RK11/RK05-F-J

DYNAMIC TEST

MD-11-DZRKL-C

EP DZRKL-C-DL-A

OCT 1976

FICHE 1 OF 1

digital

Made in U.S.A.

DZRKLC.P11
.REM %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZRKL-C-D
PRODUCT NAME:	RK11/RK05 DYNAMIC TEST
DATE CREATED:	NOVEMBER 1975
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	JIM KAPADIA
REVISED BY:	PERVEZ ZAKI TOM SAWYER

TABLE OF CONTENTS

1.0 ABSTRACT

THE RK11/RK05 DYNAMIC TEST AIMS AT
1. DEMONSTRATING THE ELECTROMECHANICAL INTREGRITY OF THE DRIVE.
2. CHECKING THE LINEAR POSITIONER CONTROL AND SPEED CONTROL
3. VERIFYING THE INTREGITY OF THE READ/WRITE LOGIC
4. PROVIDING A TIMER FOR THE SEEK FUNCTION.

THIS IS A TEST ONE LEVEL HIGHER THAN THE BASIC RK11 LOGIC TESTS.

2.0 REQUIREMENTS

2.1 EQUIPMENT

A. PDP11 WITH CONSOLE TELETYPE.
B. 8K OF MEMORY
C. RK11 CONTROLLER
D. RK05 DRIVE/S F AND J TYPE DRIVES MAY BE MIXED
E. DISK PLATTER - NEED NOT BE FORMATTED.

2.2 PRELIMINARY PROGRAMS

RK11 LOGIC TESTS-I (MAINDEC-11-DZRKJ)
RK11 LOGIC TESTS-II (MAINDEC-11-DZRKK)

2.3 EXECUTION TIME

ERROR FREE FIRST PASS ON PDP11/20 WITH CORE MEMORY TAKES APPROXIMATELY 5 MINUTES (WITHOUT THE SEEK TIMER AND GRAPH, ADDITIONAL 3.5 MINUTES FOR THESE). LESS FOR FASTER MACHINES OR MEMORIES.

3.0 STARTING ADDRESS

200 FOR ANY NORMAL MODE OF OPERATION. ALL SWITCHESS DOWN

210 FOR FUNCTION SELECTING PROGRAM (CONVERSATIONAL MODE).

4.0 PROGRAM CONTROL MODES & OPERATOR ACTION

PAPER TAPE LOADING
RKDP DUMP MODE
RKDP CHAIN MODE

156 ACT11

4.1 PAPER TAPE LOADING

4.1.1 LOAD PROGRAM INTO MEMORY USING STANDARD PROCEDURE FOR ABSOLUTE 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 6.0)

PRESS START.

4.1.5 THE PROGRAM IDENTIFIES ITSELF

MAINDEC-11-DZRKL-C

THEN IT PROCEEDS TO FIND WHICH DRIVES ARE PRESENT AND PRINTS OUT THE DRIVES FOUND. IF AN RK-05F IS DETECTED, AN F IS APPENDED TO THE DRIVE NUMBER:

DRIVES PRESENT
0
1

AFTER TYPING OUT THE DRIVE NUMBER THAT IS GOING TO BE TESTED, EXECUTION OF THE TESTS START.

AFTER ALL THE TESTS HAVE BEEN EXECUTED ON ONE DRIVE THEY ARE EXECUTED ON THE NEXT DRIVE, IF PRESENT. THIS IS REPEATED TILL ALL DRIVES ARE TESTED.

AT THE END OF A PASS THE FOLLOWING IS TYPED OUT:

END PASS X X=0,1,2.....

CONTROL IS TRANSFERRED BACK TO THE BEGINNING OF THE PROGRAM AND RE-EXECUTION BEGINS.

4.2 RKDP DUMP MODE

4.2.1 THE PROGRAM IS LOADED BY THE RKDP MONITOR.

4.2.2 SET SA=200. SELECT ANY SWITCHES YOU WANT AND PRESS START.

4.2.3 THE PROGRAM IDENTIFIES ITSELF AND PRINTS OUT:

213

```
                                REPLACE DRO RKDP-PAK BY OTHER, TYPE CR WHEN DONE
                                IF NOT PUT DRO ON LOAD
                                IN RESPONSE TO THIS, REPLACE THE RKDP PAK  ON  DRIVE
                                0.  IF  YOU  WANT  DRIVE  0  TESTED.   THEN  TYPE IN
                                CARRIAGE RETURN.  IF YOU DO NOT WANT TO TEST DRIVE 0
                                (OR REPLACE RKDP PAK), PUT DRIVE 0 ON 'LOAD', 'WRITE
                                PROTECT' AND THEN TYPE IN CARRIAGE RETURN.


                        4.3     RKDP CHAIN MODE

                                THE PROGRAM IS CHAIN LOADED FROM RKDP PAK  ON  DRIVE
                                0.   AFTER IDENTIFYING ITSELF, THE FOLLOWING MESSAGE
                                APPEARS:

                                DRO NOT TESTED

                                DRIVE 0 WILL NOT BE TESTED SINCE THE RKDP PAK IS  ON
                                THAT DRIVE.


                        4.4     ACT11 MODE

                                THE PROGRAM IS LOADED BY THE ACT11  MONITOR.   AFTER
                                IDENTIFYING  ITSELF, ASCERTAINS THE NUMBER OF DRIVES
                                PRESENT AND PROCEEDS TO TEST EACH OF THEM AS BEFORE.


                        5.0     DRIVE SELECTION

                                IF ANY  PARTICULAR  DRIVE  IS  TO  BE  SELECTED  FOR
                                TESTING,  PUT  THAT  DRIVE ON 'RUN', 'WRITE ENABLE'.
                                PUT THE REST OF THE DRIVES ON 'LOAD', 'WRITE  LOCK'.
                                THEN START AS USUAL.
```

```
                                6.0     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        DUMP ALL RK11 REGISTERS ON ERROR
                                SW<10>=1        RING BELL ON ERROR
                                SW<09>=1        LOOP ON SPECIFIC ERROR
                                SW<08>=1        LOOP ON TEST INDICATED BY USER  (SEE
                                                SEC.  6.8)
                                SW<06>=1        TYPE SEEK TIMER
                                SW<05>=1        TYPE THE GRAPHS
                                SW<04>=1        PRINT THE COMPLETE GRAPH
```

```
                        SW<03>=1        TERMINATE FUNCTION SELECTED BY USER
                        SW<02>=1        DROP THE DRIVE AFTER MAXIMUM
                                        ALLOWABLE NUMBER OF ERRORS OCCUR
                        SW<00>=1        ASK FOR PATTERN TO BE WRITTEN OR
                                        WRITE CHECKED (FUNCTION SELECTION
                                        PROGRAM)

                6.1     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.


                6.2     SW<14>

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


                6.3     SW<13>

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


                6.4     SW<12>

                        THIS SWITCH ALLOWS THE PROGRAM TO CYCLE FROM THE
                        POINT OF ERROR TO THE PREVIOUS SCOPE STATEMENT.
                        NOTE THAT IN DOING SO ANY INITIALIZATION BEING DONE
                        AT THE BEGINNING OF THE SUBTEST WILL BE DONE AGAIN
                        AND AGAIN. SEE SEC. 6.7 FOR A DIFFERENT KIND OF
                        SCOPE LOOP.


                6.5     SW<11>

                        THIS SWITCH ALLOWS DUMPING OF ALL RK11 REGISTERS ON
                        ENCOUNTERING AN ERROR.


                6.6     SW<10>

                        RINGS A BELL ON ERROR, USEFUL WHEN ERROR TYPEOUT IS
                        INHIBITED.

                6.7     SW<09>
```

THIS SWITCH PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP. NOTE THAT UNLIKE 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.)

6.8 SW<08>

THIS SWITCH IS USED TO SELECT A PARTICULAR TEST FOR EXECUTION. WHEN THE PROGRAM IS STARTED (200) WITH THIS SWITCH SET, THE FOLLOWING MESSAGE APEARS:

OCTAL TEST#?

THE USER SHOULD REPLY WITH THE TEST NUMBER (OCTAL) HE WANTS TO SELECT, FOLLOWED BY CARRIAGE RETURN.

THE SELECTED TEST IS EXECUTED AGAIN AND AGAIN. TO GET OUT OF THIS LOOP, PUT SW 8 BACK TO 0. THIS WILL RESUME NORMAL OPERATION OF THE PROGRAM. NOTE THAT BEFORE TEST 4 CAN BE EXECUTED TEST 2 SHOULD HAVE BEEN DONE AND TEST 6 SHOULD HAVE BEEN DONE BEFORE TEST 7.

6.9 SW<06>

THIS SWITCH WHEN SET MAKES THE PROGRAM TYPE THE SEEK TIMER. THIS SWITCH CAN BE SET OR RESET BEFORE OR DURING THE SEEK TIMER EXECUTION, AND EVEN WHILE THE TYPEOUT IS OCCURING.

6.10 SW<05>

THIS SWITCH MAKES THE PROGRAM TYPE THE GRAPHS. IF RESET BEFORE THE GRAPH-PLOTTING ROUTINE IS ENTERED, THE GRAPHS WILL BE SKIPPED ENTIRELY. IT CAN BE RESET EVEN AFTER SOME OF THE POINTS HAVE BEEN PLOTTED, TO SKIP PLOTTING REST OF THE POINTS.

6.11 SW<04>

THIS SWITCH IS USED TO SELECT THE COMPLETE GRAPH OUTPUT (SEEK TIMES OF ALL CYLINDERS ARE PLOTTED) NORMALLY WHEN THIS SWITCH IS NOT SET, THE SMALL GRAPH (ONLY SELECTED CYLINDERS PLOTTED) IS PRINTED OUT.

6.12 SW<03>

THIS SWITCH WHEN SET TERMINATES THE EXECUTION OF THE FUNCTION SELECTED BY THE USER (SA=210). A NEW FUNCTION MAY BE INITIATED NOW. IF YOU WANT TO KEEP ON LOOPING ON THE SAME FUNCTION, PUT SW 3 DOWN. SEE SEC. 9.0.

6.13 SW<02>

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 6, AFTER 6 ERRORS HAVE OCCURED THE DRIVE IS DROPPED AND A MESSAGE (DRIVE # XXXXX DROPPED) IS PRINTED.

6.14 SW<00>

THIS SWITCH IS TO BE USED WITH THE FUNCTION SELECTION PROGRAM (SA=210). IF A WRITE OR A WRITE CHECK FUNCTION IS SELECTED WITH THIS SW SET, THE PROGRAM WILL ASK FOR THE PATTERN TO BE WRITTEN OR WRITE CHECKED (PATRN?). THE USER SHOULD TYPE IN THE (OCTAL) PATTERN. THIS PATTERN WILL BE WRITTEN (OR WRITE CHECKED) ON THE DISK. FOR FURTHER INFORMATION REFER TO SEC. 9.0.

7.0 PROGRAM DESCRIPTION

THE FIRST TEST IS AIMED AT DETECTING IMPEPENDING ELECTRO- MECHANICAL FAILURES IN THE DRIVE AND INNER/OUTER LIMIT SWITCHES.

IN THE NEXT TWO TESTS, THE DISK IS FORMATTED AND CHECKED FOR CORRECT FORMATTING. IF THE DISK IS AN RK-05F, THE ENTIRE DISK IS FORMATTED EACH TIME THE EVEN DRIVE IS TESTED. NO FORMATTING IS DONE WHEN THE ODD DRIVE IS TESTED. THE DISK IS CHECKED EACH TIME FOR PROPER FORMAT, HOWEVER.

IN NEXT TWO TESTS THE SEEK LOGIC, POSITIONER, ETC ARE CHECKED OUT BY DOING IMPLIED SEEK, USING TWO DIFFERENT SEEKING PATTERNS. THE FIRST ONE IS A DECREASING SAW-TOOTH PATTERN (0-312-0-311-0-310....), THE SECOND ONE IS A

CONVERGING-DIVERGING PATTERN (0-312-1-311-2-310....). ON GETTING AN ERROR, FURTHER ANALYSIS IS DONE TO FIND OUT MORE ABOUT THE NATURE OF ERROR. MANY TIMES ADDITIONAL INFORMATION IS GIVEN FOR THE CONVIENCE OF THE USER. RETRIES ARE DONE WHENEVER AN ERROR OCCURS.

IN THE SUBSEQUENT TESTS EXTENSIVE WRITING IS DONE USING MORE THAN 2000 DIFFERENT PATTERNS. THE DATA IS READ, (SOFTWARE) COMPARED, AND WRITE CHECKED.

EVERYTIME AN ERROR OCCURS RETRIES ARE DONE, TO CHECK IF IT WAS A RECOVERABLE ERROR OR NOT. THE USER CAN CHANGE THE PATTERNS TO BE WRITTEN ON THE DISK. THE DATA TRANSFER BUFFERS CAN BE RE-LOCATED BY THE USER TO DIFFERENT PARTS OF MEMORY. REFER TO LOCATIONS 'PBUF0', 'PBUF1', 'PAT1', 'PTRN01' IN THE LISTINGS FOR MORE DETAILS. SEE SEC 7.1.

THE SHUNT CURRENT CHANGE TEST WRITES, READS AND CHECKS FOR ERRORS ON CYLINDERS 127 AND 128. THIS REGION HAS CRITICAL "PACKING DENSITY" TO "WRITE CURRENT" RATIOS.

THE SEEK TIMER PROVIDES SEEK TIMES AND GRAPHS AS EXPLAINED IN SEC 8.0

A FUNCTION SELECTION SUB-PROGRAM IS PROVIDED FOR USER SELECTION OF FUNCTIONS. SEE SEC 9.0

EVERY TEST IN THE PROGRAM IS PRECEEDED BY AN EXPLANATION OF THAT TEST. THE USER IS ADVISED TO REFER TO THAT, IF MORE INFORMATION IS NEEDED.

7.1 PERMISSIBLE USER PROGRAM MODIFICATIONS

THE USER CAN MAKE MINOR CHANGES IN POINTERS, TABLES, ETC. TO TAKE CARE OF HIS SPECIAL NEEDS. IT IS ADVISABLE TO MAKE CHANGES IF ANY, RIGHT AT THE BEGINING.

7.1.1 SEEK TIMING CAN BE DONE BETWEEN ANY TWO CYLINDERS, BY MAKING CHANGES DESCRIBED IN THE CYLINDER ADDRESS TABLE AT LOCATIONS. 'SOAD' AND 'SIAD' IN THE LISTINGS.

7.1.2 IN CASE YOU HAVE A LINE PRINTER AND WANT YOUR OUTPUT ON THE LINE PRINTER, CHANGE LOCATION '$TPS' TO 177514 AND LOCATION '$TPB' TO 177516 (LINE PRINTER VECTORS).

7.1.3 INPUT/OUTPUT DATA BUFFERS (FROM WHERE DATA TRANSFERS WILL BE DONE TO AND FROM THE DISK) CAN BE RELOCATED TO ANYWHERE IN THE 28K OF MEMORY (DO NOT OVERLAY THE PROGRAM). THIS CAN BE DONE BY CHANGING THE CONTENTS OF LOCATIONS 'PBUF0' AND 'PBUF1' TO THE STARTING ADDRESSES OF THE TWO USER SELECTED BUFFERS. IT SHOULD BE NOTED THAT EACH OF THE TWO BUFFERS SHOULD BE 768 (DECIMAL) WORD LONG.

7.1.4 FOUR DIFFERENT PATTERN GENERATOR ROUTINES HAVE BEEN USED IN THIS PROGRAM: A. PTGEN0 B. PTGEN1 C. PTGEN2 D. PTGEN3. THEY HAVE BEEN DESCRIBED IN DETAIL AT CORRESPONDING LOCATIONS IN THE LISTING. THE ORDER IN WHICH THEY ARE CALLED IS DESCRIBED AT THE BEGINING OF TEST 6. THIS CALLING ORDER CAN BE CHANGED BY MAKING CHANGES IN THE FOUR POINTERS A.'PAT0' B. 'PAT1' C. 'PAT2' D. 'PAT3'. THESE 4 POINTERS CONTAIN THE STARTING ADDRESS OF EACH ROUTINE.

7.1.5 AS A SPECIAL CASE OF THE ABOVE, YOU CAN WRITE THE SAME TWO (OR ONE) PATTERN/S ON THE ENTIRE DISK USING 'PTGEN0' ROUTINE. TO WRITE THE SAME ONE PATTERN:
CHANGE LOCATION 'PAT1' TO 'PTGEN0' (STARTING ADDRESS OF PTGEN0)
CHANGE LOCATION 'PAT2' TO 'PTGEN0' (STARTING ADDRESS OF PTGEN0)
CHANGE LOCATION 'PAT3' TO 'PTGEN0' (STARTING ADDRESS OF PTGEN0)
FILL LOCATIONS 'PTRN01' AND 'PTRN02' WITH THE PATTERN YOU WANT.
TO WRITE 2 DIFFERENT PATTERNS (IN ALTERNATING SECTORS):
CHANGE 'PAT1', 'PAT2' AND 'PAT3' AS DESCRIBED ABOVE.
FILL 'PATRN01' AND 'PATRN02' WITH THE TWO PATTERNS YOU WANT.

7.1.6 IN TEST 10, IF YOU WANT TO WRITE AND CHECK CYLINDERS 127 AND 128 WITH PATTERNS OTHER THAN THE 12 USED, CHANGE ANY OR ALL OF THE 12 POINTERS 'SP1' THROUGH 'SP12' TO CONTAIN PATTERNS YOU WANT.

8.0 SEEK TIMER & GRAPHS

THE LAST TEST IN THIS PROGRAM IS THE SEEK TIMER. IN ORDER TO TIME THE SEEKS, THE SECTOR COUNTER HAS BEEN USED AS A TIME BASE. THUS THE ACCURACY OF THE TIMES RECORDED IS AS GOOD AS THE ACCURACY OF THE SECTOR COUNTER (WHICH IN TURN DEPENDS ON THE ROTATION SPEED OF THE DISK).

IN THE FIRST PART OF THIS TIMER, SOME CRITICAL SEEKS

HAVE BEEN TIMED (CYLINDERS 0-1, 179-181, 0-3, 0-16, 0-32, 0-202, 0=100) EACH SEEK IS DONE 100 TIMES, TIMES ARE RECORDED, THEN THE TIMES ARE SORTED OUT AND A PRINTOUT IS GIVEN SHOWING HOW MANY TIMES A PARTICULAR SEEK TIME WAS OBTAINED. EXAMPLE: SEEK BETWEEN 0 AND LAST CYLINDER WAS DONE 100 TIMES. 99 TIMES A SEEK TIME OF 95 MS WAS OBTAINED, ONCE IT GAVE 100 MS. THIS GIVES THE USER AN IDEA OF HOW CONSISTENT ARE THE SEEK TIMES.

IF YOU WANT TO TIME SEEK BETWEEN ANY OTHER SET OF CYLINDERS, YOU CAN DO BY FOLLOWING THE INSTRUCTIONS AT LOCATION 'SOAD' IN LISTINGS. SEE SEC 7.1

IN THE SECOND PART, A GRAPH OF THE 'CYLINDER SEEKED FROM 0' IS PLOTTED AGAINST 'SEEK TIME'. TWO GRAPHS ARE AVAILABLE, NORMALLY THE SMALL GRAPH IS PRINTED OUT. THE SMALL GRAPH PLOTS THE SEEK TIMES FOR SELECTED CYLINDERS (ABOUT 49) COVERING THE RANGE FROM CYLINDER 0 TO 202. IT GIVES THE USER A QUICK SEEK CHARACTERISTICS OF A DRIVE.

THE OPTIONAL COMPLETE GRAPH (SW 4) GIVES A GRAPH SIMILAR TO THE ABOVE ONE, BUT PLOTS ALL THE CYLINDERS (203).

THE GRAPH SHOWN ON LAST PAGE IS A SAMPLE OUTPUT. IT SHOULD BE REALIZED THAT DIFFERENT DRIVES MAY HAVE A SLIGHTLY DIFFERENT CHARACTERISTIC.

9.0 FUNCTION SELECTION PROGRAM

THIS PROGRAM GIVES THE USER A CAPABILITY TO SELECT A FUNCTION AND EXECUTE IT, FROM THE CONSOLE TELETYPE.

STARTING ADDRESS=210

ON STARTING THE PROGRAM AT 210, THE FOLLOWING QUESTION APPEARS:

FUNCTION?

THE REPLY SHOULD BE:

WR	FOR WRITE
WC	FOR WRITE CHECK
RD	FOR READ
RC	FOR READ CHECK
CR	FOR CONTROL RESET
DR	FOR DRIVE RESET
SK	FOR SEEK DR

ALL COMMANDS SHOULD BE TERMINATED BY A CARRIAGE RETURN. DEPENDING ON WHICH FUNCTION IS GIVEN THE

FOLLOWING QUESTIONS APPEAR:

RKBA? TYPE IN THE BUS ADDRESS (OCTAL) FOLLOWED BY A C.R.

RKDA? TYPE IN THE DISK ADDRES (OCTAL) FOLLOWED BY C.R.

IF A NON-EXISTENT CYLINDER OR SECTOR IS SELECTED, THE QUESTION IS REPEATED AGAIN.

#WORDS? TYPE IN THE NUMBER OF WORDS YOU WANT TO TRANSFER. IT SHOULD BE IN OCTAL. THUS IF YOU WANT TO READ A SECTOR TYPE IN 400 FOLLOWED BY C.R. ANY NUMBER OF WORDS CAN BE TRANSFERRED DEPENDING ON THE BUFFER SIZE AVAILABLE.

FOR A WRITE FUNCTION: IF SW0 IS SET TO 1 THE PROGRAM WILL ASK FOR THE DATA PATTERN TO BE WRITTEN:
PATRN? THE USER SHOULD TYPE IN THE DATA PATTERN (OCTAL) TO BE WRITTEN,FOLLOWED BY <CR>. THE PATTERN WILL BE WRITTEN ON THE DISK. NOTE THE NUMBER OF WORDS TO BE WRITTEN AND THE DISK ADDRESS SHOULD BE SPECIFIED.

FOR A WRITE CHECK FUNCTION: IF SW 0 IS SET TO 1, THE USER IS ASKED FOR THE PATTERN TO BE WRITE CHECKED: PATRN? THE USER SHOULD TYPE IN THE (OCTAL) PATTERN.

FOR A SEEK FUNCTION: CYL1? CYL2? IN REPLY TO THESE, TYPE IN THE CYLINDER NUMBERS (OCTAL) BETWEEN WHICH THE SEEK IS TO BE DONE. IF A NON EXISTENT CYLINDER IS TYPED IN THE QUESTION IS REPEATED AGAIN.

THE FUNCTION IS EXECUTED AGAIN AND AGAIN. TO GET OUT OF THIS LOOP SW 3 SHOULD BE SET, AT THIS POINT THE QUESTION (FUNCTION?) IS ASKED AGAIN.

IF UPON EXECUTION OF A FUNCTION AN ERROR OCCURS IT IS REPORTED. ALL SWITCH OPTIONS WHICH APPLY TO ANY OTHER ERROR, ALSO APPLY TO THIS ERROR.

IF ON INPUTTING A NUMBER OR COMMAND A MISTAKE IS MADE, THE INPUT STRING CAN BE DELETED BY HITTING 'RUBOUT' KEY, THE NEW STRING CAN BE TYPED IN AGAIN.

10.0 ERROR INFORMATION

WHENEVER AN ERROR MESSAGE IS PRINTED OUT, ALL REGISTERS AND OTHER DATA PERTAININNG 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 ADVISED TO 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. 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.

AT TIMES WHEN AN ERROR OCCURS BESIDES THE ERROR PRINTOUT MORE PRINTOUTS OCCUR. THEY ARE GIVEN TO HELP THE USER UNDERSTAND THE PROBLEM.

11.0 UNEXPECTED TIMEOUTS AND RK11 INTERRUPTS

WHEN AN UNEXPECTED TIMEOUT OCCURS, THE PC AT WHICH TIME OUT OCCURRED 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 BEGINNING.

12.0 COMMONLY USED SUBROUTINES

A BRIEF EXPLAINATION OF EVERY SUBROUTINE IS GIVEN IN THE LISTINGS (JUST BEFORE THE CODE FOR THAT SUBROUTINE). ALL SUB-ROUTINES ARE LISTED IN THE 'TABLE OF CONTENTS' FOUND AT THE BEGINNING OF LISTINGS. THESE ARE TWO WAYS IN WHICH ROUTINES ARE CALLED, 1. JSR PC,ROUTINE 2. THROUGH AN ENCODED TRAP INSTRUCTION. THE LOWER BYTE OF THE 'TRAP' INSTRUCTION IS USED TO INDEX THROUGH THE TRAP TABLE ($TRPAD) FOR THE STARTING ADDRESS OF THE DESIRED ROUTINE.

13.0 SAMPLE GRAPH AND SEEK TIMER OUTPUTS

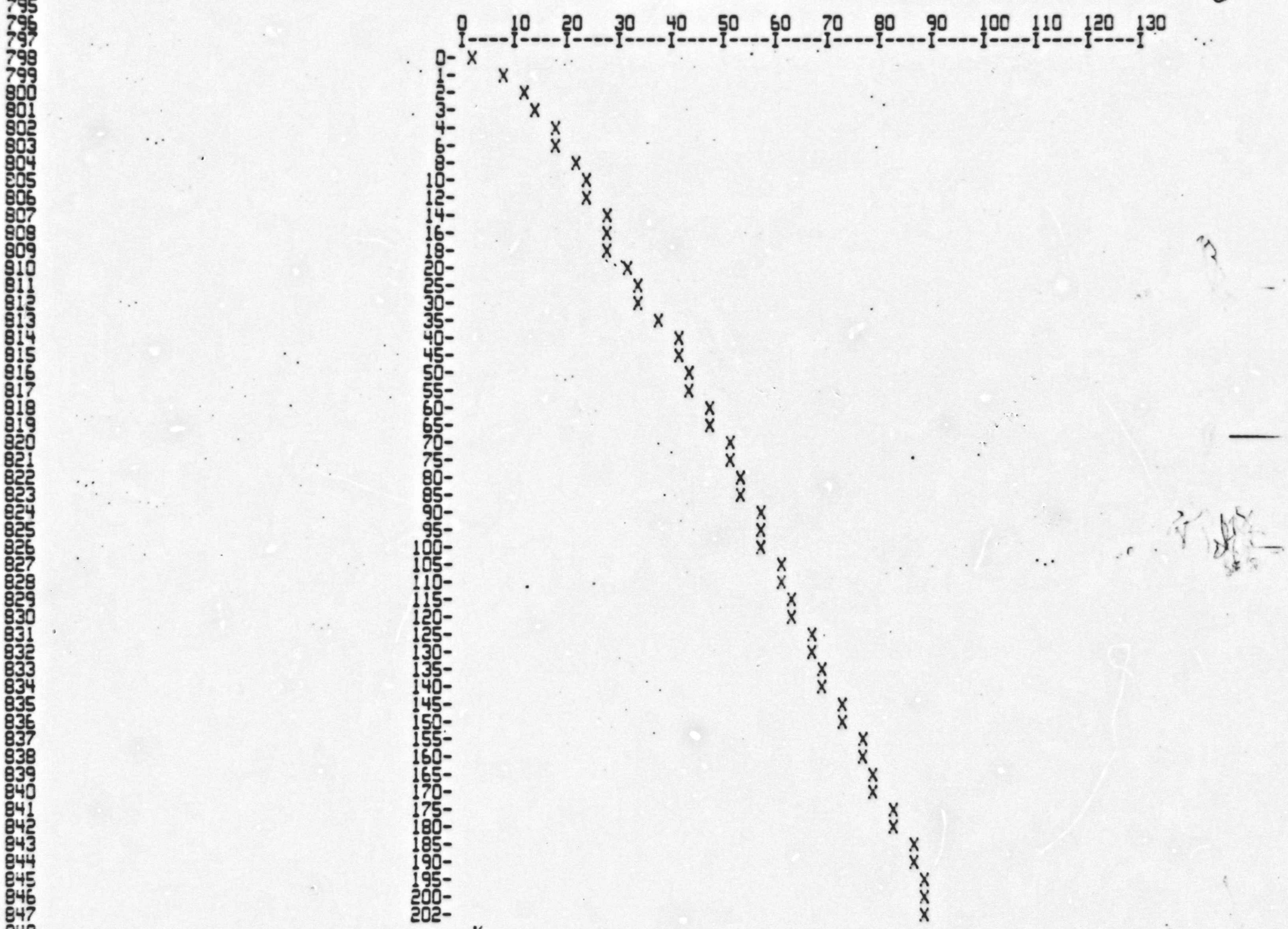
'# OF SEEKS' INDICATES THE NUMBER OF TIMES A PARTICULAR
'SEEK TIME' WAS OBTAINED. NOTE THAT TIMES ARE RECORDED FOR
BOTH FORWARD AND REVERSE SEEKS, BETWEEN A SET OF CYLINDERS.

SEEK TIME SCALE FACTOR=0.01 MILI SECS

# OF SEEKS	SEEK TIME	# OF SEEKS	SEEK TIME
CYLS:0-202			
FRWRD		REVRSE	
100	9075	100	9075
CYLS:0-1			
FRWRD		REVRSE	
100	825	100	1155
CYLS:179-181			
FRWRD		REVRSE	
100	1155	100	1155
CYLS:0-3			
FRWRD		REVRSE	
100	1485	100	1485
CYLS:0-16			
FRWRD		REVRSE	
100	3135	100	3135
CYLS:0-32			
FRWRD		REVRSE	
100	3795	100	3795
CYLS:0-100			
FRWRD		REVRSE	
100	5775	100	5775

