```
.REM %
```

IDENTIFICATION

PRODUCT CODE:	AC-9244F-MC
PRODUCT NAME:	CZRKKF0 RK11 BASIC LOGIC TEST 2
DATE CREATED:	JUNE 1978
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	JIM KAPADIA
REVISED BY:	PERVEZ ZAKI TOM SAWYER CHUCK HESS

QUICK LOOK-UP OPERATING INSTRUCTIONS

FOR A QUICK REFERENCE, LOOK UP THE FOLLOWING SECTIONS:

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

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

TABLE OF CONTENTS

1.0 ABSTRACT

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

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

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

2.0 REQUIREMENTS

2.1 EQUIPMENT

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

2.2 PRELIMINARY PROGRAMS

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

2.3 EXECUTION TIME

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

3.0 STARTING ADDRESS

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

4.0 PROGRAM CONTROL MODES & OPERATOR ACTION

PAPER TAPE LOADING
RKDP DUMP MODE
RKDP CHAIN MODE
ACT11

4.1 PAPER TAPE LOADING

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

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

4.1.3 LOAD ADDRESS 200

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

PRESS START.

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

DRIVES TO BE TESTED?

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

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

THE DRIVES DO NOT HAVE TO BE IN LOGICAL ORDER.

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

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

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

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

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

"^U" DELETES THE ENTIRE LINE

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

THE CORRECT ANSWER CAN NOW BE RETYPED AGAIN.

4.1.8 THE DRIVE NUMBER BEING TESTED OUT IS PRINTED:

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

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

END PASS # X

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

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

```
RK11 BASIC LOGIC TEST 2
MAINDEC-11-CZRKKF
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 SW<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 EFROR 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-->TST' ..

        4.  ERROR 2,3; SW 12 SET.

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

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

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

```
TST1:   SCOPE
        :



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

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

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

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

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

9.0 PROGRAM STRUCTURE

THERE ARE THREE DISTINCT PARTS OF THE PROGRAM.

SET-UP PHASE
DRIVE-DEPENDENT CONTROLLER TESTS

9.1 SET-UP PHASE

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

WHICH DRIVES ARE TO BE TESTED, ETC.

9.2 DRIVE DEPENDENT CONTROLLER TESTS

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

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

THE DRIVE NUMBER BEING TESTED IS PRINTED OUT:

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

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

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

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

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

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

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

THUS ONE PASS OF THE PROGRAM INVOLVES DOING

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

10.0 ERROR REPORTING

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

ENTRIES.

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

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

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

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

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

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

11.0 ERROR INTERPRETATION

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

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

12.0 HANDLERS AND COMMON ROUTINES

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

A. AS A SUBROUTINE THROUGH 'JSR' CALL

B. THROUGH A 'TRAP' HANDLER

12.1 TRAP HANDLER

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

12.2 SCOPE HANDLER

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

12.3 ERROR HANDLER

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

12.4 CONTROL RESET ROUTINE

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

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

12.5 CONTROL READY ROUTINE

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

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

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

12.6 DRIVE RESET ROUTINE

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

12.7 TIME DELAY ROUTINE

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

12.8 WAIT FOR INTERRUPT ROUTINE

THIS ROUTINE PROVIDES A VARIABLE TIME LIMIT DURING WHICH RK11 INTERRUPT MAY OCCUR. THE IS
WAT.INT .N N=1 TO 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 RK11 INTERRUPT OCCURS THE PROGRAM TYPES OUT THE PC AT WHICH THE INTERRUPT CAME IN AND THEN HALTS. PRESSING CONTINUE WOULD RESTART THE PROGRAM FROM BEGINING. SW 9- LOOPING CAITY IS PROVIDED AS A TROUBLE SHOOTING AID.

14.0 QUICK VERIFYING MODE

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

%

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


                        ;*************************************************************

                        ;YOU ARE ADVISED TO READ THE DOCUMENT BEFORE USING THIS PROGRAM.

                        ;ON GETTING AN ERROR REFER TO THE LISTINGS AT THE PC POINTED
```

```
                                ;OUT IN THE ERROR MESSAGE. ADJACENT ERROR MESSAGES IF FOLLOWED
                                ;CAREFULLY COULD LEAD TO AN EASY PINPOINTING OF THE FAULT

                                ;*****************************************************************
                                .SBTTL  BASIC DEFINITIONS

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

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

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

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

                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                          SW15=   100000
040000                          SW14=   40000
020000                          SW13=   20000
010000                          SW12=   10000
004000                          SW11=   4000
002000                          SW10=   2000
001000                          SW09=   1000
000400                          SW08=   400
000200                          SW07=   200
000100                          SW06=   100
```



```
000040                          SW05=   40
000020                          SW04=   20
000010                          SW03=   10
000004                          SW02=   4
000002                          SW01=   2
000001                          SW00=   1
                                .EQUIV  SW09,SW9
                                .EQUIV  SW08,SW8
                                .EQUIV  SW07,SW7
                                .EQUIV  SW06,SW6
                                .EQUIV  SW05,SW5
                                .EQUIV  SW04,SW4
                                .EQUIV  SW03,SW3
                                .EQUIV  SW02,SW2
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0

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

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

```
1006            000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
1007            000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
1008                                    .SBTTL  TRAP CATCHER
1009
1010            000000                          .=0
1011                                    ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
1012                                    ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1013                                    ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1014            000174                          .=174
1015  000174    000000                DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
1016  000176    000000                SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
1017                                    .SBTTL  STARTING ADDRESS(ES)
1018  000200    000137  002636                JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
1019                                    .SBTTL  ACT11 HOOKS
1020
1021                                    ;;*****************************************************************
1022                                    ;HOOKS REQUIRED BY ACT11
1023            000204                          $SVPC=.                 ;SAVE PC
1024            000046                          .=46
1025  000046    020740                        $ENDAD                    ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1026            000052                          .=52
1027  000052    000000                        .WORD   0                 ;;2)SET LOC.52 TO ZERO
1028            000204                        .=$SVPC                   ;; RESTORE PC
```



```
1029                                    .SBTTL  COMMON TAGS
1030
1031                                    ;;*****************************************************************
1032                                    ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
1033                                    ;*USED IN THE PROGRAM.
1034
1035            001100                          .=1100
1036  001100                            $CMTAG:
1037  001100    000000                  $PASS:  .WORD   0               ;;START OF COMMON TAGS
1038  001102       000                  $TSTNM: .BYTE   0               ;;CONTAINS PASS COUNT
1039  001103       000                  $ERFLG: .BYTE   0               ;;CONTAINS THE TEST NUMBER
1040  001104    000000                  $ICNT:  .WORD   0               ;;CONTAINS ERROR FLAG
1041  001106    000000                  $LPADR: .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
1042  001110    000000                  $LPERR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
1043  001112    000000                  $ERTTL: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
1044  001114       000                  $ITEMB: .BYTE   0               ;;CONTAINS TOTAL ERRORS DETECTED
1045  001115       001                  $ERMAX: .BYTE   1               ;;CONTAINS ITEM CONTROL BYTE
1046  001116    000000                  $ERRPC: .WORD   0               ;;CONTAINS MAX. ERRORS PER TEST
1047  001120    000000                  $GDADR: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
1048  001122    000000                  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
1049  001124    000000                  $GDDAT: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
1050  001126    000000                  $BDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
1051  001130    000000                          .WORD   0               ;;CONTAINS 'BAD' DATA
1052  001132    000000                          .WORD   0               ;;RESERVED--NOT TO BE USED
1053  001134       000                  $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
1054  001135       000                  $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
1055  001136    000000                          .WORD   0
1056  001140    177570                  $SWR:   .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
1057  001142    177570                  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
1058  001144    177560                  $TKS:   177560                  ;;TTY KBD STATUS
1059  001146    177562                  $TKB:   177562                  ;;TTY KBD BUFFER
1060  001150    177564                  $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
1061  001152    177566                  $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
1062  001154       000                  $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
1063  001155       002                  $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
1064  001156       012                  $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
1065  001157       000                  $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
1066  001160    000000                  $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                                        ;;WHICH  ($REG0) WAS OBTAINED
1067
1068  001162    000000                  $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
1069  001164    000000                  $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
1070  001166    000000                  $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
1071  001170    000000                  $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
1072  001172    000000                  $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
1073  001174    000000                  $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
1074  001176    000000                  $REG6:  .WORD   0               ;;CONTAINS (($REGAD)+14)
1075  001200    000000                  $REG7:  .WORD   0               ;;CONTAINS (($REGAD)+16)
1076  001202    000000                  $REG10: .WORD   0               ;;CONTAINS (($REGAD)+20)
1077  001204    000000                  $REG11: .WORD   0               ;;CONTAINS (($REGAD)+22)
1078  001206    000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
1079  001210    000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
1080  001212       077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
1081  001213       015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
1082  001214    000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
1083                                    ;;*****************************************************************
1084  001216    005015  051104  053111  MSG1:   .ASCIZ  <15><12>/DRIVE PRESNT/
```

```
1085  001224  020105  051120  051505
1086  001232  052116     000
1087          001236                    .EVEN
1088  001236  005015  047516  042516  MSG2:   .ASCIZ  <15><12>/NONE/
1089  001244     000
1090
1091  001245     015  041412  052116  MSG3:   .ASCIZ  <15><12>/CNT RDY DIDN'T SET/
1092  001252  051040  054504  042040
1093  001260  042111  023516  020124
1094  001266  042523  000124
1095
1096  001272  005015  051104  053111  MSG4:   .ASCIZ  <15><12>/DRIVE /
1097  001300  020105     000
1098
1099  001303     015  040412  046114  MSG5:   .ASCII  <15><12>/ALL DRVS/
1100  001310  042040  053122     123
1101
1102  001315     040  051104  050117  MSG6:   .ASCIZ  / DROPD/<15><12>
1103  001322  006504  000012
1104                                    .EVEN
1105
1106                                    ;RK11 REGISTERS
1107                                    ;IF FOR ANY REASON THE REGISTER ADDRESSES ARE DIFFERENT FROM THESE
1108                                    ;(GIVEN BELOW), THE CONTENTS OF THE APPROPRIATE POINTERS SHOULD BE
1109                                    ;MODIFIED SO THAT THE CORRECT ADDRESS IS USED.
1110                                    ;
1111                                    .EVEN
1112  001326  177400                    RKDS:   177400
1113  001330  177402                    RKER:   177402
1114  001332  177404                    RKCS:   177404
1115  001334  177406                    RKWC:   177406
1116  001336  177410                    RKBA:   177410
1117  001340  177412                    RKDA:   177412
1118  001342  177416                    RKDB:   177416
1119
1120
1121                                    ;TAGS AND GENERAL DATA AREA
1122                                    ;
1123                                    ;
1124
1125  001344  000000                    SIMUL:  0               ;FLAG TO BE SET TO 1 WHEN ON SIMULATOR
1126  001346  000000                    FTITLE: 0               ;FLAG FOR PRINTING PROGRAM TITLE
1127  001350  000000                    DRIVAD: 0               ;CONTAINS ADDRESS OF THE DRIVE UNDER TEST
1128  001352  000000                    DRVDON: 0               ;CONTAINS THE NUMBER OF DRIVES CHECKED.
1129                                                            ;IT IS INCREMENTED EACH TIME THE TESTS FOR
1130                                                            ;A DRIVE IS COMPLETED.
1131  001354  000000                    DRVPTR: 0               ;CONTAINS THE POINTER TO THE DRIVE FLAG (DRIVE0
1132                                                            ;-DRIVE7) OF THE DRIVE TO BE CHECKED NEXT.
1133  001356  000000                    INDX1:  0               ;GENERAL INDEX FOR KEEPING COUNT
1134  001360  000000                    INDX2:  0               ; GENERAL INDEX
1135  001362  000000                    COUNT:  0               ;GENERAL COUNT REGISTER
1136  001364  000000                    COUNT1: 0               ;COUNT REGISTER USED FOR 'DRESET' SUBROUTINE
1137  001366  000000                    TIMER:  0               ;TIMER REGISTER
1138  001370  000000                    EFLG1:  0               ;SET, TO INDICATE A PARTICULAR
1139                                                            ;ERROR CONDITION
1140
```



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

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS 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  025402                          EM12    ;'SIN' IS SET
001444  032327                          DH44    ;PC     RKCS    RKER    RKDS    RKDA
001446  031742                          DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
001450  000000                          0

                                ;ITEM   2

001452  027253                          EM70    ;CNTRL RDY DIDN'T SET ON READ/FMT FROM DISK ADDRESS
001454  032113                          DH14    ;PC     RKCS    RKER    RKWC
001456  031762                          DT26    ;$ERRPC $REG0   $REG1   $REG2
001460  000000                          0

                                ;ITEM   3

001462  025427                          EM16    ;RKDA WRONG AFTER SSE
001464  032047                          DH4     ;PC     EXPCT   RECVD
001466  031732                          DT2     ;$ERRPC $REG0   $REG1
001470  000000                          0

                                ;ITEM   4

001472  025456                          EM21    ;RKDS ERROR
001474  032213                          DH34    ;PC     RKDS
001476  031724                          DT1     ;$ERRPC $REG0
001500  000000                          0

                                ;ITEM   5

001502  025470                          EM30    ;'DPL' BIT SET, CHECK DRIVE POWER
001504  032155                          DH30    ;PC     RKCS    RKER    RKDS
001506  031762                          DT26    ;$ERRPC $REG0   $REG1   $REG2
001510  000000                          0

                                ;ITEM   6

001512  025500                          EM31    ;'DRU' BIT SET, CHECK DRIVE
001514  032155                          DH30    ;PC     RKCS    RKER    RKDS
001516  031762                          DT26    ;$ERRPC $REG0   $REG1   $REG2
001520  000000                          0

                                ;ITEM   7

001522  025510                          EM32    ;'RK05' BIT NOT SET
001524  032213                          DH34    ;PC     RKDS
001526  031724                          DT1     ;$ERRPC $REG0
001530  000000                          0

                                ;ITEM   10

001532  025531                          EM33    ;'DRY' NOT SET
001534  032327                          DH44    ;PC     RKCS    RKER    RKDS    RKDA
001536  031742                          DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
001540  000000                          0
```

```
1294                                    ;ITEM   11
1295
1296  001542  025551                            EM34    ;'SOK' DID NOT SET
1297  001544  032213                            DH34    ;PC     RKDS
1298  001546  031724                            DT1     ;$ERRPC $REG0
1299  001550  000000                            0
1300
1301                                    ;ITEM   12
1302
1303  001552  025570                            EM35    ;'SEC COUNTR' DIDN'T COUNT TO 0
1304  001554  032231                            DH35    ;PC     SEC-CNTR
1305  001556  031724                            DT1     ;$ERRPC $REG0
1306  001560  000000                            0
1307
1308                                    ;ITEM   13
1309
1310  001562  025623                            EM36    ;'SEC COUNTR' DIDN'T INCREMENT
1311  001564  032251                            DH36    ;PC     PRSNT-COUNT     NXT-COUNT
1312  001566  031732                            DT2     ;$ERRPC $REG0           $REG1
1313  001570  000000                            0
1314
1315                                    ;ITEM   14
1316
1317  001572  025653                            EM37    ;'SECTOR COUNTER' INCREMENTED WRONG
1318  001574  032047                            DH4     ;PC     EXPCTD  RECVD
1319  001576  031732                            DT2     ;$ERRPC $REG0   $REG1
1320  001600  000000                            0
1321
1322                                    ;ITEM   15
1323
1324  001602  025707                            EM40    ;DIDN'T GET SC=SA FOR THIS SECTOR
1325  001604  032301                            DH40    ;PC     SECTOR  RKDS
1326  001606  031732                            DT2     ;$ERRPC $REG0   $REG1
1327  001610  000000                            0
1328
1329                                    ;ITEM   16
1330
1331  001612  025747                            EM41    ;ERROR-'R/W/S RDY' SHOULD BE SET
1332  001614  032213                            DH34    ;PC     RKDS
1333  001616  031724                            DT1     ;$ERRPC $REG0
1334  001620  000000                            0
1335
1336                                    ;ITEM   17
1337
1338  001622  025415                            EM13    ;RKBA ERROR
1339  001624  032047                            DH4     ;PC     EXPCT   RECVD
1340  001626  031732                            DT2     ;$ERRPC $REG0   $REG1
1341  001630  000000                            0
1342
1343                                    ;ITEM   20
1344
1345  001632  026004                            EM43    ;UNEXPECTED RK11 INTERRUPT
1346  001634  032150                            DH21    ;PC
1347  001636  031756                            DT21    ;$ERRPC
1348  001640  000000                            0
1349
```



```
1350                                    ;ITEM   21
1351
1352  001642  026036                            EM44    ;'CNTRL RDY' DIDN'T SET AFTER SEEK OR DRIVE RESET
1353  001644  032327                            DH44    ;PC     RKCS    RKER    RKDS    RKDA
1354  001646  031742                            DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3.
1355  001650  000000                            0
1356
1357                                    ;ITEM   22
1358
1359  001652  026112                            EM45    ;'ERR' OR 'HE' SET ON SEEK OR DRIVE RESET
1360  001654  032327                            DH44    ;PC     RKCS    RKER    RKDS    RKDA
1361  001656  031742                            DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
1362  001660  000000                            0
1363
1364                                    ;ITEM   23
1365
1366  001662  026160                            EM46    ;RKER BIT, ON SEEK OR DRIVE RESET
1367  001664  032155                            DH30    ;PC     RKCS    RKER    RKDS
1368  001666  031762                            DT26    ;$ERRPC $REG0   $REG1   $REG2
1369  001670  000000                            0
1370
1371                                    ;ITEM   24
1372
1373  001672  026216                            EM47    ;RKCS CHANGED AFTER FUNCTION WAS DONE
1374  001674  032047                            DH4     ;PC     EXPCT   RECVD
1375  001676  031732                            DT2     ;$ERRPC $REG0   $REG1
1376  001700  000000                            0
1377
1378                                    ;ITEM   25
1379
1380  001702  026260                            EM50    ;'R/W/S RDY' DID NOT CLEAR
1381  001704  032155                            DH30    ;PC     RKCS    RKER    RKDS
1382  001706  031762                            DT26    ;$ERRPC $REG0   $REG1   $REG2
1383  001710  000000                            0
1384
1385                                    ;ITEM   26
1386
1387  001712  026307                            EM51    ;'R/W/S RDY' DIDN'T SET AFTER SEEK OR DRIVE RESET
1388  001714  032327                            DH44    ;PC     RKCS    RKER    RKDS    RKDA
1389  001716  031742                            DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
1390  001720  000000                            0
1391
1392                                    ;ITEM   27
1393
1394  001722  026362                            EM52    ;RKDA CHANGED AFTER SEEK
1395  001724  032047                            DH4     ;PC     EXPCTD  REGVD
1396  001726  031732                            DT2     ;$ERRPC $REG0   $REG1
1397  001730  000000                            0
1398
1399                                    ;ITEM   30
1400
1401  001732  026407                            EM53    ;'CNTRL RDY' DIDN'T CLEAR AS GO WAS SET
1402  001734  032155                            DH30    ;PC     RKCS    RKER    RKDS
1403  001736  031762                            DT26    ;$ERRPC $REG0   $REG1   $REG2
1404  001740  000000                            0
1405
```

```
                                ;ITEM   31
                                        EM54    ;'CNTRL RDY' DIDN'T SET ON DOING WRITE/FMT STARTING
                                                ;FROM <DSK-ADRES>
                                        DH54    ;PC     RKCS    RKER    RKDS    RKDA
                                                ;DRV#   CYL  <DSK-ADRES>        SUR     SECTR
                                        DT54    ;$ERRPC $REG0   $REG1   $REG2   $REG3
                                                ;$REG4  $REG5   $REG6   $REG>
                                        0
                                ;ITEM   32
                                        EM55    ;'HE' OR 'ERR'  ON WRITE/FMT STARTING FROM
                                                ;<DSK-ADRES>
                                        DH54    ;PC     RKCS    RKER    RKDS    RKDA
                                                ;DRV#   CYL  <DSK-ADRES>        SUR     SECTR
                                        DT54    ;$ERRPC $REG0   $REG1   $REG2   $REG3
                                                ;$REG4  $REG5   $REG6   $REG7
                                        0
                                ;ITEM   33
                                        EM56    ;RKDA INCREMENTED WRONG ON WRITE OR WRITE FORMAT
                                        DH56    ;PC     EXPCT:  DRV#    CYL     SUR     SECTR
                                                ;RECVD: DRV#    CYL     SUR     SECTR
                                        DT54    ;$ERRPC         $REG0   $REG1   $REG2   $REG3
                                                ;$REG4  $REG5   $REG6   $REG7
                                        0
                                ;ITEM   34
                                        EM57    ;RKWC DIDN'T OVERFLOW ON WRITE OR WRITE FORMAT
                                        DH5     ;PC     RECVD
                                        DT1     ;$ERRPC $REG0
                                        0
                                ;ITEM   35
                                        EM60    ;RKBA INCREMENTED WRONG ON WRITE OR WRITE FORMAT
                                        DH4     ;PC     EXPCT   RECVD
                                        DT2     ;$ERRPC $REG0   $REG1
                                        0
                                ;ITEM   36
                                        EM61    ;RKER SET, ON WRITE/READ/FORMAT
                                        DH30    ;PC     RKCS    RKER    RKDS
                                        DT26    ;$ERRPC $REG0   $REG1   $REG2
                                        0
                                ;ITEM   37
                                        EM62    ;RKDB ERROR
                                        DH4     ;PC     EXPCT   RECVD
                                        DT2     ;$ERRPC $REG0   $REG1
                                        0
```



```
                                ;ITEM   40
                                        EM63    ;RKDA INCREMENTED WRONG ON READ OR READ FORMAT
                                        DH56    ;PC     EXPCT:  DRV#    CYL     SUR     SECTR
                                                ;RECVD: DRV#    CYL     SUR     SECTR
                                        DT54    ;$ERRPC         $REG0   $REG1   $REG2   $REG3
                                                ;$REG4  $REG5   $REG6   $REG7
                                        0
                                ;ITEM   41
                                        EM64    ;RKWC DID NOT OVERFLOW ON READ OR READ FORMAT
                                        DH64    ;PC     RKWC    RKDA
                                        DT2     ;$ERRPC $REG0   $REG1
                                        0
                                ;ITEM   42
                                        EM65    ;RKBA INCREMENTED WRONG ON READ OR READ FORMAT
                                        DH4     ;PC     EXPCT   RECVD
                                        DT2     ;$ERRPC $REG0   $REG1
                                        0
                                ;ITEM   43
                                        EM66    ;INCORRECT HEADER FROM 'SECTOR'
                                        DH66    ;PC     SECTR   EXPCT   RECVD
                                        DT26    ;$ERRPC $REG0   $REG1   $REG2.
                                        0
                                ;ITEM   44
                                        EM67    ;DATA ERROR
                                        DH67    ;PC     EXPCT   RECVD   DSK-ADRES
                                        DT26    ;$ERRPC $REG0   $REG1   $REG2
                                        0
                                ;ITEM   45
                                        EM70    ;'CNTRL RDY' DIDN'T SET ON DOING READ/FMT STARTING
                                                ; FROM <DSK-ADRES>
                                        DH54    ;PC     RKCS    RKER    RKDS    RKDA
                                                ;DRV#   CYL     <DSK-ADRES>     SUR     SECTR
                                        DT54    ;$ERRPC $REG0   $REG1   $REG2   $REG3
                                                ;$REG4  $REG5   $REG6   $REG7
                                        0
                                ;ITEM   46
                                        EM71    ;'HE' OR 'ERR' BIT SET ON READ/FMT STARTING
                                                ; FROM <DSK-ADRES>
                                        DH54    ;PC     RKCS    RKER    RKDS    RKDA
                                                ;DRV#   CYL     <DSK-ADRES>     SUR     SECTR
                                        DT54    ;$ERRPC $REG0   $REG1   $REG2   $REG3
                                                ;$REG4  $REG5   $REG6   $REG7
```

```
1518  002120  000000                        0
1519
1520                                  ;ITEM   47
1521
1522  002122  027422                        EM72    ;WRONG DRIVE ID IN RKDS AFTER SEEK
1523  002124  032047                        DH4     ;PC     EXPCT   RECVD
1524  002126  031732                        DT2     ;$ERRPC $REG0   $REG1
1525  002130  000000                        0
1526
1527                                  ;ITEM   50
1528
1529  002132  027464                        EM73    ;HARDWARE POLL, DRIVE ID BITS(13-15) SHOULD BE CLEAR
1530  002134  032213                        DH34    ;PC     RKDS
1531  002136  031732                        DT2     ;$ERRPC $REG0
1532  002140  000000                        0
1533
1534                                  ;ITEM   51
1535
1536  002142  027536                        EM74    ;HARDWARE POLL, INTERRUPTING DRIVE # NOT PRESENT
1537  002144  032732                        DH74    ;PC     DRIVE #
1538  002146  031724                        DT1     ;$ERRPC $REG0
1539  002150  000000                        0
1540
1541                                  ;ITEM   52
1542
1543  002152  027606                        EM75    ;'DRIVE #' DID NOT INTERRUPT DURING HARDWARE POLL
1544  002154  032732                        DH74    ;PC     DRIVE #
1545  002156  031724                        DT1     ;$ERRPC $REG0
1546  002160  000000                        0
1547
1548                                  ;ITEM   53
1549
1550  002162  027656                        EM76    ;SCP DID NOT SET AFTER WAS DONE
1551  002164  033106                        DH117   ;PC     RKCS
1552  002166  031724                        DT1     ;$ERRPC $REG0
1553  002170  000000                        0
1554
1555                                  ;ITEM   54
1556
1557  002172  027721                        EM77    ;RKDA CHANGED AFTER 'DRIVE RESET'
1558  002174  032047                        DH4     ;PC     EXPCT   RECVD
1559  002176  031732                        DT2     ;$ERRPC $REG0   $REG1
1560  002200  000000                        0
1561
1562                                  ;ITEM   55
1563
1564  002202  027756                        EM100   ;DATA ERROR AT WORD#
1565  002204  032753                        DH100   ;PC     WORD#   EXPCT   RECVD
1566  002206  031762                        DT26    ;$ERRPC $REG0   $REG1   $REG2
1567  002210  000000                        0
1568
1569                                  ;ITEM   56
1570
1571  002212  030001                        EM101   ;CNTRL RDY DID NOT SET AFTER READ CHECK
1572  002214  032327                        DH44    ;PC     RKCS    RKER    RKDS    RKDA
1573  002216  031742                        DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
```



```
1574  002220  000000                        0
1575
1576                                  ;ITEM   57
1577
1578  002222  030043                        EM102   ;'ERR' OF 'HE' SET ON READ CHECK
1579  002224  032155                        DH30    ;PC     RKCS    RKER    RKDS
1580  002226  031762                        DT26    ;$ERRPC $REG0   $REG1   $REG2
1581  002230  000000                        0
1582
1583                                  ;ITEM   60
1584
1585  002232  030067                        EM103   ;'CSE' ON READ CHECK
1586  002234  033010                        DH103   ;PC     RKER
1587  002236  031724                        DT1     ;$ERRPC $REG0
1588  002240  000000                        0
1589
1590                                  ;ITEM   61
1591
1592  002242  030105                        EM104   ;RKWC DID NOT OVERFLOW ON READ CHECK OR WRITE CHECK
1593  002244  033024                        DH104   ;PC     RECVD   RKCS
1594  002246  031732                        DT2     ;$ERRPC $REG0   $REG1
1595  002250  000000                        0
1596
1597                                  ;ITEM   62
1598
1599  002252  030156                        EM105   ;RKDA INCREMENTED WRONG ON READ CHECK
1600  002254  032047                        DH4     ;PC     EXPCT   RECVD
1601  002256  031732                        DT2     ;$ERRPC $REG0   $REG1
1602  002260  000000                        0
1603
1604                                  ;ITEM   63
1605
1606  002262  030214                        EM106   ;RKBA CHANGED AFTER READ CHECK
1607  002264  032047                        DH4     ;PC     EXPCT   RECVD
1608  002266  031732                        DT2     ;$ERRPC $REG0   $REG1
1609  002270  000000                        0
1610
1611                                  ;ITEM   64
1612
1613  002272  030245                        EM107   ;MEMORY WORD CHANGED AFTER READ CHECK
1614  002274  033050                        DH107   ;PC     LOC     EXPCT   RECVD
1615  002276  031762                        DT26    ;$ERRPC $REG0   $REG1   $REG2
1616  002300  000000                        0
1617
1618                                  ;ITEM   65
1619
1620  002302  030306                        EM110   ;CNTRL RDY DID NOT SET AFTER WRITE CHECK
1621  002304  032327                        DH44    ;PC     RKCS    RKER    RKDS    RKDA
1622  002306  031742                        DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
1623  002310  000000                        0
1624
1625                                  ;ITEM   66
1626
1627  002312  030351                        EM111   ;HE OR ERR BIT SET AFTER DOING WRITE CHECK
1628  002314  032155                        DH30    ;PC     RKCS    RKER    RKDS
1629  002316  031762                        DT26    ;$ERRPC $REG0   $REG1   $REG2
```

```
1630  002320  000000                        0
1631
1632                                  ;ITEM   67
1633
1634  002322  030376                        EM112   ;WRITE CHECK ERROR
1635  002324  032155                        DH30    ;PC     RKCS    RKER    RKDS
1636  002326  031762                        DT26    ;$ERRPC $REG0   $REG1   $REG2
1637  002330  000000                        0
1638
1639                                  ;ITEM   70
1640
1641  002332  030417                        EM113   ;RKDA INCREMENTED WRONG ON WRITE CHECK
1642  002334  032047                        DH4     ;PC     EXPCT   RECVD
1643  002336  031732                        DT2     ;$ERRPC $REG0   $REG1
1644  002340  000000                        0
1645
1646                                  ;ITEM   71
1647
1648  002342  030456                        EM114   ;RKBA INCREMENTED WRONG ON WRITE CHECK
1649  002344  032047                        DH4     ;PC     EXPCT   RECVD
1650  002346  031732                        DT2     ;$ERRPC $REG0   $REG1
1651  002350  000000                        0
1652
1653                                  ;ITEM   72
1654
1655  002352  030515                        EM115   ;RKBA INCREMENTED WITH IBA SET
1656  002354  032047                        DH4     ;PC     EXPCT   RECVD
1657  002356  031732                        DT2     ;$ERRPC $REG0   $REG1
1658  002360  000000                        0
1659
1660                                  ;ITEM   73
1661
1662  002362  030551                        EM116   ;WRONG MEMORY LOCATION CHANGED WITH IBA SET
1663  002364  032753                        DH100   ;PC     WORD#   EXPCT   RECVD
1664  002366  031762                        DT26    ;$ERRPC $REG0   $REG1   $REG2
1665  002370  000000                        0
1666
1667                                  ;ITEM   74
1668
1669  002372  030624                        EM117   ;RK11 DID NOT INTERRUPT WHEN IDE WAS SET
1670  002374  033106                        DH117   ;PC     RKCS
1671  002376  031724                        DT1     ;$ERRPC $REG0
1672  002400  000000                        0
1673
1674                                  ;ITEM   75
1675
1676  002402  030671                        EM120   ;RK11 DID NOT INTERRUPT AFTER SEEK WAS INITIATED
1677  002404  033106                        DH117   ;PC     RKCS
1678  002406  031724                        DT1     ;$ERRPC $REG0
1679  002410  000000                        0
1680
1681                                  ;ITEM   76
1682
1683  002412  030744                        EM121   ;SCP SET BEFORE SEEK COMPLETED
1684  002414  033106                        DH117   ;PC     RKCS
1685  002416  031724                        DT1     ;$ERRPC $REG0
```



```
1686  002420  000000                        0
1687
1688                                  ;ITEM   77
1689
1690  002422  031002                        EM122   ;RK11 DID NOT INTERRUPT AFTER SEEK COMPLETED
1691  002424  032155                        DH30    ;PC     RKCS    RKER    RKDS
1692  002426  031762                        DT26    ;$ERRPC $REG0   $REG1   $REG2
1693  002430  000000                        0
1694
1695                                  ;ITEM   100
1696
1697  002432  031051                        EM123   ;CNTRL RESET DID NOT CLEAR 'SCP' BIT
1698  002434  033106                        DH117   ;PC     RKCS
1699  002436  031724                        DT1     ;$ERRPC $REG0
1700  002440  000000                        0
1701
1702                                  ;ITEM   101
1703
1704  002442  031110                        EM124   ;RK11 DID NOT INTERRUPT AFTER READ WAS DONE
1705  002444  033106                        DH117   ;PC     RKCS
1706  002446  031724                        DT1     ;$ERRPC $REG0
1707  002450  000000                        0
1708
1709                                  ;ITEM   102
1710
1711  002452  031152                        EM125   ;CNTRL RESET DID NOT CLEAR REGISTER
1712  002454  032020                        DH2     ;PC     REGADD  RECVD
1713  002456  031732                        DT2     ;$ERRPC $REG0   $REG1
1714  002460  000000                        0
1715
1716                                  ;ITEM   103
1717
1718  002462  031211                        EM126   ;RK11 DID NOT INTERRUPT AT CPU LEVEL
1719  002464  033122                        DH126   ;PC     LEVEL   RKCS
1720  002466  031732                        DT2     ;$ERRPC $REG0   $REG1
1721  002470  000000                        0
1722
1723                                  ;ITEM   104
1724
1725  002472  031252                        EM127   ;RK11 INTERRUPTED AT WRONG CPU LEVEL
1726  002474  033122                        DH126   ;PC     LEVEL   RKCS
1727  002476  031732                        DT2     ;$ERRPC $REG0   $REG1
1728  002500  000000                        0
1729
1730                                  ;ITEM   105
1731
1732  002502  031314                        EM130   ;'ERR BIT' DID NOT SET IN RKER
1733  002504  033150                        DH130   ;PC     RKCS    RKER    ERR BIT
1734  002506  031762                        DT26    ;$ERRPC $REG0   $REG1   $REG2
1735  002510  000000                        0
1736
1737
1738                                  ;ITEM   106
1739
1740  002512  031351                        EM131   ;HE OR ERR DID NOT SET
1741  002514  033207                        DH131   ;PC     RKCS    RKER
```

```
1742  002516  031732                        DT2     ;$ERRPC $REG0   $REG1
1743  002520  000000                        0
1744
1745                                  ;ITEM   107
1746
1747  002522  031376                        EM132   ;RKER ERROR
1748  002524  032047                        DH4     ;PC     EXPCT   RECVD
1749  002526  031732                        DT2     ;$ERRPC $REG0   $REG1
1750  002530  000000                        0
1751
1752                                  ;ITEM   110
1753
1754  002532  031410                        EM133   ;NXC BIT DID NOT SET
1755  002534  033235                        DH133   ;PC     RKCS    RKER    RKDA
1756  002536  031762                        DT26    ;PC     $REG0   $REG1   $REG2
1757  002540  000000                        0
1758
1759                                  ;ITEM   111
1760
1761  002542  031433                        EM134   ;RK11 DIDN'T INTERRUPT ON SOFT ERROR
1762  002544  033207                        DH131   ;PC     RKCS    RKER
1763  002546  031732                        DT2     ;$ERRPC $REG0   $REG1
1764  002550  000000                        0
1765
1766                                  ;ITEM   112
1767
1768  002552  031474                        EM135   ;MEX BITS INCREMENTED WRONG IN RKCS
1769  002554  032047                        DH4     ;PC     EXPCTD  RECVD
1770  002556  031732                        DT2     ;$ERRPC $REG0   $REG1
1771  002560  000000                        0
1772
1773                                  ;ITEM   113
1774
1775  002562  030306                        EM110   ;CNTRL RDY DID NOT SET AFTER WRT CHK
1776  002564  032113                        DH14    ;PC     RKCS    RKER    RKWC
1777  002566  031762                        DT26    ;$ERRPC $REG0   $REG1   $REG2
1778  002570  000000                        0
1779
1780                                  ;ITEM   114
1781
1782  002572  031531                        EM137   ;'WPS' NOT CLEAR
1783  002574  032327                        DH44    ;PC     RKCS    RKER    RKDS    RKDA
1784  002576  031742                        DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
1785  002600  000000                        0
1786
1787                                  ;ITEM   115
1788
1789  002602  031547                        EM140   ;DATA ERROR ON TRANSFER FROM DISK TO TTY
1790  002604  033273                        DH140   ;PC     EXPCT   RECVD   RKBA    RKCS
1791  002606  031742                        DT20    ;$ERRPC $REG0   $REG1   $REG2   $REG3
1792  002610  000000                        0
1793
1794
1795                                  ;ITEM   116
1796
1797  002612  031616                        EM141   ;'DRIVE #' PRESENT, BUT NOT SPECIFIED
```



```
1798  002614  032732                        DH74    ;PC     DRIVE #
1799  002616  031724                        DT1     ;$ERRPC $REG0
1800  002620  000000                        0
1801
1802                                  ;ITEM   117
1803
1804  002622  025370                        EM11    ;RKWC ERROR
1805  002624  032047                        DH4     ;PC     EXPCT   RECVD
1806  002626  031732                        DT2     ;$ERRPC $REG0   $REG1
1807  002630  000000                        0
1808                                  ;ITEM   120
1809  002632  031662                        EM142
1810  002634  000000                        0
1811
1812
1813
```

```
1814  002636  000005                          START:  RESET                   ;CLEAR THE BUS
1815                                          ;;GIVE DRIVES TIME TO LOAD HEADS IN CASE OF AN APT START.
1816  002640  023737  000042  000046                  CMP     @#42,@#46       ;ARE WE IN ACT11 AUTOMATIC MODE?
1817  002646  001016                                  BNE     STARTA          ;NO, SKIP DELAY
1818  002650  005077  176464                          CLR     @RKDA           ;SELECT UNIT 0
1819  002654  012700  000250                          MOV     #250,R0         ;WAIT FOR..
1820  002660  032777  000200  176440          20$:    BIT     #200,@RKDS      ;DRIVE READY..
1821  002666  001006                                  BNE     STARTA          ;IN CASE..
1822  002670  005001                                  CLR     R1              ;OF APT..
1823  002672  005301                                  DEC     R1              ;START, BUT..
1824  002674  001376                                  BNE     .-2             ;DON'T WAIT..
1825  002676  005300                                  DEC     R0              ;FOREVER.
1826  002700  001367                                  BNE     20$
1827  002702  000000                                  HALT                    ;RKDS BIT 7 (DRIVE READY) NEVER SET
1828  002704                                  STARTA:
1829                                          .SBTTL  INITIALIZE THE COMMON TAGS
1830                                          ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1831  002704  012706  001100                          MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1832  002710  005026                                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1833  002712  022706  001140                          CMP     #SWR,R6 ;;DONE?
1834  002716  001374                                  BNE     .-6             ;;LOOP BACK IF NO
1835  002720  012706  001100                          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1836                                          ;;INITIALIZE A FEW VECTORS
1837  002724  012737  022140  000020                  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1838  002732  012737  000340  000022                  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1839  002740  012737  022412  000030                  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1840  002746  012737  000340  000032                  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1841  002754  012737  024676  000034                  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1842  002762  012737  000340  000036                  MOV     #340,@#TRAPVEC+2;LEVEL 7
1843  002770  012737  024776  000024                  MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1844  002776  012737  000340  000026                  MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1845  003004  005037  001206                          CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1846  003010  005037  001210                          CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1847  003014  112737  000001  001115                  MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1848  003022  012737  003022  001106                  MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1849  003030  012737  003030  001110                  MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1850                                          ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1851                                          ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1852  003036  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1853  003042  012737  003076  000004                  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1854  003050  012737  177570  001140                  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1855  003056  012737  177570  001142                  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1856  003064  022777  177777  176046                  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1857  003072  001012                                  BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1858                                                                          ;;AND  THE HARDWARE SWR IS NOT = -1
1859  003074  000403                                  BR      65$             ;;BRANCH IF NO TIMEOUT
1860  003076  012716  003104          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1861  003102  000002                                  RTI
1862  003104  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1863  003112  012737  000174  001142                  MOV     #DISPREG,DISPLAY
1864  003120  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1865
1866  003124  023737  000042  000046                  CMP     @#42,@#46       ;ARE WE IN ACT11 AUTOMATIC MODE?
1867  003132  001416                                  BEQ     69$             ;YES, SKIP TITLE
1868                                          .SBTTL  TYPE PROGRAM NAME
1869                                          ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
```



```
1870  003134  005227  177777                          INC     #-1             ;;FIRST TIME?
1871  003140  001043                                  BNE     67$             ;;BRANCH IF NO
1872  003142  104401  003200                          TYPE    ,68$            ;;TYPE ASCIZ STRING
1873                                          .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1874  003146  005737  000042                          TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1875  003152  001006                                  BNE     69$             ;;BRANCH IF YES
1876  003154  023727  001140  000176                  CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1877  003162  001005                                  BNE     70$             ;;BRANCH IF NO
1878  003164  104406                                  GTSWR                   ;;GET SOFT-SWR SETTINGS
1879  003166  000403                                  BR      70$
1880  003170  112737  000001  001134  69$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
1881  003176                          70$:
1882  003176  000424                                  BR      67$             ;;GET OVER THE ASCIZ
1883                                          ;;68$:  .ASCIZ  <CRLF>/RK11 LOGIC TEST 2/<15><12>/MAINDEC-11-CZRKKF/<CRLF>
1884  003250                          67$:
1885  003250  012700  001410                          MOV     #DDPCH,R0
1886  003254  012701  177764                          MOV     #-14,R1
1887  003260  005020                  1$:     CLR     (R0)+
1888  003262  005201                                  INC     R1
1889  003264  001375                                  BNE     1$
1890  003266  005227  177777                          INC     #-1             ;FIRST START ?
1891  003272  001020                                  BNE     START1          ;BR IF NOT
1892  003274  013746  000004                          MOV     ERRVEC,-(SP)    ;SAVE ERROR VECTOR ADDRESS
1893  003300  012737  003314  000004                  MOV     #2$,ERRVEC      ;NEW VECTOR ADDRESS
1894  003306  005737  177776                          TST     PS              ;SEE IF PROGRAM CAN REFERENCE THE
1895                                                                          ;PROCESSOR STATUS WORD
1896  003312  000406                                  BR      3$              ;BR IF REFERENCE DIDN'T CAUSE TRAP
1897  003314  012737  000140  001400  2$:     MOV     #140,RKPRI      ;SETUP INTERRUPTING PRIORITY TO VALUE
1898                                                                          ;WHICH WILL ALLOW INTERRUPT ON AN LSI-11
1899  003322  012716  003330                          MOV     #3$,(SP)        ;SETUP RETURN ADDRESS
1900  003326  000002                                  RTI                     ;RETURN
1901  003330  012637  000004          3$:     MOV     (SP)+,ERRVEC    ;RESTORE THE ERROR VECTOR
1902                                          ;
1903                                          ;FIND OUT IF ACT11, 'XXDP' CHAIN OR DUMP MODE
1904                                          ;
1905  003334  012700  001410          START1: MOV     #DDPCH,R0
1906  003340  012701  177766                          MOV     #-12,R1         ;CLEAR OUT DRIVE TABLE AREA
1907  003344  005020                  1$:     CLR     (R0)+
1908  003346  005201                                  INC     R1
1909  003350  001375                                  BNE     1$
1910  003352  122737  000002  000041                  CMPB    #2,41           ;LOADED FROM AN RK05 ?
1911  003360  001166                                  BNE     ST2             ;BR IF NOT
1912  003362  013737  000040  001410                  MOV     40,DDPCH        ;GET DEVICE INDICATOR AND DRIVE ADDRESS OF
1913                                                                          ;LOADING RK05
1914  003370  122737  000010  001410                  CMPB    #10,DDPCH       ;VALID DRIVE NUMBER IN BYTE 40 ?
1915  003376  101002                                  BHI     2$              ;BR IF YES
1916  003400  105037  001410                          CLRB    DDPCH           ;MUST BE DRIVE ZERO WHICH LOADED
1917                                                                          ;THIS PROGRAM
1918  003404  005737  000042          2$:     TST     42              ;CHAIN MODE OR ACT11 AUTO ACCEPT ?
1919  003410  001432                                  BEQ     4$              ;BR IF NEITHER
1920  003412  005737  001410                          TST     DDPCH           ;RUNNING FROM AN RK05 ?
1921  003416  001002                                  BNE     3$              ;BR IF YES
1922  003420  000137  004262                          JMP     ST3             ;FIND OUT NUMBER OF DRIVES
1923  003424                          3$:
1924  003424  104401  003432                          TYPE    ,65$            ;;TYPE ASCIZ STRING
1925  003430  000413                                  BR      64$             ;;GET OVER THE ASCIZ
```

```
                                        ;;65$:  .ASCIZ  <15><12>/NOT TESTING DRIVE /
                                        64$:
003460  005046                                    CLR     -(SP)           ;CLEAR WORD ON STACK
003462  113716  001410                            MOVB    DDPCH,(SP)      ;GET DRIVE ADDRESS
003466  104403                                    TYPOS                   ;TYPE THE ADDRESS
003470     001                                    .BYTE   1               ;ONLY 1 CHARACTER
003471     000                                    .BYTE   0               ;SUPRESS LEADING ZEROS
003472  000137  004262                            JMP     ST3             ;GET NUMBER OF DRIVES
003476  005227  177777                    4$:     INC     #-1             ;FIRST TIME THROUGH HERE ?
003502  001115                                    BNE     ST2             ;BR IF NOT
003504  104401  003512                            TYPE    ,67$            ;;TYPE ASCIZ STRING
003510  000411                                    BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  <15><12>/TO TEST DRIVE /
003534                                  66$:
003534  005046                                    CLR     -(SP)           ;CLEAR WORD ON THE STACK
003536  113716  001410                            MOVB    DDPCH,(SP)      ;GET DRIVE ADDRESS
003542  104403                                    TYPOS                   ;TYPE THE DRIVE ADDRESS
003544     001                                    .BYTE   1               ;ONLY 1 CHARACTER
003545     000                                    .BYTE   0               ;SUPRESS LEADING ZEROS
003546  104401  003554                            TYPE    ,69$            ;;TYPE ASCIZ STRING
003552  000431                                    BR      68$             ;;GET OVER THE ASCIZ
                                        ;;69$:  .ASCIZ  / HALT PROGRAM, REMOVE RKDP PACK AND REPLACE IT/<15><12>
003636                                  68$:
003636  104401  003644                            TYPE    ,71$            ;;TYPE ASCIZ STRING
003642  000435                                    BR      70$             ;;GET OVER THE ASCIZ
                                        ;;71$:  .ASCIZ  /WITH A WORK PACK, CLEAR LOCATION 40, AND RESTART PROGRAM/
003736                                  70$:

                                        ;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.
003736  012700  001412                    ST2:    MOV     #DRIVS,R0
003742  012701  177765                            MOV     #-13,R1
003746  005020                            13$:    CLR     (R0)+
003750  005201                                    INC     R1
003752  001375                                    BNE     13$
003754  104401  003762                            TYPE    ,65$            ;;TYPE ASCIZ STRING
003760  000415                                    BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <15><12>/DRIVES TO BE TESTED ?/<15><12>
004014                                  64$:
004014  104411                                    RDLIN
004016  012600                                    MOV     (SP)+,R0        ;GET STARTING ADRES OF ASCII STRING
004020  012701  177770                            MOV     #-10,R1         ;SET UP COUNT
004024  112002                            1$:     MOVB    (R0)+,R2        ;GET ASCII CHARACTER
004026  042702  177400                            BIC     #177400,R2      ;MASK UNWANTED BITS
004032  012703  001414                            MOV     #DRIV0,R3
004036  012704  177770                            MOV     #-10,R4
004042  012705  000060                            MOV     #60,R5
004046  020502                            2$:     CMP     R5,R2           ;WAS THE TYPED IN CHARACTER
                                                                        ;A NUMBER BETWEEN 0-7?
004050  001414                                    BEQ     3$              ;YES, BRANCH
004052  005205                                    INC     R5              ;NO, INCREMENT
004054  005723                                    TST     (R3)+           ;INCREMENT POINTER TO DRV FLAG
004056  005204                                    INC     R4              ;CHARACTER THAT WAS INPUT
```



```
004060  001372                                    BNE     2$              ;SHOULD BE 0-7, IF ANY OTHER
                                                                        ;TYPE ?? & AGAIN ASK FOR
                                                                        ;DRIVS TO BE TSTD?
004062  005702                                    TST     R2              ;IS IT A TERMINATOR?
004064  001461                                    BEQ     6$              ;YES, EXIT. NO DRIVES INDICATED.
004066                                  4$:
004066  104401  004074                            TYPE    ,67$            ;;TYPE ASCIZ STRING
004072  000402                                    BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  /??/
004100                                  66$:
004100  000716                                    BR      ST2             ;GO, AGAIN ASK QUESTION
004102  005713                            3$:     TST     @R3     ;SEE IF ALL READY SELECTED
004104  001370                                    BNE     4$      ;ERROR IF SELECTED  ALL READY
004106  005213                                    INC     @R3             ;SET UP FLAG FOR THE DRIVE
004110  005237  001412                            INC     DRIVS           ;INCREMENT TOTAL NO OF DRIVES PRESENT
004114  111002                            11$:    MOVB    @R0,R2          ;GET NEXT CHAR
004116  042702  177400                            BIC     #177400,R2      ;CHARACTER ONLY
004122  022702  000106                            CMP     #'F,R2          ;IS IT F?
004126  001026                                    BNE     8$              ;NO, GO ON
004130  052713  100000                            BIS     #BIT15,@R3      ;SET BIT 15 TO SHOW RK05F
004134  032705  000001                            BIT     #BIT0,R5        ;EVEN DRIVE?
004140  001407                                    BEQ     9$              ;EVEN DRIVE SO BRANCH
004142  005763  177776                            TST     -2(R3)          ;CHECK EVEN DRIVE
004146  001347                                    BNE     4$              ;EVEN ALL READY SELECTED
004150  012763  100001  177776                    MOV     #BIT15!BIT0,-2(R3)      ;SELECT EVEN DRIVE
004156  000406                                    BR      10$             ;CONTINUE
004160  005763  000002                    9$:     TST     2(R3)           ;CHECK ODD DRIVE
004164  001340                                    BNE     4$              ;ERROR IF SELECTED BEFORE
004166  012763  100001  000002                    MOV     #BIT15!BIT0,2(R3)       ;SELECT ODD DRIVE
004174  005237  001412                    10$:    INC     DRIVS           ;COUNT DRIVES SELECTED
004200  105720                                    TSTB    (R0)+           ;POINT TO NEXT CHAR
004202  000744                                    BR      11$             ;CHECK FOR COMMA
004204  022702  000054                    8$:     CMP     #54,R2          ;IS IT A 'COMMA'?
004210  001403                                    BEQ     5$              ;YES, GO PROCESS NXT WORD
004212  005702                                    TST     R2              ;NO, IS IT A TERMINATOR?
004214  001324                                    BNE     4$              ;IF NOT, SOMETHING WRONG
                                                                        ;GO ASK QUESTION AGAIN
004216  000404                                    BR      6$              ;EXIT, IF A TERMINATOR
004220  105720                            5$:     TSTB    (R0)+           ;INCREMENT PTR TO NXT BYTE
                                                                        ;IN INPUT BUFFER
004222  005201                                    INC     R1              ;THERE SHOULD BE NO MORE THAN
004224  001277                                    BNE     1$              ;8 DRIVES, HENCE IF MORE
004226  000717                                    BR      4$              ;THAN 8 DIFFERENT NOS. TYPED IN, ERROR!
                                                                        ;GO AGAIN ASK THE QUESTION

004230  005037  001440                    6$:     CLR     SIZYET          ;NO SIZING NEEDED
004234  032777  002000  174676                    BIT     #SW10,@SWR      ;TESTING ON SIMULATOR?
004242  001003                                    BNE     7$              ;YES, BRANCH
004244  005037  001344                            CLR     SIMUL           ;NO, CLR FLAG
004250  000502                                    BR      ST4

004252  012737  000001  001344            7$:     MOV     #1,SIMUL        ;SET FLAG TO INDICATE SIMULATOR
004260  000476                                    BR      ST4
```

```
                                ;CHECK NUMBER OF DRIVES
004262  012737  177777  001440  ST3:    MOV     #-1,SIZE        ;CHECK FOR RK05F LATER
004270  012737  004442  000004          MOV     #5$,@#4         ;SET UP ADRES FOR TIME-OUT VECTOR
004276  005777  175024                  TST     @RKDS           ;REFERENCE RKDS
004302  005777  175032                  TST     @RKDA           ;REFERENCE RKDA
004306  012737  004534  000004          MOV     #BADTMO,@#4
004314  104401                          TYPE
004316  001216                          MSG1
004320  012700  177770                  MOV     #-10,R0         ;INITIALIZE COUNT FOR THE 8 DRIVES
004324  005037  001412                  CLR     DRIVS           ;INITIALIZE # OF DRIVES PRESENT TO 0
004330  005001                          CLR     R1              ;INITIALIZE ADDRESS TO DRIVE 0
004332  005004                          CLR     R4
004334  012702  001414                  MOV     #DRIV0,R2
004340  010177  174774          1$:     MOV     R1,@RKDA        ;ADDRESS THE DRIVE
004344  020177  174770                  CMP     R1,@RKDA        ;CHECK, WAS IT ADDRESSED?
004350  001405                          BEQ     3$              ;YES
004352  012703  004356                  MOV     #2$,R3
004356  004737  021026          2$:     JSR     PC,TYERM        ;WHILE CHECKING NUMBER OF DRIVE
                                                                ;UNDER NON-MANUAL MODE :-
                                                                ;RKDA HAD TO BE ADRESED BUT
                                                                ;IT WAS FOUND THAT THE DRIVE NO
                                                                ;THAT WAS WRITTEN COULD NOT BE READ BACK
                                                                ;CORRECTLY.
004362  000413                          BR      4$
004364  032777  000200  174734  3$:     BIT     #200,@RKDS      ;CHECK IF 'DRY' BIT IS SET, IF SET DRIVE IS
                                                                ;PRESENT
004372  001407                          BEQ     4$              ;
004374  104401                          TYPE
004376  001213                          $CRLF
004400  005237  001412                  INC     DRIVS           ;IF PRESENT, INCREMENT # OF DRIVES
004404  005212                          INC     (R2)            ;SET UP FLAG INDICATING THIS DRIVE PRESENT
004406  010446                          MOV     R4,-(SP)
004410  104402                          TYPOC
004412  005722          4$:     TST     (R2)+           ;SHIFT POINTER TO NXT DRIVE INDICATOR
004414  062701  020000                  ADD     #20000,R1       ;SET UP ADDRESS FOR THE NEXT DRIVE
004420  005204                          INC     R4              ;HAVE U CHECKED FOR ALL 8 DRIVES
004422  005200                          INC     R0
004424  001345                          BNE     1$
004426  005737  001412                  TST     DRIVS
004432  001011                          BNE     ST4
004434  104401                          TYPE
004436  001236                          MSG2
004440  000406                          BR      ST4             ;GO CHECK THE DRIVE INDEPENDENT
                                                                ;CONTROLLER LOGIC
004442  011603          5$:     MOV     (SP),R3         ;GET PC WHERE TIMEOUT OCCURED
004444  022626                  CMP     (SP)+,(SP)+     ;RESTORE STACK
004446  062703  177776                  ADD     #-2,R3
004452  004737  021026                  JSR     PC,TYERM        ;GO TYPE ERROR MESSAGE
                                                                ;WHILE CHECKING FOR THE NUMBER OF
                                                                ;DRIVES IN NON-MANUAL MODE:-
                                                                ;RKDS AND RKDA HAD TO BE REFERENCED, TIMEOUT
                                                                ;OCCURED ON REFERENCING.PC IN THE ERROR
                                                                ;MESSAGE INDICATES WHERE THE TIMEOUT OCCURED.
                                ;
```



```
004456  005037  001434          ST4:    CLR     T56FLG
004462  005737  001412                  TST     DRIVS
004466  001004                          BNE     1$
004470  004737  021742                  JSR     PC,WATIME
004474  000137  020652                  JMP     $EOP
004500  012737  001414  001354  1$:     MOV     #DRIV0,DRVPTR
004506  005037  001352                  CLR     DRVDON          ;INITIALIZE THE NO. OF DRIVES
                                                                ;THAT HAVE BEEN CHECKED
004512  005037  001350                  CLR     DRIVAD          ;INITIALIZE DRIVE ADDRESS TO
                                                                ;THE FIRST DRIVE
004516  012737  004534  000004          MOV     #BADTMO,@#4     ;SET TIME OUT VECTOR FOR UNEXPECTED
                                                                ;TIME OUTS
004524  012777  004600  174650          MOV     #BADINT,@RKVEC  ;SET UP RK11 INTERRUPT VECTOR FOR
                                                                ;UNEXPECTED INTERRUPTS FROM RK11
004532  000465                          BR      TST1            ;GO TO TEST 1


                                ;THIS ROUTINE HANDLES UNEXPECTED TIME OUTS

004534  011600          BADTMO: MOV     (SP),R0 ;SAVE PC WHERE TIME OUT OCCURED
004536  005740                  TST     -(R0)
004540  022626                  CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
004542  104401  004550          TYPE    ,65$            ;;TYPE ASCIZ STRING
004546  000407                  BR      64$             ;;GET OVER THE ASCIZ
                        ;;65$:  .ASCIZ  <15><12>/TIMOUT,PC=/
004566                  64$:
004566  010046                  MOV     R0,-(SP)        ;SET UP FOR TYPING OUT PC
004570  104402                  TYPOC                   ;GO TYPE OUT OCTAL PC
004572  000000                  HALT
004574  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.

