THAT
                                                        ;DRIVE NXD WON'T SET, BUT U WILL
                                                        ;GET ONLY A DRE,HE & ERR.
015744  022710  140214  3$:     CMP     #140214,@R0   ;DID HE & ERR SET WHEN NXD SET?
015750  001403                  BEQ     4$            ;YES BRANCH
015752  004737  020716          JSR     PC,GT2RG      ;HE OR ERR BIT DID NOT SET
015756  104106                  ERROR   106           ;WHEN NXD WAS SIMULATED
                                                        ;CLEAR NXD, HE, ERR BITS
015760  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
016762  004737  021264          JSR     PC,CHKECLR    ;CHECK IF 'NXD' BIT WAS CLEARED BY
                                                        ;CON.RESET. IF NOT, RETURN HERE.
016766  104102                  ERROR   102           ;CNTRL RESET DID NOT CLEAR
                                                        ;NXD BIT IN RKER
016770  004737  021310    5$:   JSR     PC,CHKCCLR    ;CHECK IF 'HE' & 'ERR' BITS WERE CLEARED
                                                        ;BY CON.RESET. IF NOT RETURN HERE.
016774  104102                  ERROR   102           ;CNTRL RESET DID NOT CLEAR
                                                        ;HE OR ERR BIT IN RKCS
016776  004737  021344          JSR     PC,TSTRWS     ;GO CHECK & WAIT FOR R/W/S RDY
                                                        ;TO SET.  IF SET SKIP ERROR
017002  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
                        ;;**************************************************************
017004  000004          TST45:  SCOPE
017006  013700  001332          MOV     RKCS,R0
017012  012737  177773  001362  2$:    MOV     #-5,COUNT     ;ALLOW 'ERROR 133' ONLY 5 TIMES
017020  013702  001350          MOV     DRIVAD,R2     ;GET ADRES OF DRIVE
017024  052702  014540          BIS     #14540,R2     ;SET BITS FOR CYL 313
017030  012737  017036  001110  MOV     #3$,$LPERR    ;SET RETURN ADRES FOR
                                                        ;LUPING ON EROR (SW9)
017036  104413          3$:     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
                                                        ;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
017040  004737  021344          JSR     PC,TSTRWS     ;GO CHECK & WAIT FOR R/W/S RDY
                                                        ;TO SET. IF SET SKIP ERROR BELOW
017044  104016                  ERROR   16            ;R/W/S RDY IS NOT SET
017046  104421                  TST.SIN               ;CHECK IF SIN IS SET, IF SET
                                                        ;DO DRIVE RESET TO CLR IT
017050  010277  163264          MOV     R2,@RKDA      ;ADRES DRIVE, NXC CYLINDER
017054  012710  000011          MOV     #11,@R0       ;SEEK, GO TO NXC CYL
017060  104412                  CHKCRDY               ;GO CHECK IF CONTROL RDY IS SET
                                                        ;IF SO, SKIP THE EROR MESSAGE.
017062  104021                  ERROR   21            ;SEEK WAS TRIED TO A NON EXISTENT
                                                        ;CYLINDER, NXC SHOULD HAVE OCCURED
                                                        ;SETTING CNTRL RDY. BUT CNTRL RDY
                                                        ;DID NOT SET.
017064  032777  000100  163236  9$:    BIT     #100,@RKER    ;DID NXC SET?
017072  001020                  BNE     4$            ;YES, BRANCH
017074  004737  020716          JSR     PC,GT2RG      ;GO GET RKCS, ER
```

```
 4379  016100  017737  163234  001166          MOV     @RKDA,$REG2     ;GET RKDA
 4380  016106  104110                          ERROR   110             ;NXC DID NOT SET WHEN SEEK
 4381                                                                  ;WAS TRIED TO CYLINDER AS INDICATED
 4382                                                                  ;IN RKDA
 4383  016110  004737  021344                  JSR     PC,TSTRWS       ;CHECK & WAIT FOR R/W/S RDY.
 4384                                                                  ;IF SET SKIP ERROR
 4385  016114  104016                          ERROR   16              ;R/W/S SHOULD BE SET
 4386  016116  104413                          CNT.RESET               ;GO DO CONTROL RESET
 4387  016120  004737  021412                  JSR     PC,DRESET       ;GO DO DRIVE RESET
 4388  016124  104026                          ERROR   26              ;NXC DID NOT SET AND DRIVE MAY
 4389                                                                  ;HAVE TRIED TO DO A SEEK, AFTER
 4390                                                                  ;WHICH R/W/S RDY DID NOT SET
 4391  016126  005237  001362                  INC     COUNT           ;ALLOW ONLY 5 MESSAGES FOR
 4392  016132  001405                          BEQ     5$              ;ERROR 133
 4393  016134  062702  000040          4$:     ADD     #40,R2          ;ADRES THE NXT CYL(IN NON-EXISTENT ZONE.
 4394  016140  032702  017740                  BIT     #17740,R2       ;CHKD FOR ALL NXC'S?
 4395  016144  001334                          BNE     3$              ;IF NOT, LUP BAK & CHK THE NXT NXC
 4396
 4397  016146  032710  140000          5$:     BIT     #140000,@R0     ;DID HE & ERR BIT SET WHEN NXC BIT SET?
 4398  016152  001003                          BNE     6$              ;YES, BRANCH
 4399  016154  004737  020716                  JSR     PC,GT2RG        ;GET RKCS, ER
 4400  016160  104106                          ERROR   106             ;HE OR ERR BIT DID NOT SET IN RKCS
 4401                                                                  ;WHEN NXC ERPOR WAS SIMULATED
 4402                                                                  ;CLEAR HE, ERR, NXC BITS
 4403  016162  104413                  6$:     CNT.RESET               ;GO, DO CONTROL RESET
 4404                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
 4405                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
 4406                                                                  ;ISSUED AND AFTER A CERTAIN TIME
 4407                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
 4408                                                                  ;AN ERROR IS REPORTED. NOTE THAT
 4409                                                                  ;THE PC IN ERROR MESSAGE IS THE
 4410                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
 4411                                                                  ;THIS IS A VERY BASIC ERR& IF IT
 4412                                                                  ;OCCURS GO BACK TO TEST 10
 4413  016164  004737  021264                  JSR     PC,CHKECLR      ;CHECK IF 'NXC' BIT WAS CLEARED BY
 4414                                                                  ;CON.RESET. IF NOT, RETURN HERE.
 4415  016170  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
 4416                                                                  ;NXC BIT IN RKER.
 4417  016172  032710  140000          7$:     BIT     #140000,@R0     ;DID HE & ERR BITS GET CLEARED?
 4418  016176  001405                          BEQ     TST46           ;;YES, EXIT
 4419  016200  010037  001162                  MOV     R0,$REG0        ;GET ADRES OF RKCS
 4420  016204  011037  001164                  MOV     @R0,$REG1       ;GET RKCS CONTENTS
 4421  016210  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
 4422                                                                  ;HE OR ERR BIT IN RKCS
 4423
 4424                          ;;*****************************************************************
 4425                          ;*TEST 46       SIMULATE & CHECK NXS ERROR
 4426                                  ;*THIS TEST SIMULATES NON-EXISTENT SECTOR ERROR & CHECKS THAT
 4427                                  ;*IT IS DETECTED BY NXS BIT OF RKER.  IT IS CHECKED THAT
 4428                                  ;*WHEN NXS SETS HE & ERR OF RKER ALSO SETS, AND ALL THREE
 4429                                  ;*CAN BE CLEARED BY CONTROL RESET.
 4430                          ;;*****************************************************************
 4431  016212  000004          TST46:  SCOPE
 4432  016214  104413                  CNT.RESET               ;GO, DO CONTROL RESET
 4433                                                          ;THIS IS A CALL FOR THE 'CNTRL-
 4434                                                          ;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
016216  013700  001332          MOV     RKCS,R0
016222  013777  001350  163110  MOV     DRIVAD,@RKDA    ;GET ADRES OF DRIVE
016230  052777  000014  163102  BIS     #14.,@RKDA      ;SET BITS FOR SECTOR 12 (DECIMAL)
016236  012777  177777  163070  MOV     #-1,@RKWC       ;READ 1 WORD
016244  012777  033240  163064  MOV     #OUTBUF,@RKBA   ;INTO THIS BUS ADRES
016252  012710  000005          MOV     #5,@R0          ;READ, GO (FROM NX SECTOR)
016256  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
                                                        ;NXS ERROR SHOULD OCCUR NOW
016260  017702  163044          MOV     @RKER,R2
016264  032702  000040          BIT     #40,R2          ;DID NXS BIT SET IN RKER?
016270  001006                  BNE     1$              ;YES, BRANCH
016272  004737  020716          JSR     PC,GT2RG        ;GO GET RKCS, RKER
016276  012737  000040  001166  MOV     #40,$REG2       ;THIS BIT (NXS) IN RKER DID NOT SET
016304  104105                  ERROR   105             ;NXS BIT DID NOT SET ON SIMULATING
                                                        ;NXS ERROR
016306  042702  000040  1$:     BIC     #40,R2          ;MASK NXS BIT
016312  001407                  BEQ     2$              ;CHECK IF ANY OTHER
                                                        ;RKER BIT SET
016314  012737  000040  001162  MOV     #40,$REG0       ;GET EXPCTD RKER
016322  017737  163002  001164  MOV     @RKER,$REG1     ;GET RKER RECVD
016330  104107                  ERROR   107             ;ONLY 'NXS' SHOULD BE SET
                                                        ;IN RKER, ANOTHER RKER BIT
                                                        ;WAS SET. (NOTE 'NXS' WAS
                                                        ;SIMULATED)
016332  022710  140204  2$:     CMP     #140204,@R0     ;DID HE & ERR BITS SET?
016336  001403                  BEQ     3$              ;YES, BRANCH
016340  004737  020716          JSR     PC,GT2RG        ;GO GET RKCS, RKER
016344  104106                  ERROR   106             ;HE OR ERR BIT DID NOT SET WHEN
                                                        ;NXS ERROR OCCURED
                                                        ;CLEAR NXS, HE, ERR BITS
016346  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
                                                        ;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
016350  004737  021264          JSR     PC,CHKECLR      ;CHECK IF 'NXS' BIT WAS CLEARED BY
                                                        ;CON.RESET. IF NOT, RETUEN HERE.
016354  104102                  ERROR   102             ;CNTRL RESET DID NOT CLEAR
                                                        ;NXS BIT IN RKER
```

```
                                4$:     JSR     PC,CHKCCLR      ;CHECL IF 'HE' & 'ERR' BITS WERE CLEARED
                                                                ;BY CON.RESET, IF NOT, RETURN HERE.
016362  104102                          ERROR   102             ;RKCS BITS ERR OR HE WERE NOT
                                                                ;CLEARED BY CNTRL RESET


                                ;;*****************************************************************
                                ;*TEST 47        SIMULATE & CHECK WCE
                                        ;*THIS TEST SIMULATES A WRITE CHECK ERROR AND CHECKS THAT IT
                                        ;*IS DETECTED BY WCE BIT OF RKER.  FOR COMPARISON IT USES
                                        ;*THE 256 WORDS DATA BLOCK WRITTEN ON SECTOR 0, CYLINDER 0
                                        ;*IN A PREVIOUS TEST.  THIS BLOCK IS COMPARED WITH THE 256 WORDS
                                        ;*MEMORY BUFFER STARTING AT 'OUTBUF'.  WCE IS SIMULATED BY
                                        ;*DROPPING A BIT FROM ONE OF THE WORDS IN THE MEMORY BUFFER.
                                ;;*****************************************************************
016364  000004                  TST47:  SCOPE
016366  013700  001332                  MOV     RKCS,R0
016372  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
016374  104421                          TST.SIN                 ;CHECK IF SIN IS SET, IF
                                                                ;SET DO DRV-RESET TO CLR IT
016376  012701  033240                  MOV     #OUTBUF,R1      ;THIS CODE SETS UP A MEMORY
016402  012702  177400                  MOV     #-400,R2        ;BUFFER OF 256 WORDS STARTING
016406  012703  177777                  MOV     #177777,R3      ;AT OUTBUF
                                                                ;FIRST WORD  177400
                                                                ;SECOND      177001
016412  062703  177401          1$:     ADD     #177401,R3
016416  010321                          MOV     R3,(R1)+        ;LAST WORD   000377
016420  005202                          INC     R2              ;HAVE U GENERATED ALL 256 WORDS?
016422  001373                          BNE     1$              ;IF NOT, LUP BAK & GENERATE NXT

016424  012737  170007  033256          MOV     #170007,OUTBUF+16 ;WCE WILL B SIMULATED BY DROPPING A
                                                                ;BIT IN THE EIGHTH WORD WHICH IS
                                                                ;SUPPOSED TO B 174007
016432  012777  177400  162674          MOV     #-400,@RKWC     ;WRT CHK 400 WORDS
016440  012777  033240  162670          MOV     #OUTBUF,@RKBA   ;STARTING AT THIS BUS ADRES
016446  013777  001350  162664          MOV     DRIVAD,@RKDA    ;WITH THIS DISK ADRES, SEC 0, CYL 0
016454  012710  000007                  MOV     #7,@R0          ;WRT CHK, GO

016460  104412                          CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
                                                                ;IF SO, SKIP THE EROR MESSAGE.
016462  104065                          ERROR   65              ;CNTRL RDY DID NOT SET
                                                                ;AFTER WRT CHK
016464  032777  000001  162636  3$:     BIT     #1,@RKER        ;DID WCE BIT SET?
016472  001006                          BNE     4$
016474  004737  020716                  JSR     PC,GT2RG        ;GO, GET RKCS, RKER
016500  012737  000001  001166          MOV     #1,$REG2        ;THIS BIT (WCE) DID NOT SET
016506  104105                          ERROR   105             ;WCE DID NOT SET ON SIMULATING
                                                                ;WCE CONDITIONS
```

```
4547  016510  022710  100206          4$:     CMP     #100206,@R0     ;IS RKCS CORRECT?
4548  016514  001403                          BEQ     5$              ;YES. BRANCH
4549  016516  004737  020716                  JSR     PC,GT2RG        ;GO, GET RKCS, RKER
4550  016522  104106                          ERROR   106             ;HE OR ERR BIT DID NOT SET WHEN
4551                                                                  ;WCE WAS SIMULATED
4552  016524  104413                  5$:     CNT.RESET               ;CNTRL RESET
4553  016526  004737  021264                  JSR     PC,CHKECLR      ;WAS 'WCE' BIT CLEARED?
4554                                                                  ;IF NOT, RETURN HERE.
4555  016532  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
4556                                                                  ;WCE BIT IN RKER
4557  016534  004737  021310          6$:     JSR     PC,CHKCCLR      ;CHECK IF 'ERR' BIT WAS CLEARED. IF
4558                                                                  ;NOT RETURN HERE.
4559  016540  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
4560                                                                  ;RKCS
4561
4562                                  ;;**********************************************************************
4563                                  ;*TEST 50        CHECK THAT SSE STOPS ALL CONTROL ACTION ON SOFT ERROR
4564                                          ;*THIS TEST CHECKS THAT WHEN 'STOP ON SOFT ERROR' BIT IS SET IN
4565                                          ;*RKCS AND A SOFT ERROR IS ENCOUNTERED ALL CONTROL ACTION WILL
4566                                          ;*STOP AT THE END OF THE CURRENT SECTOR IF IDE BIT IS CLEAR.
4567                                          ;*SOFT ERROR IS SIMULATED BY A WCE AS IN THE PREVIOUS
4568                                          ;*TEST. THE PREVIOUS TEST & THE TEST WHICH WRITES DATA
4569                                          ;*BLOCK ON CYLINDER 0, SECTOR 0, SHOULD BE DONE PRIOR
4570                                          ;*TO THIS TEST. A TWO SECTOR 'WRT CHK' WILL BE DONE,
4571                                          ;*CONTROL ACTION SHOULD STOP AFTER THE FIRST SECTOR DURING
4572                                          ;*WHICH A SOFT ERROR IS SIMULATED.
4573                                  ;;**********************************************************************
4574  016542  000004                  TST50:  SCOPE
4575  016544  104413                          CNT.RESET               ;GO, DO CONTROL RESET
4576                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
4577                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
4578                                                                  ;ISSUED AND AFTER A CERTAIN TIME
4579                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
4580                                                                  ;AN ERROR IS REPORTED. NOTE THAT
4581                                                                  ;THE PC IN ERROR MESSAGE IS THE
4582                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
4583                                                                  ;THIS IS A VERY BASIC ERR& IF IT
4584                                                                  ;OCCURS GO BACK TO TEST 10
4585  016546  104421                          TST.SIN                 ;CHECK IF SIN IS SET, IF
4586                                                                  ;SET DO DRIVE RESET TO CLR IT
4587  016550  013700  001332                  MOV     RKCS,R0
4588  016554  012737  170007  033256          MOV     #170007,OUTBUF+16 ;WCE IS SIMULATED BY DROPPING A BIT
4589                                                                  ;IN THE EIGTH WORD (WHICH IS ACTUALLY
4590                                                                  ;174007). NOTE THAT 256 WORD MEMORY
4591                                                                  ;BUFFER IS CREATED IN THE PREVIOUS TEST.
4592  016562  013701  001350                  MOV     DRIVAD,R1
4593  016566  012777  177000  162540          MOV     #-1000,@RKWC    ;WRT CHK 1000 (OCTAL) WORDS, 2 SECTORS
4594  016574  012777  033240  162534          MOV     #OUTBUF,@RKBA   ;FROM THIS BUS ADRES
4595  016602  010177  162532                  MOV     R1,@RKDA        ;WITH THIS DISK ADRES, SEC 0, CYL 0
4596  016606  012710  000407                  MOV     #407,@R0        ;WRT CHK, GO, SSE
4597  016612  104412                          CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
4598                                                                  ;IF SO, SKIP THE EROR MESSAGE.
4599  016614  104065                          ERROR   65              ;CNTRL RDY DID NOT SET AFTER WRT
4600                                                                  ;CHK. A SOFT ERROR (WCE) IN
4601                                                                  ;SECTOR 0 SHOULD HAVE STOPPED
4602                                                                  ;ALL CONTROL ACTION.
```