X AXIS - SEEK TIME - MILI SECS
Y AXIS - CYLINDER SEEKED FROM 0

SAMPLE OUTPUT



```
                        .TITLE  MAINDEC-11-DZRKL-C
                        ;*COPYRIGHT (C) 1974,1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY JIM KAPADIA
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-CO),MAR 21, 1976.
                        ;*
                        ;*JANUARY 1975
                        ;*
                        ;*REVISED MARCH 1976 BY TOM SAWYER
                        ;*


                        .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'
                        ;*        10            BELL ON ERROR
                        ;*         9            LOOP ON ERROR
                        ;*         8            SELECT TEST TYPED IN BY USER
                        ;*         6            SKIP SEEK TIMER
                        ;*         5            SKIP THE GRAPHS
                        ;*         4            TYPE THE COMPLETE GRAPH (ALL SEEK TIMES)
                        ;*                      NOTE, OTHERWISE YOU GET SMALL GRAPH
                        ;*         3            TERMINATE FUNCTION SELECTED BY USER
                        ;*                      (FOR FUNCTION SELECTION PROGRAM SA=210)
                        ;*         2            DROP THE DRIVE AFTER MAXIMUM ALLOWABLE
                        ;*                      NUMBER OF ERRORS HAVE OCCURED
                        ;*         0            ASK FOR PATTERN TO BE WRITTEN (OR WRITE
                        ;*                      CHECKED), IN FUNCTION SELECTION PROGRAM
                        ;*        11            DUMP OUT ALL RK REGISTERS ON ERROR


                        ;*      YOU ARE ADVISED TO READ THE DOCUMENT FOR THIS PROGRAM.
                        ;*      FUNCTION SELECTION PROGRAM STARTS AT 210.
```

```
                                .SBTTL  ACT11 HOOKS

                                ;;**********************************************************************
                                ;HOOKS REQUIRED BY ACT11
000000  000046                          $SVPC=.                 ;SAVE PC
000046  015122                          .=46
                                        $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
000052  000000                          .=52
                                        .WORD   0               ;;2)SET LOC.52 TO ZERO
000000                                  .=$SVPC                 ;; RESTORE PC
                                .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
                                .EQUIV  R6,SP           ;;STACK POINTER
                                .EQUIV  R7,PC           ;;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
                        000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
                        000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR

                                                .SBTTL  TRAP CATCHER

                        000000                          .=0
                                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
                        000174                          .=174
000174  000000                                 DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                                 SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                                .SBTTL  STARTING ADDRESS(ES)
000200  000137  002460                                  JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM

                        000210                          .=210
000210  105237  001517                                  INCB    FFUNC           ;SET FLAG INDICATING SELECTION OF
000214  000137  002460                                  JMP     @#START         ;FUNCTION PROGRAM.
```

```
                                .SBTTL  COMMON TAGS

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

                        001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000          $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
001102      000         $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103      000         $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114      000         $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115      001         $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134      000         $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135      000         $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154      000         $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155      002         $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156      012         $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157      000         $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001162  000000          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001164  000000          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001166  000000          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001170  000000          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001172  000000          $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001174  000000          $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001176  000000          $REG6:  .WORD   0               ;;CONTAINS (($REGAD)+14)
001200  000000          $REG7:  .WORD   0               ;;CONTAINS (($REGAD)+16)
001202  000000          $REG10: .WORD   0               ;;CONTAINS (($REGAD)+20)
001204  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001206  177607  000377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001212      077         $QUES:  .ASCII  /?/             ;;QUESTION MARK
001213      015         $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001214  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;**************************************************************
```

```
                                .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

001216                          $ERRTB:

                                ;IN CASE YOU WANT THE OUTPUT TO COME OUT ON LINE PRINTER, (IF YOU HAVE
                                ;ONE), MAKE THE FOLLOWING CHANGES ABOVE:

                                ;CHANGE CONTENTS OF '$TPS' TO 177514  (LPT VECTOR)
                                ;CHANGE CONTENTS OF '$TPB' TO 177516    ( " ")
                                        ;ERROR ITEMS TABLE
                                        ;
                                        ;
                                        ;
                                ;ITEM   1

001216  023506                          EM1     ;CNTRL RDY DIDN'T SET AFTER SEEK
001220  024652                          DH1     ;PC     RKCS    RKER    RKDS    RKDA
001222  025506                          DT1     ;$ERRPC $REG0   $REG1   $REG2   $REG3
001224  000000                          0

                                ;ITEM   2

001226  023545                          EM2     ;SIN ON SEEK
001230  024652                          DH1     ;PC     RKCS    RKER    RKDS    RKDA
001232  025506                          DT1     ;$ERRPC $REG0   $REG1   $REG2   $REG3
001234  000000                          0

                                ;ITEM   3

001236  023561                          EM3     ;DRE ON SEEK
001240  024652                          DH1     ;PC     RKCS    RKER    RKDS    RKDA
001242  025506                          DT1     ;$ERRPC $REG0   $REG1   $REG2   $REG3
001244  000000                          0

                                ;ITEM   4

001246  023575                          EM4     ;'ERR' ON SEEK
001250  024652                          DH1     ;PC     RKCS    RKER    RKDS    RKDA
001252  025506                          DT1     ;$ERRPC $REG0   $REG1   $REG2   $REG3
001254  000000                          0

                                ;ITEM   5
```

```
1141  001256  023613                    EM5     ;'DRU' ON SEEK; PUT DRIVE ON 'LOAD' BACK TO 'RUN'
1142  001260  024652                    DH1     ;PC     RKCS    RKER    RKDS    RKDA
1143  001262  025506                    DT1     ;$ERRPC $REG0   $REG1   $REG2   $REG3
1144  001264  000000                    0
1145
1146                          ;ITEM   6
1147
1148  001266  023662                    EM6     ;R/W/S RDY NOT SET AFTER SEEK
1149  001270  024652                    DH1     ;PC     RKCS    RKER    RKDS    RKDA
1150  001272  025506                    DT1     ;$ERRPC $REG0   $REG1   $REG2   $REG3
1151  001274  000000                    0
1152
1153                          ;ITEM   7
1154
1155  001276  023717                    EM7     ;SIN ON WRITE FMT
1156  001300  024750                    DH7     ;PC     RKCS    RKER    RKDS    RKDA    CYLINDER
1157  001302  025522                    DT7     ;$ERRPC $REG0   $REG1   $REG2   $REG3   $REG4
1158  001304  000000                    0
1159
1160                          ;ITEM   10
1161
1162  001306  023736                    EM10    ;'ERR' ON DOING WRITE FMT
1163  001310  024750                    DH7     ;PC     RKCS    RKER    RKDS    RKDA    CYLINDER
1164  001312  025522                    DT7     ;$ERRPC $REG0   $REG1   $REG2   $REG3   $REG4
1165  001314  000000                    0
1166
1167                          ;ITEM   11
1168
1169  001316  023767                    EM11    ;SIN ON READ FMT
1170  001320  025025                    DH11    ;PC     RKCS    RKER    RKDS    RKDA:
1171                                            ;DRV#   CYL     SUR     SEC
1172  001322  025540                    DT11    ;$ERRPC $REG0   $REG1   $REG2
1173                                            ;$REG4  $REG5   $REG6   $REG7
1174  001324  000000                    0
1175
1176                          ;ITEM   12
1177
1178  001326  024005                    EM12    ;'ERR' ON READ FMT
1179  001330  024750                    DH7     ;PC     RKCS    RKER    RKDS    RKDA    CYLINDER
1180  001332  025522                    DT7     ;$ERRPC $REG0   $REG1   REG2    $REG3   $REG4
1181  001334  000000                    0
1182
1183                          ;ITEM   13
1184
1185  001336  024027                    EM13    ;WRONG HEADERS FROM 'SEC #
1186  001340  025122                    DH13    ;SECTOR #       HEADER RECVD
1187  001342  000000                    0
1188  001344  015250                    ESR13   ;USE THIS SUBROUTINE FOR TYPING OUT ERROR DATA
1189
1190                          ;ITEM   14
1191
1192  001346  024056                    EM14    ;ERROR ON IMPLIED SEEK FROM CYLA TO CYLB
1193  001350  025141                    DH14    ;PC     CYLA    CYLB    RKER    RKDS    TRY#
1194  001352  025522                    DT7     ;$ERRPC $REG0   $REG1   $REG2   $REG3   $REG4
1195  001354  000000                    0
1196
```

```
1197                                    ;ITEM   15
1198
1199    001356  024131                          MS15    ;READ WRONG HDRS FROM CYLB ABOV
1200    001360  025122                          DH13    ;SEC#   HEADER RECVD
1201    001362  000000                          0
1202    001364  015156                          ESR15   ;GO TO "ESR15" FOR TYPING OUT
1203
1204                                    ;ITEM   16
1205
1206    001366  024173                          EM16    ;READ WRONG, FIRST WORD FROM SECTOR 0, 'CYLB' (ON IMPLIED SEEK FROM CYLA
1207    001370  025220                          DH16    ;PC     CYLA    CYLB    EXPCT   RECVD   TRY#
1208    001372  025522                          DT7     ;$ERRPC $REG0   $REG1   $REG2   $REG3   $REG4
1209    001374  000000                          0
1210
1211                                    ;ITEM   17
1212
1213    001376  024300                          MS17    ;READ FIRST WORD FROM SECTOR 1, 'CYLB' ABOVE
1214    001400  025275                          DH17    ;PC     CYLB    EXPCT   RECVD
1215    001402  025562                          DT17    ;$ERRPC $REG0   $REG1   $REG2
1216    001404  000000                          0
1217
1218                                    ;ITEM   20
1219
1220    001406  024346                          EM20    ;READ WRONG HEADER ON IMPLIED SEEK FROM 'CYLA' TO 'CYLB'
1221    001410  025122                          DH13    ;SECTOR # HEADER RECVD
1222    001412  000000                          0
1223    001414  015324                          ESR20   ;USE THIS SUBROUTINE FOR TYPING OUT ERROR DATA
1224
1225
1226                                    ;ITEM   21
1227
1228    001416  024434                          EM21    ;EROR ON DOING WRITE ON DSK
1229    001420  024652                          DH1     ;PC     RKCS    RKER    RKDS    RKDA
1230    001422  025506                          DT1     ;$ERRPC $REG0   $REG1   $REG2   $REG3
1231    001424  000000                          0
1232
1233                                    ;ITEM   22
1234
1235    001426  024470                          EM22    ;SIN ON DOING WRITE
1236    001430  024652                          DH1     ;PC     RKCS    RKER    RKDS    RKDA
1237    001432  025505                          DT1     ;$ERRPC $REG0   $REG1   $REG2   $REG3
1238    001434  000000                          0
1239
1240                                    ;ITEM   23
1241
1242    001436  024513                          EM23    ;HE ON DOING READ
1243    001440  025025                          DH11    ;PC     RKCS    RKER    RKDS    RKDA:
1244                                                    ;DRV#   CYL     SUR     SEC
1245    001442  025540                          DT11    ;$ERRPC $REG0   $REG1   $REG2
1246                                                    ;$REG4  $REG5   $REG6   $REG7
1247    001444  000000                          0
1248
1249                                    ;ITEM   24
1250
1251    001446  024534                          EM24    ;CSE ON READ
1252    001450  025333                          DH24    ;PC     TRY#    RKCS    RKER    RKDS    RKDA:
```

```
                                                        ;DRV#   CYL     SUR     SEC
001452  025574                              DT24      ;$ERRPC $REG10  $REG0   $REG1   $REG2
                                                        ;$REG4  $REG5   $REG6   $REG7
001454  000000                              0

                            ;ITEM   25

001456  024550                              EM25      ;DATA ERROR ON READ FROM DISK ADDRESS
001460  025440                              DH25      ;WORD#  EXPCT   RECVD   CYL     SUR     SEC
001462  000000                              0
001464  015412                              ESR25     ;USE THIS ROUTINE FOR ERROR REPORTING

                            ;ITEM   26

001466  024612                              EM26      ;HE ON WRT CHK
001470  025025                              DH11      ;PC     RKCS    RKER    RKDS    RKDA:
                                                        ;DRV#   CYL     SUR     SEC
001472  025540                              DT11      ;$ERRPC $REG0   $REG1   $REG2
                                                        ;$REG4  $REG5   $REG6   $REG7
001474  000000                              0

                            ;ITEM   27

001476  024630                              EM27      ;WRT CHK EROR
001500  025333                              DH24      ;PC     TRY#    RKCS    RKER    RKDS    RKDA:
                                                        ;DRV#   CYL     SUR     SEC
001502  025574                              DT24      ;$ERRPC $REG10  $REG0   $REG1   $REG2
                                                        ;$REG4  $REG5   $REG6   $REG7
001504  000000                              0

                            ;ITEM   30

001506  024645                              EM30      ;ERROR
001510  024652                              DH1       ;PC     RKCS    RKER    RKDS    RKDA
001512  025506                              DT1       ;$ERRPC $REG0   $REG1   $REG2   $REG3
001514  000000                              0

                            ;TAGS AND GENERAL DATA AREA

001516  000                 FTITLE: .BYTE   0         ;FLAG, SET AFTER PRINTING PROGRAM TITLE
001517  000                 FFUNC:  .BYTE   0         ;FLAG SET, TO INDICATE ENTRY INTO FUNCTION PROGRAM
001520  000                 RKDPCH: .BYTE   0         ;FLAG, SET TO INDICATE RKDP CHAIN MODE
001521  000                 LUPSW:  .BYTE   0         ;FLAG, SET TO INDICATE THAT A
                                                      ;PARTICULAR TEST WAS SELECTED BY USER (SW 8)

001522  000                 DRVDON: .BYTE   0         ;CONTAINS NUMBER OF DRIVES THAT HAVE
                                                      ;BEEN ALREADY CHECKED
001523  000                 DRIVS:  .BYTE   0         ;CONTAINS TOTAL # OF DRIVES PRESENT
```

```
                                        .EVEN

001524  000000                  DRVPTR: 0               ;CONTAINS POINTER TO INDICATOR STARTING
                                                        ;WHICH CHECKING SHOULD BE DONE FOR NEXT
                                                        ;AVAILABLE DRIVE
001526  000000                  DRIVAD: 0               ;CONTAINS THE ADDRESS OF THE DRIVE
                                                        ;BEING TESTED

001530  000000                  DRIV0:  000000          ;THESE ARE FLAGS TO INDICATE
001532  020000                  DRIV1:  020000          ;THAT A PARTICULAR DRIVE IS
001534  040000                  DRIV2:  040000          ;PRESENT.  BIT 0 IS SET TO
001536  060000                  DRIV3:  060000          ;INDICATE THAT.  BITS 13, 14, 15
001540  100000                  DRIV4:  100000          ;CONTAIN THE LOGICAL DRIVE
001542  120000                  DRIV5:  120000          ;ADDRESS
001544  140000                  DRIV6:  140000
001546  160000                  DRIV7:  160000

001550  000000                  RETRY1: 0               ;GENERAL REGISTERS
001552  000000                  RETRY2: 0
001554  000000                  RETRY3: 0

001556  000000                  INADR:  0               ;CONTAINS INNER ADDRESS
001560  000000                  OUTADR: 0               ;CONTAINS OUTER ADDRESS
001562  000000                  TIMER:  0

001564  000015                  BUFR:   .BLKW   13.     ;GENERAL BUFFERS
001616  000015                  BUFR1:  .BLKW   13.
001650  000015                  BUFR2:  .BLKW   13.

                                ;IN CASE, YOU WANT TO USE BUFFERS STARTING AT SOME OTHER MEMORY
                                ;ADDRESS YOU CAN DO SO BY CHANGING THE FOLLOWING POINTERS.
                                ;BOTH THE BUFFERS SHOULD BE 768 (DECIMAL) WORDS LONG.

001702  025620                  PBUF0:  IOBUF0          ;POINTER TO THE STARTING ADDRESS OF THE
                                                        ;BUFFER USED TO READ INTO FROM DIBK.
001704  030620                  PBUF1:  IOBUF1          ;POINTER TO STARTING ADDRESS OF BUFFER
                                                        ;IN WHICH PATTERNS ARE GENERATED. (WRITING
                                                        ;IS DONE FROM THIS BUFFER)
001706  000000                  BUFLG0: .WORD   0       ;FLAG FOR 'IOBUF0'
001710  000000                  BUFLG1: .WORD   0       ;FLAG FOR 'IOBUF1'

001712  007670                  PAT0:   PTGEN0          ;ADRES OF 'PATRN GENERATOR 0'
                                                        ;ROUTINE
001714  007752                  PAT1:   PTGEN1          ;ADRES OF 'PATRN GENERATOR 1'

001716  010054                  PAT2:   PTGEN2          ;ADRES OF 'PATRN GENRATOR 2'

001720  010116                  PAT3:   PTGEN3          ;ADRES OF 'PATRN GENRATOR 3'
```

```
1365
1366  001722  000000          PRSPAT: .WORD   0       ;CONTAINS THE POINTER TO THE
1367                                                  ;ADRES OF 1 OF THE 3 'PATRN
1368                                                  ;GENRATOR' ROUTINES
1369  001724  000000          NXTPAT: .WORD   0       ;SAME AS ABOVE
1370
1371  001726  000000          PGSUBR: .WORD   0
1372
1373  001730  000000          DSKADR: .WORD   0       ;CONTAINS DISK ADRES (DA)
1374
1375  001732  000000          BUSADR: .WORD   0       ;CONTAINS BUS ADRES (BA)
1376
1377  001734  000000          WRDCNT: .WORD   0       ;CONTAINS WORD COUNT
1378
1379  001736  000000          WDSKAD: .WORD   0       ;CONTAINS DISK ADRES
1380
1381  001740  000000          WBUSAD: .WORD   0       ;CONTAINS BUS ADRES
1382
1383  001742  000000          WWRDCN: .WORD   0       ;CONTAINS WORD COUNT
1384
1385  001744  000000          BUFNO:  .WORD   0       ;CONTAINS STARTING ADRES
1386
1387  001746  000000          ADRES:  .WORD   0       ;OF A BUFFER
1388
1389                          ;RK11 REGISTERS
1390                          ;IF FOR ANY REASON THE REGISTER ADDRESSES ARE DIFFERENT FROM
1391                          ;THESE (BELOW), THE CONTENTS OF THE APPROPRIATE POINTERS SHOULD
1392                          ;BE MODIFIED SO THAT THE CORRECT REGISTER ADDRESS IS USED.
1393
1394
1395  001750  177400          RKDS:   177400
1396  001752  177402          RKER:   177402
1397  001754  177404          RKCS:   177404
1398  001756  177406          RKWC:   177406
1399  001760  177410          RKBA:   177410
1400  001762  177412          RKDA:   177412
1401  001764  177416          RKDB:   177416
1402
1403  001766  000200          RKPRI:  200             ;CONTAINS THE CPU LEVEL (4) AT WHICH
1404                                                  ;RK11 NORMALLY INTERRUPTS.  THIS WORD
1405                                                  ;SHOULD BE CHANGED IF RK11 IS DESIGNATED
1406                                                  ;A BR LEVEL OTHER THAN 5.EXP: IF IT
1407                                                  ;IS CHANGED TO 6, THE CPU LEVEL WOULD
1408                                                  ;BE 1 LESS (5) & HENCE THIS WORD
1409                                                  ;SHOULD BE 240 (BIT POSITIONS ARE
1410                                                  ;IDENTICAL TO THE PRIORITY BITS IN PSW)
1411  001770  000220          RKVEC:  220             ;CONTAINS THE NORMAL VECTOR ADDRESS
1412                                                  ;TO WHICH THE RK11 INTERRUPTS.  IF THE
1413                                                  ;VECTOR ADDRESS HAS BEEN CHANGED, MODIFY
1414                                                  ;THIS WORD.
1415
1416
1417
1418
1419  001772  000000          INDX1:  0               ;GENERAL INDEX REGISTERS
1420  001774  000000          INDX2:  0
```

```
1421  001776  000000                  INDX3:  0
1422  002000  000000                  INDX4:  0
1423
1424  002002  000000                  ERCNT1: 0               ;GENERAL REGISTERS
1425  002004  000000                  ERCNT2: 0               ;GENERAL REGISTERS
1426  002006  000000                  ERCNT3: 0               ;GENERAL REGISTERS
1427  002010  000000                  ERCNT4: 0
1428  002012  000000                  ERCNT5: 0
1429  002014  000000                  ERCNT6: 0
1430  002016  000000                  ERCNT7: 0
1431  002020  000000                  ERCNT8: 0
1432
1433
1434                                  ;*THE FOLLOWING TABLE CONTAINS THE CYLINDERS BETWEEN WHICH THE SEEKS WILL BE
1435                                  ;*TIMED. THEY HAVE BEEN SELECTED TO GIVE SOME TYPICAL SEEKS TIMES FOR THE
1436                                  ;*3 SEEK SPEEDS. IF FOR  ANY REASON YOU WANT  TO TIME SEEKS BETWEEN ANY
1437                                  ;*OTHER SET OF CYLINDERS, MAKE CHANGES IN THE CORRESPONDING SEEK CYLINDER
1438                                  ;*ADDRESSES.
1439
1440                                  ;*OUTER CYLINDER ADDRESS, FROM WHERE SEEK WILL BE DONE
1441  002022  000000                  SOAD:   0               ;CYLINDER 0
1442  002024  000000                          0               ;  "  0
1443  002026  013140                          13140           ;  "  179
1444  002030  000000                          0               ;  "  0
1445  002032  000000                          0               ;  "  0
1446  002034  000000                          0               ;  "  0
1447  002036  000000                          0               ;  "  0
1448
1449                                  ;*INNER ADDRESS, TO WHICH SEEK WILL BE DONE
1450  002040  014500                  SIAD:   14500           ;CYLINDDER 202, LAST
1451  002042  000040                          40              ;  "  1
1452  002044  013240                          13240           ;  "  181
1453  002046  000140                          140             ;  "  3
1454  002050  001000                          1000            ;  "  16
1455  002052  002000                          2000            ;  "  32
1456  002054  006200                          6200            ;  "  100
1457
1458
1459
1460                                  ;FOLLOWING POINTERS ARE USED TO TRANSFER CONROL TO THE
1461                                  ;TEST SELECTED BY USING SW 8.  IF ANY MORE TESTS ARE
1462                                  ;ADDED TO THIS PROGRAM ADDITIONAL POINTERS SHOULD BE INSERTED.
1463  002056  004044                  PT1:    TST1+2
1464  002060  004410                  PT2:    TST2+2
1465  002062  004706                  PT3:    TST3+2
1466  002064  005376                  PT4:    TST4+2
1467  002066  006404                  PT5:    TST5+2
1468  002070  007142                  PT6:    TST6+2
1469  002072  010222                  PT7:    TST7+2
1470  002074  011760                  PT10:   TST10+2
1471  002076  012522                  PT11:   TST11+2
1472
1473
1474                                  ;MESSAGES & ASCII STRINGS
1475  002100  005015  044523  000116  MSG1:   .ASCIZ  <15><12>/SIN/
1476
```

```
1477  002106  005015  045523  000105  MSG2:   .ASCIZ  <15><12>/SKE/
1478
1479  002114  005015  042524  052123  MSG3:   .ASCIZ  <15><12>/TEST # ABORTED:/
1480  002122  021440  040440  047502
1481  002130  052122  042105  000072
1482
1483  002136  005015  051120  043517  MSG4:   .ASCIZ  <15><12>/PROG ABORTED/
1484  002144  040440  047502  052122
1485  002152  042105     000
1486
1487  002155     015  051012  040505  MSG5:   .ASCIZ  <15><12>/READ HDRS OK FROM CYLB ABOVE/
1488  002162  020104  042110  051522
1489  002170  047440  020113  051106
1490  002176  046517  041440  046131
1491  002204  020102  041101  053117
1492  002212  000105
1493
1494  002214  054105  041520  042124  MSG6:   .ASCIZ  /EXPCTD HDR= /
1495  002222  044040  051104  020075
1496  002230     000
1497
1498  002231     040  050040  036503  MSG7:   .ASCIZ  /  PC= /
1499  002236  000040
1500
1501  002240  005015  047103  051124  MSG10:  .ASCIZ  <15><12>/CNTRL RDY DIDN'T SET/
1502  002246  020114  042122  020131
1503  002254  044504  047104  052047
1504  002262  051440  052105     000
1505
1506  002267     123  041505  051124  MSG11:  .ASCIZ  /SECTR  EXPC P-HDR  RECV P-HDR/
1507  002274  020040  054105  041520
1508  002302  050040  044055  051104
1509  002310  020040  042522  053103
1510  002316  050040  044055  051104
1511  002324     000
1512
1513  002325     015  051012  053457  MSG12:  .ASCIZ  <15><12>"R/W/S RDY NOT SET"
1514  002332  051457  051040  054504
1515  002340  047040  052117  051440
1516  002346  052105     000
1517
1518  002351     040  052040  054522  MSG13:  .ASCIZ  /  TRY #:/
1519  002356  021440  000072
1520
1521
1522  002362  005015  051104  053111  MSG14:  .ASCIZ  <15><12>/DRIVE /
1523  002370  020105     000
1524
1525  002373     040  020040          BLNK13: .ASCII  /   /
1526  002376     040                  BLNK10: .ASCII  / /
1527  002377     040                  BLNKS9: .ASCII  / /
1528  002400     040                  BLNKS8: .ASCII  / /
1529  002401     040                  BLNKS7: .ASCII  / /
1530  002402     040                  BLNKS6: .ASCII  / /
1531  002403     040                  BLNKS5: .ASCII  / /
1532  002404     040                  BLNKS4: .ASCII  / /
```

```
  1533  002405  040                 BLNKS3: .ASCII  / /
  1534  002406  040                 BLNKS2: .ASCII  / /
  1535  002407  040  000             BLNKS1: .ASCIZ  / /
  1536
  1537          002412                      .EVEN
  1538  002412  000000              FDRIVE: 0
  1539  002414  000000              FDRVE1: 0
  1540  002416  000000              DRHOLD: 0
  1541
```

```
1542
1543                                    ;THIS IS THE HANDLER FOR UNEXPECTED TIME OUT. PRESSING CONTINUE WILL
1544                                    ;RESTART THE PROGRAM.
1545
1546
1547  002420  011600                    BADTMO: MOV     (SP),R0 ;SAVE PC WHERE TIME OUT OCCURED
1548  002422  005740                            TST     -(R0)
1549  002424  022626                            CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
1550  002426  104400  002434                   TYPE    ,65$            ;;TYPE ASCIZ STRING
1551  002432  000407                            BR      64$             ;;GET OVER THE ASCIZ
1552                                    ;;65$:  .ASCIZ  <15><12>/TIMOUT:PC=/
1553  002452                            64$:
1554  002452  010046                            MOV     R0,-(SP)        ;SET UP FOR TYPING OUT PC
1555  002454  104401                            TYPOC                   ;GO TYPE OUT OCTAL PC
1556  002456  000000                            HALT
1557
1558  002460                            START:
1559                                    .SBTTL  INITIALIZE THE COMMON TAGS
1560                                    ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1561  002460  012706  001100                    MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1562  002464  005026                            CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1563  002466  022706  001140                    CMP     #SWR,R6 ;;DONE?
1564  002472  001374                            BNE     .-6             ;;LOOP BACK IF NO
1565  002474  012706  001100                    MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1566                                    ;;INITIALIZE A FEW VECTORS
1567  002500  012737  016600  000020            MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1568  002506  012737  000340  000022            MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1569  002514  012737  022070  000034            MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1570  002522  012737  000340  000036            MOV     #340,@#TRAPVEC+2;LEVEL 7
1571  002530  012737  022162  000024            MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1572  002536  012737  000340  000026            MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1573  002544  012737  002544  001106            MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1574  002552  012737  002552  001110            MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1575                                    ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1576                                    ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1577  002560  013746  000004                    MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1578  002564  012737  002620  000004            MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1579  002572  012737  177570  001140            MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1580  002600  012737  177570  001142            MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1581  002606  022777  177777  176324            CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1582  002614  001012                            BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1583                                                                    ;;AND  THE HARDWARE SWR IS NOT = -1
1584  002616  000403                            BR      65$             ;;BRANCH IF NO TIMEOUT
1585  002620  012716  002626          64$:      MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1586  002624  000002                            RTI
1587  002626  012737  000176  001140  65$:      MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1588  002634  012737  000174  001142            MOV     #DISPREG,DISPLAY
1589  002642  012637  000004          66$:      MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1590
1591  002646  012737  016754  000030            MOV     #$ERROR,@#EMTVEC
1592  002654  012737  000340  000032            MOV     #340,@#EMTVEC+2
1593  002662  000005                            RESET
1594
1595
1596                                    .SBTTL  TYPE PROGRAM NAME
1597                                    ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
```

```
1598  002664  005227  177777                    INC     #-1             ;;FIRST TIME?
1599  002670  001044                            BNE     67$             ;;BRANCH IF NO
1600  002672  104400  002730                    TYPE    ,68$            ;;TYPE ASCIZ STRING
1601                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1602  002676  005737  000042                    TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1603  002702  001006                            BNE     69$             ;;BRANCH IF YES
1604  002704  023727  001140  000176            CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1605  002712  001005                            BNE     70$             ;;BRANCH IF NO
1606  002714  104405                            GTSWR                   ;;GET SOFT-SWR SETTINGS
1607  002716  000403                            BR      70$
1608  002720  112737  000001  001134  69$:     MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
1609  002726                        70$:
1610  002726  000425                            BR      67$             ;;GET OVER THE ASCIZ
1611                                ;;68$:  .ASCIZ  <CRLF><15><12>/RK11 DYNAMIC TESTS MAINDEC-11-DZRKL-C/<CRLF>
1612  003002                        67$:
1613
1614
1615  003002  105737  001517        START1: TSTB    FFUNC           ;FUNCTION PROGRAM SELECTED?
1616  003006  001404                        BEQ     7$              ;NO
1617  003010  105037  001517                CLRB    FFUNC           ;YES, CLEAR THE FLAG
1618  003014  000137  022344                JMP     @#FUNBEG        ;GO TO 'FUNTION SELECTION PROGRAM'
1619  003020  012700  001520        7$:     MOV     #RKDPCH,R0      ;CLEAR FLAGS FROM
1620  003024  105020                5$:     CLRB    (R0)+           ;'RKDPCH' TO 'DRIVAD'
1621  003026  020027  001530                CMP     R0,#DRIVAD+2
1622  003032  001374                        BNE     5$
1623  003034  012701  177770                MOV     #-10,R1
1624  003040  042720  000003        6$:     BIC     #3,(R0)+        ;CLEAR BIT 0'S IN 'DRIVE
1625  003044  005201                        INC     R1              ;PRESENT' FLAGS.
1626  003046  001374                        BNE     6$
1627
1628
1629                                ;THE FOLLOWING CODE FINDS OUT THE PROGRAM CONTROL MODE:
1630                                ;PAPER TAPE (MANUAL), ACT11, RKDP CHAIN OR DUMP
1631
1632
1633  003050  005737  000042                    TST     @#42            ;IS LOC 42 0?
1634  003054  001005                            BNE     1$              ;YES, BRANCH
1635  003056  123727  000041  000002            CMPB    @#41,#2         ;DOES BYTE 41 CONTAIN 2?
1636  003064  001410                            BEQ     2$              ;YES, IT IS RKDP DUMP MODE
1637                                                                    ;NO, PROGRAM LOADED BY PAPER TP
1638  003066  000501                            BR      ST2
1639  003070  123727  000041  000002  1$:      CMPB    @#41,#2         ;DOES BYTE 41 CONTAIN 2?
1640  003076  001456                            BEQ     3$              ;YES, RKDP CHAIN MODE
1641  003100  105737  000041                    TSTB    @#41            ;BYTE41 0?
1642  003104  001472                            BEQ     ST2             ;ACT11
1643
1644  003106                        2$:
1645  003106  104400  003114                    TYPE    ,65$            ;;TYPE ASCIZ STRING
1646  003112  000425                            BR      64$             ;;GET OVER THE ASCIZ
1647                                ;;65$:  .ASCIZ  <15><12>/REPLACE DRO RKDP-PAK BY OTHER & TYPE CR/
1648  003166                        64$:
1649  003166  104400  003174                    TYPE    ,67$            ;;TYPE ASCIZ STRING
1650  003172  000415                            BR      66$             ;;GET OVER THE ASCIZ
1651                                ;;67$:  .ASCIZ  <15><12>/IF NOT PUT DRO ON LOAD/
1652  003226                        66$:
1653
```

```
1654  003226  104407                          RDCHR                   ;READ CHAR FROM KBRD
1655  003230  005726                          TST     (R6)+           ;POP UP STACK
1656  003232  000417                          BR      ST2             ;RKDP DUMP MODE
1657
1658  003234                          3$:
1659  003234  104400  003242                  TYPE    ,69$            ;;TYPE ASCIZ STRING
1660  003240  000411                          BR      68$             ;;GET OVER THE ASCIZ
1661                                  ;;69$:  .ASCIZ  <15><12>/DRO NOT TESTED/
1662  003264                          68$:
1663  003264  112737  000001  001520          MOVB    #1,RKDPCH       ;SET FLAG INDICATING RKDP CHAIN MODE
1664
1665  003272  012737  002420  000004  ST2:    MOV     #BADTMO,@#4     ;SET TIME OUT VECTOR FOR
1666                                                                  ;UNEXPECTED TIME OUT
1667
1668                                  ;THIS CODE FINDS WHICH DRIVES ARE PRESENT & PRINTS OUT THE DRIVE
1669                                  ;DRIVE NUMBERS THAT WERE FOUND ON LINE.
1670
1671  003300  104400  003306                  TYPE    ,65$            ;;TYPE ASCIZ STRING
1672  003304  000411                          BR      64$             ;;GET OVER THE ASCIZ
1673                                  ;;65$:  .ASCIZ  <15><12>/DRIVES PRESENT/
1674  003330                          64$:
1675  003330  105037  001523                  CLRB    DRIVS           ;INITIALIZE NO. OF DRVS PRESENT
1676  003334  005001                          CLR     R1