004600  011600          BADINT: MOV     (SP),R0         ;SAVE PC WHERE INTERRUPT OCCURED
004602  005740                  TST     -(R0)
004604  032777  020000  174326          BIT     #20000,@SWR     ;INHIBIT ERROR TYPEOUT?
004612  001014                  BNE     1$              ;YES, DON'T TYPE OUT
004614  104401                  TYPE
004616  001213                  $CRLF
004620  104401                  TYPE
004622  026004                  EM43                    ;TYPE 'UNEXPEXTED RK11 INTERRUPT'
                                                        ;TYPE ' AT PC='
004624  104401  004632          TYPE    ,65$            ;;TYPE ASCIZ STRING
004630  000403                  BR      64$             ;;GET OVER THE ASCIZ
                        ;;65$:  .ASCIZ  /,PC=/
004640                  64$:
004640  010046                  MOV     R0,-(SP)        ;SET UP FOR TYPING OUT PC
```

```
                        TYPOC                   ;GO TYPE OCTAL PC WHERE BAD
                                                ;INTERUPT OCCURED
004644  032777  001000  174266  1$:   BIT     #1000,@SWR      ;LOOP ON ERROR?
004652  001403                      BEQ     2$              ;NO, BRANCH
004654  022626                      CMP     (SP)+,(SP)+     ;YES, REPOSITION STACK
004656  000177  174224              JMP     @$LPADR         ;GO TO THE STARTING ADDRESS OF
                                                ;THE TEST THAT GAVE UNEXPECTED INTERRUPT
004662  032777  040000  174250  2$:   BIT     #40000,@SWR     ;LOOP ON TEST?
004670  001401                      BEQ     3$              ;NO, BRANCH
004672  000002                      RTI                     ;YES, LOOP. GO BACK WHER U INTERRUPTED FROM.
004674  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.
004676  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.

004702  004737  021742  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.)
                        ;;**************************************************************
004706  000004              TST1:   SCOPE

004710  012700  001414              MOV     #DRIV0,R0       ;INITIALIZE POINTER
004714  005001                      CLR     R1              ;INITIALIZE DRIVE ADRES 0
004716  005002                      CLR     R2              ;INITIALIZE DRIVE # 0
004720  005737  001410      1$:     TST     DDPCH           ;LOADED FROM AN RK05 ?
004724  001403                      BEQ     2$              ;B IF NOT
004726  120237  001410              CMPB    R2,DDPCH        ;LOADED FROM THIS DRIVE ?
004732  001435                      BEQ     4$              ;BR IF YES
004734  010177  174400      2$:     MOV     R1,@RKDA        ;ADRES THE DRIVE
004740  105777  174362              TSTB    @RKDS           ;DRIVE READY?
004744  100005                      BPL     3$              ;NO, THIS DRIVE NOT PRESENT
                                                ;YES, THIS DRIVE SELECTED
004746  005710                      TST     @R0             ;WAS THIS DRIVE SPECIFIED BY
```



```
                                                ;THE USER?
004750  001026                      BNE     4$              ;YES, OK
                                                ;NO, THIS DRIVE # WAS NOT SPECIFIED
                                                ;BY THE USER, BUT STILL IS GIVING
                                                ;'DRY' WHEN ADRESED. REPORT EROR.
004752  010237  001162              MOV     R2,$REG0        ;GET DRIVE #
004756  104116                      ERROR   116             ;THIS DRIVE # WAS NOT SPECIFIED BY
                                                ;THE USER, BUT WHEN ADRESED GAVE
                                                ;'DRY'. CHECK THAT THIS DRIVE # IF
                                                ;PHYSICALLY PRESENT IS ON 'LOAD'. IF
                                                ;THIS IS NOT THE CASE, THEN ONE DRIVE
                                                ;MAY BE GETTING SELECTED BY TWO DIFFERENT
                                                ;LOGICAL ADDRESSES.
004760  005710              3$:     TST     @R0             ;CHECK THAT THIS DRIVE WAS NOT INDICATED
004762  001421                      BEQ     4$              ;IF IT WAS, & IT IS NOT FOUND TO BE
                                                ;PRESENT (DRY CLEAR), REPORT ERROR.
004764  004737  020774              JSR     PC,GT4RG        ;GET RKCS, ER, DS, DA
004770  104010                      ERROR   10              ;DRIVE # (AS IN RKDA) WAS INDICATED BY
                                                ;THE USER, BUT WAS NOT FOUND TO BE PRESENT.
                                                ;CHECK THAT THE  ROTARY DRIVE SELECTION
                                                ;SWITCH ON THE MODULE IS SET TO THE RIGHT
                                                ;DRIVE #.

004772  005010                      CLR     @R0             ;THIS DRIVE IS NOT FOUND TO BE PRESENT
                                                ;HENCE DROP IT FROM THE SELECTION TABLE.
004774  010003                      MOV     R0,R3           ;DRIVE ADDR
004776  162703  001414              SUB     #DRIV0,R3       ;MINUS OFFSET FOR TABLE
005002  042703  000003              BIC     #3,R3           ;EVEN DRIVE OF PAIR
005006  062703  001414              ADD     #DRIV0,R3       ;POINT TO EVEN OF PAIR IF RK05 F
005012  042723  100000              BIC     #100000,(R3)+   ;NOT SPECIFIED AS F MODEL
005016  042713  100000              BIC     #100000,(R3)    ;SAME
005022  005337  001412              DEC     DRIVS           ;DECREMENT DRIVE COUNT
005026  005202              4$:     INC     R2              ;INCRMNT DRIVE #
005030  005720                      TST     (R0)+           ;INCRMNT POINTER
005032  062701  020000              ADD     #20000,R1       ;INCRMNT ADRES TO NXT DRIVE
005036  001330                      BNE     1$              ;LUP BAK IF NOT DONE


                        ;THIS PART OF THE PROGRAM IS GOING TO BE REPEATED FOR
                        ;EACH DRIVE PRESENT
                        ;
                        ;'DRIVAD' CONTAINS IN BITS 15,14,13 THE ADDRESS OF THE
                        ;DRIVE BEING CURRENTLY CHECKED.
                        ;
005040              NUDRV:


                        ;;**************************************************************
                        ;*TEST 2         FIND OUT NEXT DRIVE TO BE CHECKED
                        ;THIS CODE FINDS OUT THE NEXT DRIVE THAT IS PRESENT AND THEN  SETS UP
                        ;THE ADDDRESS IN DRIVAD (BITS 13,14,15). THUS THROUGHOUT THE FOLLOWING TESTS
                        ;THE DRIVE TESTED IS THE DRIVE WHOOSE ADDRESS IS IN 'DRIVAD'.
                        ;;**************************************************************
005040  000004              TST2:   SCOPE
005042  012737  000001  001206      MOV     #1,$TIMES       ;;DO 1 ITERATION
005050  012737  000002  001102      MOV     #2,$TSTNM       ;RESET POINTER TO THIS TEST
```

```
                                                    ;NO. CHANGE THIS (2) IN CASE THE
                                                    ;TEST NO. CHANGES
005056  005037  001112                CLR     $ERTTL          ;CLEAR TOTAL ERROR COUNT
005062  005737  001412                TST     DRIVS           ;R THERE ANY DRIVES PRESENT?
005066  001002                      BNE     .+6             ;YES, BRANCH
005070  000137  020652          4$:   JMP     $EOP            ;NO, JMP TO THE END
005074  013701  001354                MOV     DRVPTR,R1       ;GET THAT POINTER TO THE NEXT
                                                    ;DRIVE FLAG
005100  032721  000001          2$:   BIT     #BIT0,(R1)+     ;IS THIS DRIVE PRESENT?
005104  001005                      BNE     1$              ;YES
005106  062737  020000  001350  6$:   ADD     #20000,DRIVAD   ;FORM NXT DRIVE ADRES
005114  001371                      BNE     2$
005116  000764                      BR      4$
005120  005737  001410          1$:   TST     DDPCH           ;PROGRAM LOADED FROM AN RK05 ?
005124  001413                      BEQ     3$              ;NO, BRANCH
005126  013746  001350                MOV     DRIVAD,-(SP)    ;PUT TEST DRIVE ADDRESS ON THE STACK
005132  000316                      SWAB    (SP)            ;SETUP TO RIGHT JUSTIFY THE ADDRESS
005134  006216                      ASR     (SP)            ;RIGHT JUSTIFY THE ADDRESS
005136  006216                      ASR     (SP)            ;RIGHT JUSTIFY THE ADDRESS
005140  006216                      ASR     (SP)            ;RIGHT JUSTIFY THE ADDRESS
005142  006216                      ASR     (SP)            ;RIGHT JUSTIFY THE ADDRESS
005144  006216                      ASR     (SP)            ;RIGHT JUSTIFY THE ADDRESS
005146  122637  001410                CMPB    (SP)+,DDPCH     ;PROGRAM LOADED FROM THIS DRIVE ?
005152  001755                      BEQ     6$              ;BR IF YES, DON'T TEST THE DRIVE
005154  010137  001354          3$:   MOV     R1,DRVPTR       ;STORE POINTER TO THE.NEXT
                                                    ;DRIVE FLAG
                                    TYPE    ,MSG4
005160  104401  001272                MOV     DRIVAD,-(R6)    ;GET THE DRIVE ADDRESS
005164  013746  001350                JSR     PC,SHFTRT       ;GO SHIFT IT TO THE RIGHT
005170  004737  021200                CLR     FFLAG
005174  005037  001404                MOV     (R6),R0         ;DRIVE NUMBER
005200  011600                      TYPOS                   ;GO TYPE THE OCTAL # FOR THE
005202  104403                                              ;DRIVE THAT IS BEING CHECKED
                                    .BYTE   1,0
005204     001     000              ASL     R0              ;INDEX TO TABLE
005205  006300                      TST     DRIV0(R0)       ;SEE IF F
005210  005760  001414                BPL     5$              ;NO
005214  100006                      TYPE    ,65$            ;;TYPE ASCIZ STRING
005216  104401  005224                BR      64$             ;;GET OVER THE ASCIZ
005222  000401              ;;65$:  .ASCIZ  /F/
                            64$:
005226                              INC     FFLAG           ;SET F FLAG
005226  005237  001404          5$:   TYPE
005232  104401                      $CRLF                   ;TYPE CR, LF
005234  001213              ;;**************************************************************
                            ;*TEST 3         CHECK THAT DRIVE IS SUPPLIED WITH POWER-DPL BIT
                            ;;**************************************************************
005236  000004              TST3:   SCOPE
005240  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
005242  013700  001326                MOV     RKDS,R0
005246  013777  001350  174064        MOV     DRIVAD,@RKDA    ;ADDRESS THE DRIVE UNDER TEST
005254  005710                      TST     @R0             ;CHECK IF ANY BIT OF RKDS IS SET?
005256  001003                      BNE     1$              ;IF SET, BRANCH
005260  011037  001162                MOV     @R0,$REG0       ;GET RKDS
005264  104004                      ERROR   4               ;RKDS ERROR! RKDS IF ADDRESSED
                                                    ;CORRECTLY SHOULD BE NON-ZERO
005266  012777  000015  174036  1$:   MOV     #15,@RKCS       ;ISSUE A DRV RESET, IF DRVE
                                                    ;POWER IS LO, DPL WILL SET
005274  005001                      CLR     R1
005276  032710  010000          2$:   BIT     #10000,@R0      ;IS 'DPL' BIT SET?
005302  001003                      BNE     3$              ;DPL IS SET, BRANCH
005304  005201                      INC     R1              ;WAIT FOR SOME TIME TO
005306  001373                      BNE     2$              ;SEE IF DPL WOULD SET
005310  000403                      BR      4$-2            ;OK, DPL NOT SET
005312  004737  021002          3$:   JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
005316  104005                      ERROR   5               ;DPL BIT OF RKDS IS SET, CHECK DRIVE POWER

005320  005001                      CLR     R1
005322  032710  000100          4$:   BIT     #100,@R0        ;DID R/W/S RDY BIT SET?
005326  001010                      BNE     TST4            ;;YES, EXIT
005330  104417  000011                DELAY   ,11             ;TIME DELAY
005334  005201                      INC     R1              ;WAIT FOR R/W/S RDY
005336  001371                      BNE     4$
005340  017737  173762  001162        MOV     @RKDS,$REG0     ;GET RKDS
005346  104016                      ERROR   16              ;R/W/S RDY DID NOT SET AFTER
                                                    ;DRIVE RESET. DRIVE RESET WAS DONE
                                                    ;TO CHECK 'DPL'BIT . THIS TEST
                                                    ;IS NOT FOR CHECKING DRIVE RESET.
                                                    ;U MIGHT WANT TO USE THE TEST PROVIDED
                                                    ;FOR CHECKING DRIVE RESET.

                            ;;**************************************************************
                            ;*TEST 4         CHECK THAT 'DRIVE UNSAFE' IS CLEAR, 'HDEN' IS SET, 'WPS' IS CLEAR
                            ;;**************************************************************
005350  000004              TST4:   SCOPE
005352  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
005354  013777  001350  173756        MOV     DRIVAD,@RKDA    ;SET DRIVE ADDRESS
005362  017700  173740                MOV     @RKDS,R0                ;GET RKDS
005366  032700  002000                BIT     #2000,R0        ;IS 'DRU' BIT OF RKDS SET?
005372  001403                      BEQ     1$              ;NO
005374  004737  021002                JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
005400  104006                      ERROR   6               ;'DRU' BIT OF RKDS IS SET, CHECK
                                                    ;DRIV BY PUTTING RUN/LOAD SW TO LOAD
```

```
                                                        ;THEN BACK TO RUN
005402  032700  004000          1$:     BIT     #4000,R0        ;IS 'HDEN' BIT SET?
005406  001004                          BNE     2$              ;YES, BRANCH
005410  017737  173712  001162          MOV     @RKDS,$REG0     ;GET RKDS
005416  104007                          ERROR   7               ;ERROR, 'RKOS' BIT IS NOT SET
005420  032777  000040  173700  2$:     BIT     #40,@RKDS       ;IS 'WPS' CLEAR?
005426  001403                          BEQ     TST5            ;;YES, EXIT
005430  004737  020774                  JSR     PC,GT4RG        ;GET RKCS, ER, DS, DA
005434  104114                          ERROR   114             ;'WPS'-WRITE PROTECT STATUS- BIT OF
                                                        ;OF RKDS SHOULD BE CLEAR, IF THIS DRIVE
                                                        ;IS WRITE ENABLED, CHECK & SEE IF THIS
                                                        ;DRIVE IS WRITE ENABLED, IF IT IS NOT,
                                                        ;WRITE ENABLE IT.

                                        ;
                                ;;**************************************************************
                                ;*TEST 5         CHECK THAT 'DRIVE READY' IS SET IN RKDS
                                ;;**************************************************************
005436  000004                  TST5:   SCOPE
005440  104413                          CNT.RESET               ;GO, DO CONTROL RESET
                                                        ;THIS IS A CALL FOR THE 'CNTRL-
                                                        ;RESET' ROUTINE. A CONTROL RESET IS
                                                        ;ISSUED AND AFTER A CERTAIN TIME
                                                        ;IF THE 'CNTRL RDY' DOES NOT SET
                                                        ;AN ERROR IS REPORTED. NOTE THAT
                                                        ;THE PC IN ERROR MESSAGE IS THE
                                                        ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                        ;THIS IS A VERY BASIC ERR & IF IT
                                                        ;OCCURS GO BACK TO TEST 10
005442  013777  001350  173670          MOV     DRIVAD,@RKDA    ;ADDRS THE DRIVE
005450  105777  173652                  TSTB    @RKDS           ;IS 'DRY' SET?
005454  100403                          BMI     TST6            ;;YES, OK
005456  004737  020774                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
005462  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
                                ;;**************************************************************
005464  000004                  TST6:   SCOPE
005466  013777  001350  173644          MOV     DRIVAD,@RKDA    ;ADDRS THE DRIVE
005474  005001                          CLR     R1              ;INITIALIZE COUNT FOR TIMING WAIT LOOP
005476  032777  000400  173622  1$:     BIT     #400,@RKDS      ;IS SOK SET?
005504  001006                          BNE     TST7            ;;EXIT
005506  005201                          INC     R1              ;NO, WAIT
005510  001372                          BNE     1$              ;WAITED LONG?
005512  017737  173610  001162          MOV     @RKDS,$REG0     ;GET RKDS
005520  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)
                                ;;**************************************************************
005522  000004                  TST7:   SCOPE
005524  104413                          CNT.RESET               ;GO, DO CONTROL RESET
                                                        ;THIS IS A CALL FOR THE 'CNTRL-
                                                        ;RESET' ROUTINE. A CONTROL RESET IS
                                                        ;ISSUED AND AFTER A CERTAIN TIME
                                                        ;IF THE 'CNTRL RDY' DOES NOT SET
                                                        ;AN ERROR IS REPORTED. NOTE THAT
                                                        ;THE PC IN ERROR MESSAGE IS THE
                                                        ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                        ;THIS IS A VERY BASIC ERR & IF IT
                                                        ;OCCURS GO BACK TO TEST 10
005526  013777  001350  173604          MOV     DRIVAD,@RKDA
005534  013700  001326                  MOV     RKDS,R0         ;INITIALIZE
005540  005037  001356                  CLR     INDX1           ;'COUNT' - TO TIME 'ERROR 35'
005544  005005                          CLR     R5              ;INITIALIZE 'COUNT' - TO TIME
                                                        ;'ERROR 36' (WAIT LOOP)
005546  012704  177764                  MOV     #-14,R4         ;INITIALIZE 'COUNT' - FOR THE 12 SECTORS.
005552  012703  000001                  MOV     #1,R3           ;R3 CONTAINS THE 'NEXT' COUNT OF SEC-CNTR
                                                        ;R1 CONTAINS THE 'PREVIOUS' COUNT OF SEC-CNTR
                                                        ;R2 CONTAINS THE 'PRESENT' COUNT OF SEC-CNTR
005556  005037  001360          1$:     CLR     INDX2           ;INITIALIZE 'COUNT' - TO TIME
                                                        ;(WAIT LOOP) 'ERROR 34'
005562  005237  001356                  INC     INDX1           ;KEEP TIMING FOR 'ERROR 35'
005566  001440                          BEQ     6$              ;BRANCH & REPORT ERROR IF WAITED LONG?
005570  005237  001360          2$:     INC     INDX2           ;KEEP TIMING FOR 'ERROR 34'
005574  001441                          BEQ     7$              ;BRANCH & REPORT ERROR IF WAITED LONG?