```
4603  016616  022777  000001  162504  2$:     CMP     #1,@RKER        ;CHECK ONLY 'WCE' BIT SHOULD
4604                                                                  ;BE SET?
4605  016624  001407                          BEQ     3$              ;YES. BRANCH
4606  016626  012737  000001  001162          MOV     #1,$REG0        ;GET EXPCTD RKER
4607  016634  017737  162470  001164          MOV     @RKER,$REG1     ;GET RKER RECVD
4608  016642  104107                          ERROR   107             ;ONLY BIT 'WCE' OF RKER
4609                                                                  ;SHOULD BE SET (WCE WAS
4610                                                                  ;SIMULATED ABOVE).  ERROR
4611                                                                  ;IF IT'S NOT
4612  016644  005201                  3$:     INC     R1              ;CHECK THAT RKDA INCREMENTED BY
4613  016646  020177  162466                  CMP     R1,@RKDA        ;1 SECTOR ONLY IMPLYING THAT
4614                                                                  ;CNTRL ACTION DID STOP AFTER
4615                                                                  ;SOFT ERROR IN SECTOR 0
4616  016652  001406                          BEQ     TST51           ;;YES, EXIT
4617  016654  010137  001162                  MOV     R1,$REG0        ;GET EXPCTD RKDA
4618  016660  017737  162454  001164          MOV     @RKDA,$REG1     ;GET RKDA RECVD
4619  016666  104070                          ERROR   70              ;RKDA SHOULD HAVE INCRMNTD
4620                                                                  ;BY 1 SECTOR ONLY, IT DIDN'T.
4621                                                                  ;WCE WAS SIMULATED IN THE
4622                                                                  ;FIRST SECTOR & A WRT CHK
4623                                                                  ;OF 2 SECTORS WAS ISSUED.
4624                                                                  ;CONTROLLER SHOULD STOP AFTER
4625                                                                  ;DETECTING WCE IN THE FIRST
4626                                                                  ;SECTOR.  HENCE RKDA SHOULD
4627                                                                  ;INCREMENT BY 1 SECTOR ONLY
4628
4629
4630                                  ;;*****************************************************************
4631                                  ;*TEST 51       CHECK THAT RK11 INTERRUPTS ON SOFT ERROR WHEN SSE & IDE ARE SET
4632                                          ;*THIS TEST CHECKS WHEN SSE BIT IS SET WITH IDE SET AND A SOFT
4633                                          ;*ERROR OCCURS, THEN ALL CONTROL ACTION WILL STOP AND A BUS
4634                                          ;*REQUEST (INTERRUPT) WILL OCCUR AT THE END OF THE CURRENT
4635                                          ;*SECTOR.  SOFT ERROR IS SIMULATED BY WCE AS IN PREVIOUS
4636                                          ;*TEST.  PREREQUISITES FOR THIS TEST ARE THE SAME AS THOSE
4637                                          ;*FOR THE PREVIOUS TEST.
4638                                  ;;*****************************************************************
4639  016670  000004                  TST51:  SCOPE
4640  016672  104413                          CNT.RESET                       ;GO, DO CONTROL RESET
4641                                                                          ;THIS IS A CALL FOR THE 'CNTRL-
4642                                                                          ;RESET' ROUTINE. A CONTROL RESET IS
4643                                                                          ;ISSUED AND AFTER A CERTAIN TIME
4644                                                                          ;IF THE 'CNTRL RDY' DOES NOT SET
4645                                                                          ;AN ERROR IS REPORTED. NOTE THAT
4646                                                                          ;THE PC IN ERROR MESSAGE IS THE
4647                                                                          ;PC WHERE 'CNT.RESET' IS LOCATED.
4648                                                                          ;THIS IS A VERY BASIC ERR& IF IT
4649                                                                          ;OCCURS GO BACK TO TEST 10
4650  016674  104421                          TST.SIN                         ;CHECK IF SIN IS SET, IF
4651                                                                          ;SET DO DRIVE RESET TO CLR IT
4652  016676  012737  170007  033256          MOV     #170007,OUTBUF+16 ;WCE IS SIMULATED BY DROPPING A BIT
4653                                                                          ;IN THE EIGHTH WORD (WHICH IS 174007)
4654                                                                          ;NOTE THAT THE 256 WORD MEMORY
4655                                                                          ;BUFFER (STARTING AT OUTBUF) IS
4656                                                                          ;CREATED IN A PREVIOUS TEST.
4657  016704  013701  001350                  MOV     DRIVAD,R1
4658  016710  012777  177000  162416          MOV     #-1000,@RKWC    ;WRT CHK 1000 (OCTAL) WORDS, 2 SECTORS
```

```
 4659  016716  012777  033240  162412          MOV     #OUTBUF,@RKBA   ;FROM THIS BUS ADRES
 4660  016724  010177  162410                  MOV     R1,@RKDA        ;WITH THIS DISK ADRES, SEC 0, CYL 0
 4661  016730  013700  001402                  MOV     RKVEC,R0
 4662  016734  012720  016766                  MOV     #1$,(R0)+       ;SET UP INTERRUPT VECTOR FOR RK11
 4663  016740  012710  000340                  MOV     #340,@R0        ;SET PSW ON INTERRUPT
 4664  016744  012777  000507  162360          MOV     #507,@RKCS      ;WRT CHK, GO. SSE, IDE SET
 4665  016752  104420  177777                  WAT.INT 177777          ;WAIT FOR INTERRUPT FROM RK11
 4666                                                                  ;TIME=485 MS FOR 11/20,
 4667                                                                  ;97 MS FOR 11/45
 4668  016756  004737  020716                  JSR     PC,GT2RG                ;11/05
 4669  016762  104111                          ERROR   111             ;RK11 DID NOT INTERRUPT AFTER A SOFT
 4670                                                                  ;ERROR (SIMULATED) IN  SECTOR 0
 4671  016764  000417                          BR      2$
 4672
 4673  016766  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER (FROM RK11 INTRUPT,
 4674  016770  022626                          CMP     (SP)+,(SP)+     ;POP STACK (FROM WAT.INT)
 4675  016772  012777  004526  162402          MOV     #BADINT,@RKVEC  ;RESTORE RK11 INTERRUPT VECTOR
 4676                                                                  ;ADRES FOR UNEXPECTED INTERRUPTS
 4677  017000  005201                          INC     R1
 4678  017002  020177  162332                  CMP     R1,@RKDA        ;CHECK THAT RKDA INCREMENTED
 4679                                                                  ;BY ONLY 1 SECTOR BEFORE INTERRUPT
 4680                                                                  ;OCCURRED
 4681  017006  001406                          BEQ     2$
 4682  017010  010137  001162                  MOV     R1,$REG0        ;GET EXPCTD RKDA
 4683  017014  017737  162320  001164          MOV     @RKDA,$REG1     ;GET RKDA RECVD
 4684  017022  104003                          ERROR   3               ;RKDA SHOULD HAVE INCREMENTED BY
 4685                                                                  ;1 SECTOR ONLY, IF ALL CNTRL ACTION
 4686                                                                  ;HAD STOPPED AFTER SOFT ERROR
 4687                                                                  ;(SIMULATED) IN SECTOR 0. IT DID NOT.
 4688  017024                          2$:
 4689  017024  012746  000340                  MOV     #340,-(SP)
 4690  017030  012746  017036                  MOV     #64$,-(SP)
 4691  017034  000002                          RTI
 4692  017036                          64$:
 4693  017036  005077  162270                  CLR     @RKCS           ;CLEAR THE IDE BIT
 4694
 4695
 4696                                  ;;**************************************************************
 4697                                  ;*TEST 52        CHECK THE MEX BITS IN RKCS
 4698                                          ;*THIS TEST CHECKS OUT THE EXTENDED MEMORY BITS OF THE RKCS.
 4699                                          ;*THE RKBA IS SET TO 177776 AND A ONE WORD WRITE CHECK IS TRIED.
 4700                                          ;*THIS COULD GIVE RISE TO NXM ERROR, BUT EVEN THEN THE RKBA
 4701                                          ;*SHOULD OVERFLOW INTO THE MEX BITS. SIMILIARLY IT IS CHECKED
 4702                                          ;*THAT THE OVERFLOWING BIT CAN MAKE THE MEX BITS COUNT
 4703                                          ;*01,10,11,00.
 4704                                  ;;**************************************************************
 4705  017042  000004                  TST52:  SCOPE
 4706  017044  013700  001332                  MOV     RKCS,R0
 4707  017050  012701  177774                  MOV     #-4,R1          ;SET UP THE COUNT
 4708  017054  005002                          CLR     R2              ;INITIALIZE MEX BITS TO B SET IN RKCS
 4709  017056  012737  017064  001110          MOV     #1$,$LPERR      ;SET RETURN ADRES FOR
 4710                                                                  ;LUPING ON EROR (SW9)
 4711  017064  104417  000142          1$:     DELAY   ,142            ;TIME DELAY
 4712  017070  004737  021344                  JSR     PC,TSTRWS       ;WAIT FOR R/W/S RDY
 4713  017074  104016                          ERROR   16              ;R/W/S RDY IS NOT SET
 4714  017076  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
017100  010210                        MOV     R2,@R0          ;SET MEX BITS (AS IN R2) IN RKCS
017102  012777  177777  162224        MOV     #-1,@RKWC       ;WRT CHK 1 WORD
017110  013777  001350  162222        MOV     DRIVAD,@RKDA    ;THIS DISK ADRES, SEC 0, CYL 0
017116  012777  177776  162212        MOV     #177776,@RKBA   ;THIS BUS ADRES. NOTE THIS BA
                                                                ;IN CONJUCTION WITH MEX BITS OF RKCS
017124  052710  000007                BIS     #7,@R0          ;WRT CHK, GO
                                                                ;THERE MAY BE A NXM OR WCE BUT
                                                                ;WHATEVER THE CASE RKBA SHOULD
                                                                ;OVERFLOW MAKING THE MEX BITS COUNT
017130  104412                        CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
                                                                ;IF SO, SKIP THE EROR MESSAGE.
017132  104065                        ERROR   65              ;CNTRL RDY DID NOT SET AFTER WRT CHK
017134  010205                3$:     MOV     R2,R5
017136  062705  000020                ADD     #20,R5          ;MEX BITS SHOULD INCREMENT BY 1 TO THIS
017142  042705  000100                BIC     #100,R5         ;MASK OUT IDE BIT POSITION, IF SET
017146  011004                        MOV     @R0,R4          ;GET RKCS
017150  042704  177717                BIC     #177717,R4      ;MASK OUT ALL BITS EXCEPT MEX
017154  020504                        CMP     R5,R4           ;DID MEX BITS INCREMENT CORRECTLY?
017156  001405                        BEQ     4$              ;YES, BRANCH
017160  010537  001162                MOV     R5,$REG0        ;GET EXPCTD MEX BITS
017164  010437  001164                MOV     R4,$REG1        ;GET MEX BITS RECVD
017170  104112                        ERROR   112             ;MEX BITS DID NOT INCREMENT AS
                                                                ;'EXPCTD' WHEN RKBA OVERFLOWED.
                                                                ;NOTE THAT BIT POSITION 4 & 5
                                                                ;REFLECT MEX BITS 0 & 1 IN THE
                                                                ;ERROR MESSAGE.
017172  017703  162132        4$:     MOV     @RKER,R3        ;GET RKER
017176  010305                        MOV     R3,R5
017200  042703  003001                BIC     #3001,R3        ;MASK WCE,DLT,NXM BIT, IF SET
017204  001410                        BEQ     5$              ;BRANCH IF REST OF RKER CLR
017206  042705  177776                BIC     #177776,R5      ;MASK NON-WCE BITS
017212  010537  001162                MOV     R5,$REG0        ;THIS IS THE EXPCTD RKER
017216  017737  162106  001164        MOV     @RKER,$REG1     ;GET RKER RECVD
017224  104107                        ERROR   107             ;ERROR IN RKER.  IT SHOULD
                                                                ;BE AS EXPECTED IN
                                                                ;ERROR MESSAGE
017226  062702  000020        5$:     ADD     #20,R2          ;INCREMENT TO NXT MEX BIT
017232  005201                        INC     R1              ;HAVE U CHKD THE MEX BITS 4 TIMES?
017234  001313                        BNE     1$              ;IF NOT, LUP BACK

                        ;;**************************************************************
                        ;*TEST 53       TRANSFER FROM DISK TO TTY
                                ;* THIS TEST CHECKS THE HIGH ORDER BITS OF THE ADDRESS
                                ;* LINES.  FIRST A ONE WORD (100) IS WRITTEN ON SECTOR,
                                ;* 2, CYL 0.  THEN IT IS READ BACK, BUT THE NPR IS DONE
                                ;* NOT TO THE MEMORY, BUT THE TELETYPE BUFFER (TKS 177560)
                                ;* AND IT CHECKED THAT THE WORD WAS RECIEVED CORRECTLY.
```

```
                        ;*IF IT IS NOT, AN ERROR IS REPORTED. THIS TEST IS
                        ;*SKIPPED ON AN 11/05.
                        ;;*************************************************************
017236  000004         TST53:  SCOPE
017240  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
                                                                ;THIS CODE FINDS OUT IF THE CPU
                                                                ;IS AN 11/05 OR ELSE.
                                                                ;ON AN 11/05, R0 (177700) CAN BE
                                                                ;ADDRESSED AS A MEMORY LOCATION, BUT
                                                                ;ON ANY OTHER CPU IF 177700 IS REFERENCED
                                                                ;A TIME OUT WILL OCCUR.
017246  012737  017270  000004          MOV     #5$,@#4         ;SET UP TIME OUT VECTOR
017254  005737  177700                  TST     @#177700        ;REFERENCE R0
017260  012737  004462  000004          MOV     #BADTMO,@#4     ;R0 WAS REFERENCED W/O TIMEOUT
                                                                ;HENCE 11/05
017266  000520                          BR      TST54           ;;SKIP THIS TEST
017270  022626                  5$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
017272  012737  004462  000004          MOV     #BADTMO,@#4     ;RESTORE TIMEOUT VECTOR
017300  012746  000340                  MOV     #340,-(SP)
017304  012746  017312                  MOV     #64$,-(SP)
017310  000002                          RTI
017312                          64$:
017312  013700  001332                  MOV     RKCS,R0
017316  104413                          CNT.RESET               ;GO, DO CONTROL RESET
                                                                ;THIS IS A CALL FOR THE 'CNTRL-
                                                                ;RESET' ROUTINE. A CONTROL RESET IS
                                                                ;ISSUED AND AFTER A CERTAIN TIME
                                                                ;IF THE 'CNTRL RDY' DOES NOT SET
                                                                ;AN ERROR IS REPORTED. NOTE THAT
                                                                ;THE PC IN ERROR MESSAGE IS THE
                                                                ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                                ;THIS IS A VERY BASIC ERR8 IF IT
                                                                ;OCCURS GO BACK TO TEST 10
017320  012701  033240                  MOV     #OUTBUF,R1
017324  013704  001336                  MOV     RKBA,R4
017330  012711  000100                  MOV     #100,@R1        ;WRITE THIS WORD
017334  012777  177777  161772          MOV     #-1,@RKWC       ;WRITE 1 WORD
017342  013702  001350                  MOV     DRIVAD,R2
017346  052702  000002                  BIS     #2,R2           ;ON CYL 0, SEC 2
017352  010277  161762                  MOV     R2,@RKDA
017356  010114                          MOV     R1,@R4          ;FROM THIS MEMORY LOC
017360  012710  000003                  MOV     #3,@R0          ;WRITE, GO
017364  005003                          CLR     R3
017366  105710                  1$:     TSTB    @R0
017370  100410                          BMI     2$
017372  005203                          INC     R3
017374  001374                          BNE     1$
017376  004737  020702                  JSR     PC,GT4RG        ;GET RKCS, ER, DS
017402  010237  001202                  MOV     R2,$REG10       ;GET THE STARTING ADRES
017406  104416                          BRKDA4                  ;BREAK IT INTO DRV #, CYL, SUR, SEC #
017410  104031                          ERROR   31              ;CNTRL RDY DID NOT SET AFTER
                                                                ;WRITE OF 1 WORD ON CYL 0, SEC 2
017412  012777  177777  161714  2$:     MOV     #-1,@RKWC       ;READ 1 WORD
017420  010277  161714                  MOV     R2,@RKDA        ;FROM SEC 2, CYL 0
017424  013714  001144                  MOV     $TKS,@R4        ;INTO TTY STAUS REGISTER
017430  005077  161510                  CLR     @$TKS           ;CLEAR TTY KEY BRD STATUS REG
```

```
                                                ;READ, MEX BITS SET
017434  012710  000065          MOV     #65,@R0
017440  005003                  CLR     R3
017442  105710          3$:     TSTB    @R0
017444  100410                  BMI     4$
017446  005203                  INC     R3
017450  001374                  BNE     3$
017452  004737  020702          JSR     PC,GT4RG
017456  010237  001202          MOV     R2,$REG10       ;GET THE STARTING ADRES
017462  104416                  BRKDA4                  ;BREAK IT INTO DR#, CYL, SUR, SEC#
017464  104045                  ERROR   45              ;CNTRL RDY DIDN'T SET AFTER
                                                        ;READ OF 1 WORD FROM CYL 0, SEC 2.
                                                        ;IN EROR MSGE, <DSK-ADRES> GIVES
                                                        ;ADRES WHERE READ BEGAN. 'RKDA'
                                                        ;GIVES CONTENTS OF RKDA AT TIME OF EROR
017466  032737  000100  001144  4$:     BIT     #100,$TKS       ;WAS THE CORRECT WORD READ INTO
                                                        ;THE TTY  STATUS REGISTER?
017474  001015                  BNE     TST54           ;;YES, EXIT
017476  017705  161442          MOV     @$TKS,R5                ;GET THE WORD RECVD FROM DISK
017502  010537  001164          MOV     R5,$REG1
017506  052705  000100          BIS     #100,R5         ;THIS WORD WAS EXPCTD
017512  010537  001162          MOV     R5,$REG0        ;STORE EXPCTD WORD
017516  011437  001166          MOV     @R4,$REG2       ;GET RKBA
017522  011037  001170          MOV     @R0,$REG3       ;GET RKCS
017526  104115                  ERROR   115             ;DATA ERROR. A ONE WORD (100)
                                                        ;NPR WAS TRIED FROM DISK TO
                                                        ;TTY KEYBOARD STATUS REGISTER
                                                        ;(17756) . BIT 6 SHOULD HAVE BEEN
                                                        ; SET AS RESULT OF THIS
                                                        ;BUT IT WAS NOT


                        ;;*****************************************************************
                        ;*TEST 54        CHECK THAT RKBA CAN COUNT CORRECTLY
                                ;*THIS TEST CHECKS THAT RKBA CAN COUNT CORRECTLY. IT IS SET
                                ;*TO THE DESIRED INITIAL VALUE.  THEN A ONE WORD WRITE CHECK
                                ;*IS TRIED, WITH MEX (MEMORY EXTENSION) BITS SET. IF THERE 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.
                        ;;*****************************************************************
017530  000004          TST54:  SCOPE
017532  012737  000005  001206          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
017540  104421                  TST.SIN                 ;CHECK IF SIN SET, IF SET DRV RESET
017542  005001                  CLR     R1              ;INITIALIZE (VALUE OF RKBA)
017544  012702  000002                  MOV     #2,R2           ;INITIALIZE (INCMNTD VALUE OF RKBA)

017550  012737  017562  001110          MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
                                                        ;ON EROR

017556  013705  001336          MOV     RKBA,R5
017562  004737  021344  1$:     JSR     PC,TSTRWS       ;WAIT FOR R/W/S RDY
017566  104016                  ERROR   16              ;R/W/S RDY IS NOT SET
017570  104413                  CNT.RESET               ;DO CONTROL RESET
017572  012777  177777  161534          MOV     #-1,@RKWC       ;WRITE CHK 1 WORD
```