1677  003336  012702  001530                  MOV     #DRIV0,R2
1678  003342  005003                          CLR     R3              ;INITIALIZE COUNT TO 0
1679  003344  010177  176412          1$:     MOV     R1,@RKDA        ;ADRES A DRIVE
1680  003350  105777  176374                  TSTB    @RKDS           ;IS IT PRESENT?
1681  003354  100004                          BPL     2$              ;NO, BRANCH
1682  003356  105237  001523                  INCB    DRIVS           ;INCREMENT TOTAL # OF DRVS
1683  003362  052712  000001                  BIS     #1,(R2)         ;SET FLAG INDICATING THIS DRV PRSNT
1684  003366  005722                  2$:     TST     (R2)+
1685  003370  005203                          INC     R3              ;INCREMENT COUNT
1686  003372  062701  020000                  ADD     #20000,R1       ;ADRES THE NXT DRV
1687                                                                  ;CHKD ALL 8 DRIVES?
1688  003376  001362                          BNE     1$              ;IF NOT, GO CHK IF NEXT DRV PRSNT
1689
1690  003400  004737  023422                  JSR     PC,SIZEF        ;FIND WHICH ARE FS
1691  003404  105737  001523                  TSTB    DRIVS           ;WERE ANY DRIVES FOUND?
1692  003410  001010                          BNE     3$              ;YES, BRANCH
1693  003412  104400  003420                  TYPE    ,67$            ;;TYPE ASCIZ STRING
1694  003416  000403                          BR      66$             ;;GET OVER THE ASCIZ
1695                                  ;;67$:  .ASCIZ  / NONE/
1696  003426                          66$:
1697  003426  000137  015040                  JMP     $EOP            ;IF NONE WERE FOUND, GO
1698                                                                  ;TO THE END OF PROGRAM
1699  003432  005002                  3$:     CLR     R2              ;DRIVE NUMBER
1700  003434  012700  001530                  MOV     #DRIV0,R0       ;TABLE OF AVAIL DRIVES
1701  003440  105710                  5$:     TSTB    (R0)            ;DRIVE HERE?
1702  003442  001414                          BEQ     4$              ;NO
1703  003444  104400                          TYPE
1704  003446  001213                          $CRLF
1705  003450  010246                          MOV     R2,-(SP)        ;PUSH NO ON THE STACK
1706  003452  104402                          TYPOS                   ;TO TYPE OCTAL NO.
1707  003454     001                          .BYTE   1               ;TYPE 1 DIGIT, SUPRESS LDG 0'S
1708  003455     000                          .BYTE   0
1709  003456  032710  000002                  BIT     #2,(R0)         ;IS IT RK05F?
```

```
1710  003462  001404                        BEQ     4$              ;NO
1711  003464  104400  003472                TYPE    ,69$            ;;TYPE ASCIZ STRING
1712  003470  000401                        BR      68$             ;;GET OVER THE ASCIZ
1713                                ;;69$:  .ASCIZ  /F/
1714  003474                        68$:
1715  003474  005202                4$:     INC     R2              ;POINT TO NEXT DRIVE #
1716  003476  005720                        TST     (R0)+           ;NEXT DRIVE IN TABLE
1717  003500  020027  001547                CMP     R0,#DRIV7+1     ;ALL DONE?
1718  003504  002755                        BLT     5$              ;NO, CHECK REST
1719                                ;FIND OUT THE FIRST (FROM 0-7) DRIVE THAT IS PRESENT.  PUT THE ADDRESS
1720                                ;OF THAT DRIVE IN 'DRIVAD'.  INDICATE THAT DRIVE # WILL
1721                                ;BE TESTED.
1722
1723  003506  012737  001530  001524  ST3:    MOV     #DRIV0,DRVPTR
1724  003514  105037  001522                CLRB    DRVDON
1725  003520  005037  001526                CLR     DRIVAD
1726  003524  105037  001102        NXTDRV: CLRB    $TSTNM          ;RESET TEST NUMBER TO 1
1727  003530  005037  001112                CLR     $ERTTL          ;CLEAR ERROR COUNT FOR THIS DRIVE
1728  003534  013701  001524                MOV     DRVPTR,R1
1729  003540  032721  000001        1$:     BIT     #1,(R1)+        ;IS THIS DRIVE PRESENT?
1730  003544  001005                        BNE     2$              ;YES, BRANCH
1731  003546  020127  001550        4$:     CMP     R1,#DRIV7+2     ;CHECKED THE WHOLE LIST?
1732  003552  001372                        BNE     1$              ;NO
1733  003554  000137  015040                JMP     $EOP            ;YES, EXIT
1734
1735  003560  032761  160000  177776  2$:     BIT     #160000,-2(R1)  ;IS IT DRIVE 0?
1736  003566  001003                        BNE     3$              ;NO
1737  003570  105737  001520                TSTB    RKDPCH          ;IF IT IS; RKDP CHAIN MODE?
1738  003574  001364                        BNE     4$              ;YES, DON'T TEST DRIVE 0
1739  003576  010137  001524        3$:     MOV     R1,DRVPTR       ;NO, GO AHEAD
1740  003602  014104                        MOV     -(R1),R4        ;GET DRIVE NO. TO BE TESTED
1741  003604  005037  002414                CLR     FDRVE1
1742  003610  005037  002412                CLR     FDRIVE          ;SHOWS F IF -1
1743  003614  032704  000002                BIT     #2,R4           ;RK-05F?
1744  003620  001410                        BEQ     7$              ;NO
1745  003622  005237  002414                INC     FDRVE1          ;SHOWS F
1746  003626  032704  020000                BIT     #20000,R4       ;EVEN DRIVE?
1747  003632  001003                        BNE     7$              ;NO
1748  003634  012737  177777  002412        MOV     #-1,FDRIVE      ;RK05F AND EVEN
1749  003642  042704  000003        7$:     BIC     #3,R4
1750  003646  010437  001526                MOV     R4,DRIVAD       ;SET UP DRIVE ADRES
1751  003652  104400  002362                TYPE    ,MSG14
1752  003656  000241                        CLC
1753  003660  006104                        ROL     R4              ;TYPE OUT THE DRIVE NO.
1754  003662  006104                        ROL     R4
1755  003664  006104                        ROL     R4
1756  003666  006104                        ROL     R4
1757  003670  010446                        MOV     R4,-(SP)
1758  003672  104402                        TYPOS
1759  003674     001                        .BYTE   1
1760  003675     000                        .BYTE   0
1761
1762  003676  005737  002414                TST     FDRVE1          ;RK-05F?
1763  003702  001404                        BEQ     6$              ;NO
1764  003704  104400  003712                TYPE    ,65$            ;;TYPE ASCIZ STRING
1765  003710  000401                        BR      64$             ;;GET OVER THE ASCIZ
```

```
                                        ;;65$:  .ASCIZ  /F/
003714                                  64$:
003714                                  6$:
                                        ;IF SW 8 IS SET THEN FIND OUT WHICH TEST NUMBER IS
                                        ;SELECTED AND JUMP TO THAT TEST.

003714  105037  001521                       CLRB    LUPSW           ;CLEAR FLAG INDICATING SW8 SET
003720  032777  000400  175212               BIT     #SW8,@SWR       ;SW 8 SET?
003726  001445                               BEQ     TST1            ;NO, BRANCH
003730                                  5$:
003730  104400  003736                       TYPE    ,67$            ;;TYPE ASCIZ STRING
003734  000410                               BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  <15><12>/OCTAL TEST#?/
003756                                  66$:
003756  104411                               RDOCT
003760  012600                               MOV     (SP)+,R0
003762  001762                               BEQ     5$
003764  020027  000011                       CMP     R0,#11          ;CHECK TYPED IN TEST #
003770  003357                               BGT     5$              ;IS LEGAL, IF NOT ASK
003772  110037  001102                       MOVB    R0,$TSTNM
003776  005300                               DEC     R0              ;FORM POINTERS FOR THE TEST #
004000  006300                               ASL     R0
004002  016037  002056  001106               MOV     PT1(R0),$LPADR  ;ADJUST POINTERS FOR SCOPE
004010  013737  001106  001110               MOV     $LPADR,$LPERR   ;LOOP, ETC.
004016  105237  001521                       INCB    LUPSW           ;SET FLAG INDICATING TEST #
                                                                    ;SELECTED
004022  000177  175060                       JMP     @$LPADR         ;GO TO THE TEST SELECTED


                                        ;ON RECOVERY FROM POWER FALIURE RETURN HERE

                                        PWRFL:  CLR     R0
                                                CLR     R1
                                        1$:     INC     R1
                                                BNE     1$
                                                INCB    R0
                                                BNE     1$


                                        ;;**************************************************************
                                        ;*TEST 1         CHECK INNER LIMIT SWITCH & ELECTROMECHANICAL INTEGRITY
                                                ;*THIS TEST PERFORMS 200(8) PURE SEEKS FROM CYLINDER 0
                                                ;*TO CYLINDER 312 AND BACK TO 0.  IT IS SUPPOSED TO
                                                ;*CHECK THE INNER LIMIT SWITCH AND THE MECHANICAL
                                                ;*INTEGRITY OF THE DRIVE.  NOTE THAT A VISUAL CHECK FOR
                                                ;ANY MECHANICAL FAILURES WOULD BE VERY HELPFUL.
                                        ;;**************************************************************
004042  000004                         TST1:  SCOPE
004044  005000                                CLR     R0              ;INITIALIZE COUNT
004046  005001                                CLR     R1              ;INITIALIZE COUNT FOR # OF SEEKS
004050  005002                                CLR     R2              ;CONTAINS SEEK ADRES
004052  012737  004104  001110                MOV     #20$,$LPERR     ;SET RETURN ADRES FOR LUPING
```

```
1822
1823  004060  012703  001564                    MOV     #BUFR,R3        ;INITIALIZE POINTER TO THE TABLE
1824  004064  012704  177767                    MOV     #-11,R4         ;ALLOW ONLY 9 CNTRL-RDY, SIN, OR R/W/S RDY ERORS
1825  004070  012705  177770                    MOV     #-10,R5         ;ALLOW ONLY 8 DRU+DRE+ERR+DRY ERRORS
1826  004074  000402                    BR      2$
1827
1828  004076  005703                1$:   TST     R3              ;WAS THERE ANY ERROR?
1829  004100  001403                      BEQ     3$              ;NO, BRANCH
1830  004102  005003                2$:   CLR     R3              ;CLR EROR FLAG
1831  004104  104414                20$:  CON.RESET               ;GO DO CNTRL RESET. SUB ROUTINE
1832                                                              ;AT 'CNT.RST'
1833  004106  104415                      DRV.RESET               ;GO TO 'DRV.RST' & DO DRV RESET
1834
1835  004110  013777  001526  175644  3$:   MOV     DRIVAD,@RKDA    ;ADRES THE DRIVE
1836  004116  050277  175640              BIS     R2,@RKDA        ;SET SEEK ADRES
1837  004122  105777  175622              TSTB    @RKDS           ;DRIVE RDY?
1838  004126  100406                      BMI     21$             ;YES
1839  004130  004737  015754              JSR     PC,GT4RG        ;NO, GET RKCS, ER, DS, DA
1840  004134  104030                      ERROR   30              ;DRIVE RDY BIT IS NOT SET
1841                                                              ;IN RKDS
1842  004136  005203                      INC     R3              ;SET ERROR FLAG
1843  004140  005205                      INC     R5              ;ALLOW ONLY 5 ERORS, IF MORE
1844  004142  001515                      BEQ     18$             ;ABORT
1845
1846  004144  012777  000011  175602  21$:  MOV     #11,@RKCS       ;GO, SEEK
1847  004152  005200                4$:   INC     R0              ;WAIT FOR CNTRL RDY
1848  004154  001007                      BNE     5$              ;WAITED LONG?
1849                                                              ;IF YES, ERROR
1850  004156  004737  015754              JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
1851  004162  104001                      ERROR   1               ;CNTRL RDY DIDN'T SET AFTER
1852                                                              ;SEEK WAS DONE TO CYLINDER
1853                                                              ;SHOWN IN RKDA. GO TO 'RK11 LOGIC TESTS'
1854  004164  005203                      INC     R3              ;SET ERROR FLAG
1855  004166  005204                      INC     R4              ;EXIT THIS TEST IF THERE R 5 OR MORE ERRORS
1856  004170  001502                      BEQ     18$
1857  004172  000403                      BR      6$
1858  004174  105777  175554        5$:   TSTB    @RKCS           ;DID CNTRL RDY SET?
1859  004200  100364                      BPL     4$              ;IF NOT WAIT FOR IT
1860
1861  004202  005000                6$:   CLR     R0              ;INITIALIZE COUNT
1862  004204  032777  000100  175536        BIT     #100,@RKDS      ;R/W/S RDY SET?
1863  004212  001010                      BNE     7$              ;YES
1864  004214  005200                      INC     R0              ;WAIT FOR R/W/S RDY
1865  004216  001372                      BNE     6$+2
1866  004220  004737  015754              JSR     PC,GT4RG        ;GET RKCS, ER, DS, DA
1867  004224  104006                      ERROR   6               ;R/W/S RDY DID NOT SET WHEN SEEK
1868                                                              ;WAS DONE TO CYLINDER INDICATED IN RKDA
1869  004226  005203                      INC     R3              ;SET ERROR FLAG
1870  004230  005204                      INC     R4              ;IF MAXM EROR COUNT, ABORT
1871  004232  001461                      BEQ     18$
1872  004234  032777  001000  175506  7$:   BIT     #1000,@RKDS     ;SIN ERROR?
1873  004242  001406                      BEQ     8$              ;NO, BRANCH
1874  004244  004737  015754              JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
1875  004250  104002                      ERROR   2               ;SIN ERROR, ON DOING SEEK TO
1876                                                              ;CYL AS SHOWN IN RKDA
1877  004252  005203                      INC     R3              ;SET ERROR FLAG
```

```
1878  004254  005204                          INC     R4              ;IF MAXM EROR COUNT REACHED,
1879  004256  001447                          BEQ     18$             ;ABORT THE TEST
1880  004260  005777  175466          8$:     TST     @RKER           ;DRE ERROR?
1881  004264  100006                          BPL     10$             ;NO, BRANCH
1882
1883  004266  004737  015754                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
1884  004272  104003                          ERROR   3               ;DRE ON DOING SEEK TO CYLINDER
1885                                                                  ;AS SHOWN IN RKDA
1886  004274  005203                          INC     R3              ;SET ERROR FLAG
1887  004276  005205                          INC     R5              ;IF MAXM EROR COUNT REACHED,
1888  004300  001767                          BEQ     8$              ;ABORT THE TEST
1889
1890  004302  005777  175446          10$:    TST     @RKCS           ;'ERR' BIT IN RKCS SET?
1891  004306  100006                          BPL     12$             ;NO, BRANCH
1892  004310  004737  015754                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
1893  004314  104004                          ERROR   4               ;'ERR' IN RKCS SET, ON DOING SEEK
1894                                                                  ;TO CYL AS SHOWN IN RKDA. NOTE
1895                                                                  ;WHICH BIT IN RKER SET?
1896  004316  005203                          INC     R3              ;SET ERROR FLAG
1897  004320  005205                          INC     R5              ;IF MAXM EROR COUNT REACHED,
1898  004322  001425                          BEQ     18$             ;ABORT THE TEST
1899
1900  004324  032777  002000  175416  12$:    BIT     #2000,@RKDS     ;DRU SET?
1901  004332  001406                          BEQ     15$             ;NO, BRANCH
1902  004334  004737  015754                  JSR     PC,GT4RG        ;GO, GET RKCS, ER, DS, DA
1903  004340  104005                          ERROR   5               ;DRU SET, THIS IS AN IRRECOVERABLE
1904                                                                  ;ERROR.  HENCE PUT THE DRIVE ON
1905                                                                  ;LOAD, BACK TO RUN.  DRU ERROR
1906                                                                  ;SHOULD BE CLEARED, IF IT IS NOT
1907                                                                  ;1) THE HEAD POSITION TRANSDVCER LAMP
1908                                                                  ;IS INOPERATIVE
1909                                                                  ;2) OR ERASE OR WRT CURRENT PRESENT
1910                                                                  ;WITHOUT 'WRT GATE'
1911  004342  005203                          INC     R3              ;SET EROR FLAG
1912  004344  005205                          INC     R5              ;ALLOW ONLY 5 ERRORS
1913  004346  001413                          BEQ     18$             ;IF MORE THAN 5
1914                                                                  ;GO TO THE END OF THE PROGRAM
1915
1916  004350  005702                  15$:    TST     R2              ;WAS SEEKING TO 0 OR 312?
1917  004352  001402                          BEQ     16$             ;TO 0, BRANCH
1918                                                                  ;TO 312,
1919  004354  005002                          CLR     R2              ;SEEK NXT TIME TO 0
1920  004356  000647                          BR      1$              ;GO BAK & SK TO 0
1921
1922  004360  012702  014500          16$:    MOV     #14500,R2       ;SEEK NXT TIME TO 312
1923
1924  004364  005201                          INC     R1              ;DONE SEEKS 200 TIMES?
1925  004366  022701  000200                  CMP     #200,R1
1926
1927  004372  001241                          BNE     1$              ;IF NOT, GO BAK
1928  004374  000404                          BR      TST2            ;;EXIT
1929
1930
1931  004376  104400  002136          18$:    TYPE    ,MSG4
1932  004402  000137  015006                  JMP     @ST12
1933
```

```
                                ;;*****************************************************************
                                ;*TEST 2          FORMAT THE DISK
                                ;*THIS PROGRAM ASSUMES AN UNFORMATTED DISK AND ITS
                                ;*FORMATTING IS DONE IN THIS TEST. A SECTOR IS FORMATTED
                                ;*AT A TIME.  THE FIRST OWRD OF EVERY SECTOR IS WRITTEN
                                ;*TO BE A PSUEDO-HEADER CONTAINING THE DRIVE #, CYLINDER
                                ;*#, SURFACE AND SECTOR #.  THE FOLLOWING IS CHECKED
                                ;*1.  'SIN' IF 'SIN' OCCURS, A DRIVE RESET IS DONE
                                ;*AND THE SAME SECTOR IS FORMATTED AGAIN.  THREE
                                ;*RETRIES ARE DONE BEFORE AN ERROR MESSAGE IS PRINTED.
                                ;*2.  'ERR' ON FINDING THAT THE 'ERR' BIT SET, RKER
                                ;*SCANNED TO FIND OUT WHAT CAUSED IT AND THE
                                ;*ERROR IS REPORTED.
                                ;;*****************************************************************
004406  000004                  TST2:   SCOPE
004410  013737  001526  002416          MOV     DRIVAD,DRHOLD   ;SAVE DRIVE NUMBER
004416  005737  002412                  TST     FDRIVE          ;SEE IF EVEN RK-05F DRIVE
004422  001003                          BNE     11$             ;YES
004424  005737  002414                  TST     FDRVE1          ;ODD RK-05F?
004430  001125                          BNE     TST3            ;DO NOT FORMAT IF ODD RK-05F
004432                          11$:
004432  012702  177152                  MOV     #-626,R2        ;203 CYLINDERS, (406 TRAKS)
004436  012703  177764                  MOV     #-14,R3         ;12 SECTORS
004442  012701  177773                  MOV     #-5,R1          ;ALLOW ONLY 5 'SIN' ERRORS
004446  012705  177773                  MOV     #-5,R5          ;ALLOW ONLY 5 'ERR'S
004452  013704  001526                  MOV     DRIVAD,R4       ;STORE ADRES OF DRIVE.
004456  104414                  4$:     CON.RESET
004460  104415                          DRV.RESET               ;GO TO 'DR-RST' & DO DRIVE RESET
004462  005000                  1$:     CLR     R0              ;KEEP COUNT OF 'SIN' ERORS
                                                                ;ALLOW 3 RETRIES ON SIN
004464  005777  175264                  TST     @RKCS           ;ERR?
004470  100001                          BPL     3$              ;NO

004472  104414                          CON.RESET               ;GO TO 'CN-RST' & DO CONTROL RESET

004474  005046                  3$:     CLR     -(SP)
004476  012746  004504                  MOV     #12$,-(SP)
004502  000002                          RTI                     ;SET PRIORITY TO ZERO
004504  010437  025620          12$:    MOV     R4,OUTBUF       ;WRITE THIS WORD
004510  012777  025620  175242          MOV     #OUTBUF,@RKBA   ;FROM THIS ADRES
004516  010477  175240                  MOV     R4,@RKDA        ;ON THIS DISK SECTOR
004522  012777  177777  175226          MOV     #-1,@RKWC       ;WRITE 1 WORD
004530  012737  004456  001110          MOV     #4$,$LPERR      ;SET RETURN ADDRESS FOR
                                                                ;LUPING ON ERROR

004536  012777  002003  175210          MOV     #2003,@RKCS     ;GO WRT FMT

004544  104420                          CON.RDY                 ;WAIT FOR CONTROL READY
004546  032777  001000  175174  5$:     BIT     #1000,@RKDS     ;WAS THERE A SIN?
004554  001413                          BEQ     6$              ;NO, SKIP DOING DRV RESET
004556  004737  015730                  JSR     PC,GTSRG        ;GO, GET RKCS, ER, DS, DA, CYLINDER
004562  104007                          ERROR   7               ;SIN ERROR ON TRYING TO
                                                                ;WRT FMT ON CYLINDER AS
                                                                ;INDICATED IN RKDA.  3 RETRIES
                                                                ;ARE DONE
                                                                ;NOTE THAT BEFORE
```

```
1990  004564  104414                        CON.RESET               ;RETRYING A DRIVE RESET WAS DONE
1991  004566  104415                        DRV.RESET               ;GO TO 'DR-RST' & DO DRV RESET
1992  004570  005200                        INC     R0              ;INCRMNT SIN COUNT
1993  004572  022700  000003                CMP     #3,R0           ;ALLOW 3 RETRIES WERE THERE 3?
1994  004576  001332                        BNE     1$+2            ;IF NOT, GO & RETRY
1995
1996  004600  005201                        INC     R1              ;ALLOW ONLY 12 SIN ERRORS
1997                                                                ;IF MORE THAN 5 EXIT THIS TEST
1998  004602  001436                        BEQ     9$              ;IF MORE THAN 5 EXIT THIS TEST
1999  004604  005777  175144        6$:     TST     @RKCS           ;DID 'ERR' BIT SET IN RKCS?
2000  004610  100005                        BPL     7$              ;NO, BRANCH
2001  004612  004737  015730                JSR     PC,GTSRG        ;GO, GET RKCS, ER, DS,DA,CYL
2002  004616  104010                        ERROR   10              ;'ERR' OCCURED WHILE DOING
2003                                                                ;WRT FMT ON SECTOR, CYLINDER
2004                                                                ;AS INDICATED IN RKDA.
2005  004620  005205                        INC     R5              ;ALLOW ONLY 5 'ERR'S. IF
2006  004622  001426                        BEQ     9$              ;MORE THAN 5 EXIT THIS TEST
2007  004624  005204        7$:     INC     R4              ;INCRMNT DISK ADRES TO NXT SCTR
2008  004626  005203                        INC     R3              ;ALL 12 SECTORS DONE?
2009  004630  001314                        BNE     1$              ;IF NOT, GO BAK & FMT NXT SCTR
2010                                                                ;IF YES
2011  004632  012703  177764                MOV     #-14,R3         ;RESET COUNT FOR 12 SECTORS
2012  004636  042704  000017                BIC     #17,R4          ;CLR OUT SEC BITS
2013
2014  004642  062704  000020        8$:     ADD     #20,R4          ;ADRES THE NXT TRAK TO B FMTED
2015  004646  005202                        INC     R2              ;ALL TRAKS FMTED?
2016  004650  001304                        BNE     1$              ;IF NOT GO BAK B FMT NXT CYL, SUR 0
2017  004652  005237  002412                INC     FDRIVE          ;EVEN RK05F?
2018  004656  001004                        BNE     10$             ;NO
2019  004660  062737  020000  001526        ADD     #20000,DRIVAD   ;FORMAT ODD DRIVE OF F
2020  004666  000661                        BR      11$
2021  004670  013737  002416  001526  10$:  MOV     DRHOLD,DRIVAD   ;RESTORE DRIVE ADDR
2022  004676  000402                        BR      TST3            ;;EXIT
2023
2024  004700  004737  016564        9$:     JSR     PC,ABRT
2025
2026                          ;;*****************************************************************
2027                          ;*TEST 3         READ FORMAT OF THE DISK
2028                                  ;* IN THIS TEST, THE HEADERS FROM ALL THE SECTORS ARE READ
2029                                  ;* & CHECKED IF THEY ARE CORRECT.  THE FOLLOWING IS THE
2030                                  ;* TEST SEQUENCE.
2031                                  ;* 1.  READ 12 SECTORS (HDRS ONLY) AT A TIME
2032                                  ;* 2.  IF THERE IS A 'SIN' ERROR RETRY ONCE MORE, IF SAW AGAIN
2033                                  ;* REPORT ERROR & READ HEADER FROM NEXT CYLINDER
2034                                  ;* 3.  IF THERE IS 'ERR' IN RKCS, DO A CONTROL RESET, REPORT
2035                                  ;* ERROR & READ HEADER FROM NEXT CYLINDER.  IF THERE ARE
2036                                  ;* MORE THAN 5 ERRORS OF THIS KIND, THIS TEST WILL BE EXITED
2037                                  ;* 4.  THE 12 HEADERS ARE CHECKED.  IF THEY ARE CORRECT THE
2038                                  ;* NEXT CYLINDER IS READ.
2039                                  ;* IF THEY ARE NOT CORRECT, A RETRY IS DONE; IF AGAIN CORRECT
2040                                  ;* HEADERS ARE NOT RECIEVED, AN ERROR IS REPORTED.  THE
2041                                  ;* SECTOR #'S GIVING THE BAD HEADERS, & THE BAD HEADERS ARE
2042                                  ;* STORED.
2043                                  ;* 5.  IF INHIBIT TYPEOUT' SWITCH IS NOT SET, THE FIRST WORDS OF
2044                                  ;* THE 12 SECTORS (PSUEDO-HEADERS) ARE READ.  IN A PREVIOUS
2045                                  ;* TEST THE FIRST WORD OF EVERY SECTOR WAS WRITTEN
```

```
                                        ;* AS A SOFTWARE HEADER (CONSISTING OF DRIVE #, CYL#, SUR,SEC#)
                                        ;* THEN THE SECTOR # GIVING BAD HEADER, EXPECTED PSUEDO-HEADER.
                                        ;* & THE PS-HEADER RECIEVED ARE TYPED OUT.  THIS WOULD
                                        ;* WRONG, HEADER WAS READ WRONG, ETC.
                                        ;* 6.  THE NEXT CYLINDER IN LINE IS READ.  ORDER OF READING IS
                                        ;* CYL0,SUR0    CYL0,SUR1        CYL312,SUR1
                                    ;;*****************************************************************
004704  000004                  TST3:   SCOPE

004706  012737  177773  002002          MOV     #-5,ERCNT1      ;ALLOW ONLY 5 ERRORS (OF BAD HEADER
                                                                ;KIND FROM 5 CYLINDERS)
004714  012737  177766  002004          MOV     #-12,ERCNT2     ;ALLOW ONLY 12 'ERR'S
004722  012737  177773  002006          MOV     #-5,ERCNT3      ;ALLOW ONLY 5 ERRORS
004730  013705  001526                  MOV     DRIVAD,R5       ;SET DRIVE #,CYL ADRES=0
004734  012737  177152  001774          MOV     #-626,INDX2     ;313 CYLS (626 TRAKS) TO B READ
004742  104414                  4$:     CON.RESET               ;GO DO CONTROL RESET
004744  104415                          DRV.RESET               ;GO DO DRIVE RESET

004746  005037  001552          1$:     CLR     RETRY2          ;ALLOW 2 RETRIES IF HDRS READ WRONG
004752  005037  001550          2$:     CLR     RETRY1          ;ALLOW 2 RETRIES FOR 'SINS'

004756  012777  025620  174774  3$:     MOV     #OUTBUF,@RKBA   ;RD HDRS INTO LOC STARTING AT THIS
004764  010577  174772                  MOV     R5,@RKDA        ;FROM THIS DSK ADRES
004770  012777  177764  174760          MOV     #-14,@RKWC      ;12 HDRS TO BE READ
004776  012737  004742  001110          MOV     #4$,$LPERR      ;SET RETURN ADRES FOR LUPING ON ERROR

005004  012777  002005  174742          MOV     #2005,@RKCS     ;GO, RD FMT OF THIS CYLINDER

005012  104420                          CON.RDY                 ;WAIT FOR CNTRL RDY TO SET

005014  032777  001000  174726  5$:     BIT     #1000,@RKDS     ;'SIN' ERROR?
005022  001420                          BEQ     6$              ;NO, BRANCH
005024  004737  016006                  JSR     PC,GETINF

005030  104011                          ERROR   11              ;'SIN' OCCURED WHEN DOING RD FMT
                                                                ;FROM CYL SHOWN IN RKDA.  IT
005032  104414                          CON.RESET               ;DO CNTRL RESET
005034  104415                          DRV.RESET               ;GO, DO DRIVE RESET
005036  005237  001550                  INC     RETRY1          ;ALLOW ONLY 2 RETRIES FOR THIS  ERROR
005042  022737  000002  001550          CMP     #2,RETRY1       ;IF TRIED 2 TIMES REPORT
005050  001342                          BNE     3$              ;ERROR, OTHERWISE GO BAK & RETRY
                                                                ;WAS TRIED TWICE, BUT 'SIN'.
005052  005237  002006                  INC     ERCNT3          ;ALLOW 5 ERRORS AT MOST
005056  001002                          BNE     6$
005060  000137  005370                  JMP     16$

005064  005777  174664          6$:     TST     @RKCS           ;'ERR' IN RKCS?
005070  100010                          BPL     7$              ;NO, BRANCH
005072  004737  015730                  JSR     PC,GT5RG        ;GO, GET RKCS, ER,DS,DA,CYLNDR
005076  104012                          ERROR   12              ;'ERR' SET WHILE DOING RD FMT
                                                                ;FROM CYL SHOWN IN RKDA
005100  104414                          CON.RESET               ;GO DO CNTRL RESET
                                                                ;ALLOW ONLY 12 ERRORS OF THIS
005102  005237  002004                  INC     ERCNT2          ;KIND, IF MORE THAN FIVE ERRORS
                                                                ;SKIP THIS TEST
005106  001532                          BEQ     TST4            ;;EXIT
```

```
2102  005110  000520                        BR      14$             ;GO SET UP TO RD FMT FROM NXT
2103                                                                ;CYL IN LINE
2104                                                                ;CHECK THAT CORRECT HEADERS WERE RECVD.
2105                                                                ;SECTR # HAVING BAD HDR IS STORED ALONG
2106                                                                ;WITH BAD HDR
2107
2108  005112  004737  007042          7$:   JSR     PC,CHKHDRS      ;GO CHECK IF CORRECT HEADERS WERE READ
2109
2110  005116  005737  001776                TST     INDX3           ;WAS THERE A MISCOMPARISON?
2111  005122  001513                        BEQ     14$             ;IF NOT, GO SET UP TO RD FMT
2112                                                                ;NXT CYL IN LINE
2113  005124  012737  004752  001110        MOV     #2$,$LPERR
2114  005132  104013                        ERROR   13              ;CORRECT HDRS WERE NOT RECVD
2115                                                                ;FROM SECTRS AS TYPED OUT.
2116                                                                ;THE SAME CYLINDER WAS READ TWICE
2117  005134  005237  001552                INC     RETRY2          ;RETRY RD FMT ON SAME CYL AGAIN
2118  005140  022737  000002  001552        CMP     #2,RETRY2       ;TRIED RDING SAME CYL TWICE
2119  005146  001301                        BNE     2$              ;IF NOT, GO RD AGAIN
2120                                                                ;YES, REPORT ERROR
2121  005150  005237  002002                INC     ERCNT1          ;ALLOW ONLY 5 ERRORS OF THE
2122                                                                ;ABOVE TYPE.  IF MORE THAN 12
2123                                                                ;EXIT THIS TEST
2124  005154  001505                        BEQ     16$
2125
2126  005156                          20$:                          ;THE PSUEDO-HEADERS (FIRST WORD OF EVERY
2127                                                                ;SECTOR) FROM THIS CYLINDER (ABOVE,
2128                                                                ;THE CYLINDER THAT GAVE WRONG HEADERS)
2129                                                                ;WILL BE READ, NOW.  FOLLOWING WILL B TYPD OUT:
2130                                                                ;SEC#, EXPCTD PSUEDO-HDR, RECVD PHDR.
2131                                                                ;IF "INHIBIT TYPEOUT" SW IS SET THIS ENTIRE
2132                                                                ;READING & TYPING WILL BE SKIPPED
2133  005156  032777  020000  173754        BIT     #20000,@SWR     ;INHIBIT TYPEOUT?
2134  005164  001072                        BNE     14$             ;YES, SKIP THE FOLLOWING & GO
2135                                                                ;SET UP TO RD FMT NXT CYL IN LINE
2136
2137  005166  012701  177764                MOV     #-14,R1         ;READ FROM 12 SECTRS
2138  005172  010577  174564                MOV     R5,@RKDA        ;FROM THIS DSK-ADRES
2139  005176  012777  025620  174554        MOV     #OUTBUF,@RKBA   ;INTO THIS BUS-ADRES
2140  005204  012777  177777  174544  10$:   MOV     #-1,@RKWC       ;RD 1 WRD
2141
2142  005212  012777  000005  174534        MOV     #5,@RKCS        ;GO, RD
2143  005220  104420                        CON.RDY                 ;WAIT FOR CNTRL RDY
2144  005222  005777  174526                TST     @RKCS           ;ANY EROR?
2145  005226  100002                        BPL     15$             ;NO, PROCEED
2146  005230  104414                        CON.RESET               ;CLEAR THE EROR
2147  005232  000447                        BR      14$             ;EROR, SO COULDN'T READ PSUEDO-HDRS
2148
2149  005234  005201                  15$:  INC     R1              ;READ FROM ALL 12 SECS
2150  005236  001362                        BNE     10$             ;IF NOT GO RD THE NXT ONE
2151
2152                                                                ;TYPE OUT PSUEDO-HDRS CORRESPONDING TO
2153                                                                ;THE SECTORS WHICH GAVE BAD HEADERS
2154                                                                ;TYPE: SEC #, EXPC P-HDR, RECVD P-HDR
2155
2156  005240  104400                        TYPE                    ;TYPE OUT
2157  005242  002267                        MSG11
```

```
2158  005244  012701  001564          MOV     #BUFR,R1        ;SEC #'S ARE STORED HERE
2159
2160  005250  104400          11$:    TYPE                    ;TYPE CR, LF
2161  005252  001213                  $CRLF
2162
2163  005254  011102                  MOV     (R1),R2
2164  005256  012703  025620          MOV     #OUTBUF,R3      ;PSUEDO-HEADERS WHICH WERE
2165                                                           ;READ ARE STORED HERE
2166
2167  005262  005702          12$:    TST     R2              ;IS THIS SEC # CORRESPONDING TO THE
2168  005264  001403                  BEQ     13$             ;ONE IN ERROR
2169  005266  005302                  DEC     R2              ;R2 CONTAINS THE SEC #
2170  005270  005723                  TST     (R3)+
2171  005272  000773                  BR      12$
2172
2173  005274  011146          13$:    MOV     (R1),-(SP)      ;GO TYPEOUT SEC # GIVING
2174  005276  104402                  TYPOS                   ;MISCOMPARISON OF HEADERS
2175  005300     002                  .BYTE   2
2176  005301     000                  .BYTE   0               ;SUPRES LDG 0'S
2177
2178  005302  104400                  TYPE                    ;TYPE 2 BLANKS
2179  005304  002403                  BLNKS5
2180
2181  005306  010546                  MOV     R5,-(SP)        ;GO TYPE EXPCTD PSUEDO HEADER
2182  005310  051116                  BIS     (R1),(SP)
2183  005312  104401                  TYPOC
2184
2185  005314  104400                  TYPE                    ;TYPE 2 BLNKS
2186  005316  002401                  BLNKS7
2187
2188  005320  011346                  MOV     (R3),-(SP)      ;GO TYPE PSUEDO-HEADER RECVD
2189  005322  104401                  TYPOC
2190
2191  005324  005721                  TST     (R1)+           ;TYPED OUT ALL SEC #'S IN ERROR.
2192  005326  021127  177777          CMP     (R1),#177777
2193  005332  001346                  BNE     11$             ;IF NOT GO BAK & TYPE NXT
2194