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

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



```
2542  005752  013700  001326                MOV     RKDS,R0
2543  005756  012703  177764                MOV     #-14,R3         ;INITIALIZE COUNT FOR # OF SECTORS
2544  005762  010477  173352          1$:     MOV     R4,@RKDA        ;ADDRESS THE DRIVE
2545  005766  005005                        CLR     R5              ;INITIALIZE COUNT - FOR TIMING ERROR
2546  005770  005205                2$:     INC     R5              ;KEEP TIMING FOR ERROR
2547  005772  001410                        BEQ     3$              ;REPORT ERROR IF WAITED LONG
2548  005774  011001                        MOV     @R0,R1          ;GET RKDS
2549  005776  032701  000020                BIT     #20,R1          ;IS SC=SA SET?
2550  006002  001772                        BEQ     2$              ;NO, WAIT FOR IT
2551  006004  005204                4$:     INC     R4              ;ADDRS THE NEXT SECTOR
2552  006006  005203                        INC     R3              ;ARE ALL SECTORS CHECKED FOR SC=SA
2553  006010  001364                        BNE     1$              ;NO, GO & CHECK NEXT
2554  006012  000406                        BR      TST11           ;;YES, EXIT
2555
2556  006014  110437  001162          3$:     MOVB    R4,$REG0        ;GET SECTOR ADDRESS
2557  006020  010137  001164                  MOV     R1,$REG1        ;GET RKDS
2558  006024  104015                        ERROR   15              ;COULD NOT GET SC=SA FOR THIS
                                              ;'SECTOR ADDRESS'
2559
2560  006026  000766                        BR      4$              ;GO CHK FOR THE REST
2561                                          ;
2562                                  ;;*****************************************************************
2563                                  ;*TEST 11       CHECK THAT 'R/W/S RDY' IS SET & 'SIN' IS CLEAR
2564                                  ;;*****************************************************************
2565  006030  000004                TST11:  SCOPE
2566  006032  104413                        CNT.RESET               ;GO, DO CONTROL RESET
2567  006034  013777  001350  173276        MOV     DRIVAD,@RKDA    ;ADDRESS THE DRIVE
2568  006042  005001                        CLR     R1
2569  006044  017700  173256          1$:     MOV     @RKDS,R0        ;GET RKDS
2570  006050  032700  000100                BIT     #100,R0         ;IS R/W/S RDY SET?
2571  006054  001007                        BNE     2$              ;YES, BRANCH
2572  006056  005201                 3$:    INC     R1              ;INCREASE LOOP TIME
2573  006060  001376                        BNE     3$              ;FOR DRIVE RESET OF HEADS
2574  006062  005201                        INC     R1              ;WAITED LONG ENOUGH?
2575  006064  001367                        BNE     1$              ;IF NOT LUP BAK & WAIT
2576  006066  010037  001162                MOV     R0,$REG0        ;GET RKDS
2577  006072  104016                        ERROR   16              ;R/W/S RDY SHOULD BE SET
2578  006074  032700  001000          2$:     BIT     #1000,R0        ;IS SIN CLEAR?
2579  006100  001403                        BEQ     TST12           ;;YES, EXIT
2580  006102  004737  020774                JSR     PC,GT4RG        ;GET RKCS,ER,DS,DA
2581  006106  104001                        ERROR   1               ;'SIN' SHOULD HAVE BEEN CLEAR
                                              ;IT WAS NOT CLEAR
2582
2583                                          ;NEXT TEST IS GOING TO CHECK
2584                                          ;DRIVE RESET, SIN SHOULD BE
2585                                          ;CLEARED THEN. IT WILL BE CHECKED
2586                                          ;THERE.
2587
2588                                  ;;*****************************************************************
2589                                  ;*TEST 12       CHECK 'DRIVE RESET'
2590                                          ;*THIS TEST CHECKS THE VERY BASIC DRIVE RESET LOGIC.
2591                                          ;*SINCE THE HEADS ARE AT CYLINDER 0 (GOING INTO THIS
2592                                          ;*TEST) DRIVE RESET RETRACTS THEM BACK BEYOND CYLINDER 0,
2593                                          ;*AFTER WHICH THEY ARE PUSHED FORWARD TO CYLINDER 0 AGAIN.
2594                                          ;*IN THE LATER PART OF THIS PROGRAM THERE IS A DRIVE RESET
2595                                          ;*TEST WHICH DOES THE RESET FROM LAST CYLINDER.
2596                                  ;;*****************************************************************
2597  006110  000004                TST12:  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
006114  013700  001332                  MOV     RKCS,R0
006120  005004                          CLR     R4              ;INITIALIZ COUNT - TO TIME ERROR
006122  013777  001350  173210          MOV     DRIVAD,@RKDA    ;ADDRESS THE DRIVE
006130  012710  000015                  MOV     #15,@R0         ;'DRIVE RESET', GO
006134  104412                          CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
                                                                ;IF SO, SKIP THE EROR MESSAGE.
006136  104021                          ERROR   21              ;CNTRL RDY DID NOT SET AFTER
                                                                ;SENDING CYL ADDR TO THE DRIV.
                                                                ;'ADD ACK' SHOULD HAVE COME BACK
                                                                ;FROM DRIVE, THEREUPON SETTING 'CN RDY'
006140  012705  177776          2$:     MOV     #-2,R5          ;SET UP DELAY COUNTER
006144  032777  000100  173154  6$:     BIT     #100,@RKDS      ;CHECK FOR R/W/S READY
006152  001402                          BEQ     .+6
006154  000137  006176                  JMP     3$
006160  005204                          INC     R4
006162  001370                          BNE     6$
006164  005205                          INC     R5
006166  001366                          BNE     6$
006170  004737  020774                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
006174  104026                          ERROR   26              ;R/W/S RDY DID NOT SET AFTER
                                                                ;DRIVE RESET
006176  032777  001000  173122  3$:     BIT     #1000,@RKDS     ;DID SIN SET?
006204  001403                          BEQ     5$              ;NO, BRANCH
006206  004737  020774                  JSR     PC,GT4RG        ;GO, GET RKCS,ER,DS,DA
006212  104001                          ERROR   1               ;SIN SET, AFTER A
                                                                ;DRIVE RESET.
006214  032710  140000          5$:     BIT     #140000,@R0     ;WAS 'ERR' BIT OR 'HE' BIT SET?
006220  001403                          BEQ     4$              ;NO
006222  004737  020774                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
006226  104022                          ERROR   22              ;'ERR' OR 'HE' BIT SET WHILE DOING
                                                                ;DRIVE RESET
006230  022710  000214          4$:     CMP     #214,@R0        ;DOES RKCS STILL CONTAIN THE
                                                                ;'DRIV RES' BITS
006234  001406                          BEQ     TST13           ;;YES, EXIT
006236  012737  000214  001162          MOV     #214,$REG0      ;GET EXPCTD RKCS
006244  011037  001164                  MOV     @R0,$REG1       ;GET RKCS, RECVD
006250  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.
```



```
                                ;;*************************************************************
006252  000004                  TST13:  SCOPE
006254  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
006256  104421                          TST.SIN                 ;GO CHECK IF SIN SET. IF SET
                                                                ;A DO DRIVE RESET TO CLEAR IT
006260  013700  001332                  MOV     RKCS,R0
006264  013777  001350  173046          MOV     DRIVAD,@RKDA    ;ADDRESS THE DRIVE

006272  012710  000011                  MOV     #11,@R0         ;'SEEK' GO
006276  104412                          CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
                                                                ;IF SO, SKIP THE EROR MESSAGE.
006300  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'
006302  005005                  2$:     CLR     R5
006304  032777  000100  173014          BIT     #100,@RKDS      ;DID R/W/S RDY BIT SET?
006312  001005                          BNE     3$              ;YES, BRANCH
006314  005205                          INC     R5              ;WAITED LONG ENOUGH?
006316  001372                          BNE     2$+2            ;IF NOT, LUP BAK & WAIT
006320  004737  020774                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
006324  104026                          ERROR   26              ;R/W/S RDY DID NOT SET AFTER SEEK
006326  032777  001000  172772  3$:     BIT     #1000,@RKDS     ;DID SIN SET?
006334  001403                          BEQ     6$              ;NO, BRANCH
006336  004737  020774                  JSR     PC,GT4RG        ;GO, GET RKCS,ER,DS,DA
006342  104001                          ERROR   1               ;SIN SET ON DOING SEEK
                                                                ;TO CYL 0 NOTE THIS IS THE
                                                                ;FIRST TIME THE HEADS HAVE
                                                                ;BEEN MOVED

006344  032710  140000          6$:     BIT     #140000,@R0     ;WAS 'ERR' OR 'HE' BIT SET?
006350  001403                          BEQ     4$

006352  004737  020774                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
006356  104022                          ERROR   22              ;'ERR' OR 'HE' BIT SET WHILE DOING 'SEEK'

006360  005777  172744          4$:     TST     @RKER           ;WAS ANY BIT IN RKER SET?
006364  001403                          BEQ     5$              ;NO
006366  004737  021002                  JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
006372  104023                          ERROR   23              ;RKER SHOWS AN ERROR BIT, CHECK

006374  022710  000210          5$:     CMP     #210,@R0        ;DOES RKCS STILL CONTAIN 'SEEK' FUNCTION
006400  001406                          BEQ     TST14           ;;YES, EXIT
006402  012737  000210  001162          MOV     #210,$REG0      ;GET EXPCTD RKCS
006410  011037  001164                  MOV     @R0,$REG1       ;GET RKCS RECVD
006414  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 IS THE FIRST TIME HEADS ARE MADE TO MOVE BY SEEKING
                                        ;*TO CYLINDER 2.
                                ;;*******************************************************
006416  000004                TST14:  SCOPE
006420  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
006422  104421                        TST.SIN                 ;GO CHECK IF SIN IS SET
                                                                ;IF SET DO DRV-RESET TO CLR IT
006424  004737  021504                JSR     PC,DRESET       ;MAKE SURE HEADS R ON CYL 0
006430  104026                        ERROR   26              ;R/W/S RDY DIDN'T SET
                                                                ;AFTER THE ABOVE DRV RESET
006432  005005                        CLR     R5
006434  013777  001350  172676        MOV     DRIVAD,@RKDA
006442  052777  000100  172670        BIS     #100,@RKDA      ;SEEK CYLINDER 2
006450  013701  001326                MOV     RKDS,R1
006454  012777  000011  172650        MOV     #11,@RKCS       ;SEEK, GO
006462  032711  000100        1$:     BIT     #100,@R1        ;DID R/W/S RDY CLR?
006466  001405                        BEQ     2$              ;YES, BRANCH
006470  005205                        INC     R5
006472  100373                        BPL     1$
006474  004737  021002                JSR     PC,GT3RG
006500  104025                        ERROR   25              ;R/W/S RDY WAS NOT CLEAR WHEN HEADS
                                                                ;WERE SEEKING TO CYLINDER 2
006502  004737  021436        2$:     JSR     PC,TSTRWS       ;GO, WAIT FOR R/W/S RDY TO SET
006506  104016                        ERROR   16              ;R/W/S RDY DID NOT SET AFTER SEEK
                                                                ;WAS TRIED TO CYLINDER 2 (ABOVE).
                                                                ;NOTE THIS WAS THE FIRST TIME A SEEK
                                                                ;WAS TRIED TO A CYLINDER OTHER THAN
                                                                ;0.
                                ;;*******************************************************
                                ;*TEST 15       CHECK 'WRITE' FORMAT FUNCTION-CYLINDER 0, SECTOR 0
                                        ;*THIS TEST CHECKS THE LOGIC INVOLVED IN THE WRITE FMT
                                        ;*FUNCTION. ON ISSUING A WRT FMT, THE FOLLOWING IS CHECKED
                                        ;*1)  CNTRL RDY WAS CLEARED AS GO WAS SET.
                                        ;*2)  CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION OF FUNCTION
```



```
                                        ;*3)  IF 'HE' OR 'ERR' BIT SET?
                                        ;*4)  IF RKDA INCREMENTED CORRECTLY FROM 0 TO 1?
                                        ;*5)  IF RKWC OVERFLOWED CORRECTLY TO 0?
                                        ;*6)  IF RKBA INCREMENTED CORRECTLY BY 2?
                                        ;*7)  IF ANY BIT IN RKER SET?
                                        ;*8)  IF THE 'WRT FMT' FUNCTION BITS ARE STILL IN THE RKCS?
                                        ;*NOTE THAT ONE WORD '125252' WAS WRITTEN ON SECTOR
                                        ;*0 & IT WILL BE CHECKED IN THE NEXT TESTS.
                                ;;*******************************************************
006510  000004                TST15:  SCOPE
006512  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
006514  104421                        TST.SIN                 ;GO CHECK IF SIN IS SET
                                                                ;IF SET, DO DRIVE RESET TO CLR IT
006516  012703  033342                MOV     #OUTBUF,R3
                                                                ;THIS CODE SETS UP A 256 WORD BUFFER
                                                                ;WHICH WILL BE USED TO WRITE 1 SECTOR
                                                                ;ON THE DISK
                                                                ;1ST WORD       000001
                                                                ;2ND WORD       177777  2'S COMPLEMENT
                                                                ;3RD WORD       000002  OF ABOVE
                                                                ;4TH WORD       177776
                                                                ;  ..
                                                                ;253RD WORD     000177
                                                                ;254TH WORD     177601
                                                                ;255TH WORD     000000
                                                                ;256TH WORD     125252
006522  012700  000001                MOV     #1,R0           ;SET COUNT
006526  010023        9$:     MOV     R0,(R3)+        ;SET UP DATA WORDS
006530  010013                MOV     R0,(R3)
006532  005423                NEG     (R3)+
006534  005200                INC     R0
006536  022700  000200                CMP     #200,R0         ;DONE?
006542  001371                BNE     9$
006544  005023                CLR     (R3)+           ;SET 255TH WORD TO 0
006546  012713  125252                MOV     #125252,@R3     ;SET 256TH WORD
006552  012703  033342                MOV     #OUTBUF,R3      ;RESET POINTER TO OUTBUF
006556  013701  001332                MOV     RKCS,R1
006562  013702  001336                MOV     RKBA,R2
006566  010312                MOV     R3,@R2          ;FROM HERE-SET UP CURRENT ADDRESS
006570  012777  177400  172536        MOV     #-400,@RKWC     ;SET UP WORD COUNT 400 WORDS
006576  013777  001350  172534        MOV     DRIVAD,@RKDA    ;SET UP DISK ADDRS, SECTOR 0, CYLINDER 0
006604  012711  002003                MOV     #2003,@R1       ;WRITE FORMAT, GO
006610  105711        1$:     TSTB    @R1             ;WAS 'CNTRL RDY' CLEARED AS GO WAS SET?
```

```
2822  006612  100003                  BPL     2$              ;YES, BRANCH
2823  006614  004737  021002          JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
2824  006620  104030                  ERROR   30              ;'CNTRL RDY' DIDN'T CLEAR AS GO
2825                                                          ;WAS SET TO 'WRITE FORMAT'
2826  006622  005000          2$:     CLR     R0
2827  006624  105711                  TSTB    @R1             ;WAS 'CNTRL RDY' SET ON COMPLETION OF WRITE?
2828  006626  100411                  BMI     3$              ;YES, BRANCH
2829  006630  005200                  INC     R0              ;NO, HAVE U WAITED LONG ENOUGH?
2830  006632  001374                  BNE     2$+2            ;IF NOT, LOOP BACK & WAIT
2831                                                          ;IF YES, REPORT ERROR
2832  006634  004737  020774          JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
2833  006640  013737  001350  001202  MOV     DRIVAD,$REG10
2834  006646  104416                  BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
2835                                                          ;$REG10 INTO DR #,CYL,SUR,SEC BITS
2836  006650  104031                  ERROR   31              ;'CNTRL RDY' DIDN'T SET ON COMPLETION
2837                                                          ;OF WRITE FORMAT
2838                                                          ;WRT FMT WAS DONE STARTING AT <DSK-ADRES>
2839                                                          ;INDICATED IN EROR MSGE.
2840  006652  004737  021234  3$:     JSR     PC,CHKHE        ;GO CHECK IF 'HE' OR 'ERR' BIT SET.
2841                                                          ;IF YES, SAVE RKCS, ER, DS, DA.
2842                                                          ;RETURN HERE IF ERROR.
2843  006656  104032                  ERROR   32              ;'HE' OR 'ERR' BIT SET WHILE DOING
2844                                                          ;A WRITE FORMAT
2845                                                          ;WRT FMT WAS DONE STARTING AT <DSK-ADRES>
2846                                                          ;INDICATED IN EROR MSGE.
2847  006660  004737  021262  4$:     JSR     PC,CHKDA        ;GO CHECK IF RKDA INCREMENTED CORRECTLY
2848                                                          ;IF NOT, RETURN HERE.
2849  006664  104033                  ERROR   33              ;RKDA SHOULD HAVE INCREMENTED BY
2850                                                          ;1 SECTOR, IT DID NOT
2851  006666  004737  021316  5$:     JSR     PC,CHKWC        ;CHECK IF WORD COUNT OVERFLOWED, IF
2852                                                          ;NOT RETURN HERE.
2853  006672  104034                  ERROR   34              ;RKWC DID NOT OVERFLOW TO 0, AFTER
2854                                                          ;XFER ON WRITE FORMAT
2855  006674  022712  034342  6$:     CMP     #OUTBUF+1000,@R2        ;DID RKBA INCREMENT CORRECTLY?
2856  006700  001406                  BEQ     7$              ;YES, BRANCH
2857  006702  012737  034342  001162  MOV     #OUTBUF+1000,$REG0 ;GET EXPCTD RKBA
2858  006710  011237  001164          MOV     @R2,$REG1       ;GET ACTUAL RKBA
2859  006714  104035                  ERROR   35              ;RKBA DIDN'T INCREMENT BY 1000 AFTER
2860                                                          ;WRITE FORMAT OF 400 WORDS
2861  006716  004737  021342  7$:     JSR     PC,CHKER        ;CHECK IOF ANY BIT IN RKER SET,
2862                                                          ;IF YES RETURN HERE.
2863  006722  104036                  ERROR   36              ;RKER BIT SET ON DOING 1 WORD
2864                                                          ;WRITE FORMAT
2865  006724  022711  002202  8$:     CMP     #2202,@R1       ;DOES RKCS STILL HAVE 'WRT FMT' BITS?
2866  006730  001406                  BEQ     TST16           ;;YES, EXIT
2867  006732  012737  002202  001162  MOV     #2202,$REG0     ;GET EXPCTD RKCS
2868  006740  011137  001164          MOV     @R1,$REG1       ;GET ACTUAL RKCS
2869  006744  104024                  ERROR   24              ;RKCS DIDN'T CONTAIN 'WRT FMT' BITS
2870                                                          ;AFTER THE FUNCTION WAS COMPLETED
2871                                  ;
2872                                  ;;*************************************************************
2873                                  ;*TEST 16       CHECK 'READ FORMAT' FUNCTION-CYLINDER 0, SECTOR 0
2874                                          ;*THIS TEST CHECKS THE LOGIC INVOLVED IN THE WRITE FMT
2875                                          ;*FUNCTION. ON ISSUING A WRT FMT, THE FOLLOWING IS CHECKED
2876                                          ;*1)  CNTRL RDY WAS CLEARED AS GO WAS SET.
2877                                          ;*2)  CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION OF FUNCTION
```



```
2878                                          ;*3)  IF 'HE' OR 'ERR' BIT SET?
2879                                          ;*4)  IF RKDA INCREMENTED CORRECTLY FROM 0 TO 1?
2880                                          ;*5)  IF RKWC OVERFLOWED CORRECTLY TO 0?
2881                                          ;*6)  IF RKBA INCREMENTED CORRECTLY BY 2?
2882                                          ;*7)  IF ANY BIT IN RKER SET?
2883                                          ;*8)  IF THE CORRECT HEADER WAS RECEIVED?
2884                                          ;*9)  FOR RK11C, AFTER RD FMT RKDB CONTAINS THE CHECKSUM
2885                                          ;*FOR THAT SECTOR. (125252 IN THIS CASE, BECAUSE THE
2886                                          ;*FIRST WORD IN SEC 0 WAS WRITTEN AS 125252 IN
2887                                          ;*THE PREVIOUS TEST)
2888                                          ;*10)  FOR RK11D, AFTER RD FMT RKDB SHOULD CONTAIN
2889                                          ;*A ZERO
2890                                          ;*11)  IF THE RD FMT FUNCTION BITS ARE STILL IN
2891                                          ;*THE RKCS?
2892                                  ;;*************************************************************
2893  006746  000004          TST16:  SCOPE
2894  006750  005000                  CLR     R0
2895  006752  104413                  CNT.RESET               ;GO, DO CONTROL RESET
2896                                                          ;THIS IS A CALL FOR THE 'CNTRL-
2897                                                          ;RESET' ROUTINE. A CONTROL RESET IS
2898                                                          ;ISSUED AND AFTER A CERTAIN TIME
2899                                                          ;IF THE 'CNTRL RDY' DOES NOT SET
2900                                                          ;AN ERROR IS REPORTED. NOTE THAT
2901                                                          ;THE PC IN ERROR MESSAGE IS THE
2902                                                          ;PC WHERE 'CNT.RESET' IS LOCATED.
2903                                                          ;THIS IS A VERY BASIC ERR & IF IT
2904                                                          ;OCCURS GO BACK TO TEST 10
2905  006754  104421                  TST.SIN                 ;GO CHECK IF SIN IS SET
2906                                                          ;IF SET, DO DRIVE RESET TO CLR IT
2907  006756  013701  001332          MOV     RKCS,R1
2908  006762  013702  001336          MOV     RKBA,R2
2909  006766  012703  033342          MOV     #OUTBUF,R3
2910  006772  010312                  MOV     R3,@R2          ;SETUP ADRS WHERE HEADER WORD IS TO BE
2911                                                          ;X-FERRED
2912  006774  012777  177777  172332  MOV     #-1,@RKWC       ;SET UP WORD COUNT
2913  007002  013777  001350  172330  MOV     DRIVAD,@RKDA    ;SET UP DISK ADRS, SECTOR 0, CYLINDER 0
2914  007010  012711  002005          MOV     #2005,@R1       ;READ FORMAT, GO
2915
2916  007014  105711          1$:     TSTB    @R1             ;WAS 'CNTRL RDY' CLEARED AS GO WAS SET?
2917  007016  100003                  BPL     2$              ;YES, BRANCH
2918  007020  004737  021002          JSR     PC,GT3RG        ;GO, GET RKCS, RKER
2919  007024  104030                  ERROR   30              ;CNTRL RDY DIDN'T CLEAR AS GO WAS
2920                                                          ;SET TO 'READ FORMAT'
2921  007026  005000          2$:     CLR     R0
2922  007030  105711                  TSTB    @R1             ;WAS 'CNTRL RDY' SET ON COMPLETION OF
2923                                                          ;TRANSFER
2924  007032  100411                  BMI     3$              ;YES, BRANCH
2925  007034  005200                  INC     R0              ;NO, HAVE U WAITED LONG ENOUGH?
2926  007036  001374                  BNE     2$+2            ;IF NOT, LOOP BACK & WAIT
2927                                                          ;IF YES, REPORT ERROR
2928  007040  004737  020774          JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
2929  007044  013737  001350  001202  MOV     DRIVAD,$REG10
2930  007052  104416                  BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
2931                                                          ;$REG10 INTO DR #,CYL,SUR,SEC BITS
2932  007054  104045                  ERROR   45              ;'CNTRL RDY' DIDN'T SET ON COMPLETION
2933                                                          ;OF READ FORMAT
```

```
                                                        ;READ FMT WAS DONE STARTING AT <DSK-ADRES>
                                                        ;INDICATED IN EROR MESGE
007056  004737  021234          3$:     JSR     PC,CHKHE        ;CHECK IF 'ERR' OR 'HE' BIT SET, IF
                                                        ;YES RETURN HERE.
007062  104046                          ERROR   46              ;'HE' OR 'ERR' BIT SET WHILE
                                                        ;DOING A 'READ FORMAT'
                                                        ;READ FMT WAS DONE STARTING AT <DSK-ADRES>
                                                        ;INDICATED IN EROR MESGE
007064  004737  021262          4$:     JSR     PC,CHKDA        ;CHECK IF RKDA INCREMENTED CORRECTLY
                                                        ;IF NOT, RETURN HERE.
007070  104040                          ERROR   40              ;RKDA SHOULD HAVE INCREMENTED
                                                        ;BY 1 SECTOR, IT DID NOT
007072  004737  021316          5$:     JSR     PC,CHKWC        ;CHECK IF RKWC OVERFLOWED TO 0, IF
                                                        ;NOT RETURN HERE.
007076  104041                          ERROR   41              ;RKWC DID NOT OVERFLOW TO 0
                                                        ;AFTER XFER ON READ FORMAT
007100  022712  033344          6$:     CMP     #OUTBUF+2,@R2   ;DID RKBA INCREMENT TO NXT WORD ADDRS?
007104  001406                          BEQ     7$              ;YES, BRANCH
007106  012737  033344  001162          MOV     #OUTBUF+2,$REG0 ;GET EXPCTD RKBA
007114  011237  001164                  MOV     @R2,$REG1       ;GET ACTUAL RKBA
007120  104042                          ERROR   42              ;RKBA DIDN'T INCREMENT BY 2 AFTER
                                                        ;'READ FORMAT' OF 1 WORD
007122  004737  021342          7$:     JSR     PC,CHKER        ;CHECK IF ANY BIT IN RKER SET, IF
                                                        ;YES RETURN HERE.
007126  104036                          ERROR   36              ;RKER BIT SET ON DOING
                                                        ;1 WORD READ FORMAT
007130  005713                  8$:     TST     @R3             ;DOES OUTBUF CONTAIN THE HEADER
                                                        ;WORD-0
007132  001407                          BEQ     9$              ;YES, BRANCH
007134  005037  001162                  CLR     $REG0           ;GET SECTOR NO.
007140  005037  001164                  CLR     $REG1           ;EXPCTD HEADER
007144  011337  001166                  MOV     @R3,$REG2       ;GET HEADER RECVD
007150  104043                          ERROR   43              ;CORRECT HEADER WORD-0-WAS
                                                        ;NOT RECEIVED ON READ FORMAT
007152  022711  002204          9$:     CMP     #2204,@R1       ;DOES RKCS HAVE THE 'RDFMT' BITS?
007156  001406                          BEQ     TST17           ;;YES, BRANCH
007160  012737  002204  001162          MOV     #2204,$REG0     ;GET EXPCTD RKCS
007166  011137  001164                  MOV     @R1,$REG1       ;GET ACTUAL RKCS
007172  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 'HE' 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
                                ;;*************************************************************
007174  000004                  TST17:  SCOPE
007176  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
007200  104421                          TST.SIN                 ;GO CHECK IF SIN IS SET
                                                        ;IF SET, DO DRIVE RESET TO CLR IT
007202  013701  001332                  MOV     RKCS,R1
007206  005000                          CLR     R0
007210  013702  001336                  MOV     RKBA,R2
007214  012703  033342                  MOV     #OUTBUF,R3
007220  010312                          MOV     R3,@R2          ;SET UP ADDRS WHERE DATA WORD IS
                                                        ;TO BE X-FERRED
007222  012777  177400  172104          MOV     #-400,@RKWC     ;SET UP WORD COUNT
007230  013777  001350  172102          MOV     DRIVAD,@RKDA    ;SET UP DISK ADRS, SECTOR 0, CYLINDER 0
007236  012711  000005                  MOV     #5,@R1          ;READ, GO

007242  105711          1$:     TSTB    @R1             ;WAS 'CNTRL RDY' CLEARED AS GO WAS SET?
007244  100003                  BPL     2$              ;YES, BRANCH
007246  004737  021002                  JSR     PC,GT3RG        ;GO, GET RKCS, ER
007252  104030                          ERROR   30              ;CNTRL RDY DID NOT CLEAR AS GO
                                                        ;WAS SET TO 'READ'
007254  005000          2$:     CLR     R0
007256  105711                  TSTB    @R1             ;WAS CNTRL RDY SET ON COMPLETION
                                                        ;OF TRANSFER?
007260  100411                  BMI     3$              ;YES, BRANCH
007262  005200                  INC     R0              ;NO, HAVE U WAITED LONG ENOUGH?
007264  001374                  BNE     2$+2            ;IF NOT, LOOP BACK & WAIT
                                                        ;IF YES, REPORT ERROR
007266  004737  020774                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
007272  013737  001350  001202          MOV     DRIVAD,$REG10
007300  104416                          BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                        ;$REG10 INTO DR #,CYL,SUR,SEC BITS
007302  104045                          ERROR   45              ;CNTRL RDY DID NOT SET ON
                                                        ;COMPLET.ON OF READ
                                                        ;READ WAS DONE STARTING AT <DSK-ADRES>
                                                        ;INDICATED IN EROR MESGE

007304  004737  021234          3$:     JSR     PC,CHKHE        ;CHECK IF 'ERR' OR 'HE' BIT IS SET
                                                        ;IF YES, RETURN HERE
007310  104046                          ERROR   46              ;'HE' OR 'ERR' BIT SET WHILE
                                                        ;DOING A READ.
                                                        ;READ WAS DONE STARTING AT <DSK-ADRES>
                                                        ;INDICATED IN EROR MESGE
007312  004737  021262          4$:     JSR     PC,CHKDA        ;CHECK IF RKDA INCREMENTED CORRECTLY,
```

```
                                                        ;IF NOT RETURN HERE.
007316  104040                          ERROR   40      ;RKDA DID NOT INCREMENT
                                                        ;BY 1 (SECTOR)
007320  004737  021316          5$:     JSR     PC,CHKWC        ;CHECK IF RKWC OVERFLOWED TO 0,
                                                        ;IF NOT RETURN HERE.
007324  104041                          ERROR   41      ;RKWC DID NOT OVERFLOW TO 0,
                                                        ;AFTER X-FER ON READ
007326  022712  034342          6$:     CMP     #OUTBUF+1000,@R2        ;DID RKBA INCREMENT CORRECTLY?
007332  001406                          BEQ     7$              ;YES, BRANCH
007334  012737  034342  001162          MOV     #OUTBUF+1000,$REG0      ;GET EXPCTD RKBA
007342  011237  001164                  MOV     @R2,$REG1       ;GET ACTUAL RKBA
007346  104042                          ERROR   42      ;RKBA DID NOT INCREMENT BY 2
                                                        ;AFTER 'READ' OF 1 WORD
007350  004737  021342          7$:     JSR     PC,CHKER        ;CHECK IF ANY BIT IN RKER SET,
                                                        ;IF YES RETURN HERE.
007354  104036                          ERROR   36      ;RKER BIT SET ON DOING 1
                                                        ;WORD 'READ'
007356  022713  000001          8$:     CMP     #1,@R3          ;DOES OUTBUF CONTAIN THE RIGHT
                                                        ;DATA WORD
007362  001411                          BEQ     9$              ;YES BRANCH
007364  012737  000001  001162          MOV     #1,$REG0        ;GET EXPCTD DATA WORD
007372  011337  001164                  MOV     (R3),$REG1      ;GET RECVD DATA WORD
007376  013737  001350  001166          MOV     DRIVAD,$REG2    ;GET DISK ADRS FROM WHICH READ WAS DONE
007404  104044                          ERROR   44      ;DID NOT READ THE CORRECT
                                                        ;DATA WORD--FROM DISK ADRES,

                                                        ;SEC 0, CYL 0, SUR 0

                                                        ;AFTER 1 SECTOR READ RKDB CONTAINS
                                                        ;FOR RK11C
                                                        ;THE CHECKSUM FOR THAT SECTOR
                                                        ;FOR RK11D
                                                        ;THE LAST WORD TRANSFERRED TO MEMORY
                                                        ;
                                                        ;IT SO HAPPENS THAT WITH THE SECTOR
                                                        ;THAT WAS READ, RKDB CONTAINS THE
                                                        ;SAME INFORMATION FOR BOTH RK11C
                                                        ;AND RK11D
007406  022777  125252  171726  9$:     CMP     #125252,@RKDB   ;DOES RKDB CONTAIN THE EXPCTD WORD?
007414  001407                          BEQ     10$             ;YES, BRANCH
007416  012737  125252  001162          MOV     #125252,$REG0   ;GET EXPCTD RKDB
007424  017737  171712  001164          MOV     @RKDB,$REG1     ;GET RECVD RKDB
007432  104037                          ERROR   37      ;RKDB DOESD NOT CONTAIN THE
                                                        ;EXPCTD WORD AFTER A READ OF SEC 0
                                                        ;CYL 0
007434  022711  000204          10$:    CMP     #204,@R1        ;DOES RKCS HAVE THE 'READ' BITS?
007440  001406                          BEQ     11$             ;YES , BRANCH
007442  012737  000204  001162          MOV     #204,$REG0      ;GET EXPCTD RKCS
007450  011137  001164                  MOV     @R1,$REG1       ;GET RECVD RKCS
007454  104024                          ERROR   24      ;RKCS DID NOT CONTAIN 'READ'
                                                        ;FUNCTION BITS AFTER OPERATION
                                                        ;WAS COMPLETED
007456  104413                  11$:    CNT.RESET               ;GO DO CONTROL RESET
007460  005777  171656                  TST     @RKDB           ;DID CONTROL RESET CLEAR RKDB?
007464  001407                          BEQ     TST20           ;;YES, EXIT
007466  013737  001342  001164          MOV     RKDB,$REG1      ;GET ADRES OF RKDB
```



```
007474  017737  171642  001164          MOV     @RKDB,$REG1     ;GET CONTENTS OF RKDB
007502  104102                          ERROR   102     ;CONTROL RESET DIDN'T CLR RKDB

                                ;;**************************************************************
                                ;*TEST 20        CHECK 'WRITE FORMAT' -CYLINDER 0, SECTOR 0-13
                                        ;*THIS TEST GOES ONE STEP FURTHE. & PERFORMS A WRT
                                        ;*FMT ON CYLINDER 0 & CHECKS THE FOLLOWING
                                        ;*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)  IF THE RKDB IS CLEAR?
                                        ;*WRT FMT IS DONE ONE SECTOR AT A TIME
                                        ;*THE FIRST WORD OF EVERY SECTOR IS WRITTEN AS A
                                        ;*PSUEDO-HEADER CONSISTING OF DRIVE #, CYLINDER #, SURFACE
                                        ;*& SECTOR #. THIS WILL BE READ & CHECKED IN THE FOLLOWING TEST.
                                ;;**************************************************************
007504  000004                  TST20:  SCOPE
007506  013703  001332                  MOV     RKCS,R3
007512  012702  177764                  MOV     #-14,R2         ;SET UP COUNT FOR 12 SECTORS
007516  013704  001340                  MOV     RKDA,R4
007522  013701  001350                  MOV     DRIVAD,R1       ;GET DRIVE ADDRESS
007526  010105                          MOV     R1,R5           ;STORE IT
007530  005205                          INC     R5
007532  012737  007540  001110          MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
                                                        ;ON     ERROR (SW 9)
007540  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
007542  104421                          TST.SIN                 ;GO CHECK IF SIN IS SET
                                                        ;IF SET, DO DRIVE RESET TO CLR IT
007544  005000                          CLR     R0
007546  010137  033342                  MOV     R1,OUTBUF       ;THIS WORD TO BE X-FERRED. FIRST
                                                        ;WORD OF EACH SECTOR WILL BE THE
                                                        ;ACTUAL DRIVE-ADDRS CONSISTING OF
                                                        ;DRIVE NO, CYL ADDRS, SURFACE
                                                        ;SECTOR NO.
007552  012777  033342  171556          MOV     #OUTBUF,@RKBA   ;ADRS FROM WHICH DATA WORD IS TO
                                                        ;X-FERRED
007560  012777  177777  171546          MOV     #-1,@RKWC       ;SET UP WORD COUNT
007566  010114                          MOV     R1,@R4  ;ADDRS THE DRIVE, CYL 0, & CORRECT SECTOR
007570  012713  002003                  MOV     #2003,@R3       ;WRITE FORMAT, GO

007574  105777  171532          2$:     TSTB    @RKCS           ;DID 'CNTRL RDY' SET?
007600  100410                          BMI     3$              ;YES, BRANCH
007602  005200                          INC     R0              ;NO, HAVE U WAITED LONG?
007604  001373                          BNE     2$              ;IF NOT, LOOP BACK & WAIT
                                                        ;IF YES, REPORT ERROR
007606  004737  020774                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
```

```
                                ;GET DISK ADRS (UNIT,CYL,SUR,SEC) TO WHICH
                                ;WRITE FORMAT WAS DONE
                                ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                ;$REG10 INTO DR #,CYL,SUR,SEC BITS
                                ;'CNTRL RDY' DID NOT SET ON COMPLETION
                                ;OF 'WRITE FORMAT'
                                ;WRT FMT WAS DONE STARTING AT <DSK-ADRES>
                                ;INDICATED IN EROR MSGE.
                                ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
                                ;IF YES RETURN HERE.
                                ;'HE' OR 'ERR' BIT SET WHILE DOING
                                ;WRITE FORMAT ON CYLINDER 0,
                                ;SECTOR IN ERROR IS AS SHOWN IN
                                ;DISK-ADRES BITS 0-3
                                ;WRT FMT WAS DONE STARTING AT <DSK-ADRES>
                                ;INDICATED IN EROR MSGE.

                                ;CHECK IF RKDA INCREMENTED CORRECTLY?

                                ;RKDA DID NOT INCREMENT CORRECT
                                ;AFTER 1 WORD 'WRITE FORMAT' ON
                                ;CYLINDER 0, SECTOR IN ERROR IS 1
                                ;LESS THAN THAT SHOWN IN EXPCTD RKDA
                                ;CHECK THAT RKDB DOES CONTAIN A 0
                                ;AFTER WRT BECAUSE LAST WORD WRITTEN
                                ;WAS SERIALLY SHIFTED OUT TO THE DISK
                                ;YES, BRANCH
                                ;THIS IS WHAT RKDB SHOULD CONTAIN
                                ;GET RKDB
                                ;RKDB SHOULD BE 0 AFTER WRT SINCE THE
                                ;LAST WORD WRITTEN WAS SERIALLY SHIFTED
                                ;OUT OF RKDB
                                ;INCREMENT DRIVE ADDRS TO NXT SECTOR
                                ;
                                ;R U GOING TO CHECK THE LAST SECTOR?
                                ;IF NOT,BRANCH
                                ;IF YES,INCREMENT R5 CORRECTLY TO 'EXPCTD RKDA'
                                ;AFTER HAVING CHECKED THE LAST SECTOR
                                ;HAVE U FORMATTED ALL 12 SECTORS?
                                ;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.
;;*****************************************************************
TST21:  SCOPE
        CLR     R5
        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                 ;GO CHECK IF SIN IS SET
                                ;IS SET, DO DRIVE RESET TO CLR IT
        MOV     RKCS,R1
        MOV     #-14,R0         ;SET UP COUNT FOR 12 SECTORS
        MOV     RKDA,R2
        MOV     DRIVAD,@R2      ;ADDRESS THE DRIVE
        MOV     #OUTBUF,R4
        MOV     R4,@RKBA        ;ADRS TO WHICH X-FER DATA FROM DSK
        MOV     #-14,@RKWC      ;SET UP WORD COUNT FOR 12 HEADERS TO BREAD
        MOV     #2005,@RKCS     ;READ FORMAT, GO

1$:     TSTB    @RKCS           ;DID CNTRL RDY SET ON COMPLETION?
        BMI     2$              ;YES, BRANCH
        INC     R5              ;NO, WAIT FOR IT TO SET
        BNE     1$              ;IF WAITED LONG ENOUGH REPORT
                                ;ERROR, OTHERWISE LOOP BACK & WAIT
        JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
        MOV     DRIVAD,$REG10
        BRKDA4                  ;GO TO 'BDA4' & BREAK CONTENTS OF
                                ;$REG10 INTO DR#,CYL,SUR,SEC BITS
        ERROR   45              ;CNTRL RDY DID NOT SET ON COMPLETION
                                ;OF READ FORMAT-OF CYLINDER 0,
                                ;SECTORS 0-13
                                ;READ FMT WAS DONE STARTING AT <DSK-ADRES>
                                ;INDICATED IN EROR MESGE

2$:     JSR     PC,CHKHE        ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
                                ;IF YES RETURN HERE.
        ERROR   46              ;'ERR' OR 'HE' BIT SET ON DOING
                                ;READ FMT-OF CYLINDER 0, SEC 0-13
                                ;READ FMT WAS DONE STARTING AT <DSK-ADRES>
                                ;INDICATED IN EROR MESGE

3$:     MOV     DRIVAD,R5
        ADD     #20,R5          ;RKDA SHOULD HAVE INCREMENTD TO (R2)

        JSR     PC,CHKDA1       ;CHECK IF RKDA INCREMENTED CORRECTLY,
                                ;IF NOT, RETURN HERE
        ERROR   40              ;RKDA DID NOT INCREMENT BY 12
                                ;AFTER A 'RD FMT' OF 12 HEADERS OF
                                ;CYLINDER 0, SECTORS 0-13
```

```
                                                        ;RKBA SHOULD INCREMENT BY 24 BYTES
                                                        ;AT THE END OF X-FER
010032  022777  033372  171276  4$:     CMP     #OUTBUF+30,@RKBA ;DID RKBA INCREMENT CORRECTLY?
010040  001407                          BEQ     5$              ;YES, BRANCH
010042  012737  033372  001162          MOV     #OUTBUF+30,$REG0        ;GET EXPCTD RKBA
010050  017737  171262  001164          MOV     @RKBA,$REG1     ;GET ACTUAL RKBA
010056  104042                          ERROR   42              ;RKBA DID NOT INCREMENT CORRECTLY
                                                        ;AFTER READ FORMAT OF 12 HEADERS

010060  004737  021316          5$:     JSR     PC,CHKWC        ;GO CHECK IF RKWC OVERFLOWED TO 0
                                                        ;IF NOT RETURN HERE.
010064  104041                          ERROR   41              ;RKWC DID NOT OVERFLOW TO 0
                                                        ;AFTER 'RD FMT' OF 12 HEADERS
                                                        ;OF CYLINDER 0

010066  005724                  6$:     TST     (R4)+           ;WAS THE CORRECT HEADER RECIEVED?
010070  001413                          BEQ     7$              ;YES, BRANCH
010072  010037  001162                  MOV     R0,$REG0        ;GET SECTOR FOR WHICH THE HEADER
010076  062737  000014  001162          ADD     #14,$REG0       ;COULD NOT BE READ CORRECT
010104  005037  001164                  CLR     $REG1           ;EXPCTD HEADER-0, FOR CYL 0
010110  014437  001166                  MOV     -(R4),$REG2     ;GET WRONG HEADER RECVD
010114  104043                          ERROR   43              ;HEADER WAS NOT READ RIGHT FOR
                                                        ;SECTOR (AS IN ER MSGE), & CYL 0

010116  005724                          TST     (R4)+           ;WAS THE CORRECT HEADER RECVD?
010120  005200                  7$:     INC     R0              ;YES, HAVE U CHECKED FOR ALL 12 SECTORS?
010122  001361                          BNE     6$              ;IF NOT, LOOP BACK & CHK HDR FRM NXT SECTR

010124  004737  021342                  JSR     PC,CHKER        ;CHECK IF ANY BIT IN RKER IS SET,
                                                        ;IF YES, RETURN HERE.
010130  104036                          ERROR   36              ;RKER BIT SET ON DOING RD FMT
                                                        ;OF CYL 0, SECTORS 0-13

010132  022711  002204          8$:     CMP     #2204,@R1       ;DOES RKCS STILL CONTAIN FUNCTION BITS?
010136  001406                          BEQ     TST22           ;;YES, EXIT
010140  012737  002204  001164          MOV     #2204,$REG0     ;GET EXPCTD RKCS
010146  011137  001164                  MOV     @R1,$REG1       ;GET ACTUAL RKCS
010152  104024                          ERROR   24              ;RKCS DID NOT CONTAIN 'RD FMT'
                                                        ;FUNCTION BITS ON COMPETION OF
                                                        ;THE FUNCTION


                                ;;**************************************************************
                                ;*TEST 22       CHECK 'READ',CYLINDER 0, SECTORS 0 TO 13
                                        ;*THIS TEST PERFORMS A READ OF ALL THE SECTORS OF CYLINDER 0
                                        ;*& CHECKS THE FOLLOWING
                                        ;*1) CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
                                        ;*OF THE FUNCTION
                                        ;*2) IF 'HE' OR 'ERR' BIT SET?
                                        ;*3) IF THE CORRECT PSUEDO-HEADER (FIRST WORD OF EVERY)
                                        ;*SECTOR, WRITTEN IN A PREVIOUS TEST) WAS RECEIVED.
                                        ;*4) IF RKDS CONTAINS THE CORRECT WORD.
                                        ;*4) IF RKDA INCREMENTED CORRECTLY.
                                        ;*5) IF REST OF THE (377) WORDS IN EACH SECTOR ARE '0' , NOTE
                                        ;*PREVIOUSLY ONE WORD WAS WRITTEN PER SECTOR.
                                        ;*6) IF RKCS STILL CONTAINS THE 'READ' FUNCTION BITS
                                        ;*7) IF CONTROL RESET CLEARS RKDB.
                                        ;*  IF TESTING IS BEING DONE ON A SIMULATOR ONLY LAST SECTOR(13)
                                        ;*IS READ BECAUSE THE SIMULATOR CAN STORE ONLY 1 SECTOR (256 WORDS).
```



```
                                        ;*HENCE ONLY THE DATA WRITTEN LAST CAN BE READ BACK.
                                ;;**************************************************************
000004                  TST22:  SCOPE
010156  012737  010230  001110          MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
                                                        ;ON       ERROR (SW 9)
010164  013703  001332                  MOV     RKCS,R3
010170  013701  001350                  MOV     DRIVAD,R1
010174  010105                          MOV     R1,R5
010176  012704  033342                  MOV     #OUTBUF,R4
010202  005737  001344                  TST     SIMUL           ;TESTING ON SIMULATOR?
010206  001405                          BEQ     9$              ;NO, BRANCH
                                                        ;IF TESTING ON SIMULATOR READ
                                                        ;SECTOR  13  ONLY
010210  052701  000013                  BIS     #13,R1          ;SET BITS FOR SEC 13
010214  052705  000020                  BIS     #20,R5          ;RKDA SHOULD INCRMNT TO THIS AFTER READ
010220  000403                          BR      1$
010222  012702  177764          9$:     MOV     #-14,R2         ;SET COUNT FOR 12 SECTORS
010226  005205                          INC     R5              ;RKDA SHOULD INCREMENT TO
                                                        ;THIS AFTER 1 SECTOR  READ
010230  104413                  1$:     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
010232  104421                          TST.SIN                 ;GO CHECK IF SIN IS SET
                                                        ;IF SET, DO DRIVE RESET TO CLR IT
010234  010177  171100                  MOV     R1,@RKDA        ;ADDRESS THE DRIVE
010240  010477  171072                  MOV     R4,@RKBA        ;ADRS TO WHICH X-FER DATA FROM DISK
010244  012777  177400  171062          MOV     #-400,@RKWC     ;SETUP WORD COUNT
010252  012713  000005                  MOV     #5,@R3          ;READ,GO

010256  005000                          CLR     R0
010260  105713                  2$:     TSTB    @R3             ;DID CNTRL RDY SET ON COMPETION?
010262  100410                          BMI     3$              ;YES, BRANCH
010264  005200                          INC     R0              ;NO, WAIT FOR IT TO SET
010266  001374                          BNE     2$              ;IF WAITED LONG ENOUGH, REPORT
                                                        ;ERROR, OTHERWISE LOOP BAK & WAIT
010270  004737  020774                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
010274  010137  001202                  MOV     R1,$REG10       ;GET SECTOR ADDRES WHERE ERROR OCCURED
010300  104416                          BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                        ;$REG10 INTO DR #,CYL,SUR,SEC BITS
010302  104045                          ERROR   45              ;CNTRL RDY DID NOT SET ON COMPLETION
                                                        ;OF READ OF CYLINDER 0, SECTOR
                                                        ;AS SHOWN IN <DSK-ADRES>
                                                        ;READ WAS DONE STARTING AT <DSK-ADRES>
                                                        ;INDICATED IN ERROR MESGE
010304  004737  021226          3$:     JSR     PC,CHKHE1       ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
                                                        ;IF YES RETURN HERE.
010310  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 ERROR MESGE
010312  020114                  4$:   CMP     R1,(R4)   ;WAS THE DATA WORD RECVD, CORRECT?
                                                        ;THE FIRST DATA WORD OF EACH SECTOR
                                                        ;IS AN ADRS WORD COMRISING OF DRIVE NO,
                                                        ;CYLINDER ADRS, SUR, SECTOR ADRS
010314  001407                        BEQ     5$
010316  010137  001162              MOV     R1,$REG0    ;GET EXPCTD DATA WORD FROM DISK
010322  011437  001164              MOV     (R4),$REG1  ;GET THE DATA WORD RECVD
010326  010137  001166              MOV     R1,$REG2    ;GET DISK ADRES
010332  104044                      ERROR   44          ;DID NOT RECIEVE CORRECT DATA WORD ON
                                                        ;READ, OF CYLINDER 0, SECTOR AS SHOWN IN 'DSK
                                                        ;ADRES' OF EXPCTD DATA WORD
010334  004737  021270          5$:   JSR     PC,CHKDA1 ;CHECK IF RKDA INCREMENTED CORRECTLY,
                                                        ;IF NOT RETURN HERE.
010340  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-
                                                        ;ZERO (PSUEDO-HDR), REST 377 WORDS
                                                        ;ARE ALL 0'S.
                                                        ;CHECK IF THE REST OF THE 377
                                                        ;WORDS ARE ALL 0'S
010342  012737  177775  001370      MOV     #-3,EFLG1   ;ALLOW ONLY 3 ERRORS
010350  012700  033344              MOV     #OUTBUF+2,R0 ;INITIALIZE PTR TO 2ND WRD IN BUFR
010354  012737  177401  001362      MOV     #-377,COUNT ;CHECK 377 WORDS IN THE BUFFER
010362  005710              11$:    TST     @R0         ;IS THIS WRD 0?
010364  001005                      BNE     12$         ;NO, ERROR
010366  005720                      TST     (R0)+       ;INCRMNT PTR TO NXT WRD
010370  005237  001362              INC     COUNT       ;CHKD ALL 377 WRDS?
010374  001372                      BNE     11$
010376  000412                      BR      7$          ;YES, BRANCH
010400  005037  001162      12$:    CLR     $REG0       ;GET EXPCTD WORD
010404  012037  001164              MOV     (R0)+,$REG1 ;GET WORD RECVD
010410  010137  001166              MOV     R1,$REG2    ;GET DISK ADRES, ERROR IN THIS
                                                        ;SECTOR
010414  104044                      ERROR   44          ;DATA ERROR, THE LAST 377 WORDS
                                                        ;READ FROM EACH SECTOR SHOULD BE 0
                                                        ;IN A PREVIOUS TEST, FIRST WORD OF
                                                        ;EVERY SEC (CYL 0) WAS WRITTEN AS A
                                                        ;PSUEDO-HDR, REST OF THE WORDS IN THE
                                                        ;SECTR ARE AUTOMATICALLY WRITTEN AS
                                                        ;0'S. THIS ERROR MAY MEAN THAT IT
                                                        ;DIDN'T HAPPEN SO
010416  005237  001370              INC     EFLG1       ;ALLOW ONLY 3 DATA ERORS OF THIS KIND
010422  001357                      BNE     11$

010424  005737  001344      7$:     TST     SIMUL       ;TESTING ON SIMULATOR?
010430  001011                      BNE     10$         ;YES BRANCH
                                                        ;IF NOT TESTING ON SIMULATOR GO AHEAD
                                                        ; & READ ALL 12 SECTORS ON CYL 0
010432  005201                      INC     R1          ;INCREMENT DRIV-ADRES TO NXT SECTOR
```



```
010434  005205                      INC     R5          ;INCREMENT 'EXPCTD DRIV-ADRES'
010436  122705  000014              CMPB    #14,R5      ;R U GOING TO READ THE LAST SECTOR?
010442  001002                      BNE     .+6         ;IF NOT, BRANCH
010444  062705  000004              ADD     #4,R5       ;IF YES, INCREMENT 'EXPCTD RKDA'
                                                        ;CORRECTLY
010450  005202                      INC     R2          ;HAVE U READ ALL 12 SECTORS?
010452  001266                      BNE     1$          ;IF NOT LOOP BACK & READ THE
                                                        ;NXT SECTOR
010454  022713  000204      10$:    CMP     #204,@R3    ;DOES RKCS, STILL HAVE THE 'READ' FUNCTION
010460  001406                      BEQ     8$          ;YES, BRANCH
010462  012737  000204  001162      MOV     #204,$REG0  ;GET EXPCTD RKCS
010470  011337  001164              MOV     @R3,$REG1   ;GET RKCS RECVD
010474  104024                      ERROR   24          ;RKCS SHOULD STILL CONTAIN THE 'READ'
                                                        ;FUNCTION BITS
010476  104413              8$:     CNT.RESET           ;GO ,DO CONTROL RESET
                                                        ;THIS IS A CALL FOR THE 'CNTRL-
                                                        ;RESET' ROUTINE. A CONTROL RESET IS
                                                        ;ISSUED AND AFTER A CERTAIN TIME
                                                        ;IF THE 'CNTRL RDY' DOES NOT SET
                                                        ;AN ERROR IS REPORTED. NOTE THAT
                                                        ;THE PC IN ERROR MESSAGE IS THE
                                                        ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                        ;THIS IS A VERY BASIC ERR & IF IT
                                                        ;OCCURS GO BACK TO TEST 10
010500  005777  170636              TST     @RKDB       ;DID CNTRL RESET CLEAR RKDB?
010504  001407                      BEQ     TST23       ;;YES, EXIT
010506  013737  001342  001162      MOV     RKDB,$REG0  ;GET ADRES OF RKDB
010514  017737  170622  001164      MOV     @RKDB,$REG1 ;GET CONTENTS OF RKDB
010522  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.
                                    ;;*****************************************************************
010524  000004                    TST23:  SCOPE
010526  012737  000001  001206            MOV     #1,$TIMES       ;;DO 1 ITERATION
010534  012737  010564  001110            MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
                                                                    ;ON  ERROR (SW 9)
010542  005003                           CLR     R3              ;(R3)=0, SURFACE 0 BEING WRITTEN
                                                                    ;(R3)-1, SURFACE 1 BEING WRITTEN
010544  012704  177465                   MOV     #-313,R4        ;SET UP COUNT FOR 203 CYLINDERS
010550  012702  177764                   MOV     #-14,R2         ;SET UP COUNT FOR 12 SECTORS
010554  013701  001350                   MOV     DRIVAD,R1       ;GET DRIVE ADRES
010560  010105                           MOV     R1,R5           ;STORE IT
010562  005205                           INC     R5
010564  104413                    1$:    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
010566  104421                           TST.SIN                 ;GO CHECK IF SIN IS SET
                                                                    ;IF SET, DO DRIVE RESET TO CLR IT
010570  005037  001362            7$:    CLR     COUNT
010574  010137  033342                   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
010600  012777  033342  170530           MOV     #OUTBUF,@RKBA   ;ADRES FROM WHICH WORD IS TO B X-FERRED
010606  012777  177777  170520           MOV     #-1,@RKWC       ;SET UP WORD COUNT
010614  010177  170520                   MOV     R1,@RKDA        ;ADRES THE DRIVE, WITH CORRECT CYL
                                                                    ;& SECTOR ADRES
010620  012777  002003  170504           MOV     #2003,@RKCS     ;WRITE FORMAT, GO

010626  105777  170500            2$:    TSTB    @RKCS           ;DID CNTRL RDY SET
010632  100411                           BMI     3$              ;YES, BRANCH
010634  005237  001362                   INC     COUNT           ;NO, HAVE U WAITED LONG ENOUGH?
010640  001372                           BNE     2$              ;IF NOT, LOOP BACK & WAIT
                                                                    ;IF YES, REPORT ERROR
010642  004737  020774                   JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
010646  010137  001202                   MOV     R1,$REG10       ;GET DISK ADRES, WHERE ERROR OCCURED
010652  104416                           BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                                    ;$REG10 -NTO DR #,CYL,SUR,SEC BITS
010654  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.
010656  032777  001000  170442  3$:    BIT     #1000,@RKDS     ;DID SIN BIT SET?
010664  001405                           BEQ     4$              ;NO, BRANCH
010666  004737  021002                   JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
010672  010137  001170                   MOV     R1,$REG3        ;GET, DISK-ADRES WHERE ERROR OCCURED
010676  104001                           ERROR   1               ;SIN SET WHILE DOING WRT FMT
```



```
                                                                    ;TO DISK-ADRES (AS IN $REG3)

010700  004737  021226            4$:    JSR     PC,CHKHE1       ;CHECK IF 'ERR' OR 'HE' BIT IS SET
                                                                    ;IF YES, RETURN HERE.
010704  104032                           ERROR   32              ;HE OR ERR SET WHILE DOING WRITE
                                                                    ;FORMAT ON SECTOR AS INDICATED IN
                                                                    ;<DSK-ADRES>
                                                                    ;WRT FMT WAS DONE STARTING AT <DSK-ADRES>
                                                                    ;INDICATED IN EROR MSGE.
010706  004737  021270            5$:    JSR     PC,CHKDA1       ;CHECK IF RKDA INCREMENTED CORRECTLY,
                                                                    ;IF NOT, RETURN HERE.
010712  104033                           ERROR   33              ;RKDA DID NOT INCREMENT CORRECTLY
                                                                    ;AFTER 'WRITE FORMAT' WAS DONE
                                                                    ;TO THE SECTOR PREVIOUS TO THAT
                                                                    ;INDICATED IN 'EXPCTD' RKDA
010714  005201            6$:    INC     R1              ;INCREMENT TO THE NXT SECTOR
010716  005205                   INC     R5              ;INCREMENT R5, TO WHAT RKDA WILL INCREMENT
010720  022702  177776           CMP     #-2,R2          ;R U GOING TO FORMAT THE LAST SECTOR
                                                            ;IN THE CYLINDER ?
010724  001002                   BNE     .+6             ;IF NOT, BRANCH
010726  062705  000004           ADD     #4,R5           ;INCREMENT R5 CORRECTLY TO 'EXPCTD RKDA'
010732  005202                   INC     R2              ;HAVE U FORMATTED ALL 12 SECTORS
                                                            ;ON THIS CYLINDER
010734  001313                   BNE     1$              ;IF NOT, LOOP BACK & FORMAT THE
                                                            ;NEXT SECTOR
                                                            ;YES
010736  012702  177764           MOV     #-14,R2         ;RESET THE COUNT FOR 12 SECTORS
010742  042701  000037           BIC     #37,R1          ;CLEAR THE SEC ADRES BITS
010746  005703                   TST     R3              ;SURFACE 1?
010750  001006                   BNE     8$              ;YES, BRANCH
010752  005203                   INC     R3              ;NO, SET FLAG
010754  062701  000020           ADD     #20,R1          ;INCREMENT TO THE NXT SURFACE
010760  010105                   MOV     R1,R5           ;THIS IS WHAT RKDA SHOULD
010762  005205                   INC     R5              ;INCREMENT TO.
010764  000677                   BR      1$              ;GO, DO NXT SURFACE
010766  062701  000040    8$:    ADD     #40,R1          ;INCREMENT TO NXT CYL
010772  010105                   MOV     R1,R5           ;POSITION FOR
010774  005205                   INC     R5              ;EXPCTD RKDA
010776  005003                   CLR     R3
011000  005204                   INC     R4              ;HAVE U FORMATTED ALL 203 CYLINDERS
011002  001270                   BNE     1$              ;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.
                                ;;*****************************************************************
011004  000004                  TST24:  SCOPE
011006  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
011014  012737  011100  001110          MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
                                                                ;ON      ERROR (SW 9)
011022  005037  001356                  CLR     INDX1           ;INDX1=0, SURFACE 0 BEING READ
                                                                ;INDX1=1, SURFACE 1 BEING READ
011026  013701  001350                  MOV     DRIVAD,R1       ;GET DRIVE ADRES
011032  010102                          MOV     R1,R2
011034  005737  001344                  TST     SIMUL           ;TESTING ON SIMULATOR?
011040  001410                          BEQ     12$             ;NO, BRANCH
011042  052701  014533                  BIS     #14533,R1       ;SET BITS FOR CYL 312, SEC 13, SUR 1
                                                                ;ON SIMULATOR, CHECK ONLY CYL 312,
                                                                ;SECTOR 13, SURFACE 1
011046  052702  014540                  BIS     #14540,R2       ;RKDA SHOULD INCRMNT TO THIS AFTR
                                                                ;RD FMT OF 1 SECTOR
011052  012737  177777  001370          MOV     #-1,EFLG1       ;SET COUNT FOR READING HDR
                                                                ;FROM 1 SECTOR ONLY
011060  000407                          BR      1$
011062  012705  177465          12$:    MOV     #-313,R5        ;SET UP COUNT FOR 203 CYLINDERS
011066  012737  177764  001370          MOV     #-14,EFLG1      ;SET COUNT FOR 12 HDRS TO BE
                                                                ;READ FROM EACH CYLINDER
011074  062702  000020                  ADD     #20,R2          ;THIS IS WHAT RKDA SHOULD INCREMENT
                                                                ;BY, AFTER 'RD FMT' OF EACH CYLINDER
011100  104413                  1$:     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
011102  104421                          TST.SIN                 ;CHECK IF SIN IS SET
                                                                ;IF SET DO DRV-RESET TO CLR IT
011104  012703  033342                  MOV     #OUTBUF,R3      ;STORE ADRES OF BUFFER
011110  005037  001360          11$:    CLR     INDX2
011114  010377  170216                  MOV     R3,@RKBA        ;ADRES TO WHICH DATA IS TO BE X-FERRED
                                                                ;FROM THE DISK
011120  013777  001370  170206          MOV     EFLG1,@RKWC     ;SET UP WORD COUNT FOR 12 HEADERS
                                                                ;TO BE READ OFF EACH CYLINDER
                                                                ;(ONLY 1 FOR SIMULATOR)
011126  010177  170206                  MOV     R1,@RKDA        ;ADRES THE DRIVE WITH CORRECT
                                                                ;CYLINDER & SECTOR ADRES
011132  012777  002005  170172          MOV     #2005,@RKCS     ;READ FORMAT, GO
```



```
011140  105777  170166          2$:     TSTB    @RKCS           ;DID CNTR1 RDY SET?
011144  100411                          BMI     3$              ;YES, BRANCH
011146  005237  001360                  INC     INDX2           ;NO, HAVE U WAITED LONG ENOUGH?
011152  001372                          BNE     2$              ;IF NOT, LOOP BACK & WAIT FOR IT
                                                                ;IF YES, REPORT ERROR
011154  004737  020774                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
011160  010137  001202                  MOV     R1,$REG10       ;GET DRIV-ADRES STARTING WHICH
                                                                ;'READ FORMAT' WAS DONE
011164  104416                          BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                                ;$REG10 INTO DR #,CYL,SUR,SEC BITS
011166  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.
011170  032777  001000  170130  3$:     BIT     #1000,@RKDS     ;DID 'SIN' SET?
011176  001405                          BEQ     4$              ;NO, BRANCH
011200  004737  021002                  JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
011204  010137  001170                  MOV     R1,$REG3        ;GET DISK-ADRES WHERE 'SIN'
                                                                ;OCCURED
011210  104001                          ERROR   1               ;SIN ERROR ON DOING RD FMT
                                                                ;TO CYL INDICATED IN $REG3
011212  004737  021226          4$:     JSR     PC,CHKHE1       ;CHECK IF 'ERR' OR 'HE' BIT IS SET.
                                                                ;IF YES, RETURN HERE.
011216  104046                          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
011220  020277  170114          5$:     CMP     R2,@RKDA        ;DID RKDA INCREMENT CORRECTLY BY 12 SEC
011224  001410                          BEQ     6$
011226  010237  001202                  MOV     R2,$REG10       ;GET EXPCTD RKDA
011232  104415                          BRKDA0                  ;GO TO 'BDA0' & BREAK  CONTENTS OF
                                                                ;$REG10 INTO DR #,CYL,SUR,SEC BITS
011234  017737  170100  001202          MOV     @RKDA,$REG10    ;GET RECVD RDKA
011242  104416                          BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
                                                                ;$REG10 INTO DR #,CYL,SUR,SEC BITS
011244  104040                          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
011246  013700  001370          6$:     MOV     EFLG1,R0        ;SET UP COUNT FOR 12 HEADERS TO B CHKD
                                                                ;(ONLY 1, IF SIMULATOR)
011252  010104                          MOV     R1,R4           ;GET DRIV-ADRES FROM WHERE RDFMT WAS DONE
011254  042704  160037                  BIC     #160037,R4      ;GET THE CYLINDER ADRES ONLY. (HEADER)
011260  020413                  7$:     CMP     R4,(R3)         ;IS THE RECVD HEADER SAME AS EXPCTD?
011262  001412                          BEQ     8$
011264  010437  001164                  MOV     R4,$REG1        ;GET EXPCTD HEADER WORD
011270  011337  001166                  MOV     (R3),$REG2      ;GET HEADER WORD RECVD
011274  010037  001162                  MOV     R0,$REG0
011300  062737  000014  001162          ADD     #14,$REG0       ;GET THE SECTOR (OCTAL NO) WHICH DID
                                                                ;NOT GIVE THE CORRECT HEADER
011306  104043                          ERROR   43              ;DID NOT RECIEVE THE CORRECT HEADER
```

```
3718                                                    ;WORD FROM 'SECTOR' AS INDICATED
3719                                                    ;(NOTE SECTOR # IS OCTAL)
3720  011310  005723              8$:     TST     (R3)+           ;INCREMENT POINTER TO THE NXT WORD
3721                                                    ;IN MEMORY WHERE THE RECVD HDR IS STORED
3722  011312  005200                      INC     R0              ;HAVE U CHECKED ALL 12 HEADERS?
3723  011314  001361                      BNE     7$              ;IF NOT, LOOP BACK & CHK THE NXT.
3724                                                    ;YES, ALL HEADERS FOR THIS CYLINDER
3725                                                    ;CHECKED.
3726  011316  004737  021316              JSR     PC,CHKWC        ;CHECK IF RKWC OVERFLOWED TO 0, IF
3727                                                    ;NOT RETURN HERE.
3728  011322  104041                      ERROR   41              ;RKWC DID NOT OVERFLOW AFTER DOING
3729                                                    ;RDFMT OF 12 SECTORS ON THE CYLINDER
3730                                                    ;NOTE THAT 'RKDA' IS THE INCREMENTED
3731                                                    ;RKDA AFTER THE RDFMT
3732  011324  005737  001344      9$:     TST     SIMUL           ;TSTING ON SIMULATOR?
3733  011330  001031                      BNE     TST25           ;;IF YES, EXIT
3734                                                    ;NO
3735  011332  005737  001356              TST     INDX1           ;DOING SURFACE 1
3736  011336  001011                      BNE     10$             ;YES, BRANCH
3737  011340  005237  001356              INC     INDX1           ;NO
3738  011344  062701  000020              ADD     #20,R1          ;INCREMENT DRIV ADRES TO THE NXT SURFACE
3739  011350  010102                      MOV     R1,R2
3740  011352  062702  000020              ADD     #20,R2          ;THIS IS WHAT RKDA SHOULD INCREMENT
3741                                                    ;TO, AFTER READ FMT OF THE CYLINDER
3742  011356  000137  011100              JMP     1$              ;GO RD FMT THE NXT SURFACE
3743  011362  005037  001356      10$:    CLR     INDX1           ;
3744  011366  042701  000037              BIC     #37,R1          ;CLR SEC, SURFACE BITS
3745  011372  062701  000040              ADD     #40,R1          ;INCREMENT TO NXT CYL
3746  011376  010102                      MOV     R1,R2           ;THIS IS WHAT RKDA SHOULD BE
3747  011400  062702  000020              ADD     #20,R2          ;AFTER RD FMT OF CYLINDER
3748  011404  005205                      INC     R5              ;HAVE U DONE ALL CYLINDERS?
3749  011406  001402                      BEQ     TST25           ;;EXIT
3750  011410  000137  011100              JMP     1$              ;IF NOT, LOOP BACK & READ FMT FROM
3751                                                    ;THE NXT CYLINDER
3752
3753
3754                                      ;;*************************************************************
3755                                      ;*TEST 25        CHECK 'READ' OF THE ENTIRE DISK
3756                                      ;*READ OF THE ENTIRE DISK (ONE WORD PER SECTOR) IS DONE
3757                                      ;*IN THIS TEST.  IN A PREVIOUS TEST THE FIRST WORD OF
3758                                      ;*EVERY SECTOR WAS WRITTEN LIKE A PSUEDO-HEADER (DRIVE #,
3759                                      ;*CYLINDER #, SURFACE & SECTOR #).  THESE PSUEDO HEADERS
3760                                      ;*WILL BE READ & CHECKED IN THIS TEST, PROVING THAT ANY
3761                                      ;*SECTOR CAN BE ACCESSED AND READ.
3762                                      ;*THE FOLLOWING CHECKING IS DONE
3763                                      ;*1. CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
3764                                      ;*OF FUNCTION.
3765                                      ;*2. IF 'SIN' OCCURRED?
3766                                      ;*3. IF 'HE' OR 'ERR' OCCURRED?
3767                                      ;*4. THE CORRECT FIRST WORD FROM EVERY SECTOR
3768                                      ;*WAS RECEIVED.  THIS WORD REFLECTS THE ABSOLUTE
3769                                      ;*DISK ADDRESS (DRV #, CYL #, SUR, SEC#) OF THAT SECTOR.
3770                                      ;*5. IF RKDB CONTAINED THE CORRECT WORD.
3771                                      ;*IF 'SIN' OCCURS DRIVE RESET IS DONE BEFORE READING
3772                                      ;*THE NEXT SECTOR.  READ IS DONE IN THIS ORDER SEC 0-11
3773                                      ;*CYL 0 SUR 0 -> SEC 0-11 CYL 0 SUR 1 -> SEC 0-11 CYL 1,....
```



```
3774                                      ;*IF TESTING ON SIMULATOR ONLY LAST CYLINDER (312), LAST
3775                                      ;*SECTOR (13), SURFACE 1 IS READ.
3776                                      ;;*************************************************************
3777  011414  000004              TST25:  SCOPE
3778  011416  012737  000001  001206      MOV     #1,$TIMES       ;;DO 1 ITERATION
3779  011424  012737  011470  001110      MOV     #1$,$LPERR      ;SET RETURN ADRES FOR
3780                                                    ;LOOPING ON ERROR (SW9)
3781  011432  012703  033342              MOV     #OUTBUF,R3
3782  011436  005004                      CLR     R4              ;FLAG, CLEAR WHEN READING SURFACE 0
3783                                                    ;SET WHEN READING SURFACE 1
3784  011440  013701  001350              MOV     DRIVAD,R1       ;GET DRIVE ADDRESS
3785  011444  005737  001344              TST     SIMUL           ;TSTING ON SIMULATOR?
3786  011450  001403                      BEQ     10$             ;IF NOT BRANCH
3787  011452  052701  014533              BIS     #14533,R1       ;SET ADRES BITS FOR LAST CYL (312)
3788  011456  000404                      BR      1$              ;LAST SECTOR (13), SURFACE 1
3789  011460  012700  177764      10$:    MOV     #-14,R0         ;SET COUNT FOR 12 SECTORS
3790  011464  012705  177465              MOV     #-313,R5        ;SET UP COUNT FOR 203 CYLINDERS
3791
3792  011470  104413              1$:     CNT.RESET               ;GO, DO CONTROL RESET
3793                                                    ;THIS IS A CALL FOR THE 'CNTRL-
3794                                                    ;RESET' ROUTINE. A CONTROL RESET IS
3795                                                    ;ISSUED AND AFTER A CERTAIN TIME
3796                                                    ;IF THE 'CNTRL RDY' DOES NOT SET
3797                                                    ;AN ERROR IS REPORTED. NOTE THAT
3798                                                    ;THE PC IN ERROR MESSAGE IS THE
3799                                                    ;PC WHERE 'CNT.RESET' IS LOCATED.
3800                                                    ;THIS IS A VERY BASIC ERR & IF IT
3801                                                    ;OCCURS GO BACK TO TEST 10
3802  011472  104421                      TST.SIN                 ;GO CHECK SIN, IF SET DO
3803                                                    ;DRIVE RESET TO CLR IT
3804  011474  005037  001356      8$:     CLR     INDX1
3805  011500  010377  167632              MOV     R3,@RKBA        ;ADRES TO WHICH DATA IS TO B X-FERRED
3806                                                    ;FROM THE DISK
3807  011504  012777  177777  167622      MOV     #-1,@RKWC       ;SET UP WORD COUNT
3808  011512  010177  167622              MOV     R1,@RKDA        ;ADRES THE DRIVE WITH CORRECT
3809                                                    ;CYLINDER & SECTOR ADRES
3810  011516  012777  000005  167606      MOV     #5,@RKCS        ;READ, GO
3811
3812  011524  105777  167602      2$:     TSTB    @RKCS           ;DID CNTRL RDY SET?
3813  011530  100411                      BMI     3$              ;YES, BRANCH
3814  011532  005237  001356              INC     INDX1           ;NO, HAVE U WAITED LONG ENOUGH
3815  011536  001372                      BNE     2$              ;IF NOT, LOOP BACK & WAIT FOR IT
3816                                                    ;IF YES, REPORT ERROR
3817  011540  004737  020774              JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS,DA
3818  011544  010137  001202              MOV     R1,$REG10       ;GET DISK-ADRES WHERE ERROR OCCURED
3819  011550  104416                      BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
3820                                                    ;$REG10 INTO DR #,CYL,SUR,SEC BITS
3821  011552  104045                      ERROR   45              ;CNTRL RDY DID NOT SET AFTER DOING
3822                                                    ;A 1 WORD READ FROM ADRES AS
3823                                                    ;INDICATED IN <DISK-ADRES>
3824                                                    ;'RKDA' IN EROR MSGE GIVES THE
3825                                                    ;CONTENTS OF RKDA AT THE TIME OF ERROR
3826
3827  011554  032777  001000  167544  3$:     BIT     #1000,@RKDS     ;DID 'SIN' SET?
3828  011562  001405                      BEQ     4$              ;NO, BRANCH
3829  011564  004737  021002              JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS
```

```
3830  011570  010137  001170          MOV     R1,$REG3        ;GET DISK-ADRES WHERE SIN OCCURED3
3831  011574  104001                  ERROR   1               ;'SIN' ERROR ON DOING READ FROM
3832                                                          ;DISK-ADRES INDICATED IN $REG3
3833  011576  004737  021226          4$:     JSR     PC,CHKHE1       ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
3834                                                          ;IF YES, RETURN HERE.
3835  011602  104046                  ERROR   46              ;'HE' OR 'ERR' ON DOING A READ OF
3836                                                          ;1 WORD FROM ADRES AS INDICATED
3837                                                          ;IN <DISK-ADRES>
3838                                                          ;'RKDA' IN EROR MSGE GIVES THE
3839                                                          ;CONTENTS OF RKDA AT THE TIME OF EROR
3840  011604  020113                  5$:     CMP     R1,(R3)         ;WAS THE CORRECT DATA WORD RECVD?
3841  011606  001407                  BEQ     6$
3842  011610  010137  001162          MOV     R1,$REG0        ;GET EXPCTD DATA WORD
3843  011614  011337  001164          MOV     (R3),$REG1      ;GET DATA WORD RECVD
3844  011620  010137  001166          MOV     R1,$REG2        ;GET DISK-ADRES
3845  011624  104044                  ERROR   44              ;DID NOT RECIEVE THE CORRECT
3846                                                          ;DATA WORD FROM DISK ON DOING
3847                                                          ;1 WORD READ FROM 'DISK-ADRES'
3848                                                          ;AS INDICATED BY 'EXPCTD' DATA WORD
3849                                                          ;NOTE THAT IN A PREVIOUS TEST THE
3850                                                          ;FIRST WORD OF EACH SECTOR IS UNIQUELY
3851                                                          ;WRITTEN WITH A WORD GIVING THE
3852                                                          ;ABSOLUTE ADDRESS OF THAT SECTOR IN
3853                                                          ;TERMS OF, DRIV #, CYL ADRES, SUR, SEC ADRS.
3854  011626  020177  167510          6$:     CMP     R1,@RKDB        ;DOES RKDB CONTAIN CORRECT WORD
3855  011632  001406                  BEQ     7$              ;YES, BRANCH
3856  011634  010137  001162          MOV     R1,$REG0        ;NO, GET EXPCTD RKDB
3857  011640  017737  167476  001164  MOV     @RKDB,$REG1     ;GET RKDB RECVD
3858  011646  104037                  ERROR   37              ;RKDB ERROR ON READ.
3859                                                          ;FOR RK11C, AFTER A READ RKDB
3860                                                          ;CONTAINS CHECKSUM FOR THE SECTOR
3861                                                          ;READ.
3862                                                          ;WHEREAS FOR RK11D, AFTER READ
3863                                                          ;RKDB CONTAINS THE LAST WORD
3864                                                          ;READ FROM THAT SECTOR &
3865                                                          ;X-FERRED TO MEMORY
3866  011650  005737  001344          7$:     TST     SIMUL           ;TESTING ON SIMULATOR?
3867  011654  001022                  BNE     TST26           ;;IF YES, EXIT
3868  011656  005201                  INC     R1              ;INCREMENT  TO ADRES NEXT SECTOR
3869  011660  005200                  INC     R0              ;HAVE U CHKD ALL 12 SECTORS?
3870  011662  001302                  BNE     1$              ;IF NOT, LUP BAK & CHK THE NXT
3871                                                          ;IF YES...
3872  011664  012700  177764          MOV     #-14,R0         ;RESET THE COUNT FOR 12 SECTORS
3873  011670  042701  000037          BIC     #37,R1          ;CLEAR SECTOR, SURFACE BITS
3874  011674  005704                  TST     R4              ;DOING SURFACE 1?
3875  011676  001004                  BNE     9$              ;YES, BRANCH
3876  011700  005204                  INC     R4              ;NO
3877  011702  062701  000020          ADD     #20,R1          ;INCREMENT THE ADRES TO NXT SURFACE
3878  011706  000670                  BR      1$              ;GO READ SURFACE 1
3879  011710  005004                  9$:     CLR     R4
3880  011712  062701  000040          ADD     #40,R1          ;INCREMENT TO NXT CYL
3881  011716  005205                  INC     R5              ;HAVE U CHKD ALL 203 CYLINDERS
3882  011720  001263                  BNE     1$              ;IF NOT, LOOP BACK & CHK THE NXT CYLINDER
3883                                                          ;YES
3884
3885
```



```
                                    ;;*****************************************************************
3886                                ;*TEST 26        CHECK 'SEEK' FUNCTION, WITH DIFFERRENT VELOCITY MODES
3887                                ;* THIS TEST CHECKS SEEK IN DIFFERENT VELOCITY MODES (DIFF <3,
3888                                ;* 3 < DIFF < 31, DIFF > 31).  FOR THESE 3 BASIC VELOCITIES SEEK IS DONE BOTH
3889                                ;* IN FWD AND REV DIRECTION TO CHECK THE ADDER & DIFFERENCE LOGIC.  IF
3890                                ;* WHILE DOING A SEEK 'SIN' OCCURS, A DRIVE RESET IS DONE TO INITIALIZE
3891                                ;* THE POSITIONING LOGIC
3892                                ;;*****************************************************************
3893  011722  000004                TST26:  SCOPE
3894  011724  012737  000005  001206        MOV     #5,$TIMES       ;;DO 5 ITERATIONS
3895  011732  012703  001372        MOV     #SEEK0,R3       ;INITIALIZE POINTER TO THE FIRST
3896                                                          ;SEEK ADDRESS
3897  011736  005037  001356        CLR     INDX1           ;INDX1, WHEN 0 INDICATES SEEK IN FWD DIRECTION
3898                                ;       WHEN 1 INICATES SEEK IN REV DIRECTION
3899  011742  013700  001332        MOV     RKCS,R0
3900  011746  013701  001326        MOV     RKDS,R1
3901  011752  013702  001330        MOV     RKER,R2
3902  011756  012737  011764  001110        MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING ON
3903                                                          ;EROR (SW 9)
3904  011764  000240                1$:     NOP
3905  011766  104413                2$:     CNT.RESET               ;GO, DO CONTROL RESET
3906                                                          ;THIS IS A CALL FOR THE 'CNTRL-
3907                                                          ;RESET' ROUTINE. A CONTROL RESET IS
3908                                                          ;ISSUED AND AFTER A CERTAIN TIME
3909                                                          ;IF THE 'CNTRL RDY' DOES NOT SET
3910                                                          ;AN ERROR IS REPORTED. NOTE THAT
3911                                                          ;THE PC IN ERROR MESSAGE IS THE
3912                                                          ;PC WHERE 'CNT.RESET' IS LOCATED.
3913                                                          ;THIS IS A VERY BASIC ERR & IF IT
3914                                                          ;OCCURS GO BACK TO TEST 10
3915
3916  011770  104421                        TST.SIN                 ;GO, CHECK IF SIN IS SET, IF SET
3917                                                          ;DO DRV-RESET TO CLEAR IT
3918
3919
3920  011772  013704  001350        MOV     DRIVAD,R4       ;GET DRIV-ADRES
3921  011776  051304                BIS     (R3),R4         ;SET CYLINDER BITS
3922  012000  010477  167334        MOV     R4,@RKDA        ;ADDRS THE DRIVE
3923  012004  012710  000011        MOV     #11,@R0         ;SET 'SEEK', 'GO'
3924
3925  012010  104412                        CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
3926                                                          ;IF SO, SKIP THE EROR MESSAGE.
3927  012012  104021                        ERROR   21              ;'CNTRL RDY' DID NOT SET AFTER
3928                                                          ;SENDING CYL ADD TO THE DRIV, 'ADD ACK'
3929                                                          ;FROM DRIVE SHLD HAVE COME BACK
3930                                                          ;THEREUPON SETTING 'CNTRL RDY'
3931  012014  005005                4$:     CLR     R5
3932  012016  032711  000100        5$:     BIT     #100,@R1        ;DID R/W/S RDY SET?
3933  012022  001005                BNE     6$              ;YES, BRANCH
3934  012024  005205                INC     R5              ;NO, WAIT
3935  012026  001373                BNE     5$              ;WAITED LONG?
3936  012030  004737  020774        JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
3937  012034  104026                ERROR   26              ;R/W/S RDY DID NOT SET ON
3938                                                          ;COMPLETION OF SEEK
3939  012036  032711  001000        6$:     BIT     #1000,@R1       ;DID SIN SET?
3940  012042  001403                BEQ     7$              ;NO, BRANCH
3941  012044  004737  020774        JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
```

```
3942  012050  104001                        ERROR   1               ;SIN SET ON DOING SEEK
3943  012052  032710  140000          7$:   BIT     #140000,@R0     ;DID 'HE' OR 'ERR' SET?
3944  012056  001403                        BEQ     8$              ;YES
3945  012060  004737  020774                JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
3946  012064  104022                        ERROR   22              ;'ERR OF 'HE' BIT SET WHEN
3947                                                                ;SEEKING TO CYL AS INDICATED
3948                                                                ;IN RKDA
3949
3950  012066  022710  000210          8$:   CMP     #210,@R0        ;DOES RKCS STILL CONTAIN THE 'SEEK' FNCTION
3951  012072  001406                        BEQ     9$              ;YES - EXIT
3952  012074  011037  001164                MOV     @R0,$REG1       ;NO, GET RKCS RECVD
3953  012100  012737  000210  001162        MOV     #210,$REG0      ;GET EXPCTD RKCS
3954  012106  104024                        ERROR   24              ;RKCS SHOULD CONTAIN THE 'SEEK' BITS
3955                                                                ;IF NOT, ERROR
3956
3957  012110  020477  167224          9$:   CMP     R4,@RKDA        ;DID RKDA CHANGE?
3958  012114  001406                        BEQ     10$             ;NO
3959  012116  010437  001162                MOV     R4,$REG0        ;YES, GET EXPCTD?
3960  012122  017737  167212  001164        MOV     @RKDA,$REG1     ;GET RKDA
3961  012130  104027                        ERROR   27              ;RKDA CHANGED AFTER DOING SEEK
3962
3963  012132  010477  167202          10$:  MOV     R4,@RKDA        ;ADRES THE DRIVE,SEC 0
3964  012136  012777  033342  167172        MOV     #OUTBUF,@RKBA   ;READ ONE HEADER INTO THIS
3965  012144  012777  177777  167162        MOV     #-1,@RKWC       ;BUS ADRES
3966  012152  012710  002005                MOV     #2005,@R0       ;GO,READ FORMAT
3967  012156  104414                        CNT.RDY                 ;WAIT FOR CNTRL RDY
3968  012160  021337  033342                CMP     (R3),OUTBUF     ;WAS THE CORRECT READE4R READ (FROM
3969  012164  001410                        BEQ     11$             ;CYLINDER TO WHICH SEEK WAS DONE BEFORE)
3970  012166  005037  001162                CLR     $REG0           ;STORE SEC # FROME  WHERE HDR WAS RD (0)
3971  012172  011337  001164                MOV     (R3),$REG1      ;GET EXPCTD HEADER
3972  012176  013737  033342  001166        MOV     OUTBUF,$REG2    ;GET HDR RECVD
3973  012204  104043                        ERROR   43              ;WRONG HDR WAS RECVD FROM CYLINDER (ADRES
3974                                                                ;IN ER MSGE). NOTE THAT A PURE SEEK WAS
3975                                                                ;DONE TO THIS CYL BEFORE  R_ADING HDR
3976                                                                ;USING READ FORMAT
3977  012206  005737  001356          11$:  TST     INDX1           ;SEEK IN REVRSE DIRECTION?
3978  012212  001007                        BNE     12$             ;YES, BRANCH
3979  012214  005723                        TST     (R3)+           ;NO, INCREMENT PTR TO NXT SEEK ADRES
3980  012216  022703  001400                CMP     #SEEK2+2,R3     ;DONE WITH ALL SKS IN FWD DIR?
3981  012222  001260                        BNE     1$              ;NO, GO & DO NXT ONE
3982  012224  005237  001356                INC     INDX1           ;SET FLAG INDICATING SK IN REVRSE
3983  012230  005743                        TST     -(R3)
3984  012232  005743                12$:    TST     -(R3)           ;POSITION PTR TO NXT SK IN REV
3985  012234  022703  001370                CMP     #SEEK0-2,R3     ;DONE WITH ALL?
3986  012240  001251                        BNE     1$              ;IF NOT, DO NXT ONE
3987
3988
3989
3990                                    ;;*************************************************************
3991                                    ;*TEST 27       CHECK DRIVE RESET FROM LAST CYLINDER
3992                                            ;*THE HEADS ARE POSITIONED ON THE LAST CYLINDER (DOING
3993                                            ;*AN IMPLIED SEEK-READ).  THEN A DRIVE RESET IS ISSUED.
3994                                            ;*IT'S CHECKED IF THE HEADS WERE BROUGHT BACK TO 0 BY
3995                                            ;*DOING A 1 WORD READ & CHECKING THAT THE CORRECT WORD
3996                                            ;*WAS RECEIVED. IF TESTING ON SIMULATOR THIS TEST IS SKIPPED.
3997                                    ;;*************************************************************
```



```
3998  012242  000004                TST27:  SCOPE
3999  012244  012737  000005  001206        MOV     #5,$TIMES       ;;DO 5 ITERATIONS
4000  012252  005737  001344                TST     SIMUL           ;R U ON A SIMULATOR?
4001  012256  001124                        BNE     TST30           ;;YES, EXIT
4002  012260  013701  001332                MOV     RKCS,R1
4003  012264  104413                        CNT.RESET               ;GO, DO CONTROL RESET
4004                                                                ;THIS IS A CALL FOR THE 'CNTRL-
4005                                                                ;RESET' ROUTINE. A CONTROL RESET IS
4006                                                                ;ISSUED AND AFTER A CERTAIN TIME
4007                                                                ;IF THE 'CNTRL RDY' DOES NOT SET
4008                                                                ;AN ERROR IS REPORTED. NOTE THAT
4009                                                                ;THE PC IN ERROR MESSAGE IS THE
4010                                                                ;PC WHERE 'CNT.RESET' IS LOCATED.
4011                                                                ;THIS IS A VERY BASIC ERR & IF IT
4012                                                                ;OCCURS GO BACK TO TEST 10
4013  012266  005000                        CLR     R0
4014  012270  012703  033342                MOV     #OUTBUF,R3      ;ADRES WHERE DATA WILL BE READ INTO
4015  012274  013704  001350                MOV     DRIVAD,R4
4016  012300  010405                        MOV     R4,R5
4017  012302  052705  014500                BIS     #14500,R5       ;SET CYL ADRES=312 (OCTAL)
4018  012306  010577  167026                MOV     R5,@RKDA        ;ADRES THE DRIVE, LAST CYLINDER
4019  012312  012777  177777  167014        MOV     #-1,@RK.C       ;READ 1 WORD
4020  012320  010377  167012                MOV     R3,@RKBA        ;INTO THIS MEMORY ADRES
4021
4022  012324  012711  000005                MOV     #5,@R1          ;READ, GO
4023
4024  012330  005000                        CLR     R0
4025  012332  104414                1$:     CNT.RDY                 ;THIS IS A CALL FOR CN.RDY ROUTINE
4026                                                                ;WHICH WAITS FOR CNTRL RDY TO SET.
4027                                                                ;A RETURN IS MADE AFTER CNTRL RDY
4028                                                                ;SETS. IF WITHIN A CERTAIN TIME
4029                                                                ;CNTRL RDY DOESN'T SET AN ERROR
4030                                                                ;MESSAGE IS GIVEN. WAITING TIME
4031                                                                ;883 MS FOR 11/20, 175 MS FOR 11/45
4032  012334  020513                2$:     CMP     R5,@R3          ;WAS THE CORRECT WORD READ?
4033  012336  001407                        BEQ     3$              ;YES, SEEK TO 312 WAS DONE CORRECTLY5,@
4034  012340  010537  001162                MOV     R5,$REG0        ;GET EXPCTD WORD
4035  012344  011337  001164                MOV     @R3,$REG1       ;GET WORD RECVD
4036  012350  010537  001166                MOV     R5,$REG2        ;GET DSK-ADRES FROM WHERE WORD WAS READ
4037  012354  104044                        ERROR   44              ;DID NOT READ BACK CORRECT WORD FROM
4038                                                                ;LAST CYL, SEC 0. IF TEST 45 & 46
4039                                                                ;WERE SUCCESSFULLY DONE THIS
4040                                                                ;ERROR MEANS THAT IMPLIED SEEK
4041                                                                ;TO CYL 312 COULD NOT B DONE
4042  012356  012711  000015                MOV     #15,@R1         ;DRIVE RESET, GO
4043  012362  104414                        CNT.RDY                 ;THIS IS A CALL FOR CN.RDY ROUTINE
4044                                                                ;WHICH WAITS FOR CNTRL RDY TO SET.
4045                                                                ;A RETURN IS MADE AFTER CNTRL RDY
4046                                                                ;SETS. IF WITHIN A CERTAIN TIME
4047                                                                ;CNTRL RDY DOESN'T SET AN ERROR
4048                                                                ;MESSAGE IS GIVEN. WAITING TIME
4049                                                                ;883 MS FOR 11/20, 175 MS FOR 11/45
4050  012364  005000                        CLR     R0
4051  012366  032777  000100  166732  4$:   BIT     #100,@RKDS      ;DID R/W/S RDY SET?
4052  012374  001011                        BNE     5$              ;YES, BRANCH
4053  012376  012702  177763                MOV     #-15,R2         ;IF U R ON A SLOWER MACHINE
```

```
4054  012402  005202                        INC     R2              ;& DO NOT NEED SUCH A LARGE MACHINE
4055  012404  001376                        BNE     .-2             ;TIME LOOP, CHANGE THESE 3
4056                                                                  ;INSTRUCTIONS TO 'NOP' THE
4057                                                                  ;LOOP TIME WILL BE REDUCED
4058                                                                  ;TO 1100 MS
4059                                                                  ;
4060                                                                  ;THE TOTAL TIME FOR THE ABOVE
4061                                                                  ;LOOPS (W/O PUTTING 'NOP'S) IS
4062                                                                  ;5304 MS FOR 11/20 AND
4063                                                                  ;1061 MS FOR 11/45 WITH MOS
4064                                                                  ;OR BIPOLAR MEMORY
4065  012406  005200                        INC     R0              ;WAITED LONG?
4066  012410  001366                        BNE     4$              ;IF NOT, LUP BAK & WAIT
4067                                                                  ;IF YES, ERROR
4068  012412  004737  020774                JSR     PC,GT4RG        ;GET RKCS,ER,DS,DA
4069  012416  104026                        ERROR   26              ;R/W/S RDY DID NOT SET AFTER
4070                                                                  ;DOING DRIVE RESET
4071  012420  032711  140000        5$:     BIT     #140000,@R1     ;DID HE OR ERR BIT SET?
4072  012424  001403                        BEQ     6$              ;IF NOT, BRANCH
4073
4074  012426  004737  020774                JSR     PC,GT4RG        ;GET RKCS,ER,DS,DA FOR ERROR MESSAGE
4075  012432  104022                        ERROR   22              ;HE OR ERR BIT SET ON DOING DRIVE
4076                                                                  ;RESET FROM LAST CYLINDER
4077  012434  005205                6$:     INC     R5              ;POSITION R5 TO EXPCTD RKDA
4078  012436  020577  166676                CMP     R5,@RKDA        ;DID THE CYL ADRES BITS IN RKDA GET CHANGED?
4079  012442  001406                        BEQ     7$              ;NO, BRANCH
4080  012444  010537  001162                MOV     R5,$REG0        ;GET EXPCTD RKDA
4081  012450  017737  166664  001164        MOV     @RKDA,$REG1     ;GET RKDA RECVD
4082  012456  104054                        ERROR   54              ;CYLINDER ADRES BITS IN RKDA
4083                                                                  ;GOT CHANGED AFTER
4084                                                                  ;DRIVE RESET, FROM LAST CYLINDER
4085  012460  012777  177777  166646  7$:     MOV     #-1,@RKWC       ;READ 1 WORD
4086  012466  010377  166644                MOV     R3,@RKBA        ;INTO THIS ADRES
4087  012472  010477  166642                MOV     R4,@RKDA        ;FROM THIS DSK ADRES-CYL 0, SEC 0
4088
4089  012476  012711  000005                MOV     #5,@R1          ;READ, GO
4090
4091  012502  005000                        CLR     R0
4092  012504  104414                8$:     CNT.RDY                 ;THIS IS A CALL FOR CN.RDY ROUTINE
4093                                                                  ;WHICH WAITS FOR CNTRL RDY TO SET.
4094                                                                  ;A RETURN IS MADE AFTER CNTRL RDY
4095                                                                  ;SETS. IF WITHIN A CERTAIN TIME
4096                                                                  ;CNTRL RDY DOESN'T SET AN ERROR
4097                                                                  ;MESSAGE IS GIVEN. WAITING TIME
4098                                                                  ;883 MS FOR 11/20, 175 MS FOR 11/45
4099  012506  020413                9$:     CMP     R4,@R3          ;WAS THE CORRECT WORD READ?
4100  012510  001407                        BEQ     TST30           ;;YES, EXIT
4101  012512  010437  001162                MOV     R4,$REG0        ;GET EXPCTD WORD
4102  012516  011337  001164                MOV     @R3,$REG1       ;GET WORD RECVD
4103  012522  010437  001166                MOV     R4,$REG2        ;GET DISK ADRES WHERE ERROR OCCURED
4104  012526  104044                        ERROR   44              ;DID NOT READ CORRECT WORD FROM
4105                                                                  ;CYL 0, SEC 0. IF TEST 45 & 46
4106                                                                  ;WERE SUCCESSFULLY DONE THIS
4107                                                                  ;ERROR COULD MEAN THAT DRIVE-RESET
4108                                                                  ;DID NOT BRING HEADS BACK TO 0.
4109
```



```
4110
4111                                ;;*************************************************************
4112                                ;*TEST 30        'WRITE' - 256 WORD BLOCK ON SECTOR 0, CYLINDER 0
4113                                        ;THE TEST BELOW SHOULD BE CONSIDERED AS A SET UP PHASE FOR
4114                                        ;THE FOLLOWING TEST.  IT WRITES A BLOCK OF 256 WORDS IN
4115                                        ;SECTOR 0, CYLINDER 0 WITH A SPECIFIC PATTERN AND THIS WRITTEN
4116                                        ;BLOCK WILL BE MADE USE OF IN THE NEXT TEST TO CHECK
4117                                        ;OUT 'WRITE-CHECK' AND 'READ CHECK' FUNCTIONS.
4118                                ;;*************************************************************
4119  012530  000004                TST30:  SCOPE
4120  012532  104413                        CNT.RESET               ;GO, DO CONTROL RESET
4121                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
4122                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
4123                                                                  ;ISSUED AND AFTER A CERTAIN TIME
4124                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
4125                                                                  ;AN ERROR IS REPORTED. NOTE THAT
4126                                                                  ;THE PC IN ERROR MESSAGE IS THE
4127                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
4128                                                                  ;THIS IS A VERY BASIC ERR& IF IT
4129                                                                  ;OCCURS GO BACK TO TEST 10
4130  012534  104421                        TST.SIN                 ;CHECK IF SIN IS SET, IF SET
4131                                                                  ;DO DRIVE RESET TO CLEAR IT
4132  012536  013704  001332                MOV     RKCS,R4
4133                                                                  ;THE FOLLOWING CODE IS FOR SETTING
4134                                                        ;UP THE I/O BUFFER IN MEMORY (STARTING AT
4135                                                        ;OUTBUF), WITH A PARTICULAR 256 WORD PATTERN.
4136                                                        ;STARTING FROM THE FIRST WORD IN THE BUFFER
4137                                                        ;THE LO BYTE WILL BE A COUNT PATTERN
4138                                                        ;FROM 0 TO 255 (DECIMAL), WHEREAS THE
4139                                                        ;HI-BYTE WILL BE THE COMPLEMENT OF LO BYTE,
4140                                                        ;A DECREASING COUNT PATTERN FROM 255 TO 0.
4141                                                        ;I.E.THE BUFFER WILL LOOK LIKE:
4142                                                        ;OUTBUF         (1 111 111 1 00 000 000)
4143                                                        ;OUTBUF+2       (1 111 111 0 00 000 001)
4144                                                        ;
4145                                                        ;LAST WORD      (0 000 000 0 11 111 111)
4146
4147  012542  012700  033342                MOV     #OUTBUF,R0
4148  012546  012701  177401                MOV     #177401,R1      ;PATTERN GENERATING NUMBER
4149  012552  012702  177400                MOV     #-400,R2        ;SET UP COUNT FOR 256 WORDS
4150  012556  012703  177400                MOV     #177400,R3      ;SET UP THE FIRST PATTERN TO B WRITTEN
4151
4152  012562  010320                        MOV     R3,(R0)+        ;SET UP FIRST WORD IN I/O BUFFER
4153  012564  005202                        INC     R2              ;INCREMENT COUNT
4154  012566  060103                1$:     ADD     R1,R3           ;SET UP NEXT WORD PATTERN
4155  012570  010320                        MOV     R3,(R0)+        ;WRITE IT IN NXT I/O BUFFER WORD
4156  012572  005202                        INC     R2              ;HAVE U WRITTEN ALL 256 WORDS
4157  012574  001374                        BNE     1$              ;IF NOT GO & WRITE NEXT PATTERN
4158
4159  012576  012777  177400  166530        MOV     #-400,@RKWC     ;WRITE 256 WORDS
4160  012604  012777  033342  166524        MOV     #OUTBUF,@RKBA   ;STARTING FROM THIS BUS ADRES
4161  012612  013777  001350  166520        MOV     DRIVAD,@RKDA    ;TO THIS DISK ADRES, CYL 0, SEC 0
4162
4163  012620  012714  000003                MOV     #3,@R4          ;WRITE, GO
4164
4165  012624  105714                2$:     TSTB    @R4             ;WAS CNTRL RDY CLEARED AS GO WAS SET?
```

```
                                        BPL     3$-2            ;YES, BRANCH
                                        JSR     PC,GT3RG        ;GET RKCS, ER, DS
                                        ERROR   30              ;CNTRL RDY DID NOT CLEAR AS GO WAS SET
                                                                ;TO 'WRITE'

                                        CLR     R2
                                3$:     TSTB    @RKCS           ;DID CNTRL RDY SET?
                                        BMI     4$              ;YES, BRANCH
                                        INC     R2              ;WAITED LONG ENOUGH?
                                        BNE     3$              ;IF NOT, LUP BAK & WAIT
                                                                ;IF YES, ERROR
                                        JSR     PC,GT4RG        ;GO, GET RKCD, ER, DS, DA
                                        MOV     DRIVAD,$REG10   ;GET THE STARING ADRES
                                        BRKDA4                  ;BREAK CONTENTS OF $REG10 INTO
                                                                ;DRV #, CYL, SUR, SEC #
                                        ERROR   31              ;CNTRL RDY DID NOT SET ON COMPLETION
                                                                ;OF WRITE OF 256 WORDS ON CYL 0, SEC 0
                                                                ;'RKDA' IN EROR MSGE GIVES THE
                                                                ;CONTENTS OF RKDA AT THE TIME OF EROR
                                                                ;WRITE WAS DONE STARTING AT <DSK-ADRES>
                                                                ;INDICATED IN EROR MSGE
                                4$:     JSR     PC,CHKHE        ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
                                                                ;IF YES, RETURN HERE
                                        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 EROR
                                5$:     CMP     R0,@RKBA        ;DID RKBA INCREMENT CORRECTLY?
                                        BEQ     6$              ;YES, BRANCH
                                        MOV     R0,$REG0        ;GET EXPCTD RKBA
                                        MOV     @RKBA,$REG1     ;GET RKBA RECVD
                                        ERROR   35              ;RKBA DID NOT INCREMENT CORRECTLY
                                                                ;(BY 1000 OCTAL BYTES) AFTER WRITE
                                                                ;OF 400 (OCTAL) WORDS ON SEC 0, CYL 0
                                6$:     JSR     PC,CHKWC        ;CHECK IF RKWC OVERFLOWED TO 0,
                                                                ;IF NOT RETURN HERE.
                                        ERROR   34              ;RKWC DID NOT OVERFLOW, AFTER A
                                                                ;WRITE OF 256 WORDS ON CYL 0, SEC 0
                                7$:     JSR     PC,CHKDA        ;CHECK IF RKDA INCREMENTED CORRECTLY,
                                                                ;IF NOT RETURN HERE
                                        ERROR   33              ;RKDA DID NOT INCREMENT BY 1 AFTER
                                                                ;A WRITE OF 256 WORDS IN CYL 0, SEC 0
                                8$:     JSR     PC,CHKER        ;CHECK IF ANY BIT RKER IS SET
                                                                ;IF YES RETURN HERE.
                                        ERROR   36              ;RKER BIT SET ON DOING WRITE ON
                                                                ;CYLINDER 0, SECTOR 0
                                9$:     CMP     #202,@R4        ;DOES RKCS STILL CONTAIN THE WRITE BITS?
                                        BEQ     TST31           ;;YES, EXIT
                                        MOV     #202,$REG0      ;GET EXPECTED RKCS
                                        MOV     @R4,$REG1       ;GET RKCS RECVD
                                        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
                                ;;*************************************************************
                                TST31:  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     #-400,R0        ;SET COUNT FOR 400 WORDS
                                        MOV     #OUTBUF,R1      ;TO BE CLEARED IN THE BUFFER
                                8$:     CLR     (R1)+           ;CLR THE 400 WORD BUFFER
                                        INC     R0              ;STARTING AT 'OUTBUF'
                                        BNE     8$
                                        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
```

```
4278  013116  012705  177773            MOV     #-5,R5
4279  013122  062702  177401    6$:     ADD     #177401,R2
4280  013126  020213                    CMP     R2,(R3) ;WAS THE READ WORD SAME AS THE WORD
4281                                                    ;THAT WAS SUPPOSE TO BE WRITTEN
4282  013130  001414                    BEQ     7$      ;YES, BRANCH
4283                                                    ;NO, ERROR
4284  013132  010137  001162            MOV     R1,$REG0        ;GET THE # OF WORD
4285  013136  062737  000401  001162    ADD     #401,$REG0      ;THAT IS IN ERROR (EXAMPLE=1,2--376,377,400)
4286  013144  010237  001164            MOV     R2,$REG1        ;GET EXPCTD WORD (THAT WAS SUPPOSED TO
4287                                                    ;BE WRITTEN)
4288  013150  011337  001166            MOV     (R3),$REG2      ;GET WORD RECVD (THAT WAS READ BAK)
4289  013154  104055                    ERROR   55              ;DID NOT READ BACK WORD THAT WAS SUPPOSED
4290                                                    ;TO HAVE BEEN WRITTEN PREVIOUSLY. POSITION
4291                                                    ;OF WORD IN ERROR IS AS INDICATED BY
4292                                                    ;WORD # ($REG0), SEC 0, CYL 0
4293  013156  005205                    INC     R5
4294  013160  001403                    BEQ     TST32           ;;EXIT
4295  013162  005723            7$:     TST     (R3)+           ;INCREMENT POINTER TO NXT WORD (THAT
4296                                                    ;WAS READ BACK)
4297  013164  005201                    INC     R1              ;HAVE U CHKD ALL 256 WORDS?
4298  013166  001355                    BNE     6$              ;IF NOT, LUP BAK & CHK THE NXT WORD
4299                                                    ;IF YES, EXIT
4300
4301                              ;;*************************************************************
4302                              ;*TEST 32       CHECK 'READ CHECK' FUNCTION - CYLINDER 0, SECTOR 0
4303                                      ;*THIS TEST CHECKS OUT THE BASIC :READ CHECK' LOGIC, USING THE DATA BLOCK
4304                                      ;*'CYLINDER, SECTOR 0) WRITTEN IN A PREVIOUS TEST.  HENCE THE TEST WHICH
4305                                      ;*WRITES THE DATA BLOCK SHOULD BE DONE PRIOR TO THIS TEST.
4306                              ;;*************************************************************
4307  013170  000004            TST32:  SCOPE
4308  013172  104413                    CNT.RESET               ;GO, DO CONTROL RESET
4309                                                    ;THIS IS A CALL FOR THE 'CNTRL-
4310                                                    ;RESET' ROUTINE. A CONTROL RESET IS
4311                                                    ;ISSUED AND AFTER A CERTAIN TIME
4312                                                    ;IF THE 'CNTRL RDY' DOES NOT SET
4313                                                    ;AN ERROR IS REPORTED. NOTE THAT
4314                                                    ;THE PC IN ERROR MESSAGE IS THE
4315                                                    ;PC WHERE 'CNT.RESET' IS LOCATED.
4316                                                    ;THIS IS A VERY BASIC ERR& IF IT
4317                                                    ;OCCURS GO BACK TO TEST 10
4318  013174  104421                    TST.SIN                 ;CHECK IF SIN IS SET, IF SET
4319                                                    ;DO DRIVE RESET TO CLEAR IT
4320  013176  013701  001332            MOV     RKCS,R1
4321  013202  013702  001334            MOV     RKWC,R2
4322  013206  013703  001340            MOV     RKDA,R3
4323  013212  013704  001336            MOV     RKBA,R4
4324  013216  012737  052525  033342    MOV     #52525,OUTBUF
4325  013224  012712  177400            MOV     #-400,@R2       ;READ CHECK 256 WORDS
4326  013230  013713  001350            MOV     DRIVAD,@R3      ;STARTING FROM CYL 0, SECTOR 0
4327  013234  012714  033342            MOV     #OUTBUF,@R4
4328  013240  012711  000013            MOV     #13,@R1         ;READ CHECK, GO
4329
4330  013244  105711            1$:     TSTB    @R1             ;DID CNTRL RDY GET CLEARED AS GO WAS SET?
4331  013246  100003                    BPL     2$              ;YES, BRANCH
4332  013250  004737  021002            JSR     PC,GT3RG        ;GET RKCS, ER, DS
4333  013254  104030                    ERROR   30              ;CNTRL RDY DID NOT CLEAR AS GO
```



```
4334  013256  104412            2$:     CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