```
4883  017600  010115                          MOV     R1,@R5  ;THIS BUS ADRES
4884  017602  013777  001350  161530          MOV     DR!VAD,@RKDA    ;SET DISK ADRES
4885  017610  012777  000067  161514          MOV     #67,@RKCS       ;WRITE CHECK, GO, MEX BITS SET
4886  017616  104412                          CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
4887                                                                  ;!F SO, SKIP THE EROR MESSAGE.
4888  017620  104065                          ERROR   65              ;CNTRL RDY DID NOT SET AFTER
4889                                                                  ;WRT CHK WAS TRIED TO NXM LOC
4890                                                                  ;U MIGHT WANT TO USE TESTS
4891                                                                  ;CHECKING MEX BITS & NXM.
4892  017622  005237  001356                  INC     INDX1           ;ALLOW ONLY 5 ERRORS OF -BOVE KIND
4893  017626  001417                          BEQ     5$
4894
4895  017630  020215                  3$:     CMP     R2,@R5          ;DID RKBA INCREMENT BY 1 FROM
4896                                                                  ;ITS INITIAL VALUE?
4897  017632  001410                          BEQ     4$              ;YES, BRANCH
4898  017634  010137  001162                  MOV     R1,$REGO        ;GET EXPCTD RKBA
4899  017640  011537  001164                  MOV     @R5,$REG1       ;GET RKBA RECVD
4900  017644  104017                          ERROR   17              ;RKBA DID NOT INCREMENT BY
4901                                                                  ;1 FROM ITS INITIAL VALUE.
4902                                                                  ;ONE WORD WRT CHK WAS TRIED
4903                                                                  ;TO A NXM LOCATION. THERE
4904                                                                  ;WILL BE AN NXM ERROR,
4905                                                                  ;BUT STILL RKBA SHOULD
4906                                                                  ;INCREMENT BY 1 FROM ITS
4907                                                                  ;INITIAL VALUE.
4908  017646  005237  001360                  INC     INDX2           ;ALLOW ONLY 5 ERRORS OF
4909  017652  001405                          BEQ     5$              ;THE ABOVE KIND
4910  017654  060201                  4$:     ADD     R2,R1           ;SET NXT VALUE OF RKBA
4911  017656  010102                          MOV     R1,R2
4912  017660  062702  000002                  ADD     #2,R2           ;SET EXPCTD VALUE OF RKBA
4913  017664  001336                          BNE     1$              ;ALL DONE?
4914
4915  017666                          5$:                             ;DUMMY EXIT POINT
4916
4917
4918                                          ;;*****************************************************************
4919                                          ;*TEST 55       CHECK FOR RK-05F
4920                                                  ;*THIS TEST CHECKS RK-05F TYPE DRIVES
4921                                                  ;*TO INSURE THAT IF SEEKS ARE ISSUED ON ONE
4922                                                  ;*DRIVE, THE OTHER DRIVE BECOMES BUSY
4923                                          ;;*****************************************************************
4924  017666  000004                  1ST55:  SCOPE
4925  017670  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
4926  017676  005737  001404                  TST     FFLAG           ;SEE IF RK-05F
4927  017702  001403                          BEQ     1$              ;NOT F
4928  017704  004537  025056                  JSR     R5,FCHECK       ;SEE IF OTHER GOES BUSY
4929  017710  104120                          ERROR   120
4930
4931  017712                          1$:
4932
4933                                          ;;*****************************************************************
4934                                          ;*TEST 56       END OF PROGRAM
4935                                                  ;*THIS IS NOT A TEST, BUT A LINKAGE PROVIDED TO PERFORM
4936                                                  ;*THE ABOVE SUB-TESTS FOR ALL DRIVES THAT ARE PRESENT.
4937                                                  ;*NOTE THAT THE NEXT TEST- HARDWARE POLLING LOGIC-
4938                                                  ;*IS DONE USING ALL THE DRIVES THAT ARE INDICATED PRESENT.
```

```
                                        ;*DO NOT LOOP ON THIS 'TEST'.
                                ;;**************************************************************
017712  000004                  T$T56:  SCOPE
017714  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
017722  005237  001352                  INC     DRVDON          ;INCREMENT THE COUNT FOR THE NUMBER
                                                                ;OF DRIVES THAT ARE CHECKED
017726  004737  021412                  JSR     PC,DRESET       ;RESET THE DRIVE
017732  104026                          ERROR   26              ;R/W/S DIDN'T SET AFTER DRIVE RESET
017734  023737  001412  001352  BTEOP:  CMP     DRIVS,DRVDON    ;HAVE U TESTED ALL THE DRIVES
                                                                ;THAT ARE PRESENT?
017742  001405                          BEQ     1$              ;IF YES, EXIT
017744  062737  020000  001350          ADD     #20000,DRIVAD   ;ADRES THE NXT POSSIBLE DRIVE
017752  000137  004766                  JMP     NUDRV           ;GO BACK AND TEST THE NEXT
                                                                ;DRIVE PRESENT
017756  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.
                                ;;**************************************************************
017762  000004                  T$T57:  SCOPE
017764  012737  000005  001206          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
017772  005237  001440                  INC     SIZYET          ;FOUNR RK05F YET?
017776  001002                          BNE     25$             ;YES
020000  004737  025202                  JSR     PC,SIZEF        ;FIND WHICH ARE RK-05F
020004  005037  001436          25$:    CLR     PHYDRV          ;NUMBER OF ACTUAL DRIVES
020010  012700  001414                  MOV     #DRIV0,R0       ;TABLE
020014  005710                  23$:    TST     (R0)            ;DRIVE HERE+?
020016  001405                          BEQ     22$             ;NO
020020  005237  001436                  INC     PHYDRV          ;COUNT DRIVE
020024  005710                          TST     (R0)            ;RK05F?
020026  100001                          BPL     22$             ;NO
020030  005720                          TST     (R0)+           ;DONT COUNT F TWICE
020032  005720                  22$:    TST     (R0)+   ;NEXT DRIVE
020034  020027  001433                  CMP     R0,#DRIV7+1     ;ALL YET
020040  002765                          BLT     23$             ;NO
020042  005037  001406                  CLR     ODDEVN          ;EVEN DRIVES FIRST IF F
020046  005737  001412          T56:    TST     DRIVS           ;ANY DRIVES PRESENT?
020052  001002                          BNE     20$             ;YES
020054  000137  020560                  JMP     $EOP            ;NO
020060  005237  001434          20$:    INC     T56FLG
020064  013700  001332                  MOV     RKCS,R0
020070  005037  001356                  CLR     INDX1           ;FLAG TO INDICATE:
```

```
                                                        ;(INDX1)=0 POLLING DONE AFTER ALL
                                                        ;          DRIVES SEEK TO CYL 0
                                                        ;(INDX1)=1 POLLING DONE AFTER ALL
                                                        ;          DRIVES SEEK TO CYL 4
020074  005037  001360          15$:    CLR     INDX2   ;FLAG INDICATING TYPE OF INTERRUPT
                                                        ;SET TO NON-ZERO TO INDICATE
                                                        ;THAT THE INTERRUPT IS DUE TO
                                                        ;SEEK DONE
020100  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
020102  005737  001356          TST     INDX1   ;PERFORMING SEEKS TO CYL 4
020106  001002                  BNE     .+6     ;YES, BRANCH
020110  005002                  CLR     R2      ;NO
020112  000402                  BR      .+6
020114  012702  000200          MOV     #200,R2 ;SET ADRES FOR FOURTH CYLINDER
020120  012701  001414          MOV     #DRIV0,R1       ;INITIALIZE POINTER
020124  012703  177770          MOV     #-10,R3 ;SET COUNT FOR 8 DRIVES
020130  012705  033240          MOV     #OUTBUF,R5      ;INITIALIZE POINTER TO INDICATOR AREA
020134  005025                  CLR     (R5)+   ;CLEAR OUT THE 8-WORD INDICATOR
020136  005203                  INC     R3      ;AREA WHICH IS USED FOR DOING
020140  001375                  BNE     .-4     ;SOFTWARE POLLING LATER ON
020142  012703  177770          MOV     #-10,R3 ;SET COUNT FOR 8 POSSIBLE DRIVES
020146  012705  033240          MOV     #OUTBUF,R5      ;INITIALIZE POINTER TO INDICATOR AREA
020152                  1$:
020152  012746  000340          MOV     #340,-(SP)
020156  012746  020164          MOV     #64$,-(SP)
020162  000002                  RTI
020164                  64$:
020164  032711  000001          BIT     #BIT0,(R1)      ;IS THIS DRIVE PRESENT?
020170  001433                  BEQ     4$      ;IF NOT, BRANCH
020172  005711                  TST     (R1)    ;RK06F?
020174  100012                  BPL     17$     ;NO, CONTINUE
020176  032702  020000          BIT     #BIT13,R2       ;DRIVE EVEN?
020202  001404                  BEQ     16$     ;YES
020204  005737  001406          TST     ODDEVN  ;DO WE WANT ODD?
020210  001423                  BEQ     4$      ;NO, SO DO NOT TEST
020212  000403                  BR      17$     ;ADD THIS DRIVE TO LIST
020214  005737  001406  16$:     TST     ODDEVN  ;DO WE WANT EVEN?
020220  001017                  BNE     4$      ;NO, SO SKIP
020222  010215          17$:    MOV     R2,(R5) ;SET UP THIS WORD IN THE
                                                        ;INDICATOR AREA SHOWING THAT THIS
                                                        ;DRIVE (AS IN BITS 13-15 OF R2)
                                                        ;IS PRESENT
020224  042725  017777          BIC     #17777,(R5)+    ;MASK OUT UNWANTED BITS (CYL,SUR,SEC BITS)
020230  005004                  CLR     R4
020232  105710          2$:     TSTB    @R0     ;IS CNTRL RDY SET?
020234  100405                  BMI     3$      ;YES, BRANCH
020236  005204                  INC     R4      ;NO, WAIT FOR IT
```

```
5051  020240  001374                      BNE     2$              ;IF WAITED LONG REPORT ERROR
5052  020242  004737  020702              JSP     PC,GT4RG                ;GO, GET RKCS,ER,DS,DA
5053  020246  104021                      ERROR   21              ;CNTRL RDY DID NOT SET AFTER ACCEPTING
5054                                                              ;ADRES FROM PREVIOUS SEEK
5055  020250  010277  161064      3$:     MOV     R2,@RKDA        ;ADRES THIS DRIVE, CYL 0 OR CYL 4
5056                                                              ;(WHICHEVER THE CASE MAY BE)
5057  020254  012710  000111              MOV     #111,@R0        ;SEEK,GO,IDE SET
5058  020260  005721              4$:     TST     (R1)+           ;NEXT DRIVE DATA
5059  020262  062702  020000              ADD     #20000,R2       ;INCREMENT DRIVE ADRES (BITS 15,14,13)
5060  020266  005203                      INC     R3              ;TO NEXT ONE
5061  020270  001330                      BNE     1$              ;BRANCH BACK IF ALL DRIVES ARE
5062                                                              ;NOT CHECKED TO SEE IF THE NEXT
5063                                                              ;DRIVE IS PRESENT (& IF SO ISSUE A
5064                                                              ;SEEK TO IT)
5065                                                              ;BY NOW SEEKS HAVE BEEN ISSUED
5066                                                              ;TO ALL DRIVES PRESENT & POLLING
5067                                                              ;HAS BEGUN
5068  020272  005004                      CLR     R4
5069  020274  013702  001402      5$:     MOV     RKVEC,R2
5070  020300  012722  020332              MOV     #6$,(R2)+       ;SET ADRES FOR RK11 TO INTERUPT
5071  020304  012712  000340              MOV     #340,(R2)       ;SET PSW ON INTERUPT
5072  020310  013746  001400              MOV     RKPRI,-(SP)     ;DROP CPU PRIORITY TO 4 SO THAT
5073  020314  012746  020322              MOV     #18$,-(SP)              ;RK11 CAN INTERUPT
5074  020320  000002                      RTI
5075  020322  000240              18$:    NOP                     ;THIS IS A TIME LOOP DURING
5076  020324  005204                      INC     R4              ;WHICH ALL DRIVES PRESENT SHOULD
5077  020326  001375                      BNE     18$             ;INTERRUPT
5078  020330  000452                      BR      11$             ;BRANCH AND CHECK IF ALL AVAILABLE
5079                                                              ;DRIVES INTERRUPTED CORRECTLY
5080  020332  022626              6$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
5081  020334  005737  001360              TST     INDX2           ;WAS THIS FIRST INTERRUPT
5082                                                              ;DUE TO 'ADRES ACK' AFTER INITIATION
5083                                                              ;OF SEEK?
5084  020340  001021                      BNE     9$              ;IF YES, CHECK THE FOLLOWING
5085
5086  020342  032710  020000              BIT     #20000,@R0      ;CHECK THAT SCP IS NOT SET
5087  020346  001403                      BEQ     7$              ;BRANCH IF SCP CLEAR
5088  020350  011037  001162              MOV     @R0,$REG0       ;GET RKCS
5089  020354  104076                      ERROR   76              ;AFTER THE FIRST INTERRUPT WHICH
5090                                                              ;IS DUE TO INITIATION OF SEEK, SCP
5091                                                              ;SHOULD NOT HAVE SET. IT DID
5092  020356  017701  160744      7$:     MOV     @RKDS,R1
5093  020362  032701  160000              BIT     #160000,R1      ;RKDS BITS 15-13 SHLOULD BE CLR
5094  020366  001403                      BEQ     8$
5095  020370  010137  001162              MOV     R1,$REG0        ;GET RKDS
5096  020374  104050                      ERROR   50              ;SEEK, WITH IDE SET WAS ISSUED TO
5097                                                              ;ALL AVAILABLE DRIVES. THE FIRST
5098                                                              ;INTERUPT IS DUE TO SEEK INITIATED
5099                                                              ;BY FRST DRV. DRV ID BITS 13-15
5100                                                              ;SHOULD BE CLR AFTR THIS FRST INRUPT.
5101                                                              ;THEY WERE NOT IF THIS ERROR OCCURS.
5102  020376  005237  001360      8$:     INC     INDX2           ;SET UP FLAG INDICATING
5103                                                              ;THAT   E FIRST INTERRUPT DUE
5104                                                              ;TO INITIATION OF SEEK WAS
5105                                                              ;PROCESSED
5106  020402  000734                      BR      5$              ;GO BACK TO THE WAIT LOOP & WAIT
```

```
                                                ;FOR NEXT INTERRUPT FROM RK11
020404  013703  001436          9$:   MOV   PHYDRV,R3     ;SET COUNT OF # OF DRIVES PRESENT
020410  012705  033240                MOV   #OUTBUF,R5    ;INITIALIZE POINTER
020414  017701  160706                MOV   @RKDS,R1      ;GET RKDS
020420  042701  017777                BIC   #17777,R1     ;MASK BITS 0-12
                                                ;THE FOLLOWING CODE IS A SOFTWARE
                                                ;POLL WHICH FINDS OUT WHICH DRIVE
                                                ;CAUSED THE PRESENT INTERRUPT
                                                ;AND SETS UP A FLAG BIT FOR
                                                ;THE DRIVE #, INDICATING THAT
                                                ;THIS DRIVE # INTERRUPTED
020424  020125                      CMP   R1,(R5)+
020426  001411                      BEQ   10$           ;BRANCH IF INTERRUPTING DRIVE WAS FOUND
020430  005303                      DEC   R3            ;HAVE U CHKD ALL DRIVES PRESENT?
020432  001374                      BNE   .-6           ;IF NOT LUP BAK & CHK
                                                ;REPORT ERROR IF THE INTERRUPTING
                                                ;DRIVE # (AS IN RKDS 13-15) WAS NOT
                                                ;ANY ONE OF THOSE THAT ARE PRESENT
020434  010146                      MOV   R1,-(R6)      ;GET WORD TO B SHFTD RT
020436  004737  021106                JSR   PC,SHFTRT     ;GO SHIFT IT
020442  012637  001162                MOV   (R6)+,$REGO   ;THIS DRIVE # WAS RECVD IN RKDS AS
                                                ;THE INTERRUPTING DRIVE, BUT THIS
                                                ;DRIVE IS NOT PHYSICALY PRESENT
020446  104051                      ERROR 51            ;RKDS INDICATES AN INTERRUPTING
                                                ;DRIVE # (DURING H'WARE POLL) BUT
                                                ;THAT DRIVE IS ACTUALLY NOT PRESENT
020450  000401                      BR    10$+2
020452  005245          10$:  INC   -(R5)         ;SET UP FLAG INDICATING THAT
                                                ;THE INTERRUPT FOR THIS DRIVE
                                                ;(AFTER IT HAD COMPLETED ITS SEEK)
                                                ;WAS PROCESSED
020454  000707                      BR    5$            ;GO BAK & WAIT FOR FURTHER INTRUPTS
020456  013703  001436          11$:  MOV   PHYDRV,R3     ;GET # OF DRIVES
020462  012705  033240                MOV   #OUTBUF,R5    ;INITIALIZE POINTER

020466  105715          14$:  TSTB  (R5)          ;DID THIS DRIVE INTERRUPT?
020470  001006                      BNE   13$           ;YES, BRANCH
020472  011546                      MOV   (R5),-(R6)    ;GET THIS DRIVE #
020474  004737  021106                JSR   PC,SHFTRT     ;SHIFT IT TO THE RIGHT
020500  012637  001162                MOV   (R6)+,$REGO   ;THIS DRIVE # DID NOT INTERRUPT
                                                ;DURING H'WARE POLL
020504  104052                      ERROR 52            ;DRIVE # (AS IN $REGO) DID NOT
                                                ;INTERRUPT DURING HARDWARE POLL
020506  062705  000002          13$:  ADD   #2,R5         ;INCREMENT POINTER TO THE NEXT FLAG
020512  005303                      DEC   R3            ;CHKD FOR ALL DRIVES?
020514  001364                      BNE   14$           ;IF NOT LUP BACK

020516  005737  001356                TST   INDX1         ;DONE POLLING FOR SEEKS TO CYL 312?
020522  001004                      BNE   TSTEND        ;IF YES, EXIT
020524  005237  001356                INC   INDX1         ;IF NOT, INCREMENT FLAG
020530  000137  020074                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

020534  005237  001406          TSTEND: INC     ODDEVN          ;NOW ODD IF RK05F
020540  022737  000002  001406          CMP     #2,ODDEVN       ;SEE IF DONE
020546  001402                        BEQ     21$             ;ALL DONE
020550  000137  020046                JMP     T56             ;TEST AGAIN
020554  005037  001434          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

020560                          $EOP:
020560  000004                          SCOPE
020562  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
020566  005037  001206                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
020572  005237  001100                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
020576  042737  100000  001100          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
020604  005327                          DEC     (PC)+           ;;LOOP?
020606  000001                  $EOPCT: .WORD   1
020610  003022                          BGT     $DOAGN          ;;YES
020612  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
020614  000001                  $ENDCT: .WORD   1
020616  020606                          $EOPCT
020620  104401  020665                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
020624  013746  001100                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
020630  104405                          TYPOS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
020632  104401  020662                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
020636  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
020642  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
020644  000005                          RESET                   ;;CLEAR THE WORLD
020646  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
020650  000240                          NOP                     ;;SAVE ROOM
```

```
                                        NOP                     ;;FOR
                                        NOP                     ;;PDT11
                        $DOAGN:
                                        JMP     @(PC)+          ;;RETURN
                        $RTNAD:         .WORD   ST4
                        $ENULL:         .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
                        $ENDMG:         .ASCIZ  <15><12>/END PASS */
```

```
                                .SBTTL  GT2RG:  ROUTINE FOR GETTING RKCS,RKER

                        ;SUBROUTINE FOR TRANSFERRING THE CONTENTS OF RKCS, RKER
                        ;TO $REG0, $REG1 RESPECTIVELY BEFORE TYPING OUT AN ERROR MESSAGE.
                        ;CALL:  JSR     PC,GT2RG

                                .SBTTL  GT3RG:  ROUTINE FOR GETTING RKCS, RKER, RKDS

                        ;GT3RG
                        ;SUBROUTINE FOR TRANSFERRING THE CONTENTS OF RKCS, RKER, RKDS
                        ;TO  $REG0, $REG1, $REG2 RESPECTIVELY BEFORE TYPING OUT AN
                        ;ERROR MESSAGE.
                        ;CALL:  JSR     PC,GT3RG

                                .SBTTL  GT4RG:  ROUTINE FOR GETTING RKCS, RKER, RKDS, RKDA

                        ;GT4RG
                        ;SUBROUTINE FOR TRANSFERRING CONTENTS OF RKCS, RKER, RKDS
                        ;RKDA TO $REG0, $REG1, $REG2, $REG3 RESPECTIVELY BEFORE
                        ;TYPING OUT AN ERROR MESSAGE.
                        ;CALL:  JSR     PC,GT4RG

020702  017737  160432  001170  GT4RG:  MOV     @RKDA,$REG3     ;GET RKDA
020710  017737  160412  001166  GT3RG:  MOV     @RKDS,$REG2     ;GET RKDS
020716  017737  160406  001164  GT2RG:  MOV     @RKER,$REG1     ;GET RKER
020724  017737  160402  001162          MOV     @RKCS,$REG0
020732  000207                          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:
020734                                  TYPE    ,65$            ;;TYPE ASCIZ STRING
020734  104401  020742                  BR      64$             ;;GET OVER THE ASCIZ
020740  000406                  ;;65$:  .ASCIZ  <15><12>/EROR,PC=/
020756                          64$:
020756  010346                          MOV     R3,-(SP)
020760  104402                          TYPOC
020762  000207                          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
```