2195  005334  104400  002231          TYPE    ,MSG7           ;TYPE OUT PC
2196  005340  012746  005156          MOV     #20$,-(SP)
2197  005344  104401                  TYPOC
2198  005346  104400  001213          TYPE    ,$CRLF
2199                                                           ;TYPE ROUTINE ENDS HERE
2200
2201                                                           ;FIND OUT NXT TRAK TO B READ
2202                                                           ;FORMATTED
2203
2204  005352  062705  000020  14$:    ADD     #20,R5          ;SET ADRES FOR SUR 0, NXT CYL IN LINE
2205  005356  005237  001774          INC     INDX2           ;READ ALL 313 CYLINDERS (626 TRAKS)?
2206  005362  001404                  BEQ     TST4            ;;EXIT
2207  005364  000137  004746          JMP     1$              ;IF NOT, GO BAK & READ NXT
2208
2209  005370  004737  016564  16$:    JSR     PC,ABRT
2210
2211                          ;;*****************************************************************
2212                          ;*TEST 4         SEEK PATTERNS: 0-312-0-311-...,USING IMPLIED SEEK
2213
```

```
                                        ;****TEST 2 (WRITING PSUEDO-HEADERS) SHOULD HAVE BEEN DONE BEFORE****
                                        ;**** DOING THIS TEST****
                                        ;*THIS TEST PERFORMS SEEKS (IMPLIED SEEKS USING 'READS') IN THE
                                        ;*FOLLOWING PATTERN.
                                        ;*0-312-0-311-0-310-.....0-1-0-0

                                        ;*THE FIRST WORD OF EVERY SECTOR IS A PSEUDO-HEADER (WRITTEN IN
                                        ;*A PREVIOUS TEST) CONSISTING OF DRIVE NO, CYLINDER NO., SURFACE
                                        ;*AND SECTOR NO.  AN IMPLIED SEEK IS DONE BY ISSUING A 'READ' FOR
                                        ;*THE PSEUDO-HEADER OF SECTOR 0, SURFACE 0.

                                        ;*IF A 'SIN' OCCURS TWO TRIES ARE DONE BEFORE ABORTING.  IF A 'SKE'
                                        ;*OCCURS IT COULD MEAN THAT 1) EITHER THE HEADERS WAS READ WRONG
                                        ;*OR 2) THE HEADS GOT POSITIONED ON THE WRONG CYLINDER.  IN
                                        ;*ORDER TO PROVIDE A FURTHER INSIGHT INTO THE PROBLEM, THE FOLLOWING
                                        ;*IS DONE:
                                        ;*THE HEADERS ARE READ FROM THE CYLINDER THAT GAVE 'SKE'.  IF THE HEADERS
                                        ;*ARE CORRECT IT IS SO REPORTED.  IF THE HEADERS ARE INCORECT, THEN THE
                                        ;*EXPECTED HEADERS AND THE RECEIVED ONES ARE REPORTED.  ONE MORE TRY IS
                                        ;*DONE (THE IMPLIED SEEK IS TRIED AGAIN BETWEEN THE CYLINDERS THAT GAVE RISE
                                        ;*TO 'SKE')

                                        ;*THE FOLLOWING ACTION IS TAKEN WHEN THERE IS NO 'SKE' OR 'SIN' BUT STILL THE
                                        ;*PSEUDO-HEADER IS READ WRONG:
                                        ;*FIRST THE HEADERS ARE READ FROM THAT CYLINDER AND CHECKED.  IF THEY ARE
                                        ;**CORRECT, IT IS SO REPORTED.  IF THEY ARE WRONG THEN THE EXPECTED AND RECEIVED
                                        ;*HEADERS ARE REPORTED.  THEN THE PSEUDO-HEADER FROM SECTOR 1 IS READ AND REPORT
                                        ;*ONE MORE TRY IS DONE BY REPEATING THE WHOLE PROCESS. (IMPLIED SEEK
                                        ;*BETWEEN THE TWO CYLINDERS AND READING PSEUDO-HEADER FROM SECTOR 0 OF THE DESTI
                                        ;*CYLINDER).

                                        ;*UP TO 12 ERRORS OF EACH KIND (SIN, SKE, BAD PSEUDO-HEADER) ARE ALLOWED.
                                        ;*IF ANY ERROR OCCURS MORE THAN 12 TIMES THE TEST IS ABORTED.
                                    ;;**************************************************************
005374  000004                  TST4:   SCOPE
005376  104414                          CON.RESET               ;GO DO CONTROL RESET
005400  104415                          DRV.RESET               ;GO DO DRIVE RESET

005402  005004                          CLR     R4              ;FLAG, CLR IF DOING IMPLIED
                                                                ;SEEK IN FROM 0 TO 'INADR'
                                                                ;=1, IF GOING FROM 'INADR'
                                                                ;OUT TO CYL 0
005404  012737  177465  001774          MOV     #-313,INDX2     ;313 SEEK PATTERNS
005412  012700  177764                  MOV     #-14,R0
005416  010037  002002                  MOV     R0,ERCNT1       ;ALLOW ONLY 12 ERRORS
005422  010037  002004                  MOV     R0,ERCNT2       ;OF THESE KINDS
005426  010037  002006                  MOV     R0,ERCNT3

005432  012737  014500  001556          MOV     #14500,INADR    ;'INADR' CONTAINS THE INER
                                                                ;CYL TO WHICH IMPLIED SEEK WILL
                                                                ;BE DONE

005440  005704                  1$:     TST     R4              ;GOING IN OR OUT?
005442  001005                          BNE     2$              ;GOING OUT, BRANCH
005444  013705  001556                  MOV     INADR,R5        ;SET CYL ADRES BITS FOR GOING IN
005450  053705  001526                  BIS     DRIVAD,R5       ;FORM DISK ADRES FOR INNER
```

```
2270  005454  000402                          BR      3$              ;CYLINDER
2271  005456  013705  001526          2$:     MOV     DRIVAD,R5       ;FORM DISK ADRES FOR OUTER
2272                                                                  ;CYLINDER - 0
2273                                                                  ;ALLOW 2 TRIES WHEN
2274  005462  012737  177776  001552  3$:     MOV     #-2,RETRY2      ;ERRORS OCCUR
2275  005470  012737  177776  001550  13$:    MOV     #-2,RETRY1
2276  005476  012737  177777  001554  4$:     MOV     #-1,RETRY3
2277  005504  000404                          BR      5$
2278  005506  104414                  6$:     CON.RESET
2279  005510  104415                          DRV.RESET
2280  005512  004737  006310                  JSR     PC,SBR1         ;REPOSITION HEADS TO PRE-ERROR CYL
2281
2282  005516  012777  177777  174232  5$:     MOV     #-1,@RKWC       ;READ 1 WORD
2283  005524  010577  174232                  MOV     R5,@RKDA        ;FROM THIS CYLINDER, SEC 0
2284  005530  012777  025620  174222          MOV     #OUTBUF,@RKBA   ;INTO THIS BUS ADRES
2285  005536  012737  005506  001110          MOV     #6$,$LPERR      ;SET RETURN ADRES FOR LUPING
2286                                                                  ;ON 'ERROR'
2287
2288  005544  012777  000005  174202          MOV     #5,@RKCS        ;GO, READ
2289
2290  005552  104420                          CON.RDY                 ;WAIT FOR CNTRL RDY
2291
2292  005554  032777  001000  174166          BIT     #1000,@RKDS     ;SIN?
2293  005562  001434                          BEQ     8$              ;NO, BRANCH
2294                                                                  ;YES, THERE WAS A SIN
2295  005564  004737  016170                  JSR     PC,ERR1         ;GO GET, CYLS BETW'N WHICH SK WAS TRIED
2296  005570  017737  174154  001170          MOV     @RKDS,$REG3
2297  005576  017737  174150  001166          MOV     @RKER,$REG2
2298  005604  104417  002100                  TYPMSG  ,MSG1
2299  005610  013737  001554  001172          MOV     RETRY3,$REG4    ;SAVE TRY # ON 'SIN'
2300  005616  062737  000002  001172          ADD     #2,$REG4
2301  005624  104014                          ERROR   14              ;AN IMPLIED SEEK WAS TRIED
2302                                                                  ;FROM 'CYLA' TO 'CYLB' (INDICATED
2303                                                                  ;IN EROR MESAGE), 'SIN' OCCURRED.
2304                                                                  ;2 TRIES ARE DONE BEFORE
2305                                                                  ;ABORTING
2306  005626  005737  001554                  TST     RETRY3          ;DONE RETRIES
2307  005632  001403                          BEQ     7$              ;YES, BRANCH
2308  005634  005237  001554                  INC     RETRY3          ;GO DO 2ND TRY
2309  005640  000722                          BR      6$
2310
2311  005642  005237  002002          7$:     INC     ERCNT1          ;ALLOW LESS THAN 12 ERORS OF THIS TYPE
2312  005646  001103                          BNE     19$             ;IF MORE SKIP THIS TEST
2313  005650  000137  006376                  JMP     EXT4            ;EXIT THIS TEST
2314
2315  005654  032777  010000  174070  8$:     BIT     #10000,@RKER    ;SKE?
2316  005662  001506                          BEQ     20$
2317  005664  004737  016170          15$:    JSR     PC,ERR1         ;GO GET 2 CYL NOS. BETWEEN WHICH
2318  005670  017737  174056  001166          MOV     @RKER,$REG2     ;IMPLIED SEEK WAS DONE
2319  005676  017737  174046  001170          MOV     @RKDS,$REG3
2320  005704  013737  001550  001172          MOV     RETRY1,$REG4    ;GET TRY # ON 'SKE'
2321  005712  062737  000003  001172          ADD     #3,$REG4
2322  005720  104417  002106                  TYPMSG  ,MSG2           ;GO PRINT 'SKE'
2323  005724  104014                          ERROR   14              ;IMPLIED SEEK WAS TRIED FROM
2324                                                                  ;'CYLA' TO 'CYLB' (INDICATED
2325                                                                  ;IN EROR MESAGE); 'SKE' OCCURRED.
```

```
2326
2327  005726  104414                          CON.RESET               ;2 TRIES ARE DONE.
2328                                                                  ;DO CONTROL RESET
2329  005730  004737  006334          9$:     JSR     PC,SBR2         ;GO READ 12 HEADERS FROM
2330                                                                  ;THIS CYLINDER & COMPARE
2331                                                                  ;THEM.  NOTE R5 CONTAINS THE
2332                                                                  ;DISK ADRES THAT WILL BE USED.
2333
2334  005734  012777  000015  174012      MOV     #15,@RKCS       ;GO DO DRIVE RESET
2335                                                                  ;WHILE THE DRIVE IS DOING RESET
2336                                                                  ;THE HDRS THAT WERE READ
2337                                                                  ;ABOVE ARE CHECKED, PRINTED
2338
2339  005742  005737  001776              TST     INDX3           ;WAS THERE A MISCOMPARISON
2340                                                                  ;IN ANY HEADER?
2341  005746  001006                      BNE     10$             ;IF INDX3>0, THERE WAS.
2342                                                                  ;NO, THERE WASN'T. HDRS OK
2343  005750  005237  001550              INC     RETRY1          ;ONLY 2 TRIES FOR SKE
2344  005754  001414                      BEQ     12$             ;BRANCH IF THIS WAS A 2ND TRY
2345  005756  104417  002155              TYPMSG  ,MSG5           ;TYPE OUT THAT HDRS WERE READ
2346                                                                  ;CORRECTLY.  THIS WAS TRY # 1
2347
2348  005762  000405                      BR      11$
2349
2350                                                                  ;HDRS WERE READ INCORRECT.
2351  005764  005237  001550          10$:     INC     RETRY1          ;ALLOW 2 TRIES FOR SKE
2352  005770  001411                      BEQ     14$             ;BRANCH, IF THIS WAS 2ND TRY
2353
2354                                                                  ;THERE WAS SKE ON DOING IMPLIED
2355  005772  104416  000015              MESAGE  ,15             ;SEEK TO 'CYL B'. THEN HDRS WERE
2356                                                                  ;READ FROM CYL B, WRONG HDRS
2357                                                                  ;RECVD
2358
2359  005776  104422          11$:        RESDON                  ;WAIT FOR PREVIOUS DRIVE RESET
2360                                                                  ;TO BE DONE
2361  006000  004737  006310              JSR     PC,SBR1         ;GO, REPOSITION HEADS
2362  006004  000634                      BR      4$
2363
2364                                                                  ;2ND TRY, SKE THIS TIME ALSO. BUT
2365                                                                  ;READ HDRS CORRECTLY FROM
2366                                                                  ;CYLINDER THAT GAVE SKE
2367                                                                  ;NOTE THIS WAS THE 2ND TRY
2368  006006  104417  002155          12$:    TYPMSG  ,MSG5           ;TYPE OUT THAT HDRS WERE
2369                                                                  ;READ CORRECTLY.
2370
2371  006012  000402                      BR      16$
2372
2373                                                                  ;2ND TRY, SKE THIS TIME ALSO.
2374  006014  104416  000015          14$:    MESAGE  ,15             ;READ HDRS FROM CYL THAT
2375                                                                  ;GAVE SKE, THEY WERE INCORRECT.
2376
2377  006020  104422          16$:        RESDON                  ;WAIT FOR PREVIOUS DRIVE RESET
2378                                                                  ;TO BE DONE
2379
2380  006022  005237  002004              INC     ERCNT2          ;ALLOW ONLY LESS THAN 10 ERRORS OF
2381                                                                  ;THIS TYPE (SKE)
```

```
2382  006026  001002                      BNE     17$
2383  006030  000137  006376              JMP     EXT4            ;EXIT THIS TEST IF MORE
2384
2385  006034  005704              17$:    TST     R4              ;WENT LAST TIME IN OR OUT?
2386  006036  001007                      BNE     19$             ;OUT
2387                                                              ;IN
2388  006040  005703                      TST     R3              ;WERE HDRS CORRECT?
2389  006042  001005                      BNE     19$             ;NO
2390                                                              ;YES
2391
2392  006044  005204              18$:    INC     R4
2393  006046  004737  006310              JSR     PC,SBR1         ;GO POSITION HEADS BAK ON INNER
2394                                                              ;CYL
2395  006052  000137  005440              JMP     1$              ;GO BAK & SEEK OUT NOW
2396
2397  006056  005237  001774      19$:    INC     INDX2           ;ALL SEEK PATTERNS DONE?
2398  006062  001547                      BEQ     TST5            ;;EXIT
2399
2400  006064  162737  000040  001556      SUB     #40,INADR       ;SET ADDRESS FOR THE NXT
2401                                                              ;INNER CYLINDER
2402  006072  005004                      CLR     R4              ;INDICATE THAT NOW SEEK IS GOING
2403                                                              ;TO BE IN
2404  006074  000137  005440              JMP     1$              ;GO BAK & SEEK IN TO 'INADR'
2405
2406  006100              20$:                                    ;IF THERE WAS NO SIN OR SKE
2407                                                              ;ENTER HERE
2408
2409  006100  012737  005462  001110      MOV     #3$,$LPERR      ;SET RETURN ADRES FOR LUPING
2410                                                              ;ON ERROR
2411  006106  020537  025620              CMP     R5,OUTBUF       ;CORRECT PSUEDO-HEADER READ?
2412  006112  001471                      BEQ     24$             ;YES, BRANCH
2413  006114  013737  001552  001172      MOV     RETRY2,$REG4    ;GET TRY #
2414  006122  062737  000003  001172      ADD     #3,$REG4
2415  006130  004737  016170              JSR     PC,ERR1         ;GO GET CYL #'S BETW'N
2416                                                              ;WHICH IMPLIED SEEK (READ)
2417                                                              ;WAS DONE
2418  006134  010537  001166              MOV     R5,$REG2        ;GET EXPCTD PSUEDO-HDR
2419  006140  013737  025620  001170      MOV     OUTBUF,$REG3    ;GET PSUEDO-HDR RECVD
2420  006146  104016                      ERROR   16              ;IMPLIED SEEK FROM CYLA TO CYLB WAS DONE.
2421                                                              ;READ PSEUDO-HEADER OF SEC 0,
2422                                                              ;CYLB (IN EROR MESAGE), BUT
2423                                                              ;THE WRONG PSEUDO-HEADER WAS
2424                                                              ;RECEIVED
2425  006150  005237  001552              INC     RETRY2
2426  006154  001402                      BEQ     21$
2427  006156  000137  005470              JMP     13$
2428
2429
2430  006162  004737  006334      21$:    JSR     PC,SBR2         ;GO READ HEADERS (12) FROM
2431                                                              ;THIS CYLINDER, & CHECK THEM.
2432                                                              ;IF MISCOMPARISON INDX3 WILL
2433                                                              ;BE > 0.
2434  006166  005737  001776              TST     INDX3
2435  006172  001003                      BNE     22$
2436  006174  104417  002155              TYPMSG  ,MSG5           ;WRONG PSUEDO-HDR WAS READ
2437                                                              ;BUT WHEN HDRS WERE READ
```

```
2438
2439  006200  000402                          BR      23$             ;FROM THE SAME CYLINDER, THEY
2440                                                                  ;WERE CORRECT
2441  006202  104416  000015          22$:    MESAGE  ,15             ;WRONG PSUEDO-HDR WAS READ
2442                                                                  ;FROM 'CYLB' (IN ERROR MESAGE).
2443                                                                  ;THEN HEADERS WERE READ FROM THE
2444                                                                  ;SAME CYLINDER.  THEY WERE ALSO
2445                                                                  ;WRONG.
2446  006206  010500                  23$:    MOV     R5,R0           ;NOW READ THE PSUEDO-HEADER
2447  006210  005200                          INC     R0              ;FROM THE NEXT SECTOR (1)
2448  006212  010077  173544                  MOV     R0,@RKDA        ;SAME CYLINDER
2449  006216  012777  025620  173534          MOV     #OUTBUF,@RKBA
2450  006224  012777  177777  173524          MOV     #-1,@RKWC
2451  006232  012777  000005  173514          MOV     #5,@RKCS
2452  006240  104420                          CON.RDY
2453  006242  010537  001162                  MOV     R5,$REG0
2454  006246  004737  016226                  JSR     PC,GCYL         ;GO GET CYL # & STORE IT IN $REG0
2455  006252  010037  001164                  MOV     R0,$REG1        ;GET EXPCT PSUEDO-HDR FROM SEC 1
2456  006256  013737  025620  001166          MOV     OUTBUF,$REG2
2457  006264  104416  000017                  MESAGE  ,17             ;PSUEDO-HEADER FROM SEC 1, CYLB
2458                                                                  ;(IN MESAGE) WAS READ.  THE EXPCTD
2459                                                                  ;& RECVD DATA WORDS ARE REPORTED.
2460  006270  005237  002006                  INC     ERCNT3          ;ALLOW ONLY LESS THAN 10 ERRORS
2461                                                                  ;OF THIS TYPE (WRONG PS-HDRS)
2462  006274  001440                          BEQ     EXT4
2463
2464  006276  005704                  24$:    TST     R4              ;SEEKED IN OR OUT LAST TIME?
2465  006300  001266                          BNE     19$             ;IF OUT, GO SEEK NXT INNER CYL
2466                                                                  ;IF IN, GO SEEK BAK TO 0
2467  006302  005204                          INC     R4              ;INDICATE THAT SEEK OUT (0)
2468  006304  000137  005440                  JMP     1$              ;WILL BE DONE NOW
2469
2470
2471                                                          ;THIS ROUTINE IS USED IN THIS TEST ONLY.
2472                                                          ;R4=0 INDICATES SEEK BEING DONE FROM
2473                                                          ;CYL 0 TO INNER CYL.
2474                                                          ;R4=1 INDICATES SEEK BEING DONE FROM
2475                                                          ;INNER CYL TO 0. THIS ROUTINE POSITIONS
2476                                                          ;THE HEADS ON 'INADR' CYL IF R4=1
2477
2478  006310  005704                  SBR1:   TST     R4
2479  006312  001407                          BEQ     1$
2480  006314  013777  001556  173440          MOV     INADR,@RKDA
2481  006322  012777  000011  173424          MOV     #11,@RKCS
2482  006330  104421                          TST.RWS
2483  006332  000207                  1$:     RTS     PC
2484
2485                                                          ;THIS ROUTINE IS USED IN THIS TEST
2486                                                          ;ONLY. IT READS 12 HEADERS FROM CYLINDER
2487                                                          ;WHOSE ADRES IS IN R5. THEN IT CHECKS
2488                                                          ;IF THE EXPECTED HEADER IS RECEIVED.
2489                                                          ;IF IT IS NOT, INDX3 IS INCREMENTED INDICATING
2490                                                          ;THE ERROR
2491
2492  006334  012700  177764          SBR2:   MOV     #-14,R0
2493  006340  012701  025620                  MOV     #OUTBUF,R1
```

```
2494  006344  010077  173406                  MOV     R0,@RKWC        ;READ 12 HDRS
2495  006350  010177  173404                  MOV     R1,@RKBA,       ;INTO THIS ADRES
2496  006354  010577  173402                  MOV     R5,@RKDA        ;FROM THIS CYLINDER
2497  006360  012777  002005  173366          MOV     #2005,@RKCS     ;RD FMT, GO
2498  006366  104420                          CON.RDY
2499
2500  006370  004737  007042                  JSR     PC,CHKHDRS      ;GO CHECK IF CORRECT HEADERS WERE READ
2501
2502  006374  000207                          RTS     PC              ;EXIT
2503
2504  006376  004737  016564          EXT4:   JSR     PC,ABRT
2505
2506                                  ;;*****************************************************************
2507                                  ;*TEST 5         PERFORM CONVERGING-DIVERGING (IMPLIED) SEEKS
2508                                          ;*THIS TEST PERFORMS A CONVERGING-DIVERGING SEEK PATTERN
2509                                          ;*USING IMPLIED SEEK (READ FORMAT).  THE SEEK SEQUENCE IS:
2510                                          ;*0-312-1-311-2-310-3-307-------310-2-311-1-312
2511                                          ;ALL READ FORMATS ARE DONE FROM SURFACE 0.
2512                                          ;*THE CYLINDER ADDRESSES, BETWEEN WHICH THE IMPLIED SEEK IS
2513                                          ;*PERFRORMED, ARE CONTAINED IN 'OUTADR' & 'INADR'.  IF 'SIN' OCCURS
2514                                          ;*AN ERROR IS REPORTED AND A RETRY IS DONE.  ON READING INCORRECT
2515                                          ;*HEADERS AN ERROR IS REPORTED AND A RETRY IS DONE.  NOTE THAT IF
2516                                          ;*ALL THE 12 HEADERS ARE INCORRECT, IT COULD MEAN THAT THE HEADS
2517                                          ;*COULD NOT POSITION CORRECTLY.  THIS WOULD BE CONFIRMED IF IN
2518                                          ;*PREVIOUS TESTS BAD HEADERS WERE NOT RECIEVED FROM THE SAME
2519                                          ;*CYLINDER.  IF THAT CYLINDER GAVE BAD HEADERS IN ALL THE PREVIOUS
2520                                          ;*TESTS THE PROBLEM COULD BE DIFFERENT.
2521                                          ;*MAXIMUM 12 ERRORS OF ANY KIND ARE ALLOWED.
2522                                          ;*IF MORE THAN 12 ERRORS OCCUR THE TEST IS ABORTED.
2523                                  ;;*****************************************************************
2524  006402  000004                  TST5:   SCOPE
2525  006404  104414                          CON.RESET               ;GO,DO CONTROL RESET
2526  006406  104415                          DRV.RESET               ;GO,DO DRIVE RESET
2527
2528  006410  005004                          CLR     R4              ;(R4)=0 SEEKING FROM 'OUTADR' TO 'INADR'
2529                                                                   ;(R4)=1 SEEKING FROM 'INADR' TO 'OUTADR'
2530
2531  006412  012737  177466  001774          MOV     #-312,INDX2     ;SET COUNT FOR DOING 312 TIMES
2532  006420  012700  177764                  MOV     #-14,R0
2533  006424  010037  002002                  MOV     R0,ERCNT1       ;ALLOW ONLY 12 ERRORS
2534  006430  010037  002004                  MOV     R0,ERCNT2
2535
2536  006434  005037  001560                  CLR     OUTADR          ;INITIALIZE 'OUTADR' TO 0
2537  006440  012737  014500  001556          MOV     #14500,INADR    ;INITIALIZE 'INADR' TO 312
2538
2539  006446  005704                  1$:     TST     R4              ;GOING IN OR OUT?
2540  006450  001005                          BNE     2$              ;GOING OUT,BRANCH
2541  006452  013705  001556                  MOV     INADR,R5        ;SET CYL ADRES BITS FOR GOING IN
2542  006456  053705  001526                  BIS     DRIVAD,R5       ;FORM DISK ADRES FOR INNER CYLINDER
2543  006462  000404                          BR      3$
2544
2545  006464  013705  001560          2$:     MOV     OUTADR,R5       ;SET CYL ADRES BITS FOR GOING OUT
2546  006470  053705  001526                  BIS     DRIVAD,R5       ;FORM DISK ADRES FOR GOING OUT
2547
2548  006474  005037  001552          3$:     CLR     RETRY2          ;ALLOW 2 RETRIES
2549  006500  012737  177777  001550  4$:     MOV     #-1,RETRY1      ;WHEN ERRORS OCCUR
```

```
2550  006506  000404                        BR      7$
2551  006510  104414              5$:       CON.RESET
2552  006512  104415                        DRV.RESET
2553  006514  004737  007006                JSR     PC,SBR3         ;GO REPOSITION HEADS
2554
2555  006520  012777  177764  173230  7$:   MOV     #-14,@RKWC      ;READ ALL HDRS FROM THIS CYLINDER
2556  006526  010577  173230                MOV     R5,@RKDA        ;FROM THIS CYL, SEC 0
2557  006532  012777  025620  173220        MOV     #OUTBUF,@RKBA   ;INTO THIS BUS ADRES
2558  006540  012737  006510  001110        MOV     #5$,$LPERR      ;SET RETURN ADRES FOR LOOPING ON ERROR
2559
2560  006546  012777  002005  173200        MOV     #2005,@RKCS     ;READ FORMAT,GO
2561
2562  006554  104420                        CON.RDY                 ;WAIT FOR CONTRL RDY
2563
2564  006556  032777  001000  173164        BIT     #1000,@RKDS     ;SIN?
2565  006564  001443                        BEQ     8$              ;NO, BRANCH
2566  006566  017737  173160  001166        MOV     @RKER,$REG2     ;SAVE RKER
2567  006574  017737  173150  001170        MOV     @RKDS,$REG3     ;SAVE RKDS
2568  006602  013737  001550  001172        MOV     RETRY1,$REG4    ;GET RETRY #
2569  006610  062737  000002  001172        ADD     #2,$REG4
2570  006616  004737  016112                JSR     PC,ERR2         ;GET CYL #'S BELOW 'N WHICH
2571                                                                ;SEEK WAS TRIED
2572  006622  104417  002100                TYPMSG  ,MSG1           ;TYPE 'SIN'
2573  006626  104014                        ERROR   14
2574                                                                ;'SIN' OCCURRED ON DOING IMPLIED
2575                                                                ;SEEK FROM 'CYLA' TO 'CYLB' (IN
2576                                                                ;EROR MESAGE).
2577  006630  005737  001550                TST     RETRY1          ;DONE 2 TRIES?
2578  006634  001403                        BEQ     6$              ;YES, BRANCH
2579  006636  005237  001550                INC     RETRY1          ;NO, RETRY
2580  006642  000722                        BR      5$
2581  006644  104414              6$:       CON.RESET
2582  006646  104415                        DRV.RESET
2583
2584
2585  006650  005237  002002                INC     ERCNT1          ;ALLOW LESS THAN 12 ERORS OF THE
2586  006654  001527                        BEQ     EXT5            ;ABOVE KIND
2587                                                                ;IF MORE SKIP THIS TEST
2588                                                                ;SIN OCCURED WHEN GOING TO CYL (IN
2589                                                                ;R5).  A DRVE RESET HAS BEEN DONE,
2590                                                                ;NOW TRY POSITIONING HEADS ON
2591                                                                ;THAT CYL.
2592  006656  010577  173100                MOV     R5,@RKDA        ;SET CYL ADRES
2593  006662  012777  000011  173064        MOV     #11,@RKCS       ;SEEK,GO
2594  006670  104421                        TST.RWS
2595  006672  000424                        BR      11$
2596                                                                ;IF NO SIN, ENTER HERE
2597  006674  004737  007042        8$:     JSR     PC,CHKHDRS      ;GO CHECK IF CORRECT HEADERS WERE READ
2598
2599  006700  012737  006500  001110        MOV     #4$,$LPERR      ;SET LUP ADDRESS
2600  006706  005737  001776                TST     INDX3           ;WAS THERE A BAD HDR?
2601  006712  001414                        BEQ     11$             ;NO, BRANCH
2602  006714  104020                        ERROR   20              ;WRONG HEADERS WERE READ FROM
2603                                                                ;'SEC #'S, ON DOING AN IMPLIED
2604                                                                ;SEEK (RDFMT) FROM 'CYLA' TO 'CYLB'
2605  006716  005237  001552                INC     RETRY2          ;ALLOW 2 TRIES
```

```
2606  006722  022737  000002  001552          CMP     #2,RETRY2
2607  006730  001263                          BNE     4$              ;GO TRY 2ND TIME
2608  006732  005237  002004                  INC     ERCNT2          ;ALLOW ONLY 12 ERRORS
2609  006736  001002                          BNE     11$
2610  006740  000137  007134                  JMP     EXT5            ;IF MORE, EXIT THIS TEST
2611
2612  006744  005704                  11$:    TST     R4              ;GOING WHICH WAY?
2613  006746  001006                          BNE     12$             ;'INADR' TO 'OUTADR', BRANCH
2614                                                                  ;'OUTADR' TO 'INADR'
2615  006750  005204                          INC     R4              ;INDICATE THAT NXT TIME GOING
2616                                                                  ;FROM 'INADR' TO 'OUTADR'
2617  006752  062737  000040  001560          ADD     #40,OUTADR      ;INCREMENT CYCLINDER ADRES
2618  006760  000137  006446                  JMP     1$              ;GO BAK & DO IMPLIED SEEK
2619                                                                  ;FROM 'INADR' TO 'OUTADR'
2620
2621  006764  005004                  12$:    CLR     R4              ;INDICATE THAT NXT TIME GOING
2622                                                                  ;FROM 'OUTADR' TO 'INADR'
2623  006766  162737  000040  001556          SUB     #40,INADR       ;DECREMENT CYLINDER ADRES
2624  006774  005237  001774                  INC     INDX2           ;DONE ALL 312 FORWARD-BACKWARD
2625                                                                  ;SEEK PATTERNS
2626  007000  001457                          BEQ     TST6            ;;IF YES, EXIT
2627
2628  007002  000137  006446                  JMP     1$              ;IF NOT, GO BAK & DO IMPLIED
2629                                                                  ;SEEK FROM 'OUTADR' TO 'INADR'
2630
2631                          ;THIS SUBROUTINE IS ENTERED AFTER A 'SIN' ERROR OCCURED ON GOING FROM
2632                          ;'CYLA' TO 'CYLB' AS INDICATED IN THE ERROR MESSAGE. BEFORE RETRYING THE
2633                          ;HEADS HAVE TO BE POSITIONED BACK TO 'CYLA'. NOTE THAT A DRIVE RESET
2634                          ;WAS DONE TO CLEAR SIN.
2635                          ;R4=0, INDICATES SEEK IS BEING DONE FROM 'OUTADR' CYLINDER TO 'INADR'CYLINDER.
2636                          ;R4=1, INDICATES THAT SEEK IS BEING DONE FROM 'INADR' TO 'OUTADR'.
2637  007006  005704          SBR3:   TST     R4              ;GOING WHICH WAY?
2638  007010  001404                  BEQ     1$              ;IF FROM 'OUTADR' TO 'INADR', BRANCH.
2639  007012  013777  001556  172742          MOV     INADR,@RKDA
2640  007020  000403                  BR      2$
2641  007022  013777  001560  172732  1$:     MOV     OUTADR,@RKDA
2642  007030  012777  000011  172716  2$:     MOV     #11,@RKCS
2643  007036  104421                  TST.RWS
2644  007040  000207                  RTS     PC
2645
2646
2647                          ;CHKHDRS
2648                          ;THIS ROUTINE CHECKS THAT THE HEADERS READ PREVIOUSLY WERE CORRECT.
2649                          ;IF NOT, THE BAD HEADERS AND THE SECTOR #'S FROM WHICH THEY WERE READ
2650                          ;ARE STORED.
2651                          ;ON ENTRY:
2652                          ;R5 CONTAINS DISK ADDRESS FROM WHERE HEADERS WERE READ.
2653                          ;OUTBUF - 12 HEADRERS READ PREVIOUSLY ARE STORED STARTING 'OUTBUF'.
2654                          ;ON EXIT:
2655                          ;INDX3=0, IF THE HEADERS WERE CORRECT
2656                          ;INDX3=1, IF THE HEADERS WERE INCORRECT
2657                          ;BUFR - SECTOR #'S GIVING BAD HEADRERS ARE STORED STARTING AT 'BUFR'.
2658                          ;BUFR1 - BAD HEADERS FOR THE ABOVE SECTORS ARE STORED STARTING 'BUFR1'.
2659                          ;THE BAD SECTOR TABLE IS TERMINATED BY A '177777' WORD.
2660
2661  007042  005000          CHKHDRS: CLR    R0              ;INITIALIZE FOR 14 HDRS
```

```
 2662  007044  012701  025620                    MOV     #OUTBUF,R1      ;INITIZLIZE PTR TO HDRS RECVD
 2663  007050  012702  001564                    MOV     #BUFR,R2        ;INITIALIZE PTR TO SECTOR TABLE
 2664  007054  012703  001616                    MOV     #BUFR1,R3       ;INITIALIZE PTR TO BAD HDR TABLE
 2665  007060  010537  001120                    MOV     R5,$GDADR
 2666  007064  042737  160037  001120            BIC     #160037,$GDADR  ;GET EXPCTD HEADER
 2667  007072  005037  001776                    CLR     INDX3           ;CLR FLG INDICATING BAD HDRS
 2668
 2669  007076  023711  001120          9$:       CMP     $GDADR,(R1)     ;HEADER OK?
 2670  007102  001406                            BEQ     10$             ;YES,BRANCH
 2671  007104  011123                            MOV     (R1),(R3)+      ;SAVE BAD HDR
 2672  007106  010022                            MOV     R0,(R2)+        ;SAVE BAD SECTR #
 2673
 2674  007110  012712  177777                    MOV     #177777,(R2)    ;PUT TERMINATR ON SECTR TABLE
 2675  007114  005237  001776                    INC     INDX3           ;SET FLG INDICATING BAD HDR
 2676  007120  005721                    10$:    TST     (R1)+           ;INCRMNT PTR TO NXT HDR
 2677  007122  005200                            INC     R0              ;ALL HDRS CHKD?
 2678  007124  022700  000014                    CMP     #14,R0
 2679  007130  001362                            BNE     9$              ;IF NOT, LUP BAK
 2680  007132  000207                            RTS     PC
 2681
 2682  007134  004737  016564          EXT5:     JSR     PC,ABRT
 2683
 2684                                    ;;*******************************************************************
 2685                                    ;*TEST 6         WRITE PATTERNS ON THE DISK
 2686                                    ;*IN THIS TEST DIFFERENT PATTERNS ARE WRITTEN ON THE ENTIRE DISK.  768
 2687                                    ;*WORDS (3 SECTORS) ARE WRITTEN AT A TIME.  TWO 768 WORDS LONG
 2688                                    ;*BUFFERS HAVE BEEN USED IN THIS TEST.  WHILE ONE BUFFER IS
 2689                                    ;*USED TO WRITE ON THE DISK, THE OTHER BUFFER GETS FILLED UP WITH
 2690                                    ;*PATTERNS TO BE USED NEXT!  THIS OVERLAPPING IS DONE TO SAVE TIME.
 2691
 2692                                    ;*THREE PATTERN-GENERATOR SUB-ROUTINES ARE USED:
 2693                                    ;*1.  PTGEN0    2.  PTGEN1      3.  PTGEN2      4.  PTGEN3
 2694                                    ;*THESE THREE ROUTINES ARE CALLED IN A CYCLIC ORDER:
 2695                                    ;*      PTGEN0-PTGEN1-PTGEN2-PTGEN3-PTGEN0-PTGEN1.....
 2696                                    ;*THE FOLLOWING ORDER IS MAINTAINED IN WRITING BLOCKS (EACH
 2697                                    ;*3 SECTORS LONG) USING ONE OF THE FOUR PATTERN GENERATORS:
 2698
 2699                                    ;*CYL   SECTORS       CYL   SECTORS       CYL   SECTORS       CYL   SECTORS
 2700                                    ;*    SUR       ROUTINE   SUR       ROUTINE   SUR       ROUTINE   SUR       ROUTI
 2701                                    ;* 0  0   0-2  PTGEN0  0  0  6-10  PTGEN1  0  0   3-5  PTGEN2  0  0 11-13 PTGEN3
 2702                                    ;* 0  1   0-2  PTGEN0  0  1  6-10  PTGEN1  0  1   3-5  PTGEN2  0  1 11-13 PTGEN3
 2703                                    ;* 1  0   0-2  PTGEN0  1  0  6-10  PTGEN1  1  0   3-5  PTGEN2  1  0 11-13 PTGEN3
 2704                                    ;* 1  1   0-2  PTGEN0  1  1  6-10  PTGEN1  1  1   3-5  PTGEN2  1  1 11-13 PTGEN3