4335                                                    ;IF SO, SKIP THE EROR MESSAGE.
4336                                                    ;WAS SET TO 'READ CHECK'
4337  013260  104056                    ERROR   56              ;CNTRL RDY DID NOT SET ON DOING
4338                                                    ;'READ CHECK' FROM CYL 0, SEC 0
4339  013262  032711  140000    3$:     BIT     #140000,@R1     ;DID 'ERR' OR 'HE' BIT SET?
4340  013266  001403                    BEQ     4$              ;NO, BRANCH
4341  013270  004737  021002            JSR     PC,GT3RG        ;GO, GET RKCS,ER,DS FOR ERROR MESSAGE
4342  013274  104057                    ERROR   57              ;'ERR' OR 'HE' BIT SET ON DOING
4343                                                    ;'READ CHECK' ON CYLINDER 0, SEC 0
4344  013276  032777  000002  166024    4$:     BIT     #2,@RKER        ;DID 'CSE' BIT SET IN RKER?
4345  013304  001404                    BEQ     5$              ;NO, BRANCH
4346  013306  017737  166016  001162    MOV     @RKER,$REG0     ;GET RKER
4347  013314  104060                    ERROR   60              ;SOFT ERROR - CSE - ON DOING 'READ
4348                                                    ;CHECK' ON CYLINDER 0, SECTOR 0
4349                                                    ;U SHOULD HAVE GOT ERRROR 102 ALSO
4350  013316  005712            5$:     TST     @R2             ;DID WORD COUNT OVERFLOW TO 0?
4351  013320  001405                    BEQ     6$              ;YES, BRANCH
4352  013322  011237  001162            MOV     @R2,$REG0       ;GET RKWC
4353  013326  011137  001164            MOV     @R1,$REG1       ;GET RKCS
4354  013332  104061                    ERROR   61              ;WORD COUNT DID NOT OVERFLOW
4355                                                    ;ON DOING 'READ CHK' ON CYL 0, SEC 0
4356  013334  013702  001350    6$:     MOV     DRIVAD,R2       ;RKDA SHOULD INCREMENT
4357  013340  005202                    INC     R2              ;TO THIS AFTER 'RD CHK' IS DONE
4358  013342  020213                    CMP     R2,@R3          ;DID RKDA INCREMENT CORRECTLY?
4359  013344  001405                    BEQ     7$
4360  013346  010237  001162            MOV     R2,$REG0        ;GET EXPCTD RKDA
4361  013352  011337  001164            MOV     @R3,$REG1       ;GET RKDA RECVD
4362  013356  104062                    ERROR   62              ;RKDA DID NOT INCREMENT CORRECTLY
4363                                                    ;(BY 1) ON DOING 'READ CHK' ON
4364                                                    ;CYL 0, SEC 0
4365  013360  022714  033342    7$:     CMP     #OUTBUF,@R4     ;DID RKBA GET CHANGED?
4366  013364  001406                    BEQ     8$              ;NO, BRANCH (RKBA WON'T CHANGE, NO NPR'S)
4367  013366  012737  033342  001162    MOV     #OUTBUF,$REG0   ;GET EXPCTD RKBA
4368  013374  011437  001164            MOV     @R4,$REG1       ;GET RKBA RECVD
4369  013400  104063                    ERROR   63              ;RKBA CHANGED AFTER DOING 'READ CHK'
4370                                                    ;ON CYLINDER 0, SECTOR 0. SHOULD
4371                                                    ;NOT CHANGE, FOR, NO NPR'S.
4372  013402  022737  052525  033342  8$:     CMP     #52525,OUTBUF   ;'OUTBUF' SHOULD STILL CONTAIN THE
4373                                                    ;SAME WORD AS IT DID BEFORE 'RD CHK'
4374                                                    ;NOTE THAT AT THE BEGINING OF THIS TEST
4375                                                    ;52525 WAS WRITTEN INTO 'OUTBUF'
4376  013410  001412                    BEQ     TST33           ;;YES, EXIT
4377                                                    ;REPORT ERROR IF 'OUTBUF' CHANGED
4378  013412  012737  033342  001162    MOV     #OUTBUF,$REG0   ;GET ADRES OF OUTBUF
4379  013420  012737  052525  001164    MOV     #52525,$REG1    ;GET EXPCTD WORD IN 'OUTBUF'
4380  013426  013737  033342  001166    MOV     OUTBUF,$REG2    ;GET WORD FOUND IN 'OUTBUF'
4381  013434  104064                    ERROR   64              ;AS MENTIONED ABOVE, IF 'WRITE' OF
4382                                                    ;256 WORD DATA BLOCK WAS DONE
4383                                                    ;CORRECTLY BEFORE, THEN THIS ERROR
4384                                                    ;COULD MEAN THAT AN NPR WAS DONE
4385                                                    ;ON 'READ CHECK'.
4386
4387                              ;;*************************************************************
4388                              ;*TEST 33       CHECK THE 'WRITE CHECK' FUNCTION - ON CYLINDER 0, SECTOR 0
4389                                      ;*THIS TEST CHECKS OUT THE BASIC 'WRITE CHECK' LOGIC, USING THE 256
```

```
                                        ;*WORD DATA BLOCK (SECTOR 0, CYLINDER 0) WRITTEN IN A PREVIOUS
                                        ;*TEST. THE BUFFER IN MEMORY, USED FOR COMPARISON OF DATA, IS THE
                                        ;*ONE STARTING AT 'OUTBUF'. HENCE THE TEST WHICH WRITES THE
                                        ;*256 WORD BLOCK ON THE DISK (AS WELL AS CREATING THE 256
                                        ;*256 WORD MEMORY BUFFER) SHOULD BE DONE BEFORE THIS TEST.
                                        ;;*************************************************************
013436  000004                TST33:  SCOPE
013440  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
013442  104421                        TST.SIN                 ;CHECK IF SIN IS SET, IF SET
                                                                ;DO DRIVE RESET TO CLEAR IT
013444  013701  001332                MOV     RKCS,R1
013450  012700  177400                MOV     #-400,R0
013454  012702  033342                MOV     #OUTBUF,R2
013460  012703  177777                MOV     #177777,R3
013464  062703  177401        1$:     ADD     #177401,R3
013470  010322                        MOV     R3,(R2)+
013472  005200                        INC     R0
013474  001373                        BNE     1$
013476  012777  177400  165630        MOV     #-400,@RKWC     ;WRITE CHECK 256 WORDS
013504  012777  033342  165624        MOV     #OUTBUF,@RKBA   ;STARTING AT THIS BUS ADRES
013512  013777  001350  165620        MOV     DRIVAD,@RKDA    ;WITH THIS DISK DATA BLOCK (CYL 0, SEC 0)
013520  012711  000007                MOV     #7,@R1          ;WRITE CHECK, GO

013524  005000                        CLR     R0              ;GIVE SOME TIME
013526  105711                2$:     TSTB    @R1             ;DID CNTRL RDY CLEAR AS GO WAS SET?
013530  100003                        BPL     3$              ;YES BRANCH
013532  004737  021002                JSR     PC,GT3RG        ;GET RKCS, ER, DS
013536  104030                        ERROR   30              ;CNTRL RDY DID NOT CLEAR AS GO WAS
                                                                ;SET TO DO WRITE CHECK
013540  104412                3$:     CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
                                                                ;IF SO, SKIP THE ERROR MESSAGE.
013542  104065                        ERROR   65              ;CNTRL RDY DID NOT SET AFTER
                                                                ;COMPLETING WRITE CHECK ON
                                                                ;CYLINDER 0, SECTOR 0
013544  032711  140000        4$:     BIT     #140000,@R1     ;DID HE OR ERR BIT SET
013550  001403                        BEQ     5$              ;NO, BRANCH
013552  004737  021002                JSR     PC,GT3RG        ;GO GET RKCS ER DS FOR ERROR MESSAGE
013556  104066                        ERROR   66              ;HE OR ERR BIT SET ON DOING WRITE
                                                                ;CHK ON CYLINDER 0, SEC 0
013560  032777  000001  165542 5$:     BIT     #1,@RKER        ;DID WCE SET IN RKER?
013566  001403                        BEQ     6$              ;NO, BRANCH
013570  004737  021002                JSR     PC,GT3RG        ;YES GET RKCS, ER, DS
013574  104067                        ERROR   67              ;WCE ON WRITE CHECK OF CYL 0, SEC 0
                                                                ;NOTE THAT IF A PREVIOUS TEST
                                                                ;& THEN COMPARED WITH MEMORY BUFFER
                                                                ;TO SEE IF IT WAS WRITTEN CORRECT WAS
                                                                ;DONE RIGHT BEFORE, THIS ERROR SHOULD NOT
```



```
                                                                ;HAPPEN UNLESS THERE IS A FAULT IN THE
                                                                ;COMPARING LOGIC OF 'WRT CHK'
013576  005777  165532        6$:     TST     @RKWC           ;DID RKWC OVERFLOW?
013602  001406                        BEQ     7$              ;YES, BRANCH
013604  017737  165524  001162        MOV     @RKWC,$REG0     ;NO, GET RKWC
013612  011137  001164                MOV     @R1,$REG1       ;GET RKC:
013616  104061                        ERROR   61              ;RKWC DID NOT OVERFLOW AFTER
                                                                ;WRITE CHECK ON CYL 0, SEC 0
013620  013704  001350        7$:     MOV     DRIVAD, R4      ;RKDA SHOULD INCREMENT
013624  005204                        INC     R4              ;TO THIS AFTER WRT CHK
013626  020477  165506                CMP     R4,@RKDA        ;DID RKDA INCREMENT CORRECTLY?
013632  001406                        BEQ     8$              ;YES, BRANCH
013634  010437  001162                MOV     R4,$REG0        ;NO, GET EXPCTD RKDA
013640  017737  165474  001164        MOV     @RKDA,$REG1     ;GET RKDA RECVD
013646  104070                        ERROR   70              ;RKDA DID NOT INCREMENT CORRECTLY
                                                                ;(BY 1 SECTOR) AFTER WAT CHK ON SEC 0, CYL 0
013650  022777  034342  165460 8$:     CMP     #OUTBUF+1000,@RKBA ;DID RKBA INCREMENT CORRECTLY?
013656  001407                        BEQ     9$              ;YES, EXIT
013660  012737  034342  001162        MOV     #OUTBUF+1000,$REG0 ;GET EPCTD RKBA
013666  017737  165444  001164        MOV     @RKBA,$REG1     ;GET RKBA RECVD
013674  104071                        ERROR   71              ;RKBA DID NOT INCREMENT CORRECTLY
                                                                ;(BY 1000 BYTES) AFTER A WRT CHK
                                                                ;OF 256 WORDS ON CYL 0, SEC 0
013676  022711  000206        9$:     CMP     #206,@R1        ;DOES RKCS STILL CONTAIN THE WRT CHK BITS?
013702  001406                        BEQ     TST34           ;;YES, BRANCH
013704  012737  000206  001162        MOV     #206,$REG0      ;NO, GET EXPCTD RKCS
013712  011137  001164                MOV     @R1,$REG1       ;GET RKCS RECVD
013716  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.
                                        ;;*************************************************************
013720  000004                TST34:  SCOPE
013722  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
013724  104421                        TST.SIN                 ;CHECK IF SIN IS SET, IF SET
                                                                ;DO DRIVE RESET TO CLEAR IT
013726  013701  001332                MOV     RKCS,R1
```

```
 4502  013732  012700  177400              MOV     #-400,R0        ;SET UP COUNT FOR 256 WORDS
 4503  013736  012702  033342              MOV     #OUTBUF,R2
 4504  013742  010203                      MOV     R2,R3
 4505
 4506  013744  005023              1$:     CLR     (R3)+           ;CLEAR OUT THE 256
 4507  013746  005200                      INC     R0              ;WORD MEMORY BUFFER STARTING
 4508  013750  001375                      BNE     1$              ;AT 'OUTBUF'
 4509  013752  012777  177400  165354      MOV     #-400,@RKWC     ;READ BACK 256 WORDS
 4510  013760  010277  165352              MOV     R2,@RKBA        ;INTO THIS BUS ADRES (IBA WILL B SET)
 4511  013764  013777  001350  165346      MOV     DRIVAD,@RKDA    ;FROM THIS DSK ADRES (SEC 0, CYL 0)
 4512                                                              ;NOTE: SEC 0 HAS BEEN WRITTEN IN A
 4513                                                              ;PREVIOUS TEST WITH A UNIQUE PATTERN
 4514  013772  012711  004005              MOV     #4005,@R1       ;READ, GO, IBA SET
 4515
 4516  013776  005037  001362              CLR     COUNT
 4517  014002  105711              2$:     TSTB    @R1             ;DID CNTRL RDY SET?
 4518  014004  100412                      BMI     3$              ;YES, BRANCH
 4519  014006  005237  001362              INC     COUNT           ;WAITED LONG ENOUGH?
 4520  014012  001373                      BNE     2$              ;IF NOT, LUP BAK & WAIT
 4521  014014  004737  020774              JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
 4522  014020  013737  001350  001202      MOV     DRIVAD,$REG10   ;GET THE STARTING ADRES
 4523  014026  104416                      BRKDA4                  ;BREAK CONTENTS OF $REG10
 4524                                                              ;INTO DR #, CYL, SUR, SEC
 4525  014030  104045                      ERROR   45              ;CNTRL RDY DID NOT SET AFTER DOING
 4526                                                              ;READ
 4527  014032  004737  021234      3$:     JSR     PC,CHKHE        ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
 4528                                                              ;IF YES, RETURN HERE.
 4529  014036  104046                      ERROR   46              ;ERR BIT SET ON DOING READ FROM SEC 0,
 4530                                                              ;CYL 0 (INDICATED  IN <DSK-ADRES>)
 4531                                                              ;'RKDA' IN EROR MSGE GIVES THE
 4532                                                              ;CONTENTS OF RKDA AT THE TIME OF EROR
 4533
 4534  014040  020277  165272      4$:     CMP     R2,@RKBA        ;DID RKBA INCREMENT?
 4535  014044  001406                      BEQ     5$              ;OK IF NOT, BRANCH
 4536  014046  010237  001162              MOV     R2,$REG0        ;GET EXPCTD RKBA
 4537  014052  017737  165260  001164      MOV     @RKBA,$REG1     ;GET RKBA RECVD
 4538  014060  104072                      ERROR   72              ;RKBA INCREMNTED WHEN IBA BIT WAS
 4539                                                              ;SET, SHOULD NOT HAVE
 4540  014062  032777  001000  165236  5$: BIT     #1000,@RKDS     ;IS SIN SET?
 4541  014070  001042                      BNE     TST35           ;;IF YES, EXIT
 4542  014072  012700  177400              MOV     #-400,R0
 4543  014076  022712  000377              CMP     #377,@R2        ;CHECK THAT THE FIRST WORD IN
 4544                                                              ;'OUTBUF' IS 377 (LAST WORD OF SEC 0,
 4545                                                              ;CYL 0). NOTE THAT READ WAS DONE
 4546  014102  001411                      BEQ     6$              ;INTO THIS SAME WRD WITH IBA SET
 4547  014104  012737  000377  001162      MOV     #377,$REG0       ;GET EXPCTD WORD (LAST WORD OF THE BUFFER
 4548  014112  011237  001164              MOV     (R2),$REG1      ;GET WORD RECVD (LAST WRD FROM SEC 0)
 4549  014116  013737  001350  001166      MOV     DRIVAD,$REG2    ;DISK ADRES WHERE ERROR OCCURED
 4550                                                              ;(SEC 0, CYL 0 LAST WORD)
 4551                                                              ;DATA ERROR
 4552  014124  104044                      ERROR   44              ;THE FIRST WORD IN MEM BUFFER (OUTBUF)
 4553                                                              ;SHOULD BE NON-ZERO & SHOULD CONTAIN
 4554                                                              ;THE LAST WORD READ BACK FROM SEC 0
 4555                                                              ;CYL 0,THIS DID NOT HAPPEN IF THE ERROR OCCURS
 4556  014126  005722              6$:     TST     (R2)+           ;INCREMENT POINTER TO THE NXT WORD
 4557  014130  012705  177773              MOV     #-5,R5          ;ALLOW ONLY 5 MESAGES FOR ERR 116
```



```
 4558  014134  005200              7$:     INC     R0              ;CHKD ALL 256 WORDS IN THE BUFFER?
 4559  014136  001417                      BEQ     TST35           ;;YES, EXIT
 4560  014140  005722                      TST     (R2)+           ;IS THIS WORD 0?
 4561  014142  001774                      BEQ     7$              ;YES, LUP BAK & CHK THE NXT WORD?
 4562  014144  005037  001164              CLR     $REG1           ;ERROR. GET EXPCTED WORD - 0
 4563  014150  014237  001166              MOV     -(R2),$REG2     ;GET WORD THAT WAS FOUND IN THE BUFFER
 4564  014154  010004                      MOV     R0,R4
 4565  014156  062704  000401              ADD     #401,R4
 4566  014162  010437  001162              MOV     R4,$REG0        ;THIS 'WORD #' IN MEMORY BUFFER
 4567                                                              ;SHOULD HAVE BEEN ZERO
 4568  014166  104073                      ERROR   73              ;THE 256 WORD BUFER (STARTING AT
 4569                                                              ;OUTBUF) WAS CLEARED BEFORE READING
 4570                                                              ;BAK SEC 0 INTO IT. SINCE THE IBA
 4571                                                              ;BIT WAS SET DURING THE READ, ONLY
 4572                                                              ;THE FIRST WORD OF (OUTBUF) SHOULD
 4573                                                              ;HAVE CHANGED, THE REST OF THE WORDS
 4574                                                              ;SHOULD BE STILL 0. IF THIS ERROR
 4575                                                              ;OCCURS, 'WORD #' (OF THE BUFFER) AS
 4576                                                              ;INDICATED IN THE EROR MESAGE) GOT
 4577                                                              ;CHANGED WHEN READ WAS DONE FROM
 4578                                                              ;THE DISK, INDICATING THAT WITH IBA
 4579                                                              ;SET X-FER WAS NOT DONE INTO THE
 4580                                                              ;SAME MEMORY LOCATION. 'WORD #'
 4581                                                              ;IS OCTAL & SPECIFIES THE POSITION
 4582                                                              ;IN THE BUFFER (FIRST WORD IS 'WORD #' 1)
 4583  014170  005205                      INC     R5
 4584  014172  001401                      BEQ     TST35           ;;EXIT
 4585  014174  000757                      BR      7$
 4586                              ;
 4587                              ;;*************************************************************
 4588                              ;*TEST 35        CHECK THAT RK11 INTERRUPTS WHEN IDE IS SET
 4589                                      ;*THIS TEST CHECKS IF RK11 INTERRUPTS TO ITS DESIGNATED VECTOR
 4590                                      ;*ADDRESS WHEN IDE BIT IS SET, WITH CONTROL READY SET & GO CLEAR.
 4591                                      ;* IT IS NORMALLY 220, UNLESS IT HAS BEEN CHANGED. IF IT HAS BEEN
 4592                                      ;*CHANGED RK11 WILL INTERRUPT TO 'RKVEC'. NOTE 'RKVEC' HAS
 4593                                      ;*TO BE SET UP BY THE USER.
 4594                              ;;*************************************************************
 4595  014176  000004              TST35:  SCOPE
 4596  014200  104413                      CNT.RESET               ;GO, DO CONTROL RESET
 4597                                                              ;THIS IS A CALL FOR THE 'CNTRL-
 4598                                                              ;RESET' ROUTINE. A CONTROL RESET IS
 4599                                                              ;ISSUED AND AFTER A CERTAIN TIME
 4600                                                              ;IF THE 'CNTRL RDY' DOES NOT SET
 4601                                                              ;AN ERROR IS REPORTED. NOTE THAT
 4602                                                              ;THE PC IN ERROR MESSAGE IS THE
 4603                                                              ;PC WHERE 'CNT.RESET' IS LOCATED.
 4604                                                              ;THIS IS A VERY BASIC ERR& IF IT
 4605                                                              ;OCCURS GO BACK TO TEST 10
 4606  014202  104421                      TST.SIN                 ;CHECK IF SIN IS SET, IF SET
 4607                                                              ;DO DRIVE RESET TO CLEAR IT
 4608  014204  012746  000340              MOV     #340,-(SP)
 4609  014210  012746  014216              MOV     #64$,-(SP)
 4610  014214  000002                      RTI
 4611  014216                      64$:
 4612  014216  013701  001332              MOV     RKCS,R1
 4613  014222  013700  001402              MOV     RKVEC,R0        ;GET POINTER TO RK VECTOR ADRES
```

```
4614  014226  012720  014262                  MOV     #1$,(R0)+       ;SET UP INTERRUPT VECTOR FOR RK11
4615  014232  012710  000340                  MOV     #340,(R0)       ;SET PSW ON INTERRUPT
4616  014236  105711                          TSTB    @R1             ;WAIT FOR CNTRL RDY TO SET
4617  014240  100376                          BPL     .-2             ;
4618  014242  012711  000100                  MOV     #100,@R1        ;SET IDE BIT IN RKCS
4619  014246  104420  000005                  WAT.INT ,5              ;WAIT FOR INTERRUPT, ATLEAST
4620                                                                  ;37 US FOR 11/20, 7 US FOR 11/45
4621  014252  011137  001162                  MOV     @R1,$REG0       ;GET RKCS
4622  014256  104074                          ERROR   74              ;RK11 DID NOT INTERRUPT WHEN IDE
4623                                                                  ;WAS SET, WITH CNTRLE RDY SET & GO
4624                                                                  ;CLEAR
4625  014260  000400                          BR      1$
4626  014262  022626                  1$:     CMP     (SP)+,(SP)+     ;RK11 INTERRUPTED CORRECTLY TO
4627                                                                  ;THIS. RESTORE STACK POINTER
4628                                                                  ;(FROM RK11 INTERRUPT)
4629  014264  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
4630                                                                  ;(FROM  WAT.INT)
4631  014266  012777  014302  165106          MOV     #2$,@RKVEC      ;IF THERE IS FAULTY POLLING OR INTERUPT
4632                                                                  ; LOGIC SECOND INTERRUPT MIGHT OCCUR
4633  014274  104420  000005                  WAT.INT ,5              ;WAIT FOR INTERRUPT,IF ANY
4634                                                                  ;DUE TO FAULTY LOGIC
4635
4636  014300  000403                          BR      3$
4637
4638  014302  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE STACK PTR (FROM RK11 INTRUPT)
4639  014304  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK PTR (FROM WAT.INT)
4640  014306  104020                          ERROR   20              ;AN UNEXPECTED RK11 INTERRUPT
4641                                                                  ;OCCURED. THERE SHOULD HAVE BEEN
4642                                                                  ;ONLY 1 INTERRUPT (TO 1$ ABOVE)
4643  014310  012777  004600  165064  3$:     MOV     #BADINT,@RKVEC  ;RESTORE VECTOR ADRES FOR
4644                                                                  ;UNEXPECTED RK11 INTERRUPT,
4645  014316  012746  000340                  MOV     #340,-(SP)
4646  014322  012746  014330                  MOV     #65$,-(SP)
4647  014326  000002                          RTI
4648  014330                          65$:
4649
4650
4651                                  ;;**************************************************************
4652                                  ;*TEST 36        CHECK THAT WITH IDE SET RK11 INTERRUPTS AFTER INTIATION & COMPLETION OF
4653                                          ;*THIS TEST CHECKS THAT AN INTERRUPT FROM RK11 OCCURS AFTER
4654                                          ;*A SEEK IS INITIATED WITH 'IDE' BIT SET, AND THEN A SECOND
4655                                          ;*INTERRUPT OCCURS AFTER THE SEEK IS DONE. IT ALSO CHECKS THAT
4656                                          ;*AFTER THE FIRST INTERRUPT 'SCP' BIT IS NOT SET, WHEREAS AFTER
4657                                          ;*THE SECOND INTERRUPT 'SCP' IS SET.
4658                                          ;*THIS TEST ALSO CHECKS A PART OF THE POLLING LOGIC.
4659                                  ;;**************************************************************
4660  014330  000004                  TST36:  SCOPE
4661  014332  012737  000005  001206          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
4662  014340  104413                          CNT.RESET               ;GO, DO CONTROL RESET
4663                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
4664                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
4665                                                                  ;ISSUED AND AFTER A CERTAIN TIME
4666                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
4667                                                                  ;AN ERROR IS REPORTED. NOTE THAT
4668                                                                  ;THE PC IN ERROR MESSAGE IS THE
4669                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
```



```
4670                                                                  ;THIS IS A VERY BASIC ERR& IF IT
4671                                                                  ;OCCURS GO BACK TO TEST 10
4672  014342  013700  001332                  MOV     RKCS,R0
4673  014346  013777  001350  164764          MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE
4674  014354  004737  021504                  JSR     PC,DRESET       ;GO, DO DRIVE RESET
4675  014360  104026                          ERROR   26              ;R/W/S RDY DIDN'T SET AFTER DOING
4676                                                                  ;ABOVE DRIVE RESET
4677  014362  013701  001402          2$:     MOV     RKVEC,R1
4678  014366  012721  014432                  MOV     #3$,(R1)+       ;SET UP VECTOR ADRES FOR RK11 INTERUPT
4679  014372  012711  000340                  MOV     #340,(R1)       ;SET UP PSW ON INTERRUPT
4680  014376  052777  000040  164734          BIS     #40,@RKDA       ;ADRES  CYLINDER #1
4681  014404  012710  000111                  MOV     #111,@R0        ;SEEK, GO WITH IDE SET
4682  014410  104420  000300                  WAT.INT ,300            ;WAIT FOR THE DRIVE TO
4683                                                                  ;INTERRUPT AFTER ADRES WAS RECVD
4684                                                                  ;WAITING TIME= 1.4 MS FOR 11/20
4685                                                                  ;280 US FOR 11/45
4686                                                                  ;ERROR, IF INTERUPT DID NOT OCCUR
4687                                                                  ;BY NOW
4688  014414  012777  004600  164760          MOV     #BADINT,@RKVEC  ;RESTORE UNEXPECTED RK11 INTERRUPT
4689  014422  011037  001162                  MOV     @R0,$REG0       ;GET RKCS
4690  014426  104075                          ERROR   75              ;INTERRUPT DID NOT OCCUR AFTER
4691                                                                  ;SEEK WAS INITIATED WITH IDE SET
4692  014430  000402                          BR      3$+4
4693  014432  022626                  3$:     CMP     (SP)+,(SP)+     ;OK, IF RK11 INTERRUPTED TO THIS
4694                                                                  ;RESTORE STACK POINTER (FROM RK11 INTERRUPT)
4695  014434  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER (FROM
4696                                                                  ;WAT.INT)
4697  014436  012777  014502  164736          MOV     #5$,@RKVEC      ;SET UP NEW VECTOR ADRES FOR RK11
4698  014444  032710  020000                  BIT     #20000,@R0      ;IS SCP CLEAR
4699  014450  001403                          BEQ     4$              ;YES, BRANCH
4700  014452  011037  001162                  MOV     @R0,$REG0       ;GET RKCS
4701  014456  104076                          ERROR   76              ;SCP SET BEFORE SEEK TO LAST
4702                                                                  ;CYLINDER WAS DONE
4703  014460  104420  056700          4$:     WAT.INT ,56700          ;WAIT FOR DRIVE TO INTERRUPT
4704                                                                  ;AFTER SEEK WAS COMPLETED
4705                                                                  ;WAITING TIME=180 MS FOR 11/20
4706                                                                  ;36 MS FOR 11/45
4707  014464  012777  004600  164710          MOV     #BADINT,@RKVEC  ;IT'S AN ERROR IF BY THIS TIME
4708                                                                  ;INTERRUPT HAS NOT OCCURERED
4709  014472  004737  021002                  JSR     PC,GT3RG                ;GO GET RKCS, ER, DS
4710  014476  104077                          ERROR   77              ;RK11 DID NOT INTERRUPT AFTER SEEK (TO
4711                                                                  ;LAST CYLINDER) WAS DONE WITH IDE SET
4712  014500  000401                          BR      5$+2
4713  014502  022626                  5$:     CMP     (SP)+,(SP)+     ;OK, IF RK11 INTERUPTED TO THIS AFTER
4714                                                                  ;SEEK WAS COMPLETED. RESTORE
4715                                                                  ;STACK POINTER (FROM RK11 INTERRUPT)
4716  014504  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER (FROM
4717                                                                  ;WAT.INT)
4718  014506  012777  004600  164666          MOV     #BADINT,@RKVEC  ;RESTORE RK11 INTERRUPT VECTOR ADRES
4719                                                                  ;FOR UNEXPECTED INTERUTS
4720  014514  032710  020000                  BIT     #20000,@R0      ;DID SCP BIT SET?
4721  014520  001003                          BNE     6$              ;YES, BRANCH
4722  014522  011037  001162                  MOV     @R0,$REG0       ;GET RKCS
4723  014526  104053                          ERROR   53              ;SCP DID NOT SET AFTER RK11 INTERRUPTED
4724                                                                  ;INDICATING SEEK WAS DONE
4725  014530  017701  164572          6$:     MOV     @RKDS,R1        ;GET RKDS
```

```
                                        BIC     #17777,R1       ;MASK NON-ID BITS IN RKDS
                                        CMP     R1,DRIVAD       ;CORRECT ID BITS IN RKDS?
                                        BEQ     7$              ;YES, BRANCH

                                        MOV     DRIVAD,-(SP)    ;PUSH DRV ADRES ON THE STACK
                                        JSR     PC,SHFTRT       ;GO, SHIFT RIGHT DRV #
                                        MOV     (SP)+,$REG0     ;GET EXPCTD DRV #
                                        MOV     R1,-(SP)        ;PUSH ID BITS ON THE STACK
                                        JSR     PC,SHFTRT       ;GO SHIFT THEM RIGHT
                                        MOV     (SP)+,$REG1     ;POP THE RECVD ID BITS
                                        ERROR   47              ;WRONG ID BITS WERE RECVD IN
                                                                ;RKDS AFTER SEEK WAS DONE (INTRUPT
                                                                ;MODE). 'EXPCT' INDICATES THE DRIVE
                                                                ;# THAT SHOULD HAVE BEEN IN THE
                                                                ;ID BITS, 'RECVD' INDICATES THE
                                                                ;DRIVE # THAT WAS RECVD IN THE ID BITS

                                7$:
                                        MOV     #340,-(SP)
                                        MOV     #64$,-(SP)
                                        RTI
                                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,$REG0     ;GET RKDS
                                        ERROR   50              ;CONTROL RESET DIDN'T CLEAR THE
                                                                ;DRIVE ID BITS (13-15) IN RKDS

                                8$:     CMP     #200,@R0        ;WAS SCP BIT CLEARED BY CNTRL RESET?
                                        BEQ     TST37           ;;YES, EXIT
                                        MOV     @R0,$REG0       ;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$,(R5)+       ;SET UP VECTOR ADRES FOR RK11 TO INTRUPT
                                        MOV     #340,(R5)       ;SET PSW ON INTERUPT
                                        MOV     #105,@R0        ;READ, GO, IDE SET
                                        WAT.INT ,127710         ;WAIT FOR RK11 TO INTERRUPT ON
                                                                ;COMPLETION OF READ
                                                                ;WAITING TIME= 337 MS FOR 11/20
                                                                ;67 MS FOR 11/45
                                        MOV     #BADINT,@RKVEC  ;RESTORE UNEXPCTED INTERRUPT VECTOR ADRES
                                        MOV     @R0,$REG0       ;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)
                                        CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER (FROM WAT.INT)
                                        MOV     #BADINT,@RKVEC  ;RESTORE UNEXPECTED RK11 INTERRUPT
                                                                ;VECTOR ADRES
                                        JSR     PC,CHKER        ;CHECK IF ANY BIT IN RKER IS SET,
                                                                ;IF YES, RETURN HERE.
                                        ERROR   36              ;RKER SET ON DOING READ FROM SEC 0,
                                                                ; CYL 13 IN INTERRUPT MODE
                                4$:     ADD     #5,R1           ;RKDA SHOULD HAVE INCREMENTED TO THIS
                                        CMP     R1,@R2          ;DID RKDA INCREMENT CORRECTLY?
                                        BEQ     2$              ;YES BRANCH
                                        MOV     R1,$REG0        ;GET EXPCTD RTDA
                                        MOV     @R2,$REG1       ;GET RKDA RECVD
                                        ERROR   40              ;RKDA INCREMENTED WRONG ON DOING
                                                                ;A READ ON CYL 0, SEC 13
                                2$:     JSR     PC,CHKWC        ;CHECK THAT RKWC OVERFLOWED TO 0,
                                                                ;IF NOT RETURN HERE.
                                        ERROR   41              ;RKWC DIDN'T OURFLO AFTER
                                                                ;A READ OF 200 WORDS

                                3$:
                                        MOV     #340,-(SP)
                                        MOV     #64$,-(SP)
                                        RTI
                                64$:
                                        CMP     #OUTBUF+400,@R4 ;DID RKBA INCREMENT CORRECTLY?
                                        BEQ     TST40           ;;YES, EXIT
                                        MOV     #OUTBUF+400,$REG0 ;GET EXPCT RKBA
                                        MOV     @R4,$REG1       ;GET RKBA RECVD
                                        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.
                        ;;*************************************************************
015070  000004         TST40:  SCOPE
015072  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
015074  104421                 TST.SIN                ;CHECK IF SIN IS SET, IF SET
                                                       ;DO DRIVE RESET TO CLEAR IT
015076  012737  015132  001110         MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
                                                       ;ON ERROR (SW 9)
015104  013700  001332         MOV     RKCS,R0
015110  013777  001350  164222         MOV     DRIVAD,@RKDA
015116  012701  000007         MOV     #7,R1           ;PRIORITY LEVEL 7
015122  012702  000340         MOV     #340,R2         ;BR LEVEL 7 FOR PSW
015126  013703  001400         MOV     RKPRI,R3        ;NOTE, IF RK11 INTERRUPT LEVEL IS
                                                       ;CHANGED FROM 5 TO ANY OTHER LEVEL
                                                       ;THEN CHANGE CONTENTS OF 'RKPRI'
                                                       ; ACCORDINGLY
015132  013704  001402         1$:     MOV     RKVEC,R4
015136  012724  015244         MOV     #3$,(R4)+       ;SET UP ADRES FOR RK11 TO INTERUPT
015142  012714  000340         MOV     #340,(R4)       ;SET UP PSW ON INTERUPT
015146  010246                 MOV     R2,-(SP)                ;SET PROCESSOR PRIORITY LEVEL AS
015150  012746  015156         MOV     #4$,-(SP)
015154  000002                 RTI
015156                         4$:                     ;INDICATED BY R2
015156  012710  000100         MOV     #100,@R0        ;SET THE IDE BIT
015162  012705  177760         MOV     #-20,R5         ;WAIT FOR THE RK11 INTERRUPT
015166  005205                 INC     R5              ;WAITING TIME=78 US FOR 11/20
015170  001376                 BNE     .-2             ;13 US FOR 11/45
015172  020203                 CMP     R2,R3           ;WAS THE CPU PRIORITY LEVEL LESS THAN
015174  003005                 BGT     2$              ;THE RK11 LEVEL?  IF YES, RK11
                                                       ;SHOULD HAVE INTERRUPTED. ERROR,
                                                       ;IF IT D.D NOT
015176  010137  001162         MOV     R1,$REG0        ;GET CPU BR LEVEL
015202  011037  001164         MOV     @R0,$REG1       ;GET RKCS
015206  104103                 ERROR   103             ;THOUGH CPU LEVEL WAS LESS THAN
                                                       ;THE RK11 LEVEL (5), RK11 DID NOT
                                                       ;INTERRUPT
015210  005010         2$:     CLR     @R0             ;CLEAR RKCS
015212  062702  177740         ADD     #-40,R2         ;DECREASE THE PRIORITY LEVEL (FOR
                                                       ;CPU) BY 1
015216  005301                 DEC     R1              ;CPU WILL B AT THIS LEVEL
015220  001344                 BNE     1$              ;LUP BAK & CHK FOR THIS BR LEVEL.
```



```
                                                       ;DONE WITH CHKING FOR ALL LEVELS.
015222  012777  004600  164152         MOV     #BADINT,@RKVEC  ;RESTORE UNEXPECTED RK11 INTERRUPT
                                                       ;VECTOR
015230  012746  000340         MOV     #340,-(SP)
015234  012746  015242         MOV     #64$,-(SP)
015240  000002                 RTI
015242                         64$:
015242  000414                 BR      TST41           ;;EXIT,TO NXT TST
015244  022626         3$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
015246  012777  004600  164126         MOV     #BADINT,@RKVEC  ;RESTORE UNEXPECTED RK11 INTERRUPT
                                                       ;VECTOR
015254  020203                 CMP     R2,R3           ;IF THIS INTERRUPT OCCURED WHEN
015256  003754                 BLE     2$              ;CPU LEVEL WAS LESS THAN THE
                                                       ;RK11 PRIORITY LEVEL (5) THEN IT IS
                                                       ;OK.  IF NOT SO, ERROR
015260  010137  001162         MOV     R1,$REG0        ;GET CPU BR LEVEL
015264  011037  001164         MOV     @R0,$REG1       ;GET RKCS
015270  104104                 ERROR   104             ;RK11 INTERRUPTED WHEN THE CPU
                                                       ;LEVEL (AS POINTED BY R1) WAS
                                                       ;HIGHER OR SAME AS THE RK11
                                                       ;LEVEL (5)
015272  000746                 BR      2$              ;GO BACK & CHK THE NXT LEVEL

                        ;;*************************************************************
                        ;*TEST 41        SIMULATE & CHECK 'OVR' ERROR
                                ;*THIS TEST SIMULATES OVERRUN ERROR AND CHECKS IF THE OVR
                                ;*BIT IN RKER GETS SET.  THEN IT IS CLEARED USING CNTRL RESET
                                ;*& CHECKED THAT IT WAS CLEARED.  OVR CONDITION IS SIMULATED
                                ;*BY TRYING TO READ 401(OCTAL) WORDS FROM LAST CYLINDER(312),
                                ;*LAST SECTOR (13), SURFACE 1.
                        ;;*************************************************************
015274  000004         TST41:  SCOPE
015276  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
015300  104421                 TST.SIN                ;CHECK IF SIN IS SET, IF
                                                       ;SET, DO DRIVE RESET TO CLR IT
015302  013701  001350         MOV     DRIVAD,R1       ;GET ADRES OF DRIVE
015306  052701  014533         BIS     #14533,R1       ;SET BITS FOR LAST CYLINDER (312),
                                                       ;SUR 1, LAST SECTOR (13)
015312  012777  177377  164014         MOV     #-401,@RKWC     ;READ 401 WORDS
015320  012777  033342  164010         MOV     #OUTBUF,@RKBA   ;INTO THIS MEMORY BUFFER
015326  010177  164006         MOV     R1,@RKDA        ;FROM THIS DSK ADRES, LAST CYL,
                                                       ;LAST SEC, SURFACE 1
015332  012777  000005  163772         MOV     #5,@RKCS                ;READ, GO

015340  005002                 CLR     R2
015342  105777  163764         1$:     TSTB    @RKCS           ;DID CNTRL RDY SET?
```

```
4950  015346  100410                          BMI     2$              ;YES, BRANCH
4951  015350  005202                          INC     R2              ;NO, WAIT FOR IT
4952  015352  001373                          BNE     1$              ;IF WAITED LONG, REPORT ERROR MESSAGE BECAUSE
4953                                                                  ;OVR SHOULD HAVE SET HE CAUSING
4954                                                                  ;CNTRL RDY TO SET BY NOW
4955  015354  017737  163754  001166          MOV     @RKWC,$REG2
4956  015362  004737  021010                  JSR     PC,GT2RG        ;GO, GET RKCS, ER
4957  015366  104002                          ERROR   2               ;CNTRL RDY DID NOT SET AFTER DOING
4958                                                                  ;AN OVR READ.  HE SHOULD HAVE OCCURED
4959                                                                  ;SETTING CNTRL RDY (HE BECAUSE OF
4960                                                                  ;OVR CONDITIONS)
4961  015370  032777  040000  163732  2$:     BIT     #40000,@RKER    ;DID OVR BIT SET IN RKER?
4962  015376  001006                          BNE     3$
4963  015400  004737  021010                  JSR     PC,GT2RG        ;GET RKCS, ER
4964  015404  012737  040000  001166          MOV     #40000,$REG2    ;THIS BIT (OVR) DID NOT SET.
4965  015412  104105                          ERROR   105             ;OVR ERROR BIT DID NOT SET IN RKER
4966                                                                  ;ON SIMULATING OVR CONDITIONS
4967  015414  022777  140204  163710  3$:     CMP     #140204,@RKCS   ;DID HE & ERR SET WHEN OVR SET IN RKER?
4968  015422  001403                          BEQ     4$              ;YES, BRANCH
4969  015424  004737  021010                  JSR     PC,GT2RG        ;GET RKCS, ER
4970  015430  104106                          ERROR   106             ;HE OR ERR BIT DID NOT SET IN RKCS WHEN
4971                                                                  ;AN OVR ERROR WAS SIMULATED
4972                                                                  ;CLEAR OVER, ERR, HE BITS
4973  015432  104413                  4$:     CNT.RESET               ;GO, DO CONTROL RESET
4974                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
4975                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
4976                                                                  ;ISSUED AND AFTER A CERTAIN TIME
4977                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
4978                                                                  ;AN ERROR IS REPORTED. NOTE THAT
4979                                                                  ;THE PC IN ERROR MESSAGE IS THE
4980                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
4981                                                                  ;THIS IS A VERY BASIC ERR& IF IT
4982                                                                  ;OCCURS GO BACK TO TEST 10
4983  015434  004737  021356                  JSR     PC,CHKECLR      ;CHECK IF 'OVR' BIT WAS CLEARED BY
4984                                                                  ;CON.RESET, IF NOT RETURN HERE.
4985  015440  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR OVR
4986                                                                  ;BIT IN RKER
4987  015442  004737  021402          5$:     JSR     PC,CHKCCLR      ;CHECK IF 'ERR' & 'HE' BIT GOT CLEARED BY
4988                                                                  ;CON.RESET, IF NOT RETURN HERE.
4989  015446  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
4990                                                                  ;HE OR ERR BIT IN RKCS.
4991  015450  004737  021504          6$:     JSR     PC,DRESET       ;GO DO DRIVE RESET
4992  015454  104026                          ERROR   26              ;R/W/S RDY DIDN'T SET
4993                                                                  ;AFTER THE ABOVE DRIVE RESET
4994
4995                                  ;;*****************************************************************
4996                                  ;*TEST 42        SIMULATE & CHECK PGE ERROR
4997                                          ;*THIS TEST SIMULATES 'PROGRAMMING ERROR' & CHECKS IF IT IS
4998                                          ;*DETECTED BY PGE BIT IN RKER.  THEN A CNTRL RESET IS DONE &
4999                                          ;*IT IS CHECKED IF PGE BIT WAS CLEARED.  IT IS ALSO CHECKED IF
5000                                          ;*THE SETTING & CLEARING OF PGE BIT SETS & CLEARS HE, ERR
5001                                          ;*BITS IN RKCS.
5002                                  ;;*****************************************************************
5003  015456  000004                  TST42:  SCOPE
5004  015460  104413                          CNT.RESET               ;GO, DO CONTROL RESET
5005                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
```



```
5006                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
5007                                                                  ;ISSUED AND AFTER A CERTAIN TIME
5008                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
5009                                                                  ;AN ERROR IS REPORTED. NOTE THAT
5010                                                                  ;THE PC IN ERROR MESSAGE IS THE
5011                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
5012                                                                  ;THIS IS A VERY BASIC ERR& IF IT
5013                                                                  ;OCCURS GO BACK TO TEST 10
5014  015462  104421                          TST.SIN                 ;GO CHECK IF SIN IS SET. IF
5015                                                                  ;SET DO DRIVE RESET TO CLR IT
5016  015464  013701  001330                  MOV     RKER,R1
5017  015470  013777  001350  163641          MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE, CYLINDER 0
5018
5019  015476  012777  002011  163626          MOV     #2011,@RKCS     ;SEEK, GO WITH FMT SET
5020                                                                  ;THIS IS A PGE SIMULATION
5021  015504  104414                          CNT.RDY                 ;THIS IS A CALL FOR 'CN.RDY'
5022                                                                  ;ROUTINE WHICH WAITS FOR CNT
5023                                                                  ;RDY TO SET. IF CNTRL RDY DOES
5024                                                                  ;NOT SET WITHIN 883 MS/ 11-20
5025                                                                  ;(176 MS FOR 11-45 WITH BIPOLAR)
5026                                                                  ;AN ERROR IS REPORTED
5027  015506  032711  004000                  BIT     #4000,@R1       ;DID PGE BIT IN RKER SET?
5028  015512  001006                          BNE     1$              ;YES, BRANCH
5029  015514  012737  004000  001166          MOV     #4000,$REG2     ;THIS BIT IN RKER (PGE) DID NOT SET
5030  015522  004737  021010                  JSR     PC,GT2RG        ;GO GET RKCS, ER FOR MESAGE
5031  015526  104105                          ERROR   105             ;PGE BIT DID NOT SET IN RKER
5032                                                                  ;ON SIMULATION OF PGE CONDITION
5033                                                                  ;$REG2 CONTAINS THE RKER BIT (PGE)
5034                                                                  ;THAT SHOULD HAVE SET.
5035  015530  022777  142210  163574  1$:     CMP     #142210,@RKCS   ;DID HE & ERR BITS SET?
5036  015536  001403                          BEQ     2$              ;YES, BRANCH
5037  015540  004737  021010                  JSR     PC,GT2RG        ;GO, GET RKCS, ER
5038  015544  104106                          ERROR   106             ;HE OR ERR BIT DID NOT SET WHEN
5039                                                                  ;PGE SET IN RKER.
5040                                                                  ;CLEAR PGE, HE, ERR BITS
5041  015546  104413                  2$:     CNT.RESET               ;GO, DO CONTROL RESET
5042                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
5043                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
5044                                                                  ;ISSUED AND AFTER A CERTAIN TIME
5045                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
5046                                                                  ;AN ERROR IS REPORTED. NOTE THAT
5047                                                                  ;THE PC IN ERROR MESSAGE IS THE
5048                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
5049                                                                  ;THIS IS A VERY BASIC ERR& IF IT
5050                                                                  ;OCCURS GO BACK TO TEST 10
5051  015550  004737  021356                  JSR     PC,CHKECLR      ;CHECK IF 'PGE' BIT GOT CLEARED BY
5052                                                                  ;CONTROL RESET, IF NOT RETURN HERE.
5053  015554  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
5054                                                                  ;PGE BIT IN RKER
5055  015556  004737  021402          3$:     JSR     PC,CHKCCLR      ;CHECK IF 'ERR' BITGOT CLEARED BY
5056                                                                  ;CON.RESET, IF NOT RETURN HERE.
5057  015562  104102                          ERROR   102             ;RKCS BITS HE OR ERR DID NOT
5058                                                                  ;GET CLEARED BY CNTRL RESET
5059
5060                                  ;;*****************************************************************
5061                                  ;*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.
                                        ;;*************************************************************
015564  000004                TST43:  SCOPE
015566  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
015570  104421                        TST.SIN                 ;GO CHECK IF SIN IS SET
                                                                ;IF SET DO DRIVE RESET TO CLR IT
015572  005002                        CLR     R2
015574  013700  001332                MOV     RKCS,R0
015600  012777  177777  163526        MOV     #-1,@RKWC       ;WRITE CHECK 1 WORD
015606  012777  160000  163522        MOV     #160000,@RKBA   ;AT THIS BUS ADRES
015614  013777  001350  163516        MOV     DRIVAD,@RKDA    ;WITH THIS DISK ADRES (CYL 0, SEC 0)
015622  012710  000067                MOV     #67,@R0         ;WRT CHK. GO, MEX BITS SET
015626  105777  163500        1$:     TSTB    @RKCS           ;DID CNTRL RDY SET AS A RESULT OF HE?
015632  100410                        BMI     2$              ;YES, BRANCH
015634  005202                        INC     R2              ;WAITED LONG ENOUGH?
015636  001373                        BNE     1$              ;IF NOT LUP BAK & WAIT
015640  004737  021010                JSR     PC,GT2RG        ;GET RKCS, ER
015644  017737  163464  001166        MOV     @RKWC,$REG2     ;GET RKWC
015652  104113                        ERROR   113             ;CNTRL RDY DID NOT SET ON DOING
                                                                ;A WRT CHK WITH A NXM LOCATION.
                                                                ;THIS HE SHOULD HAVE SET THE
                                                                ;CNTRL RDY BIT IN RKCS
015654  032777  002000  163446  2$:     BIT     #2000,@RKER     ;DID NXM BIT IN RKER SET?
015662  001006                        BNE     3$              ;YES, BRANCH
015664  004737  021010                JSR     PC,GT2RG        ;GO GET RKCS, RKER
015670  012737  002000  001166        MOV     #2000,$REG2     ;THIS BIT (NXM) DID NOT SET IN RKER
015676  104105                        ERROR   105             ;NXM BIT DID NOT SET IN RKER ON
                                                                ;SIMULATING NXM CONDITION.
015700  022710  140266        3$:     CMP     #140266,@R0     ;DID HE & ERR BIT SET?
015704  001403                        BEQ     4$              ;YES, BRANCH
015706  004737  021010                JSR     PC,GT2RG                ;GO, GET RKCS, RKER
015712  104106                        ERROR   106             ;HE OR ERR BIT DID NOT SET WHEN
                                                                ;NXM ERROR WAS SIMULATED
                                                                ;CLEAR NXM, HE, ERR BITS
015714  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
015716  004737  021356                JSR     PC,CHKECLR      ;CHECK IF 'NXM' BIT GOT C;LEARED BY
                                                                ;CON.RESET, IF NOT RETURN HERE.
015722  104102                        ERROR   102             ;CNTRL RESET DID NOT CLEAR
                                                                ;NXM BIT IN RKER
015724  004737  021402        5$:     JSR     PC,CHKCCLR      ;CHECK IF 'HE' & 'ERR' BITS GOT CLEARED
                                                                ;BY CON.RESET, IF NOT RETURN HERE.
015730  104102                        ERROR   102             ;CNTRL RESET DID NOT CLEAR
                                                                ;HE OR ERR BIT IN RKCS.
015732  004737  021436        6$:     JSR     PC,TSTRWS       ;GO CHECK IF R/W/S RDY IS SET &
                                                                ;WAIT FOR IT.  SKIP ERROR IF IT IS SET
015736  104016                        ERROR   16              ;R/W/S RDY IS NOT SET
                                        ;;*************************************************************
                                        ;*TEST 44       SIMULATE & CHECK NXD ERROR
                                                ;*THIS TEST SIMULATES NON-EXISTENT DISK ERROR & CHECKS IF
                                                ;*IT IS DETECTED BY NXD BIT OF RKER.  IF ALL EIGHT ARE PRESENT
                                                ;*THEN THIS TEST IS ABORTED FOR SIMULATION CANNOT BE DONE.
                                        ;;*************************************************************
015740  000004                TST44:  SCOPE
015742  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
015744  104421                        TST.SIN                 ;CHECK IF SIN IS SET, IF SET
                                                                ;DO DRV RESET TO CLR IT
015746  013700  001332                MOV     RKCS,R0
015752  012702  160000                MOV     #160000,R2      ;ADRES DRIVE 7 TO FIND
                                                                ;IF IT IS PRESENT
015756  010277  163356        1$:     MOV     R2,@RKDA        ;ADRES DRIVE # POINTED TO BY R2
015762  104417  000001                DELAY   ,1              ;TIME DELAY, 7.5 US ON 11/20,
                                                                ;1.5 US ON 11/45
015766  105777  163334                TSTB    @RKDS           ;IS IT PRESENT?
015772  100004                        BPL     2$              ;NO, BRANCH
015774  062702  160000                ADD     #-20000,R2      ;ADRES THE NXT DRIVE IN THE
                                                                ;REVERSE ORDER. I.E. 7,6,...
016000  001366                        BNE     1$              ;LUP BAK & TRY TO FIND A DRIVE
                                                                ;THAT'S NOT PRESENT
016002  000435                        BR      TST45           ;;EXIT TO THE NXT TST
016004  012710  000015        2$:     MOV     #15,@R0         ;DRIVE RESET, ON A NX DRIVE
016010  104417  000106                DELAY   ,106            ;TIME DELAY, 525 US ON 11/20
                                                                ;105 US ON 11/45
016014  105777  163310                TSTB    @RKER           ;DID NXD BIT IN RKER SET?
016020  001006                        BNE     3$              ;YES, BRANCH
016022  004737  021010                JSR     PC,GT2RG        ;GET RKCS, RKER
016026  012737  000200  001166        MOV     #200,$REG2      ;THIS BIT (NXD) IN RKER DID NOT SET
016034  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.
016036  022710  140214          3$:     CMP     #140214,@R0     ;DID HE & ERR SET WHEN NXD SET?
016042  001403                          BEQ     4$              ;YES BRANCH
016044  004737  021010                  JSR     PC,GT2RG        ;HE OR ERR BIT DID NOT SET
016050  104106                          ERROR   106             ;WHEN NXD WAS SIMULATED
                                                        ;CLEAR NXD, HE, ERR BITS
016052  104413                  4$:     CNT.RESET               ;GO, DO CONTROL RESET
                                                        ;THIS IS A CALL FOR THE 'CNTRL-
                                                        ;RESET' ROUTINE. A CONTROL RESET IS
                                                        ;ISSUED AND AFTER A CERTAIN TIME
                                                        ;IF THE 'CNTRL RDY' DOES NOT SET
                                                        ;AN ERROR IS REPORTED. NOTE THAT
                                                        ;THE PC IN ERROR MESSAGE IS THE
                                                        ;PC WHERE 'CNT.RESET' IS LOCATED.
                                                        ;THIS IS A VERY BASIC ERR& IF IT
                                                        ;OCCURS GO BACK TO TEST 10
016054  004737  021356                  JSR     PC,CHKECLR      ;CHECK IF 'NXD' BIT WAS CLEARED BY
                                                        ;CON.RESET. IF NOT, RETURN HERE.
016060  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
                                                        ;NXD BIT IN RKER
016062  004737  021402          5$:     JSR     PC,CHKCCLR      ;CHECK IF 'HE' & 'ERR' BITS WERE CLEARED
                                                        ;BY CON.RESET. IF NOT RETURN HERE.
016066  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
                                                        ;HE OR ERR BIT IN RKCS
016070  004737  021436                  JSR     PC,TSTRWS       ;GO CHECK & WAIT FOR R/W/S RDY
                                                        ;TO SET.  IF SET SKIP ERROR
016074  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
                                ;;*****************************************************************
016076  000004                  TST45:  SCOPE
016100  013700  001332                  MOV     RKCS,R0
016104  012737  177773  001362  2$:     MOV     #-5,COUNT       ;ALLOW 'ERROR 133' ONLY 5 TIMES
016112  013702  001350                  MOV     DRIVAD,R2       ;GET ADRES OF DRIVE
016116  052702  014540                  BIS     #14540,R2       ;SET BITS FOR CYL 313
016122  012737  016130  001110          MOV     #3$,$LPERR      ;SET RETURN ADRES FOR
                                                        ;LUPING ON EROR (SW9)
016130  104413                  3$:     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
016132  004737  021436                  JSR     PC,TSTRWS       ;GO CHECK & WAIT FOR R/W/S RDY
                                                        ;TO SET. IF SET SKIP ERROR BELOW
016136  104016                          ERROR   16              ;R/W/S RDY IS NOT SET
016140  104421                          TST.SIN                 ;CHECK IF SIN IS SET, IF SET
                                                        ;DO DRIVE RESET TO CLR IT
016142  010277  163172                  MOV     R2,@RKDA        ;ADRES DRIVE, NXC CYLINDER
016146  012710  000011                  MOV     #11,@R0         ;SEEK, GO TO NXC CYL
016152  104412                          CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
                                                        ;IF SO, SKIP THE EROR MESSAGE.
016154  104021                          ERROR   21              ;SEEK WAS TRIED TO A NON EXISTENT
                                                        ;CYLINDER, NXC SHOULD HAVE OCCURED
                                                        ;SETTING CNTRL RDY. BUT CNTRL RDY
                                                        ;DID NOT SET.
016156  032777  000100  163144  9$:     BIT     #100,@RKER      ;DID NXC SET?
016164  001020                          BNE     4$              ;YES, BRANCH
016166  004737  021010                  JSR     PC,GT2RG        ;GO GET RKCS, ER
016172  017737  163142  001166          MOV     @RKDA,$REG2     ;GET RKDA
016200  104110                          ERROR   110             ;NXC DID NOT SET WHEN SEEK
                                                        ;WAS TRIED TO CYLINDER AS INDICATED
                                                        ;IN RKDA
016202  004737  021436                  JSR     PC,TSTRWS       ;CHECK & WAIT FOR R/W/S RDY,
                                                        ;IF SET SKIP ERROR
016206  104016                          ERROR   16              ;R/W/S SHOULD BE SET
016210  104413                          CNT.RESET               ;GO DO CONTROL RESET
016212  004737  021504                  JSR     PC,DRESET       ;GO DO DRIVE RESET
016216  104026                          ERROR   26              ;NXC DID NOT SET AND DRIVE MAY
                                                        ;HAVE TRIED TO DO A SEEK, AFTER
                                                        ;WHICH R/W/S RDY DID NOT SET
016220  005237  001362                  INC     COUNT           ;ALLOW ONLY 5 MESSAGES FOR
016224  001405                          BEQ     5$              ;ERROR 133
016226  062702  000040          4$:     ADD     #40,R2          ;ADRES THE NXT CYL(IN NON-EXISTENT ZONE)
016232  032702  017740                  BIT     #17740,R2       ;CHKD FOR ALL NXC'S?
016236  001334                          BNE     3$              ;IF NOT, LUP BAK & CHK THE NXT NXC

016240  032710  140000          5$:     BIT     #140000,@R0     ;DID HE & ERR BIT SET WHEN NXC BIT SET?
016244  001003                          BNE     6$              ;YES, BRANCH
016246  004737  021010                  JSR     PC,GT2RG        ;GET RKCS, ER
016252  104106                          ERROR   106             ;HE OR ERR BIT DID NOT SET IN RKCS
                                                        ;WHEN NXC ERROR WAS SIMULATED
                                                        ;CLEAR HE, ERR, NXC BITS
016254  104413                  6$:     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
```

```
5286  016256  004737  021356                  JSR     PC,CHKECLR      ;CHECK IF 'NXC' BIT WAS CLEARED BY
5287                                                                    ;CON.RESET. IF NOT, RETURN HERE.
5288  016262  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
5289                                                                    ;NXC BIT IN RKER.
5290  016264  032710  140000          7$:     BIT     #140000,@R0     ;DID HE & ERR BITS GET CLEARED?
5291  016270  001405                          BEQ     TST46           ;;YES, EXIT
5292  016272  010037  001162                  MOV     R0,$REG0        ;GET ADRES OF RKCS
5293  016276  011037  001164                  MOV     @R0,$REG1       ;GET RKCS CONTENTS
5294  016302  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
5295                                                                    ;HE OR ERR BIT IN RKCS
5296
5297                                  ;;*****************************************************************
5298                                  ;*TEST 46        SIMULATE & CHECK NXS ERROR
5299                                          ;*THIS TEST SIMULATES NON-EXISTENT SECTOR ERROR & CHECKS THAT
5300                                          ;*IT IS DETECTED BY NXS BIT OF RKER.  IT IS CHECKED THAT
5301                                          ;*WHEN NXS SETS HE & ERR OF RKER ALSO SETS, AND ALL THREE
5302                                          ;*CAN BE CLEARED BY CONTROL RESET.
5303                                  ;;*****************************************************************
5304  016304  000004                  TST46:  SCOPE
5305  016306  104413                          CNT.RESET               ;GO, DO CONTROL RESET
5306                                                                    ;THIS IS A CALL FOR THE 'CNTRL-
5307                                                                    ;RESET' ROUTINE. A CONTROL RESET IS
5308                                                                    ;ISSUED AND AFTER A CERTAIN TIME
5309                                                                    ;IF THE 'CNTRL RDY' DOES NOT SET
5310                                                                    ;AN ERROR IS REPORTED. NOTE THAT
5311                                                                    ;THE PC IN ERROR MESSAGE IS THE
5312                                                                    ;PC WHERE 'CNT.RESET' IS LOCATED.
5313                                                                    ;THIS IS A VERY BASIC ERR& IF IT
5314                                                                    ;OCCURS GO BACK TO TEST 10
5315  016310  013700  001332                  MOV     RKCS,R0
5316  016314  013777  001350  163016          MOV     DRIVAD,@RKDA    ;GET ADRES OF DRIVE
5317  016322  052777  000014  163010          BIS     #14,@RKDA       ;SET BITS FOR SECTOR 12 (DECIMAL)
5318  016330  012777  177777  162776          MOV     #-1,@RKWC       ;READ 1 WORD
5319  016336  012777  033342  162772          MOV     #OUTBUF,@RKBA   ;INTO THIS BUS ADRES
5320  016344  012710  000005                  MOV     #5,@R0          ;READ, GO (FROM NX SECTOR)
5321  016350  104414                          CNT.RDY                 ;THIS IS A CALL FOR 'CN.RDY'
5322                                                                    ;ROUTINE WHICH WAITS FOR CNT
5323                                                                    ;RDY TO SET. IF CNTRL RDY DOES
5324                                                                    ;NOT SET WITHIN 883 MS/ 11-20
5325                                                                    ;(176 MS FOR 11-45 WITH BIPOLAR)
5326                                                                    ;AN ERROR IS REPORTED
5327                                                                    ;NXS ERROR SHOULD OCCUR NOW
5328  016352  017702  162752                  MOV     @RKER,R2
5329  016356  032702  000040                  BIT     #40,R2          ;DID NXS BIT SET IN RKER?
5330  016362  001006                          BNE     1$              ;YES, BRANCH
5331  016364  004737  021010                  JSR     PC,GT2RG        ;GO GET RKCS, RKER
5332  016370  012737  000040  001166          MOV     #40,$REG2       ;THIS BIT (NXS) IN RKER DID NOT SET
5333  016376  104105                          ERROR   105             ;NXS BIT DID NOT SET ON SIMULATING
5334                                                                    ;NXS ERROR
5335  016400  042702  000040          1$:     BIC     #40,R2          ;MASK NXS BIT
5336  016404  001407                          BEQ     2$              ;CHECK IF ANY OTHER
5337                                                                    ;RKER BIT SET
5338  016406  012737  000040  001160          MOV     #40,$REG0       ;GET EXPCTD RKER
5339  016414  017737  162710  001164          MOV     @RKER,$REG1     ;GET RKER RECVD
5340  016422  104107                          ERROR   107             ;ONLY 'NXS' SHOULD BE SET
5341                                                                    ;IN RKER, ANOTHER RKER BIT
```



```
5342                                                                    ;WAS SET. (NOTE 'NXS' WAS
5343                                                                    ;SIMULATED)
5344  016424  022710  140204          2$:     CMP     #140204,@R0     ;DID HE & ERR BITS SET?
5345  016430  001403                          BEQ     3$              ;YES, BRANCH
5346  016432  004737  021010                  JSR     PC,GT2RG        ;GO GET RKCS, RKER
5347  016436  104106                          ERROR   106             ;HE OR ERR BIT DID NOT SET WHEN
5348                                                                    ;NXS ERROR OCCURED
5349                                                                    ;CLEAR NXS, HE, ERR BITS
5350  016440  104413                  3$:     CNT.RESET               ;GO, DO CONTROL RESET
5351                                                                    ;THIS IS A CALL FOR THE 'CNTRL-
5352                                                                    ;RESET' ROUTINE. A CONTROL RESET IS
5353                                                                    ;ISSUED AND AFTER A CERTAIN TIME
5354                                                                    ;IF THE 'CNTRL RDY' DOES NOT SET
5355                                                                    ;AN ERROR IS REPORTED. NOTE THAT
5356                                                                    ;THE PC IN ERROR MESSAGE IS THE
5357                                                                    ;PC WHERE 'CNT.RESET' IS LOCATED.
5358                                                                    ;THIS IS A VERY BASIC ERR& IF IT
5359                                                                    ;OCCURS GO BACK TO TEST 10
5360  016442  004737  021356                  JSR     PC,CHKECLR      ;CHECK IF 'NXS' BIT WAS CLEARED BY
5361                                                                    ;CON.RESET. IF NOT, RETUEN HERE.
5362  016446  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
5363                                                                    ;NXS BIT IN RKER
5364  016450  004737  021402          4$:     JSR     PC,CHKCCLR      ;CHECL IF 'HE' & 'ERR' BITS WERE CLEARED
5365                                                                    ;BY CON.RESET. IF NOT, RETURN HERE.
5366  016454  104102                          ERROR   102             ;RKCS BITS ERR OR HE WERE NOT
5367                                                                    ;CLEARED BY CNTRL RESET
5368
5369                                  ;;*****************************************************************
5370                                  ;*TEST 47        SIMULATE & CHECK WCE
5371                                          ;*THIS TEST SIMULATES A WRITE CHECK ERROR AND CHECKS THAT IT
5372                                          ;*IS DETECTED BY WCE BIT OF RKER.  FOR COMPARISON IT USES
5373                                          ;*THE 256 WORDS DATA BLOCK WRITTEN ON SECTOR 0, CYLINDER 0
5374                                          ;*IN A PREVIOUS TEST.  THIS BLOCK IS COMPARED WITH THE 256 WORDS
5375                                          ;*MEMORY BUFFER STARTING AT 'OUTBUF'.  WCE IS SIMULATED BY
5376                                          ;*DROPPING A BIT FROM ONE OF THE WORDS IN THE MEMORY BUFFER.
5377                                  ;;*****************************************************************
5378  016456  000004                  TST47:  SCOPE
5379  016460  013700  001332                  MOV     RKCS,R0
5380  016464  104413                          CNT.RESET               ;GO, DO CONTROL RESET
5381                                                                    ;THIS IS A CALL FOR THE 'CNTRL-
5382                                                                    ;RESET' ROUTINE. A CONTROL RESET IS
5383                                                                    ;ISSUED AND AFTER A CERTAIN TIME
5384                                                                    ;IF THE 'CNTRL RDY' DOES NOT SET
5385                                                                    ;AN ERROR IS REPORTED. NOTE THAT
5386                                                                    ;THE PC IN ERROR MESSAGE IS THE
5387                                                                    ;PC WHERE 'CNT.RESET' IS LOCATED.
5388                                                                    ;THIS IS A VERY BASIC ERR& IF IT
5389                                                                    ;OCCURS GO BACK TO TEST 10
5390  016466  104421                          TST.SIN                 ;CHECK IF SIN IS SET, IF
5391                                                                    ;SET DO DRV-RESET TO CLR IT
5392  016470  012701  033342                  MOV     #OUTBUF,R1      ;THIS CODE SETS UP A MEMORY
5393  016474  012702  177400                  MOV     #-400,R2        ;BUFFER OF 256 WORDS STARTING
5394  016500  012703  177777                  MOV     #177777,R3      ;AT OUTBUF
5395                                                                    ;FIRST WORD  177400
5396                                                                    ;SECOND      177001
5397  016504  062703  177401          1$:     ADD     #177401,R3
```

```
5398  016510  010321                          MOV     R3,(R1)+        ;LAST WORD   000377
5399  016512  005202                          INC     R2              ;HAVE U GENERATED ALL 256 WORDS?
5400  016514  001373                          BNE     1$              ;IF NOT, LUP BAK & GENERATE NXT
5401
5402  016516  012737  170007  033360          MOV     #170007,OUTBUF+16 ;WCE WILL B SIMULATED BY DROPPING A
5403                                                                  ;BIT IN THE EIGHTH WORD WHICH IS
5404                                                                  ;SUPPOSED TO B 174007
5405  016524  012777  177400  162602          MOV     #-400,@RKWC     ;WRT CHK 400 WORDS
5406  016532  012777  033342  162576          MOV     #OUTBUF,@RKBA   ;STARTING AT THIS BUS ADRES
5407  016540  013777  001350  162572          MOV     DRIVAD,@RKDA    ;WITH THIS DISK ADRES, SEC 0, CYL 0
5408  016546  012710  000007                  MOV     #7,@R0          ;WRT CHK, GO
5409
5410  016552  104412                          CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
5411                                                                  ;IF SO, SKIP THE EROR MESSAGE.
5412  016554  104065                          ERROR   65              ;CNTRL RDY DID NOT SET
5413                                                                  ;AFTER WRT CHK
5414  016556  032777  000001  162544  3$:     BIT     #1,@RKER        ;DID WCE BIT SET?
5415  016564  001006                          BNE     4$
5416  016566  004737  021010                  JSR     PC,GT2RG        ;GO, GET RKCS, RKER
5417  016572  012737  000001  001166          MOV     #1,$REG2        ;THIS BIT (WCE) DID NOT SET
5418  016600  104105                          ERROR   105             ;WCE DID NOT SET ON SIMULATING
5419                                                                  ;WCE CONDITIONS
5420  016602  022710  100206          4$:     CMP     #100206,@R0     ;IS RKCS CORRECT?
5421  016606  001403                          BEQ     5$              ;YES, BRANCH
5422  016610  004737  021010                  JSR     PC,GT2RG        ;GO, GET RKCS, RKER
5423  016614  104106                          ERROR   106             ;HE OR ERR BIT DID NOT SET WHEN
5424                                                                  ;WCE WAS SIMULATED
5425  016616  104413          5$:     CNT.RESET                       ;CNTRL RESET
5426  016620  004737  021356                  JSR     PC,CHKECLR      ;WAS 'WCE' BIT CLEARED?
5427                                                                  ;IF NOT, RETURN HERE.
5428  016624  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
5429                                                                  ;WCE BIT IN RKER
5430  016626  004737  021402          6$:     JSR     PC,CHKCCLR      ;CHECK IF 'ERR' BIT WAS CLEARED. IF
5431                                                                  ;NOT RETURN HERE.
5432  016632  104102                          ERROR   102             ;CNTRL RESET DID NOT CLEAR
5433                                                                  ;RKCS
5434
5435                                  ;;*************************************************************
5436                                  ;*TEST 50        CHECK THAT SSE STOPS ALL CONTROL ACTION ON SOFT ERROR
5437                                          ;*THIS TEST CHECKS THAT WHEN 'STOP ON SOFT ERROR' BIT IS SET IN
5438                                          ;*RKCS AND A SOFT ERROR IS ENCOUNTERED ALL CONTROL ACTION WILL
5439                                          ;*STOP AT THE END OF THE CURRENT SECTOR IF IDE BIT IS CLEAR.
5440                                          ;*SOFT ERROR IS SIMULATED BY A WCE AS IN THE PREVIOUS
5441                                          ;*TEST. THE PREVIOUS TEST & THE TEST WHICH WRITES DATA
5442                                          ;*BLOCK ON CYLINDER 0, SECTOR 0, SHOULD BE DONE PRIOR
5443                                          ;*TO THIS TEST. A TWO SECTOR 'WRT CHK' WILL BE DONE.
5444                                          ;*CONTROL ACTION SHOULD STOP AFTER THE FIRST SECTOR DURING
5445                                          ;*WHICH A SOFT ERROR IS SIMULATED.
5446                                  ;;*************************************************************
5447  016634  000004          TST50:  SCOPE
5448  016636  104413                  CNT.RESET                       ;GO, DO CONTROL RESET
5449                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
5450                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
5451                                                                  ;ISSUED AND AFTER A CERTAIN TIME
5452                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
5453                                                                  ;AN ERROR IS REPORTED. NOTE THAT
```



```
5454                                                                  ;THE PC IN ERROR MESSAGE IS THE
5455                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
5456                                                                  ;THIS IS A VERY BASIC ERR& IF IT
5457                                                                  ;OCCURS GO BACK TO TEST 10
5458  016640  104421                          TST.SIN                 ;CHECK IF SIN IS SET, IF
5459                                                                  ;SET DO DRIVE RESET TO CLR IT
5460  016642  013700  001332                  MOV     RKCS,R0
5461  016646  012737  170007  033360          MOV     #170007,OUTBUF+16 ;WCE IS SIMULATED BY DROPPING A BIT
5462                                                                  ;IN THE EIGTH WORD (WHICH IS ACTUALLY
5463                                                                  ;174007). NOTE THAT 256 WORD MEMORY
5464                                                                  ;BUFFER IS CREATED IN THE PREVIOUS TEST.
5465  016654  013701  001350                  MOV     DRIVAD,R1
5466  016660  012777  177000  162446          MOV     #-1000,@RKWC    ;WRT CHK 1000 (OCTAL) WORDS, 2 SECTORS
5467  016666  012777  033342  162442          MOV     #OUTBUF,@RKBA   ;FROM THIS BUS ADRES
5468  016674  010177  162440                  MOV     R1,@RKDA        ;WITH THIS DISK ADRES, SEC 0, CYL 0
5469  016700  012710  000407                  MOV     #407,@R0        ;WRT CHK, GO, SSE
5470  016704  104412                          CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
5471                                                                  ;IF SO, SKIP THE EROR MESSAGE.
5472  016706  104065                          ERROR   65              ;CNTRL RDY DID NOT SET AFTER WRT
5473                                                                  ;CHK. A SOFT ERROR (WCE) IN
5474                                                                  ;SECTOR 0 SHOULD HAVE STOPPED
5475                                                                  ;ALL CONTROL ACTION.
5476  016710  022777  000001  162412  2$:     CMP     #1,@RKER        ;CHECK ONLY 'WCE' BIT SHOULD
5477                                                                  ;BE SET?
5478  016716  001407                          BEQ     3$              ;YES, BRANCH
5479  016720  012737  000001  001162          MOV     #1,$REG0        ;GET EXPCTD RKER
5480  016726  017737  162376  001164          MOV     @RKER,$REG1     ;GET RKER RECVD
5481  016734  104107                          ERROR   107             ;ONLY BIT 'WCE' OF RKER
5482                                                                  ;SHOULD BE SET (WCE WAS
5483                                                                  ;SIMULATED ABOVE).  ERROR
5484                                                                  ;IF IT'S NOT
5485  016736  005201          3$:     INC     R1                      ;CHECK THAT RKDA INCREMENTED BY
5486  016740  020177  162374                  CMP     R1,@RKDA        ;1 SECTOR ONLY IMPLYING THAT
5487                                                                  ;CNTRL ACTION DID STOP AFTE.
5488                                                                  ;SOFT ERROR IN SECTOR 0
5489  016744  001406                          BEQ     TST51           ;;YES, EXIT
5490  016746  010137  001162                  MOV     R1,$REG0        ;GET EXPCTD RKDA
5491  016752  017737  162362  001164          MOV     @RKDA,$REG1     ;GET RKDA RECVD
5492  016760  104070                          ERROR   70              ;RKDA SHOULD HAVE INCRMNTD
5493                                                                  ;BY 1 SECTOR ONLY, IT DIDN'T.
5494                                                                  ;WCE WAS SIMULATED IN THE
5495                                                                  ;FIRST SECTOR & A WRT CHK
5496                                                                  ;OF 2 SECTORS WAS ISSUED.
5497                                                                  ;CONTROLLER SHOULD STOP AFTER
5498                                                                  ;DETECTING WCE IN THE FIRST
5499                                                                  ;SECTOR.  HENCE RKDA SHOULD
5500                                                                  ;INCREMENT BY 1 SECTOR ONLY
5501
5502
5503                                  ;;*************************************************************
5504                                  ;*TEST 51        CHECK THAT RK11 INTERRUPTS ON SOFT ERROR WHEN SSE & IDE ARE SET
5505                                          ;*THIS TEST CHECKS WHEN SSE BIT IS SET WITH IDE SET AND A SOFT
5506                                          ;*ERROR OCCURS, THEN ALL CONTROL ACTION WILL STOP AND A BUS
5507                                          ;*REQUEST (INTERRUPT) WILL OCCUR AT THE END OF THE CURRENT
5508                                          ;*SECTOR.  SOFT ERROR IS SIMULATED BY WCE AS IN PREVIOUS
5509                                          ;*TEST.  PREREQUISITES FOR THIS TEST ARE THE SAME AS THOSE
```

```
                        ;*FOR THE PREVIOUS TEST.
                        ;;*****************************************************************
016762  000004         TST51:  SCOPE
016764  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
016766  104421                 TST.SIN                ;CHECK IF SIN IS SET, IF
                                                       ;SET DO DRIVE RESET TO CLR IT
016770  012737  170007  033360         MOV     #170007,OUTBUF+16 ;WCE IS SIMULATED BY DROPPING A BIT
                                                       ;IN THE EIGHTH WORD (WHICH IS 174007)
                                                       ;NOTE THAT THE 256 WORD MEMORY
                                                       ;BUFFER (STARTING AT OUTBUF) IS
                                                       ;CREATED IN A PREVIOUS TEST.
016776  013701  001350         MOV     DRIVAD,R1
017002  012777  177000  162324 MOV     #-1000,@RKWC    ;WRT CHK 1000 (OCTAL) WORDS, 2 SECTORS
017010  012777  033342  162320 MOV     #OUTBUF,@RKBA   ;FROM THIS BUS ADRES
017016  010177  162316         MOV     R1,@RKDA        ;WITH THIS DISK ADRES, SEC 0, CYL 0
017022  013700  001402         MOV     RKVEC,R0
017026  012720  017060         MOV     #1$,(R0)+       ;SET UP INTERRUPT VECTOR FOR RK11
017032  012710  000340         MOV     #340,@R0        ;SET PSW ON INTERRUPT
017036  012777  000507  162266 MOV     #507,@RKCS      ;WRT CHK, GO. SSE, IDE SET
017044  104420  177777         WAT.INT,177777         ;WAIT FOR INTERRUPT FROM RK11
                                                       ;TIME=485 MS FOR 11/20.
                                                       ;97 MS FOR 11/45
017050  004737  021010         JSR     PC,GT2RG               ;11/05
017054  104111                 ERROR   111             ;RK11 DID NOT INTERRUPT AFTER A SOFT
                                                       ;ERROR (SIMULATED) IN  SECTOR 0
017056  000417                 BR      2$

017060  022626         1$:     CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER (FROM RK11 INTRUPT)
017062  022626                 CMP     (SP)+,(SP)+     ;POP STACK (FROM WAT.INT)
017064  012777  004600  162310 MOV     #BADINT,@RKVEC  ;RESTORE RK11 INTERRUPT VECTOR
                                                       ;ADRES FOR UNEXPECTED INTERRUPTS
017072  005201                 INC     R1
017074  020177  162240         CMP     R1,@RKDA        ;CHECK THAT RKDA INCREMENTED
                                                       ;BY ONLY 1 SECTOR BEFORE INTERRUPT
                                                       ;OCCURRED
017100  001406                 BEQ     2$
017102  010137  001162         MOV     R1,$REG0        ;GET EXPCTD RKDA
017106  017737  162226  001164 MOV     @RKDA,$REG1     ;GET RKDA RECVD
017114  104003                 ERROR   3               ;RKDA SHOULD HAVE INCREMENTED BY
                                                       ;1 SECTOR ONLY, IF ALL CNTRL ACTION
                                                       ;HAD STOPPED AFTER SOFT ERROR
                                                       ;(SIMULATED) IN SECTOR 0. IT DID NOT.
017116          2$:
017116  012746  000340         MOV     #340,-(SP)
017122  012746  017130         MOV     #64$,-(SP)
017126  000002                 RTI
017130          64$:
```



```
017130  005077  162176         CLR     @RKCS           ;CLEAR THE IDE BIT


                        ;;*****************************************************************
                        ;*TEST 52        CHECK THE MEX BITS IN RKCS
                                ;*THIS TEST CHECKS OUT THE EXTENDED MEMORY BITS OF THE RKCS.
                                ;*THE RKBA IS SET TO 177776 AND A ONE WORD WRITE CHECK IS TRIED.
                                ;*THIS COULD GIVE RISE TO NXM ERROR, BUT EVEN THEN THE RKBA
                                ;*SHOULD OVERFLOW INTO THE MEX BITS. SIMILIARLY IT IS CHECKED
                                ;*THAT THE OVERFLOWING BIT CAN MAKE THE MEX BITS COUNT
                                ;*01,10,11,00.
                        ;;*****************************************************************
017134  000004         TST52:  SCOPE
017136  013700  001332         MOV     RKCS,R0
017142  012701  177774         MOV     #-4,R1          ;SET UP THE COUNT
017146  005002                 CLR     R2              ;INITIALIZE MEX BITS TO B SET IN RKCS
017150  012737  017156  001110 MOV     #1$,$LPERR      ;SET RETURN ADRES FOR
                                                       ;LUPING ON EROR (SW9)
017156  104417  000142 1$:     DELAY   ,142            ;TIME DELAY
017162  004737  021436         JSR     PC,TSTRWS       ;WAIT FOR R/W/S RDY
017166  104016                 ERROR   16              ;R/W/S RDY IS NOT SET
017170  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
017172  010210                 MOV     R2,@R0          ;SET MEX BITS (AS IN R2) IN RKCS
017174  012777  177777  162132 MOV     #-1,@RKWC       ;WRT CHK 1 WORD
017202  013777  001350  162130 MOV     DRIVAD,@RKDA    ;THIS DISK ADRES, SEC 0, CYL 0
017210  012777  177776  162120 MOV     #177776,@RKBA   ;THIS BUS ADRES. NOTE THIS BA
                                                       ;IN CONJUCTION WITH MEX BITS OF RKCS
017216  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
017222  104412                 CHKCRDY                ;GO CHECK IF CONTROL RDY IS SET
                                                       ;IF SO, SKIP THE EROR MESSAGE.
017224  104065                 ERROR   65              ;CNTRL RDY DID NOT SET AFTER WRT CHK
017226  010205         3$:     MOV     R2,R5
017230  062705  000020         ADD     #20,R5          ;MEX BITS SHOULD INCREMENT BY 1 TO THIS
017234  042705  000100         BIC     #100,R5         ;MASK OUT IDE BIT POSITION, IF SET
017240  011004                 MOV     @R0,R4          ;GET RKCS
017242  042704  177717         BIC     #177717,R4      ;MASK OUT ALL BITS EXCEPT MEX
017246  020504                 CMP     R5,R4           ;DID MEX BITS INCREMENT CORRECTLY?
017250  001405                 BEQ     4$              ;YES, BRANCH
017252  010537  001162         MOV     R5,$REG0        ;GET EXPCTD MEX BITS
017256  010437  001164         MOV     R4,$REG1        ;GET MEX BITS RECVD
017262  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.
017264  017703  162040          4$:   MOV   @RKER,R3      ;GET RKER
017270  010305                        MOV   R3,R5
017272  042703  003001                BIC   #3001,R3      ;MASK WCE,DLT,NXM BIT, IF SET
017276  001410                        BEQ   5$            ;BRANCH IF REST OF RKER CLR
017300  042705  177776                BIC   #177776,R5    ;MASK NON-WCE BITS
017304  010537  001162                MOV   R5,$REG0      ;THIS IS THE EXPCTD RKER
017310  017737  162014  001164        MOV   @RKER,$REG1   ;GET RKER RECVD
017316  104107                        ERROR 107           ;ERROR IN RKER.  IT SHOULD
                                                        ;BE AS EXPECTED IN
                                                        ;ERROR MESSAGE
017320  062702  000020          5$:   ADD   #20,R2        ;INCREMENT TO NXT MEX BIT
017324  005201                        INC   R1            ;HAVE U CHKD THE MEX BITS 4 TIMES?
017326  001313                        BNE   1$            ;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.
                                ;;*************************************************************
017330  000004                  TST53:  SCOPE
017332  012737  000001  001206        MOV   #1,$TIMES     ;;DO 1 ITERATION
                                                        ;THIS CODE FINDS OUT IF THE CPU
                                                        ;IS AN 11/05 OR ELSE.
                                                        ;ON AN 11/05, R0 (177700) CAN BE
                                                        ;ADDRESSED AS A MEMORY LOCATION, BUT
                                                        ;ON ANY OTHER CPU IF 177700 IS REFERENCED
                                                        ;A TIME OUT WILL OCCUR.
017340  012737  017362  000004        MOV   #5$,@#4       ;SET UP TIME OUT VECTOR
017346  005737  177700                TST   @#177700      ;REFERENCE R0
017352  012737  004534  000004        MOV   #BADTMO,@#4   ;R0 WAS REFERENCED W/O TIMEOUT
                                                        ;HENCE 11/05
017360  000520                        BR    TST54         ;;SKIP THIS TEST
017362  022626                  5$:   CMP   (SP)+,(SP)+   ;RESTORE STACK POINTER
017364  012737  004534  000004        MOV   #BADTMO,@#4   ;RESTORE TIMEOUT VECTOR
017372  012746  000340                MOV   #340,-(SP)
017376  012746  017404                MOV   #64$,-(SP)
017402  000002                        RTI
017404                          64$:
017404  013700  001332                MOV   RKCS,R0
017410  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
017412  012701  033342                MOV   #OUTBUF,R1
```



```
017416  013704  001336                MOV   RKBA,R4
017422  012711  000100                MOV   #100,@R1      ;WRITE THIS WORD
017426  012777  177777  161700        MOV   #-1,@RKWC     ;WRITE 1 WORD
017434  013702  001350                MOV   DRIVAD,R2
017440  052702  000002                BIS   #2,R2         ;ON CYL 0, SEC 2
017444  010277  161670                MOV   R2,@RKDA
017450  010114                        MOV   R1,@R4        ;FROM THIS MEMORY LOC
017452  012710  000003                MOV   #3,@R0        ;WRITE, GO
017456  005003                        CLR   R3
017460  105710                  1$:   TSTB  @R0
017462  100410                        BMI   2$
017464  005203                        INC   R3
017466  001374                        BNE   1$
017470  004737  020774                JSR   PC,GT4RG      ;GET RKCS, ER, DS
017474  010237  001202                MOV   R2,$REG10     ;GET THE STARTING ADRES
017500  104416                        BRKDA4              ;BREAK IT INTO DRV #, CYL, SUR, SEC #
017502  104031                        ERROR 31            ;CNTRL RDY DID NOT SET AFTER
                                                        ;WRITE OF 1 WORD ON CYL 0, SEC 2
017504  012777  177777  161622  2$:   MOV   #-1,@RKWC     ;READ 1 WORD
017512  010277  161622                MOV   R2,@RKDA      ;FROM SEC 2, CYL 0
017516  013714  001144                MOV   $TKS,@R4      ;INTO TTY STAUS REGISTER
017522  005077  161416                CLR   @$TKS         ;CLEAR TTY KEY BRD STATUS REG

017526  012710  000065                MOV   #65,@R0       ;READ, MEX BITS SET
017532  005003                        CLR   R3
017534  105710                  3$:   TSTB  @R0
017536  100410                        BMI   4$
017540  005203                        INC   R3
017542  001374                        BNE   3$
017544  004737  020774                JSR   PC,GT4RG
017550  010237  001202                MOV   R2,$REG10     ;GET THE STARTING ADRES
017554  104416                        BRKDA4              ;BREAK IT INTO DR#, CYL, SUR, SEC#
017556  104045                        ERROR 45            ;CNTRL RDY DIDN'T SET AFTER
                                                        ;READ OF 1 WORD FROM CYL 0, SEC 2.
                                                        ;IN ERROR MSGE, <DSK-ADRES> GIVES
                                                        ;ADRES WHERE READ BEGAN. 'RKDA'
                                                        ;GIVES CONTENTS OF RKDA AT TIME OF ERROR
017560  032737  000100  001144  4$:   BIT   #100,$TKS     ;WAS THE CORRECT WORD READ INTO
                                                        ;THE TTY  STATUS REGISTER?
017566  001015                        BNE   TST54         ;;YES, EXIT
017570  017705  161350                MOV   @$TKS,R5              ;GET THE WORD RECVD FROM DISK
017574  010537  001164                MOV   R5,$REG1
017600  052705  000100                BIS   #100,R5       ;THIS WORD WAS EXPCTD
017604  010537  001162                MOV   R5,$REG0      ;STORE EXPCTD WORD
017610  011437  001166                MOV   @R4,$REG2     ;GET RKBA
017614  011037  001170                MOV   @R0,$REG3     ;GET RKCS
017620  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.
                                ;;*************************************************************
017622  000004                  TST54:  SCOPE
017624  012737  000005  001206          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
017632  104421                          TST.SIN                 ;CHECK IF SIN SET, IF SET DRV RESET
017634  005001                          CLR     R1              ;INITIALIZE (VALUE OF RKBA)
017636  012702  000002                  MOV     #2,R2           ;INITIALIZE (INCMNTD VALUE OF RKBA)
                                
017642  012737  017654  001110          MOV     #1$,$LPERR      ;SET RETURN ADRES FOR LUPING
                                                                ;ON EROR
                                
017650  013705  001336                  MOV     RKBA,R5
017654  004737  021436          1$:     JSR     PC,TSTRWS       ;WAIT FOR R/W/S RDY
017660  104016                          ERROR   16              ;R/W/S RDY IS NOT SET
017662  104413                          CNT.RESET               ;DO CONTROL RESET
017664  012777  177777  161442          MOV     #-1,@RKWC       ;WRITE CHK 1 WORD
017672  010115                          MOV     R1,@R5  ;THIS BUS ADRES
017674  013777  001350  161436          MOV     DRIVAD,@RKDA    ;SET DISK ADRES
017702  012777  000067  161422          MOV     #67,@RKCS       ;WRITE CHECK, GO, MEX BITS SET
017710  104412                          CHKCRDY                 ;GO CHECK IF CONTROL RDY IS SET
                                                                ;IF SO, SKIP THE EROR MESSAGE.
017712  104065                          ERROR   65              ;CNTRL RDY DID NOT SET AFTER
                                                                ;WRT CHK WAS TRIED TO NXM LOC
                                                                ;U MIGHT WANT TO USE TESTS
                                                                ;CHECKING MEX BITS & NXM.
017714  005237  001356                  INC     INDX1           ;ALLOW ONLY 5 ERRORS OF ABOVE KIND
017720  001417                          BEQ     5$
                                
017722  020215                  3$:     CMP     R2,@R5          ;DID RKBA INCREMENT BY 1 FROM
                                                                ;ITS INITIAL VALUE?
017724  001410                          BEQ     4$              ;YES, BRANCH
017726  010137  001162                  MOV     R1,$REG0        ;GET EXPCTD RKBA
017732  011537  001164                  MOV     @R5,$REG1       ;GET RKBA RECVD
017736  104017                          ERROR   17              ;RKBA DID NOT INCREMENT BY
                                                                ;1 FROM ITS INITIAL VALUE.
                                                                ;ONE WORD WRT CHK WAS TRIED
                                                                ;TO A NXM LOCATION. THERE
                                                                ;WILL BE AN NXM ERROR,
                                                                ;BUT STILL RKBA SHOULD
                                                                ;INCREMENT BY 1 FROM ITS
                                                                ;INITIAL VALUE.
017740  005237  001360                  INC     INDX2           ;ALLOW ONLY 5 ERRORS OF
017744  001405                          BEQ     5$              ;THE ABOVE KIND
017746  060201                  4$:     ADD     R2,R1           ;SET NXT VALUE OF RKBA
017750  010102                          MOV     R1,R2
017752  062702  000002                  ADD     #2,R2           ;SET EXPCTD VALUE OF RKBA
017756  001336                          BNE     1$              ;ALL DONE?
                                
017760                          5$:                             ;DUMMY EXIT POINT
```



```
                                ;;*************************************************************
                                ;*TEST 55        CHECK FOR RK-05F
                                        ;*THIS TEST CHECKS RK-05F TYPE DRIVES
                                        ;*TO INSURE THAT IF SEEKS ARE ISSUED ON ONE
                                        ;*DRIVE, THE OTHER DRIVE BECOMES BUSY
                                ;;*************************************************************
017760  000004                  TST55:  SCOPE
017762  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
017770  005737  001404                  TST     FFLAG           ;SEE IF RK-05F
017774  001403                          BEQ     1$              ;NOT F
017776  004537  025160                  JSR     R5,FCHECK       ;SEE IF OTHER GOES BUSY
020002  104120                          ERROR   120
                                
020004                          1$:
                                
                                ;;*************************************************************
                                ;*TEST 56        END OF PROGRAM
                                        ;*THIS IS NOT A TEST, BUT A LINKAGE PROVIDED TO PERFORM
                                        ;*THE ABOVE SUB-TESTS FOR ALL DRIVES THAT ARE PRESENT.
                                        ;*NOTE THAT THE NEXT TEST- HARDWARE POLLING LOGIC-
                                        ;*IS DONE USING ALL THE DRIVES THAT ARE INDICATED PRESENT.
                                        ;*DO NOT LOOP ON THIS 'TEST'.
                                ;;*************************************************************
020004  000004                  TST56:  SCOPE
020006  012737  000001  001206          MOV     #1,$TIMES       ;;DO 1 ITERATION
020014  005237  001352                  INC     DRVDON          ;INCREMENT THE COUNT FOR THE NUMBER
                                                                ;OF DRIVES THAT ARE CHECKED
020020  004737  021504                  JSR     PC,DRESET       ;RESET THE DRIVE
020024  104026                          ERROR   26              ;R/W/S DIDN'T SET AFTER DRIVE RESET
020026  023737  001412  001352  BTEOP:  CMP     DRIVS,DRVDON    ;HAVE U TESTED ALL THE DRIVES
                                                                ;THAT ARE PRESENT?
020034  001405                          BEQ     1$              ;IF YES, EXIT
020036  062737  020000  001350          ADD     #20000,DRIVAD   ;ADRES THE NXT POSSIBLE DRIVE
020044  000137  005040                  JMP     NUDRV           ;GO BACK AND TEST THE NEXT
                                                                ;DRIVE PRESENT
020050  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.
                                ;;*************************************************************
020054  000004                  TST57:  SCOPE
```

```
5846  020056  012737  000005  001206          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
5847  020064  005237  001440                  INC     SIZYET          ;FOUNR RK05F YET?
5848  020070  001002                          BNE     25$             ;YES
5849  020072  004737  025304                  JSR     PC,SIZEF        ;FIND WHICH ARE RK-05F
5850  020076  005037  001436          25$:    CLR     PHYDRV          ;NUMBER OF ACTUAL DRIVES
5851  020102  012700  001414                  MOV     #DRIV0,R0       ;TABLE
5852  020106  005710                  23$:    TST     (R0)            ;DRIVE HERE+?
5853  020110  001405                          BEQ     22$             ;NO
5854  020112  005237  001436                  INC     PHYDRV          ;COUNT DRIVE
5855  020116  005710                          TST     (R0)            ;RK05F?
5856  020120  100001                          BPL     22$             ;NO
5857  020122  005720                          TST     (R0)+           ;DONT COUNT F TWICE
5858  020124  005720                  22$:    TST     (R0)+   ;NEXT DRIVE
5859  020126  020027  001433                  CMP     R0,#DRIV7+1     ;ALL YET
5860  020132  002765                          BLT     23$             ;NO
5861  020134  005037  001406                  CLR     ODDEVN          ;EVEN DRIVES FIRST IF F
5862  020140  005737  001412          T56:    TST     DRIVS           ;ANY DRIVES PRESENT?
5863  020144  001002                          BNE     20$             ;YES
5864  020146  000137  020652                  JMP     $EOP            ;NO
5865  020152  005237  001434          20$:    INC     T56FLG
5866  020156  013700  001332                  MOV     RKCS,R0
5867  020162  005037  001356                  CLR     INDX1                   ;FLAG TO INDICATE:
5868                                                                  ;(INDX1)=0 POLLING DONE AFTER ALL
5869                                                                          ;DRIVES SEEK TO CYL 0
5870                                                                  ;(INDX1)=1 POLLING DONE AFTER ALL
5871                                                                          ;DRIVES SEEK TO CYL 4
5872  020166  005037  001360          15$:    CLR     INDX2           ;FLAG INDICATING TYPE OF INTERRUPT
5873                                                                  ;SET TO NON-ZERO TO INDICATE
5874                                                                  ;THAT THE INTERRUPT IS DUE TO
5875                                                                  ;SEEK DONE
5876  020172  104413                          CNT.RESET               ;GO, DO CONTROL RESET
5877                                                                  ;THIS IS A CALL FOR THE 'CNTRL-
5878                                                                  ;RESET' ROUTINE. A CONTROL RESET IS
5879                                                                  ;ISSUED AND AFTER A CERTAIN TIME
5880                                                                  ;IF THE 'CNTRL RDY' DOES NOT SET
5881                                                                  ;AN ERROR IS REPORTED. NOTE THAT
5882                                                                  ;THE PC IN ERROR MESSAGE IS THE
5883                                                                  ;PC WHERE 'CNT.RESET' IS LOCATED.
5884                                                                  ;THIS IS A VERY BASIC ERR& IF IT
5885                                                                  ;OCCURS GO BACK TO TEST 10
5886  020174  005737  001356                  TST     INDX1           ;PERFORMING SEEKS TO CYL 4
5887  020200  001002                          BNE     .+6             ;YES, BRANCH
5888  020202  005002                          CLR     R2              ;NO
5889  020204  000402                          BR      .+6
5890  020206  012702  000200                  MOV     #200,R2         ;SET ADRES FOR FOURTH CYLINDER
5891  020212  012701  001414                  MOV     #DRIV0,R1       ;INITIALIZE POINTER
5892  020216  012703  177770                  MOV     #-10,R3         ;SET COUNT FOR 8 DRIVES
5893  020222  012705  033342                  MOV     #OUTBUF,R5      ;INITIALIZE POINTER TO INDICATOR AREA
5894  020226  005025                          CLR     (R5)+           ;CLEAR OUT THE 8-WORD INDICATOR
5895  020230  005203                          INC     R3              ;AREA WHICH IS USED FOR DOING
5896  020232  001375                          BNE     .-4             ;SOFTWARE POLLING LATER ON
5897  020234  012703  177770                  MOV     #-10,R3         ;SET COUNT FOR 8 POSSIBLE DRIVES
5898  020240  012705  033342                  MOV     #OUTBUF,R5      ;INITIALIZE POINTER TO INDICATOR AREA
5899  020244                          1$:
5900  020244  012746  000340                  MOV     #340,-(SP)
5901  020250  012746  020256                  MOV     #64$,-(SP)
```



```
5902  020254  000002                          RTI
5903  020256                          64$:
5904  020256  032711  000001                  BIT     #BIT0,(R1)      ;IS THIS DRIVE PRESENT?
5905  020262  001433                          BEQ     4$              ;IF NOT, BRANCH
5906  020264  005711                          TST     (R1)            ;RK06F?
5907  020266  100012                          BPL     17$             ;NO, CONTINUE
5908  020270  032702  020000                  BIT     #BIT13,R2       ;DRIVE EVEN?
5909  020274  001404                          BEQ     16$             ;YES
5910  020276  005737  001406                  TST     ODDEVN          ;DO WE WANT ODD?
5911  020302  001423                          BEQ     4$              ;NO, SO DO NOT TEST
5912  020304  000403                          BR      17$             ;ADD THIS DRIVE TO LIST
5913  020306  005737  001406          16$:    TST     ODDEVN          ;DO WE WANT EVEN?
5914  020312  001017                          BNE     4$              ;NO, SO SKIP
5915  020314  010215                  17$:    MOV     R2,(R5)         ;SET UP THIS WORD IN THE
5916                                                                  ;INDICATOR AREA SHOWING THAT THIS
5917                                                                  ;DRIVE (AS IN BITS 13-15 OF R2)
5918                                                                  ;IS PRESENT
5919  020316  042725  017777                  BIC     #17777,(R5)+    ;MASK OUT UNWANTED BITS (CYL,SUR,SEC BITS)
5920  020322  005004                          CLR     R4
5921  020324  105710                  2$:     TSTB    @R0             ;IS CNTRL RDY SET?
5922  020326  100405                          BMI     3$              ;YES, BRANCH
5923  020330  005204                          INC     R4              ;NO, WAIT FOR IT
5924  020332  001374                          BNE     2$              ;IF WAITED LONG REPORT ERROR
5925  020334  004737  020774                  JSR     PC,GT4RG                ;GO, GET RKCS,ER,DS,DA
5926  020340  104021                          ERROR   21              ;CNTRL RDY DID NOT SET AFTER ACCEPTING
5927                                                                  ;ADRES FROM PREVIOUS SEEK
5928  020342  010277  160772          3$:     MOV     R2,@RKDA        ;ADRES THIS DRIVE, CYL 0 OR CYL 4
5929                                                                  ;(WHICHEVER THE CASE MAY BE)
5930  020346  012710  000111                  MOV     #111,@R0        ;SEEK,GO,IDE SET
5931  020352  005721                  4$:     TST     (R1)+           ;NEXT DRIVE DATA
5932  020354  062702  020000                  ADD     #20000,R2       ;INCREMENT DRIVE ADRES (BITS 15,14,13)
5933  020360  005203                          INC     R3              ;TO NEXT ONE
5934  020362  001330                          BNE     1$              ;BRANCH BACK IF ALL DRIVES ARE
5935                                                                  ;NOT CHECKED TO SEE IF THE NEXT
5936                                                                  ;DRIVE IS PRESENT (& IF SO ISSUE A
5937                                                                  ;SEEK TO IT)
5938                                                                  ;BY NOW SEEKS HAVE BEEN ISSUED
5939                                                                  ;TO ALL DRIVES PRESENT & POLLING
5940                                                                  ;HAS BEGUN
5941  020364  005004                          CLR     R4
5942  020366  013702  001402          5$:     MOV     RKVEC,R2
5943  020372  012722  020424                  MOV     #6$,(R2)+       ;SET ADRES FOR RK11 TO INTERUPT
5944  020376  012712  000340                  MOV     #340,(R2)       ;SET PSW ON INTERUPT
5945  020402  013746  001400                  MOV     RKPRI,-(SP)     ;DROP CPU PRIORITY TO 4 SO THAT
5946  020406  012746  020414                  MOV     #18$,-(SP)              ;RK11 CAN INTERUPT
5947  020412  000002                          RTI
5948  020414  000240          18$:    NOP                             ;THIS IS A TIME LOOP DURING
5949  020416  005204                  INC     R4                      ;WHICH ALL DRIVES PRESENT SHOULD
5950  020420  001375                  BNE     18$                     ;INTERRUPT
5951  020422  000452                  BR      11$                     ;BRANCH AND CHECK IF ALL AVAILABLE
5952                                                                  ;DRIVES INTERRUPTED CORRECTLY
5953  020424  022626          6$:     CMP     (SP)+,(SP)+             ;RESTORE STACK POINTER
5954  020426  005737  001360          TST     INDX2                   ;WAS THIS FIRST INTERRUPT
5955                                                                  ;DUE TO 'ADRES ACK' AFTER INITIATION
5956                                                                  ;OF SEEK?
5957  020432  001021                  BNE     9$                      ;IF YES, CHECK THE FOLLOWING
```

```
                                BIT     #20000,@R0      ;CHECK THAT SCP IS NOT SET
                                BEQ     7$              ;BRANCH IF SCP CLEAR
                                MOV     @R0,$REG0       ;GET RKCS
                                ERROR   76              ;AFTER THE FIRST INTERRUPT WHICH
                                                        ;IS DUE TO INITIATION OF SEEK, SCP
                                                        ;SHOULD NOT HAVE SET. IT DID
                        7$:     MOV     @RKDS,R1
                                BIT     #160000,R1      ;RKDS BITS 15-13 SHLOULD BE CLR
                                BEQ     8$
                                MOV     R1,$REG0        ;GET RKDS
                                ERROR   50              ;SEEK, WITH IDE SET WAS ISSUED TO
                                                        ;ALL AVAILABLE DRIVES. THE FIRST
                                                        ;INTERUPT IS DUE TO SEEK INITIATED
                                                        ;BY FRST DRV. DRV ID BITS 13-15
                                                        ;SHOULD BE CLR AFTR THIS FRST INRUPT.
                                                        ;THEY WERE NOT IF THIS ERROR OCCURS.
                        8$:     INC     INDX2           ;SET UP FLAG INDICATING
                                                        ;THAT THE FIRST INTERRUPT DUE
                                                        ;TO INITIATION OF SEEK WAS
                                                        ;PROCESSED
                                BR      5$              ;GO BACK TO THE WAIT LOOP & WAIT
                                                        ;FOR NEXT INTERRUPT FROM RK11
                        9$:     MOV     PHYDRV,R3       ;SET COUNT OF # OF DRIVES PRESENT
                                MOV     #OUTBUF,R5      ;INITIALIZE POINTER
                                MOV     @RKDS,R1        ;GET RKDS
                                BIC     #17777,R1       ;MASK BITS 0-12
                                                        ;THE FOLLOWING CODE IS A SOFTWARE
                                                        ;POLL WHICH FINDS OUT WHICH DRIVE
                                                        ;CAUSED THE PRESENT INTERRUPT
                                                        ;AND SETS UP A FLAG BIT FOR
                                                        ;THE DRIVE #, INDICATING THAT
                                                        ;THIS DRIVE # INTERRUPTED
                                CMP     R1,(R5)+
                                BEQ     10$             ;BRANCH IF INTERRUPTING DRIVE WAS FOUND
                                DEC     R3              ;HAVE U CHKD ALL DRIVS PRESENT?
                                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
                                MOV     R1,-(R6)        ;GET WORD TO B SHFTD RT
                                JSR     PC,SHFTRT       ;GO SHIFT IT
                                MOV     (R6)+,$REG0     ;THIS DRIVE # WAS RECVD IN RKDS AS
                                                        ;THE INTERRUPTING DRIVE, BUT THIS
                                                        ;DRIVE IS NOT PHYSICALY PRESENT
                                ERROR   51              ;RKDS INDICATES AN INTERRUPTING
                                                        ;DRIVE # (DURING H'WARE POLL) BUT
                                                        ;THAT DRIVE IS ACTUALLY NOT PRESENT
                                BR      10$+2
                        10$:    INC     -(R5)           ;SET UP FLAG INDICATING THAT
                                                        ;THE INTERRUPT FOR THIS DRIVE
                                                        ;(AFTER IT HAD COMPLETED ITS SEEK)
                                                        ;WAS PROCESSED
                                BR      5$              ;GO BAK & WAIT FOR FURTHER INTRUPTS
                        11$:    MOV     PHYDRV,R3       ;GET # OF DRIVES
                                MOV     #OUTBUF,R5      ;INITIALIZE POINTER
```



```
                        14$:    TSTB    (R5)            ;DID THIS DRIVE INTERRUPT?
                                BNE     13$             ;YES, BRANCH
                                MOV     (R5),-(R6)      ;GET THIS DRIVE #
                                JSR     PC,SHFTRT       ;SHIFT IT TO THE RIGHT
                                MOV     (R6)+,$REG0     ;THIS DRIVE # DID NOT INTERRUPT
                                                        ;DURING H'WARE POLL
                                ERROR   52              ;DRIVE # (AS IN $REG0) DID NOT
                                                        ;INTERRUPT DURING HARDWARE POLL
                        13$:    ADD     #2,R5           ;INCREMENT POINTER TO THE NEXT FLAG
                                DEC     R3              ;CHKD FOR ALL DRIVES?
                                BNE     14$             ;IF NOT LUP BACK

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

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

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

                        .SBTTL  END OF PASS ROUTINE

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