```
020764  010046                  BDA0:   MOV     R0,-(SP)        ;PUSH R0 ONTO THE STACK
020766  012700  001172                  MOV     #$REG3+2,R0     ;SET UP POINTER
020772  000403                          BR      BDAR
020774  010046                  BDA4:   MOV     R0,-(SP)        ;PUSH R0 ONTO THE STACK
020776  012700  001202                  MOV     #$REG7+2,R0     ;SET UP POINTER
021002  032777  020000  160130  BDAR:   BIT     #20000,@SWR     ;INHIBIT TYPEOUT?
021010  001034                          BNE     2$              ;YES, BRANCH TO EXIT POINT
021012  010146                          MOV     R1,-(SP)        ;PUSH R1 ON STACK
021014  010246                          MOV     R2,-(SP)        ;PUSH R2 ON STACK
021016  013701  001202                  MOV     $REG10,R1       ;GET THE ADDRESS WHICH
                                                                ;HAS TO BE BROKEN
021022  042701  177760                  BIC     #177760,R1      ;EXTRACT SECTOR BITS 0-3
021026  010140                          MOV     R1,-(R0)        ;MOVE SECTOR BITS TO $REG3 OR $REG7
021030  013701  001202                  MOV     $REG10,R1       ;GET THE DSK-ADRES TO BE BROKEN
021034  006201                          ASR     R1              ;SHIFT RIGHT 4 TIMES
021036  006201                          ASR     R1
021040  006201                          ASR     R1
021042  006201                          ASR     R1
021044  010102                          MOV     R1,R2           ;STORE THIS
021046  042702  177776                  BIC     #177776,R2      ;EXTRACCT THE SURFACE BIT
021052  010240                          MOV     R2,-(R0)        ;MOVE SURFACE BIT TO $REG3 OR $REG6
```

```
5331  021054  006201                          ASR     R1
5332  021056  010102                          MOV     R1,R2           ;STORE IT
5333  021060  042702  177400                  BIC     #177400,R2      ;EXTRACT THE CYLINDER BITS
5334  021064  010240                          MOV     R2,-(R0)        ;MOVE CYLINDER BITS TO $REG1 OR $REG5
5335  021066  000301                          SWAB    R1              ;SWAB HI-LO BYTES
5336  021070  042701  177770                  BIC     #177770,R1      ;EXTRACT THE DRIVE #
5337  021074  010140                          MOV     R1,-(R0)        ;MOVE DRIVE # TO $REG0 OR $REG4
5338
5339  021076  012602                          MOV     (SP)+,R2        ;RESTORE R2
5340  021100  012601                          MOV     (SP)+,R1        ;RESTORE R1
5341  021102  012600                  2$:     MOV     (SP)+,R0        ;RESTORE R0 FROM THE STACK
5342  021104  000002                          RTI                     ;RETURN FROM INTERRUPT, EXIT THIS
5343                                                                   ;ROUTINE
5344
5345
5346
5347                                          .SBTTL  SHFTRT: SHIFT RIGHT ROUTINE
5348
5349                                  ;SHFTRT
5350                                  ;THIS ROUTINE SHIFTS A WORD TO THE RIGHT 13 TIMES. THE WORD TO BE SHIFTED
5351                                  ;IS PUT ON THE STACK BEFORE ENTERING THIS ROUTINE AND IT IS POPPED UP
5352                                  ;FROM THE STACK AFTER THE SHIFT HAS BEEN DONE.
5353                                  ;CALL:  JSR     PC,SHFTRT
5354
5355  021106  012737  177763  021132  SHFTRT: MOV     #-15,2$         ;SET UP A COUNT OF 13
5356  021114  000241                          CLC                     ;CLEAR THA C BIT
5357  021116  006066  000002          1$:     ROR     2(R6)           ;ROTATE RIGHT THE WORD TO B SHFTD
5358  021122  005237  021132                  INC     2$              ;SHIFTED 13 TIMES?
5359  021126  001373                          BNE     1$              ;IF NOT LUP BAK & SHIFT
5360  021130  000207                          RTS     PC              ;EXIT FROM THIS SUBROUTINE
5361  021132  000000                  2$:     0
5362
5363
5364
5365
5366
5367                                          .SBTTL  CHKHE:  CHECK FOR 'ERR'OR
5368                                          .SBTTL  CHKHE1: CHECK FOR 'ERR'OR
5369
5370                                  ;;CHKHE
5371                                  ;THIS ROUTINE CHECKS IF 'HE' OR 'ERR' BITS IN RKCS ARE SET. IF ANY OF THE
5372                                  ;TWO BITS ARE SET, THE CONTENTS OF RKCS, ER, DS, AND DA ARE SAVED AND A
5373                                  ;RETURN IS MADE TO THE ERROR MESSAGE FOLLOWING THE 'JSR' CALL.
5374                                  ;AT THE TIME OF ENTRY 'DRIVAD' CONTAINS THE DISK ADDRESS WHICH IS TO
5375                                  ;BE BROKEN DOWN INTO DRIVE #, CYLINDER, SURFACE AND SECTOR #. THIS INFORMATION
5376                                  ;IS SAVED TO BE USED LATER FOR ERROR REPORTING. IF THE BITS ARE NOT SET,
5377                                  ;RETURN IS MADE TO SKIP THE ERROR MESSAGE.
5378
5379                                  ;CHKHE1
5380                                  ;THIS ROUTINE CHECKS IF 'HE' OR 'ERR' BITS IN RKCS ARE SET. IF ANY OF THE
5381                                  ;TWO BITS ARE SET, THE CONTENTS OF RKCS, ER, DS, AND DA ARE SAVED AND A
5382                                  ;RETURN IS MADE TO THE ERROR MESSAGE FOLLOWING THE 'JSR' CALL.
5383                                  ;AT THE TIME OF ENTRY R1 CONTAINS THE DISK ADDRESS WHICH IS TO BE BROKEN
5384                                  ;DOWN INTO DRIVE #, CYLINDER, SURFACE AND SECTOR #. THIS INFORMATION IS
5385                                  ;SAVED TO BE USED LATER FOR ERROR REPORTING. IF THE BITS ARE NOT SET,
5386                                  ;RETURN IS MADE TO SKIP THE ERROR MESSAGE.
```

```
                                                                ;SAVE THE DISK ADRES
021134  010137  001202          CHKHE1: MOV     R1,$REG1O       ;SAVE THE DISK ADRES
021140  000403                          BR      CHE1            ;NO
                                                                ;GET RKCS,ER,DS, DA
021142  013737  001350  001202  CHKHE:  MOV     DRIVAD,$REG1O   ;SAVE THE DISK ADRES
021150  032777  140000  160154  CHE1:   BIT     #140000,@RKCS   ;IS 'HE' OR 'ERR' BIT SET?
021156  001467                          BEQ     CRETRN          ;NO
021160  004737  020702                  JSR     PC,GT4RG        ;GET RKCS,ER,DS, DA
021164  104416                          BRKDA4                  ;GO TO 'BDA4' & BREAK CONTENTS O
                                                                ;$REG1O INTO DR#, CYL, SUR, SEC BITS
021166  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.
021170  013705  001350  CHKDA:  MOV     DRIVAD,R5       ;RKDA SHOULD INCREMENT TO THIS
021174  005205                  INC     R5              ;AFTER DATA TRANSFER IS DONE
021176  020577  160136  CHKDA1: CMP     R5,@RKDA        ;DID RKDA INCREMENT CORRECTLY?
021202  001455                  BEQ     CRETRN          ;IF YES, BRANCH
                                                        ;IF NOT, REPORT ERROR
021204  010537  001202          MOV     R5,$REG1O       ;GET EXPCTD RKDA
021210  104415                  BRKDA0                  ;GO TO 'BDA0' & BREAK  CONTENTS OF
                                                        ;$REG1O INTO DR #,CYL,SUR,SEC BITS
021212  017737  160122  001202  MOV     @RKDA,$REG1O    ;GET ACTUAL RKDA
021220  104416                  BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                        ;$REG1O INTO DR #,CYL,SUR,SEC BITS
021222  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.
021224  005777  160104  CHKWC:  TST     @RKWC           ;DID WORD COUNT OVERFLOW TO 0?
021230  001442                  BEQ     CRETRN          ;IF YES, BRANCH
                                                        ;IF NOT, ERROR
021232  017737  160076  001162  MOV     @RKWC,$REG0     ;GET RKWC
021240  017737  160074  001164  MOV     @RKDA,$REG1     ;GET RKDA
021246  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.
```

```
5443  021250  005777  160054          CHKER:  TST     @RKER           ;DID ANY BIT IN RKER SET?
5444  021254  001430                          BEQ     CRETRN          ;NO, BRANCH
5445                                                                  ;YES, ERROR
5446  021256  004737  020710                  JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
5447
5448  021262  000207                          RTS     PC              ;RETURN TO THE ERROR MESSAGE
5449
5450
5451                                  ;CHKECLR
5452                                  ;THIS ROUTINE CHECKS THAT RKER IS CLEAR. IF NOT, THE CONTENTS OF RKER
5453                                  ;ARE SAVED AND A RETURN IS MADE TO THE ERROR MESSAGE FOLLOWING THE "JSR"
5454                                  ;CALL. IF RKER IS CLEAR THE ERROR MESSAGE IS SKIPPED ON RETURN.
5455
5456  021264  005777  160040          CHKECLR: TST    @RKER           ;ANY BIT IN RKER SET?
5457  021270  001422                          BEQ     CRETRN          ;NO
5458  021272  013737  001330  001162          MOV     RKER,$REG0      ;GET ADRES OF RKER
5459  021300  017737  160024  001164          MOV     @RKER,$REG1     ;GET CONTENTS OF RKER
5460  021306  000207                          RTS     PC              ;RETURN TO THE ERROR MESSAGE
5461
5462
5463                                  ;CHKCCLR
5464                                  ;THIS ROUTINE CHECKS THAT RKCS IS CLEAR. IF NOT, THE CONTENTS OF RKCS ARE
5465                                  ;SAVED AND A RETURN IS MADE TO THE ERROR MESSAGE. IF RKCS IS CLEAR THE
5466                                  ;ERROR MESSAGE IS SKIPPED ON RETURN.
5467  021310  022777  000200  160014  CHKCCLR: CMP     #200,@RKCS      ;IS RKCS CLEAR?
5468  021316  001407                          BEQ     CRETRN          ;YES
5469  021320  013737  001332  001162          MOV     RKCS,$REG0      ;SAVE ADRES OF RKCS
5470  021326  017737  160000  001164          MOV     @RKCS,$REG1     ;SAVE THE CONTENT OF RKCS
5471  021334  000207                          RTS     PC              ;RETURN TO THE ERROR MESSAGE
5472
5473  021336  062716  000002          CRETRN: ADD     #2,(SP)         ;SKIP ERROR MESSAGE ON
5474  021342  000207                          RTS     PC              ;RETURN
5475
5476
5477                                          .SBTTL  TSTRWS: WAIT FOR R/W/S RDY ROUTINE
5478
5479                                  ;TSTRWS
5480                                  ;THIS ROUTINE WAITS FOR R/W/S RDY TO SET. WHEN IT SETS, THE RETURN PC
5481                                  ;IS INCREMENTED SO THAT ON RETURN (TO THE MAIN PROGRAM) THE ERROR
5482                                  ;MESSAGE FOLLOWING THE 'JSR' CALL IS SKIPPED. IF R/W/S RDY DOES NOT SET
5483                                  ;THEN A RETURN IS MADE TO THE ERROR MESSAGE (FOLLOWING THE 'JSR' CALL).
5484                                  ;WAITING TIME IS APPROX. 1040 MS FOR 11/20, APPROX. 208 MS FOR 11/45
5485                                  ;CALL:  JSR     TSTRWS
5486
5487  021344  013777  001350  157766  TSTRWS: MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE
5488  021352  005037  001366                  CLR     TIMER           ;INITIALIZE COUNT
5489  021356  032777  000100  157742  1$:     BIT     #100,@RKDS      ;DID R/W/S RDY SET?
5490  021364  001007                          BNE     2$              ;YES, BRANCH
5491  021366  005237  001366                  INC     TIMER           ;WAIT FOR R/W/S RDY
5492  021372  001371                          BNE     1$              ;ERROR IF IT'S NOT SET BY NOW
5493  021374  017737  157726  001162          MOV     @RKDS,$REG0     ;GET RKDS
5494  021402  000207                          RTS     PC              ;EXIT (TO ERROR FOOLOWING 'JSR TSTRWS')
5495
5496  021404  062716  000002          2$:     ADD     #2,(SP)         ;ADJUST RETURN ADRES TO SKIP OVER
5497                                                                  ;ERROR (FOLLOWING 'JSR TSTRWS')
5498  021410  000207                          RTS     PC              ;EXIT
```

```
                        .SBTTL  DRESET: DRIVE RESET ROUTINE

                ;DRESET
                ;THIS ROUTINE DOES A DRIVE RESET ON THE DRIVE WHOOSE ADDRESS IS IN
                ;RKDA. MULTIPLE RETURN ADDRESSES FOR THIS ROUTINE ARE PROVIDED.
                ;IF THERE IS NO ERROR (R/W/S RDY SETS WITHIN CERTAIN TIME) , THEN BEFORE
                ;EXITING FROM THIS ROUTINE THE RETURN ADDRESS IS INCREMENTED BY 2, TO SKIP
                ;THE ERROR MESSAGE ON RETURN. IF THERE IS AN ERROR, THE 3 REGISTERS (CS,ER,DS)
                ;ARE STORED AND THEN A NORMAL EXIT IS MADE FROM THIS ROUTINE TO THE
                ;ERROR MESSAGE FOLLOWING THE CALL FOR THIS ROUTINE.
                ;CALL:   JSR     PC,DRESET

021412  005037  001364        DRESET: CLR     COUNT1          ;INITIALIZE THE COUNT
021416  013777  001350  157714        MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE
021424  012777  000015  157700        MOV     #15.,@RKCS      ;DRIVE RESET, GO
021432  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
021434  032777  000100  157664  1$:     BIT     #100.,@RKDS             ;DID R/W/S RDY SET?
021442  001013                        BNE     2$
021444  012746  177770                MOV     #-10.,-(SP)     ;PUSH COUNT ON SP
021450  005216                        INC     (SP)            ;COUNT IT DOWN
021452  001376                        BNE     .-2
021454  005726                        TST     (SP)+           ;POP UP SP
021456  005237  001364                INC     COUNT1                  ;IF NOT WAIT
021462  001364                        BNE     1$                      ;WAITED LONG?
021464  004737  020702                JSR     PC,GT4RG
021470  000402                        BR      2$+4
021472  062716  000002        2$:     ADD     #2,@R6
021476  000207                        RTS     PC


                        .SBTTL  TSTSIN: CHECK 'SIN' ROUTINE

                ;TSTSIN
                ;THIS ROUTINE CHECKS IF 'SIN' IS SET, IF IT IS SET A
                ;DRIVE RESET IS DONE TO CLEAR 'SIN' AND INITIALIZE POSITIONER.
                ;CALL:           TST.SIN
                ;IF ON DOING DRIVE RESET R/W/S RDY DOES NOT SET  A MESSAGE
                ;       ERROR PC=XXXXXX  IS GIVEN.
                ;XXXXXX=PC IN THE MAIN PROGRAM WHERE 'TST.SIN' CALL IS LOCATED.

021500  013777  001350  157632  TSTSIN: MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE
021506  032777  001000  157612          BIT     #1000,@RKDS     ;IS SIN SET?
021514  001403                        BEQ     1$
```

```
                021516  004737  021412                  JSR     PC,DRESET       ;GO DO DRIVE RESET, SIN  SET
                021522  000401                          BR      2$              ;REPORT ERROR
                021524  000002                  1$:     RTI
                021526  022777  020000  157404  2$:     BIT     #SW13,@SWR      ;INHIBIT TYPEOUT?
                021534  001373                          BNE     1$              ;IF YES, SKIP TYPEOUT
                021536  104401  021544                  TYPE    ,65$            ;;TYPE ASCIZ STRING
                021542  000406                          BR      64$             ;;GET OVER THE ASCIZ
                                                65$:     .ASCIZ  /ERROR PC= /
                021560                          64$:
                021560  011646                          MOV     (SP),-(SP)
                021562  062716  177776                  ADD     #-2,(SP)        ;GET THE PC WHERE 'TST.SIN' IS LOCATED
                021566  104402                          TYPOC                   ;GO TYPE OUT PC
                021570  000755                          BR      1$


                                                        .SBTTL  DELAY:  TIME DELAY ROUTINE

                                                ;DELAY
                                                ;THIS ROUTINE PROVIDES  A VARIABLE TIME DELAY. THE CALL FOR THIS
                                                ;ROUTINE IS AN ENCODED 'TRAP' INSTRUCTION.
                                                ;CALL:          DELAY   ,N      N IS ANY OCTAL NO. FROM 1 TO 177777
                                                ;THE DELAY PROVIDED IS 7.5N US (CONVERT N TO DECIMAL) FOR 11/20
                                                ;1.5N US FOR 11/45
                                                ;IF THE USER WANTS TO CHANGE THE DELAY TIME (EXMP: SHORTER DELAY TO
                                                ;GET A TIGHTER SCOPE LOOP) THE VARIABLE 'N' FOLLOWING 'DELAY' SHOULD
                                                ;BE CHANGED TO SUIT THE INDIVIDUAL NEED.