 2705                                    ;* 2  0   0-2  PTGEN0  2  0  6-10  PTGEN1  2  0   3-5  PTGEN2  2  0 11-13 PTGEN3
 2706                                    ;*ETC, ETC.....
 2707                                    ;*THE ABOVE SHOWN STAGGERING (OF SECTORS TO BE WRITTEN) IS DONE TO
 2708                                    ;*SAVE ONE REV (40MS) EVERY TIME A BLOCK (3 SECTORS) IS WRITTEN.
 2709                                    ;*IF YOU WANT TO USE BUFFERS STARTING AT SOME OTHER MEMORY ADDRESS
 2710                                    ;*MAKE THE FOLLOWING CHANGES:
 2711                                    ;*CHANGE 'PBUF0' TO STARTING ADDRESS OF THE FIRST 768 WORDS LONG BUFFER.
 2712                                    ;*CHANGE 'PBUF1' TO STARTING ADDRESS OF SECOND 768 WORDS LONG BUFFER.
 2713
 2714                                    ;*IF YOU WANT TO WRITE YOUR OWN PATTERNS USING PATTERN GENERATOR 'PTGEN0'
 2715                                    ;*CHANGE 'PTRN01' AND 'PTRN02' TO THE PATTERNS YOU WANT.
 2716
 2717                                    ;*TO WRITE THE SAME TWO (OR ONE) PATTERNS ON THE ENTIRE DISK CHANGE
```

```
                                        ;*LOCATION 'PAT1' TO 'PTGEN0' THE STARTING ADDRESS OF PAT-GENERATOR 0
                                        ;*LOCATION 'PAT2' TO 'PTGEN0' THE STARTING ADDRES OF PAT-GENERATOR 0
                                        ;*LOCATION 'PAT3' TO 'PTGEN0' THE STARTING ADDRESS OF PAT-GENERATOR 0
                                    ;;**********************************************************************
007140  000004                TST6:   SCOPE
007142  012737  177764  002002          MOV     #-14,ERCNT1
007150  012737  177764  002004          MOV     #-14,ERCNT2
007156  012737  177152  001772          MOV     #-626,INDX1     ;SET COUNT FOR 313X2 TRACKS
007164  005037  001706                  CLR     BUFLG0          ;CLR FLAG FOR BUFR 0
007170  005037  001710                  CLR     BUFLG1          ;CLR FLAG FOR BUFR 1
                                                                ;BIT 7 OF ABOVE FLAGS WHEN SET
                                                                ;INDICATES, THAT BUFR TO BE USED
                                                                ;FOR WRITING ON DSK
007174  013737  001526  001730          MOV     DRIVAD,DSKADR   ;GET DRIVE #, DISK ADRES
007202  012737  001712  001722          MOV     #PAT0,PRSPAT    ;INITIALIZE PTR TO THE FIRST
                                                                ;PATRN GENERATOR
007210  004737  007626                  JSR     PC,GETBUF
007214  017737  172502  001726          MOV     @PRSPAT,PGSUBR
007222  004777  172500                  JSR     PC,@PGSUBR      ;GO GENERATE PATRNS FOR
                                                                ;3 SECTOR (400X3)
007226  005005                  1$:     CLR     R5              ;INITIALIZE COUNT FOR 4 BLOCKS
007230  005737  001706          2$:     TST     BUFLG0          ;FIND OUT WHICH BUFR TO USE
007234  100407                          BMI     3$              ;FOR WRITING ON DSK
007236  013737  001704  001732          MOV     PBUF1,BUSADR    ;USE 'IOBUF1' FOR TRANSFER
                                                                ;OR THE ONE INDICATED BY THE USER
007244  052737  000200  001710          BIS     #BIT7,BUFLG1    ;SET FLAG TO INDICATE THAT
                                                                ;WRTING ON DSK WILL B DONE FROM
                                                                ;THIS BUFR (BUF 1)
007252  000406                          BR      13$

007254  013737  001702  001732  3$:     MOV     PBUF0,BUSADR    ;USE 'IOBUF0' FOR TRANSFER
                                                                ;OR THE ONE INDICATED BY THE USER

007262  052737  000200  001706          BIS     #BIT7,BUFLG0    ;INDICATE THAT 'IOBUF0' WILL
                                                                ;B USED FOR WRTING ON DISK
007270  012737  007276  001110  13$:     MOV     #4$,$LPERR
007276  013777  001730  172456  4$:     MOV     DSKADR,@RKDA    ;SET RKDA
007304  013777  001732  172446          MOV     BUSADR,@RKBA
007312  012777  176400  172436          MOV     #-1400,@RKWC
007320  012777  000003  172426          MOV     #3,@RKCS        ;WRITE THE 4 SECTORS ON
                                                                ;DISK
007326  013737  001722  001724          MOV     PRSPAT,NXTPAT   ;WHILE THE PATRNS R BEING WRITTEN
007334  023727  001722  001720          CMP     PRSPAT,#PAT3    ;GO GENERATE THE NXT PATRNS
007342  001004                          BNE     5$              ;TO B WRITTEN
007344  012737  001712  001724          MOV     #PAT0,NXTPAT    ;KEEP GENERATING PATRNS IN THIS
007352  000403                          BR      6$              ;WAY "PAT0"-"PAT1"-"PAT2"-"PAT3"-"PAT0"-
007354  062737  000002  001724  5$:     ADD     #2,NXTPAT
007362  004737  007626          6$:     JSR     PC,GETBUF
007366  017737  172332  001726          MOV     @NXTPAT,PGSUBR
007374  004777  172326                  JSR     PC,@PGSUBR      ;GO GENERATE THESE PATRNS.
                                                                ;(3 X 400) WORDS
007400  104420                          CON.RDY
007402  032777  140000  172344          BIT     #140000,@RKCS   ;ANY ERROR?
007410  001411                          BEQ     12$             ;GET RKCS,ER,DS,DA
007412  004737  015754                  JSR     PC,GT4RG
007416  104021                          ERROR   21              ;ERROR ON DOING WRITE
```

```
2774  007420  104414                      CON.RESET                ;CLEAR IT
2775  007422  005237  002002              INC     ERCNT1           ;ALLOW 12 ERRORS AT MOST
2776  007426  001002                      BNE     12$
2777  007430  000137  010214              JMP     EXT6             ;IF MORE, EXIT
2778  007434  032777  001000  172306  12$: BIT     #BIT9,@RKDS      ;SIN, ON DOING WRITE?
2779  007442  001412                      BEQ     7$
2780  007444  004737  015754              JSR     PC,GT4RG
2781  007450  104022                      ERROR   22               ;SIN ERROR ON DOING WRITE
2782  007452  104414                      CON.RESET
2783  007454  104415                      DRV.RESET
2784  007456  005237  002004              INC     ERCNT2           ;ALLOW 12 ERRORS AT MOST
2785  007462  001002                      BNE     7$
2786  007464  000137  010214              JMP     EXT6
2787                                                               ;FIGURE OUT WHICH BUFFER IS
2788                                                               ;AVAILABLE FOR USE
2789  007470  105737  001706        7$:   TSTB    BUFLG0           ;WAS PREVIOUS DSK-WRITE DONE
2790  007474  100003                      BPL     8$               ;USING BUFR 0?
2791  007476  005037  001706              CLR     BUFLG0           ;YES, CLR FLAG INDICATING IT'S
2792                                                               ;AVAILABLE NOW
2793  007502  000402                      BR      9$
2794  007504  005037  001710        8$:   CLR     BUFLG1           ;CLR FLAG INDICATING BUFR1
2795                                                               ;IS AVAILABLE NOW
2796  007510  013737  001724  001722  9$:   MOV     NXTPAT,PRSPAT    ;'PRSPAT'S PATRNS WILL BE USED
2797                                                               ;ON NEXT WRITE
2798  007516  010500                      MOV     R5,R0            ;FORM SEC # TO BE USED NXT TIME
2799  007520  116000  010210              MOVB    SECPTR(R0),R0    ;GET SEC #
2800  007524  042737  000017  001730  10$: BIC     #17,DSKADR       ;MASK SECTOR BITS FROM DSK-ADRES
2801  007532  050037  001730              BIS     R0,DSKADR        ;FORM NXT DSK-ADRES
2802  007536  005205                      INC     R5               ;DONE WITH 12 SECTRS (3 BLOCKS)?
2803  007540  022705  000004              CMP     #4,R5
2804  007544  001231                      BNE     2$               ;ON THIS SURFACE? IF NOT, GO
2805                                                               ;DO NXT 4 SECTRS
2806  007546  032737  000001  001772      BIT     #BIT0,INDX1      ;WHICH SURFACE WAS DONE, 0 OR 1?
2807  007554  001415                      BEQ     11$              ;IF 0, GO DO SURFACE 1
2808  007556  005237  001772              INC     INDX1            ;COUNT # OF TRACKS
2809  007562  001002                      BNE     .+6
2810  007564  000137  010220              JMP     TST7             ;EXIT IF DONE
2811  007570  042737  000020  001730      BIC     #20,DSKADR       ;GO TO SUR 0, NEXT CYLINDER
2812  007576  062737  000040  001730      ADD     #40,DSKADR
2813  007604  000137  007226              JMP     1$               ;GO BACK AND DO WRITE
2814  007610  005237  001772        11$:  INC     INDX1            ;COUNT # OF TRACKS
2815  007614  052737  000020  001730      BIS     #20,DSKADR       ;SET BIT FOR SUR 1
2816  007622  000137  007226              JMP     1$               ;GO, WRITE PATRNS ON SURFACE 1
2817                                    ;*GET BUF#
2818                                    ;*THIS ROUTINE FINDS OUT  WHICH BUFFER (IOBUF0, IOBUF1) OR ONE OF
2819                                    ;*THE TWO BUFFERS SELECTED BY THE USER) TO USE
2820                                    ;*FOR GENERATING PATTERNS. THEN IT SETS UP A FLAG INDICATING THAT
2821                                    ;*BUFFER HAS TO BE FILLED UP. THE STARTING ADDRESS OF THE
2822                                    ;*BUFFER IS STORED IN 'BUF #'.
2823
2824  007626  005737  001706        GETBUF: TST     BUFLG0           ;BUFR 0 AVAILABLE FOR USE?
2825  007632  100007                      BPL     1$               ;YES, BRANCH
2826  007634  052737  100000  001710      BIS     #BIT15,BUFLG1    ;NO, USE BUFR 1. INDICATE SO.
2827  007642  013737  001704  001744      MOV     PBUF1,BUFNO      ;SAVE STARTING ADRES OF BUFR1
2828  007650  000207                      RTS     PC
2829  007652  052737  100000  001706  1$:  BIS     #BIT15,BUFLG0    ;INDICATED, USING BUF 0
```

```
2830  007660  013737  001702  001744          MOV     PBUF0,BUFNO     ;SAVE STARTING ADRES OF BUFR 0
2831  007666  000207                          RTS     PC              ;RETURN
2832                                  ;*PTGEN0
2833                                  ;*THIS ROUTINE GENERATES A 768 (3X256) WORDS LONG
2834                                  ;*BUFFER CONTAINING THE FOLLOWING PATTERNS
2835                                  ;*FIRST BLOCK OF 256 WORDS:     125252
2836                                  ;*SECOND BLOCK OF 256 WORDS:    052525 (COMPLEMENT OF ABOVE)
2837                                  ;*THIRD BLOCK OF 256 WORDS:     010421
2838                                  ;*YOU CAN USE ANY OTHER PATTERN/S (& ITS COMPLEMENT)
2839                                  ;*MAKING THE CHANGES IN THE 2 LOCATIONS SHOWN BELOW.
2840
2841  007670  013700  001744          PTGEN0: MOV     BUFNO,R0                ;GET STARTING ADRES OF BUFR
2842  007674  013701  007746                  MOV     PTRN01,R1       ;GET PATRN TO BE GENERATED
2843  007700  012702  177400                  MOV     #-400,R2        ;IN THE FIRST 400 WORD BLOCK
2844
2845  007704  010120                  1$:     MOV     R1,(R0)+        ;GENERATE THE FIRST BLOCK
2846  007706  005202                          INC     R2              ;WITH 'PAT01' PATRN
2847  007710  001375                          BNE     1$              ;ALL DONE?
2848
2849  007712  012702  177400                  MOV     #-400,R2
2850  007716  005101                          COM     R1              ;COMPLEMENT 'PAT01' PATAN
2851  007720  010120                  2$:     MOV     R1,(R0)+        ;GENERATE 2ND BLOCK WITH
2852  007722  005202                          INC     R2              ;'PAT01'S COMPLEMENT PATRN
2853  007724  001375                          BNE     2$              ;ALL DONE?
2854  007726  012702  177400                  MOV     #-400,R2
2855  007732  013701  007750                  MOV     PTRN02,R1       ;GET PATRN TO BE GENERATED
2856                                                                  ;FOR 3RD BLOCK
2857  007736  010120                  3$:     MOV     R1,(R0)+        ;GENERATE 3RD BLOCK USING
2858                                                                  ;'PAT02' PATRN
2859  007740  005202                          INC     R2
2860  007742  001375                          BNE     3$              ;ALL DONE?
2861
2862  007744  000207                          RTS     PC              ;RETURN
2863
2864  007746  125252          PTRN01: 125252                          ;CHANGE THESE LOCATIONS IF
2865  007750  010421          PTRN02: 010421                          ;YOU WANT ANY OTHER PATTERNS
2866
2867                                  ;*PTGEN1
2868                                  ;*THIS ROUTINE GENERATES A 768 (3X256) WORDS
2869                                  ;*LONG BUFFER CONTAINING THE FOLLOWING PATTERNS:
2870
2871                                  ;*FIRST BLOCK-256 WORDS 000001  FILL 1'S
2872                                  ;*                      000003
2873                                  ;*                      :
2874                                  ;*                      177777
2875
2876                                  ;*SECOND BLOCK          177776  FILL 0'S
2877                                  ;*                      177774
2878                                  ;*                      :
2879                                  ;*                      000000
2880
2881                                  ;*THIRD BLOCK           000001  FLOAT A 1
2882                                  ;*                      000020
2883                                  ;*                      :
2884                                  ;*                      100000
2885
```

```
                                        ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE
                                        ;*BUFFER.
                                        
007752  012703  000001          PTGEN1: MOV     #1,R3           ;INITIALIZE PATRNS
007756  012704  177776                  MOV     #177776,R4
007762  013700  001744                  MOV     BUFNO,R0                ;GET STARTING ADRES OF BUFR
007766  012702  177760                  MOV     #-20,R2         ;SET COUNT
007772  010301          1$:     MOV     R3,R1           ;INITIALIZE PATRN
007774  010120          2$:     MOV     R1,(R0)+        ;GENERATE THE FIRST
007776  000261                  SEC                     ;BLOCK USING "FILL 1'S"
010000  006101                  ROL     R1
010002  103374                  BCC     2$
010004  005202                  INC     R2
010006  001371                  BNE     1$              ;ALL DONE?

010010  012702  177760                  MOV     #-20,R2
010014  010401          3$:     MOV     R4,R1           ;INITIALIZE PATRN
010016  010120          4$:     MOV     R1,(R0)+        ;GENERATE 2ND BLOCK
010020  000241                  CLC                     ;USING "FILL 0'S"
010022  006101                  ROL     R1
010024  103774                  BCS     4$
010026  005202                  INC     R2
010030  001371                  BNE     3$              ;DONE?

010032  012702  177760                  MOV     #-20,R2         ;SET COUNT
010036  010301          5$:     MOV     R3,R1           ;INITIALIZE PATRN
010040  010120          6$:     MOV     R1,(R0)+        ;GENERATE THE 3RD BLOCK
010042  006301                  ASL     R1              ;USING "FLOAT A 1"
010044  103375                  BCC     6$
010046  005202                  INC     R2
010050  001372                  BNE     5$              ;DONE?
010052  000207                  RTS     PC              ;RETURN

                                        ;*PTGEN2
                                        ;*THIS ROUTINE GENERATES A 768 (3X256) WORDS LONG
                                        ;*BUFFER CONTAINING THE FOLLOWING PATTERNS:

                                        ;*FIRST BLOCK-256 WORDS 000000  COUNT PATRN-LOWER BYTE
                                        ;*                      000001  0-377
                                        ;*                      000002
                                        ;*                      .
                                        ;*                      000377

                                        ;*SECOND BLOCK          000000  COUNT PATRN-HIGHER BYTE
                                        ;*                      000400  0-377
                                        ;*                      001000
                                        ;*                      .
                                        ;*                      177400

                                        ;*THIRD BLOCK           000000  COUNT PATRN-HIGHER & LOWER BYTE
                                        ;*                      000401  0-377, 0-377
                                        ;*                      .
                                        ;*                      177777

                                        ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
```

```
2942  010054  005001                  PTGEN2: CLR     R1              ;INITIALIZE PATRN
2943
2944  010056  013700  001744                  MOV     BUFNO,R0
2945  010062  010120                  1$:     MOV     R1,(R0)+        ;GENERATE 1ST BLOCK USING
2946  010064  105201                          INCB    R1              ;USING 'COUNT UP LOWER BYTE'
2947  010066  001375                          BNE     1$              ;DONE?
2948
2949  010070  005001                          CLR     R1
2950  010072  010120                  2$:     MOV     R1,(R0)+        ;GENERATE 2ND BLOCK
2951  010074  062701  000400                  ADD     #400,R1         ;USING 'COUNT UP HIGHER BYTE'
2952  010100  103374                          BCC     2$              ;DONE?
2953  010102  005001                          CLR     R1
2954  010104  010120                  3$:     MOV     R1,(R0)+        ;GENERATE 3RD BLOCK USING
2955  010106  062701  000401                  ADD     #401,R1         ;'COUNT UP HIGHER & LOWER BYTE'
2956  010112  103374                          BCC     3$              ;ALL DONE?
2957  010114  000207                          RTS     PC              ;RETURN
2958
2959
2960                                  ;PTGEN3
2961                                  ;*THIS ROUTINE GENERATES A 768 (3X256) WORDS LONG
2962                                  ;*BUFFER CONTAINING THE FOLLOWING PATTERNS:
2963
2964                                  ;*FIRST BLOCK OF 256 WORDS:     167356 (COMPLEMENT OF 010421)
2965
2966                                  ;*SECOND BLOCK          177776  FLOAT A 0
2967                                  ;*                      177775
2968                                  ;*                      :
2969                                  ;*                      077777
2970
2971                                  ;*THIRD BLOCK           000377  COUNT UP HIGHER BYTE 0-377
2972                                  ;*                      000776  COUNT DOWN LOWER BYTE 377-0
2973                                  ;*                      :
2974                                  ;*                      177400
2975
2976
2977                                  ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2978  010116  013700  001744  PTGEN3:  MOV     BUFNO,R0
2979  010122  012702  177400          MOV     #-400,R2
2980  010126  013701  007750          MOV     PTRN02,R1       ;GET PATTERN
2981  010132  005101                  COM     R1              ;COMPLEMENT 'PAT02' PATRN
2982  010134  010120          4$:     MOV     R1,(R0)+        ;GENERATE 1ST BLOCK
2983  010136  005202                  INC     R2
2984  010140  001375                  BNE     4$              ;ALL DONE?
2985
2986                                                          ;2ND BLOCK
2987  010142  012702  177760          MOV     #-20,R2
2988  010146  000261          7$:     SEC
2989  010150  012701  177776          MOV     #177776,R1
2990  010154  010120          8$:     MOV     R1,(R0)+
2991  010156  006101                  ROL     R1
2992  010160  103775                  BCS     8$
2993  010162  005202                  INC     R2
2994  010164  001370                  BNE     7$              ;ALL DONE?
2995
2996                                                          ;3RD BLOCK
2997  010166  012701  000377          MOV     #377,R1
```

```
 2998  010172  010102                      MOV     R1,R2           ;GENERATE 3RD BLOCK USING
 2999  010174  010120              9$:     MOV     R1,(R0)+        ;'COUNT DOWN LOWER BYTE'
 3000  010176  060201                      ADD     R2,R1           ;'COUNT UP HIGHER BYTE'
 3001  010200  022701  177777              CMP     #-1,R1
 3002  010204  001373                      BNE     9$              ;ALL DONE?
 3003  010206  000207                      RTS     PC              ;RETURN
 3004
 3005                              ;SECTOR POINTER TABLE. DATA TRANSFERS ARE DONE IN BLOCKS (3 SECTORS
 3006                              ;EACH) IN THE CYCLIC ORDER:  0-2, 6-10, 3-5, 11-13, 0-2, AND SO ON.
 3007  010210      006             SECPTR: .BYTE   6
 3008  010211      003                     .BYTE   3
 3009  010212      011                     .BYTE   11
 3010  010213      000                     .BYTE   0
 3011
 3012  010214  004737  016564      EXT6:   JSR     PC,ABRT
 3013
 3014
 3015                              ;;*****************************************************************
 3016                              ;*TEST 7         READ, SOFTWARE COMPARE, WRITE CHECK OF THE PATTERNS
 3017
 3018                                      ;****TEST 6 (WRITING PATTERNS ON DISK) SHOULD BE DONE BEFORE DOING****
 3019                                      ;****THIS TEST.****
 3020
 3021                                      ;*IN THIS TEST THE PATTERNS THAT WERE WRITTEN BEFORE (IN THE
 3022                                      ;*PREVIOUS TEST) ARE READ BACK AND SOFTWARE COMPARED.
 3023                                      ;*WHILE THE SOFTWARE COMPARISON IS TAKING PLACE, AN OVERLAPPING
 3024                                      ;*'WRITE CHECK' IS DONE FOR THE SAME BLOCK.  THE READING
 3025                                      ;*BACK OF EACH BLOCK (4 SECTORS) IS DONE IN THE SAME
 3026                                      ;*MANNER AS DESCRIBED IN THE BEGINNING OF PREVIOUS TEST.
 3027                                      ;*OVERLAPPING OPERATIONS AND STAGGERING OF BLOCKS ARE DONE IN
 3028                                      ;*A SIMILIAR MANNER.
 3029                                      ;*THE FOLLOWING IS A DESCRIPTION OF HOW ERRORS ARE HANDLED.
 3030                                      ;*IF A 'SIN' OR 'HE' OCCURS IT IS REPORTED AND THAT BLOCK
 3031                                      ;*IS SKIPPED.  IN THIS STAGE OF TESTING THESE ERRORS SHOULD NOT
 3032                                      ;*NORMALLY OCCUR.
 3033                                      ;*IF A CHECKSUM ERROR OCCURS, CONTROL IS TRANSFERRED TO
 3034                                      ;*'CSERROR'.  FIRST THE CSE IS REPORTED.  THE SECTOR GIVING
 3035                                      ;*CSE IS READ AGAIN.  IN ALL 3 TRIES ARE DONE.  IF STILL
 3036                                      ;*THE ERROR PRESISTS THAT SECTOR IS ABORTED AND THE REST
 3037                                      ;*OF THE SECTORS ARE READ AND CHECKED FOR CSE.  IF
 3038                                      ;*AGAIN SOME OTHER SECTOR GIVES CSE, REPORTING AND RETRIES
 3039                                      ;*ARE DONE AGAIN.
 3040                                      ;*NEXT SOFTWARE COMPARISON IS DONE OF THE DATA THAT WAS
 3041                                      ;*READ BACK.  ON GETTING A DATA MISCOMPARISON
 3042                                      ;*THE RELEVANT INFO(GOOD DATA, BAD DATA, ADDRESS ETC.) IS
 3043                                      ;*STORED.  AFTER THE WHOLE BLOCK HAS BEEN CHECKED, THE DATA
 3044                                      ;*ERROR/S IS/ARE REPORTED.  THEN THE BLOCK IS READ AGAIN.  IN ALL
 3045                                      ;*THREE TRIES ARE DONE.
 3046                                      ;*WHILE THE DATA COMPARISON (SOFTWARE) IS GOING ON THE 'WRITE
 3047                                      ;*CHECK' IS ALSO IN PROGRESS.  IF A WRITE CHECK ERROR OCCURS THE
 3048                                      ;*CONTROL IS TRANSFERRED TO 'WCEROR'.  WRITE CHECK OF THE SECTOR
 3049                                      ;*THAT GAVE WCE IS DONE AGAIN.  IN ALL THREE TRIES ARE DONE.
 3050                                      ;*NOTE THAT EVERY WCE IS REPORTED. IF ALL THE 4 SECTOR OF
 3051                                      ;*A BLOCK GAVE WCE'S, ALL 4 WILL BE REPORTED AND RETRIED.
 3052
 3053                                      ;*DEPENDING ON THE NATURE OF THE PROBLEM, THE ABOVE ERRORS
```

```
                                        ;*CAN OCCUR IN ANY COMBINATION.  IT IS RECOMMENDED THAT ALL
                                        ;*THE ERROR MESSAGE BE CAREFULLY CO-RELATED AND EXAMINED.  IT
                                        ;*WILL PROVIDE A DEEP INSIGHT INTO THE PROBLEM.
                                    ;;**************************************************************
010220  000004                    TST7:   SCOPE
010222  012737  177764  002006            MOV     #-14,ERCNT3     ;ALLOW 12 ERRORS AT THE MOST
010230  012737  177773  002010            MOV     #-5,ERCNT4      ;ALLOW 5 ERRORS AT THE MOST
010236  012737  177742  002012            MOV     #-36,ERCNT5     ;ALLOW 10 ERRORS AT THE MOST
010244  012737  177742  002014            MOV     #-36,ERCNT6     ;ALLOW 10 ERRORS AT THE MOST
010252  012737  177764  002016            MOV     #-14,ERCNT7     ;ALLOW 12 ERRORS AT THE MOST
010260  012737  177742  002020            MOV     #-36,ERCNT8     ;ALLOW 10 ERRORS AT THE MOST
010266  012737  177152  001772            MOV     #-626,INDX1     ;SET COUNT FOR 626 TRACKS
010274  005037  001774                    CLR     INDX2           ;CLR COUNT FOR 4 BLOCKS ON A TRACK

010300  012737  001712  001722            MOV     #PAT0,PRSPAT    ;INITLZE PTR TO PATRN GENRTR
010306  013737  001526  001746            MOV     DRIVAD,ADRES    ;INITLZE DRV#,ADRES

010314  005037  002002            BEGIN:  CLR     ERCNT1          ;IF > 0, MEANS THAT RETRIES
010320  005037  002004                    CLR     ERCNT2          ;DONE AFTER CSE OR CSE CHKD
010324  012737  177775  001550            MOV     #-3,RETRY1      ;RETRY COUNT FOR CSE
010332  012737  177776  001552            MOV     #-2,RETRY2      ;RETRY COUNT FOR SFTWRE MISCMP'N
010340  012737  000003  001554            MOV     #3,RETRY3       ;RETRY COUNT FOR WCE

010346  013737  001746  001736            MOV     ADRES,WDSKAD    ;DISK ADRES TO WRT CHK WITH
010354  013737  001704  001740            MOV     PBUF1,WBUSAD    ;USE THIS BUFR 1 TO WRT CHK
                                                                  ;OR THE BUFR INDICATED BY THE USER
010362  012737  176400  001742            MOV     #-1400,WWRDCN   ;WRT CHK 1 BLOCK=3SECS=1400 WRDS

010370  013737  001746  001730  READ:     MOV     ADRES,DSKADR    ;DISK ADRES TO READ FROM
010376  012737  176400  001734            MOV     #-1400,WRDCNT   ;1 BLOCK = 3 SECTORS = 1400 WRDS
010404  013737  001702  001732            MOV     PBUF0,BUSADR    ;USE 'IOBUF0' TO READ INTO
                                                                  ;OR THE BUFR INDICATED BY THE USER

010412  000404                            BR      RDAGAIN

010414  104414                LUPSIN: CON.RESET
010416  104415                        DRV.RESET
010420  000401                        BR      RDAGAIN

010422  104414                LUPHE:  CON.RESET

010424  013777  001730  171330  RDAGAIN:        MOV     DSKADR,@RKDA    ;READ FROM THIS DSK-ADRES
010432  013777  001734  171316            MOV     WRDCNT,@RKWC    ;THIS # OF WORDS
010440  013777  001732  171312            MOV     BUSADR,@RKBA    ;INTO THIS BUFR

010446  012777  000405  171300            MOV     #405,@RKCS      ;READ,SSE,GO


010454  013737  001704  001744            MOV     PBUF1,BUFNO     ;SET UP STARTING ADRES
010462  017737  171234  001726            MOV     @PRSPAT,PGSUBR
010470  004777  171232                    JSR     PC,@PGSUBR      ;GO GENERATE A BUFFER
                                                                  ;OF 1400 WORDS USING THIS
                                                                  ;PATRN GENRATR

010474  104420                            CON.RDY                 ;DONE WITH PATRN GENRTNG,
```

```
                                                                        ;WAIT FOR CNT RDY TO SET
                                                                        ;(FROM PREVIOUS READ)
                                                                        ;CNT RDY SET
010476  032777  040000  171250          BIT     #BIT14,@RKCS          ;HARD ERROR?
010504  001416                         BEQ     NOHE
010506  012737  010422  001110          MOV     #LUPHE,$LPERR
010514  004737  016006                 JSR     PC,GETINF
010520  104023                         ERROR   23                    ;HARD ERROR
010522  104414                         CON.RESET
010524  005237  002006                 INC     ERCNT3                ;ALLOW 12 ERRORS AT MOST
010530  001002                         BNE     1$
010532  000137  011752                 JMP     EXT7                  ;IF MORE, EXIT
010536  000137  011274          1$:    JMP     FINISH
010542  032777  001000  171200  NOHE:  BIT     #BIT9,@RKDS           ;SIN SET?
010550  001417                         BEQ     NOSIN                 ;NO
010552  012737  010414  001110          MOV     #LUPSIN,$LPERR
010560  004737  016006                 JSR     PC,GETINF
010564  104011                         ERROR   11                    ;SIN ON READ
010566  104414                         CON.RESET
010570  104415                         DRV.RESET
010572  005237  002010                 INC     ERCNT4                ;ALLOW 5 ERRORS AT MOST
010576  001002                         BNE     1$
010600  000137  011752                 JMP     EXT7                  ;IF MORE, EXIT
010604  000137  011274          1$:    JMP     FINISH
010610  005737  002002  NOSIN:  TST     ERCNT1                ;CHECKING CSE FOR 1ST TIME
010614  001031                         BNE     WRTCHK                ;NO,BRANCH
010616  005237  002002                 INC     ERCNT1                ;INDICATE THAT CSE HAS BEEN
                                                                        ;CHECKED ONCE
010622  032777  000002  171122          BIT     #BIT1,@RKER           ;CHECK SUM EROR?
010630  001423                         BEQ     WRTCHK
010632  012737  010314  001110          MOV     #BEGIN,$LPERR
010640  004737  016006                 JSR     PC,GETINF
010644  013737  001550  001202          MOV     RETRY1,$REG10         ;GET THE RETRY #
010652  062737  000004  001202          ADD     #4,$REG10             ;SAVE IT FOR TYPEOUT
010660  104024                         ERROR   24                    ;CSE
010662  005237  002012                 INC     ERCNT5                ;ALLOW 10 ERRORS AT MOST
010666  001002                         BNE     1$
010670  000137  011752                 JMP     EXT7                  ;IF MORE, EXIT
010674  000137  011436          1$:    JMP     CSEROR                ;GO, SERVICE CSE
010700  005037  002004  WRTCHK: CLR     ERCNT2                ;CLR FLAG INDICATING SOFTWARE
                                                                        ;COMPARE DONE
010704  022737  000003  001554          CMP     #3,RETRY3             ;WRT CHK DONE BEFORE OR
010712  001016                         BNE     SFTCMP                ;IT'S 1ST TIME?
                                                                        ;IF DONE,BRANCH OTHERWISE DO IT
010714  013777  001736  171040  WCAGAIN: MOV    WDSKAD,@RKDA          ;WRT CHK FROM THIS DSK-ADRES
010722  013777  001742  171026          MOV     WWRDCN,@RKWC          ;THIS # FO WORDS
010730  013777  001740  171022          MOV     WBUSAD,@RKBA          ;WITH THIS BUFFER
010736  012777  000407  171010          MOV     #407,@RKCS            ;WRT CHK,GO,SSE
```

```
3166  010744  005337  001554                    DEC     RETRY3          ;INDICATE WRT CHK DONE
3167
3168  010750  005737  002004          SFTCMP:   TST     ERCNT2          ;SFTWARE CMPARE DONE ONCE BEFORE?
3169  010754  001060                    BNE     WCREPT          ;IF SFTWARE CMPARE HAS BEEN DONE
3170                                                            ;ONCE DON'T DO IT AGAIN. OTHERWISE,
3171  010756  005237  002004                    INC     ERCNT2          ;DO IT. INDICATE IT IS DONE.
3172                                                            ;MORE THAN ONCE BEFORE.
3173                                                            ;IF THIS IS 1ST TIME THRU &
3174                                                            ;WRT CHK WAS DONE ONCE BEFORE
3175                                                            ;DO SOFTWARE COMPARISON OF
3176                                                            ;THE DATA THAT WAS READ FROM
3177                                                            ;THE DISK
3178
3179  010762  012702  001564                    MOV     #BUFR,R2        ;INITLZE PTR TO BUFR STORING
3180                                                            ;ADRES OF BAD DATA
3181  010766  012703  001616                    MOV     #BUFR1,R3       ;STORE EXPCTD DATA STARTING HERE
3182  010772  012704  001650                    MOV     #BUFR2,R4       ;STORE RECVD (BAD) DATA
3183                                                            ;STARTING HERE
3184
3185  010776  005037  001776                    CLR     INDX3           ;CLR FLAG INDICATING MISCMPRE
3186
3187  011002  013700  001702          COMPAR:   MOV     PBUF0,R0        ;INITLZE PTR TO 'RECVD DATA' BUFR
3188  011006  012737  177764  002000            MOV     #-14,INDX4      ;STORE AND REPORT 12 OR LESS DATA ERRORS
3189  011014  013701  001704                    MOV     PBUF1,R1        ;INITLZE PTR TO 'EXPCTD DATA' BUFR
3190  011020  012737  176400  001556            MOV     #-1400,INADR    ;SET COUNT
3191  011026  021011                  CMPAGAN:  CMP     (R0),(R1)       ;CORRECT WORD READ FROM DISK?
3192  011030  001402                            BEQ     1$
3193  011032  000137  011600                    JMP     MISCMP          ;BRANCH IF MISCMPRE ERROR
3194  011036  005720                  1$:       TST     (R0)+           ;INCRMNT PTRS
3195  011040  005721                            TST     (R1)+           ;TO NXT WORDS
3196  011042  005237  001556                    INC     INADR           ;DONE WITH CMPRISON?
3197  011046  001367                    BNE     CMPAGAN         ;IF NOT, COMPARE THE REST
3198
3199  011050  005737  001776                    TST     INDX3           ;WAS THERE A BAD DATA WORD
3200                                                            ;(EVEN AFTR RETRYING)
3201  011054  001420                    BEQ     WCREPT          ;NO, BRANCH
3202  011056  012737  010370  001110    REPMSC:   MOV     #READ,$LPERR
3203  011064  104025                            ERROR   25              ;DATA ERROR
3204  011066  005237  002014                    INC     ERCNT6          ;ALLOW 10 ERRORS AT MOST
3205  011072  001002                    BNE     1$
3206  011074  000137  011752                    JMP     EXT7            ;IF MORE, EXIT
3207  011100  005737  001552          1$:       TST     RETRY2
3208  011104  001404                    BEQ     WCREPT
3209  011106  005237  001552                    INC     RETRY2
3210  011112  000137  010370                    JMP     READ
3211
3212  011116  104420                  WCREPT:   CON.RDY                 ;WAIT FOR CNTRL RDY FROM
3213                                                            ;PREVIOUS WRT CHK
3214  011120  022737  177776  001552            CMP     #-2,RETRY2      ;IF THERE WAS A RETRY AFTER MISC
3215                                                            ;-OMPARISON, DO WRT CHK AGAIN
3216  011126  001417                            BEQ     ERWCHK
3217  011130  000401                            BR      LUPWCE+2
3218  011132  104414                  LUPWCE:   CON.RESET
3219  011134  013777  001736  170620            MOV     WDSKAD,@RKDA    ;WRT CHK WITH THIS DSK-ADRES
3220  011142  013777  001742  170606            MOV     WWRDCN,@RKWC    ;THIS # OF WORDS
3221  011150  013777  001740  170602            MOV     WBUSAD,@RKBA    ;THIS BUS ADRES
```

```
3222  011156  012777  000407  170570          MOV     #407,@RKCS
3223
3224  011164  104420                          CON.RDY
3225  011166  012737  011132  001110  ERWCHK: MOV     #LUPWCE,$LPERR
3226  011174  032777  040000  170546          BIT     #BIT14,@RKDS    ;HARD EROR?(FROM WRT CHK)
3227  011202  001410                          BEQ     XHE             ;NO,BRANCH
3228
3229  011204  004737  016006                  JSR     PC,GETINF
3230  011210  104026                          ERROR   26              ;HE ON WRT CHK
3231  011212  005237  002016                  INC     ERCNT7          ;ALLOW 12 ERRORS AT MOST
3232  011216  001002                          BNE     XHE
3233  011220  000137  011752                  JMP     EXT7            ;IF MORE, EXIT
3234
3235  011224  032777  000001  170520  XHE:    BIT     #BIT0,@RKER     ;WRITE CHECK EROR?
3236  011232  001420                          BEQ     FINISH
3237  011234  004737  016006                  JSR     PC,GETINF
3238  011240  012737  000003  001202          MOV     #3,$REG10       ;GET TRY #
3239  011246  163737  001554  001202          SUB     RETRY3,$REG10   ;SAVE IT FOR TYPEOUT
3240  011254  104027                          ERROR   27              ;WRT CHK EROR
3241  011256  005237  002020                  INC     ERCNT8          ;ALLOW 10 ERRORS AT MOST
3242  011262  001002                          BNE     1$
3243  011264  000137  011752                  JMP     EXT7            ;IF MORE, EXIT