```
6070
6071  020652                                $EOP:
6072  020652  000004                               SCOPE
6073  020654  005037  001102                       CLR     $TSTNM          ;;ZERO THE TEST NUMBER
6074  020660  005037  001206                       CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
6075  020664  005237  001100                       INC     $PASS           ;;INCREMENT THE PASS NUMBER
6076  020670  042737  100000  001100               BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
6077  020676  005327                               DEC     (PC)+           ;;LOOP?
6078  020700  000001                     $EOPCT: .WORD   1
6079  020702  003022                               BGT     $DOAGN          ;;YES
6080  020704  012737                               MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
6081  020706  000001                     $ENDCT: .WORD   1
6082  020710  020700                             $EOPCT
6083  020712  104401  020757                       TYPE    ,$ENDMG         ;;TYPE "END PASS #"
6084  020716  013746  001100                       MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
6085  020722  104405                               TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
6086  020724  104401  020754                       TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
6087  020730  013700  000042             $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
6088  020734  001405                               BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
6089  020736  000005                               RESET                   ;;CLEAR THE WORLD
6090  020740  004710                     $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
6091  020742  000240                               NOP                     ;;SAVE ROOM
6092  020744  000240                               NOP                     ;;FOR
6093  020746  000240                               NOP                     ;;ACT11
6094  020750                             $DOAGN:
6095  020750  000137                               JMP     @(PC)+          ;;RETURN
6096  020752  004456                     $RTNAD: .WORD   ST4
6097  020754     377     377     000     $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
6098  020757     015  042412  042116     $ENDMG: .ASCIZ  <15><12>/END PASS #/
6099  020764  050040  051501  020123
6100  020772  000043
6101
6102
6103
6104
6105
6106
6107
6108                                               .SBTTL  GT2RG:  ROUTINE FOR GETTING RKCS,RKER
6109
6110                                     ;SUBROUTINE FOR TRANSFERRING THE CONTENTS OF RKCS, RKER
6111                                     ;TO $REG0, $REG1 RESPECTIVELY BEFORE TYPING OUT AN ERROR MESSAGE.
6112                                     ;CALL:  JSR     PC,GT2RG
6113
6114                                             .SBTTL  GT3RG:  ROUTINE FOR GETTING RKCS, RKER, RKDS
6115
6116                                     ;GT3RG
6117                                     ;SUBROUTINE FOR TRANSFERRING THE CONTENTS OF RKCS, RKER, RKDS
6118                                     ;TO  $REG0, $REG1, $REG2 RESPECTIVELY BEFORE TYPING OUT AN
6119                                     ;ERROR MESSAGE.
6120                                     ;CALL:  JSR     PC,GT3RG
6121
6122                                             .SBTTL  GT4RG:  ROUTINE FOR GETTING RKCS, RKER, RKDS, RKDA
6123
6124                                     ;GT4RG
6125                                     ;SUBROUTINE FOR TRANSFERRING CONTENTS OF RKCS, RKER, RKDS
```