                021572  017637  000000  001366  DELAY:  MOV     @(SP),TIMER     ;GET 'AMOUNT' (N) FOR WHICH
                021600  062716  000002                  ADD     #2,(SP)         ;DELAY IS TO BE PROVIDED
                                                                                ;ADJUST STACK POINTER TO SKIP OVER 'N'
                021604  005337  001366          1$:     DEC     TIMER           ;COUNT DOWN TO 0
                021610  001375                          BNE     1$

                021612  000002                          RTI                     ;RETURN TO MAIN PROGRAM


                                                        .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'.
```

```
                021614  017637  000000  001366  WATINT: MOV     @(SP),TIMER     ;GET 'AMOUNT' (N) FOR WHICH
                021622  062716  000002          ADD     #2,(SP)         ;WAITING IS TO BE DONE
                                                                        ;ADJUST STACK POINTER FOR CORRECT RETURN
                021626  013746  000400          MOV     RKPRI,-(SP)     ;DROP CPU PRIORITY SO THAT RK11 CAN
                021632  012746  021640          MOV     #1$,-(SP)       ; INTERRUPT
                021636  000002                  RTI
                021640  005337  001366  1$:     DEC     TIMER           ;WAIT FOR RK11 TO INTERRUPT
                021644  001375                  BNE     1$
                                                                        ;IF INTERRUPT HAS NOT OCCURED BY NOW
                                                                        ;RETURN AND REPORT ERROR
                021646  000002                  RTI                     ;EXIT


                                        ;WATIME

                021650  005000  WATIME: CLR     R0
                021652  005001          CLR     R1
                021654  005200  1$:     INC     R0
                021656  001376          BNE     1$
                021660  105201          INCB    R1
                021662  001374          BNE     1$
                021664  000207          RTS     PC


                                        .SBTTL  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
                                        ;       ---             ;RETURN HERE IF NO ERROR

                021666  005037  001366  CH.CRDY: CLR TIMER
                021672  105777  157434  1$:     TSTB    @RKCS           ;CNTRL RDY SET?
                021676  100406                  BMI     2$              ;YES
                021700  005237  001366          INC     TIMER
                021704  001372                  BNE     1$              ;NO, WAIT
                021706  004737  020702          JSR     PC,GT4RG        ;SAVE RKCS, ER, DS, DA
                021712  000002                  RTI

                021714  062716  000002  2$:     ADD     #2,(SP)         ;ADJUST RETURN ADDRESS TO
                021720  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
021722  012777  000001  157402  CN.RST: MOV     #1,@RKCS        ;ISSUE A CONTROL RESET
021730  012737  177500  001170          MOV     #-300,$REG3     ;SET UP COUNT
021736  000402                          BR      CN.RDY+4                ;SKIP OVER CN.RDY
021740  005037  001170          CN.RDY: CLR     $REG3
021744  105777  157362          1$:     TSTB    @RKCS           ;DID CNTRL-RDY SET?
021750  100435                          BMI     3$              ;YES, EXIT
021752  005237  001170                  INC     $REG3           ;WAITED LONG?
021756  001372                          BNE     1$              ;IF NOT, GO BAK & WAIT
021760  032777  020000  157152  2$:     BIT     #SW13,@SWR      ;INHIBIT TYPEOUT?
021766  001026                          BNE     3$              ;IF YES, SKIP TYPEOUT
021770  104401                          TYPE
021772  001245                          MSG3
021774  104401  022002                  TYPE    ,65$            ;;TYPE ASCIZ STRING
022000  000403                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/PC=/
022010                          64$:
022010  011646                          MOV     (SP),-(SP)
022012  162716  000002                  SUB     #2,(SP)
022016  104402                          TYPOC                   ;GO TYPE PC IN THE MAIN PROGRAM,
                                                                ; WHERE ERROR OCCURRED
022020  104401  022026                  TYPE    ,67$            ;;TYPE ASCIZ STRING
022024  000404                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  / RKCS=/
022036                          66$:
022036  017746  157270                  MOV     @RKCS,-(SP)     ;GET RKCS
022042  104402                          TYPOC                   ;GO TYPE IT

022044  000002          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

022046                          $SCOPE:
022046  104407                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
022050  032777  040000  157062  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
022056  001111                          BNE     $OVER           ;;YES IF SW14=1
                                ;*****START OF CODE FOR THE XOR TESTER*****
022060  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
022062  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
022066  012737  022106  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
022074  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
022100  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
022104  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
022106  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
022110  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
022114  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
022116                          6$:;*****END OF CODE FOR THE XOR TESTER*****
022116  032777  000400  157014          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
022124  001404                          BEQ     2$              ;;BR IF NO
022126  127737  157006  001102          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
022134  001462                          BEQ     $OVER           ;;BR IF YES
022136  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
022142  001421                          BEQ     3$              ;;BR IF NO
022144  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
022152  101015                          BHI     3$              ;;BR IF NO
022154  032777  001000  156756          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
022162  001404                          BEQ     4$              ;;BR IF NO
022164  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
022172  000443                          BR      $OVER
022174  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
022200  005037  001206                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
022204  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
022206  032777  004000  156724  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
022214  001011                          BNE     1$              ;;BR IF YES
022216  005737  001100                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
022222  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
022224  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
022230  023737  001206  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
022236  002021                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
022240  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
022246  013737  022316  001206          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
022254  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
022260  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
022264  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
```

```
5779  022270  005037  001210                CLR     $ESCAPE          ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
5780  022274  112737  000001  001115        MOVB    #1,$ERMAX        ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
5781  022302  013777  001102  156632 $OVER: MOV     $TSTNM,@DISPLAY  ;;DISPLAY TEST NUMBER
5782  022310  013716  001106               MOV     $LPADR,(SP)      ;;FUDGE RETURN ADDRESS
5783  022314  000002                       RTI                      ;;FIXES PS
5784  022316  000050              $MXCNT: 50                       ;;MAX. NUMBER OF ITERATIONS
5785
5786
5787                                ;;*******************************************************************************
5788
5789                                .SBTTL  ERROR HANDLER ROUTINE
5790
5791                                ;*SW15=1        HALT ON ERROR
5792                                ;*SW13=1        INHIBIT ERROR TYPEOUTS
5793                                ;*SW10=1        TESTING ON SIMULATOR
5794                                ;*SW09=1        LOOP ON ERROR
5795                                ;*SW12=1        CYCLE ON ERROR TO PREVIOUS 'SCOPE'
5796                                ;*SW06=1        DROP DRIVE AFTER MAXIMUM (ALLOWABLE) ERRORS ON THE DRIVE
5797                                ;*GO TO $ERRTYP ON ERROR
5798
5799  022320  104407              $ERROR: CKSWR                            ;CHECK FOR SOFTWARE SWITCH REGISTER REQUEST
5800  022322  105237  0011C3      7$:     INCB    $ERFLG                   ;SET THE ERROR FLAG
5801  022326  001775                      BEQ     7$                       ;DON'T LET THE FLAG GO TO ZERO
5802  022330  013777  001102  156604      MOV     $TSTNM,@DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
5803  022336  005237  001112      1$:     INC     $ERTTL                   ;COUNT THE NUMBER OF ERRORS
5804
5805  022342  032777  000100  156570      BIT     #BIT6,@SWR               ;DESELECT DRIVE SW SET?
5806  022350  001404                      BEQ     6$                       ;NO
5807  022352  023727  001112  000005      CMP     $ERTTL,#5                ;MORE THAN 5 ERRORS ON THIS DRIVE?
5808  022360  101047                      BHI     8$                       ;YES, DESELCT THE DRIVE
5809
5810  022362  011637  001116      6$:     MOV     (SP),$ERRPC              ;GET ADDRESS OF ERROR INSTRUCTION
5811  022366  162737  000002  001116      SUB     #2,$ERRPC
5812  022374  117737  156516  001114      MOVB    @$ERRPC,$ITEMB           ;STRIP AND SAVE THE ERROR ITEM CODE
5813  022402  032777  020000  156530      BIT     #SW13,@SWR               ;SKIP TYPEOUT IF SET
5814  022410  001004                      BNE     2$                       ;SKIP TYPEOUTS
5815  022412  004737  022632              JSR     PC,@#$ERRTYP             ;GO TO USER ERROR ROUTINE
5816  022416  104401  001213              TYPE    ,$CRLF
5817  022422  005777  156512      2$:     TST     @SWR                     ;HALT ON ERROR
5818  022426  100002                      BPL     3$                       ;SKIP IF CONTINUE
5819  022430  000000                      HALT                             ;HALT ON ERROR!
5820  022432  104407                      CKSWR                            ;CHECK FOR SOFTWARE SWITCH REGIATER REQUEST
5821  022434  032777  010000  156476  3$: BIT     #SW12,@SWR               ;SW 12 SET?
5822  022442  001402                      BEQ     .+6                      ;NO, BRANCH
5823  022444  013716  001106              MOV     $LPADR,(SP)              ;ADJUST RETURN ADRES FOR SW12
5824  022450  032777  001000  156462      BIT     #SW09,@SWR               ;LOOP ON ERROR SWITCH SET?
5825  022456  001402                      BEQ     4$                       ;BR IF NO
5826  022460  013716  001110              MOV     $LPERR,(SP)              ;FUDGE RETURN FOR LOOPING
5827  022464  005737  001210      4$:     TST     $ESCAPE                  ;CHECK FOR AN ESCAPE ADDRESS
5828  022470  001402                      BEQ     5$                       ;BR IF NONE
5829  022472  013716  001210              MOV     $ESCAPE,(SP)             ;FUDGE RETURN ADDRESS FOR ESCAPE
5830  022476  000002              5$:     RTI                              ;RETURN
5831
5832  022500  005737  001434      8$:     TST     T56FLG                   ;IF EROR WAS IN LAST TEST (POLL)
5833                                                                       ;DROP ALL THE DRIVES
5834  022504  001407                      BEQ     10$
```

```
 5835  022506  104401  001303            TYPE    ,MSG5
 5836  022512  005037  001412            CLR     DRIVS
 5837  022516  022626                    CMP     (SP)+,(SP)+
 5838  022520  000137  020560            JMP     $EOP
 5839  022524  013746  001354    10$:    MOV     DRVPTR,-(SP)        ;DROP THE DRIVE FROM THE
 5840  022530  162716  000002            SUB     #2,(SP)             ;SELECTION LIST
 5841  022534  013746  001350            MOV     DRIVAD,-(SP)        ;DRIVE ADDR TO STACK
 5842  022540  004737  021106            JSR     PC,SHFTRT           ;RIGHT JUSTIFY
 5843  022544  042716  000001            BIC     #1,(R6)             ;MAKE EVEN
 5844  022550  062716  001414            ADD     #DRIVO,(SP)         ;POINTS TO TABLE FOR EVEN DRIVE
 5845  022554  042776  100000  000000    BIC     #BIT15,@(R6)        ;TEST REMAINING DRIVE AS RKOSE
 5846  022562  062716  000002            ADD     #2,(R6)             ;POINT TO ODD
 5847  022566  042736  100000            BIC     #BIT15,@(SP)+       ;TEST AS RK-OSE
 5848  022572  012736  010000            MOV     #BIT12,@(SP)+       ;INDICATE THIS DRIVE DROPPED
 5849  022576  104401  001272            TYPE    ,MSG4
 5850  022602  013746  001350            MOV     DRIVAD,-(R6)        ;PUSH DRIVE # ON STACK
 5851  022606  004737  021106            JSR     PC,SHFTRT           ;SHIFT IT BEFORE TYPING
 5852  022612  104402                    TYPOC                       ;TYPE OUT DRIVE #
 5853  022614  104401  001315            TYPE    ,MSG6
 5854  022620  005337  001412            DEC     DRIVS               ;DECREMENT # OF DRIVES PRESNT
 5855  022624  022626            9$:     CMP     (SP)+,(SP)+         ;RESTORE STACK
 5856  022626  000137  017734            JMP     BTEOP               ;GO BACK TO THE END OF PROGRM
 5857                                                                ;LINKAGE.
 5858
 5859                            .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
 5860
 5861                            ;;*****************************************************************
 5862                            ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETEPMINE WHICH
 5863                            ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
 5864                            ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
 5865
 5866  022632            $ERRTYP:
 5867  022632  104401  001213            TYPE    ,$CRLF              ;;"CARRIAGE RETURN" & "LINE FEED"
 5868  022636  010046                    MOV     R0,-(SP)            ;;SAVE R0
 5869  022640  005000                    CLR     R0                  ;;PICKUP THE ITEM INDEX
 5870  022642  153700  001114            BISB    @#$ITEMB,R0
 5871  022646  001004                    BNE     1$                  ;;IF ITEM NUMBER IS ZERO, JUST
 5872                                                                ;;TYPE THE PC OF THE ERROR
 5873  022650  013746  001116            MOV     $ERRPC,-(SP)        ;;SAVE $ERRPC FOR TYPEOUT
 5874                                                                ;;ERROR ADDRESS
 5875  022654  104402                    TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 5876  022656  000426                    BR      6$                  ;;GET OUT
 5877  022660  005300            1$:     DEC     R0                  ;;ADJUST THE INDEX SO THAT IT WILL
 5878  022662  006300                    ASL     R0                  ;;      WORK FOR THE ERROR TABLE
 5879  022664  006300                    ASL     R0
 5880  022666  006300                    ASL     R0
 5881  022670  062700  001442            ADD     #$ERRTB,R0          ;;FORM TABLE POINTER
 5882  022674  012037  022704            MOV     (R0)+,2$            ;;PICKUP "ERROR MESSAGE" POINTER
 5883  022700  001404                    BEQ     3$                  ;;SKIP TYPEOUT IF NO POINTER
 5884  022702  104401                    TYPE                        ;;TYPE THE "ERROR MESSAGE"
 5885  022704  000000            2$:     .WORD   0                   ;;"ERROR MESSAGE" POINTER GOES HERE
 5886  022706  104401  001213            TYPE    ,$CRLF              ;;"CARRIAGE RETURN" & "LINE FEED"
 5887  022712  012037  022722    3$:     MOV     (R0)+,4$            ;;PICKUP "DATA HEADER" POINTER
 5888  022716  001404                    BEQ     5$                  ;;SKIP TYPEOUT IF 0
 5889  022720  104401                    TYPE                        ;;TYPE THE "DATA HEADER"
 5890  022722  000000            4$:     .WORD   0                   ;;"DATA HEADER" POINTER GOES HERE
```

```
5891  022724  104401  001213          TYPE    $CRLF           ;;"CARRIAGE RETURN" & "LINE FEED"
5892  022730  011000                  5$:     MOV     @(R0),R0        ;;PICKUP "DATA TABLE" POINTER
5893  022732  001004                          BNE     7$              ;;GO TYPE THE DATA
5894  022734  012600                  6$:     MOV     (SP)+,R0        ;;RESTORE R0
5895  022736  104401  001213                  TYPE    $CRLF           ;;"CARRIAGE RETURN" & "LINE FEED"
5896  022742  000207                          RTS     PC              ;;RETURN
5897  022744                          7$:
5898  022744  013046                          MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
5899  022746  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
5900  022750  005710                          TST     (R0)            ;;IS THERE ANOTHER NUMBER?
5901  022752  001770                          BEQ     6$              ;;BR IF NO
5902  022754  104401  022762                  TYPE    ,8$             ;;TYPE TWO(2) SPACES
5903  022760  000771                          BR      7$              ;;LOOP
5904  022762  020040    000           8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
5905          022766                          .EVEN
5906
5907                                  .SBTTL  TYPE ROUTINE
5908
5909                                  ;;**************************************************************
5910                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
5911                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
5912                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
5913                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
5914                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
5915                                  ;*
5916                                  ;*CALL:
5917                                  ;*1) USING A TRAP INSTRUCTION
5918                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
5919                                  ;*OR
5920                                  ;*      TYPE
5921                                  ;*      MESADR
5922                                  ;*
5923
5924  022766  105737  001157  $TYPE:   TSTB    $TPFLG          ;;IS THERE A TERMINAL?
5925  022772  100002                          BPL     1$              ;;BR IF YES
5926  022774  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
5927  022776  000407                          BR      3$              ;;LEAVE
5928  023000  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
5929  023002  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
5930  023006  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
5931  023010  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
5932  023012  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
5933  023014  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
5934  023016  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
5935  023022  000002                          RTI                     ;;RETURN
5936  023024  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
5937  023030  001430                          BEQ     8$
5938  023032  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
5939  023036  001006                          BNE     5$
5940  023040  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
5941  023042  104401                          TYPE                    ;;TYPE A CR AND LF
5942  023044  001213                          $CRLF
5943  023046  105037  023202                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
5944  023052  000755                          BR      2$              ;;GET NEXT CHARACTER
5945  023054  004737  023136          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
5946  023060  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
```

```
 5947  023064  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
 5948  023066  013746  001154                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 5949                                                                ;;AND THE NULL CHAR.
 5950  023072  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 5951  023076  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 5952  023100  004737  023136                JSR     PC,$TYPEC       ;;GO TYPE A NULL
 5953  023104  105337  023202                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 5954  023110  000770                        BR      7$              ;;LOOP
 5955
 5956                                        ;HORIZONTAL TAB PROCESSOR
 5957
 5958  023112  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 5959  023116  004737  023136        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
 5960  023122  132737  000007  023202        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 5961  023130  001372                        BNE     9$              ;;TAB STOP
 5962  023132  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
 5963  023134  000724                        BR      2$              ;;GET NEXT CHARACTER
 5964  023136  105777  156006        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 5965  023142  100375                        BPL     $TYPEC
 5966  023144  116677  000002  156000        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 5967  023152  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 5968  023160  001003                        BNE     1$              ;;BRANCH IF NO
 5969  023162  105037  023202                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 5970  023166  000406                        BR      $TYPEX          ;;EXIT
 5971  023170  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 5972  023176  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
 5973  023200  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
 5974  023202  000000        $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
 5975  023204  000207        $TYPEX: RTS     PC
 5976
 5977
 5978                                        .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 5979
 5980                                        ;;**************************************************************
 5981                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 5982                                        ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 5983                                        ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 5984                                        ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 5985                                        ;*REPLACED WITH SPACES.
 5986                                        ;*CALL:
 5987                                        ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 5988                                        ;*      TYPDS                   ;;GO TO THE ROUTINE
 5989
 5990  023206                        $TYPDS:
 5991  023206  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 5992  023210  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 5993  023212  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 5994  023214  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 5995  023216  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 5996  023220  012746  020200                MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 5997  023224  016605  000020                MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 5998  023230  100004                        BPL     1$              ;;BR IF INPUT IS POS.
 5999  023232  005405                        NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 6000  023234  112766  000055  000001        MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 6001  023242  005000                1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 6002  023244  012703  023422                MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
```

```
                        023250  112723  000040          MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
                        023254  005002                  2$:     CLR     R2      ;;CLEAR THE BCD NUMBER
                        023256  016001  023412                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
                        023262  160105                  3$:     SUB     R1,R5   ;;FORM THIS BCD DIGIT
                        023264  002402                          BLT     4$      ;;BR IF DONE
                        023266  005202                          INC     R2      ;;INCREASE THE BCD DIGIT BY 1
                        023270  000774                          BR      3$
                        023272  060105                  4$:     ADD     R1,R5   ;;ADD BACK THE CONSTANT
                        023274  005702                          TST     R2      ;;CHECK IF BCD DIGIT=0
                        023276  001002                          BNE     5$      ;;FALL THROUGH IF 0
                        023300  105716                          TSTB    (SP)    ;;STILL DOING LEADING 0'S?
                        023302  100407                          BMI     7$      ;;BR IF YES
                        023304  106316                  5$:     ASLB    (SP)    ;;MSD?
                        023306  103003                          BCC     6$      ;;BR IF NO
                        023310  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
                        023316  052702  000060          6$:     BIS     #'0,R2  ;;MAKE THE BCD DIGIT ASCII
                        023322  052702  000040          7$:     BIS     #' ,R2  ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
                        023326  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
                        023330  005720                          TST     (R0)+   ;;JUST INCREMENTING
                        023332  020027  000010                  CMP     R0,#10  ;;CHECK THE TABLE INDEX
                        023336  002746                          BLT     2$      ;;GO DO THE NEXT DIGIT
                        023340  003002                          BGT     8$      ;;GO TO EXIT
                        023342  010502                          MOV     R5,R2   ;;GET THE LSD
                        023344  000764                          BR      6$      ;;GO CHANGE TO ASCII
                        023346  105726                  8$:     TSTB    (SP)+   ;;WAS THE LSD THE FIRST NON-ZERO?
                        023350  100003                          BPL     9$      ;;BR IF NO
                        023352  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
                        023360  105013                  9$:     CLRB    (R3)    ;;SET THE TERMINATOR
                        023362  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
                        023364  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
                        023366  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
                        023370  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
                        023372  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
                        023374  104401  023422                  TYPE    ,$DBLK  ;;NOW TYPE THE NUMBER
                        023400  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
                        023406  012616                          MOV     (SP)+,(SP)
                        023410  000002                          RTI                     ;;RETURN TO USER
                        023412  023420                  $DTBL:  10000.
                        023414  001750                          1000.
                        023416  000144                          100.
                        023420  000012                          10.
                        023422  000004                  $DBLK:  .BLKW   4

                                                        .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                                        ;;**************************************************************
                                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                                        ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                                        ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                                        ;*CALL:
                                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                                        ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                                        ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                                        ;*      .BYTE   M               ;;M=1 OR 0
                                                        ;*                                      ;;1=TYPE LEADING ZEROS
                                                        ;*                                      ;;0=SUPPRESS LEADING ZEROS
```