3244  011270  000137  011644          1$:     JMP     WCEROR
3245
3246
3247                                                                  ;THERE WAS NO WCE. DONE
3248                                                                  ;WITH ALL CHECKING FOR SOFT
3249                                                                  ;ERORS,ETC. MODIFY PARAMETERS
3250                                                                  ;TO CHECK NXT BLOCK ON
3251                                                                  ;THE DISK
3252
3253  011274  022737  001720  001722  FINISH: CMP     #PAT3,PRSPAT    ;FIND OUT THE NXT PATRN GENRATR
3254  011302  001404                          BEQ     1$              ;TO USE FOR GENRATING THE
3255  011304  062737  000002  001722          ADD     #2,PRSPAT       ;BUFR,STORE POINTER TO
3256  011312  000403                          BR      2$              ;GENRATR ROUTINE IN 'PRSPAT'
3257  011314  012737  001712  001722  1$:     MOV     #PAT0,PRSPAT    ;NOTE THER R 4 PAT-GENRATRS
3258                                                                  ;PAT0-PAT1-PAT2-PAT3----PAT0-
3259  011322  013701  001774          2$:     MOV     INDX2,R1        ;FORM SECTR # TO BE USED NEXT
3260  011326  116101  010210                  MOVB    SECPTR(R1),R1   ;GET SECTR #
3261  011332  042737  000017  001746  3$:     BIC     #17,ADRES       ;MASK SECTR BITS
3262  011340  050137  001746                  BIS     R1,ADRES        ;FORM THE NEW DISK-ADRES
3263                                                                  ;FORM THE NXT BLOCK TO BE
3264                                                                  ;CHECKED.
3265  011344  005237  001774                  INC     INDX2           ;DONE ALL 4 BLOCKS(3 SECS
3266                                                                  ;EACH) ON THIS TRACK?
3267  011350  022737  000004  001774          CMP     #4,INDX2
3268  011356  001025                          BNE     GOBAK           ;IF NOT, GO BAK & DO THE
3269                                                                  ;NXT BLOCK ON THIS TRACK
3270
3271  011360  005037  001774          DONTRK: CLR     INDX2           ;REINITLZE COUNT FOR
3272                                                                  ;4 BLOCKS ON THIS TRACK
3273  011364  032737  000001  001772          BIT     #BIT0,INDX1     ;WHICH SUR TO DO NXT? 0 OR 1
3274  011372  001407                          BEQ     DOSUR1          ;BRANCH, DO SUR 1 NXT
3275
3276  011374  062737  000040  001746          ADD     #40,ADRES
3277
```

```
3278  011402  042737  000020  001746          BIC     #20,ADRES       ;CLR SUR BIT, DO SUR 0 NXT
3279  011410  000403                          BR      DONE
3280
3281  011412  052737  000020  001746  DOSUR1: BIS     #20, ADRES      ;SET SUR BIT, DO SUR 1 NXT
3282
3283  011420  005237  001772          DONE:   INC     INDX1           ;DONE WITH ALL 626 TRACKS?
3284  011424  001002                          BNE     GOBAK
3285  011426  000137  011756                  JMP     TST10
3286
3287  011432  000137  010314          GOBAK:  JMP     BEGIN           ;IF NOT, GO BAK & CHECK
3288                                                                  ;THE NXT TRACK
3289
3290                          ;CSEROR                                 ;THIS IS THE ENTRY POINT
3291                                                                  ;FOR SERVICE ROUTINE FOR CHECK
3292  011436                  CSEROR:                                 ;SUM EROR. CONTROL IS XFERED
3293                                                                  ;HERE ONCE CSE OCCURS
3294
3295  011436  017700  170320                  MOV     @RKDA,R0        ;GET RKDA AFTER CSE
3296  011442  010001                          MOV     R0,R1           ;SAVE RKDA
3297  011444  032700  000017                  BIT     #17,R0          ;FORM THE ADRES OF THE SECTR
3298  011450  001002                          BNE     1$              ;WHICH GAVE CSE
3299  011452  162700  000004                  SUB     #4,R0
3300  011456  005300          1$:     DEC     R0                      ;R0 CONTAINS DSK-ADRES WHERE
3301                                                                  ;CSE OCURED
3302  011460  020037  001730                  CMP     R0,DSKADR       ;DID A PREVIOUS RETRY (IF ANY)
3303                                                                  ;GIVE CSE ON SAME ADRES
3304  011464  001021                          BNE     2$              ;NO, THIS IS A FRESH CSE, BRANCH.
3305                                                                  ;IF THIS WAS A CSE ON A
3306  011466  005237  001550                  INC     RETRY1          ;RETRY INCMNT RETRY COUNT.
3307                                                                  ;BRANCH IF 3 RETRIES HAVEN'T
3308  011472  001021                          BNE     3$              ;BEEN DONE
3309                                                                  ;GO REPORT CSE
3310                                                                  ;IF CSE WAS IN THE LAST SECTR OF
3311                                                                  ;THE 3 SEC BLOCK, GO TO 'WRTCHK'
3312                                                                  ;OTHERWISE CHECK THE REST OF
3313                                                                  ;THE SECTORS FOR CSE.
3314  011474  010102                          MOV     R1,R2
3315  011476  042702  177760                  BIC     #177760,R2
3316  011502  001002          7$:     BNE     8$
3317  011504  000137  010700                  JMP     WRTCHK
3318  011510  162702  000003          8$:     SUB     #3,R2
3319  011514  100372                          BPL     7$
3320
3321  011516  010100          6$:     MOV     R1,R0
3322  011520  012737  177775  001550          MOV     #-3,RETRY1
3323  011526  000403                          BR      3$
3324
3325  011530  012737  177776  001550  2$:     MOV     #-2,RETRY1      ;ALLOW 2 MORE TRIES FOR
3326                                                                  ;THE CSE ON SAME DISK-ADRES
3327
3328  011536  010037  001730          3$:     MOV     R0,DSKADR       ;SAVE DSK-ADRES FOR DOING
3329                                                                  ;THE RETRY-READ
3330  011542  163700  001746                  SUB     ADRES,R0        ;MODIFY THE
3331  011546  005200                          INC     R0              ;BUS ADRES & WORD COUNT
3332  011550  005300          4$:     DEC     R0                      ;TO BE USE ON RETRY-
3333  011552  001407                          BEQ     5$              ;READ
```

```
3334  011554  062737  001000  001732          ADD     #1000,BUSADR
3335  011562  062737  000400  001734          ADD     #400,WRDCNT
3336  011570  000767                          BR      4$
3337
3338  011572  104414                  5$:     CON.RESET                       ;CLR THE CSE IN RKER
3339  011574  000137  010424                  JMP     RDAGAIN                 ;GO BACK, READ AGAIN
3340                                                                          ;RETRY
3341
3342
3343
3344                                  ;MISCMP                                 ;THIS IS THE ENTRY POINT
3345                                                                          ;FOR SERVICE ROUTINE, FOR
3346                                                                          ;DATA EROR (MISCOMPARISON)
3347                                                                          ;ON SOFTWARE COMPARISON
3348                                                                          ;OF DATA READ FROM THE DISK BLOCK
3349
3350
3351  011600  104420                  MISCMP: CON.RDY                         ;WAIT FOR CNTRL RDY FROM
3352                                                                          ;PREVIOUS WRT CHK
3353  011602  032777  040000  170144          BIT     #BIT14,@RKCS
3354  011610  001401                          BEQ     1$
3355  011612  104414                          CON.RESET                       ;CLR WCE IN RKER
3356                                                                          ;BUT STILL DATA EROR ENTER HERE
3357  011614  010022                  1$:     MOV     R0,(R2)+                ;STORE MEM ADRES WHERE
3358                                                                          ;DATA EROR OCCURED
3359  011616  012123                          MOV     (R1)+,(R3)+             ;SAVE EXPCTD DATA
3360  011620  012024                          MOV     (R0)+,(R4)+             ;SAVE DATA RECVD (BAD)
3361  011622  005237  001776                  INC     INDX3                   ;INDICATE MISCMPRISON
3362
3363  011626  005237  002000                  INC     INDX4                   ;STORE ONLY 12(DEC) ERORS
3364  011632  001402                          BEQ     4$                      ;IF 12 ERORS, GO REPORT THEM
3365
3366  011634  000137  011026                  JMP     CMPAGAN                 ;GO BACK & CMPARE THE REST
3367
3368  011640  000137  011056          4$:     JMP     REPMSC
3369
3370
3371
3372                                                                          ;THIS THE ENTRY POINT FOR
3373                                                                          ;EROR SERVICE ON GETTING A
3374                                                                          ;WRITE CHECK EROR.
3375
3376  011644  005737  001554          WCEROR: TST     RETRY3                  ;DONE 3 TRIES?
3377  011650  003035                          BGT     CLRERR                  ;IF NOT SKIP, OTHERWISE REPORT
3378                                                                          ;W C EROR
3379
3380  011652  012737  000003  001554          MOV     #3,RETRY3               ;SET CONT FOR RETRIES
3381
3382  011660  017700  170076                  MOV     @RKDA,R0                ;IF WCE WAS IN THE LAST SECTOR
3383  011664  010002                          MOV     R0,R2                   ;OF THE BLOCK, NO MORE SECS
3384  011666  042702  177760                  BIC     #177760,R2              ;TO CHECK, GO TO 'FINISH'
3385  011672  001002                  3$:     BNE     4$                      ;IF IT WASN'T LAST SECTOR OF THE
3386  011674  000137  011274                  JMP     FINISH                  ;BLOCK, THEN CHECK REMAINING
3387  011700  162702  000003          4$:     SUB     #3,R2                   ;SECTORS. (STARTING FROM THE
3388  011704  100372                          BPL     3$                      ;SEC AFTER THE ONE GIVING WCE)
3389  011706  010001                  1$:     MOV     R0,R1                   ;SAVE DISK ADRES
```

```
3390  011710  163700  001736                    SUB     WDSKAD,R0
3391  011714  010137  001736                    MOV     R1,WDSKAD       ;GET SAVED DISK ADRES
3392  011720  005200                            INC     R0
3393  011722  005300                    2$:     DEC     R0
3394  011724  001407                            BEQ     CLRERR
3395  011726  062737  001000  001740            ADD     #1000,WBUSAD    ;FORM THE NEW BUS ADRES
3396  011734  062737  000400  001742            ADD     #400,WWRDCN     ;FORM THE NEW WORD COUNT
3397  011742  000767                            BR      2$
3398  011744  104414                    CLRERR: CON.RESET
3399  011746  000137  010714                    JMP     WCAGAIN
3400
3401  011752  004737  016564            EXT7:   JSR     PC,ABRT
3402
3403
3404                                    ;;*****************************************************************
3405                                    ;*TEST 10        WRITE, WRITE CHECK ON CYLINDERS 127, 128
3406                                            ;*THIS TEST WRITES 12 UNIQUE PATTERNS (ONE FOR EACH
3407                                            ;*SECTOR) ON CYLINDERS 127 AND 128, THEN WRITE
3408                                            ;*CHECK IS DONE TO SEE IF THEY WERE WRITTEN
3409                                            ;*CORRECTLY.  IT SHOULD BE NOTED THAT THERE IS
3410                                            ;*CHANGE IN 'WRITE' CURRENT AT THIS CYLINDER.
3411                                            ;*PATTERNS ARE RELOCATED ON THE CYLINDERS AFTER EACH
3412                                            ;*WRITE/WRITE CHECK CYCLE.  THUS THE FIRST TIME
3413                                            ;*PATTERN 'SP1' IS WRITTEN ON SECTOR 0, PATTERN 'SP2' ON
3414                                            ;*SECTOR 2, ETC.  AFTER THIS WRITE/WRITE CHECK CYCLE
3415                                            ;*IS OVER PATTERN 'SP2' IS WRITTEN ON SECTOR 0, PATTERN
3416                                            ;*:'SP3' ON SECTOR 1 AND SO ON.  THIS WRITE/WRITE
3417                                            ;*CHECK CYCLE IS REPEATED 12 TIMES, THUS
3418                                            ;*THE LAST WRITE/WRITE CHECK IS DONE USING
3419                                            ;*PATTERN 'SP12' ON SECTOR 0, PATTERN 'SP1' IS
3420                                            ;*WRITTEN ON SECTOR 1, PATTERN 'SP2' ON SECTOR 2,
3421                                            ;*ETC.  IF YOU WANT TO WRITE ANY OTHER PATTERNS
3422                                            ;*FILL IN THE PATTERNS YOU WANT IN LOCATIONS
3423                                            ;*'SP1', 'SP2',...ETC.
3424                                    ;;*****************************************************************
3425  011756  000004                    TST10:  SCOPE
3426
3427  011760  012737  177764  002002            MOV     #-14,ERCNT1     ;ALLOW 12 ERRORS AT MOST
3428  011766  012737  177764  002004            MOV     #-14,ERCNT2     ;ALLOW 12 ERRORS AT MOST
3429  011774  012737  177723  002006            MOV     #-55,ERCNT3     ;ALLOW 15 ERRORS AT MOST
3430  012002  012702  012464                    MOV     #SP1,R2         ;INITIALIZE POINTER TO PATRN
3431  012006  010201                    DOWRT:  MOV     R2,R1
3432  012010  012700  007740                    MOV     #7740,R0        ;SET UP CYL ADRES BITS (127)
3433  012014  053700  001526                    BIS     DRIVAD,R0       ;SET UP DRIVE # BITS
3434  012020  005003                    WRL01:  CLR     R3
3435  012022  104414                    WRERR:  CON.RESET
3436
3437  012024  010077  167732            WRL0:   MOV     R0,@RKDA        ;ADRES THE DRIVE
3438  012030  012777  177400  167720            MOV     #-400,@RKWC     ;WRITE 1 SECTOR
3439  012036  010177  167716                    MOV     R1,@RKBA        ;USE THIS PATTERN
3440  012042  012777  004003  167704            MOV     #4003,@RKCS     ;WRITE, GO
3441  012050  104420                            CON.RDY
3442  012052  032777  140000  167674            BIT     #140000,@RKCS   ;ANY ERROR?
3443  012060  001414                            BEQ     4$
3444  012062  012737  012022  001110            MOV     #WRERR,$LPERR   ;SET ADRES FOR LOOPING ON ERROR
3445  012070  004737  015754                    JSR     PC,GT4RG        ;GET TKCS, ER, DS, DA
```

```
3446  012074  104021                        ERROR   21              ;ERROR OCCURRED ON DOING A WRITE
3447  012076  104414                        CON.RESET               ;CLEAR THE ERROR
3448  012100  005237  002002                INC     ERCNT1          ;ALLOW 12 ERRORS ONLY
3449  012104  001002                        BNE     4$
3450  012106  000137  012514                JMP     EXT10
3451
3452  012112  005200                4$:     INC     R0
3453  012114  005203                        INC     R3              ;KEEP COUNT
3454  012116  022701  012512                CMP     #SP12,R1        ;USE PATTERNS IN A CYCLIC FASHION
3455  012122  001002                        BNE     3$
3456  012124  012701  012462                MOV     #SP1-2,R1
3457  012130  005721                3$:     TST     (R1)+           ;INCREMENT POINTERS TO NEXT PATTERN
3458  012132  020327  000014                CMP     R3,#14          ;DONE SURFACE 0?
3459  012136  002732                        BLT     WRLO            ;NO
3460  012140  001005                        BNE     2$              ;YES, IF CHANGING HEADS
3461  012142  010201                        MOV     R2,R1
3462  012144  042700  000017                BIC     #17,R0          ;SET UP CORRECT ADDRESS ETC.
3463  012150  052700  000020                BIS     #20,R0
3464
3465  012154  020327  000030        2$:     CMP     R3,#30          ;DONE WITH WRITING SURFACE 1?
3466  012160  001321                        BNE     WRLO            ;NO, BRANCH
3467
3468  012162  032700  007700        WRHI:   BIT     #7700,R0        ;DONE WITH BOTH CYLINDERS - 127 & 128?
3469  012166  001405                        BEQ     DOWCHK          ;YES
3470  012170  012700  010000                MOV     #10000,R0       ;NO, DO CYLINDER 128
3471  012174  053700  001526                BIS     DRIVAD,R0
3472  012200  000707                        BR      WRLO1           ;GO BACK
3473                                                                ;CYLINDERS 127 AND 128 HAVE BEEN
3474
3475
3476  012202  010201        DOWCHK: MOV     R2,R1           ;WRITTEN, NOW DO WRITE CHECK
3477  012204  012737  177775  001550        MOV     #-3,RETRY1      ;INITIALIZE POINTER TO FIRST PATTERN
3478  012212  012700  010000                MOV     #10000,R0       ;RETRY COUNT
3479  012216  053700  001526                BIS     DRIVAD,R0       ;DO CYLINDER 128 FIRST
3480  012222  005003        WCHI1:  CLR     R3
3481  012224  010077  167532        WCERR:  MOV     R0,@RKDA        ;ADRES THE DRIVE
3482
3483  012230  012777  177400  167520  WCHI:   MOV     #-400,@RKWC     ;WRITE CHECK 1 SECTOR
3484  012236  010177  167516                MOV     R1,@RKBA        ;WITH THIS PATTERN
3485  012242  012777  004007  167504        MOV     #4007,@RKCS     ;WRITE CHECK, GO
3486  012250  104420                        CON.RDY
3487
3488  012252  032777  040000  167470        BIT     #40000,@RKDS    ;HE?
3489  012260  001406                        BEQ     1$              ;NO
3490  012262  004737  016006                JSR     PC,GETINF
3491  012266  104026                        ERROR   26              ;HE ON DOING WRT CHK
3492  012270  005237  002004                INC     ERCNT2          ;ALLOW 12 ERRORS ONLY
3493  012274  001507                        BEQ     EXT10           ;IF MORE, EXIT
3494  012276  032777  000001  167446  1$:   BIT     #BIT0,@RKER     ;WCE?
3495  012304  001425                        BEQ     4$              ;NO
3496  012306  012737  012224  001110        MOV     #WCERR,$LPERR   ;SET ADRES FOR LOOPING ON ERROR
3497  012314  004737  016006                JSR     PC,GETINF       ;GET INFO ON ERROR
3498  012320  013737  001550  001202        MOV     RETRY1,$REG10
3499  012326  062737  000004  001202        ADD     #4,$REG10       ;WCE ON DOING WRITE CHECK, WITH
3500  012334  104027                        ERROR   27              ;PATTERN STORED IN R1
3501  012336  005237  001550                INC     RETRY1          ;DO 3 TIMES IN ALL
```

```
3502  012342  001330                        BNE     WCERR
3503  012344  005237  002006                INC     ERCNT3          ;ALLOW 15 ERRORS ONLY
3504  012350  001461                        BEQ     EXT10           ;IF MORE, EXIT
3505  012352  012737  177775  001550        MOV     #-3,RETRY1
3506
3507  012360  005200              4$:       INC     R0              ;KEEP TRACK OF DISK-ADRES
3508  012362  005203                        INC     R3              ;AND COUNT
3509  012364  022701  012512                CMP     #SP12,R1        ;USE PATTERNS IN CYCLIC
3510  012370  001002                        BNE     3$
3511  012372  012701  012462                MOV     #SP1-2,R1       ;FASHION
3512  012376  005721              3$:       TST     (R1)+           ;INCREMENT POINTER TO NEXT PATTERN
3513  012400  020327  000014                CMP     R3,#14          ;DONE SURFACE 0?
3514  012404  002711                        BLT     WCHI            ;NO
3515  012406  001005                        BNE     2$
3516  012410  010201                        MOV     R2,R1           ;IF CHANGING HEADS (0-1), SET CORRECT
3517  012412  042700  000017                BIC     #17,R0          ;ADRES BITS
3518  012416  052700  000020                BIS     #20,R0
3519
3520  012422  020327  000030        2$:     CMP     R3,#30          ;DONE WRITE CHECKING SURFACE 1?
3521  012426  001300                        BNE     WCHI            ;NO, GO BACK
3522
3523  012430  032700  007700        WCLO:   BIT     #7700,R0        ;DONE BOTH CYLINDERS - 127, 128?
3524  012434  001005                        BNE     REPEAT          ;YES, BRANCH
3525  012436  012700  007740                MOV     #7740,R0        ;DO CYLINDER 127 NOW
3526  012442  053700  001526                BIS     DRIVAD,R0
3527  012446  000665                        BR      WCHI1
3528
3529
3530  012450  005722        REPEAT: TST     (R2)+           ;RELOCATE THE PATTERNS ON THE
3531  012452  020227  012514                CMP     R2,#SP12+2      ;CYLINDERS AND DO IT AGAIN
3532  012456  001420                        BEQ     TST11           ;;EXIT
3533  012460  000137  012006                JMP     DOWRT           ;THIS TEXT)
3534
3535
3536
3537                          ;PATTERNS TO BE USED
3538
3539  012464  177777        SP1:    .WORD   177777          ;IF YOU WANT TO WRITE ANY
3540  012466  052525        SP2:    .WORD   052525          ;OTHER PATTERNS, CHANGE THESE
3541  012470  111111        SP3:    .WORD   111111          ;12 LOCATIONS TO ANY PATTERN
3542  012472  010421        SP4:    .WORD   010421          ;YOU WANT.
3543  012474  102041        SP5:    .WORD   102041
3544  012476  010101        SP6:    .WORD   010101
3545  012500  040201        SP7:    .WORD   040201
3546  012502  000401        SP8:    .WORD   000401
3547  012504  031463        SP9:    .WORD   031463
3548  012506  070707        SP10:   .WORD   070707
3549  012510  007417        SP11:   .WORD   007417
3550  012512  041020        SP12:   .WORD   041020
3551
3552  012514  004737  016564        EXT10:  JSR     PC,ABRT
3553
3554
3555
3556
3557
```

```
                                ;;**********************************************************************
                                ;*TEST 11       SEEK FUNCTION TIMER
                                ;*SEEK TIMER
                                ;*IN THIS PART OF THE PROGRAM SEEKS ARE TIMED BETWEEN A PARTICULAR SET
                                ;*OF CYLINDERS, BOTH IN THE FORWRAD  DIRECTION (0-312) AND REVERSE(312-0).
                                ;*******CAUTION******
                                ;*IT SHOULD BE NOTED THAT THE SECTOR COUNTER (IN RKDS) IS USED AS THE REAL
                                ;*TIME CLOCK TO DO THE SEEK TIMING. FOR THE TIMES TO BE RELIABLE, THE
                                ;*SECTOR COUNTER SHOULD BE ACCURATE WITHIN THE SPECIFICATIONS OF THE DISK
                                ;*SPEED:     1500 RPM = 40 MILI SECS/REV =3.33 MILI SECS/SECTOR.
                                ;*VARIATION: +-30  RPM
                                ;;**********************************************************************
012520  000004               TST11:  SCOPE
012522  032777  000100  166410        BIT     #SW6,@SWR       ;INHIBIT TIMER?
012530  001002                      BNE     .+6
012532  000137  013670              JMP     PLTGRPH
012536  104414                      CON.RESET
012540  104415                      DRV.RESET

012542  012737  177771  001550        MOV     #-7,RETRY1      ;COUNT FOR 7 DIFRNT SEEK TIMES
                                                              ;TO BE RECORDED

012550  104400  012556              TYPE    ,65$              ;;TYPE ASCIZ STRING
012554  000424                      BR      64$               ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/SEEK TIME SCALE FACTOR=0.01 MILI SECS/
012626                          64$:
012626  104400  012634              TYPE    ,67$              ;;TYPE ASCIZ STRING
012632  000421                      BR      66$               ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <15><12><12>/ #  OF    SEEK    #  OF   SEEK/
012676                          66$:
012676  104400  012704              TYPE    ,69$              ;;TYPE ASCIZ STRING
012702  000421                      BR      68$               ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  <15><12>/ SEEKS     TIME   SEEKS   TIME/<15><12>
012746                          68$:

012746  012737  002040  001556        MOV     #SIAD,INADR     ;INITLZE PTR TO INNER ADRES
012754  012737  002022  001560        MOV     #SOAD,OUTADR    ;INITLZE PTR TO OUTER ADRES

012762  017777  166572  166772  REPTIM: MOV     @OUTADR,@RKDA   ;POSITION HEADS TO OUTER CYLINDER
                                                              ;BEFORE STARTING TO TIME
012770  053777  001526  166764        BIS     DRIVAD,@RKDA    ;SET DRIVE # BITS
012776  012777  000011  166750        MOV     #11,@RKCS               ;SEEK, GO
013004  104420                      CON.RDY                 ;WAIT FOR CNTRL RDY
013006  104421                      TST.RWS                 ;WAIT FOR R/W/S RDY

013010  005037  001772              CLR     INDX1           ;INDX1 = 0, GOING IN; OTHERWISE OUT
013014  012704  026200              MOV     #BUFR10,R4      ;STORE FWRD SEEK TIMES IN THIS BUFR
013020  012705  026600              MOV     #BUFR11,R5      ;STORE REVRSE  "    "
013024  012737  177634  001774        MOV     #-144,INDX2     ;SET COUNT FOR # OF SEEKS

013032  013702  001762        BEGSK:  MOV     RKDA,R2
013036  005737  001772              TST     INDX1           ;GOING FWRD OR REVRSE?
013042  001005                      BNE     1$              ;REVRSE, BRANCH

013044  017712  166506              MOV     @INADR,@R2      ;FWRD, SET INNER CYL ADRES
```

```
3614  013050  053712  001526              BIS     DRIVAD,@R2      ;SET DRIVE # BITS
3615  013054  000404                      BR      2$
3616
3617  013056  017712  166476      1$:     MOV     @OUTADR,@R2     ;SET OUTER CYL ADRES
3618  013062  053712  001526              BIS     DRIVAD,@R2      ;SET DRIVE # BITS
3619  013066  004737  014560      2$:     JSR     PC,@#TIMSEK     ;GO, TIME THE SEEK FROM CYLINDER
3620                                                              ;0 TO THE ABOVE CYL. RETURN WITH
3621                                                              ;R3 CONTAINING THE TIME (MS, SCALE
3622                                                              ;FACTOR= 0.01) REQUIRED FOR THE SEEK.
3623  013072  005737  001772              TST     INDX1
3624  013076  001004                      BNE     3$
3625  013100  010324                      MOV     R3,(R4)+        ;STORE TIME TAKEN FOR FRWRD SEEK
3626  013102  005237  001772              INC     INDX1           ;SET FLAG FOR DOING REVRSE SEEK
3627  013106  000751                      BR      BEGSK           ;GO DO IT
3628
3629  013110  010325              3$:     MOV     R3,(R5)+        ;STORE TIME TAKEN FOR REVRSE SEEK
3630  013112  005037  001772              CLR     INDX1           ;CLR FLG FOR DOING FRWRD SEEK
3631
3632  013116  005237  001774              INC     INDX2           ;RECORDED 144 SEEK TIMES
3633  013122  001343                      BNE     BEGSK           ;IF NOT, GO BAK
3634
3635                                      ;AT THIS POINT 100 SEEKS HAVE BEEK PERFORMED BETWEEN TWO
3636                                      ;CYLINDERS (FORWARD & REVERSE DIRECTION).  FORWARD SEEK
3637                                      ;TIMES ARE STORED IN TABLE STARTING AT 'BUFR10' REVERSE
3638                                      ;STARTING AT 'BUFR11'.  THE FOLLOWING CODE FINDS OUT THE
3639                                      ;NUMBERS OF TIMES A PARTICULAR 'SEEK TIME' WAS OBTAINED.
3640                                      ;EXAMPLE: OUT OF 100 TIMES THE SEEK WAS DONE BETWEEN
3641                                      ;CYLINDER 0 & LAST.
3642                                      ;70 TIMES IT TOOK 95 MILI SECS
3643                                      ;20 TIMES IT TOOK 85 MILI SECS
3644                                      ;10 TIMES IT TOOK 100 MILI SECS
3645                                      ;THIS INDICATES HOW CONSISTENT WAS THE SEEKING TIME.
3646                                      ;NOTE THAT ONLY 5 DIFFERENT "SEEK TIMES" WILL BE TYPED
3647                                      ;OUT.
3648
3649                                      ;SORTING ROUTINE
3650
3651  013124  012705  177776      SORT:   MOV     #-2,R5          ;COUNT FOR FRWRD, REVRSE
3652  013130  012737  026200  001162      MOV     #BUFR10,$REG0   ;INTLZE PTR TO 'SEEK TIME'
3653  013136  012737  026202  001164      MOV     #BUFR10+2,$REG1
3654  013144  012737  025700  001166      MOV     #BUFR4,$REG2
3655  013152  012737  025760  001170      MOV     #BUFR5,$REG3    ;INTLZE PTR TO '# OF TIMES'
3656
3657  013160  013700  001162      1$:     MOV     $REG0,R0        ;PTR T 'SEEK TIME'
3658  013164  013701  001164              MOV     $REG1,R1
3659  013170  012702  177635              MOV     #-143,R2        ;COUNT FOR 143 ITEMS TO SORT
3660  013174  005003                      CLR     R3
3661
3662  013176  021011              2$:     CMP     (R0),(R1)       ;SORT THE ITEMS & PUT THEM
3663  013200  003404                      BLE     3$              ;IN DESCENDING ORDER
3664  013202  011004                      MOV     (R0),R4         ;LARGER ITEMS AT TOP OF LIST,
3665  013204  011110                      MOV     (R1),(R0)       ;SMALLER AT THE BOTTOM
3666  013206  010411                      MOV     R4,(R1)
3667  013210  005203                      INC     R3
3668  013212  005720              3$:     TST     (R0)+
3669  013214  005721                      TST     (R1)+
```

```
 3670  013216  005202                          INC     R2
 3671  013220  001366                          BNE     2$
 3672  013222  005703                          TST     R3              ;SORTED ALL ITEMS?
 3673  013224  001355                          BNE     1$              ;IF NOT LOOP BACK
 3674
 3675  013226  013700  001162                  MOV     $REG0,R0        ;PTR TO 'SEEK TIME'
 3676  013232  013701  001166                  MOV     $REG2,R1        ;SAVE 'SEEK TIME' HERE
 3677  013236  013702  001170                  MOV     $REG3,R2        ;SAVE '# OF TIMES' HERE
 3678  013242  010204                          MOV     R2,R4
 3679  013244  005024                          CLR     (R4)+           ;CLR OUT 5 WORDS OF
 3680  013246  005024                          CLR     (R4)+           ;'# OF TIMES'BUFR
 3681  013250  005024                          CLR     (R4)+
 3682  013252  005024                          CLR     (R4)+
 3683  013254  005024                          CLR     (R4)+
 3684  013256  012703  177773                  MOV     #-5,R3          ;SAVE ONLY 5 DIFRNT 'SEEK TIMES'
 3685
 3686  013262  011011                          MOV     (R0),(R1)       ;FIND OUT THE '# OF TIMES'
 3687  013264  012703  177634                  MOV     #-144,R3        ;EACH 'SEEK TIME' WAS
 3688  013270  022011                  4$:     CMP     (R0)+,(R1)
 3689  013272  001411                          BEQ     5$              ;OBTAINED
 3690
 3691  013274  005721                          TST     (R1)+
 3692  013276  016011  177776                  MOV     -2(R0),(R1)     ;SAVE 'SEEK TIME'
 3693  013302  005722                          TST     (R2)+
 3694  013304  012712  000001                  MOV     #1,(R2)         ;KEEP '# OF TIMES'
 3695  013310  005203                          INC     R3
 3696  013312  001404                          BEQ     6$
 3697  013314  000765                          BR      4$
 3698
 3699  013316  005212                  5$:     INC     (R2)            ;INCRMNT '# OF TIMES'
 3700  013320  005203                          INC     R3              ;ALL DONE?
 3701  013322  001362                          BNE     4$              ;IF NOT, GO BAK
 3702
 3703  013324  005205                  6$:     INC     R5              ;SORTED BOTH FRWRD, REVRSE
 3704  013326  001415                          BEQ     GOTYPE          ;'SEEK TIMES', IF YES GO TYPE
 3705
 3706  013330  012737  026600  001162          MOV     #BUFR11,$REG0   ;IF NOT, INITLZE PTR TO 'SEEK TIME'
 3707  013336  012737  026602  001164          MOV     #BUFR11+2,$REG1
 3708  013344  012737  026040  001166          MOV     #BUFR6,$REG2    ;SAVE 'SEEK TIME'
 3709  013352  012737  026120  001170          MOV     #BUFR7,$REG3    ;SAVE '# OF TIMES' HERE
 3710
 3711  013360  000677                          BR      1$              ;GO BAK & DO SORTING FOR REVRSE SEEK TIMES
 3712
 3713                                                  ;TYPE OUT CYL #'S BETWEEN WHICH SEEK
 3714                                                  ;WAS TIMED.  'OUTADR' TO 'INADR' 'INADR' TO 'OUTADR'
 3715  013362  104400                  GOTYPE: TYPE
 3716  013364  001213                          $CRLF
 3717  013366  104400  013374                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 3718  013372  000403                          BR      64$             ;;GET OVER THE ASCIZ
 3719                                  ;;65$:  .ASCIZ  /CYLS:/
 3720  013402                          64$:
 3721
 3722  013402  017700  166152                  MOV     @OUTADR,R0      ;GET OUTER CYL #
 3723  013406  006200                          ASR     R0
 3724  013410  006200                          ASR     R0
 3725  013412  006200                          ASR     R0
```

```
3726  013414  006200                          ASR     R0
3727  013416  006200                          ASR     R0
3728  013420  010046                          MOV     R0,-(SP)
3729  013422  104423                          TYPDSS                  ;TYPE IT OUT IN DECIMAL
3730  013424  104400  013432                  TYPE    ,67$            ;;TYPE ASCIZ STRING
3731  013430  000401                          BR      66$             ;;GET OVER THE ASCIZ
3732                                ;;67$:    .ASCIZ  /-/
3733  013434                      66$:
3734  013434  017701  166116                  MOV     @INADR,R1       ;GET INNER CYL #
3735  013440  006201                          ASR     R1
3736  013442  006201                          ASR     R1
3737  013444  006201                          ASR     R1
3738  013446  006201                          ASR     R1
3739  013450  006201                          ASR     R1
3740  013452  010146                          MOV     R1,-(SP)
3741  013454  104423                          TYPDSS                  ;TYPE IT OUT IN DECIMAL
3742  013456  104400  013464                  TYPE    ,69$            ;;TYPE ASCIZ STRING
3743  013462  000405                          BR      68$             ;;GET OVER THE ASCIZ
3744                                ;;69$:    .ASCIZ  <15><12>/  FRWRD/
3745  013476                      68$:
3746  013476  104400  002377                  TYPE    ,BLNKS9
3747  013502  104400  013510                  TYPE    ,71$            ;;TYPE ASCIZ STRING
3748  013506  000404                          BR      70$             ;;GET OVER THE ASCIZ
3749                                ;;71$:    .ASCIZ  /REVRSE/
3750  013520                      70$:
3751
3752                                          ;TYPE OUT THE '# OF SEEKS' & 'SEEK
3753                                          ;TIME' OBTAINED FOR EACH OF THOSE
3754                                          ;SEEKS
3755  013520  005000              TYPTIM:     CLR     R0
3756  013522  005005                          CLR     R5
3757  013524  104400              1$:         TYPE
3758  013526  001213                          $CRLF
3759  013530  016046  025760                  MOV     BUFR5(R0),-(SP) ;GET '# OF SEEKS', IF NONE (0)
3760  013534  001424                          BEQ     3$              ;SKIP TYPING (FRWRD SEEK)
3761  013536  104404                          TYPDS                   ;GO TYPE OUT DECIMAL '# OF SEEKS'
3762  013540  104400                          TYPE
3763  013542  002406                          BLNKS2
3764  013544  016046  025700                  MOV     BUFR4(R0),-(SP) ;GET 'SEEK TIME' FOR EACH OF
3765  013550  104404                          TYPDS                   ;OF THAT '# OF SEEKS'. 'GO
3766                                          ;TYPE OUT IN DECIMAL
3767
3768  013552  016046  026120              2$:         MOV     BUFR7(R0),-(SP) ;GET '# OF SEEKS', IF NONE (0)
3769  013556  001416                          BEQ     4$              ;SKIP TYPING (REVRSE SEEK)
3770  013560  005705                          TST     R5
3771  013562  001402                          BEQ     6$
3772  013564  104400  002373                  TYPE    ,BLNK13
3773  013570  104404              6$:         TYPDS                   ;TYPE OUT IN DECIMAL
3774  013572  104400                          TYPE
3775  013574  002406                          BLNKS2
3776  013576  016046  026040                  MOV     BUFR6(R0),-(SP) ;GET 'SEEK TIME' & TYPE IT
3777  013602  104404                          TYPDS                   ;OUT IN DECIMAL
3778  013604  000406                          BR      5$
3779
3780  013606  005726              3$:         TST     (SP)+           ;POP STACK
3781  013610  005205                          INC     R5
```

```
3782  013612  000757                          BR      2$
3783
3784  013614  005726                  4$:     TST     (SP)+           ;POP STACK
3785  013616  005705                          TST     R5
3786  013620  001004                          BNE     TIMDON
3787
3788  013622  005720                  5$:     TST     (R0)+           ;INCREMENT PTR TO TABLES
3789  013624  020027  000012                  CMP     R0,#12          ;ALL DONE?
3790  013630  001335                          BNE     1$              ;IF NOT GO BAK
3791
3792  013632  062737  000002  001556  TIMDON: ADD     #2,INADR        ;INCRMNT POINTER TO NEXT
3793  013640  062737  000002  001560          ADD     #2,OUTADR       ;INNER & OUTER ADRES
3794  013646  005237  001550                  INC     RETRY1          ;ALL DONE?
3795  013652  001406                          BEQ     PLTGRPH
3796  013654  032777  000100  165256          BIT     #SW6,@SWR       ;INHIBIT TIMER? FURTHER ?