```
6126                                     ;RKDA TO $REG0, $REG1, $REG2, $REG3 RESPECTIVELY BEFORE
6127                                     ;TYPING OUT AN ERROR MESSAGE.
6128                                     ;CALL:  JSR     PC,GT4RG
6129
6130  020774  017737  160340  001170     GT4RG:  MOV     @RKDA,$REG3             ;GET RKDA
6131  021002  017737  160320  001166     GT3RG:  MOV     @RKDS,$REG2             ;GET RKDS
6132  021010  017737  160314  001164     GT2RG:  MOV     @RKER,$REG1             ;GET RKER
6133  021016  017737  160310  001162             MOV     @RKCS,$REG0
6134  021024  000207                             RTS     PC
6135
6136
6137
6138
6139
6140                                             .SBTTL  TYERM:  SPECIAL ERROR MESSAGE ROUTINE
6141
6142                                     ;TYERM
6143                                     ;THIS ROUTINE TYPES OUT 'EROR AT PC=X'
6144                                     ;X IS THE PC WHERE THE  EXPLANATION AS TO WHAT HAPPENED IS GIVEN. THIS ROUTINE
6145                                     ;IS USED ONLY FOR NON-MANUAL MODE OF THE PROGRAM.
6146                                     ;CALL:          JSR     TYERM
6147
6148  021026                             TYERM:
6149  021026  104401  021034                   TYPE    ,65$            ;;TYPE ASCIZ STRING
6150  021032  000406                             BR      64$             ;;GET OVER THE ASCIZ
6151                                     ;;65$:  .ASCIZ  <15><12>/EROR,PC=/
6152  021050                             64$:
6153  021050  010346                             MOV     R3,-(SP)
6154  021052  104402                             TYPOC
6155  021054  000207                             RTS     PC
6156
6157
6158
6159
6160
6161
6162                                             .SBTTL  BDA0, BDA4:     BREAK DISK ADDRESS INTO SEC, SUR, CYL, DRIVE
6163
6164                                     ;BDA0, BDA4
6165
6166                                     ;THIS ROUTINE BREAKS A DISK ADDRESS (BITS 0-15) INTO  DRIVE #,
6167                                     ;CYLINDER #, SURFACE, SECTOR #. THE ROUTINE IS CALLED BY USING EITHER
6168                                     ;BRKDA0 OR BRKDA4, BOTH BEING 'TRAP' INSTRUCTIOS WITH THEIR LOWER BYTES
6169                                     ;ENCODED TO PROVIDE INDEXING TO 'BDA0' OR 'BDA4'. BEFORE CALLING
6170                                     ;THE ROUTINE THE DISK ADDRESS WHICH IS TO BE BROKEN AS ABOVE
6171                                     ;IS DEPOSITED IN $REG10.
6172                                     ;'BRKDA0' PUTS THE              BRKDA4 PUTS THE
6173                                     ;DRIVE # INTO $REG0             DRIVE # INTO $REG4
6174                                     ;CYLINDER # INTO $REG1          CYLINDER # INTO $REG5
6175                                     ;SURFACE # INTO $REG2           SURFACE # INTO $REG6
6176                                     ;SECTOR # INTO $REG3            SECTOR # INTO $REG7
6177                                     ;CALL:  BRKDA0                  BRKDA4
6178
6179
6180  021056  010046                     BDA0:   MOV     R0,-(SP)        ;PUSH R0 ONTO THE STACK
6181  021060  012700  001172                     MOV     #$REG3+2,R0     ;SET UP POINTER
```

```
6182  021064  000403                          BR      BDAR
6183
6184  021066  010046                  BDA4:   MOV     R0,-(SP)        ;PUSH R0 ONTO THE STACK
6185  021070  012700  001202                  MOV     #$REG7+2,R0     ;SET UP POINTER
6186
6187  021074  032777  020000  160036  BDAR:   BIT     #20000,@SWR     ;INHIBIT TYPEOUT?
6188  021102  001034                          BNE     2$              ;YES, BRANCH TO EXIT POINT
6189
6190  021104  010146                          MOV     R1,-(SP)        ;PUSH R1 ON STACK
6191  021106  010246                          MOV     R2,-(SP)        ;PUSH R2 ON STACK
6192  021110  013701  001202                  MOV     $REG10,R1       ;GET THE ADDRESS WHICH
6193                                                                  ;HAS TO BE BROKEN
6194  021114  042701  177760                  BIC     #177760,R1      ;EXTRACT SECTOR BITS 0-3
6195  021120  010140                          MOV     R1,-(R0)        ;MOVE SECTOR BITS TO $REG3 OR $REG7
6196  021122  013701  001202                  MOV     $REG10,R1       ;GET THE DSK-ADRES TO BE BROKEN
6197  021126  006201                          ASR     R1              ;SHIFT RIGHT 4 TIMES
6198  021130  006201                          ASR     R1
6199  021132  006201                          ASR     R1
6200  021134  006201                          ASR     R1
6201  021136  010102                          MOV     R1,R2           ;STORE THIS
6202  021140  042702  177776                  BIC     #177776,R2      ;EXTRACCT THE SURFACE BIT
6203  021144  010240                          MOV     R2,-(R0)        ;MOVE SURFACE BIT TO $REG3 OR $REG6
6204  021146  006201                          ASR     R1
6205  021150  010102                          MOV     R1,R2           ;STORE IT
6206  021152  042702  177400                  BIC     #177400,R2      ;EXTRACT THE CYLINDER BITS
6207  021156  010240                          MOV     R2,-(R0)        ;MOVE CYLINDER BITS TO $REG1 OR $REG5
6208  021160  000301                          SWAB    R1              ;SWAB HI-LO BYTES
6209  021162  042701  177770                  BIC     #177770,R1      ;EXTRACT THE DRIVE #
6210  021166  010140                          MOV     R1,-(R0)        ;MOVE DRIVE # TO $REG0 OR $REG4
6211
6212  021170  012602                          MOV     (SP)+,R2        ;RESTORE R2
6213  021172  012601                          MOV     (SP)+,R1        ;RESTORE R1
6214  021174  012600                  2$:     MOV     (SP)+,R0        ;RESTORE R0 FROM THE STACK
6215  021176  000002                          RTI                     ;RETURN FROM INTERRUPT, EXIT THIS
6216                                                                  ;ROUTINE
6217
6218
6219
6220                                          .SBTTL  SHFTRT: SHIFT RIGHT ROUTINE
6221
6222                                  ;SHFTRT
6223                                  ;THIS ROUTINE SHIFTS A WORD TO THE RIGHT 13 TIMES. THE WORD TO BE SHIFTED
6224                                  ;IS PUT ON THE STACK BEFORE ENTERING THIS ROUTINE AND IT IS POPPED UP
6225                                  ;FROM THE STACK AFTER THE SHIFT HAS BEEN DONE.
6226                                  ;CALL:  JSR     PC,SHFTRT
6227
6228  021200  012737  177763  021224  SHFTRT: MOV     #-15,2$         ;SET UP A COUNT OF 13
6229  021206  000241                          CLC                     ;CLEAR THA C BIT
6230  021210  006066  000002          1$:     ROR     2(R6)           ;ROTATE RIGHT THE WORD TO B SHFTD
6231  021214  005237  021224                  INC     2$              ;SHIFTED 13 TIMES?
6232  021220  001373                          BNE     1$              ;IF NOT LUP BAK & SHIFT
6233  021222  000207                          RTS     PC              ;EXIT FROM THIS SUBROUTINE
6234  021224  000000                  2$:     0
6235
6236
6237
```



```
6238
6239
6240                                          .SBTTL  CHKHE:  CHECK FOR 'ERR'OR
6241                                          .SBTTL  CHKHE1: CHECK FOR 'ERR'OR
6242
6243                                  ;;CHKHE
6244                                  ;THIS ROUTINE CHECKS IF 'HE' OR 'ERR' BITS IN RKCS ARE SET. IF ANY OF THE
6245                                  ;TWO BITS ARE SET, THE CONTENTS OF RKCS, ER, DS, AND DA ARE SAVED AND A
6246                                  ;RETURN IS MADE TO THE ERROR MESSAGE FOLLOWING THE 'JSR' CALL.
6247                                  ;AT THE TIME OF ENTRY 'DRIVAD' CONTAINS THE DISK ADDRESS WHICH IS TO
6248                                  ;BE BROKEN DOWN INTO DRIVE #, CYLINDER, SURFACE AND SECTOR #. THIS INFORMATION
6249                                  ;IS SAVED TO BE USED LATER FOR ERROR REPORTING. IF THE BITS ARE NOT SET,
6250                                  ;RETURN IS MADE TO SKIP THE ERROR MESSAGE.
6251
6252                                  ;CHKHE1
6253                                  ;THIS ROUTINE CHECKS IF 'HE' OR 'ERR' BITS IN RKCS ARE SET. IF ANY OF THE
6254                                  ;TWO BITS ARE SET, THE CONTENTS OF RKCS, ER, DS, AND DA ARE SAVED AND A
6255                                  ;RETURN IS MADE TO THE ERROR MESSAGE FOLLOWING THE 'JSR' CALL.
6256                                  ;AT THE TIME OF ENTRY R1 CONTAINS THE DISK ADDRESS WHICH IS TO BE BROKEN
6257                                  ;DOWN INTO DRIVE #, CYLINDER, SURFACE AND SECTOR #. THIS INFORMATION IS
6258                                  ;SAVED TO BE USED LATER FOR ERROR REPORTING. IF THE BITS ARE NOT SET,
6259                                  ;RETURN IS MADE TO SKIP THE ERROR MESSAGE.
6260
6261  021226  010137  001202          CHKHE1: MOV     R1,$REG10       ;SAVE THE DISK ADRES
6262  021232  000403                          BR      CHE1
6263
6264  021234  013737  001350  001202  CHKHE:  MOV     DRIVAD,$REG10   ;SAVE THE DISK ADRES
6265  021242  032777  140000  160062  CHE1:   BIT     #140000,@RKCS   ;IS 'HE' OR 'ERR' BIT SET?
6266  021250  001467                          BEQ     CRETRN          ;NO
6267  021252  004737  020774                  JSR     PC,GT4RG        ;GET RKCS,ER,DS, DA
6268  021256  104416                          BRKDA4                  ;GO TO 'BDA4' & BREAK CONTENTS O
6269                                                                  ;$REG10 INTO DR#, CYL, SUR, SEC BITS
6270  021260  000207                          RTS     PC              ;RETURN TO THE ERROR MESSAGE
6271
6272
6273
6274
6275                                          .SBTTL  CHKDA:  CHECK IF RKDA INCREMENTED CORRECTLY
6276
6277                                  ;CHKDA
6278                                  ;THIS ROUTINE CHECKS IF RKDA INCREMENTED CORRECTLY. IF RKDA INCREMENTED
6279                                  ;CORRECTLY RETURN IS MADE TO SKIP THE ERROR MESSAGE.
6280                                  ;IF RKDA DID NOT INCREMENT CORRECTLY, THE EXPECTED AND RECIEVED VALUES
6281                                  ;OF RKDA ARE SAVED AND A RETURN IS MADE TO THE ERROR MESSAGE FOLLOWING THE
6282  021262  013705  001350          ;'JSR' CALL.
6283  021266  005205                  CHKDA:  MOV     DRIVAD,R5       ;RKDA SHOULD INCREMENT TO THIS
6284  021270  020577  160044                  INC     R5              ;AFTER DATA TRANSFER IS DONE
6285  021274  001455          CHKDA1: CMP     R5,@RKDA        ;DID RKDA INCREMENT CORRECTLY?
6286                                          BEQ     CRETRN          ;IF YES, BRANCH
6287  021276  010537  001202                                          ;IF NOT, REPORT ERROR
6288  021302  104415                          MOV     R5,$REG10       ;GET EXPCTD RKDA
6289                                          BRKDA0                  ;GO TO 'BDA0' & BREAK  CONTENTS OF
6290  021304  017737  160030  001202                                  ;$REG10 INTO DR #,CYL,SUR,SEC BITS
6291  021312  104416                          MOV     @RKDA,$REG10    ;GET ACTUAL RKDA
6292                                          BRKDA4                  ;GO TO 'BDA4' & BREAK  CONTENTS OF
6293  021314  000207                                                  ;$REG10 INTO DR #,CYL,SUR,SEC BITS
                                              RTS     PC              ;RETURN TO THE ERROR MESSAGE
```

```
                                        .SBTTL  CHKWC:  CHECK IF RKWC OVERFLOWED

                                ;CHKWC
                                ;THIS ROUTINE CHECKS IF RKWC OVERFLOWED :O 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.
021316  005777  160012        CHKWC:  TST     @RKWC           ;DID WORD COUNT OVERFLOW TO 0?
021322  001442                        BEQ     CRETRN          ;IF YES, BRANCH
                                                                ;IF NOT, ERROR
021324  017737  160004  001162        MOV     @RKWC,$REG0     ;GET RKWC
021332  017737  160002  001164        MOV     @RKDA,$REG1     ;GET RKDA
021340  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.