```
                                        ;*
                                        ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                        ;*$TYPOS OR $TYPOC
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPON                   ;;CALL FOR TYPEOUT
                                        ;*
                                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOC                   ;;CALL FOR TYPEOUT
023432  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
023436  116637  000001  023655          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
023444  112637  023657                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
023450  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
023454  000406                          BR      $TYPON
023456  112737  000001  023655  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
023464  112737  000006  023657          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
023472  112737  000005  023654  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
023500  010346                          MOV     R3,-(SP)        ;;SAVE R3
023502  010446                          MOV     R4,-(SP)        ;;SAVE R4
023504  010546                          MOV     R5,-(SP)        ;;SAVE R5
023506  113704  023657                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
023512  005404                          NEG     R4
023514  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
023520  110437  023656                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
023524  113704  023655                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
023530  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
023534  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
023536  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
023540  000404                          BR      3$              ;;GO DO MSB
023542  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
023544  006105                          ROL     R5
023546  006105                          ROL     R5
023550  010503                          MOV     R5,R3
023552  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
023554  105337  023656                  DECB    $OMODE          ;;TYPE THIS DIGIT?
023560  100016                          BPL     7$              ;;BR IF NO
023562  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
023566  001002                          BNE     4$              ;;TEST FOR 0
023570  005704                          TST     R4              ;;SUPPRESS THIS 0?
023572  001403                          BEQ     5$              ;;BR IF YES
023574  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
023576  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
023602  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
023606  110337  023652                  MOVB    R3,8$           ;;SAVE FOR TYPING
023612  104401  023652                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
023616  105337  023654          7$:     DECB    $OCNT           ;;COUNT BY 1
023622  003347                          BGT     2$              ;;BR IF MORE TO DO
023624  002402                          BLT     6$              ;;BR IF DONE
023626  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
023630  000744                          BR      2$              ;;GO DO THE LAST DIGIT
023632  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
023634  012604                          MOV     (SP)+,R4        ;;RESTORE R4
023636  012603                          MOV     (SP)+,R3        ;;RESTORE R3
```

```
6115  023640  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
6116  023646  012616                          MOV     (SP)+,(SP)
6117  023650  000002                          RTI                     ;;RETURN
6118  023652     000          8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
6119  023653     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
6120  023654     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
6121  023655     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
6122  023656  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
6123
6124                          .SBTTL  TTY INPUT ROUTINE
6125
6126                          ;;**************************************************************
6127                          .ENABL  LSB
6128
6129                          ;;**************************************************************
6130                          ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
6131                          ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
6132                          ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
6133                          ;*WHEN OPERATING IN TTY FLAG MODE.
6134  023660  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
6135  023666  001074                  BNE     15$             ;;BRANCH IF NO
6136  023670  105777  155250                  TSTB    @$TKS           ;;CHAR THERE?
6137  023674  100071                  BPL     15$             ;;IF NO, DON'T WAIT AROUND
6138  023676  117746  155244                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
6139  023702  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
6140  023706  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
6141  023712  001062                  BNE     15$             ;;NO, RETURN TO USER
6142  023714  123727  001134  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
6143  023722  001456                  BEQ     15$             ;;BRANCH IF YES
6144
6145  023724  104401  024545                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
6146  023730  104401  024552  $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
6147  023734  013746  000176                  MOV     $WREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
6148  023740  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
6149  023742  104401  024563                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
6150  023746  005046          19$:    CLR     -(SP)           ;;CLEAR COUNTER
6151  023750  005046                  CLR     -(SP)           ;;THE NEW SWR
6152  023752  105777  155166  7$:     TSTB    @$TKS           ;;CHAR THERE?
6153  023756  100375                  BPL     7$              ;;IF NOT TRY AGAIN
6154
6155  023760  117746  155162                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
6156  023764  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
6157
6158
6159
6160  023770  021627  000025  9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
6161  023774  001005                  BNE     10$             ;;BRANCH IF NOT
6162  023776  104401  024540                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
6163  024002  062706  000006  20$:     ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
6164  024006  000757                  BR      19$             ;;LET'S TRY IT AGAIN
6165
6166
6167  024010  021627  000015  10$:     CMP     (SP),#15        ;;IS IT A <CR>?
6168  024014  001022                  BNE     16$             ;;BRANCH IF NO
6169  024016  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
6170  024022  001403                  BEQ     11$             ;;BRANCH IF YES
```

```
 6171  024024  016677  000002  155106          MOV     2(SP),@SWR      ;;SAVE NEW SWR
 6172  024032  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
 6173  024036  104401  001213          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
 6174  024042  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
 6175  024050  001003                          BNE     15$             ;;BRANCH IF NOT
 6176  024052  012777  000100  155064          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
 6177  024060  000002          15$:    RTI                             ;;RETURN
 6178  024062  004737  023136          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
 6179  024066  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
 6180  024072  002420                          BLT     18$             ;;BRANCH IF YES
 6181  024074  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
 6182  024100  003015                          BGT     18$             ;;BRANCH IF YES
 6183  024102  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
 6184  024106  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
 6185  024112  001403                          BEQ     17$             ;;BRANCH IF YES
 6186  024114  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
 6187  024116  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
 6188  024120  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
 6189  024122  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
 6190  024126  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
 6191  024132  000707                          BR      7$              ;;GET THE NEXT ONE
 6192  024134  104401  001212          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
 6193  024140  000720                          BR      20$             ;;SIMULATE CONTROL-U
 6194                                  .DSABL  LSB
 6195
 6196
 6197                                  ;;**************************************************************
 6198                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 6199                                  ;*CALL:
 6200                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
 6201                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
 6202                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
 6203                                  ;
 6204
 6205  024142  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
 6206  024144  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
 6207  024152  105777  154766          1$:     TSTB    @$TKS           ;;WAIT FOR
 6208  024156  100375                          BPL     1$              ;;A CHARACTER
 6209  024160  117766  154762  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
 6210  024166  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
 6211  024174  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
 6212  024202  001013                          BNE     3$              ;;BRANCH IF NO
 6213  024204  105777  154734          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
 6214  024210  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
 6215  024212  117746  154730                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
 6216  024216  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 6217  024222  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 6218  024226  001366                          BNE     2$              ;;IF NOT DISCARD IT
 6219  024230  000750                          BR      1$              ;;YES, RESUME
 6220  024232  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
 6221  024240  002407                          BLT     4$              ;;BRANCH IF YES
 6222  024242  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
 6223  024250  003003                          BGT     4$              ;;BRANCH IF YES
 6224  024252  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
 6225  024260  000002          4$:     RTI                             ;;GO BACK TO USER
 6226                                  ;;**************************************************************
```

```
MAINDEC-11-D2RAA-D     MACY11 27.1006)  04-OCT-76  16:06  PAGE 114
D2RAAC.P11    22-SEP-76 08:47          TTY INPUT ROUTINE                                SEQ 0131

                                       ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                       ;*CALL:
                                       ;*      RDLIN                    ;;INPUT A STRING FROM THE TTY
                                       ;*      RETURN HERE              ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                       ;*                               ;;TERMINATOR WILL BE A BYTE OF ALL 1'S
                                       ;
024262  010346                 $RDLIN: MOV     R3,-(SP)         ;;SAVE R3
024264  005046                         CLR     -(SP)            ;;CLEAR THE RUBOUT KEY
024266  012703  024516           1$:   MOV     #$TTYIN,R3       ;;GET ADDRESS
024272  022703  024540           2$:   CMP     #$TTYIN+22,R3    ;;BUFFER FULL?
024276  101456                         BLOS    4$               ;;BR IF YES
024300  104410                         RDCHR                    ;;GO READ ONE CHARACTER FROM THE TTY
024302  112613                         MOVB    (SP)+,(R3)       ;;GET CHARACTER
024304  122713  000177           10$:  CMPB    #177,(R3)        ;;IS IT A RUBOUT
024310  001022                         BNE     5$               ;;BR IF NO
024312  005716                         TST     (SP)             ;;IS THIS THE FIRST RUBOUT?
024314  001007                         BNE     6$               ;;BR IF NO
024316  112737  000134  024514       MOVB    #'\,9$           ;;TYPE A BACK SLASH
024324  104401  024514                 TYPE    ,9$
024330  012716  177777               MOV     #-1,(SP)         ;;SET THE RUBOUT KEY
024334  005303                 6$:   DEC     R3               ;;BACKUP BY ONE
024336  020327  024516               CMP     R3,#$TTYIN       ;;STACK EMPTY?
024342  103434                       BLO     4$               ;;BR IF YES
024344  111337  024514               MOVB    (R3),9$          ;;SETUP TO TYPEOUT THE DELETED CHAR.
024350  104401  024514               TYPE    ,9$              ;;GO TYPE
024354  000746                       BR      2$               ;;GO READ ANOTHER CHAR.
024356  005716                 5$:   TST     (SP)             ;;RUBOUT KEY SET?
024360  001406                       BEQ     7$               ;;BR IF NO
024362  112737  000134  024514       MOVB    #'\,9$           ;;TYPE A BACK SLASH
024370  104401  024514               TYPE    ,9$
024374  005016                       CLR     (SP)             ;;CLEAR THE RUBOUT KEY
024376  122713  000025           7$:   CMPB    #25,(R3)         ;;IS CHARACTER A CTRL U?
024402  001003                       BNE     8$               ;;BR IF NO
024404  104401  024540               TYPE    ,$CNTLU          ;;TYPE A CONTROL "U"
024410  000726                       BR      1$               ;;GO START OVER
024412  122713  000022           8$:   CMPB    #22,(R3)         ;;IS CHARACTER A "^R"?
024416  001011                       BNE     3$               ;;BRANCH IF NO
024420  105013                       CLRB    (R3)             ;;CLEAR THE CHARACTER
024422  104401  001213               TYPE    ,$CRLF           ;;TYPE A "CR" & "LF"
024426  104401  024516               TYPE    ,$TTYIN          ;;TYPE THE INPUT STRING
024432  000717                       BR      2$               ;;GO PICKUP ANOTHER CHACTER
024434  104401  001212           4$:   TYPE    ,$QUES           ;;TYPE A '?'
024440  000712                       BR      1$               ;;CLEAR THE BUFFER AND LOOP
024442  111337  024514           3$:   MOVB    (R3),,9$         ;;ECHO THE CHARACTER
024446  104401  024514               TYPE    ,9$
024452  122723  000015               CMPB    #15,(R3)+        ;;CHECK FOR RETURN
024456  001305                       BNE     2$               ;;LOOP IF NOT RETURN
024460  105063  177777               CLRB    -1(R3)           ;;CLEAR RETURN (THE 15)
024464  104401  001214               TYPE    ,$LF             ;;TYPE A LINE FEED
024470  005726                       TST     (SP)+            ;;CLEAN RUBOUT KEY FROM THE STACK
024472  012603                       MOV     (SP)+,R3         ;;RESTORE R3
024474  011646                       MOV     (SP),-(SP)       ;;ADJUST THE STACK AND PUT ADDRESS OF THE
024476  016666  000004  000002       MOV     4(SP),2(SP)      ;;      FIRST ASCII CHARACTER ON IT
024504  012766  024516  000004       MOV     #$TTYIN,4(SP)
024512  000002                       RTI                      ;;RETURN
024514     000           9$:   .BYTE   0                        ;;STORAGE FOR ASCII CHAR. TO TYPE
```

```
                                        .BYTE   0               ;;TERMINATOR
                        $TTYIN: .BLKB   22              ;;RESERVE 22 BYTES FOR TTY INPUT
005015      000         $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
006507  000C12          $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
053523  020122          $MSWR:  .ASCIZ  <15><12>/SWR = /
020075      000
047040  053505          $MNEW:  .ASCIZ  /  NEW = /
036440  000040
                                                        ;CONTROL U, RUBOUT CAPABILITY

                        .SBTTL  TRAP DECODER

                        ;;**************************************************************
                        ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                        ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                        ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                        ;*GO TO THAT ROUTINE.

010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
000002                          MOV     2(SP),R0        ;;GET TRAP ADDRESS
005740                          TST     -(R0)           ;;BACKUP BY 2
111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
006300                          ASL     R0              ;;POSITION FOR INDEXING
016000  024630                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
000200                          RTS     R0              ;;GO TO ROUTINE


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

011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
000002                          RTI                     ;;RESTORE THE PSW


                        .SBTTL  TRAP TABLE

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

                        ;       ROUTINE
                        ;       -------
024616                  $TRPAD: .WORD   $TRAP2
022766                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
023456                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
023432                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
023472                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
023206                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

023730                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

023660                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
024142                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
024262                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE

021666                          CH.CRDY ;;CALL=CHKCRDY  TRAP+12(104412) CHECK CONTROL READY

021722                          CN.RST  ;;CALL=CNT.RESET        TRAP+13(104413) CONTROL RESET ROUTINE
```

```
6339  024660  021740                          CN.RDY  ;;CALL=CNT.RDY  TRAP+14(104414) WAIT FOR CNTRL RDY TO SET
6340
6341  024662  020764                          BDA0    ;;CALL=BRKDAC   TRAP+15(104415) BREAK RKDA INTO DR #,CYL,SUR,SEC BITS
6342
6343  024664  020774                          BDA4    ;;CALL=BRKDA4   TRAP+16(104416) BREAK RKDA INTO DR #,CYL,SUR,SEC BITS
6344
6345  024666  021572                          DELA.Y  ;;CALL=DELAY    TRAP+17(104417) TIME DELAY ROUTINE
6346
6347  024670  021614                          WATINT  ;;CALL=WAT.INT  TRAP+20(104420) WAIT FOR RK11 INTERRUPT ROUTINE
6348
6349  024672  021500                          TSTSIN  ;;CALL=TST.SIN  TRAP+21(104421) TEST SIN ROUTINE
6350
6351
6352                                  .SBTTL  POWER DOWN AND UP ROUTINES
6353
6354                                  ;;**********************************************************************
6355                                  ;POWER DOWN ROUTINE
6356  024674  012737  025040  000024  $PWRON: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
6357  024702  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
6358  024710  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
6359  024712  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
6360  024714  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
6361  024716  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
6362  024720  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
6363  024722  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
6364  024724  017746  154210                  MOV     @$WR,-(SP)      ;;PUSH @SWR ON STACK
6365  024730  010637  025044                  MOV     SP,$SAVR6       ;;SAVE SP
6366  024734  012737  024746  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
6367  024742  000000                          HALT
6368  024744  000776                          BR      .-2             ;;HANG UP
6369
6370                                  ;;**********************************************************************
6371                                  ;POWER UP ROUTINE
6372  024746  012737  025040  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
6373  024754  013706  025044                  MOV     $SAVR6,SP       ;;GET SP
6374  024760  005037  025044                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
6375  024764  005237  025044          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
6376  024770  001375                          BNE     1$              ;;OF  WORD
6377  024772  012677  154142                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
6378  024776  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
6379  025000  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
6380  025002  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
6381  025004  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
6382  025006  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
6383  025010  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
6384  025012  012737  024674  000024          MOV     #$PWRON,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
6385  025020  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
6386  025026  104401                          TYPE                    ;REPORT THE POWER FAILURE
6387  025030  025046          $PWRMG: .WORD   $POWER                  ;;POWER FAIL MESSAGE POINTER
6388  025032  012716                  MOV     (PC)+,(SP)              ;;RESTART AT PFSTRT
6389  025034  004630          $PWRAD: .WORD   PFSTRT                  ;;RESTART ADDRESS
6390  025036  000002                  RTI
6391  025040  000000          $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
6392  025042  000776                  BR      .-2                     ;; BEFORE THE POWER DOWN WAS COMPLETE
6393  025044  000000          $SAVR6: 0                               ;;PUT THE SP HERE
6394  025046  005015  047520  042527  $POWER: .ASCIZ  <15><12>/"POWER"
```

```
                                        .EVEN
025056  004737  021412          FCHECK: JSR     PC,DRESET               ;RESETB DRIVE
025062  104026                          ERROR   26
025064  104413                          CNT.RESET
025066  013737  001350  025200          MOV     DRIVAD,DRHOLD           ;SAVE DRIVE ADRR
025074  032737  020000  001350          BIT     #20000,DRIVAD           ;SEE IF ODD
025102  001404                          BEQ     1$
025104  042737  020000  001350          BIC     #20000,DRIVAD           ;MAKE EVEN
025112  000403                          BR      2$
025114  052737  020000  001350  1$:     BIS     #20000,DRIVAD           ;MAKE ODD
025122  013777  001350  154210  2$:     MOV     DRIVAD,@RKDA    ;DRIVE ADDR
025130  012777  000011  154174          MOV     #11,@RKCS               ;DRIVE SEEK
025136  104414                          CNT.RDY
025140  013777  025200  154172          MOV     DRHOLD,@RKDA            ;OTHER DRIVE
025146  104414                          CNT.RDY
025150  032777  000100  154150          BIT     #100,@RKDS              ;HEAEDS IN MOTIONN?
025156  001001                          BNE     3$                      ;NO SO RK-05J
025160  005725                          TST     (R5)+                   ;YES RK-05F
025162  013737  025200  001350  3$:     MOV     DRHOLD,DRIVAD           ;RESTORE ADDR
025170  004737  021412                  JSR     PC,DRESET               ;WAIT FOR RESET
025174  104026                          ERROR   26
025176  000205                          RTS     R5
025200  000000          DRHOLD: 0
025202  005037  001350  SIZEF:  CLR     DRIVAD          ;START AT DRO
025206  012700  001414          MOV     #DRIV0,R0               ;TABLE OF AVAIL DRIVES
025212  005710          4$:     TST     (R0)                    ;THIS DRIVE HERE?
025214  001413                  BEQ     2$                      ;NO
025216  005760  000002          TST     2(R0)                   ;COMPLEMENT HERE?
025222  001410                  BEQ     2$                      ;NO
025224  004537  025056          JSR     R5,FCHECK               ;SEE IF F MODEL
025230  000405                  BR      2$                      ;J MODEL
025232  052710  100000          BIS     #100000,(R0)            ;SET SIGN FOR F
025236  052760  100000  000002  BIS     #100000,2(R0)           ;BOTH DRIVES
025244  005720          2$:     TST     (R0)+
025246  005720                  TST     (R0)+                   ;NEXT PAIR OF DRIVES
025250  062737  040000  001350  ADD     #40000,DRIVAD           ;NEXT ACTUL ADDR
025256  022700  001433          CMP     #DRIV7+1,R0             ;CHECKED ALL?
025262  003353                  BGT     4$                      ;NOT YET
025264  000207                  RTS     PC

                        ;ERROR MESSAGES

                        .SBTTL  ERROR MESSAGES

025266  045522  041527  042440  EM11:   .ASCIZ  /RKWC EROR/
025274  047522  000122
025300  044523  020116  051511  EM12:   .ASCIZ  /SIN IS SET/
025306  051440  052105     000
025313     122  041113  020101  EM13:   .ASCIZ  /RKBA EROR/
025320  051105  051117     000
```

```
6451  025325  122  042113  020101  EM16:   .ASCIZ  /RKDA WRONG AFTER 'SSE'/
6452  025332  051127  047117  020107
6453  025340  043101  042524  020122
6454  025346  051447  042523  000047
6455
6456  025354  045522  051504  042440  EM21:   .ASCIZ  /RKDS EROR/
6457  025362  047522  000122
6458
6459  025366  050104  020114  042523  EM30:   .ASCIZ  /DPL SET/
6460  025374  000124
6461
6462  025376  051104  020125  042523  EM31:   .ASCIZ  /DRU SET/
6463  025404  000124
6464
6465  025406  045522  032460  041040  EM32:   .ASCIZ  /RKDS BIT NOT SET/
6466  025414  052111  047040  052117
6467  025422  051440  052105  000
6468
6469  025427  104  054522  041040  EM33:   .ASCIZ  /DRY BIT NOT SET/
6470  025434  052111  047040  052117
6471  025442  051440  052105  000
6472
6473  025447  123  045517  042040  EM34:   .ASCIZ  /SOK DIDN'T SET/
6474  025454  042111  023516  020124
6475  025462  042523  000124
6476
6477  025465  042523  026503  047103  EM35:   .ASCIZ  /SEC-CNTR DIDN'T COUNT TO 0/
6478  025474  051124  042040  042111
6479  025502  023516  020124  047503
6480  025510  047125  020124  047524
6481  025516  030040  000
6482
6483  025521  123  041505  041455  EM36:   .ASCIZ  /SEC-CNTR DIDN'T INCRMNT/
6484  025526  052116  020122  044504
6485  025534  047104  052047  044440
6486  025542  041516  046522  052116
6487  025550  000
6488
6489  025551  123  041505  041455  EM37:   .ASCIZ  /SEC-COUNTR INCRMENTED WRONG/
6490  025556  052517  052116  020122
6491  025564  047111  051103  042515
6492  025572  052116  042105  053440
6493  025600  047522  043516  000
6494
6495  025605  104  042111  023516  EM40:   .ASCIZ  /DIDN'T GET SC=SA FOR THIS SECTR/
6496  025612  020124  042507  020124
6497  025620  041523  051475  020101
6498  025626  047506  020122  044124
6499  025634  051511  051440  041505
6500  025642  051124  000
6501
6502  025645  105  047522  026522  EM41:   .ASCIZ  "EROR-R/W/S RDY SHOULD BE SET"
6503  025652  027522  027527  020123
6504  025660  042122  020131  044123
6505  025666  052517  042114  041040
6506  025674  020105  042523  000124
```