3797  013662  001402                          BEQ     PLTGRPH         ;YES, BRANCH
3798  013664  000137  012762                  JMP     REPTIM          ;GO, BACK AND TIME REST
3799                                                                  ;OF SEEKS
3800
3801
3802
3803                          ;PLOT GRAPH OF 'SEEK TIME' V/S 'CYLIDERS SEEKED'
3804
3805                                  ;PERFORM SEEK FROM CYLINDER 0 TO EVERY OTHER CYLINDER AND TIME IT.
3806                                  ;0      0,0     1,----0 312. NOTE 'SECTOR COUNTER' IS USED AS A READ
3807                                  ;TIME CLOCK TO TIME THERE SEEKS.  AFTER OBTAINING THE SEEK TIMES A
3808                                  ;GRAPH IS PLOTTED OF 'SEEK TIME' V/S 'CYLINDER'.
3809
3810                                  ;TIME THE SEEKS
3811  013670  032777  000040  165242  PLTGRPH: BIT #SW5,@SWR          ;SKIP THE GRAPH?
3812  013676  001002                          BNE     .+6
3813  013700  000137  015006                  JMP     TST12           ;YES, BRANCH
3814  013704  104414                          CON.RESET
3815  013706  104415                          DRV.RESET
3816  013710  012737  177465  001776          MOV     #-313,INDX3     ;PERFORM 313 SEEKS 0-0,0-1,0-312
3817  013716  012704  026200                  MOV     #BUFR10,R4      ;STORE 'SEEK TIME' HERE
3818  013722  005037  001556                  CLR     INADR           ;CLR CYL ADRES BITS
3819
3820  013726  013777  001556  166026  1$:     MOV     INADR,@RKDA     ;ADRES THE RIGHT CYLINDER
3821  013734  053777  001526  166020          BIS     DRIVAD,@RKDA    ;ADRES THE RIGHT DRIVE
3822
3823  013742  004737  014560                  JSR     PC,TIMSEK       ;GO TIME THE SEEK FROM CYL 0
3824                                                                  ;TO THE ABOVE CYL. RETURN WITH
3825                                                                  ;R3 CONTAINING 'SEEK TIME' IN MS
3826                                                                  ;SCALE FACTOR OF 0.01
3827  013746  010324                          MOV     R3,(R4)+        ;STORE 'SEEK TIME'
3828  013750  042777  017777  166004          BIC     #17777,@RKDA            ;SEEK BACK TO CYL 0 FOR
3829  013756  012777  000011  165770          MOV     #11,@RKCS       ;TIMING NXT CYL SEEK
3830  013764  104420                          CON.RDY                 ;WAIT FOR CNTRL RDY?
3831  013766  104421                          TST.RWS                 ;WAIT FOR R/W/S RDY
3832  013770  062737  000040  001556          ADD     #40,INADR       ;FORM NXT CYL ADRES
3833  013776  005237  001776                  INC     INDX3
3834  014002  001351                          BNE     1$
3835
3836                                  ;PLOT A GRAPH USING 'SEEK TIMES' RECORDED BEFORE
3837
```

```
3838  014004                          PLOT:
3839  014004  104400  014012                TYPE    ,65$            ;;TYPE ASCIZ STRING
3840  014010  000422                    BR      64$             ;;GET OVER THE ASCIZ
3841                                  ;;65$:  .ASCIZ  <15><12><12><12>/X AXIS - SEEK TIME - MILI SECS/
3842  014056                          64$:
3843  014056  104400  014064                TYPE    ,67$            ;;TYPE ASCIZ STRING
3844  014062  000423                    BR      66$             ;;GET OVER THE ASCIZ
3845                                  ;;67$:  .ASCIZ  <15><12>/Y AXIS - CYLINDER SEEKED FROM 0/<15><12><12>
3846  014132                          66$:
3847
3848  014132  104400                    TYPE
3849  014134  002401                    BLNKS7
3850  014136  005000                    CLR     R0              ;TYPE OUT THE TIME UNITS
3851  014140  010046              1$:   MOV     R0,-(SP)        ;(MILI SECS) FOR THE X-AXIS
3852  014142  104423                    TYPDSS                  ;LIKE THIS:
3853  014144  005700                    TST     R0              ;0   20   30   40.....
3854  014146  001411                    BEQ     2$
3855  014150  022700  000144            CMP     #144,R0
3856  014154  003010                    BGT     4$
3857  014156  022700  000170            CMP     #170,R0
3858  014162  002412                    BLT     5$
3859  014164  104400                    TYPE
3860  014166  002406                    BLNKS2
3861  014170  000404                    BR      3$
3862  014172  104400              2$:   TYPE
3863  014174  002407                    BLNKS1
3864  014176  104400              4$:   TYPE
3865  014200  002405                    BLNKS3
3866  014202  062700  000012      3$:   ADD     #12,R0
3867  014206  000754                    BR      1$
3868
3869  014210  104400              5$:   TYPE
3870  014212  001213                    $CRLF
3871  014214  104400                    TYPE
3872  014216  002401                    BLNKS7
3873
3874  014220  012700  177763      PLT1: MOV     #-15,R0         ;TYPE OUT THE X-AXIS MARKERS
3875  014224                      1$:
3876  014224  104400  014232            TYPE    ,65$            ;;TYPE ASCIZ STRING
3877  014230  000403                    BR      64$             ;;GET OVER THE ASCIZ
3878                                  ;;65$:  .ASCIZ  /I----/
3879  014240                          64$:
3880  014240  005200                    INC     R0              ;I----I----I----
3881  014242  001370                    BNE     1$
3882
3883                                  ;IF SW 4 IS SET THEN TYPE THE COMPLETE GRAPH. IF NOT TYPE THE SMALL GRAPH.
3884
3885  014244  032777  000020  164666      BIT     #SW4,@SWR       ;TYPE COMPLETE GRAPH?
3886  014252  001054                    BNE     CMPGRP          ;YES BRANCH
3887                                                            ;IF NOT, TYPE SMALL GRAPH
3888
3889
3890
3891  014254  005000              SMGRP: CLR    R0
3892  014256  032777  000040  164654 1$:   BIT     #SW5,@SWR       ;SKIP REST OF GRAPH?
3893  014264  001445                    BEQ     5$              ;YES
```

```
3894  014266  104400                          TYPE                    ;IN THIS GRAPH SEEK TIMES ARE
3895  014270  001213                          $CRLF                   ;PLOTTED ONLY FOR SELECTED
3896                                                                  ;CYLINDERS (NOT ALL) SHOWN BELOW:
3897                                                                  ;0,1,2,3,4,  6,8,10,12,14,16,18,20,
3898                                                                  ;25,30,35,...,190,195,200,  203
3899  014272  010046                          MOV     R0,-(SP)                ;TYPE THE MARKERS
3900  014274  104404                          TYPDS
3901  014276  104400  014304                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3902  014302  000401                          BR      64$             ;;GET OVER THE ASCIZ
3903                                  ;;65$:  .ASCIZ  /-/
3904  014306                          64$:
3905  014306  010001                          MOV     R0,R1           ;FORM THE ADRES OF 'SEEK TIME'
3906  014310  006301                          ASL     R1
3907  014312  016103  026200                  MOV     BUFR10(R1),R3   ;GET THE SEEK TIME
3908  014316  004737  014520                  JSR     PC,PLTPT                ;GO PLOT IT
3909  014322  022700  000004                  CMP     #4,R0           ;PLOTTED UPTO CYL 4?
3910  014326  003402                          BLE     2$              ;YES
3911  014330  005200                          INC     R0
3912  014332  000751                          BR      1$
3913  014334  022700  000024          2$:     CMP     #24,R0  ;PLOTTED UPTO CYL 20?
3914  014340  003403                          BLE     3$
3915  014342  062700  000002                  ADD     #2,R0
3916  014346  000743                          BR      1$
3917  014350  022700  000310          3$:     CMP     #310,R0         ;PLOTTED UPTO CYL 200?
3918  014354  003403                          BLE     4$
3919  014356  062700  000005                  ADD     #5,R0
3920  014362  000735                          BR      1$
3921  014364  022700  000312          4$:     CMP     #312,R0         ;PLOTTED ALL CYLS?
3922  014370  001403                          BEQ     5$
3923  014372  062700  000002                  ADD     #2,R0
3924  014376  000727                          BR      1$
3925  014400  000137  015006          5$:     JMP     TST12
3926
3927
3928                                  ;IF SW 4 IS SET THE COMPLETE GRAPH IS PRINTED OUT. IT GIVES TIMES FOR
3929                                  ;SEEKS FROM CYLINDER 0 TO ALL OTHER CYLINDERS (0,1,2,3...202).
3930
3931  014404  005000                  CMPGRP: CLR     R0              ;INITLZE COUNT
3932  014406  012701  177773                  MOV     #-5,R1          ;INITLZE COUNT FOR Y-AXIS MARKER
3933  014412  012702  026200                  MOV     #BUFR10,R2      ;INITLZE PTR TO SEEK TIMES
3934  014416  104400                          TYPE
3935  014420  001213                          $CRLF
3936  014422  000412                          BR      3$
3937
3938  014424  032777  000040  164506  2$:     BIT     #SW5,@SWR               ;SKIP REST OF GRAPH?
3939  014432  001002                          BNE     .+6
3940  014434  000137  015006                  JMP     TST12
3941  014440  005201                          INC     R1              ;TYPE OUT Y-AXIS MARKER 'CYL #'
3942  014442  001005                          BNE     4$              ;IF REQUIRED
3943  014444  012701  177773                  MOV     #-5,R1
3944  014450  010046                  3$:     MOV     R0,-(SP)        ;TYPE 'CYL #' ON Y-AXIS
3945  014452  104404                          TYPDS                   ;(IN DECIMAL)
3946  014454  000402                          BR      5$
3947  014456  104400                  4$:     TYPE
3948  014460  002402                          BLNKS6
3949  014462                          5$:
```

```
3950  014462  104400  014470          TYPE    ,65$            ;;TYPE ASCIZ STRING
3951  014466  000401                  BR      64$             ;;GET OVER THE ASCIZ
3952                          ;;65$:  .ASCIZ  /-/
3953  014472                  64$:
3954
3955  014472  012203                  MOV     (R2)+,R3                ;GET SEEK TIME
3956  014474  004737  014520          JSR     PC,PLTPT        ;GO PLOT THE POINT
3957
3958
3959  014500  104400                  TYPE
3960  014502  001213                  $CRLF
3961  014504  005200                  INC     R0              ;ALL DONE?
3962  014506  022700  000312          CMP     #312,R0
3963  014512  001344                  BNE     2$              ;IF NOT, GO BAK
3964  014514  000137  015006  6$:     JMP     TST12
3965
3966                  ;PLTPT
3967                  ;THE ROUTINE IS ENTERED WITH R3 CONTAINING HORIZONTAL AXIS
3968                  ;COORDINATE- SEEK TIME
3969                  ;PLOT THE ACTUAL TIME ON THE GRAPH.  IN KEEPING WITH NORMAL
3970                  ;CONVENTION A NUMBER LESS THAN HALF THE CELL WIDTH IS
3971                  ;CONSIDERED AS FALLING UNDER THE PREVIOUS CELL, A NUMBER
3972                  ;GREATER THAN OR EQUAL TO HALF THE CELL WIDTH FALLS UNDER THE NEXT CELL
3973                  ;EX: IF SEEK TIME IS 11.5 MS, IT'S BETW'N 10 & 12, BUT > 11
3974                  ;HENCE IT WILL BE PLOTTED AS 12 IF SEEK TIME IS 10.8 MS,
3975                  ;IT'S BETW'N 10 & 12, BUT < 11 HENCE IT WILL BE PLOTTED
3976                  ;AS 10.0 MS
3977
3978  014520  162703  000310  PLTPT:  SUB     #310,R3         ;FIND OUT HOW MANY BLANKS TO
3979  014524  002403                  BLT     7$              ;INSERT TO PLOT THE POINT
3980                                                          ;NOTE THE FIRST CELL = 0 MS
3981  014526  104400                  TYPE
3982  014530  002407                  BLNKS1
3983  014532  000772                  BR      PLTPT
3984  014534  062703  000144  7$:     ADD     #144,R3
3985  014540  002402                  BLT     8$
3986  014542  104400                  TYPE
3987  014544  002407                  BLNKS1
3988
3989  014546                  8$:
3990  014546  104400  014554          TYPE    ,65$            ;;TYPE ASCIZ STRING
3991  014552  000401                  BR      64$             ;;GET OVER THE ASCIZ
3992                          ;;65$:  .ASCIZ  /X/
3993  014556                  64$:
3994  014556  000207                  RTS     PC
3995
3996                  ;TIMSEK
3997                  ;THIS ROUTINE FINDS OUT THE TIME REQUIRED TO SEEK TO THE CYLINDER
3998                  ;INDICATED IN RKDA.
3999                  ;***CAUTION*** SECTOR COUNTER IS USED AS A REAL TIME CLOCK TO KEEP TIME.
4000                  ;ENTRY: JSR     PC,TIMSEK
4001                  ;       RKDA CONTAINS THE CYLINDER TO BE SEEKED FROM THE PRESENT POSITION.
4002                  ;RETURN:        R3 CONTAINS THE SEEK TIME IN MILI SECS. SCALE FACTOR = 0.01
4003
4004
4005                                                          ;R3 WILL COUNT REVOLUTIONS OF
```

```
 4006  014560  010246                TIMSEK: MOV     R2,-(SP)        ;DISK (FROM INDEX MARK TO INDEX MARK)
 4007  014562  005003                        CLR     R3              ;40 MILI SECS FOR EACH REV
 4008  014564  013701  001750                MOV     RKDS,R1
 4009  014570  011102                1$:     MOV     @R1,R2
 4010  014572  032702  000400                BIT     #400,R2         ;WAIT FOR SOK
 4011  014576  001774                        BEQ     1$
 4012
 4013  014600  032702  000010                BIT     #BIT3,R2                ;WAIT FOR SECTOR 10 SO THAT
 4014  014604  001771                        BEQ     1$              ;U CAN START WAITING FOR
 4015                                                                ;INDEX, SEC 0
 4016
 4017  014606  011102                2$:     MOV     @R1,R2
 4018  014610  032702  000400                BIT     #400,R2         ;WAIT FOR SEC OK
 4019  014614  001774                        BEQ     2$
 4020  014616  021102                        CMP     @R1,R2
 4021  014620  001372                        BNE     2$
 4022  014622  032702  000017                BIT     #17,R2                  ;WAIT FOR SEC 0, INDEX MARK
 4023  014626  001367                        BNE     2$              ;AS SOON AS IT IS SEC 0, ISSUE
 4024                                                                ;A SEEK & START TIMING
 4025                                                                ;
 4026  014630  012777  000011  165116        MOV     #11,@RKCS       ;ISSUE A SEEK, START TIMING
 4027                                                                ;THE SEC COUNTER
 4028  014636  104420                        CON.RDY                 ;WAIT FOR CNTRL RDY
 4029
 4030  014640  011102                3$:     MOV     @R1,R2          ;GET RKDS
 4031  014642  032702  000400                BIT     #400,R2         ;WAIT FOR SOK
 4032  014646  001774                        BEQ     3$
 4033  014650  020211                        CMP     R2,@R1          ;INFO CORRECT?
 4034  014652  001372                        BNE     3$              ;NO
 4035  014654  032702  000100                BIT     #100,R2         ;R/W/S RDY SET?
 4036  014660  001025                        BNE     SKDON           ;IF YES, BRANCH
 4037  014662  032702  000017                BIT     #17,R2          ;WAIT FOR SEC CNTR TO MOVE
 4038  014666  001764                        BEQ     3$              ;FROM 0 TO 1
 4039
 4040  014670  011102                4$:     MOV     @R1,R2
 4041  014672  032702  000400                BIT     #400,R2         ;WAIT FOR SOK
 4042  014676  001774                        BEQ     4$
 4043  014700  020211                        CMP     R2,@R1
 4044  014702  001372                        BNE     4$
 4045  014704  032702  000100                BIT     #100,R2         ;R/W/S RDY SET, SEEK DONE?
 4046  014710  001005                        BNE     5$              ;YES, BRANCH
 4047  014712  032702  000017                BIT     #17,R2                  ;IF NOT KEEP TRACK OF SEC
 4048  014716  001364                        BNE     4$              ;COUNTER. INCREMENT R3 AT
 4049                                                                ;EVERY INDEX MARK,EVERY
 4050  014720  005203                        INC     R3              ;40 MILI SECS
 4051  014722  000746                        BR      3$              ;GO BAK, KEEP TIME
 4052
 4053  014724  032702  000017        5$:     BIT     #17,R2          ;CHECK, IS IT INDEX MARK -SEC 0
 4054  014730  001001                        BNE     SKDON           ;IF NOT, SKIP
 4055  014732  005203                        INC     R3              ;IF YES, INCREMENT COUNT
 4056
 4057                                                                        ;SEEK DONE, SAVE RKDS-SEC COUNTER.
 4058  014734                        SKDON:
 4059  014734  012746  000014                MOV     #14,-(SP)               ;;PUT THE MULTIPLER ON THE STACK
 4060  014740  010346                        MOV     R3,-(SP)                ;;PUT THE MULTIPLICAND ON THE STACK
 4061  014742  004737  020256                JSR     PC,@#$MULT      ;;CALL THE MULTPLY ROUTINE
```

```
4062  014746  012616                        MOV     (SP)+,(SP)      ;;DISREGARD THE MSB'S
4063  014750  012603                        MOV     (SP)+,R3                ;;GET THE LS3'S OF THE PRODUCT
4064  014752  042702  177760                BIC     #177760,R2      ;SEEK. TOTAL TIME=(IN DECIMAL)
4065  014756  060203                        ADD     R2,R3           ;[(R3)X12+SEC COUNTER]X330X0.01
4066                                                                ;NOTE THERE IS A SCALE FACTOR
4067  014760  012746  000512                MOV     #512,-(SP)              ;;PUT THE MULTIPLER ON THE STACK
4068  014764  010346                        MOV     R3,-(SP)                ;;PUT THE MULTIPLICAND ON THE STACK
4069  014766  004737  020256                JSR     PC,@#$MULT      ;;CALL THE MULTPLY ROUTINE
4070  014772  012616                        MOV     (SP)+,(SP)      ;;DISREGARD THE MSB'S
4071  014774  012603                        MOV     (SP)+,R3                ;;GET THE LSB'S OF THE PRODUCT
4072  014776  062703  000245                ADD     #245,R3         ;ASSUMPTION THAT EACH SECTOR
4073                                                                ;TAKES 3.3 MILI SECS. IF THE
4074                                                                ;DISK SPEED IS VERY MUCH DIFRNT
4075                                                                ;FROM THE SPEC SPEED OF
4076                                                                ;1500 RPM (40 MS/REV), THEN
4077                                                                ;SEC COUNTER WOULD NOT BE AN
4078                                                                ;ACCURATE TIME CLOCK.
4079  015002  012602                        MOV     (SP)+,R2        ;POP R2 BAK
4080  015004  000207                        RTS     PC              ;RETURN
4081
4082
4083                                ;;*****************************************************************
4084                                ;*TEST 12        END OF PROGRAM
4085                                        ;*THIS IS NOT A TEST BUT IS JUST A LINKAGE
4086                                        ;*PROVIDED TO TEST ALL THE DRIVES.
4087                                ;;*****************************************************************
4088  015006  000004                ^ST12:  SCOPE
4089  015010  104400  001213                TYPE    ,$CRLF
4090  015014  104400  001213                TYPE    ,$CRLF
4091  015020  105237  001522                INCB    DRVDON
4092  015024  123737  001522  001523 BTEOP: CMPB    DRVDON,DRIVS
4093  015032  001402                        BEQ     .+6
4094  015034  000137  003524                JMP     NXTDRV
4095
4096                                .SBTTL  END OF PASS ROUTINE
4097
4098                                ;;*****************************************************************
4099                                ;*INCREMENT THE PASS NUMBER ($PASS)
4100                                ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
4101                                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
4102                                ;*IF THERES A MONITOR GO TO IT
4103                                ;*IF THERE ISN'T JUMP TO ST3
4104
4105  015040                        $EOP:
4106  015040  000004                        SCOPE
4107  015042  005037  001102                CLR     $TSTNM          ;;ZERO THE TEST NUMBER
4108  015046  005237  001100                INC     $PASS           ;;INCREMENT THE PASS NUMBER
4109  015052  042737  100000  001100        BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
4110  015060  005327                        DEC     (PC)+           ;;LOOP?
4111  015062  000001                $EOPCT: .WORD   1
4112  015064  003022                        BGT     $DOAGN          ;;YES
4113  015066  012737                        MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
4114  015070  000001                $ENDCT: .WORD   1
4115  015072  015062                        $EOPCT
4116  015074  104400  015141                TYPE    ,$ENDMG         ;;TYPE "END PASS #"
4117  015100  013746  001100                MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
```

```
 4118  015104  104404                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 4119  015106  104400  015136                  TYPE    $ENULL          ;;TYPE A NULL CHARACTER
 4120  015112  013700  000042        $GET42:   MOV     @#42,R0         ;;GET MONITOR ADDRESS
 4121  015116  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 4122  015120  000005                          RESET                   ;;CLEAR THE WORLD
 4123  015122  004710                $ENDAD:   JSR     PC,(R0)         ;;GO TO MONITOR
 4124  015124  000240                          NOP                     ;;SAVE ROOM
 4125  015126  000240                          NOP                     ;;FOR
 4126  015130  000240                          NOP                     ;;ACT11
 4127  015132                        $DOAGN:
 4128  015132  000137                          JMP     @(PC)+          ;;RETURN
 4129  015134  003506                $RTNAD:   .WORD   ST3
 4130  015136     377     377     000  $ENULL:  .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 4131  015141     015  042412  042116  $ENDMG:  .ASCIZ  <15><12>/END PASS */
 4132  015146  050040  051501  020123
 4133  015154  000043
 4134
```

```
                                ;COMMON SUBROUTINES AND HANDLERS



                                        .SBTTL  ESR15
                                ;ESR15
                                ;THIS ROUTINE IS USED TO TYPE OUT ERROR DATA FOR ITEM 15
                                ;OF THE ERROR TABLE.  AT THE TIME OF ENTRY INTO THIS
                                ;ROUTINE R5 CONTAINS THE DISK ADDRESS FROM WHICH THE 12
                                ;HEADERS WERE READ, THE SECTOR #'S WHICH GAVE BAD HEADERS HAVE
                                ;BEEN STORED STARTING AT 'BUFR'.  THE CORRESPONDING BAD HEADERS
                                ;HAVE BEEN STORED STARTING AT 'BUFR1'.

                                ;THE PRINTOUT LOOKS LIKE:
                                ;SEC#   HDR RECVD
                                ;AA     BBBBBB          AA=BAD SEC #  BBBBBB=BAD HEADER
                                ;
                                ;EXPCTD HDR=XXXXXX      TRY#=  Y

015156                          ESR15:
015156  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
015160  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
015162  012701  001564                  MOV     #BUFR,R1        ;SEC #'S STORED HERE PREVIOUSLY
015166  012702  001616                  MOV     #BUFR1,R2       ;BAD HDRS STORED HERE PRVSLY
015172  012146                  1$:     MOV     (R1)+,-(SP)
015174  104402                          TYPOS                   ;GO TYPE OUT BAD SEC # (OCTAL)
015176     002                          .BYTE   2               ;ONLY 2 DIGITS
015177     000                          .BYTE   0               ;SUPRES LDG 0'S
015200  104400                          TYPE                    ;TYPE 3 BLNKS
015202  002405                          BLNKS3
015204  012246                          MOV     (R2)+,-(SP)     ;GO TYPE OUT BAD HEADER
015206  104401                          TYPOC
015210  104400                          TYPE
015212  002404                          BLNKS4
015214  104400                          TYPE
015216  001213                          $CRLF
015220  022711  177777                  CMP     #177777,(R1)    ;ALL BAD SEC #'S TYPD OUT?
015224  001362                          BNE     1$              ;IF NOT GO BAK

015226  104400                          TYPE
015230  002214                          MSG6
015232  010546                          MOV     R5,-(SP)        ;TYPE OUT EXPCTD HEADER FOR
015234  042716  160037                  BIC     #160037,(SP)
015240  104401                          TYPOC                   ;THAT CYLINDER

015242  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
015244  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
015246  000207                          RTS     PC


                                        .SBTTL  ESR13
                                ;ESR13
                                ;THIS ROUTINE IS USED WITH 'ERROR 13"' TO TYPEOUT OUT ERROR
                                ;DATA.  THE SECTOR #'S WHICH GAVE BAD HEADERS HAVE BEEN STORED
                                ;STARTING AT 'BUFR'.  THE CORRESPONDING BAD HEADERS HAVE
                                ;BEEN STORED STARTING AT 'BUFR1'.  R5 CONTAINS THE EXPECTED
```

```
                                ;HEADER FOR THAT CYLINDER.  THE TYPEOUT LOOKS LIKE
                                ;
                                ;SEC#  HDR RCVD
                                ;AA     BBBBBB              AA=BAD SEC #
                                ;                           BBBBBB=BAD HEADER
                                ;
                                ;EXPCTD HDR=XXXXXX      TRY#:  Y        SUR=Z

015250  004737  015156       ESR13:  JSR     PC,ESR15
015254  104400  015262               TYPE    ,65$                ;;TYPE ASCIZ STRING
015260  000404                       BR      64$                 ;;GET OVER THE ASCIZ
                             ;;65$:  .ASCIZ  /  SUR=/
015272                       64$:
015272  005046                       CLR     -(SP)
015274  032705  000020               BIT     #20,R5              ;SUR 0 OR 11?
015300  001401                       BEQ     1$
015302  005216                       INC     (SP)
015304  104401               1$:     TYPOC

015306  104400  002351               TYPE    ,MSG13
015312  013746  001552               MOV     RETRY2,-(SP)
015316  005216                       INC     (SP)
015320  104401                       TYPOC
015322  000207                       RTS     PC


                                        .SBTTL  ESR20
                                ;ESR20
                                ;SUBROUTINE TO TYPE OUT ERROR DATA FOR 'ERROR 20'.  AT THE TIME
                                ;OF ENTRY, TABLE STARTING AT 'BUFR' CONTAINS SECTOR #'S THAT GAVE BAD
                                ;HEADERS.  TABLE AT 'BUFR1' CONTAINS BAD HEADERS, R5 CONTAINS EXPECTED
                                ;HEADER FOR THE CYLINDER.  'INADR' AND 'OUTADR' CONTAIN THE CYLINDER
                                ;ADDRESSES BETWEEN WHICH THE IMPLIED SEEK WAS TRIED.

015324  004737  015250       ESR20:  JSR     PC,ESR13            ;GO TYPE OUT SEC #'S, BAD HDRS
015330  004737  016112               JSR     PC,ERR2             ;GET CYL #'S BETWN WHICH SEEK
                                                                 ;WAS TRIED
015334  104400  015342               TYPE    ,65$                ;;TYPE ASCIZ STRING
015340  000404                       BR      64$                 ;;GET OVER THE ASCIZ
                             ;;65$:  .ASCIZ  /  CYLA=/
015352                       64$:
015352  013746  001162               MOV     $REG0,-(SP)         ;GO TYPE CYL # FROM WHERE
015356  104402                       TYPOS                       ;SEEK BEGAN
015360     003                       .BYTE   3                   ;TYPE 3 DIGITS
015361     000                       .BYTE   0                   ;SUPRES LDG 0'S
015362  104400  015370               TYPE    ,67$                ;;TYPE ASCIZ STRING
015366  000404                       BR      66$                 ;;GET OVER THE ASCIZ
                             ;;67$:  .ASCIZ  /  CYLB=/
015400                       66$:
015400  013746  001164               MOV     $REG1,-(SP)         ;TYPE CYL # TO WHICH SEEK
015404  104402                       TYPOS                       ;WAS DONE
015406     003                       .BYTE   3                   ;TYPE 3 DIGITS
015407     000                       .BYTE   0                   ;SUPRES LDG 0'S
015410  000207                       RTS     PC                  ;RETURN


                                        .SBTTL  ESR25
```

```
4247  015412  010205                ESR25:  MOV     R2,R5           ;SAVE ADRES OF TERMINATOR
4248
4249  015414  012702  001564                MOV     #BUFR,R2        ;INITLZE PTR TO TABLE STORING
4250                                                                  ;ADRES OF BAD DATA
4251  015420  012703  001616                MOV     #BUFR1,R3       ;INITLZE PTR TO 'EXPCTD' DATA
4252  015424  012704  001650                MOV     #BUFR2,R4       ;INITLZE PTR TO 'RECVD' DATA
4253
4254  015430  032777  020000  163502  1$:   BIT     #SW13,@SWR      ;INHIBIT TYPE OUT?
4255  015436  001076                        BNE     4$              ;YES, EXIT
4256  015440  104400                        TYPE                    ;TYPE CR,LF
4257  015442  001213                        $CRLF
4258
4259  015444  163712  001702                SUB     PBUF0,(R2)      ;GET WORD # IN BUFR (0,1,2...)
4260  015450  006212                        ASR     (R2)
4261  015452  011246                        MOV     (R2),-(SP)      ;WHICH WAS BAD.  NOTE YOU
4262                                                                  ;CAN HAVE THE ACTUAL MEMORY
4263                                                                  ;ADRES BY ADDING 'IOBUF0'
4264                                                                  ;TO THIS
4265  015454  104402                        TYPOS                   ;GO TYPE WORD # THAT WAS BAD
4266  015456     004                        .BYTE   4
4267  015457     000                        .BYTE   0
4268  015460  104400                        TYPE
4269  015462  002405                        BLNKS3                  ;2 BLANKS
4270
4271  015464  012346                        MOV     (R3)+,-(SP)     ;GET EXPCTD DATA
4272  015466  104401                        TYPOC                   ;GO TYPE IT
4273  015470  104400                        TYPE
4274  015472  002406                        BLNKS2
4275  015474  012446                        MOV     (R4)+,-(SP)     ;GET RECVD DATA (BAD)
4276  015476  104401                        TYPOC                   ;GO TYPE IT
4277  015500  104400                        TYPE
4278  015502  002406                        BLNKS2
4279
4280  015504  012700  000400                MOV     #400,R0         ;GET THE DISK ADRES FROM
4281  015510  021200                2$:     CMP     (R2),R0         ;WHICH THIS (BAD) DATA WAS
4282  015512  002405                        BLT     3$              ;READ
4283  015514  062700  000400                ADD     #400,R0
4284  015520  022700  002400                CMP     #2400,R0
4285  015524  001371                        BNE     2$
4286
4287  015526  000300                3$:     SWAB    R0
4288  015530  005300                        DEC     R0
4289  015532  063700  001746                ADD     ADRES,R0        ;R0 CONTAINS THE DISK
4290                                                                  ;ADRES FROM WHICH THE (BAD)
4291  015536  010037  001170                MOV     R0,$REG3        ;DATA WAS READ
4292
4293  015542  004737  016012                JSR     PC,BRKDA        ;GO BREAK ABOVE DISK ADRES
4294                                                                  ;INTO CYL#, SUR#, SEC#
4295
4296  015546  013746  001174                MOV     $REG5,-(SP)     ;GET THE CYL#
4297  015552  104402                        TYPOS                   ;TYPE IT
4298  015554     003                        .BYTE   3               ;ONLY 3 DIGITS
4299  015555     000                        .BYTE   0               ;NO LEADING 0'S
4300  015556  104400                        TYPE
4301  015560  002405                        BLNKS3
4302
```

```
 4303  015562  013746  001176                MOV     $REG6,-(SP)     ;GET SUR #
 4304  015566  104402                        TYPOS                   ;TYPE
 4305  015570     001                        .BYTE   1               ;1 DIGIT ONLY
 4306  015571     000                        .BYTE   0
 4307
 4308  015572  104400                        TYPE
 4309  015574  002404                        BLNKS4
 4310
 4311  015576  013746  001200                MOV     $REG7,-(SP)     ;GET SEC#
 4312  015602  104402                        TYPOS                   ;TYPE
 4313  015604     002                        .BYTE   2               ;2 DIGITS
 4314  015605     000                        .BYTE   0
 4315
 4316  015606  005722                        TST     (R2)+           ;INCREMNT PTR
 4317  015610  020205                        CMP     R2,R5           ;TYPED OUT ALL BAD DATA
 4318                                                                ;INFO?
 4319  015612  001306                        BNE     1$              ;IF NOT LUP BAK
 4320  015614  104400                        TYPE
 4321  015616  002351                        MSG13                   ;'  TRY #:'
 4322  015620  013746  001552                MOV     RETRY2,-(SP)    ;GET RETRY COUNT
 4323  015624  062716  000003  —             ADD     #3,(SP)         ;FORM THE RETRY NO.
 4324  015630  104402                        TYPOS                   ;TYPE IT OUT
 4325  015632     001                        .BYTE   1
 4326  015633     000                        .BYTE   0
 4327
 4328  015634  000207              4$:       RTS     PC              ;IF YES, RETURN
 4329                              ;MESSAGE HANDLER
 4330                              ;THE MESSAGE HANDLER IS USED FOR TYPING OUT MESSAGES & DATA
 4331                              ;RELATED TO THE MESSAGE. IF SW13 IS SET, THE TYPEOUT IS
 4332                              ;INHIBITED. THE CALL IS:
 4333                              ;               MESAGE  ,XX
 4334                              ;XXX IS THE MESSAGE NUMBER & PROVIDES AN INDEX TO THE
 4335                              ;'ERROR ITEMS TABLE' WHERE THAT MESAGE ITEM
 4336                              ;IS LOCATED.
 4337                              ;THE MESAGE ITEM CONTAINS:
 4338                              ;       MS:     POINTER TO THE ASCII MESAGE
 4339                              ;       DH:     POINTER TO THE DATA HEADER
 4340                              ;       DT:     POINTER TO THE DATA
 4341                              ;       0       TERMINATOR
 4342                              ;IF 'DT' IS 0 THE DATA IS TO BE PRINTED USING THE SUBROUTINE
 4343                              ;INDICATED IN PLACE OF THE TERMINATOR
 4344
 4345  015636  032777  020000  163274  MSGE:   BIT     #SW13,@SWR      ;INHIBIT TYPEOUT?
 4346  015644  001012                        BNE     1$              ;IF YES, EXIT
 4347  015646  011637  001116                MOV     (SP),$ERRPC     ;GET ADRES OF 'MESAGE' CALL
 4348  015652  162737  000002  001116        SUB     #2,$ERRPC       ;STORE IT
 4349  015660  117637  000000  001114        MOVB    @(SP),$ITEMB    ;GET MESAGE # (INDEX TO ITEM TABLE)
 4350  015666  004737  017224                JSR     PC,@#ERRTYP     ;GO TO 'ERRTYP' & TYPE OUT
 4351                                                                ;INFO
 4352  015672  062716  000002          1$:   ADD     #2,(SP)         ;ADJUST RETURN ADDRES
 4353  015676  000002                        RTI                     ;EXIT
 4354
 4355                              ;THIS ROUTINE IS USED FOR TYPING OUT ASCII MESSAGES. BEFORE
 4356                              ;THE MESSAGE IS TYPED SW13 IS CHECKED & IF SET THE
 4357                              ;TYPEOUT IS INHIBITED & AN EXIT IS MADE.
 4358                              ;THE CALL FOR THIS ROUTINE IS "TYPMSG", AN ENCODED
```

```
                                        ;TRAP INSTRUCTION.
                                        ;THE POINTER TO THE ASCII MESSAGE TO BE TYPED IS LOCATED IN  THE
                                        ;WORD FOLLOWING THE "TYPMSG" CALL.
                                
015700  032777  020000  163232  TY.MSG:  BIT     #SW13,@SWR      ;INHIBIT TYPEOUT?
015706  001005                          BNE     2$              ;YES, EXIT
015710  017637  000000  015720          MOV     @(SP),1$        ;GET POINTER TO ASCII MESSAGE
015716  104400                          TYPE                    ;GO TYPE ASCII STRING
015720  000000                  1$:     0
015722  062716  000002          2$:     ADD     #2,(SP)         ;ADJUST RETURN ADRES, SKIP OVER
                                                                ;POINTER ON RETURN
015726  000002                          RTI                     ;EXIT
                                
                                
                                
                                        ;GT5RG
                                        ;THIS ROUTINE EXTRACTS THE CYLINDER # FROM RKDA AND STORES IT
                                        ;IN $REG4. THEN TRANSFERS RKCS, ER, DS, DA TO $REG0, $REG1, $REG2, $REG3
015730  017746  164026          GT5RG:  MOV     @RKDA,-(SP)     ;PUSH RKDA ONTO STACK
015734  042716  160037                  BIC     #160037,(SP)    ;MASK OUT NON-CYLINDER BITS
015740  006316                          ASL     (SP)            ;SHIFT 8 BITS OF CYL ADRES TO LO BYTE