021342  005777  157762        CHKER:  TST     @RKER           ;DID ANY BIT IN RKER SET?
021346  001430                        BEQ     CRETRN          ;NO, BRANCH
                                                                ;YES, ERROR
021350  004737  021002                JSR     PC,GT3RG        ;GO, GET RKCS, ER, DS

021354  000207                        RTS     PC              ;RETURN TO THE ERROR MESSAGE


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

021356  005777  157746        CHKECLR: TST    @RKER           ;ANY BIT IN RKER SET?
021362  001422                        BEQ     CRETRN          ;NO
021364  013737  001330  001162        MOV     RKER,$REG0      ;GET ADRES OF RKER
021372  017737  157732  001164        MOV     @RKER,$REG1     ;GET CONTENTS OF RKER
021400  000207                        RTS     PC              ;RETURN TO THE ERROR MESSAGE


                                ;CHKCCLR
                                ;THIS ROUTINE CHECKS THAT RKCS IS CLEAR. IF NOT, THE CONTENTS OF RKCS ARE
                                ;SAVED AND A RETURN IS MADE TO THE ERROR MESSAGE. IF RKCS IS CLEAR THE
                                ;ERROR MESSAGE IS SKIPPED ON RETURN.
021402  022777  000200  157722  CHKCCLR: CMP    #200,@RKCS      ;IS RKCS CLEAR?
021410  001407                        BEQ     CRETRN          ;YES
021412  013737  001332  001162        MOV     RKCS,$REG0      ;SAVE ADRES OF RKCS
021420  017737  157706  001164        MOV     @RKCS,$REG1     ;SAVE THE CONTENT OF RKCS
021426  000207                        RTS     PC              ;RETURN TO THE ERROR MESSAGE

021430  062716  000002        CRETRN: ADD     #2,(SP)         ;SKIP ERROR MESSAGE ON
021434  000207                        RTS     PC              ;RETURN
```



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

                                ;TSTRWS
                                ;THIS ROUTINE WAITS FOR R/W/S RDY TO SET. WHEN IT SETS, THE RETURN PC
                                ;IS INCREMENTED SO THAT ON RETURN (TO THE MAIN PROGRAM) THE ERROR
                                ;MESSAGE FOLLOWING THE 'JSR' CALL IS SKIPPED. IF R/W/S RDY DOES NOT SET
                                ;THEN A RETURN IS MADE TO THE ERROR MESSAGE (FOLLOWING THE 'JSR' CALL).
                                ;WAITING TIME IS APPROX. 1040 MS FOR 11/20, APPROX. 208 MS FOR 11/45
                                ;CALL:  JSR     TSTRWS

021436  013777  001350  157674  TSTRWS: MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE
021444  005037  001366                CLR     TIMER           ;INITIALIZE COUNT
021450  032777  000100  157650  1$:     BIT     #100,@RKDS      ;DID R/W/S RDY SET?
021456  001007                        BNE     2$              ;YES, BRANCH
021460  005237  001366                INC     TIMER           ;WAIT FOR R/W/S RDY
021464  001371                        BNE     1$              ;ERROR IF IT'S NOT SET BY NOW
021466  017737  157634  001162        MOV     @RKDS,$REG0     ;GET RKDS
021474  000207                        RTS     PC              ;EXIT (TO ERROR FOOLOWING 'JSR TSTRWS')

021476  062716  000002        2$:     ADD     #2,(SP)         ;ADJUST RETURN ADRES TO SKIP OVER
                                                                ;ERROR (FOLLOWING 'JSR TSTRWS')
021502  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
                                ;EXITNG 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

021504  005037  001364        DRESET: CLR     COUNT1          ;INITIALIZE THE COUNT
021510  013777  001350  157622        MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE
021516  012777  000015  157606        MOV     #15,@RKCS       ;DRIVE RESET, GO
021524  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
021526  032777  000100  157572  1$:     BIT     #100,@RKDS      ;DID R/W/S RDY SET?
021534  001013                        BNE     2$
021536  012746  177770                MOV     #-10,-(SP)      ;PUSH COUNT ON SP
021542  005216                        INC     (SP)            ;COUNT IT DOWN
021544  001376                        BNE     .-2
021546  005726                        TST     (SP)+           ;POP UP $P
021550  005237  001364                INC     COUNT1          ;IF NOT WAIT
```

```
                                        BNE     1$                      ;WAITED LONG?
                                        JSR     PC,GT4RG
                                        BR      2$+4
                                2$:     ADD     #2,@R6
                                        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.

021572  013777  001350  157540  TSTSIN: MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE
021600  032777  001000  157520          BIT     #1000,@RKDS     ;IS SIN SET?
021606  001403                  BEQ     1$
021610  004737  021504          JSR     PC,DRESET       ;GO DO DRIVE RESET, SIN  SET
021614  000401                  BR      2$              ;REPORT ERROR
021616  000002          1$:     RTI
021620  032777  020000  157312  2$:     BIT     #SW13,@SWR      ;INHIBIT TYPEOUT?
021626  001373                  BNE     1$              ;IF YES, SKIP TYPEOUT
021630  104401  021636          TYPE    ,65$            ;;TYPE ASCIZ STRING
021634  000406                  BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  /ERROR PC= /
021652                          64$:
021652  011646                  MOV     (SP),-(SP)
021654  062716  177776          ADD     #-2,(SP)        ;GET THE PC WHERE 'TST.SIN' IS LOCATED
021660  104402                  TYPOC                   ;GO TYPE OUT PC
021662  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.

021664  017637  000000  001366  DELA.Y: MOV     @(SP),TIMER     ;GET 'AMOUNT' (N) FOR WHICH
021672  062716  000002          ADD     #2,(SP)         ;DELAY IS TO BE PROVIDED
                                                        ;ADJUST STACK POINTER TO SKIP OVER 'N'
021676  005337  001366          1$:     DEC     TIMER           ;COUNT DOWN TO 0
021702  001375                  BNE     1$
```



```
021704  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'.

021706  017637  000000  001366  WATINT: MOV     @(SP),TIMER     ;GET 'AMOUNT' (N) FOR WHICH
021714  062716  000002          ADD     #2,(SP)         ;WAITING IS TO BE DONE
                                                        ;ADJUST STACK POINTER FOR CORRECT RETURN
021720  013746  001400          MOV     RKPRI,-(SP)     ;DROP CPU PRIORITY SO THAT RK11 CAN
021724  012746  021732          MOV     #1$,-(SP)       ; INTERRUPT
021730  000002                  RTI
021732  005337  001366          1$:     DEC     TIMER           ;WAIT FOR RK11 TO INTERRUPT
021736  001375                  BNE     1$
                                                        ;IF INTERRUPT HAS NOT OCCURED BY NOW
021740  000002                  RTI                     ;RETURN AND REPORT ERROR
                                                        ;EXIT


                                ;WATIME

021742  005000          WATIME: CLR     R0
021744  005001                  CLR     R1
021746  005200          1$:     INC     R0
021750  001376                  BNE     1$
021752  105201                  INCB    R1
021754  001374                  BNE     1$
021756  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
                                ;
                                CH.CRDY: CLR TIMER
021760  005037  001366          1$:     TSTB    @RKCS           ;CNTRL RDY SET?
021764  105777  157342          BMI     2$              ;YES
021770  100406                  INC     TIMER
021772  005237  001366          BNE     1$              ;NO, WAIT
021776  001372                  JSR     PC,GT4RG        ;SAVE RKCS, ER, DS, DA
022000  004737  020774          RTI
022004  000002
                                2$:     ADD     #2,(SP)         ;ADJUST RETURN ADDRESS TO
022006  062716  000002          RTI                     ;SKIP ERROR MESSAGE ON RETURN
022012  000002
                                        .SBTTL  CON.RESET:      CONTROL REST ROUTINE
                                ;CON.RESET
                                ;THIS ROUTINE ISSUES A CONTROL RESET AND WAITS FOR
                                ;THE 'CNTRL RDY' FLAG TO SET.  WHEN THE FLAG SETS
                                ;AN EXIT IS MADE OUT OF THE ROUTINE.  IF 'CNTRL-RDY'
                                ;DOES NOT SET WITHIN A CERTAIN TIME AN ERROR MESSAGE
                                ;       CNT RDY DIDN'T SET
                                ;       PC=XXXXXX RKCS=YYYYYY
                                ;IS GIVEN. NOTE THAT XXXXXX IS THE PC WHERE 'CNT.RESET' OR 'CNT.RDY'
                                ;IS CALLED.
                                ;CALL:  CNT.RESET
                                        .SBTTL  CNT.RDY:        WAIT FOR CONTROL READY ROUTINE
                                ;CN.RDY
                                ;THIS ROUTINE WAITS FOR THE CONTROL READY BIT TO SET AND WHEN IT
                                ;SETS EXITS OUT. IF WITHIN A CERTAIN TIME CNTRL RDY DOES
                                ;NOT SET AN ERROR IS REPORTED. WAITING TIME IS 883 MS FOR 11/20
                                ;175 MS FOR 11/45 WITH BIPOLAR MEMORY.
                                ;CALL:  CNT.RDY
022014  012777  000001  157310  CN.RST: MOV     #1,@RKCS        ;ISSUE A CONTROL RESET
022022  012737  177500  001170          MOV     #-300,$REG3     ;SET UP COUNT
022030  000402                  BR      CN.RDY+4                ;SKIP OVER CN.RDY
022032  005037  001170          CN.RDY: CLR     $REG3
022036  105777  157270          1$:     TSTB    @RKCS           ;DID CNTRL-RDY SET?
022042  100435                  BMI     3$              ;YES, EXIT
022044  005237  001170          INC     $REG3           ;WAITED LONG?
022050  001372                  BNE     1$              ;IF NOT, GO BAK & WAIT
022052  032777  020000  157060  2$:     BIT     #SW13,@SWR      ;INHIBIT TYPEOUT?
022060  001026                  BNE     3$              ;IF YES, SKIP TYPEOUT
022062  104401                  TYPE
022064  001245                  MSG3
022066  104401  022074          TYPE    ,65$            ;;TYPE ASCIZ STRING
022072  000403                  BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/PC=/
```



```
022102                          64$:
022102  011646                  MOV     (SP),-(SP)
022104  162716  000002          SUB     #2,(SP)
022110  104402                  TYPOC                   ;GO TYPE PC IN THE MAIN PROGRAM,
                                                        ; WHERE ERROR OCCURRED
022112  104401  022120          TYPE    ,67$            ;;TYPE ASCIZ STRING
022116  000404                  BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  / RKCS=/
022130                          66$:
022130  017746  157176          MOV     @RKCS,-(SP)     ;GET RKCS
022134  104402                  TYPOC                   ;GO TYPE IT
022136  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
                                $SCOPE:
022140                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
022140  104407
022142  032777  040000  156770  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
022150  001111                  BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
022152  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
022154  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
022160  012737  022200  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
022166  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
022172  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
022176  000463                  BR      $SVLAD          ;;GO TO THE NEXT TEST
022200  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
022202  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
022206  000423                  BR      7$              ;;LOOP ON THE PRESENT TEST
022210                          6$:;#####END OF CODE FOR THE XOR TESTER#####
022210  032777  000400  156727          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
022216  001404                  BEQ     2$              ;;BR IF NO
022220  127737  156714  001102          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
022226  001462                  BEQ     $OVER           ;;BR IF YES
022230  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
```

```
6630  022234  001421                    BEQ     3$              ;;BR IF NO
6631  022236  123737  001115  001103    CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
6632  022244  101015                    BHI     3$              ;;BR IF NO
6633  022246  032777  001000  156664    BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
6634  022254  001404                    BEQ     4$              ;;BR IF NO
6635  022256  013737  001110  001106  7$:  MOV   $LPERR,$LPADR  ;;SET LOOP ADDRESS TO LAST SCOPE
6636  022264  000443                    BR      $OVER
6637  022266  105037  001103          4$:  CLRB  $ERFLG          ;;ZERO THE ERROR FLAG
6638  022272  005037  001206              CLR   $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
6639  022276  000415                      BR    1$              ;;ESCAPE TO THE NEXT TEST
6640  022300  032777  004000  156632  3$:  BIT   #BIT11,@SWR    ;;INHIBIT ITERATIONS?
6641  022306  001011                      BNE   1$              ;;BR IF YES
6642  022310  005737  001100              TST   $PASS           ;;IF FIRST PASS OF PROGRAM
6643  022314  001406                      BEQ   1$              ;;      INHIBIT ITERATIONS
6644  022316  005237  001104              INC   $ICNT           ;;INCREMENT ITERATION COUNT
6645  022322  023737  001206  001104      CMP   $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
6646  022330  002021                      BGE   $OVER           ;;BR IF MORE ITERATION REQUIRED
6647  022332  012737  000001  001104  1$:  MOV   #1,$ICNT       ;;REINITIALIZE THE ITERATION COUNTER
6648  022340  013737  022410  001206      MOV   $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
6649  022346  105237  001102  $SVLAD: INCB  $TSTNM            ;;COUNT TEST NUMBERS
6650  022352  011637  001106              MOV   (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
6651  022356  011637  001110              MOV   (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
6652  022362  005037  001210              CLR   $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
6653  022366  112737  000001  001115      MOVB  #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
6654  022374  013777  001102  156540  $OVER:  MOV  $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
6655  022402  013716  001106              MOV   $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
6656  022406  000002                      RTI                   ;;FIXES PS
6657  022410  000050              $MXCNT: 50                    ;;MAX. NUMBER OF ITERATIONS
6658
6659
6660                                ;;**********************************************************************
6661
6662                                .SBTTL  ERROR HANDLER ROUTINE
6663
6664                                ;*SW15=1        HALT ON ERROR
6665                                ;*SW13=1        INHIBIT ERROR TYPEOUTS
6666                                ;*SW10=1        TESTING ON SIMULATOR
6667                                ;*SW09=1        LOOP ON ERROR
6668                                ;*SW12=1        CYCLE ON ERROR TO PREVIOUS 'SCOPE'
6669                                ;*SW06=1        DROP DRIVE AFTER MAXIMUM (ALLOWABLE) ERRORS ON THE DRIVE
6670                                ;*GO TO $ERRTYP ON ERROR
6671
6672  022412  104407              $ERROR: CKSWR                   ;CHECK FOR SOFTWARE SWITCH REGISTER REQUEST
6673  022414  105237  001103      7$:     INCB    $ERFLG          ;SET THE ERROR FLAG
6674  022420  001775                      BEQ     7$              ;DON'T LET THE FLAG GO TO ZERO
6675  022422  013777  001102  156512      MOV     $TSTNM,@DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
6676  022430  005237  001112      1$:     INC     $ERTTL          ;COUNT THE NUMBER OF ERRORS
6677
6678  022434  032777  000100  156476      BIT     #BIT6,@SWR      ;DESELECT DRIVE SW SET?
6679  022442  001404                      BEQ     6$              ;NO
6680  022444  023727  001112  000005      CMP     $ERTTL,#5       ;MORE THAN 5 ERRORS ON THIS DRIVE?
6681  022452  101053                      BHI     8$              ;YES, DESELCT THE DRIVE
6682
6683  022454  011637  001116      6$:     MOV     (SP),$ERRPC     ;GET ADDRESS OF ERROR INSTRUCTION
6684  022460  162737  000002  001116      SUB     #2,$ERRPC
6685  022466  117737  156424  001114      MOVB    @$ERRPC,$ITEMB  ;STRIP AND SAVE THE ERROR ITEM CODE
```



```
6686  022474  032777  020000  156436      BIT     #SW13,@SWR      ;SKIP TYPEOUT IF SET
6687  022502  001004                      BNE     2$              ;SKIP TYPEOUTS
6688  022504  004737  022734              JSR     PC,@#$ERRTYP    ;GO TO USER ERROR ROUTINE
6689  022510  104401  001213              TYPE    ,$CRLF
6690  022514  023737  000042  000046  2$:  CMP    @#42,@#46       ;ARE WE IN ACT11 AUTO MODE?
6691  022522  001403                      BEQ     .+10            ;YES, HALT ON ERROR
6692  022524  005777  156410              TST     @SWR            ;HALT ON ERROR?
6693  022530  100002                      BPL     3$              ;SKIP IF CONTINUE
6694  022532  000000                      HALT                    ;HALT ON ERROR!
6695  022534  104407                      CKSWR                   ;CHECK FOR SOFTWARE SWITCH REGIATER REQUEST
6696  022536  032777  010000  156374  3$:  BIT     #SW12,@SWR      ;SW 12 SET?
6697  022544  001402                      BEQ     .+6             ;NO, BRANCH
6698  022546  013716  001106              MOV     $LPADR,(SP)     ;ADJUST RETURN ADRES FOR SW12
6699  022552  032777  001000  156360      BIT     #SW09,@SWR      ;LOOP ON ERROR SWITCH SET?
6700  022560  001402                      BEQ     4$              ;BR IF NO
6701  022562  013716  001110              MOV     $LPERR,(SP)     ;FUDGE RETURN FOR LOOPING
6702  022566  005737  001210      4$:     TST     $ESCAPE         ;CHECK FOR AN ESCAPE ADDRESS
6703  022572  001402                      BEQ     5$              ;BR IF NONE
6704  022574  013716  001210              MOV     $ESCAPE,(SP)    ;FUDGE RETURN ADDRESS FOR ESCAPE
6705  022600  000002              5$:     RTI                     ;RETURN
6706
6707  022602  005737  001434      8$:     TST     T56FLG          ;IF EROR WAS IN LAST TEST (POLL)
6708                                                            ;DROP ALL THE DRIVES
6709  022606  001407                      BEQ     10$
6710  022610  104401  001303              TYPE    ,MSG5
6711  022614  005037  001412              CLR     DRIVS
6712  022620  022626                      CMP     (SP)+,(SP)+
6713  022622  000137  020652              JMP     $EOP
6714  022626  013746  001354      10$:    MOV     DRVPTR,-(SP)    ;DROP THE DRIVE FROM THE
6715  022632  162716  000002              SUB     #2,(SP)         ;SELECTION LIST
6716  022636  013746  001350              MOV     DRIVAD,-(SP)    ;DRIVE ADDR TO STACK
6717  022642  004737  021200              JSR     PC,SHFTRT       ;RIGHT JUSTIFY
6718  022646  042716  000001              BIC     #1,(R6)         ;MAKE EVEN
6719  022652  062716  001414              ADD     #DRIV0,(SP)     ;POINTS TO TABLE FOR EVEN DRIVE
6720  022656  042776  100000  000000      BIC     #BIT15,@(R6)    ;TEST REMAINING DRIVE AS RK05E
6721  022664  062716  000002              ADD     #2,(R6)         ;POINT TO ODD
6722  022670  042736  100000              BIC     #BIT15,@(SP)+   ;TEST AS RK-05E
6723  022674  012736  010000              MOV     #BIT12,@(SP)+   ;INDICATE THIS DRIVE DROPPED
6724  022700  104401  001272              TYPE    ,MSG4
6725  022704  013746  001350              MOV     DRIVAD,-(R6)    ;PUSH DRIVE # ON STACK
6726  022710  004737  021200              JSR     PC,SHFTRT       ;SHIFT IT BEFORE TYPING
6727  022714  104402                      TYPOC                   ;TYPE OUT DRIVE #
6728  022716  104401  001315              TYPE    ,MSG6
6729  022722  005337  001412              DEC     DRIVS           ;DECREMENT # OF DRIVES PRESNT
6730  022726  022626              9$:     CMP     (SP)+,(SP)+     ;RESTORE STACK
6731  022730  000137  020026              JMP     BTEOP           ;GO BACK TO THE END OF PROGRM
6732                                                            ;LINKAGE.
6733
6734                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
6735
6736                                ;;**********************************************************
6737                                ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
6738                                ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
6739                                ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
6740
6741  022734                        $ERRTYP:
```

```
                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
                        MOV     R0,-(SP)        ;;SAVE R0
                        CLR     R0              ;;PICKUP THE ITEM INDEX
                        BISB    @#$ITEMB,R0
                        BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                ;;TYPE THE PC OF THE ERROR
                        MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                ;;ERROR ADDRESS
                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                        BR      6$              ;;GET OUT
1$:                     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
                        ASL     R0              ;;      WORK FOR THE ERROR TABLE
                        ASL     R0
                        ASL     R0
                        ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
                        MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
                        BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
                        TYPE                    ;;TYPE THE "ERROR MESSAGE"
2$:                     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3$:                     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
                        BEQ     5$              ;;SKIP TYPEOUT IF 0
                        TYPE                    ;;TYPE THE "DATA HEADER"
4$:                     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
5$:                     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
                        BNE     7$              ;;GO TYPE THE DATA
6$:                     MOV     (SP)+,R0        ;;RESTORE R0
                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
                        RTS     PC              ;;RETURN
7$:
                        MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                        TST     (R0)            ;;IS THERE ANOTHER NUMBER?
                        BEQ     6$              ;;BR IF NO
                        TYPE    ,8$             ;;TYPE TWO(2) SPACES
                        BR      7$              ;;LOOP
8$:                     .ASCIZ  /  /            ;;TWO(2) SPACES
                        .EVEN

                .SBTTL  TYPE ROUTINE

                ;;*****************************************************************
                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                ;*
                ;*CALL:
                ;*1) USING A TRAP INSTRUCTION
                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                ;*OR
                ;*      TYPE
                ;*      MESADR
                ;*
```



```
$TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
        BPL     1$              ;;BR IF YES
        HALT                    ;;HALT HERE IF NO TERMINAL
        BR      3$              ;;LEAVE
1$:     MOV     R0,-(SP)        ;;SAVE R0
        MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
60$:    MOV     (SP)+,R0        ;;RESTORE R0
3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
        RTI                     ;;RETURN
4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
        BEQ     8$
        CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
        BNE     5$
        TST     (SP)+           ;;POP  <CR><LF> EQUIV
        TYPE                    ;;TYPE A CR AND LF
        $CRLF
        CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
        BR      2$              ;;GET NEXT CHARACTER
5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
        BNE     2$              ;;IF NO GO GET NEXT CHAR.
        MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                ;;AND THE NULL CHAR.
7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
        JSR     PC,$TYPEC       ;;GO TYPE A NULL
        DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
        BR      7$              ;;LOOP

        ;HORIZONTAL TAB PROCESSOR

8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
        BNE     9$              ;;TAB STOP
        TST     (SP)+           ;;POP SPACE OFF STACK
        BR      2$              ;;GET NEXT CHARACTER
$TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
        BPL     $TYPEC
        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
        BNE     1$              ;;BRANCH IF NO
        CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
        BR      $TYPEX          ;;EXIT
1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
        BEQ     $TYPEX          ;;BRANCH IF YES
        INCB    (PC)+           ;;COUNT THE CHARACTER
$CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
$TYPEX: RTS     PC


                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
```

```
                                ;;****************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                ;*REPLACED WITH SPACES.
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                ;*      TYPDS                   ;;GO TO THE ROUTINE
                                
023310                          $TYPDS:
023310  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
023312  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
023314  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
023316  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
023320  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
023322  012746  020200                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
023326  016605  000020                 MOV     20(SP),R5       ;;GET THE INPUT NUMBER
023332  100004                         BPL     1$              ;;BR IF INPUT IS POS.
023334  005405                         NEG     R5              ;;MAKE THE BINARY NUMBER POS.
023336  112766  000055  000001         MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
023344  005000                 1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
023346  012703  023524                 MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
023352  112723  000040                 MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
023356  005002                 2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
023360  016001  023514                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
023364  160105                 3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
023365  002402                         BLT     4$              ;;BR IF DONE
023370  005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
023372  000774                         BR      3$
023374  060105                 4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
023376  005702                         TST     R2              ;;CHECK IF BCD DIGIT=0
023400  001002                         BNE     5$              ;;FALL THROUGH IF 0
023402  105716                         TSTB    (SP)            ;;STILL DOING LEADING 0'S?
023404  100407                         BMI     7$              ;;BR IF YES
023406  106316                 5$:     ASLB    (SP)            ;;MSD?
023410  103003                         BCC     6$              ;;BR IF NO
023412  116663  000001  177777         MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
023420  052702  000060         6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
023424  052702  000040         7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
023430  110223                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
023432  005720                         TST     (R0)+           ;;JUST INCREMENTING
023434  020027  000010                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
023440  002746                         BLT     2$              ;;GO DO THE NEXT DIGIT
023442  003002                         BGT     8$              ;;GO TO EXIT
023444  010502                         MOV     R5,R2           ;;GET THE LSD
023446  000764                         BR      6$              ;;GO CHANGE TO ASCII
023450  105726                 8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
023452  100003                         BPL     9$              ;;BR IF NO
023454  116663  177777  177776         MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
023462  105013                 9$:     CLRB    (R3)            ;;SET THE TERMINATOR
023464  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
023466  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
023470  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
023472  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
```



```
023474  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
023476  104401  023524                 TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
023502  016666  000002  000004         MOV     2(SP),4(SP)     ;;ADJUST THE STACK
023510  012616                         MOV     (SP)+,(SP)
023512  000002                         RTI                     ;;RETURN TO USER
023514  023420                 $DTBL:  10000.
023516  001750                         1000.
023520  000144                         100.
023522  000012                         10.
023524  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

023534  017646  000000         $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
023540  116637  000001  023757         MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
023546  112637  023761                 MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
023552  062716  000002                 ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
023556  000406                         BR      $TYPON
023560  112737  000001  023757 $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
023566  112737  000006  023761         MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
023574  112737  000005  023756 $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
023602  010346                         MOV     R3,-(SP)        ;;SAVE R3
023604  010446                         MOV     R4,-(SP)        ;;SAVE R4
023606  010546                         MOV     R5,-(SP)        ;;SAVE R5
023610  113704  023761                 MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
023614  005404                         NEG     R4
023616  062704  000006                 ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
023622  110437  023760                 MOVB    R4,$OMODE       ;;SAVE IT FOR USE
023626  113704  023757                 MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
023632  016605  000012                 MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
023636  005003                         CLR     R3              ;;CLEAR THE OUTPUT WORD
023640  006105                 1$:     ROL     R5              ;;ROTATE MSB INTO "C"
023642  000404                         BR      3$              ;;GO DO MSB
```

```
6966  023644  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
6967  023646  006105                          ROL     R5
6968  023650  006105                          ROL     R5
6969  023652  010503                          MOV     R5,R3
6970  023654  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
6971  023656  105337  023760                  DECB    $OMODE          ;;TYPE THIS DIGIT?
6972  023662  100016                          BPL     7$              ;;BR IF NO
6973  023664  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
6974  023670  001002                          BNE     4$              ;;TEST FOR 0
6975  023672  005704                          TST     R4              ;;SUPPRESS THIS 0?
6976  023674  001403                          BEQ     5$              ;;BR IF YES
6977  023676  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
6978  023700  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
6979  023704  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
6980  023710  110337  023754                  MOVB    R3,8$           ;;SAVE FOR TYPING
6981  023714  104401  023754                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
6982  023720  105337  023756          7$:     DECB    $OCNT           ;;COUNT BY 1
6983  023724  003347                          BGT     2$              ;;BR IF MORE TO DO
6984  023726  002402                          BLT     6$              ;;BR IF DONE
6985  023730  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
6986  023732  000744                          BR      2$              ;;GO DO THE LAST DIGIT
6987  023734  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
6988  023736  012604                          MOV     (SP)+,R4        ;;RESTORE R4
6989  023740  012603                          MOV     (SP)+,R3        ;;RESTORE R3
6990  023742  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
6991  023750  012616                          MOV     (SP)+,(SP)
6992  023752  000002                          RTI                     ;;RETURN
6993  023754     000          8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
6994  023755     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
6995  023756     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
6996  023757     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
6997  023760  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
6998
6999                                  .SBTTL  TTY INPUT ROUTINE
7000
7001                                  ;;**************************************************************
7002                                  .ENABL  LSB
7003
7004                                  ;;**************************************************************
7005                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
7006                                  ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
7007                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
7008                                  ;*WHEN OPERATING IN TTY FLAG MODE.
7009  023762  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
7010  023770  001074                          BNE     15$             ;;BRANCH IF NO
7011  023772  105777  155146                  TSTB    @$TKS           ;;CHAR THERE?
7012  023776  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
7013  024000  117746  155142                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
7014  024004  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
7015  024010  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
7016  024014  001062                          BNE     15$             ;;NO, RETURN TO USER
7017  024016  123727  001134  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
7018  024024  001456                          BEQ     15$             ;;BRANCH IF YES
7019
7020  024026  104401  024647                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
7021  024032  104401  024654          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
```



```
7022  024036  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
7023  024042  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
7024  024044  104401  024665                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
7025  024050  005046          19$:    CLR     -(SP)           ;;CLEAR COUNTER
7026  024052  005046                  CLR     -(SP)           ;;THE NEW SWR
7027  024054  105777  155064          7$:     TSTB    @$TKS           ;;CHAR THERE?
7028  024060  100375                          BPL     7$              ;;IF NOT TRY AGAIN
7029
7030  024062  117746  155060                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
7031  024066  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
7032
7033
7034
7035  024072  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
7036  024076  001005                          BNE     10$             ;;BRANCH IF NOT
7037  024100  104401  024642                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
7038  024104  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
7039  024110  000757                          BR      19$             ;;LET'S TRY IT AGAIN
7040
7041
7042  024112  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
7043  024116  001022                          BNE     16$             ;;BRANCH IF NO
7044  024120  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
7045  024124  001403                          BEQ     11$             ;;BRANCH IF YES
7046  024126  016677  000002  155004          MOV     2(SP),@SWR      ;;SAVE NEW SWR
7047  024134  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
7048  024140  104401  001213          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
7049  024144  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
7050  024152  001003                          BNE     15$             ;;BRANCH IF NOT
7051  024154  012777  000100  154762          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
7052  024162  000002          15$:    RTI                     ;;RETURN
7053  024164  004737  023240          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
7054  024170  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
7055  024174  002420                          BLT     18$             ;;BRANCH IF YES
7056  024176  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
7057  024202  003015                          BGT     18$             ;;BRANCH IF YES
7058  024204  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
7059  024210  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
7060  024214  001403                          BEQ     17$             ;;BRANCH IF YES
7061  024216  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
7062  024220  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
7063  024222  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
7064  024224  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
7065  024230  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
7066  024234  000707                          BR      7$              ;;GET THE NEXT ONE
7067  024236  104401  001212          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
7068  024242  000720                          BR      20$             ;;SIMULATE CONTROL-U
7069                                  .DSABL  LSB
7070
7071
7072                                  ;;**************************************************************
7073                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
7074                                  ;*CALL:
7075                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
7076                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
7077                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
```

```
                                        ;
                                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
                                        MOV     4(SP),2(SP)     ;;SAVE THE PS
                                1$:     TSTB    @$TKS           ;;WAIT FOR
                                        BPL     1$              ;;A CHARACTER
                                        MOVB    @$TKB,4(SP)     ;;READ THE TTY
                                        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
                                        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
                                        BNE     3$              ;;BRANCH IF NO
                                2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
                                        BPL     2$              ;;LOOP UNTIL ITS THERE
                                        MOVB    @$TKB,-(SP)     ;;GET CHARACTER
                                        BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
                                        CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
                                        BNE     2$              ;;IF NOT DISCARD IT
                                        BR      1$              ;;YES, RESUME
                                3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
                                        BLT     4$              ;;BRANCH IF YES
                                        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
                                        BGT     4$              ;;BRANCH IF YES
                                        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
                                4$:     RTI                     ;;GO BACK TO USER
                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                ;*CALL:
                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
                                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
                                        CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
                                1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
                                2$:     CMP     #$TTYIN+22,R3   ;;BUFFER FULL?
                                        BLOS    4$              ;;BR IF YES
                                        RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
                                        MOVB    (SP)+,(R3)      ;;GET CHARACTER
                                10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
                                        BNE     5$              ;;BR IF NO
                                        TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
                                        BNE     6$              ;;BR IF NO
                                        MOVB    #'\,9$          ;;TYPE A BACK SLASH
                                        TYPE    ,9$
                                        MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
                                6$:     DEC     R3              ;;BACKUP BY ONE
                                        CMP     R3,#$TTYIN      ;;STACK EMPTY?
                                        BLO     4$              ;;BR IF YES
                                        MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
                                        TYPE    ,9$             ;;GO TYPE
                                        BR      2$              ;;GO READ ANOTHER CHAR.
                                5$:     TST     (SP)            ;;RUBOUT KEY SET?
                                        BEQ     7$              ;;BR IF NO
                                        MOVB    #'\,9$          ;;TYPE A BACK SLASH
                                        TYPE    ,9$
                                        CLR     (SP)            ;;CLEAR THE RUBOUT KEY
                                7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
```



```
                                        BNE     8$              ;;BR IF NO
                                        TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
                                        BR      1$              ;;GO START OVER
                                9$:     CMPB    #22,(R3)        ;;IS CHARACTER A "^R"?
                                        BNE     3$              ;;BRANCH IF NO
                                        CLRB    (R3)            ;;CLEAR THE CHARACTER
                                        TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
                                        TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
                                        BR      2$              ;;GO PICKUP ANOTHER CHACTER
                                4$:     TYPE    ,$QUES          ;;TYPE A '?'
                                        BR      1$              ;;CLEAR THE BUFFER AND LOOP
                                3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
                                        TYPE    ,9$
                                        CMPB    #15,(R3)+       ;;CHECK FOR RETURN
                                        BNE     2$              ;;LOOP IF NOT RETURN
                                        CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
                                        TYPE    ,$LF            ;;TYPE A LINE FEED
                                        TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
                                        MOV     (SP)+,R3        ;;RESTORE R3
                                        MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
                                        MOV     4(SP),2(SP)     ;;       FIRST ASCII CHARACTER ON IT
                                        MOV     #$TTYIN,4(SP)
                                        RTI                     ;;RETURN
                                9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
                                        .BYTE   0               ;;TERMINATOR
                                $TTYIN: .BLKB   22              ;;RESERVE 22 BYTES FOR TTY INPUT
                                $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
                                $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
                                $MSWR:  .ASCIZ  <15><12>/SWR = /
                                $MNEW:  .ASCIZ  /  NEW = /
                                                                ;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.

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


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

                                $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
                                        MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
                                        RTI                     ;;RESTORE THE PSW
```

```
                                        .SBTTL  TRAP TABLE

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

                                        ;       ROUTINE
                                        ;       -------
024732  024720                  $TRPAD: .WORD   $TRAP2
                                        $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
024734  023070                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
024736  023560                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
024740  023534                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER ,AS PER LAST CALL)
024742  023574                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
024744  023310
                                        SGTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
024746  024032
                                        $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
024750  023762                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
024752  024244                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
024754  024364
                                        CH.CRDY ;;CALL=CHKCRDY  TRAP+12(104412) CHECK CONTROL READY
024756  021760
                                        CN.RST  ;;CALL=CNT.RESET        TRAP+13(104413) CONTROL RESET ROUTINE
024760  022014
                                        CN.RDY  ;;CALL=CNT.RDY  TRAP+14(104414) WAIT FOR CNTRL RDY TO SET
024762  022032
                                        BDA0    ;;CALL=BRKDA0   TRAP+15(104415) BREAK RKDA INTO DR #,CYL,SUR,SEC BITS
024764  021056
                                        BDA4    ;;CALL=BRKDA4   TRAP+16(104416) BREAK RKDA INTO DR #,CYL,SUP,SEC BITS
024766  021066
                                        DELA.Y  ;;CALL=DELAY    TRAP+17(104417) TIME DELAY ROUTINE
024770  021664
                                        WATINT  ;;CALL=WAT.INT  TRAP+20(104420) WAIT FOR RK11 INTERRUPT ROUTINE
024772  021706
                                        TSTSIN  ;;CALL=TST.SIN  TRAP+21(104421) TEST SIN ROUTINE
024774  021572


                                        .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;;**************************************************************
                                        ;POWER DOWN ROUTINE
024776  012737  025142  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
025004  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
025012  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
025014  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
025016  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
025020  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
025022  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
025024  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
025026  017746  154106                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
025032  010637  025146                  MOV     SP,$SAVR6       ;;SAVE SP
025036  012737  025050  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
025044  000000                          HALT
025046  000776                          BR      .-2             ;;HANG UP

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



```
                                        ;POWER UP ROUTINE
025050  012737  025142  000024  $PWRUP: MOV     #$ILLUP @#PWRVEC ;;SET FOR FAST DOWN
025056  013706  025146                  MOV     $SAVR6,SP       ;;GET SP
025062  005037  025146                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
025066  005237  025146          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
025072  001375                          BNE     1$              ;;OF  WORD
025074  012677  154040                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
025100  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
025102  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
025104  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
025106  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
025110  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
025112  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
025114  012737  024776  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
025122  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
025130  104401                          TYPE                    ;REPORT THE POWER FAILURE
025132  025150          $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
025134  012716                  $PWRAD: MOV     (PC)+,(SP)      ;;RESTART AT PFSTRT
025136  004702                          .WORD   PFSTRT          ;;RESTART ADDRESS
025140  000002                          RTI
025142  000000          $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
025144  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
025146  000000          $SAVR6: 0                       ;;PUT THE SP HERE
025150  005015  047520  042527  $POWER: .ASCIZ  <15><12>*POWER*
025156  000122
                                        .EVEN

025160  004737  021504          FCHECK: JSR     PC,DRESET               ;RESETB DRIVE
025164  104026                          ERROR   26
025166  104413                          CNT.RESET
025170  013737  001350  025302          MOV     DRIVAD,DRHOLD           ;SAVE DRIVE ADRR
025176  032737  020000  001350          BIT     #20000,DRIVAD           ;SEE IF ODD
025204  001404                          BEQ     1$
025206  042737  020000  001350          BIC     #20000,DRIVAD           ;MAKE EVEN
025214  000403                          BR      2$
025216  052737  020000  001350  1$:     BIS     #20000,DRIVAD           ;MAKE ODD
025224  013777  001350  154106  2$:     MOV     DRIVAD,@RKDA    ;DRIVE ADDR
025232  012777  000011  154072          MOV     #11,@RKCS               ;DRIVE SEEK
025240  104414                          CNT.RDY
025242  013777  025302  154070          MOV     DRHOLD,@RKDA            ;OTHER DRIVE
025250  104414                          CNT.RDY
025252  032777  000100  154046          BIT     #100,@RKDS              ;HEADDS IN MOTION%?
025260  001001                          BNE     3$                      ;NO SO RK-05J
025262  005725                          TST     (R5)+                   ;YES RK-05F
025264  013737  025302  001350  3$:     MOV     DRHOLD,DRIVAD           ;RESTORE ADDR
025272  004737  021504                  JSR     PC,DRESET               ;WAIT FOR RESET
025276  104026                          ERROR   26
025300  000205                          RTS     R5
025302  000000          DRHOLD: 0
025304  005037  001350          SIZEF:  CLR     DRIVAD          ;START AT DR0
025310  012700  001414                  MOV     #DRIV0,R0               ;TABLE OF AVAIL DRIVES
025314  005710                  4$:     TST     (R0)                    ;THIS DRIVE HERE?
025316  001413                          BEQ     2$                      ;NO
025320  005760  000002                  TST     2(R0)                   ;COMPLEMENT HERE?
025324  001410                          BEQ     2$                      ;NO
025326  004537  025160                  JSR     R5,FCHECK               ;SEE IF F MODEL
```

```
7302  025332  000405                          BR      2$                      ;J MODEL
7303  025334  052710  100000                  BIS     #100000,(R0)            ;SET SIGN FOR F
7304  025340  052760  100000  000002          BIS     #100000,2(R0)           ;BOTH DRIVES
7305  025346  005720                  2$:     TST     (R0)+
7306  025350  005720                          TST     (R0)+                   ;NEXT PAIR OF DRIVES
7307  025352  062737  040000  001350          ADD     #40000,DRIVAD           ;NEXT ACTUL ADDR
7308  025360  022700  001433                  CMP     #DRIV7+1,R0             ;CHECKED ALL?
7309  025364  003353                          BGT     4$                      ;NOT YET
7310  025366  000207                          RTS     PC
7311
7312                                          ;ERROR MESSAGES
7313
7314                                          .SBTTL  ERROR MESSAGES
7315
7316  025370  045522  041527  042440  EM11:   .ASCIZ  /RKWC EROR/
7317  025376  047522  000122
7318
7319
7320  025402  044523  020116  051511  EM12:   .ASCIZ  /SIN IS SET/
7321  025410  051440  052105     000
7322
7323  025415     122  041113  020101  EM13:   .ASCIZ  /RKBA EROR/
7324  025422  051105  051117     000
7325
7326  025427     122  042113  020101  EM16:   .ASCIZ  /RKDA WRONG AFTER 'SSE'/
7327  025434  051127  047117  020107
7328  025442  043101  042524  020122
7329  025450  051447  042523  000047
7330
7331  025456  045522  051504  042440  EM21:   .ASCIZ  /RKDS EROR/
7332  025464  047522  000122
7333
7334  025470  050104  020114  042523  EM30:   .ASCIZ  /DPL SET/
7335  025476  000124
7336
7337  025500  051104  020125  042523  EM31:   .ASCIZ  /DRU SET/
7338  025506  000124
7339
7340  025510  045522  032460  041040  EM32:   .ASCIZ  /RK05 BIT NOT SET/
7341  025516  052111  047040  052117
7342  025524  051440  052105     000
7343
7344  025531     104  054522  041040  EM33:   .ASCIZ  /DRY BIT NOT SET/
7345  025536  052111  047040  052117
7346  025544  051440  052105     000
7347
7348  025551     123  045517  042040  EM34:   .ASCIZ  /SOK DIDN'T SET/
7349  025556  042111  023516  020124
7350  025564  042523  000124
7351
7352  025570  042523  026503  047103  EM35:   .ASCIZ  /SEC-CNTR DIDN'T COUNT TO 0/
7353  025576  051124  042040  042111
7354  025604  023516  020124  047503
7355  025612  047125  020124  047524
7356  025620  030040     000
7357
```



```
7358  025623     123  041505  041455  EM36:   .ASCIZ  /SEC-CNTR DIDN'T INCRMNT/
7359  025630  052116  020122  044504
7360  025636  047104  052047  044440
7361  025644  041516  046522  052116
7362  025652     000
7363
7364  025653     123  041505  041455  EM37:   .ASCIZ  /SEC-COUNTR INCRMENTED WRONG/
7365  025660  052517  052116  020122
7366  025666  047111  051103  042515
7367  025674  052116  042105  053440
7368  025702  047522  043516     000
7369
7370  025707     104  042111  023516  EM40:   .ASCIZ  /DIDN'T GET SC=SA FOR THIS SECTR/
7371  025714  020124  042507  020124
7372  025722  041523  051475  020101
7373  025730  047506  020122  044124
7374  025736  051511  051440  041505
7375  025744  051124     000
7376
7377  025747     105  047522  026522  EM41:   .ASCIZ  "EROR-R/W/S RDY SHOULD BE SET"
7378  025754  027522  027527  020123
7379  025762  042122  020131  044123
7380  025770  052517  042114  041040
7381  025776  020105  042523  000124
7382
7383  026004  047125  054105  042520  EM43:   .ASCIZ  /UNEXPECTED RK11 INTERRUPT/
7384  026012  052103  042105  051040
7385  026020  030513  020061  047111
7386  026026  042524  051122  050125
7387  026034  000124
7388
7389  026036  047103  051124  020114  EM44:   .ASCIZ  /CNTRL RDY DIDN'T SET AFTER SEEK OR DR RESET/
7390  026044  042122  020131  044504
7391  026052  047104  052047  051440
7392  026060  052105  040440  052106
7393  026066  051105  051440  042505
7394  026074  020113  051117  042040
7395  026102  020122  042522  042523
7396  026110  000124
7397
7398  026112  051105  020122  051117  EM45:   .ASCIZ  /ERR OR HE BIT SET ON SEEK OR DR RESET/
7399  026120  044040  020105  044502
7400  026126  020124  042523  020124
7401  026134  047117  051440  042505
7402  026142  020113  051117  042040
7403  026150  020122  042522  042523
7404  026156  000124
7405
7406  026160  045522  051105  041040  EM46:   .ASCIZ  /RKER BIT, ON SEEK OR DR RESET/
7407  026166  052111  020054  047117
7408  026174  051440  042505  020113
7409  026202  051117  042040  020122
7410  026210  042522  042523  000124
7411
7412  026216  045522  051503  041440  EM47:   .ASCIZ  /RKCS CHNGD AFTR FUNCTION WAS DONE/
7413  026224  047110  042107  040440
```

```
 7414  026232  052106  020122  052506
 7415  026240  041516  044524  047117
 7416  026246  053440  051501  042040
 7417  026254  047117  000105
 7418
 7419  026260  027522  027527  020123  EM50:   .ASCIZ  "R/W/S RDY DIDN'T CLEAR"
 7420  026266  042122  020131  044504
 7421  026274  047104  052047  041440
 7422  026302  042514  051101     000
 7423
 7424  026307     122  053457  051457  EM51:   .ASCIZ  "R/W/S RDY DIDN'T SET AFTR SEEK OR DR RESET"
 7425  026314  051040  054504  042040
 7426  026322  042111  023516  020124
 7427  026330  042523  020124  043101
 7428  026336  051124  051440  042505
 7429  026344  020113  051117  042040
 7430  026352  020122  042522  042523
 7431  026360  000124
 7432
 7433  026362  045522  040504  041440  EM52:   .ASCIZ  /RKDA CHNGD AFTR SEEK/
 7434  026370  047110  042107  040440
 7435  026376  052106  020122  042523
 7436  026404  045505     000
 7437
 7438  026407     103  052116  046122  EM53:   .ASCIZ  /CNTRL RDY DIDN'T CLR AS GO WAS SET/
 7439  026414  051040  054504  042040
 7440  026422  042111  023516  020124
 7441  026430  046103  020122  051501
 7442  026436  043440  020117  040527
 7443  026444  020123  042523  000124
 7444
 7445  026452  047103  051124  020114  EM54:   .ASCIZ  "CNTRL RDY DIDN'T SET ON WRT/FMT STARTING FROM <DSK-ADRES>"
 7446  026460  042122  020131  044504
 7447  026466  047104  052047  051440
 7448  026474  052105  047440  020116
 7449  026502  051127  027524  046506
 7450  026510  020124  052123  051101
 7451  026516  044524  043516  043040
 7452  026524  047522  020115  042074
 7453  026532  045523  040455  051104
 7454  026540  051505  000076
 7455
 7456  026544  042510  047440  020122  EM55:   .ASCIZ  "HE OR ERR ON WRT/FMT STARTING FROM <DSK-ADRES>"
 7457  026552  051105  020122  047117
 7458  026560  053440  052122  043057
 7459  026566  052115  051440  040524
 7460  026574  052122  047111  020107
 7461  026602  051106  046517  036040
 7462  026610  051504  026513  042101
 7463  026616  042522  037123     000
 7464
 7465  026623     122  042113  020100  EM56:   .ASCIZ  /RKDA INCRMNTD WRONG ON WRT-FMT/
 7466  026630  047111  051103  047115
 7467  026636  042124  053440  047522
 7468  026644  043516  047440  020116
 7469  026652  051127  026524  046506
```



```
 7470  026660  000124
 7471
 7472  026662  045522  041527  042040  EM57:   .ASCIZ  /RKWC DIDN'T OVRFLO ON WRT FMT/
 7473  026670  042111  023516  020124
 7474  026676  053117  043122  047514
 7475  026704  047440  020116  051127
 7476  026712  020124  046506  000124
 7477
 7478  026720  045522  040502  044440  EM60:   .ASCIZ  /RKBA INCRMNTD WRONG ON WRT FMT/
 7479  026726  041516  046522  052116
 7480  026734  020104  051127  047117
 7481  026742  020107  047117  053440
 7482  026750  052122  043040  052115
 7483  026756     000
 7484
 7485  026757     122  042513  020122  EM61:   .ASCIZ  /RKER SET,ON WRT OR RD OR FMT/
 7486  026764  042523  026124  047117
 7487  026772  053440  052122  047440
 7488  027000  020122  042122  047440
 7489  027006  020122  046506  000124
 7490
 7491  027014  045522  041104  042440  EM62:   .ASCIZ  /RKDB EROR/
 7492  027022  047522  000122
 7493
```

```
 7494  027026  045522  040504  044440  EM63:   .ASCIZ  /RKDA INCRMNTD WRONG ON RD OR RD FMT/
 7495  027034  041516  046522  052116
 7496  027042  020104  051127  047117
 7497  027050  020107  047117  051040
 7498  027056  020104  051117  051040
 7499  027064  020104  046506  000124
 7500
 7501  027072  045522  041527  042040  EM64:   .ASCIZ  /RKWC DIDN'T OVRFLO ON RD OR RD FMT/
 7502  027100  042111  023516  020124
 7503  027106  053117  043122  047514
 7504  027114  047440  020116  042122
 7505  027122  047440  020122  042122
 7506  027130  043040  052115     000
 7507
 7508  027135     122  041113  020101  EM65:   .ASCIZ  /RKBA INCRMNTD WRONG ON RD OR RD FMT/
 7509  027142  047111  051103  047115
 7510  027150  042124  053440  047522
 7511  027156  043516  047440  020116
 7512  027164  042122  047440  020122
 7513  027172  042122  043040  052115
 7514  027200     000
 7515
 7516  027201     111  041516  051117  EM66:   .ASCIZ  /INCORRECT HEADER FROM 'SECTOR'/
 7517  027206  042522  052103  044040
 7518  027214  040505  042504  020122
 7519  027222  051106  046517  023440
 7520  027230  042523  052103  051117
 7521  027236  000047
 7522
 7523  027240  040504  040524  042440  EM67:   .ASCIZ  /DATA ERROR/
 7524  027246  051122  051117     000
 7525
 7526  027253     103  052116  046122  EM70:   .ASCIZ  "CNTRL RDY DIDN'T SET ON RD/FMT STARTING FROM <DSK-ADRES>"
 7527  027260  051040  054504  042040
 7528  027266  042111  023516  020124
 7529  027274  042523  020124  047117
 7530  027302  051040  027504  046506
 7531  027310  020124  052123  051101
 7532  027316  044524  043516  043040
 7533  027324  047522  020115  042074
 7534  027332  045523  040455  051104
 7535  027340  051505  000076
 7536
 7537  027344  042510  047440  020122  EM71:   .ASCIZ  "HE OR ERR ON RD/FMT STARTING FROM <DSK-ADRES>"
 7538  027352  051105  020122  047117
 7539  027360  051040  027504  046506
 7540  027366  020124  052123  051101
 7541  027374  044524  043516  043040
 7542  027402  047522  020115  042074
 7543  027410  045523  040455  051104
 7544  027416  051505  000076
 7545
 7546  027422  051127  047117  020107  EM72:   .ASCIZ  /WRONG DRIVE ID IN RKDS AFTER SEEK/
 7547  027430  051104  053111  020105
 7548  027436  042111  044440  020116
 7549  027444  045522  051504  040440
```



```
 7550  027452  052106  051105  051440
 7551  027460  042505  000113
 7552
 7553  027464  051110  053504  042522  EM73:   .ASCIZ  /HRDWRE POLL-DRV ID BITS(13-15) SHLDBE CLR/
 7554  027472  050040  046117  026514
 7555  027500  051104  020126  042111
 7556  027506  041040  052111  024123
 7557  027514  031461  030455  024465
 7558  027522  051440  046110  041104
 7559  027530  020105  046103  000122
 7560
 7561  027536  051110  053504  042522  EM74:   .ASCIZ  /HRDWRE POLL-INTRUPTING DRIV # NOT PRSNT/
 7562  027544  050040  046117  026514
 7563  027552  047111  051124  050125
 7564  027560  044524  043516  042040
 7565  027566  044522  020126  020043
 7566  027574  047516  020124  051120
 7567  027602  047123  000124
 7568
 7569  027606  051104  053111  021440  EM75:   .ASCIZ  /DRIV # DIDN'T INTRUPT AFTER HRDWRE POLL/
 7570  027614  042040  042111  023516
 7571  027622  020124  047111  051124
 7572  027630  050125  020124  043101
 7573  027636  042524  020122  051110
 7574  027644  053504  042522  050040
 7575  027652  046117  000114
 7576
 7577  027656  041523  020120  044504  EM76:   .ASCIZ  /SCP DIDN'T SET AFTER SEEK WAS DONE/
 7578  027664  047104  052047  051440
 7579  027672  052105  040440  052106
 7580  027700  051105  051440  042505
 7581  027706  020113  040527  020123
 7582  027714  047504  042516     000
 7583
 7584  027721     122  042113  020101  EM77:   .ASCIZ  /RKDA CHANGD AFTER DRIV RESET/
 7585  027726  044103  047101  042107
 7586  027734  040440  052106  051105
 7587  027742  042040  044522  020126
 7588  027750  042522  042523  000124
 7589
 7590  027756  040504  040524  042440  EM100:  .ASCIZ  /DATA EROR AT WORD#/
 7591  027764  047522  020122  052101
 7592  027772  053440  051117  021504
 7593  030000     000
 7594
 7595  030001     103  052116  046122  EM101:  .ASCIZ  /CNTRL RDY DIDN'T SET AFTER RD CHK/
 7596  030006  051040  054504  042040
 7597  030014  042111  023516  020124
 7598  030022  042523  020124  043101
 7599  030030  042524  020122  042122
 7600  030036  041440  045510     000
 7601
 7602  030043     105  051122  047440  EM102:  .ASCIZ  /ERR OR HE ON RD CHK/
 7603  030050  020122  042510  047440
 7604  030056  020116  042122  041440
 7605  030064  045510     000
```

```
7606
7607  030067  103  042523  047440  EM103:  .ASCIZ  /CSE ON RD CHK/
7608  030074  020116  042122  041440
7609  030102  045510  000
7610
7611  030105  122  053513  020103  EM104:  .ASCIZ  /RKWC DIDN'T OVERFLO ON RD CHK OR WRT CHK/
7612  030112  044504  047104  052047
7613  030120  047440  042526  043122
7614  030126  047514  047440  020116
7615  030134  042122  041440  045510
7616  030142  047440  020122  051127
7617  030150  020124  044103  000113
7618
7619  030156  045522  040504  044440  EM105:  .ASCIZ  /RKDA INCRMNTD WRONG ON RD CHK/
7620  030164  041516  046522  052116
7621  030172  020104  051127  047117
7622  030200  020107  047117  051040
7623  030206  020104  044103  000113
7624
7625  030214  045522  040502  041440  EM106:  .ASCIZ  /RKBA CHANGD AFTER RD CHK/
7626  030222  040510  043516  020104
7627  030230  043101  042524  020122
7628  030236  042122  041440  045510
7629  030244  000
7630
7631  030245  115  046505  051117  EM107:  .ASCIZ  /MEMORY WORD CHANGED AFTER RD CHK/
7632  030252  020131  047527  042122
7633  030260  041440  040510  043516
7634  030266  042105  040440  052106
7635  030274  051105  051040  020104
7636  030302  044103  000113
7637
7638  030306  047103  051124  020114  EM110:  .ASCIZ  /CNTRL RDY DIDN'T SET AFTER WRT CHK/
7639  030314  042122  020131  044504
7640  030322  047104  052047  051440
7641  030330  052105  040440  052106
7642  030336  051105  053440  052122
7643  030344  041440  045510  000
7644
7645  030351  110  020105  051117  EM111:  .ASCIZ  /HE OR ERR ON WRT CHK/
7646  030356  042440  051122  047440
7647  030364  020116  051127  020124
7648  030372  044103  000113
7649
7650  030376  051127  052111  020105  EM112:  .ASCIZ  /WRITE CHECK EROR/
7651  030404  044103  041505  020113
7652  030412  051105  051117  000
7653
7654  030417  122  042113  020101  EM113:  .ASCIZ  /RKDA INCRMNTD WRONG ON WRT CHK/
7655  030424  047111  051103  047115
7656  030432  042124  053440  047522
7657  030440  043516  047440  020116
7658  030446  051127  020124  044103
7659  030454  000113
7660
7661  030456  045522  040502  044440  EM114:  .ASCIZ  /RKBA INCRMNTD WRONG ON WRT CHK/
```



```
7662  030464  041516  046522  052116
7663  030472  020104  051127  047117
7664  030500  020107  047117  053440
7665  030506  052122  041440  045510
7666  030514  000
7667
7668  030515  122  041113  020101  EM115:  .ASCIZ  /RKBA INCRMNTD, WITH IBA SET/
7669  030522  047111  051103  047115
7670  030530  042124  020054  044527
7671  030536  044124  044440  040502
7672  030544  051440  052105  000
7673
7674  030551  127  047522  043516  EM116:  .ASCIZ  /WRONG MEMORY LOCATION CHANGED WITH IBA SET/
7675  030556  046440  046505  051117
7676  030564  020131  047514  040503
7677  030572  044524  047117  041440
7678  030600  040510  043516  042105
7679  030606  053440  052111  020110
7680  030614  041111  020101  042523
7681  030622  000124
7682
7683  030624  045522  030461  042040  EM117:  .ASCIZ  /RK11 DIDN'T INTRUPT WHEN IDE WAS SET/
7684  030632  042111  023516  020124
7685  030640  047111  051124  050125
7686  030646  020124  044127  047105
7687  030654  044440  042504  053440
7688  030662  051501  051440  052105
7689  030670  000
7690
7691  030671  122  030513  020061  EM120:  .ASCIZ  /RK11 DIDN'T INTRUPT AFTER SK WAS INITIATED/
7692  030676  044504  047104  052047
7693  030704  044440  052116  052522
7694  030712  052120  040440  052105
7695  030720  051105  051440  020113
7696  030726  040527  020123  047111
7697  030734  052111  040511  042524
7698  030742  000104
7699
7700  030744  041523  020120  042523  EM121:  .ASCIZ  /SCP SET BEFORE SEEK COMPLETED/
7701  030752  020124  042502  047506
7702  030760  042522  051440  042505
7703  030766  020113  047503  050115
7704  030774  042514  042524  000104
7705
7706  031002  045522  030461  042040  EM122:  .ASCIZ  /RK11 DIDN'T INTRUPT AFTER SK COMPLETED/
7707  031010  042111  023516  020124
7708  031016  047111  051124  050125
7709  031024  020124  043101  042524
7710  031032  020122  045523  041440
7711  031040  046517  046120  052105
7712  031046  042105  000
7713
7714  031051  103  052116  046122  EM123:  .ASCIZ  /CNTRL RESET DIDN'T CLEAR 'SCP'/
7715  031056  051040  051505  052105
7716  031064  042040  042111  023516
7717  031072  020124  046103  040505
```