```
6507
6508  025702  047125  054105  042520  EM43:   .ASCIZ  /UNEXPECTED RK11 INTERRUPT/
6509  025710  052103  042105  051040
6510  025716  030513  020061  047111
6511  025724  042524  051122  050125
6512  025732  000124
6513
6514  025734  047103  051124  020114  EM44:   .ASCIZ  /CNTRL RDY DIDN'T SET AFTER SEEK OR DR RESET/
6515  025742  042122  020131  044504
6516  025750  047104  052047  051440
6517  025756  052105  040440  052106
6518  025764  051105  051440  042505
6519  025772  020113  051117  042040
6520  026000  020122  042522  042523
6521  026006  000124
6522
6523  026010  051105  020122  051117  EM45:   .ASCIZ  /ERR OR HE BIT SET ON SEEK OR DR RESET/
6524  026016  044040  020105  044502
6525  026024  020124  042523  020124
6526  026032  047117  051440  042505
6527  026040  020113  051117  042040
6528  026046  020122  042522  042523
6529  026054  000124
6530
6531  026056  045522  051105  041040  EM46:   .ASCIZ  /RKER BIT, ON SEEK OR DR RESET/
6532  026064  052111  020054  047117
6533  026072  051440  042505  020113
6534  026100  051117  042040  020122
6535  026106  042522  042523  000124
6536
6537  026114  045522  051503  041440  EM47:   .ASCIZ  /RKCS CHNGD AFTR FUNCTION WAS DONE/
6538  026122  047110  042107  040440
6539  026130  052106  020122  052506
6540  026136  041516  044524  047117
6541  026144  053440  051501  042040
6542  026152  047117  000105
6543
6544  026156  027522  027527  020123  EM50:   .ASCIZ  "R/W/S RDY DIDN'T CLEAR"
6545  026164  042122  020131  044504
6546  026172  047104  052047  041440
6547  026200  042514  051101     000
6548
6549  026205     122  053457  051457  EM51:   .ASCIZ  "R/W/S RDY DIDN'T SET AFTR SEEK OR DR RESET"
6550  026212  051040  054504  042040
6551  026220  042111  023516  020124
6552  026226  042523  020124  043101
6553  026234  051124  051440  042505
6554  026242  020113  051117  042040
6555  026250  020122  042522  042523
6556  026256  000124
6557
6558  026260  045522  040504  041440  EM52:   .ASCIZ  /RKDA CHNGD AFTR SEEK/
6559  026266  047110  042107  040440
6560  026274  052106  020122  042523
6561  026302  045505     000
6562
```

```
6563  026305  000103  052116  046122  EM53:   .ASCIZ  /CNTRL RDY DIDN'T CLR AS GO WAS SET/
6564  026312  051040  054504  042040
6565  026320  042111  023516  020124
6566  026326  046103  020122  051501
6567  026334  043440  020117  040527
6568  026342  020123  042523  000124
6569
6570  026350  047103  051124  020114  EM54:   .ASCIZ  "CNTRL RDY DIDN'T SET ON WRT/FMT STARTING FROM <DSK-ADRES>"
6571  026356  042122  020131  044504
6572  026364  047104  052047  051440
6573  026372  052105  047440  020116
6574  026400  051127  027524  046506
6575  026406  020124  052123  051101
6576  026414  044524  043516  043040
6577  026422  047522  020115  042074
6578  026430  045523  040455  051104
6579  026436  051505  000076
6580
6581  026442  042510  047440  020122  EM55:   .ASCIZ  "HE OR ERR ON WRT/FMT STARTING FROM <DSK-ADRES>"
6582  026450  051105  020122  047117
6583  026456  053440  052122  043057
6584  026464  052115  051440  040524
6585  026472  052122  047111  020107
6586  026500  051106  046517  036040
6587  026506  051504  026513  042101
6588  026514  042522  037123  000000
6589
6590  026521  000122  042113  020101  EM56:   .ASCIZ  /RKDA INCRMNTD WRONG ON WRT-FMT/
6591  026526  047111  051103  047115
6592  026534  042124  053440  047522
6593  026542  043516  047440  020116
6594  026550  051127  026524  046506
6595  026556  000124
6596
6597  026560  045522  041527  042040  EM57:   .ASCIZ  /RKWC DIDN'T OVRFLO ON WRT FMT/
6598  026566  042111  023516  020124
6599  026574  053117  043122  047514
6600  026602  047440  020116  051127
6601  026610  020124  046506  000124
6602
6603  026616  045522  040502  044440  EM60:   .ASCIZ  /RKBA INCRMNTD WRONG ON WRT FMT/
6604  026624  041516  046522  052116
6605  026632  020104  051127  047117
6606  026640  020107  047117  053440
6607  026646  052122  043040  052115
6608  026654  000000
6609
6610  026655  000122  042513  020122  EM61:   .ASCIZ  /RKER SET,ON WRT OR RD OR FMT/
6611  026662  042523  026124  047117
6612  026670  053440  052122  047440
6613  026676  020122  042122  047440
6614  026704  020122  046506  000124
6615
6616  026712  045522  041104  042440  EM62:   .ASCIZ  /RKDB EROR/
6617  026720  047522  000122
6618
```

```
6619  026724  045522  040504  044440  EM63:   .ASCIZ  /RKDA INCRMNTD WRONG ON RD OR RD FMT/
6620  026732  041516  046522  052116
6621  026740  020104  051127  047117
6622  026746  020107  047117  051040
6623  026754  020104  051117  051040
6624  026762  020104  046506  000124
6625
6626  026770  045522  041527  042040  EM64:   .ASCIZ  /RKWC DIDN'T OVRFLO ON RD OR RD FMT/
6627  026776  042111  023516  020124
6628  027004  053117  043122  047514
6629  027012  047440  020116  042122
6630  027020  047440  020122  042122
6631  027026  043040  052115     000
6632
6633  027033     122  041113  020101  EM65:   .ASCIZ  /RKBA INCRMNTD WRONG ON RD OR RD FMT/
6634  027040  047111  051103  047115
6635  027046  042124  053440  047522
6636  027054  043516  047440  020116
6637  027062  042122  047440  020122
6638  027070  042122  043040  052115
6639  027076     000
6640
6641  027077     111  041516  051117  EM66:   .ASCIZ  /INCORRECT HEADER FROM 'SECTOR'/
6642  027104  042522  052103  044040
6643  027112  040505  042504  020122
6644  027120  051106  046517  023440
6645  027126  042523  052103  051117
6646  027134  000047
6647
6648  027136  040504  040524  042440  EM67:   .ASCIZ  /DATA ERROR/
6649  027144  051122  051117     000
6650
6651  027151     103  052116  046122  EM70:   .ASCIZ  "CNTRL RDY DIDN'T SET ON RD/FMT STARTING FROM <DSK-ADRES>"
6652  027156  051040  054504  042040
6653  027164  042111  023516  020124
6654  027172  042523  020124  047117
6655  027200  051040  027504  046506
6656  027206  020124  052123  051101
6657  027214  044524  043516  043040
6658  027222  047522  020115  042074
6659  027230  045523  040455  051104
6660  027236  051505  000076
6661
6662  027242  042510  047440  020122  EM71:   .ASCIZ  "HE OR ERR ON RD/FMT STARTING FROM <DSK-ADRES>"
6663  027250  051105  020122  047117
6664  027256  051040  027504  046506
6665  027264  020124  052123  051101
6666  027272  044524  043516  043040
6667  027300  047522  020115  042074
6668  027306  045523  040455  051104
6669  027314  051505  000076
6670
6671  027320  051127  047117  020107  EM72:   .ASCIZ  /WRONG DRIVE ID IN RKDS AFTER SEEK/
6672  027326  051104  053111  020105
6673  027334  042111  044440  020116
6674  027342  045522  051504  040440
```

```
  6675  027350  052106  051105  051440
  6676  027356  042505  000113
  6677
  6678  027362  051110  053504  042522  EM73:   .ASCIZ  /HRDWRE POLL-DRV ID BITS(13-15) SHLDBE CLR/
  6679  027370  050040  046117  026514
  6680  027376  051104  020126  042111
  6681  027404  041040  052111  024123
  6682  027412  031461  030455  024465
  6683  027420  051440  046110  041104
  6684  027426  020105  046103  000122
  6685
  6686  027434  051110  053504  042522  EM74:   .ASCIZ  /HRDWRE POLL-INTRUPTING DRIV # NOT PRSNT/
  6687  027442  050040  046117  026514
  6688  027450  047111  051124  050125
  6689  027456  044524  043516  042040
  6690  027464  044522  020126  020043
  6691  027472  047516  020124  051120
  6692  027500  047123  000124
  6693
  6694  027504  051104  053111  021440  EM75:   .ASCIZ  /DRIV # DIDN'T INTRUPT AFTER HRDWRE POLL/
  6695  027512  042040  042111  023516
  6696  027520  020124  047111  051124
  6697  027526  050125  020124  043101
  6698  027534  042524  020122  051110
  6699  027542  053504  042522  050040
  6700  027550  046117  000114
  6701
  6702  027554  041523  020120  044504  EM76:   .ASCIZ  /SCP DIDN'T SET AFTER SEEK WAS DONE/
  6703  027562  047104  052047  051440
  6704  027570  052105  040440  052106
  6705  027576  051105  051440  042505
  6706  027604  020113  040527  020123
  6707  027612  047504  042516     000
  6708
  6709  027617     122  042113  020101  EM77:   .ASCIZ  /RkDA CHANGD AFTER DRIV RESET/
  6710  027624  044103  047101  042107
  6711  027632  040440  052106  051105
  6712  027640  042040  044522  020126
  6713  027646  042522  042523  000124
  6714
  6715  027654  040504  040524  042440  EM100:  .ASCIZ  /DATA EROR AT WORD#/
  6716  027662  047522  020122  052101
  6717  027670  053440  051117  021504
  6718  027676     000
  6719
  6720  027677     103  052116  046122  EM101:  .ASCIZ  /CNTRL RDY DIDN'T SET AFTER RD CHK/
  6721  027704  051040  054504  042040
  6722  027712  042111  023516  020124
  6723  027720  042523  020124  043101
  6724  027726  042524  020122  042122
  6725  027734  041440  045510     000
  6726
  6727  027741     105  051122  047440  EM102:  .ASCIZ  /ERR OR HE ON RD CHK/
  6728  027746  020122  042510  047440
  6729  027754  020116  042122  041440
  6730  027762  045510     000
```

```
6731
6732  027765     103  042523  047440  EM103:  .ASCIZ  /CSE ON RD CHK/
6733  027772  020116  042122  041440
6734  030000  045510     000
6735
6736  030003     122  053513  020103  EM104:  .ASCIZ  /RKWC DIDN'T OVERFLO ON RD CHK OR WRT CHK/
6737  030010  044504  047104  052047
6738  030016  047440  042526  043122
6739  030024  047514  047440  020116
6740  030032  042122  041440  045510
6741  030040  047440  020122  051127
6742  030046  020124  044103  000113
6743
6744  030054  045522  040504  044440  EM105:  .ASCIZ  /RKDA INCRMNTD WRONG ON RD CHK/
6745  030062  041516  046522  052116
6746  030070  020104  051127  047117
6747  030076  020107  047117  051040
6748  030104  020104  044103  000113
6749
6750  030112  045522  040502  041440  EM106:  .ASCIZ  /RKBA CHANGD AFTER RD CHK/
6751  030120  040510  043516  020104
6752  030126  043101  042524  020122
6753  030134  042122  041440  045510
6754  030142     000
6755
6756  030143     115  046505  051117  EM107:  .ASCIZ  /MEMORY WORD CHANGED AFTER RD CHK/
6757  030150  020131  047527  042122
6758  030156  041440  040510  043516
6759  030164  042105  040440  052106
6760  030172  051105  051040  020104
6761  030200  044103  000113
6762
6763  030204  047103  051124  020114  EM110:  .ASCIZ  /CNTRL RDY DIDN'T SET AFTER WRT CHK/
6764  030212  042122  020131  044504
6765  030220  047104  052047  051440
6766  030226  052105  040440  052106
6767  030234  051105  053440  052122
6768  030242  041440  045510     000
6769
6770  030247     110  020105  051117  EM111:  .ASCIZ  /HE OR ERR ON WRT CHK/
6771  030254  042440  051122  047440
6772  030262  020116  051127  020124
6773  030270  044103  000113
6774
6775  030274  051127  052111  020105  EM112:  .ASCIZ  /WRITE CHECK EROR/
6776  030302  044103  041505  020113
6777  030310  051105  051117     000
6778
6779  030315     122  042113  020101  EM113:  .ASCIZ  /RKDA INCRMNTD WRONG ON WRT CHK/
6780  030322  047111  051103  047115
6781  030330  042124  053440  047522
6782  030336  043516  047440  020116
6783  030344  051127  020124  044103
6784  030352  000113
6785
6786  030354  045522  040502  044440  EM114:  .ASCIZ  /RKBA INCRMNTD WRONG ON WRT CHK/
```

```
  6787  030362  041516  046522  052116
  6788  030370  020104  051127  047117
  6789  030376  020107  047117  053440
  6790  030404  052122  041440  045510
  6791  030412     000
  6792
  6793  030413     122  041113  020101  EM115:  .ASCIZ  /RKBA INCRMNTD, WITH IBA SET/
  6794  030420  047111  051103  047115
  6795  030426  042124  020054  044527
  6796  030434  044124  044440  040502
  6797  030442  051440  052105     000
  6798
  6799  030447     127  047522  043516  EM116:  .ASCIZ  /WRONG MEMORY LOCATION CHANGED WITH IBA SET/
  6800  030454  046440  046505  051117
  6801  030462  020131  047514  040503
  6802  030470  044524  047117  041440
  6803  030476  040510  043516  042105
  6804  030504  053440  052111  020110
  6805  030512  041111  020101  042523
  6806  030520  000124
  6807
  6808  030522  045522  030461  042040  EM117:  .ASCIZ  /RK11 DIDN'T INTRUPT WHEN IDE WAS SET/
  6809  030530  042111  023516  020124
  6810  030536  047111  051124  050125
  6811  030544  020124  044127  047105
  6812  030552  044440  042504  053440
  6813  030560  051501  051440  052105
  6814  030566     000
  6815
  6816  030567     122  030513  020061  EM120:  .ASCIZ  /RK11 DIDN'T INTRUPT AFTER SK WAS INITIATED/
  6817  030574  044504  047104  052047
  6818  030602  044440  052116  052522
  6819  030610  052120  040440  052106
  6820  030616  051105  051440  020113
  6821  030624  040527  020123  047111
  6822  030632  052111  040511  042524
  6823  030640  000104
  6824
  6825  030642  041523  020120  042523  EM121:  .ASCIZ  /SCP SET BEFORE SEEK COMPLETED/
  6826  030650  020124  042502  047506
  6827  030656  042522  051440  042505
  6828  030664  020113  047503  050115
  6829  030672  042514  042524  000104
  6830
  6831  030700  045522  030461  042040  EM122:  .ASCIZ  /RK11 DIDN'T INTRUPT AFTER SK COMPLETED/
  6832  030706  042111  023516  020124
  6833  030714  047111  051124  050125
  6834  030722  020124  043101  042524
  6835  030730  020122  045523  041440
  6836  030736  046517  046120  052105
  6837  030744  042105     000
  6838
  6839  030747     103  052116  046122  EM123:  .ASCIZ  /CNTRL RESET DIDN'T CLEAR 'SCP'/
  6840  030754  051040  051505  052105
  6841  030762  042040  042111  023516
  6842  030770  020124  046103  040505
```

```
 6843  030776  020122  051447  050103
 6844  031004  000047
 6845
 6846  031006  045522  030461  042040  EM124:  .ASCIZ  /RK11 DIDN'T INTRUPT AFTER RD DONE/
 6847  031014  042111  023516  020124
 6848  031022  047111  051124  050125
 6849  031030  020124  043101  042524
 6850  031036  020122  042122  042040
 6851  031044  047117  000105
 6852
 6853  031050  047103  051124  020114  EM125:  .ASCIZ  /CNTRL RESET DIDN'T CLR REGISTR/
 6854  031056  042522  042523  020124
 6855  031064  044504  047104  052047
 6856  031072  041440  051114  051040
 6857  031100  043505  051511  051124
 6858  031106     000
 6859
 6860  031107     122  030513  020061  EM126:  .ASCIZ  /RK11 DIDN'T INTRUPT AT CPU LEVEL/
 6861  031114  044504  047104  052047
 6862  031122  044440  052116  052522
 6863  031130  052120  040440  020124
 6864  031136  050103  020125  042514
 6865  031144  042526  000114
 6866
 6867  031150  045522  030461  044440  EM127:  .ASCIZ  /RK11 INTRUPTED AT WRONG CPU LEVEL/
 6868  031156  052116  052522  052120
 6869  031164  042105  040440  020124
 6870  031172  051127  047117  020107
 6871  031200  050103  020125  042514
 6872  031206  042526  000114
 6873
 6874  031212  042447  051122  041040  EM130:  .ASCIZ  /'ERR BIT' DIDN'T SET IN RKER/
 6875  031220  052111  020047  044504
 6876  031226  047104  052047  051440
 6877  031234  052105  044440  020116
 6878  031242  045522  051105     000
 6879
 6880  031247     110  020105  051117  EM131.  .ASCIZ  /HE OR ERR DIDN'T SET/
 6881  031254  042440  051122  042040
 6882  031262  042111  023516  020124
 6883  031270  042523  000124
 6884
 6885  031274  045522  051105  042440  EM132:  .ASCIZ  /RKER EROR/
 6886  031302  047522  000122
 6887
 6888  031306  054116  020103  044502  EM133:  .ASCIZ  /NXC BIT DIDN'T SET/
 6889  031314  020124  044[illegible]  047104
 6890  031322  052047  051 [illegible]  052105
 6891  031330     000
 6892
 6893  031331     122  030513  020061  EM134:  .ASCIZ  /RK11 DIDN'T INTRUPT ON SOFT EROR/
 6894  031336  044504  047104  052047
 6895  031344  044440  052116  052522
 6896  031352  052120  047440  020116
 6897  031360  047523  052106  042440
 6898  031366  047522  000122
```

```
 6899
 6900  031372  042515  020130  044502  EM135:  .ASCIZ  /MEX BITS INCRMNTD WRONG-RKCS/
 6901  031400  051524  044440  041516
 6902  031406  046522  052116  020104
 6903  031414  051127  047117  026507
 6904  031422  045522  051503     000
 6905
 6906  031427     127  051520  047040  EM137:  .ASCIZ  /WPS NOT CLEAR/
 6907  031434  052117  041440  042514
 6908  031442  051101     000
 6909
 6910  031445     104  052101  020101  EM140:  .ASCIZ  /DATA EROR ON TRANSFER FROM DISK TO TTY/
 6911  031452  051105  051117  047440
 6912  031460  020116  051124  047101
 6913  031466  043123  051105  043040
 6914  031474  047522  020115  044504
 6915  031502  045523  052040  020117
 6916  031510  052124  000131
 6917
 6918  031514  042047  044522  020126  EM141:  .ASCIZ  /'DRIV #' PRESENT, BUT NOT INDICATED/
 6919  031522  023443  050040  042522
 6920  031530  042523  052116  020054
 6921  031536  052502  020124  047516
 6922  031544  020124  047111  044504
 6923  031552  040503  042524  000104
 6924  031560  047040  020117  052502  EM142:  .ASCIZ  / NO BUSY ON OTHER HALF OF RK-05F/
 6925  031566  054523  047440  020116
 6926  031574  052117  042510  020122
 6927  031602  040510  043114  047440
 6928  031610  020106  045522  030055
 6929  031616  043065     000
 6930
 6931
 6932
 6933
 6934
 6935          031622                          .EVEN
 6936
 6937                                          .SBTTL  ERROR DATA POINTERS
 6938
 6939  031622  001116  001162  000000  DT1:    .WORD   $ERRPC,$REG0,0
 6940
 6941  031630  001116  001162  001164  DT2:    .WORD   $ERRPC,$REG0,$REG1,0
 6942  031636  000000
 6943
 6944  031640  001116  001162  001164  DT20:   .WORD   $ERRPC,$REG0,$REG1,$REG2,$REG3,0
 6945  031646  001166  001170  000000
 6946
 6947  031654  001116  000000          DT21:   .WORD   $ERRPC,0
 6948
 6949  031660  001116  001162  001164  DT26:   .WORD   $ERRPC,$REG0,$REG1,$REG2,0
 6950  031666  001166  000000
 6951
 6952  031672  001116  001162  001164  DT54:   .WORD   $ERRPC,$REG0,$REG1,$REG2,$REG3,$REG4,$REG5,$REG6,$REG7,0
 6953  031700  001166  001170  001172
 6954  031706  001174  001176  001200
```

```
 6955  031714  000000
 6956
 6957
 6958
 6959
 6960                                          .SBTTL  ERROR HEADERS
 6961
 6962  031716  020040  041520  020040  DH2:    .ASCIZ  / PC   REGADD    RECVD/
 6963  031724  051040  043505  042101
 6964  031732  020104  020040  051040
 6965  031740  041505  042126     000
 6966
 6967  031745     040  050040  020103  DH4:    .ASCIZ  / PC    EXPCT   RECVD/
 6968  031752  020040  042440  050130
 6969  031760  052103  020040  051040
 6970  031766  041505  042126     000
 6971
 6972  031773     040  050040  020103  DH5:    .ASCIZ  / PC    RECVD/
 6973  032000  020040  051040  041505
 6974  032006  042126     000
 6975
 6976  032011     040  050040  020103  DH14:   .ASCIZ  / PC    RKCS    RKER    RKWC/
 6977  032016  020040  051040  041513
 6978  032024  020123  020040  051040
 6979  032032  042513  020122  020040
 6980  032040  051040  053513  000103
 6981
 6982  032046  020040  041520     000  DH21:   .ASCIZ  / PC/
 6983
 6984  032053     040  050040  020103  DH30:   .ASCIZ  / PC     RKCS    RKER    RKDS/
 6985  032060  020040  020040  045522
 6986  032066  051503  020040  020040
 6987  032074  045522  051105  020040
 6988  032102  020040  045522  051504
 6989  032110     000
 6990
 6991  032111     040  050040  020103  DH34:   .ASCIZ  / PC     RKDS/
 6992  032116  020040  020040  045522
 6993  032124  051504     000
 6994
 6995  032127     040  050040  020103  DH35:   .ASCIZ  / PC   SEC-CNTR/
 6996  032134  020040  042523  026503
 6997  032142  047103  051124     000
 6998
 6999  032147     040  050040  020103  DH36:   .ASCIZ  / PC     PRSNT  NXT-CNT/
 7000  032154  020040  020040  051120
 7001  032162  047123  020124  047040
 7002  032170  052130  041455  052116
 7003  032176     000
 7004
 7005  032177     040  050040  020103  DH40:   .ASCIZ  / PC    SECTOR   RKDS/
 7006  032204  020040  051440  041505
 7007  032212  047524  020122  020040
 7008  032220  045522  051504     000
 7009
 7010  032225     040  050040  020103  DH44:   .ASCIZ  / PC    RKCS    RKER    RKDS    RKDA/
```

```
7011  032232  020040  051040  041513
7012  032240  020123  020040  051040
7013  032246  042513  020122  020040
7014  032254  051040  042113  020123
7015  032262  020040  051040  042113
7016  032270  000101
7017
7018  032272  020040  041520  020040  DH54:   .ASCIZ  / PC   RKCS   RKER   RKDS   RKDA   DRV#....CYL.<DSK-ADRS>.SUR..SEC/
7019  032300  020040  045522  051503
7020  032306  020040  020040  045522
7021  032314  051105  020040  020040
7022  032322  045522  051504  020040
7023  032330  020040  045522  040504
7024  032336  020040  042040  053122
7025  032344  027043  027056  041456
7026  032352  046131  036056  051504
7027  032360  026513  042101  051522
7028  032366  027076  052523  027122
7029  032374  051456  041505     000
7030
7031  032401     040  041520  020040  DH56:   .ASCIZ  / PC  EXPC: DR#   CYL    SUR    SEC  RECV: DR#   CYL    SUR    SEC/
7032  032406  054105  041520  020072
7033  032414  051104  020043  020040
7034  032422  054503  020114  020040
7035  032430  020040  052523  020122
7036  032436  020040  051440  041505
7037  032444  020040  042522  053103
7038  032452  020072  051104  020043
7039  032460  020040  054503  020114
7040  032466  020040  020040  052523
7041  032474  020122  020040  020040
7042  032502  042523  000103
7043
7044  032506  020040  041520  020040  DH64:   .ASCIZ  / PC   RKWC   RKDA/
7045  032514  020040  045522  041527
7046  032522  020040  051040  042113
7047  032530  000101
7048
7049  032532  020040  041520  020040  DH66:   .ASCIZ    PC   SECTR   EXPCT   RECVD/
7050  032540  020040  042523  052103
7051  032546  020122  020040  054105
7052  032554  041520  020124  020040
7053  032562  042522  053103  000104
7054
7055  032570  020040  041520  020040  DH67:   .ASCIZ  / PC   EXPCT   RECVD  DSK-ADRS/
7056  032576  020040  054105  041520
7057  032604  020124  020040  042522
7058  032612  053103  020104  042040
7059  032620  045523  040455  051104
7060  032626  000123
7061
7062  032630  020040  041520  020040  DH74:   .ASCIZ  / PC      DRIV #/
7063  032636  020040  020040  051104
7064  032644  053111  021440     000
7065
7066  032651     040  050040  020103  DH100:  .ASCIZ  / PC  WORD #   EXPCT   RECVD/
```

```
7067  032656  053440  051117  020104
7068  032664  020043  020040  054105
7069  032672  041520  020124  020040
7070  032700  042522  053103  000104
7071
7072  032706  020040  041520  020040  DH103:  .ASCIZ  /  PC   RKER/
7073  032714  051040  042513  000122
7074
7075  032722  020040  041520  020040  DH104:  .ASCIZ  /  PC   RECVD   RKCS/
7076  032730  051040  041505  042126
7077  032736  020040  051040  041513
7078  032744  000123
7079
7080  032746  020040  041520  020040  DH107:  .ASCIZ  /  PC     LOC    EXPCT   RECVD/
7081  032754  020040  046040  041517
7082  032762  020040  020040  054105
7083  032770  041520  020124  020040
7084  032776  042522  053103  000104
7085
7086  033004  020040  041520  020040  DH117:  .ASCIZ  /  PC   RKCS/
7087  033012  051040  041513  000123
7088
7089  033020  020040  041520  020040  DH126:  .ASCIZ  /  PC    LEVEL    RKCS/
7090  033026  020040  042514  042526
7091  033034  020114  020040  051040
7092  033042  041513  000123
7093
7094  033046  020040  041520  020040  DH130:  .ASCIZ  /  PC     RKCS    RKER  ERR BIT/
7095  033054  020040  051040  041513
7096  033062  020123  020040  051040
7097  033070  042513  020122  042440
7098  033076  051122  041040  052111
7099  033104     000
7100
7101  033105     040  050040  020103  DH131:  .ASCIZ  /  PC     RKCS    RKER/
7102  033112  020040  020040  045522
7103  033120  051503  020040  020040
7104  033126  045522  051105     000
7105
7106  033133     040  050040  020103  DH133:  .ASCIZ  /  PC     RKCS    RKER    RKDA/
7107  033140  020040  020040  045522
7108  033146  051503  020040  020040
7109  033154  045522  051105  020040
7110  033162  020040  045522  040504
7111  033170     000
7112
7113  033171     040  050040  020103  DH140:  .ASCIZ  /  PC    EXPCT   RECVD    RKBA    RKCS/
7114  033176  020040  042440  050130
7115  033204  052103  020040  051040
7116  033212  041505  042126  020040
7117  033220  020040  045522  040502
7118  033226  020040  020040  045522
7119  033234  051503     000
7120
7121
7122        033240                        .EVEN
```

```
MAINDEC-11-DZRKK-D     MACY11 27(1006)  04-OCT-76  16:06  PAGE 130
DZRKKD.P11    22-SEP-76 08:47           ERROR HEADERS                                    SEQ 0147

                                        ;DATA BUFFER
                                        
033240  000400                          OLTBUF: .BLKW   256.     ;THIS 256 WORD BUFFER IS FOR
                                                                 ;DATA TRANSFERS FROM AND
                                                                 ;TO THE DISK.
                                        
                000001                          .END
```

```
BACINT  004526    CR     = 000015    DT20    031640    EM40    025605    PFSTRT  004630
BADTMO  004462    CRETRN  021336     DT21    031654    EM41    025645    PHYDRV  001436
BDAR    021002    CRLF   = 000200    DT26    031660    EM43    025702    PIRQ   = 177772
BDAR0   020764    DDISP  = 177570    DT54    031672    EM44    025734    PIRQVE= 000240
BDAR4   020774    DDPCH   001410     EF.GI   001370    EM45    026010    PR0    = 000000
BIT0   = 000001   DELAY  = 104417     EMTVEC= 000030    EM46    026056    PR1    = 000040
BIT00  = 000001   DELAY   021572     EM100   027654    EM47    026114    PR2    = 000100
BIT01  = 000002   DH100   032657     EM101   027677    EM50    026156    PR3    = 000140
BIT02  = 000004   DH103   032706     EM102   027741    EM51    026205    PR4    = 000200
BIT03  = 000010   DH104   032722     EM103   027765    EM52    026260    PR5    = 000240
BIT04  = 000020   DH107   032746     EM104   030003    EM53    026305    PR6    = 000300
BIT05  = 000040   DH117   033004     EM105   030054    EM54    026350    PR7    = 000340
BIT06  = 000100   DH126   033020     EM106   030112    EM55    026442    PS     = 177776
BIT07  = 000200   DH130   033046     EM107   030143    EM56    026521    PSW    = 177776
BIT08  = 000400   DH131   033105     EM11    025266    EM57    026560    PWRVEC= 000024
BIT09  = 001000   DH133   033133     EM110   030204    EM60    026616    RDCHR  = 104410
BIT1   = 000002   DH14    032011     EM111   030247    EM61    026655    RDLIN  = 104411
BIT10  = 002000   DH140   033171     EM112   030274    EM62    026712    RESVEC= 000010
BIT11  = 004000   DH2     031716     EM113   030315    EM63    026724    RKBP    001336
BIT12  = 010000   DH21    032046     EM114   030354    EM64    026770    RKCS    001332
BIT13  = 020000   DH30    032053     EM115   030413    EM65    027033    RKDA    001340
BIT14  = 040000   DH34    032111     EM116   030447    EM66    027077    RKDB    001342
BIT15  = 100000   DH35    032127     EM117   030522    EM67    027136    RKDS    001326
BIT2   = 000004   DH36    032147     EM12    025300    EM70    027151    RKER    001330
BIT3   = 000010   DH4     031745     EM120   030567    EM71    027242    RKPRI   001400
BIT4   = 000020   DH40    032177     EM121   030642    EM72    027320    RKVEC   001402
BIT5   = 000040   DH44    032225     EM122   030700    EM73    027362    RKWC    001334
BIT6   = 000100   DH5     031773     EM123   030747    EM74    027434    R6     =%000006
BIT7   = 000200   DH54    032272     EM124   031006    EM75    027504    R7     =%000007
BIT8   = 000400   DH56    032401     EM125   031050    EM76    027554    SEEK0   001372
BIT9   = 001000   DH64    032506     EM126   031107    EM77    027617    SEEK1   001374
BPTVEC= 000014    DH66    032532     EM127   031150    ERRVEC= 000004    SEEK2   001376
BRKDA0= 104415    DH67    032570     EM13    025313    FCHECK  025056    SHFTRT  021106
BRKDA4= 104416    DH74    032630     EM130   031212    FFLAG   001404    SIMUL   001344
BTEOP   017734    DISPLA  001142     EM131   031247    FTITLE  001346    SIZEF   025202
CHE1    021150    DISPRE  000174     EM132   031274    GTSWR  = 104406    SIZYET  001440
CHKCCL  021310    DRESET  021412     EM133   031306    GT2RG   020716    STACK  = 001100
CHKCRD= 104412    DRHOLD  025200     EM134   031331    GT3RG   020710    START   002536
CHKDA   021170    DRIVAD  001350     EM135   031372    GT4RG   020702    START1  003262
CHKDA1  021176    DRIVS   001412     EM137   031427    HT     = 000011    STKLMT= 177774
CHKECL  021264    DRIV0   001414     EM140   031445    INDX1   001356    ST2     003664
CHKER   021250    DRIV1   001416     EM141   031514    INDX2   001360    ST3     004210
CHKHE   021142    DRIV2   001420     EM142   031560    IOTVEC= 000020    ST4     004404
CHKHE1  021134    DRIV3   001422     EM16    025325    LF     = 000012    SWR     001140
CHKWC   021224    DRIV4   001424     EM21    025354    MSG1    001216    SWREG   000176
CH.CRD  021666    DRIV5   001426     EM30    025366    MSG2    001236    SW0    = 000001
CKSWR  = 104407   DRIV6   001430     EM31    025376    MSG3    001245    SW00   = 000001
CNT.RD= 104414    DRIV7   001432     EM32    025406    MSG4    001272    SW01   = 000002
CNT.RE= 104413    DRVDON  001352     EM33    025427    MSG5    001303    SW02   = 000004
CN.RDY  021740    DRVPTR  001354     EM34    025447    MSG6    001315    SW03   = 000010
CN.RST  021722    DSWR   = 177570     EM35    025466    NUDRV   004766    SW04   = 000020
COUNT   001362    DT1     031622     EM36    025521    ODDEVN  001406    SW05   = 000040
COUNT1  001364    DT2     031630     EM37    025551    OUTBUF  033240    SW06   = 000100
```

```
SW07  = 000200
SW08  = 000400
SW09  = 001000
SW1   = 000002
SW10  = 002000
SW11  = 004000
SW12  = 010000
SW13  = 020000
SW14  = 040000
SW15  = 100000
SW2   = 000004
SW3   = 000010
SW4   = 000020
SW5   = 000040
SW6   = 000100
SW7   = 000200
SW8   = 000400
SW9   = 001000
TBITVE= 000014
TIMER   001366
TKVEC = 000060
TPVEC = 000064
TRAPVE= 000034
TRTVEC= 000014
TSTEND  020534
TSTRWS  021344
TSTSIN  021500
TST.SI= 104421
TST1    004634
TST10   005670
TST11   005756
TST12   006032
TST13   006160
TST14   006324
TST15   006416
TST16   006654
TST17   007102
TST2    004766
TST20   007412
TST21   007610
TST22   010062
TST23   010432
TST24   010712
TST25   011322
TST26   011630
TST27   012150
TST3    005164
TST30   012436
TST31   012672
TST32   013076
TST33   013344
TST34   013626
TST35   014104
TST36   014236
TST37   014562
TST4    005276
TST40   014776
TST41   015202
TST42   015364
TST43   015472
TST44   015646
TST45   016004
TST46   016212
TST47   016364
TST5    005364
TST50   016542
TST51   016670
TST52   017042
TST53   017236
TST54   017530
TST55   017666
TST56   017712
TST57   017762
TST6    005412
TST7    005450
TYERM   020734
TYPDS = 104405
TYPE  = 104401
TYPOC = 104402
TYPON = 104404
TYPOS = 104403
T56     020046
T56FLG  001434
WATIME  021650
WATINT  021614
WAT.IN= 104420
$AUTOB  001134
$BDADR  001122
$BDDAT  001126
$CHARC  023202
$CKSWR  023660
$CMTAG  001100
$CM1  = 000012
$CM2  = 000024
$CM3  = 000012
$CNTLG  024545
$CNTLU  024540
$CRLF   001213
$DBLK   023422
$DOAGN  020656
$DTBL   023412
$ENDAD  020646
$ENDCT  020614
$ENDMG  020665
$ENULL  020662
$EOP    020560
$EOPCT  020606
$ERFLG  001103
$ERMAX  001115
$ERROR  022320
$ERRPC  001116
$ERRTB  001442
$ERRTY  022632
$ERTTL  001112
$ESCAP  001210
$FILLC  001156
$FILLS  001155
$GDADR  001120
$GDDAT  001124
$GET42  020636
$GTSWR  023730
$HD   = 000000
$ICNT   001104
$ILLUP  025040
$INTAG  001135
$ITEMB  001114
$LF     001214
$LPADR  001106
$LPERR  001110
$MNEW   024563
$MSWR   024552
$MXCNT  022316
$NULL   001154
$NWTST= 000001
$OCNT   023654
$OMODE  023656
$OVER   022302
$PASS   001100
$POWER  025046
$PWRAD  025034
$PWRDN  024674
$PWRMG  025030
$PWRUP  024746
$QUES   001212
$RDCHR  024142
$RDLIN  024262
$RDSZ = 000022
$REGAD  001160
$REG0   001162
$REG1   001164
$REG10  001202
$REG11  001204
$REG2   001166
$REG3   001170
$REG4   001172
$REG5   001174
$REG6   001176
$REG7   001200
$RTNAD  020660
$SAVR6  025044
$SCOPE  022046
$SETUP= 000117
$STUP = 177777
$SVLAD  022254
$SVPC = 000204
$SWR  = 165400
$SWRMK= 000000
$TIMES  001206
$TKB    001146
$TKS    001144
$TN   = 000060
$TPB    001152
$TPFLG  001157
$TPS    001150
$TRAP   024574
$TRAP2  024616
$TRP  = 000022
$TRPAD  024630
$TSTNM  001102
$TTYIN  024516
$TYPDS  023206
$TYPE   022766
$TYPEC  023136
$TYPEX  023204
$TYPOC  023456
$TYPON  023472
$TYPOS  023432
$XTSTR  022060
$$GET4= 000000
$OFILL  023655
.     = 034240

. ABS.  034240     000

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DZRKKD,DZRKKD/LI:ME/NL:MC:MD:CND/SOL/NSG←DZRKKD.P11
 RUN-TIME: 65 62 1 SECONDS
 RUN-TIME RATIO: 505/130=3.8
 CORE USED:  24K  (47 PAGES)
```

Spooler runtime 26 Seconds, 108 KCS, 666 disk reads, 3 disk writes, 149 pages