```
 7718  031100  020122  051447  050103
 7719  031106  000047
 7720
 7721  031110  045522  030461  042040  EM124:  .ASCIZ  /RK11 DIDN'T INTRUPT AFTER RD DONE/
 7722  031116  042111  023516  020124
 7723  031124  047111  051124  050125
 7724  031132  020124  043101  042524
 7725  031140  020122  042122  042040
 7726  031146  047117  000105
 7727
 7728  031152  047103  051124  020114  EM125:  .ASCIZ  /CNTRL RESET DIDN'T CLR REGISTR/
 7729  031160  042522  042523  020124
 7730  031166  044504  047104  052047
 7731  031174  041440  051114  051040
 7732  031202  043505  051511  051124
 7733  031210     000
 7734
 7735  031211     122  030513  020061  EM126:  .ASCIZ  /RK11 DIDN'T INTRUPT AT CPU LEVEL/
 7736  031216  044504  047104  052047
 7737  031224  044440  052116  052522
 7738  031232  052120  040440  020124
 7739  031240  050103  020125  042514
 7740  031246  042526  000114
 7741
 7742  031252  045522  030461  044440  EM127:  .ASCIZ  /RK11 INTRUPTED AT WRONG CPU LEVEL/
 7743  031260  052116  052522  052120
 7744  031266  042105  040440  020124
 7745  031274  051127  047117  020107
 7746  031302  050103  020125  042514
 7747  031310  042526  000114
 7748
 7749  031314  042447  051122  041040  EM130:  .ASCIZ  /'ERR BIT' DIDN'T SET IN RKER/
 7750  031322  052111  020047  044504
 7751  031330  047104  052047  051440
 7752  031336  052105  044440  020116
 7753  031344  045522  051105     000
 7754
 7755  031351     110  020105  051117  EM131:  .ASCIZ  /HE OR ERR DIDN'T SET/
 7756  031356  042440  051122  042040
 7757  031364  042111  023516  020124
 7758  031372  042523  000124
 7759
 7760  031376  045522  051105  042440  EM132:  .ASCIZ  /RKER EROR/
 7761  031404  047522  000122
 7762
 7763  031410  054116  020103  044502  EM133:  .ASCIZ  /NXC BIT DIDN'T SET/
 7764  031416  020124  044504  047104
 7765  031424  052047  051440  052105
 7766  031432     000
 7767
 7768  031433     122  030513  020061  EM134:  .ASCIZ  /RK11 DIDN'T INTRUPT ON SOFT EROR/
 7769  031440  044504  047104  052040
 7770  031446  044440  052116  052522
 7771  031454  052120  047440  020116
 7772  031462  047523  052106  042440
 7773  031470  047522  000122
```



```
 7774
 7775  031474  042515  020130  044502  EM135:  .ASCIZ  /MEX BITS INCRMNTD WRONG-RKCS/
 7776  031502  051524  044440  041516
 7777  031510  046522  052116  020104
 7778  031516  051127  047117  026507
 7779  031524  045522  051503     000
 7780
 7781  031531     127  051520  047040  EM137:  .ASCIZ  /WPS NOT CLEAR/
 7782  031536  052117  041440  042514
 7783  031544  051101     000
 7784
 7785  031547     104  052101  020101  EM140:  .ASCIZ  /DATA EROR ON TRANSFER FROM DISK TO TTY/
 7786  031554  051105  051117  047440
 7787  031562  020116  051124  047101
 7788  031570  043123  051105  043040
 7789  031576  047522  020115  044504
 7790  031604  045523  052040  020117
 7791  031612  052124  000131
 7792
 7793  031616  042047  044522  020126  EM141:  .ASCIZ  /'DRIV #' PRESENT, BUT NOT INDICATED/
 7794  031624  023443  050040  042522
 7795  031632  042523  052116  020054
 7796  031640  052502  020124  047516
 7797  031646  020124  047111  044504
 7798  031654  040503  042524  000104
 7799  031662  047040  020117  052502  EM142:  .ASCIZ  / NO BUSY ON OTHER HALF OF RK-05F/
 7800  031670  054523  047440  020116
 7801  031676  052117  042510  020122
 7802  031704  040510  043114  047440
 7803  031712  020106  045522  030055
 7804  031720  043065     000
 7805
 7806
 7807
 7808
 7809
 7810          031724                          .EVEN
 7811
 7812                                          .SBTTL  ERROR DATA POINTERS
 7813
 7814  031724  001116  001162  000000  DT1:    .WORD   $ERRPC,$REG0,0
 7815
 7816  031732  001116  001162  001164  DT2:    .WORD   $ERRPC,$REG0,$REG1,0
 7817  031740  000000
 7818
 7819  031742  001116  001162  001164  DT20:   .WORD   $ERRPC,$REG0,$REG1,$REG2,$REG3,0
 7820  031750  001166  001170  000000
 7821
 7822  031756  001116  000000          DT21:   .WORD   $ERRPC,0
 7823
 7824  031762  001116  001162  001164  DT26:   .WORD   $ERRPC,$REG0,$REG1,$REG2,0
 7825  031770  001166  000000
 7826
 7827  031774  001116  001162  001164  DT54:   .WORD   $ERRPC,$REG0,$REG1,$REG2,$REG3,$REG4,$REG5,$REG6,$REG7,0
 7828  032002  001166  001170  001172
 7829  032010  001174  001176  001200
```

```
7830  032016  000000
7831
7832
7833
7834                                        .SBTTL  ERROR HEADERS
7835
7836
7837  032020  020040  041520  020040  DH2:    .ASCIZ  /  PC   REGADD    RECVD/
7838  032026  051040  043505  042101
7839  032034  020104  020040  051040
7840  032042  041505  042126     000
7841
7842  032047     040  050040  020103  DH4:    .ASCIZ  /  PC    EXPCT   RECVD/
7843  032054  020040  042440  050130
7844  032062  052103  020040  051040
7845  032070  041505  042126     000
7846
7847  032075     040  050040  020103  DH5:    .ASCIZ  /  PC    RECVD/
7848  032102  020040  051040  041505
7849  032110  042126     000
7850
7851  032113     040  050040  020103  DH14:   .ASCIZ  /  PC    RKCS    RKER    RKWC/
7852  032120  020040  051040  041513
7853  032126  020123  020040  051040
7854  032134  042513  020122  020040
7855  032142  051040  053513  000103
7856
7857  032150  020040  041520     000  DH21:   .ASCIZ  /  PC/
7858
7859  032155     040  050040  020103  DH30:   .ASCIZ  /  PC     RKCS    RKER    RKDS/
7860  032162  020040  020040  045522
7861  032170  051503  020040  020040
7862  032176  045522  051105  020040
7863  032204  020040  045522  051504
7864  032212     000
7865
7866  032213     040  050040  020103  DH34:   .ASCIZ  /  PC     RKDS/
7867  032220  020040  020040  045522
7868  032226  051504     000
7869
7870  032231     040  050040  020103  DH35:   .ASCIZ  /  PC   SEC-CNTR/
7871  032236  020040  042523  026503
7872  032244  047103  051124     000
7873
7874  032251     040  050040  020103  DH36:   .ASCIZ  /  PC     PRSNT  NXT-CNT/
7875  032256  020040  020040  051120
7876  032264  047123  020124  047040
7877  032272  052130  041455  052116
7878  032300     000
7879
7880  032301     040  050040  020103  DH40:   .ASCIZ  /  PC    SECTOR   RKDS/
7881  032306  020040  051440  041505
7882  032314  047524  020122  020040
7883  032322  045522  051504     000
7884
7885  032327     040  050040  020103  DH44:   .ASCIZ  /  PC    RKCS    RKER    RKDS    RKDA/
```



```
7886  032334  020040  051040  041513
7887  032342  020123  020040  051040
7888  032350  042513  020122  020040
7889  032356  051040  042113  020123
7890  032364  020040  051040  042113
7891  032372  000101
7892
7893  032374  020040  041520  020040  DH54:   .ASCIZ  /  PC    RKCS    RKER    RKDS    RKDA   DRV#....CYL.<DSK-ADRS>.SUR..SEC/
7894  032402  020040  045522  051503
7895  032410  020040  020040  045522
7896  032416  051105  020040  020040
7897  032424  045522  051504  020040
7898  032432  020040  045522  040504
7899  032440  020040  042040  053122
7900  032446  027043  027056  041456
7901  032454  046131  036056  051504
7902  032462  026513  042101  051522
7903  032470  027076  052523  027122
7904  032476  051456  041505     000
7905
7906  032503     040  041520  020040  DH56:   .ASCIZ  / PC  EXPC: DR#   CYL     SUR    SEC  RECV: DR#   CYL     SUR     SEC/
7907  032510  054105  041520  020072
7908  032516  051104  020043  020040
7909  032524  054503  020114  020040
7910  032532  020040  052523  020122
7911  032540  020040  051440  041505
7912  032546  020040  042522  053103
7913  032554  020072  051104  020043
7914  032562  020040  054503  020114
7915  032570  020040  020040  052523
7916  032576  020122  020040  020040
7917  032604  042523  000103
7918
7919  032610  020040  041520  020040  DH64:   .ASCIZ  /  PC    RKWC   RKDA/
7920  032616  020040  045522  041527
7921  032624  020040  051040  042113
7922  032632  000101
7923
7924  032634  020040  041520  020040  DH66:   .ASCIZ  /  PC    SECTR   EXPCT   RECVD/
7925  032642  020040  042523  052103
7926  032650  020122  020040  054105
7927  032656  041520  020124  020040
7928  032664  042522  053103  000104
7929
7930  032672  020040  041520  020040  DH67:   .ASCIZ  /  PC    EXPCT   RECVD  DSK-ADRS/
7931  032700  020040  054105  041520
7932  032706  020124  020040  042522
7933  032714  053103  020104  042040
7934  032722  045523  040455  051104
7935  032730  000123
7936
7937  032732  020040  041520  020040  DH74:   .ASCIZ  /  PC      DRIV #/
7938  032740  020040  020040  051104
7939  032746  053111  021440     000
7940
7941  032753     040  050040  020103  DH100:  .ASCIZ  /  PC  WORD #   EXPCT   RECVD/
```

```
 7942  032760  053440  051117  020104
 7943  032766  020043  020040  054105
 7944  032774  041520  020124  020040
 7945  033002  042522  053103  000104
 7946
 7947  033010  020040  041520  020040  DH103:  .ASCIZ  /  PC   RKER/
 7948  033016  051040  042513  000122
 7949
 7950  033024  020040  041520  020040  DH104:  .ASCIZ  /  PC   RECVD   RKCS/
 7951  033032  051040  041505  042126
 7952  033040  020040  051040  041513
 7953  033046  000123
 7954
 7955  033050  020040  041520  020040  DH107:  .ASCIZ  /  PC     LOC    EXPCT   RECVD/
 7956  033056  020040  046040  041517
 7957  033064  020040  020040  054105
 7958  033072  041520  020124  020040
 7959  033100  042522  053103  000104
 7960
 7961  033106  020040  041520  020040  DH117:  .ASCIZ  /  PC   RKCS/
 7962  033114  051040  041513  000123
 7963
 7964  033122  020040  041520  020040  DH126:  .ASCIZ  /  PC    LEVEL    RKCS/
 7965  033130  020040  042514  042526
 7966  033136  020114  020040  051040
 7967  033144  041513  000123
 7968
 7969  033150  020040  041520  020040  DH130:  .ASCIZ  /  PC     RKCS    RKER  ERR BIT/
 7970  033156  020040  051040  041513
 7971  033164  020123  020040  051040
 7972  033172  042513  020122  042440
 7973  033200  051122  041040  052111
 7974  033206     000
 7975
 7976  033207     040  050040  020103  DH131:  .ASCIZ  /  PC     RKCS    RKER/
 7977  033214  020040  020040  045522
 7978  033222  051503  020040  020040
 7979  033230  045522  051105     000
 7980
 7981  033235     040  050040  020103  DH133:  .ASCIZ  /  PC     RKCS    RKER    RKDA/
 7982  033242  020040  020040  045522
 7983  033250  051503  020040  020040
 7984  033256  045522  051105  020040
 7985  033264  020040  045522  040504
 7986  033272     000
 7987
 7988  033273     040  050040  020103  DH140:  .ASCIZ  /  PC    EXPCT   RECVD    RKBA    RKCS/
 7989  033300  020040  042440  050130
 7990  033306  052103  020040  051040
 7991  033314  041505  042126  020040
 7992  033322  020040  045522  040502
 7993  033330  020040  020040  045522
 7994  033336  051503     000
 7995
 7996
 7997          033342                        .EVEN
```



```
 7998                                  ;DATA BUFFER
 7999
 8000  033342  000400                  OUTBUF: .BLKW   256.    ;THIS 256 WORD BUFFER IS FOR
 8001                                                          ;DATA TRANSFERS FROM AND
 8002                                                          ;TO THE DISK.
 8003
 8004          000001                          .END
```

```
BADINT  004600          2108   2136#  4643   4688   4707   4718   4798   4806   4895   4904   5548
BADTMO  004534          2043   2106   2117#  5657   5661
BDAR    021074          6182   6187#
BDA0    021056          6180#  7216
BDA4    021066          6184#  7218
BIT0  = 000001           993#  2002   2006   2010   2270   5904
BIT00 = 000001           983#   993
BIT01 = 000002           982#   992
BIT02 = 000004           981#   991
BIT03 = 000010           980#   990
BIT04 = 000020           979#   989
BIT05 = 000040           978#   988
BIT06 = 000100           977#   987
BIT07 = 000200           976#   986
BIT08 = 000400           975#   985   6625
BIT09 = 001000           974#   984   6633
BIT1  = 000002           992#
BIT10 = 002000           973#
BIT11 = 004000           972#  6640
BIT12 = 010000           971#  6723
BIT13 = 020000           970#  5908
BIT14 = 040000           969#  6611
BIT15 = 100000           968#  2001   2006   2010   6720   6722
BIT2  = 000004           991#
BIT3  = 000010           990#
BIT4  = 000020           989#
BIT5  = 000040           988#
BIT6  = 000100           987#  6678
BIT7  = 000200           986#
BIT8  = 000400           985#
BIT9  = 001000           984#
BPTVEC= 000014          1000#
BRKDA0= 104415          3697   6288   7216#
BRKDA4= 104416          2834   2930   3032   3160   3249   3370   3538   3670   3700   3819   4179   4265   4523
                        5693   5709   6268   6291   7218#
BTEOP   020026          5820#  6731
CHE1    021242          6262   6265#
CHKCCL  021402          4987   5055   5124   5203   5364   5430   6340#
CHKCRD= 104412          2612   2672   3925   4334   4428   5243   5410   5470   5606   5759   7210#
CHKDA   021262          2847   2942   3045   4206   6282#
CHKDA1  021270          3175   3265   3395   3559   6284#
CHKECL  021356          4983   5051   5120   5199   5286   5360   5426   6329#
CHKER   021342          2861   2957   3059   3295   4210   4808   6316#
CHKHE   021234          2840   2936   3039   3256   4187   4527   6264#
CHKHE1  021226          3166   3377   3552   3687   3833   6261#
CHKWC   021316          2851   2947   3049   3278   3726   4202   4819   6302#
CH.CRD  021760          6522#  7210
CKSWR = 104407          6610   6672   6695   7206#
CNT.RD= 104414          3967   4025   4043   4092   5021   5321   6393   7214#  7284   7286
CNT.RE= 104413          2310   2357   2394   2446   2531   2566   2598   2656   2722   2776   2895   2996   3098
                        3128   3221   3345   3452   3508   3637   3792   3906   4003   4120   4234   4308   4397
                        4489   4596   4662   4748   4769   4847   4927   4973   5004   5041   5070   5110   5139
                        5189   5226   5259   5276   5305   5350   5380   5425   5448   5513   5587   5667   5754
                        5876   7212#  7275
CN.RDY  022032          6561   6562#  7214
CN.RST  022014          6559#  7212
```



```
COUNT   001362          1135#  3410*  3414*  3520*  3533*  4516*  4519*  5221*  5264*
COUNT1  001364          1136#  6390*  6405*
CR    = 000015           908#  6842   6852
CRETRN  021430          6266   6285   6303   6317   6330   6341   6346#
CRLF  = 000200           909#  1884   6813   6852
DDISP = 177570           915#  1057   1855
DDPCH   001410          1156#  1885   1905   1912*  1914   1916*  1920   1929   1941   2197   2199   2275   2284
DELAY = 104417          2342   5155   5166   5584   7220#
DELA.Y  021664          6456#  7220
DH100   032753          1565   1663   7941#
DH103   033010          1586   7947#
DH104   033024          1593   7950#
DH107   033050          1614   7955#
DH117   033106          1551   1670   1677   1684   1698   1705   7961#
DH126   033122          1719   1726   7964#
DH130   033150          1733   7969#
DH131   033207          1741   1762   7976#
DH133   033235          1755   7981#
DH14    032113          1248   1776   7851#
DH140   033273          1790   7988#
DH2     032020          1712   7837#
DH21    032150          1346   7857#
DH30    032155          1269   1276   1367   1381   1402   1452   1579   1628   1635   1691   7859#
DH34    032213          1262   1283   1297   1332   1530   7866#
DH35    032231          1304   7870#
DH36    032251          1311   7874#
DH4     032047          1255   1318   1339   1374   1395   1445   1459   1482   1523   1558   1600   1607   1642
                        1649   1656   1748   1769   1805   7842#
DH40    032301          1325   7880#
DH44    032327          1241   1290   1353   1360   1388   1572   1621   1783   7885#
DH5     032075          1438   7847#
DH54    032374          1410   1420   1504   1514   7893#
DH56    032503          1429   1466   7906#
DH64    032610          1475   7919#
DH66    032634          1489   7924#
DH67    032672          1496   7930#
DH74    032732          1537   1544   1798   7937#
DISPLA  001142          1057#  1855*  1863*  6654*  6675*
DISPRE  000174          1015#  1863
DRESET  021504          2734   4674   4991   5260   5818   6390#  6428   7273   7291
DRHOLD  025302          7276*  7285   7290   7294#
DRIVAD  001350          1127#  2104*  2272*  2277   2289   2321   2367   2404   2416   2456   2541   2567   2610
                        2669   2738   2818   2833   2913   2929   3015   3031   3068   3123   3236   3248   3262
                        3332   3505   3620   3784   3920   4015   4161   4178   4254   4264   4326   4356   4419
                        4454   4511   4522   4549   4673   4727   4730   4749   4785   4862   4939   5017   5086
                        5222   5316   5407   5465   5530   5599   5681   5757   5823*  6264   6282   6360   6391
                        6425   6716   6725   7276   7277   7279*  7281*  7282   7290*  7295*  7307*
DRIVS   001412          1158#  1959   1996*  2011*  2047*  2068*  2077   2097   2237*  2265   5820   5862   6711*
                        6729*
DRIV0   001414          1170#  1973   2050   2101   2194   2232   2234   2297   5851   5891   6719   7296
DRIV1   001416          1171#
DRIV2   001420          1172#
DRIV3   001422          1173#
DRIV4   001424          1174#
DRIV5   001426          1175#
DRIV6   001430          1176#
```

```
DRIV7   001432          1177#   5859    7308
DRVDON  001352          1128#   2102*   5816*   5820
DRVPTR  001354          1131#   2101*   2268    2286*   6714
DSWR    = 177570         914#   1056    1854
DT1     031724          1263    1284    1298    1305    1333    1439    1538    1545    1552    1587    1671    1678    1685
                        1699    1706    1799    7814#
DT2     031732          1256    1312    1319    1326    1340    1375    1396    1446    1460    1476    1483    1524    1531
                        1559    1594    1601    1608    1643    1650    1657    1713    1720    1727    1742    1749    1763
                        1770    1806    7816#
DT20    031742          1242    1291    1354    1361    1389    1573    1622    1784    1791    7819#
DT21    031756          1347    7822#
DT26    031762          1249    1270    1277    1368    1382    1403    1453    1490    1497    1566    1580    1615    1629
                        1636    1664    1692    1734    1756    1777    7824#
DT54    031774          1412    1422    1431    1468    1506    1516    7827#
EFLG1   001370          1138#   3408*   3429*   3629*   3633*   3655    3706
EMTVEC= 000030          1003#   1839*   1840*
EM100   027756          1564    7590#
EM101   030001          1571    7595#
EM102   030043          1578    7602#
EM103   030067          1585    7607#
EM104   030105          1592    7611#
EM105   030156          1599    7619#
EM106   030214          1606    7625#
EM107   030245          1613    7631#
EM11    025370          1804    7316#
EM110   030306          1620    1775    7638#
EM111   030351          1627    7645#
EM112   030376          1634    7650#
EM113   030417          1641    7654#
EM114   030456          1648    7661#
EM115   030515          1655    7668#
EM116   030551          1662    7674#
EM117   030624          1669    7683#
EM12    025402          1240    7320#
EM120   030671          1676    7691#
EM121   030744          1683    7700#
EM122   031002          1690    7706#
EM123   031051          1697    7714#
EM124   031110          1704    7721#
EM125   031152          1711    7728#
EM126   031211          1718    7735#
EM127   031252          1725    7742#
EM13    025415          1338    7323#
EM130   031314          1732    7749#
EM131   031351          1740    7755#
EM132   031376          1747    7760#
EM133   031410          1754    7763#
EM134   031433          1761    7768#
EM135   031474          1768    7775#
EM137   031531          1782    7781#
EM140   031547          1789    7785#
EM141   031616          1797    7793#
EM142   031662          1809    7799#
EM16    025427          1254    7326#
EM21    025456          1261    7331#
EM30    025470          1268    7334#
```



```
EM31    025500          1275    7337#
EM32    025510          1282    7340#
EM33    025531          1289    7344#
EM34    025551          1296    7348#
EM35    025570          1303    7352#
EM36    025623          1310    7358#
EM37    025653          1317    7364#
EM40    025707          1324    7370#
EM41    025747          1331    7377#
EM43    026004          1345    2143    7383#
EM44    026036          1352    7389#
EM45    026112          1359    7398#
EM46    026160          1366    7406#
EM47    026216          1373    7412#
EM50    026260          1380    7419#
EM51    026307          1387    7424#
EM52    026362          1394    7433#
EM53    026407          1401    7438#
EM54    026452          1408    7445#
EM55    026544          1418    7456#
EM56    026623          1428    7465#
EM57    026662          1437    7472#
EM60    026720          1444    7478#
EM61    026757          1451    7485#
EM62    027014          1458    7491#
EM63    027026          1465    7494#
EM64    027072          1474    7501#
EM65    027135          1481    7508#
EM66    027201          1488    7516#
EM67    027240          1495    7523#
EM70    027253          1247    1502    7526#
EM71    027344          1512    7537#
EM72    027422          1522    7546#
EM73    027464          1529    7553#
EM74    027536          1536    7561#
EM75    027606          1543    7569#
EM76    027656          1550    7577#
EM77    027721          1557    7584#
ERRVEC= 000004           996#   1852    1853*   1864*   1892    1893*   1901*   6616    6617*   6619*   6622*
FCHECK  025160          5801    7273#   7301
FFLAG   001404          1152#   2291*   2303*   5799
FTITLE  001346          1126#
GNS     = ****** U      1014    1883    1926    1938    1947    1951    1966    1990    2122    2147    2301    6151    6435
                        6573    6581    7198    7199    7200    7201    7202    7204    7206    7207    7208    7210    7212
                        7214    7216    7218    7220    7222    7224
GTSWR = 104406          1878    7204#
GT2RG   021010          4956    4963    4969    5030    5037    5092    5100    5106    5170    5186    5251    5272    5331
                        5346    5416    5422    5541    6132#
GT3RG   021002          2335    2371    2701    2746    2823    2918    3020    3547    3681    3829    4167    4332    4341
                        4425    4435    4440    4709    6131#   6319
GT4RG   020774          2222    2382    2407    2580    2626    2632    2637    2683    2687    2696    2832    2928    3030
                        3157    3247    3368    3536    3667    3817    3936    3941    3945    4068    4074    4177    4263
                        4521    5691    5707    5925    6130#   6267    6407    6527
HT      = 000011         906#   6811    6852
INDX1   001356          1133#   2458*   2467*   3618*   3735    3737*   3743*   3804*   3814*   3898*   3977    3982*   5765*
                        5867*   5886    6027    6029*
```

```
INDX2   001360          1134#  2465*  2469*  3652*  3664*  5781*  5872*  5954   5975*
IOTVEC= 000020          1001#  1837*  1838*
LF    = 000012           907#  6846   6852
MSG1    001216          1084#  2045
MSG2    001236          1088#  2080
MSG3    001245          1091#  6570
MSG4    001272          1096#  2288   6724
MSG5    001303          1099#  6710
MSG6    001315          1102#  6728
NUDRV   005040          2250#  5824
ODDEVN  001406          1153#  5861*  5910   5913   6055*  6056
OUTBUF  033342          2788   2813   2855   2857   2909   2951   2953   3011   3053   3055   3141*  3146   3237
                        3272   3274   3334   3409   3521*  3525   3651   3781   3964   3968   3972   4014   4147
                        4160   4247   4253   4277   4324*  4327   4365   4367   4372   4378   4380   4411   4418
                        4462   4464   4503   4789   4829   4831   4943   5319   5392   5402*  5408   5461*  5467
                        5525*  5532   5677   5893   5898   5982   6013   8000#
PFSTRT  004702          2174#  7264
PHYDRV  001436          1160#  5850*  5854*  5981   6012
PIRQ  = 177772           913#
PIRQVE= 000240          1007#
PR0   = 000000           930#
PR1   = 000040           931#
PR2   = 000100           932#
PR3   = 000140           933#
PR4   = 000200           934#
PR5   = 000240           935#
PR6   = 000300           936#
PR7   = 000340           937#
PS    = 177776           910#   911    1894
PSW   = 177776           911#
PWRVEC= 000024          1002#  1843*  1844*  7231*  7232*  7241*  7247*  7259*  7260*
RDCHR = 104410          7113   7207#
RDLIN = 104411          1968   7208#
RESVEC= 000010           997#
RKBA    001336          1116#  2815   2908   3010   3146*  3238*  3272   3275   3358*  3525*  3653*  3805*  3964*
                        4020*  4086*  4160*  4195   4198   4253*  4323   4418*  4462   4465   4510*  4534   4537
                        4784   4943*  5085*  5319*  5406*  5467*  5532*  5600*  5678   5751
RKCS    001332          1114#  2327*  2608   2668   2741*  2814   2907   3008   3120   3152   3233   3240*  3242
                        3331   3529*  3531   3660*  3662   3610*  3812   3900   4002   4132   4172   4256*  4258
                        4320   4409   4501   4612   4672   4782   4861   4946*  4949   4967   5019*  5035   5083
                        5088   5151   5220   5315   5379   5460   5537*  5566*  5579   5666   5758*  5866   6133
                        6265   6340   6342   6343   6392*  6523   6559*  6563   6583   7283*
RKDA    001340          1117#  1818*  2042   2051*  2052   2201*  2321*  2367*  2404*  2416*  2456*  2544*  2567*
                        2610*  2669*  2738*  2739*  2818*  2913*  3015*  3122   3235   3357*  3527*  3658*  3694
                        3699   3808*  3922*  3957   3960   3963*  4018*  4078   4081   4087*  4161*  4254*  4322
                        4419*  4456   4459   4511*  4673*  4680*  4749*  4783   4862*  4944*  5017*  5086*  5154*
                        5241*  5252   5316*  5317*  5407*  5468*  5486   5491   5533*  5551   5556   5599*  5683*
                        5697*  5757*  5928*  6130   6284   6290   6306   6360*  6391*  6425*  7282*  7285*
RKDB    001342          1118#  3084   3087   3099   3101   3102   3181   3186   3462   3464   3465   3854   3857
RKDS    001326          1112#  1820   2041   2063   2202   2320   2345   2368   2377   2380   2405   2418   2422
                        2457   2505   2542   2569   2619   2630   2679   2685   2740   3545   3679   3827   3901
                        4051   4273   4540   4725   4750   4752   5157   5965   5983   6131   6362   6366   6399
                        6426   7287
RKER    001330          1113#  2699   3902   4344   4346   4438   4961   5016   5098   5168   5249   5328   5339
                        5414   5476   5480   5623   5629   6132   6316   6329   6331   6332
RKPRI   001400          1144#  1897*  4865   5945   6487
```



```
RKVEC   001402          1149#  2108*  4613   4631*  4643*  4677   4688*  4697*  4707*  4718*  4790   4798*  4806*
                        4869   4895*  4904*  5534   5548*  5942
RKWC    001334          1115#  2817*  2912*  3014*  3148*  3239*  3359*  3526*  3655*  3807*  3965*  4019*  4085*
                        4159*  4252*  4321   4417*  4448   4450   4509*  4787*  4942*  4955   5084*  5093   5318*
                        5405*  5466*  5531*  5598*  5680*  5696*  5755*  6302   6305
SEEK0   001372          1141#  3896   3985
SEEK1   001374          1142#
SEEK2   001376          1143#  3980
SHFTRT  021200          2290   4731   4734   5999   6018   6228#  6717   6726
SIMUL   001344          1125#  2030*  2033*  3335   3433   3622   3732   3785   3866   4000
SIZEF   025304          5849   7295#
SIZYET  001440          1181#  2027*  2039*  5847*
STACK = 001100           901#  1835
START   002636          1018   1814#  2127   2164
STARTA  002704          1817   1821   1828#
START1  003334          1891   1905#
STKLMT= 177774           912#
ST2     003736          1911   1935   1959#  1992
ST3     004262          1922   1933   2039#
ST4     004456          2031   2034   2078   2081   2096#  6096
SWR     001140          1056#  1833   1854*  1856   1862*  1876   2028   2138   2152   2157   6187   6431   6567
                        6611   6625   6627   6633   6640   6678   6686   6692   6696   6699   7009   7046*  7239
                        7252*
SWREG   000176          1016#  1862   1876   7009   7022
SW0   = 000001           965#
SW00  = 000001           955#   965
SW01  = 000002           954#   964
SW02  = 000004           953#   963
SW03  = 000010           952#   962
SW04  = 000020           951#   961
SW05  = 000040           950#   960
SW06  = 000100           949#   959
SW07  = 000200           948#   958
SW08  = 000400           947#   957
SW09  = 001000           946#   956   6699
SW1   = 000002           964#
SW10  = 002000           945#  2028
SW11  = 004000           944#
SW12  = 010000           943#  6696
SW13  = 020000           942#  6431   6567   6686
SW14  = 040000           941#
SW15  = 100000           940#
SW2   = 000004           963#
SW3   = 000010           962#
SW4   = 000020           961#
SW5   = 000040           960#
SW6   = 000100           959#
SW7   = 000200           958#
SW8   = 000400           957#
SW9   = 001000           956#
TBITVE= 000014           998#
TIMER   001366          1137#  6361*  6364*  6456*  6459*  6484*  6490*  6522*  6525*
TKVEC = 000060          1005#
TPVEC = 000064          1006#
TRAPVE= 000034          1004#  1841*  1842*
TRTVEC= 000014           999#
```

```
TSTEND  020626  6028  6055#
TSTRWS  021436  2750  5128  5207  5236  5256  5585  5752  6360#
TSTSIN  021572  6425#  7224
TST.SI= 104421  2666  2732  2786  2905  3006  3138  3231  3355  3518  3648  3802  3916  4130
                4244  4318  4407  4499  4606  4779  4857  4937  5014  5080  5149  5239  5390
                5458  5523  5744  7224#
TST1    004706  2110  2192#
TST10   005742  2481  2503  2508  2515  2530#
TST11   006030  2554  2565#
TST12   006110  2579  2597#
TST13   006252  2642  2655#
TST14   006416  2705  2721#
TST15   006510  2775#
TST16   006746  2866  2893#
TST17   007174  2970  2995#
TST2    005040  2259#
TST20   007504  3100  3119#
TST21   007702  3219#
TST22   010154  3300  3328#
TST23   010524  3463  3497#
TST24   011004  3614#
TST25   011414  3733  3749  3777#
TST26   011722  3867  3894#
TST27   012242  3998#
TST3    005236  2309#
TST30   012530  4001  4100  4119#
TST31   012764  4215  4233#
TST32   013170  4274  4294  4307#
TST33   013436  4376  4396#
TST34   013720  4470  4488#
TST35   014176  4541  4559  4584  4595#
TST36   014330  4660#
TST37   014654  4758  4768#
TST4    005350  2341  2356#
TST40   015070  4830  4846#
TST41   015274  4901  4926#
TST42   015456  5003#
TST43   015564  5069#
TST44   015740  5138#
TST45   016076  5163  5219#
TST46   016304  5291  5304#
TST47   016456  5378#
TST5    005436  2381  2393#
TST50   016634  5447#
TST51   016762  5489  5512#
TST52   017134  5578#
TST53   017330  5647#
TST54   017622  5659  5717  5742#
TST55   017760  5797#
TST56   020004  5814#
TST57   020054  5845#
TST6    005454  2406  2415#
TST7    005522  2419  2445#
TYERM   021026  2055  2086  6148#
TYPDS = 104405  6085  7202#
TYPE  = 104401  1872  1924  1936  1945  1949  1964  1988  2044  2066  2079  2120  2140  2142
```



```
                2145  2288  2299  2304  6083  6086  6149  6433  6569  6571  6579  6689  6710
                6724  6728  6742  6759  6761  6764  6766  6770  6777  6816  6911  6981  7020
                7021  7024  7037  7048  7067  7120  7126  7131  7135  7140  7141  7143  7146
                7150  7198#  7261
TYPOC = 104402  2071  2125  2150  6154  6439  6577  6584  6727  6750  6774  7023  7199#
TYPON = 104404  7201#
TYPOS = 104403  1930  1942  2293  7200#
T56     020140  5862#  6058
T56FLG  001434  1179#  2096*  5865*  6059*  6707
WATIME  021742  2099  2174  6500#
WATINT  021706  6484#  7222
WAT.IN= 104420  4619  4633  4682  4703  4794  5538  7222#
$AUTOB  001134  1053#  1880*  7017  7166
$BDADR  001122  1048#
$BDDAT  001126  1050#
$CHARC  023304  6818*  6828*  6835  6844*  6849#
$CKSWR  023762  7009#  7206
$CMTAG  001100  1036#  1830  1831  1839  1845  1846
$CM1  = 000012  1068#  1069#  1070#  1071#  1072#  1073#  1074#  1075#  1076#  1077#  1078#
$CM2  = 000024  1068#  1069#  1070#  1071#  1072#  1073#  1074#  1075#  1076#  1077#  1078#
$CM3  = 000012  1066#  1068
$CNTLG  024647  7020  7161#
$CNTLU  024642  7037  7135  7160#
$CRLF   001213  1081#  2067  2141  2305  6689  6742  6761  6766  6770  6817  6852  7048  7140
                7160
$DBLK   023524  6877  6911  6919#
$DDAGN  020750  6079  6088  6094#
$DTBL   023514  6880  6915#
$ENDAD  020740  1025  6090#
$ENDCT  020706  6081#
$ENDMG  020757  6083  6098#
$ENULL  020754  6086  6097#
$EOP    020652  2100  2267  5864  6071#  6713
$EOPCT  020700  6078#  6082
$ERFLG  001103  1039#  6600  6629  6631  6637*  6658  6673*
$ERMAX  001115  1045#  1847*  6631  6653*  6658
$ERROR  022412  1839  6672#
$ERRPC  001116  1046#  6683*  6684*  6685  6748  7814  7816  7819  7822  7824  7827
$ERRTB  001442  1196#  6756
$ERRTY  022734  6688  6741#
$ERTTL  001112  1043#  2264*  5826*  6676*  6680
$ESCAP  001210  1079#  1846*  6652*  6702  6704
$FILLC  001156  1064#  6821  6852
$FILLS  001155  1063#  6852
$GDADR  001120  1047#
$GDDAT  001124  1049#
$GET42  020730  6087#
$GTSWR  024032  7021#  7204
$HD   = 000000  871
$ICNT   001104  1040#  6644*  6645  6647*  6657
$ILLUP  025142  7231  7247  7266#
$INTAG  001135  1054#  7049  7166
$ITEMB  001114  1044#  6685  6745
$LF     001214  1082#  6852  7150  7160
$LPADR  001106  1041#  1848*  2155  6635*  6650*  6655  6657  6698
$LPERR  001110  1042#  1849*  3126*  3329*  3499*  3616*  3779*  3903*  4859*  5224*  5582*  5748*  6635
```

```
                    6651*   6657    6701
$MAIL = ****** U    1866    1876    6650    6805
$MNEW   024665      7024    7164#
$MSWR   024654      7021    7162#
$MXCNT  022410      6648    6657#
$NULL   001154      1062#   6823    6852
$NWTST= 000001      2178#   2180    2253#   2255    2306#   2353#   2390#   2410#   2412    2428#   2430    2525#   2527
                    2562#   2588#   2590    2648#   2650    2711#   2713    2760#   2762    2872#   2874    2979#   2981
                    3105#   3107    3200#   3202    3309#   3311    3470#   3472    3595#   3597    3754#   3756    3886#
                    3888    3990#   3992    4111#   4113    4222#   4224    4301#   4303    4387#   4389    4475#   4477
                    4587#   4589    4651#   4653    4762#   4764    4836#   4838    4918#   4920    4995#   4997    5060#
                    5062    5132#   5134    5212#   5214    5297#   5299    5369#   5371    5435#   5437    5503#   5505
                    5569#   5571    5637#   5639    5732#   5734    5791#   5793    5806#   5808    5829#   5831
$OCNT   023756      6953*   6982*   6995#
$OMODE  023760      6948*   6952*   6957    6960*   6971*   6997#
$OVER   022374      6612    6628    6636    6646    6654#
$PASS   001100      1037#   6075*   6076*   6084    6097    6642    6658
$POWER  025150      7262    7269#
$PWRAD  025136      7264#
$PWRDN  024776      1843    7231#   7259
$PWRMG  025132      7262#
$PWRUP  025050      7241    7247#
$QUES   001212      1080#   6852    7067    7143    7160
$RDCHR  024244      7080#   7207
$RDDEC= ****** U    7209
$RDLIN  024364      7108#   7208
$RDOCT= ****** U    7209
$RDSZ = 000022      7101#
$REGAD  001160      1066#
$REG0   001162      1068#   2211*   2324*   2345*   2377*   2422*   2500*   2505*   2510*   2517*   2556*   2576*   2643*
                    2706*   2857*   2867*   2953*   2964*   2971*   3055*   3066*   3086*   3093*   3185*   3274*   3285*
                    3286*   3301*   3389*   3417*   3448*   3464*   3714*   3715*   3842*   3856*   3953*   3959*   3970*
                    4034*   4080*   4101*   4197*   4216*   4284*   4285*   4346*   4352*   4360*   4367*   4378*   4450*
                    4458*   4464*   4471*   4536*   4547*   4566*   4621*   4689*   4700*   4722*   4732*   4752*   4759*
                    4799*   4815*   4831*   4884*   4910*   5292*   5338*   5479*   5490*   5555*   5616*   5628*   5721*
                    5771*   5961*   5968*   6000*   6019*   6133*   6305*   6331*   6342*   6366*   7814    7816    7819
                    7824    7827
$REG1   001164      1069#   2511*   2518*   2557*   2644*   2707*   2858*   2868*   2954*   2965*   2972*   3056*   3067*
                    3087*   3094*   3101*   3102*   3186*   3275*   3287*   3302*   3390*   3418*   3449*   3465*   3712*
                    3843*   3857*   3952*   3960*   3971*   4035*   4081*   4102*   4198*   4217*   4286*   4353*   4361*
                    4368*   4379*   4451*   4459*   4465*   4472*   4537*   4548*   4562*   4735*   4816*   4832*   4885*
                    4911*   5293*   5339*   5480*   5491*   5556*   5617*   5629*   5719*   5772*   6132*   6306*   6332*
                    6343*   7816    7819    7824    7827
$REG10  001202      1076#   2833*   2929*   3031*   3158*   3248*   3369*   3537*   3668*   3696*   3699*   3818*   4178*
                    4264*   4522*   5692*   5708*   6192    6196    6261*   6264*   6287*   6290*
$REG11  001204      1077#
$REG2   001166      1070#   2966*   3068*   3288*   3391*   3419*   3713*   3844*   3972*   4036*   4103*   4288*   4380*
                    4549*   4563*   4955*   4964*   5029*   5093*   5101*   5171*   5252*   5332*   5417*   5722*   6131*
                    7819    7824    7827
$REG3   001170      1071#   3548*   3682*   3830*   5723*   6130*   6181    6560*   6562*   6565*   7819    7827
$REG4   001172      1072#   7827
$REG5   001174      1073#   7827
$REG6   001176      1074#   7827
$REG7   001200      1075#   6185    7827
$RTNAD  020752      6096#
$R2A  = ****** U    7209
```



```
$SAVRE= ****** U    7209
$SAVR6  025146      7240*   7248    7249*   7250*   7268#
$SCOPE  022140      1837    6609#
$SETUP= 000117      1814#   1836    1837    1839    1841    1843    1845    1846    1848    1872    1873    6073    6610
                    7004    7166
$STUP = 177777      1814#
$SVLAD  022346      6620    6649#
$SVPC = 000204      1023#   1028
$SWR  = 165400       852#    871     877     878     879     880     881     882     883     884    1078    1079    1080
                    1845    1846    1848    1849    2193    2260    2310    2357    2394    2416    2446    2531    2566
                    2598    2656    2722    2776    2894    2996    3120    3220    3329    3498    3615    3778    3895
                    3999    4120    4234    4308    4397    4489    4596    4661    4769    4847    4927    5004    5070
                    5139    5220    5305    5379    5448    5513    5579    5648    5743    5798    5815    5846    6068
                    6074    6085    6095    6097    6601    6602    6603    6604    6605    6611    6623    6625    6626
                    6629    6630    6631    6638    6639    6640    6651    6654    6657    7265
$SWRMK= 000000       884     885    6605    6606    6627
$TIMES  001206      1078#   1845*   2260*   3498*   3615*   3778*   3895*   3999*   4661*   5648*   5743*   5798*   5815*
                    5846*   6074*   6638*   6645    6648*   6657
$TKB    001146      1059#   7002    7013    7030    7084    7090
$TKS    001144      1058#   5698    5699*   5715    5718    7002    7011    7027    7051*   7082    7088
$TN   = 000060       852#    871    2178    2193#   2253    2260#   2306    2310#   2341    2353    2357#   2381    2390
                    2394#   2406    2410    2416#   2419    2428    2446#   2481    2503    2508    2515    2525    2531#
                    2554    2562    2566#   2579    2588    2598#   2642    2648    2656#   2705    2711    2722#   2760
                    2776#   2866    2872    2894#   2970    2979    2996#   3100    3105    3120#   3200    3220#   3300
                    3309    3329#   3463    3470    3498#   3595    3615#   3733    3749    3754    3778#   3867    3886
                    3895#   3990    3999#   4001    4100    4111    4120#   4215    4222    4234#   4274    4294    4301
                    4308#   4376    4387    4397#   4470    4475    4489#   4541    4559    4584    4587    4596#   4651
                    4661#   4758    4762    4769#   4830    4836    4847#   4901    4918    4927#   4995    5004#   5060
                    5070#   5132    5139#   5163    5212    5220#   5291    5297    5305#   5369    5379#   5435    5448#
                    5489    5503    5513#   5569    5579#   5637    5648#   5659    5717    5732    5743#   5791    5798#
                    5806    5815#   5829    5846#
$TPB    001152      1061#   6841*   6852
$TPFLG  001157      1065#   6799    6852
$TPS    001150      1060#   6839    6852
$TRAP   024676      1841    7175#
$TRAP2  024720      7186#   7197
$TRP  = 000022      7190#   7199#   7200#   7201#   7202#   7203#   7204    7205#   7206    7207#   7208#   7209#   7210
                    7211#   7212    7213#   7214    7215#   7216    7217#   7218    7219#   7220    7221#   7222    7223#
                    7224    7225#
$TRPAD  024732      7180    7197#
$TSTNM  001102      1038#   2261*   6073*   6600    6627    6649*   6654    6658    6675
$TTYIN  024620      7110    7111    7123    7141    7155    7159#
$TYPBN= ****** U    7203
$TYPDS  023310      6865#   7202
$TYPE   023070      6799#   7190    7198
$TYPEC  023240      6820    6827    6834    6839#   6840    7053
$TYPEX  023306      6845    6847    6850#
$TYPOC  023560      6951#   7199
$TYPON  023574      6950    6953#   7201
$TYPOS  023534      6946#   7200
$XTSTR  022152      6614#
$$GET4= 000000      6089#
$OFILL  023757      6947*   6951*   6961    6996#
$40CAT= ****** U    6611
.     = 034342      1010#   1014#   1023    1024#   1026#   1028#   1035#   1083    1087#   1824    1834    1848    1849
                    1884#   1927#   1939#   1948#   1952#   1991#   2123#   2148#   2266    2620    3193    3440    3569
```

4055 4617 4879 5887 5889 5896 5994 6097 6101 6152# 6403 6436# 6582#
6657 6658 6691 6697 6780# 6852 6919# 7002 7159# 7160 7166 7243 7267
7810# 7997# 8000#

COMMEN 1# 1008#
ENDCOM 1# 1008#
ERROR 902# 2212 2223 2325 2336 2346 2372 2378 2383 2408 2423 2501 2506 2512 2519
2558 2577 2581 2614 2627 2633 2638 2645 2674 2684 2688 2697 2702 2708 2735
2747 2751 2824 2836 2843 2849 2853 2859 2863 2869 2919 2932 2938 2944 2949
2955 2959 2967 2973 3021 3034 3041 3047 3051 3057 3061 3069 3088 3095 3103
3162 3168 3177 3187 3251 3258 3267 3276 3280 3289 3297 3303 3372 3379 3392
3397 3421 3450 3466 3540 3549 3554 3561 3672 3684 3689 3702 3717 3728 3821
3831 3835 3845 3858 3927 3937 3942 3946 3954 3961 3973 4037 4069 4075 4082
4104 4168 4181 4189 4199 4204 4208 4212 4218 4267 4289 4333 4337 4342 4347
4354 4362 4369 4381 4426 4430 4436 4441 4452 4460 4466 4473 4525 4529 4538
4552 4568 4622 4640 4675 4690 4701 4710 4723 4736 4753 4760 4800 4810 4817
4821 4833 4886 4912 4957 4965 4970 4985 4989 4992 5031 5038 5053 5057 5094
5102 5107 5122 5126 5130 5172 5187 5201 5205 5209 5238 5215 5253 5258 5261
5273 5288 5294 5333 5340 5347 5362 5366 5412 5418 5423 5428 5432 5472 5481
5492 5542 5557 5586 5608 5618 5630 5694 5710 5724 5753 5761 5773 5802 5819
5926 5962 5969 6003 6021 7274 7292
ESCAPE 1# 1008#
GETPRI 1# 1008#
GETSWR 1# 852# 1008# 1873#
MESAGE 2177# 2180 2253# 2255 2410# 2412 2427# 2430 2524# 2527 2588# 2590 2648# 2650 2711#
2713 2760# 2762 2872# 2874 2978# 2981 3105# 3107 3200# 3202 3308# 3311 3470# 3472
3594# 3597 3753# 3756 3885# 3888 3989# 3992 4110# 4113 4221# 4224 4300# 4303 4387#
4389 4475# 4477 4587# 4589 4651# 4653 4762# 4764 4836# 4838 4918# 4920 4995# 4997
5060# 5062 5132# 5134 5212# 5214 5297# 5299 5369# 5371 5435# 5437 5502# 5505 5568#
5571 5637# 5639 5731# 5734 5791# 5793 5806# 5808 5828# 5831
MORETA 1029# 1084
MULT 1# 1008#
NEWTST 1# 1008# 2178 2253 2306 2353 2390 2410 2428 2525 2562 2588 2648 2711 2760
2872 2979 3105 3200 3309 3470 3595 3754 3886 3990 4111 4222 4301 4387 4475
4587 4651 4762 4836 4918 4995 5060 5132 5212 5297 5369 5435 5503 5569 5637
5732 5791 5806 5829
POP 1# 1008# 6906 7252 7253
PRIOR7 1198# 4608 4645 4743 4824 4897 5561 5662 5899
PUSH 1# 1008# 6865 7233 7239
REPORT 1# 1008#
SCOPE 903# 2192 2259 2309 2356 2393 2415 2445 2530 2565 2597 2655 2721 2775 2893
2995 3119 3219 3328 3497 3614 3777 3894 3998 4119 4233 4307 4396 4488 4595
4660 4768 4846 4926 5003 5069 5138 5219 5304 5378 5447 5512 5578 5647 5742
5797 5814 5845 6072
SETPRI 1# 1008#
SETTRA 7190# 7199 7200 7201 7202 7204 7206 7207 7208 7210 7212 7214 7216 7218 7220
7222 7224
SETUP 1# 1008# 1828
SKIP 1# 1008# 2341 2381 2406 2419 2481 2503 2508 2515 2554 2579 2642 2705 2866
2970 3100 3300 3463 3733 3749 3867 4001 4100 4215 4274 4294 4376 4470 4541
4559 4584 4758 4830 4901 5163 5291 5489 5659 5717
SLASH 1# 1008#
SPACE 1008#
STARS 1# 1008# 1021 1031 1083 2178 2191 2253 2258 2306 2308 2353 2355 2390 2392
2410 2414 2428 2444 2525 2529 2562 2564 2588 2596 2648 2654 2711 2720 2760
2774 2872 2892 2979 2994 3105 3118 3200 3218 3309 3327 3470 3496 3595 3613
3754 3776 3886 3893 3990 3997 4111 4118 4222 4232 4301 4306 4387 4395 4475
4487 4587 4594 4651 4659 4762 4767 4836 4845 4918 4925 4995 5002 5060 5068
5132 5137 5212 5218 5297 5303 5369 5377 5435 5446 5503 5511 5569 5577 5637
5646 5732 5741 5791 5796 5806 5813 5829 5844 6064 6597 6736 6784 6855 6923

```
        7001    7004    7072    7101    7169    7229    7245
$ARSU      1#   1008#   1850#
TRMTRP  7190#
TYPBIN     1#   1008#
TYPDEC     1#   1008#   6084
TYPNAM     1#    852#   1008#   1868
TYPNUM     1#   1008#
TYPOCS     1#   1008#
TYPOCT     1#   1008#   6748    6772    7022
TYPTXT     1#   1008#   1923    1936    1945    1949    1964    1987    2120    2145    2299    6148    6433    6571    6579
$$CMRE  1029#   1068    1069    1070    1071    1072    1073    1074    1075    1076    1077
$$CMTM  1029#
$$ESCA     1#   1008#
$$NEWT     1#   1008#   2178    2253    2306    2353    2390    2410    2428    2525    2562    2588    2648    2711    2760
        2872    2979    3105    3200    3309    3470    3595    3754    3886    3990    4111    4222    4301    4387    4475
        4587    4651    4762    4836    4918    4995    5060    5132    5212    5297    5369    5435    5503    5569    5637
        5732    5791    5806    5829
$$SET   7190#   7199    7200    7201    7202    7204    7206    7207    7208    7210    7212    7214    7216    7218    7220
        7222    7224
$$SKIP     1#   1008#   2341    2381    2406    2419    2481    2503    2508    2515    2554    2579    2642    2705    2866
        2970    3100    3300    3463    3733    3749    3867    4001    4100    4215    4274    4294    4376    4470    4541
        4559    4584    4758    4830    4901    5163    5291    5489    5659    5717
.EQUAT     1#    852#    898
.HEADE     1#    852#    861
.KT11      1#
.SETUP     1#    852#   1814
.SWRHI     1#    852#    873
.SWRLO  852#    885#
.$ACT1     1#    852#   1019
.$APTB     1#
.$APTH     1#
.$APTY     1#
.$ASTA     1#
.$CATC     1#    852#   1008
.$CMTA     1#    852#   1029
.$DB2D     1#
.$DB2O     1#
.$DIV      1#
.$EOP      1#    852#   6062
.$ERRD     1#
.$ERRT     1#    852#   6734
.$MULT     1#
.$POWE     1#    852#   7227
.$RAND     1#
.$RDDE     1#    852#
.$RDOC     1#
.$READ     1#    852#   6999
.$R2AZ     1#
.$SAVE     1#
.$SB2D     1#
.$SB2O     1#
.$SCOP     1#    852#   6595
.$SIZE     1#
.$SUPR     1#
.$TRAP     1#    852#   7167
.$TYPB     1#
```



```
.$TYPD     1#    852#   6853
.$TYPE     1#    852#   6782
.$TYPO     1#    852#   6921
.$40CA     1#
.1170      1#

. ABS.  034342     000

 ERRORS DETECTED:  0

 CZRKKF,CZRKKF.LST/CRF/SOL=CZRKKF.SML,CZRKKF.P11
 RUN-TIME: 22 31 1 SECONDS
 RUN-TIME RATIO: 392/55=7.0
 CORE USED:  34K  (67 PAGES)
```