015742  006316                          ASL     (SP)
015744  006316                          ASL     (SP)
015746  000316                          SWAB    (SP)
015750  112637  001172                  MOVB    (SP)+,$REG4     ;UP STACK
                                
                                
                                        ;GT4RG
                                        ;THIS ROUTINE TRANSFERS THE CONTENTS OF RKCS, RKER, RKDS
                                        ;RKDA TO $REG0, $REG1, $REG2, $REG3 RESPECTIVELY. $REG'S
                                        ;ARE USED FOR TYPING OUT THERE CONTENTS AT THE TIME OF ERROR
                                
015754  017737  163774  001162  GT4RG:  MOV     @RKCS,$REG0     ;GET RKCS
015762  017737  163764  001164          MOV     @RKER,$REG1     ;    RKER
015770  017737  163754  001166          MOV     @RKDS,$REG2     ;    RKDS
015776  017737  163760  001170          MOV     @RKDA,$REG3     ;    RKDA
016004  000207                          RTS     PC              ;EXIT FROM THIS ROUTINE
                                
                                        ;GETINF
                                        ;THIS ROUTINE SAVES THE CONTENTS OF RKCS IN $REG0
                                        ;RKER IN $REG1, RKDS IN $REG2.  THEN IT BREAKS RKDA
                                        ;INTO DRIVE NO, CYLINDER #, SURFACE AND SECTOR #.
                                        ;AND SAVES THEM IN $REG4, $REG5, $REG6, $REG7.
                                
016006  004737  015754          GETINF: JSR     PC,GT4RG
016012  010046                  BRKDA:  MOV     R0,-(SP)
016014  010146                          MOV     R1,-(SP)
016016  010246                          MOV     R2,-(SP)
016020  012700  001202                  MOV     #$REG7+2,R0
016024  013701  001170                  MOV     $REG3,R1
016030  010102                          MOV     R1,R2
016032  042702  177760                  BIC     #177760,R2
016036  010240                          MOV     R2,-(R0)
016040  006201                          ASR     R1
```

```
 4415  016042  006201                          ASR     R1
 4416  016044  006201                          ASR     R1
 4417  016046  006201                          ASR     R1
 4418  016050  010102                          MOV     R1,R2
 4419  016052  042702  177776                  BIC     #177776,R2
 4420  016056  010240                          MOV     R2,-(R0)
 4421  016060  006201                          ASR     R1
 4422  016062  010102                          MOV     R1,R2
 4423  016064  042702  177400                  BIC     #177400,R2
 4424  016070  010240                          MOV     R2,-(R0)
 4425  016072  000301                          SWAB    R1
 4426  016074  042701  177770                  BIC     #177770,R1
 4427  016100  010140                          MOV     R1,-(R0)
 4428  016102  012602                          MOV     (SP)+,R2
 4429  016104  012601                          MOV     (SP)+,R1
 4430  016106  012600                          MOV     (SP)+,R0
 4431  016110  000207                          RTS     PC
 4432
 4433
 4434
 4435
 4436
 4437
 4438                                  .SBTTL  ERR2
 4439                                ;ERR2
 4440                                ;THIS ROUTINE GETS THE CYLINDER NUMBERS BETWEEN WHICH (IMPLIED) SEEK
 4441                                ;WAS DONE.  (R4)=0 INDICATES SEEK FROM 'OUTADR' TO 'INADR'
 4442                                ;(R4)=1 INDICATES SEEK TO 'OUTADR'.  ON EXIT $REG0 CONTAINS CYL #
 4443                                ;FROM WHICH SEEK WAS INITIATED, $REG1 CONTAINS CYL # TO WHICH SEEK WAS DONE
 4444  016112  013737  001556  001162  ERR2:   MOV     INADR,$REG0     ;GET CYL ADRES
 4445  016120  004737  016226          ERR2:   JSR     PC,GCYL         ;GO GET CYL# FROM IT
 4446  016124  013737  001162  001164          MOV     $REG0,$REG1     ;SAVE
 4447  016132  013737  001560  001162          MOV     OUTADR,$REG0    ;GET CYL ADRES
 4448  016140  004737  016226                  JSR     PC,GCYL         ;GO GET CYL # FROM IT
 4449  016144  005704                          TST     R4              ;GOING WHICH WAY?
 4450  016146  001407                          BEQ     1$              ;'OUTADR' TO 'INADR', BRANCH
 4451  016150  013746  001162                  MOV     $REG0,-(SP)     ;EXCHANG CYL" TO GET
 4452  016154  013737  001164  001162          MOV     $REG1,$REG0     ;CORRECT 'TO' & 'FROM' CYLS
 4453  016162  012637  001164                  MOV     (SP)+,$REG1
 4454  016166  000207                  1$:     RTS     PC              ;RETURN
 4455
 4456                                  .SBTTL  ERR1
 4457                                ;ERR1
 4458                                ;THIS SUBROUTINE FINDS OUT THE CYLINDER NOS. BETWEEN WHICH THE SEEK
 4459                                ;IS DONE.THE CYLINDER # WHERE THE  HEADS WHERE PRIOR TO MOVING, IS
 4460                                ;DEPOSITED IN $REG0. THE CYLINDER # WHERE THE HEADS SHOULD BE AFTER
 4461                                ;MOVEMENT, IS DEPOSITED IN $REG1.  R4 INDICATES WHICH DIRECTION THE
 4462                                ;HEADS WERE MOVING, IN OR OUT. R5 CONTAINS THE
 4463                                ;DISK ADDRESS (IN OR OUT AS THE CASE MAY BE).
 4464
 4465  016170  010537  001162          ERR1:   MOV     R5,$REG0
 4466  016174  004737  016226                  JSR     PC,GCYL ;GO GET CYL #
 4467  016200  005704                          TST     R4              ;WAS GOING IN OR OUT?
 4468  016202  001006                          BNE     1$              ;OUT
 4469  016204  013737  001162  001164          MOV     $REG0,$REG1
 4470  016212  005037  001162                  CLR     $REG0
```

```
4471  016216  000207                          RTS     PC
4472  016220  005037  001164          1$:     CLR     $REG1
4473  016224  000207                          RTS     PC
4474
4475
4476                                          .SBTTL  GCYL
4477                                  ;GCYL
4478                                  ;THIS ROUTINE EXTRACTS THE CYLINDER NO. FROM THE DISK ADDRESS
4479                                  ;CONTAINED IN '$REG0', AND THEN STORES IT BACK IN '$REG0'
4480  016226  010046                  GCYL:   MOV     R0,-(SP)        ;PUSH R0 ONTO STACK
4481  016230  013700  001162                  MOV     $REG0,R0
4482  016234  042700  160037                  BIC     #160037,R0      ;MASK OUT DRV # BITS &
4483                                                                  ;SUR, SEC BITS IF PRESENT
4484  016240  006200                          ASR     R0
4485  016242  006200                          ASR     R0              ;SHIFT CYL BITS RIGHT
4486  016244  006200                          ASR     R0              ;BY 5
4487  016246  006200                          ASR     R0
4488  016250  006200                          ASR     R0
4489  016252  010037  001162                  MOV     R0,$REG0        ;STORE CYL # IN $REG0
4490  016256  012600                          MOV     (SP)+,R0        ;POP R0 FROM STACK
4491  016260  000207                          RTS     PC              ;EXIT
4492
4493
4494                                          .SBTTL  DRV.RESET - DRIVE RESET ROUTINE
4495                                          .SBTTL  RESDON - WAIT FOR DRIVE RESET TO BE DONE
4496                                  ;DR.RST
4497                                  ;THIS ROUTINE DOES A DRIVE RESET ON THE DRIVE WHOOSE ADDRESS IS IN
4498                                  ;RKDA. MULTIPLE RETURN ADDRESSES FOR THIS ROUTINE ARE PROVIDED.
4499                                  ;IF THERE IS NO ERROR (R/W/S RDY SETS WITHIN CERTAIN TIME) , THEN
4500                                  ;A NORMAL EXIT IS MADE. IF R/W/S RDY DOES NOT SET ERROR IS REPORTED.
4501
4502
4503  016262  005037  001174  DR.RST: CLR     $REG5           ;INITIALIZE THE COUNT
4504  016266  013777  001526  163466          MOV     DRIVAD,@RKDA
4505  016274  012777  000015  163452          MOV     #15,@RKCS       ;DRIVE RESET, GO
4506  016302  104420                          CON.RDY
4507  016304  000402                          BR      RES.DO+4
4508  016306  005037  001174  RES.DO: CLR     $REG5
4509  016312  032777  000100  163430  1$:     BIT     #100,@RKDS              ;DID R/W/S RDY SET?
4510  016320  001024                          BNE     2$
4511  016322  012746  177770                  MOV     #-10,-(SP)      ;PUSH COUNT ON SP
4512  016326  005216                          INC     (SP)            ;COUNT IT DOW
4513  016330  001376                          BNE     .-2
4514  016332  005726                          TST     (SP)+           ;POP UP SP
4515  016334  005237  001174                  INC     $REG5                   ;IF NOT WAIT
4516  016340  001364                          BNE     1$                      ;WAITED LONG?
4517  016342  032777  020000  162570          BIT     #SW13,@SWR
4518  016350  001010                          BNE     2$
4519  016352  104417                          TYPMSG
4520  016354  002325                          MSG12
4521  016356  104417  002231                  TYPMSG  ,MSG7
4522  016362  011646                          MOV     (SP),-(SP)
4523  016364  162716  000002                  SUB     #2,(SP)
4524  016370  104401                          TYPOC
4525  016372  000002          2$:     RTI
4526
```

```
                                                .SBTTL  CON.RESET - CONTROL RESET ROUTINE
                                                .SBTTL  CON.RDY - WAIT FOR CONTROL READY
                                        ;CON.RESET
                                        ;CON.RDY
                                        ;THIS ROUTINE IS CALLED BY USING 'CNT.RESET' WHICH IS ACTUALLY
                                        ;'TRAP' INSTRUCTION WITH THE LOWER BYTE ENCODED TO PROVIDE
                                        ;AN INDEX TO THE CONTROL-RESET ROUTINE BELOW.
                                        ;THE 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=XXXXXX
                                        ;IS GIVEN.
                                        ;THIS ROUTINE IS CALLED THROUGH THE 'TRAP' INSTRUCTION
                                        ;USING THE LOWER BYTE AS AN INDEX TO THIS ROUTINE.
                                        ;THE TRAP DECODER LOCATED AT '$TRAP'.


                                        ;CN.RDY
                                        ;THE CN.RDY ROUTINE IS CALLED BY USING CNT.RDY WHICH IS A TRAP
                                        ;INSTRUCTION WITH ITS LOWER BYTE ENCODED.
                                        ;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.
016374  012777  000001  163352  CN.RST: MOV     #1,@RKCS        ;ISSUE A CONTROL RESET
016402  012737  177500  001170          MOV     #-300,$REG3     ;SET UP COUNT
016410  000402                  BR      CN.RDY+4                ;SKIP OVER CN.RDY
016412  005037  001170          CN.RDY: CLR     $REG3
016416  105777  163332          1$:     TSTB    @RKCS           ;DID CNTRL-RDY SET?
016422  100431                  BMI     2$              ;YES, EXIT
016424  005237  001170          INC     $REG3           ;WAITED LONG?
016430  001372                  BNE     1$              ;IF NOT, GO BAK & WAIT
016432  104417                  TYPMSG
016434  002240                  MSG10
016436  104400  016444          TYPE    ,65$            ;;TYPE ASCIZ STRING
016442  000403                  BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/PC=/
016452                          64$:
016452  011646                  MOV     (SP),-(SP)
016454  162716  000002          SUB     #2,(SP)
016460  104401                  TYPOC                   ;GO TYPE PC IN THE MAIN PROGRAM,
                                                        ; WHERE ERROR OCCURRED
016462  104400  016470          TYPE    ,67$            ;;TYPE ASCIZ STRING
016466  000404                  BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  / RKCS=/
016500                          66$:
016500  017746  163250          MOV     @RKCS,-(SP)     ;GET RKCS
016504  104401                  TYPOC                   ;GO TYPE IT

016506  000002          2$:     RTI                     ;RETURN FROM THIS
                                                        ;ROUTINE TO THE MAIN
```

```
                                                                ;PROGRAM

                                        .SBTTL  TST.RWS - WAIT FOR R/W/S RDY
                                ;TST.RWS
                                ;THIS ROUTINE WAITS FOR THE R/W/S READY TO  ET AND RETURNS
                                ;TO THE MAIN PROGRAM WHEN IT SETS.  IF IT DOES NOT SET
                                ;WITHIN A CERTAIN TIME  AN ERROR IS REPORTED.
                                ;WAITING TIME APPROX.  1040 MS FOR 11/20.  208 MS FOR 11/45

016510  005037  001562          TSTRWS: CLR     TIMER
016514  032777  000100  163226  1$:     BIT     #100,@RKDS
016522  001017                        BNE     2$
016524  005237  001562                INC     TIMER
016530  001371                        BNE     1$
016532  032777  020000  162400        BIT     #BIT13,@SWR
016540  001010                        BNE     2$
016542  104417  002325                TYPMSG  ,MSG12
016546  104417  002231                TYPMSG  ,MSG7
016552  011646                        MOV     (SP),-(SP)
016554  162716  000002                SUB     #2,(SP)
016560  104401                        TYPOC
016562  000002                  2$:   RTI

                                        .SBTTL  TEST ABORT ROUTINE
                                ;ABRT

016564  104400  002114          ABRT:   TYPE    ,MSG3
016570  113746  001102                MOVB    $TSTNM,-(SP)
016574  104401                        TYPOC
016576  000207                        RTS     PC


                                ;COMMON SUBROUTINES & HANDLERS

                                .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
                                ;*SW09=1         LOOP ON ERROR
                                ;*CALL
                                ;*      SCOPE           ;;SCOPE=IOT

016600                          $SCOPE:
016600  104406                        CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
016602  032777  000400  162330        BIT     #SW8,@SWR               ;WAS SW8 USED TO SELECT
016610  001053                        BNE     $OVER           ;A TEST? IF YES, SKIP OVER
                                                                ;THE REST, U ARE LOOPING ON
016612  032777  040000  162320  1$:   BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
016620  001047                        BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
016622  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
```

```
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
016624  013746  000004                MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
016630  012737  016650  000004        MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
016636  005737  177060                TST     @#177060        ;;TIME OUT ON XOR?
016642  012637  000004                MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
016646  000421                        BR      $SVLAD          ;;GO TO THE NEXT TEST
016650  022626                5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
016652  012637  000004                MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
016656  000407                        BR      7$              ;;LOOP ON THE PRESENT TEST
016660                        6$:;#####END OF CODE FOR THE XOR TESTER#####
016660  105737  001103        2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
016664  001412                        BEQ     $SVLAD          ;;BR IF NO
016666  032777  001000  162244        BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
016674  001404                        BEQ     4$              ;;BR IF NO
016676  013737  001110  001106  7$:   MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
016704  000415                        BR      $OVER
016706  105037  001103        4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
016712  105237  001102        $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
016716  011637  001106                MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
016722  011637  001110                MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
016726  005037  001204                CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
016732  112737  000001  001115        MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
016740  013777  001102  162174  $OVER: MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
016746  013716  001106                MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
016752  000002                        RTI                     ;;FIXES PS

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

                        .SBTTL  ERROR HANDLER ROUTINE

                        ;*SW15=1        HALT ON ERROR
                        ;*SW13=1        INHIBIT ERROR TYPEOUTS
                        ;*SW10=1        BELL ON ERROR
                        ;*SW09=1        LOOP ON ERROR
                        ;*SW12=1        CYCLE ON ERROR TO PREVIOS 'SCOPE' STATEMENT
                        ;*GO TO ERRTYP ON ERROR
                        ;*NOT FROM SYSMAC

016754                  $ERROR:
016754  105237  001103        7$:     INCB    $ERFLG          ;SET THE ERROR FLAG
016760  001775                        BEQ     7$              ;DON'T LET THE FLAG GO TO ZERO
016762  013777  001102  162152        MOV     $TSTNM,@DISPLAY
016770  032777  002000  162142        BIT     #SW10,@SWR
016776  001402                        BEQ     1$
017000  104400  001206                TYPE    ,$BELL
017004  005237  001112        1$:     INC     $ERTTL
017010  011637  001116                MOV     (SP),$ERRPC
017014  032777  000004  162116        BIT     #SW2,@SWR       ;DROP THE DRIVE?
017022  001404                        BEQ     5$              ;SW NOT SET, SKIP
017024  023727  001112  000006        CMP     $ERTTL,#6       ;MORE THAN 6 ERRORS ON THIS DRIVE?
017032  101037                        BHI     6$              ;YES, DROP THE DRIVE
017034  162737  000002  001116  5$:   SUB     #2,$ERRPC
017042  117737  162050  001114        MOVB    @$ERRPC,$ITEMB
```

```
 4695  017050  032777  020000  162062          BIT     #SW13,@SWR
 4696  017056  001004                          BNE     2$
 4697  017060  004737  017224                  JSR     PC,@#ERRTYP
 4698  017064  104400  001213                  TYPE    ,$CRLF
 4699  017070  005777  162044          2$:     TST     @SWR
 4700  017074  100001                          BPL     3$
 4701  017076  000000                          HALT
 4702  017100  032777  010000  162032  3$:     BIT     #SW12,@SWR
 4703  017106  001402                          BEQ     .+6
 4704  017110  013716  001106                  MOV     $LPADR,(SP)
 4705  017114  032777  001000  162016          BIT     #SW09,@SWR
 4706  017122  001402                          BEQ     4$
 4707  017124  013716  001110                  MOV     $LPERR,(SP)
 4708  017130  000002                  4$:     RTI
 4709
 4710  017132  013746  001524          6$:     MOV     DRVPTR,-(SP)    ;GET POINTER TO DRIVE #
 4711  017136  162716  000002                  SUB     #2,(SP)
 4712  017142  042736  000377                  BIC     #377,@(SP)+     ;CLEAR THE DRIVE PRESENT FLAG
 4713  017146  104400  002362                  TYPE    ,MSG14
 4714  017152  013746  001526                  MOV     DRIVAD,-(SP)
 4715  017156  000241                          CLC                     ;GET THE DRIVE #
 4716  017160  006116                          ROL     (SP)
 4717  017162  006116                          ROL     (SP)
 4718  017164  006116                          ROL     (SP)
 4719  017166  006116                          ROL     (SP)
 4720  017170  104401                          TYPOC                   ;TYPE IT OUT
 4721  017172  104400  017200                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 4722  017176  000405                          BR      64$             ;;GET OVER THE ASCIZ
 4723                          ;;65$:  .ASCIZ  / DROPPED/
 4724  017212                  64$:
 4725  017212  105337  001523                  DECB    DRIVS           ;DECRMNT # OF DRIVS PRESENT
 4726  017216  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK
 4727  017220  000137  015024                  JMP     BTEOP           ;EXIT
 4728
 4729  017224                  ERRTYP:
 4730  017224  104400  001213                  TYPE    ,$CRLF          ;"CARRIAGE RETURN" & LINE FEED"
 4731  017230  010046                          MOV     R0,-(SP)        ;SAVE R0
 4732  017232  005000                          CLR     R0              ;PICKUP THE ITEM INDEX
 4733  017234  153700  001114                  BISB    @#$ITEMB,R0
 4734  017240  001011                          BNE     1$              ;IF ITEM NUMBER IS ZERO, JUST
 4735
 4736                                                                  ;TYPE THE PC OF THE ERROR
 4737  017242  013746  001116                  MOV     $ERRPC,-(SP)    ;SAVE $ERRPC FOR TYPEOUT
 4738                                                                  ;ERROR ADDRESS
 4739  017246  104401                          TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4740  017250  104400                          TYPE
 4741  017252  002231                          MSG7
 4742  017254  013746  001116                  MOV     $ERRPC,-(SP)
 4743  017260  104401                          TYPOC
 4744  017262  000440                          BR      6$              ;GET OUT
 4745  017264  005300                  1$:     DEC     R0              ;ADJUST THE INDEX SO THAT IT WILL
 4746  017266  006300                          ASL     R0              ;       WORK FOR THE ERROR TABLE
 4747  017270  006300                          ASL     R0
 4748  017272  006300                          ASL     R0
 4749  017274  062700  001216                  ADD     #$ERRTB,R0      ;FORM TABLE POINTER
 4750  017300  012037  017310                  MOV     (R0)+,2$        ;PICKUP "ERROR MESSAGE" POINTER
```

```
 4751  017304  001404                          BEQ     3$              ;SKIP TYPEOUT IF NOT POINTER
 4752  017306  104400                          TYPE                    ;TYPE THE "ERROR MESSAGE"
 4753  017310  000000                  2$:     .WORD   0               ;"CARRIAGE RETURN" & LINE FEED"
 4754  017312  104400  001213                  TYPE    ,$CRLF          ;PICKUP "DATA HEADER" POINTER
 4755  017316  032777  004000  161614  3$:     BIT     #SW11,@SWR      ;DUMP OUT ALL RK REGISTERS
 4756  017324  001042                          BNE     10$             ;YES, BRANCH
 4757  017326  012037  017336                  MOV     (R0)+,4$        ;PICKUP "DATA HEADER" POINTER
 4758  017332  001412                          BEQ     5$              ;SKIP TYPEOUT IF 0
 4759  017334  104400                          TYPE                    ;TYPE THE "DATA HEADER"
 4760  017336  000000                  4$:     .WORD   0               ;"DATA HEADER" POINTER GOES HERE
 4761  017340  104400  001213                  TYPE    ,$CRLF          ;"CARRIAGE RETURN" & LINE FEED"
 4762  017344  062700  000002                  ADD     #2,R0           ;FORM POINTER TO TERMINATOR
 4763  017350  005710                          TST     (R0)            ;IS THE TERMINATOR 0?
 4764  017352  001017                          BNE     9$              ;IF NOT, BRANCH
 4765  017354  162700  000002                  SUB     #2,R0           ;YES, IT IS 0.  REPOINT TO "DATA"
 4766                                                                  ;GO TYPE OUT DATA AS USUAL
 4767  017360  011000                  5$:     MOV     (R0),R0         ;PICKUP "DATA TABLE" POINTER
 4768  017362  001004                          BNE     7$              ;GO TYPE THE DATA
 4769  017364  012600                  6$:     MOV     (SP)+,R0        ;RESTORE R0
 4770  017366  104400  001213                  TYPE    ,$CRLF          ;"CARRIAGE RETURN" & LINE FEED"
 4771  017372  000207                          RTS     PC              ;RETURN
 4772  017374                          7$:
 4773  017374  013046                          MOV     @(R0)+,-(SP)    ;SAVE @(R0)+ FOR TYPEOUT
 4774  017376  104401                          TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4775  017400  005710                          TST     (R0)            ;IS THERE ANOTHER NUMBER?
 4776  017402  001770                          BEQ     6$              ;BR IF NO
 4777  017404  104400  002406                  TYPE    ,$LNKS2
 4778  017410  000771                          BR      7$
 4779  017412  004770  000000          9$:     JSR     PC,@(R0)        ;GO TO THE SPECIAL ERROR
 4780                                                                  ;DATA HANDLING SUBROUTINE
 4781                                                                  ;NOTE THAT THIS ROUTINE IS
 4782                                                                  ;THE ONE INDICATED IN THE
 4783                                                                  ;LAST WORD OF AN ERROR
 4784                                                                  ;ITEM IN THE ERROR TABLE
 4785                                                                  ;(STARTING AT $ERRTB)
 4786  017416  104400                          TYPE
 4787  017420  002231                          MSG7
 4788  017422  013746  001116                  MOV     $ERRPC,-(SP)
 4789  017426  104401                          TYPOC
 4790  017430  000755                          BR      6$              ;GO BACK, TO THE EXIT POINT
 4791                                                                  ;FOR 'ERRTYP'
 4792
 4793  017432  004737  017440          10$:    JSR     PC,DMPREG
 4794  017436  000752                          BR      6$
 4795
 4796
 4797
 4798                                  ;DMPREG
 4799                                  ;DUMPS OUT ALL RK REGISTERS WHEN SW 11 IS SET
 4800
 4801  017440                          DMPREG:
 4802  017440  104400  017446                  .       TYPE    ,65$            ;;TYPE ASCIZ STRING
 4803  017444  000441                          BR      64$             ;;GET OVER THE ASCIZ
 4804                                  ;;65$:  .ASCIZ  <15><12>/ PC     RKDS    RKER    RKCS    RKWC    RKBA    RKDA    RKDB/<
 4805  017550                          64$:
 4806  017550  013746  001116                  MOV     $ERRPC,-(SP)
```

```
 4807  017554  104401                    TYPOC
 4808  017556  104400  002406            TYPE    ,BLNKS2
 4809  017562  010046                    MOV     R0,-(SP)
 4810  017564  012700  001750            MOV     #RKDS,R0
 4811  017570  013046              1$:   MOV     2(R0)+,-(SP)
 4812  017572  104401                    TYPOC
 4813  017574  104400  002406            TYPE    ,BLNKS2
 4814  017600  020027  001764            CMP     R0,#RKDB
 4815  017604  003771                    BLE     1$
 4816  017606  012600                    MOV     (SP)+,R0
 4817  017610  000207                    RTS     PC
 4818
 4819
 4820
 4821
 4822
 4823                              .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 4824
 4825                              ;;**********************************************************************
 4826                              ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 4827                              ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 4828                              ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 4829                              ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 4830                              ;*REPLACED WITH SPACES.
 4831                              ;*CALL:
 4832                              ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 4833                              ;*      TYPDS                   ;;GO TO THE ROUTINE
 4834
 4835  017612                      $TYPDS:
 4836  017612  010046                    MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 4837  017614  010146                    MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 4838  017616  010246                    MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 4839  017620  010346                    MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 4840  017622  010546                    MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 4841  017624  012746  020200            MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 4842  017630  016605  000020            MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 4843  017634  100004                    BPL     1$              ;;BR IF INPUT IS POS.
 4844  017636  005405                    NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 4845  017640  112766  000055  000001    MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 4846  017646  005000              1$:   CLR     R0              ;;ZERO THE CONSTANTS INDEX
 4847  017650  012703  020026            MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 4848  017654  112723  000040            MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 4849  017660  005002              2$:   CLR     R2              ;;CLEAR THE BCD NUMBER
 4850  017662  016001  020016            MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 4851  017666  160105              3$:   SUB     R1,R5           ;;FORM THIS BCD DIGIT
 4852  017670  002402                    BLT     4$              ;;BR IF DONE
 4853  017672  005202                    INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 4854  017674  000774                    BR      3$
 4855  017676  060105              4$:   ADD     R1,R5           ;;ADD BACK THE CONSTANT
 4856  017700  005702                    TST     R2              ;;CHECK IF BCD DIGIT=0
 4857  017702  001002                    BNE     5$              ;;FALL THROUGH IF 0
 4858  017704  105716                    TSTB    (SP)            ;;STILL DOING LEADING 0'S?
 4859  017706  100407                    BMI     7$              ;;BR IF YES
 4860  017710  106316              5$:   ASLB    (SP)            ;;MSD?
 4861  017712  103003                    BCC     6$              ;;BR IF NO
 4862  017714  116663  000001  177777    MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
```

```
 4863  017722  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 4864  017726  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 4865  017732  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
 4866  017734  005720                          TST     (R0)+           ;;JUST INCREMENTING
 4867  017736  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
 4868  017742  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
 4869  017744  003002                          BGT     8$              ;;GO TO EXIT
 4870  017746  010502                          MOV     R5,R2           ;;GET THE LSD
 4871  017750  000764                          BR      6$              ;;GO CHANGE TO ASCII
 4872  017752  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
 4873  017754  100003                          BPL     9$              ;;BR IF NO
 4874  017756  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
 4875  017764  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
 4876  017766  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
 4877  017770  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
 4878  017772  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 4879  017774  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 4880  017776  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 4881  020000  104400  020026                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 4882  020004  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 4883  020012  012616                          MOV     (SP)+,(SP)
 4884  020014  000002                          RTI                     ;;RETURN TO USER
 4885  020016  023420                  $DTBL:  10000.
 4886  020020  001750                          1000.
 4887  020022  000144                          100.
 4888  020024  000012                          10.
 4889  020026  000004                  $DBLK:  .BLKW   4
 4890
 4891                                  .SBTTL  TYPE ROUTINE
 4892
 4893                                  ;;**********************************************************************
 4894                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 4895                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 4896                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 4897                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 4898                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 4899                                  ;*
 4900                                  ;*CALL:
 4901                                  ;*1) USING A TRAP INSTRUCTION
 4902                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 4903                                  ;*OR
 4904                                  ;*      TYPE
 4905                                  ;*      MESADR
 4906                                  ;*
 4907
 4908  020036  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 4909  020042  100002                          BPL     1$              ;;BR IF YES
 4910  020044  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
 4911  020046  000407                          BR      3$              ;;LEAVE
 4912  020050  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
 4913  020052  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 4914  020056  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 4915  020060  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 4916  020062  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 4917  020064  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
 4918  020066  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
```

```
4919  020072  000002                        RTI                     ;;RETURN
4920  020074  122716  000011        4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
4921  020100  001430                        BEQ     8$
4922  020102  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
4923  020106  001006                        BNE     5$
4924  020110  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
4925  020112  104400                        TYPE                    ;;TYPE A CR AND LF
4926  020114  001213                        $CRLF
4927  020116  105037  020252                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
4928  020122  000755                        BR      2$              ;;GET NEXT CHARACTER
4929  020124  004737  020206        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
4930  020130  123726  001156        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
4931  020134  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
4932  020136  013746  001154                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
4933                                                                ;;AND THE NULL CHAR.
4934  020142  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
4935  020146  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
4936  020150  004737  020206                JSR     PC,$TYPEC       ;;GO TYPE A NULL
4937  020154  105337  020252                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
4938  020160  000770                        BR      7$              ;;LOOP
4939
4940                                  ;HORIZONTAL TAB PROCESSOR
4941
4942  020162  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
4943  020166  004737  020206        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
4944  020172  132737  000007  020252        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
4945  020200  001372                        BNE     9$              ;;TAB STOP
4946  020202  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
4947  020204  000724                        BR      2$              ;;GET NEXT CHARACTER
4948  020206  105777  160736        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
4949  020212  100375                        BPL     $TYPEC
4950  020214  116677  000002  160730        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
4951  020222  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
4952  020230  001003                        BNE     1$              ;;BRANCH IF NO
4953  020232  105037  020252                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
4954  020236  000406                        BR      $TYPEX          ;;EXIT
4955  020240  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
4956  020246  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
4957  020250  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
4958  020252  000000        $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
4959  020254  000207        $TYPEX: RTS     PC
4960
4961
4962
4963                                  .SBTTL  INTEGER MULTIPLY ROUTINE
4964
4965                                  ;;*****************************************************************
4966                                  ;*CALL
4967                                  ;*      MOV     MULTIPLER,-(SP)
4968                                  ;*      MOV     MULTIPLICAND,-(SP)
4969                                  ;*      JSR     PC,@#$MULT
4970                                  ;*      RETURN  ;;PRODUCT IS ON THE STACK
4971                                  ;*
4972                                  ;*      STACK   PRODUCT
4973                                  ;*      -----   -------
4974                                  ;*      TOP     LSB'S
```

```
 4975                                   ;*      +2      MSB'S
 4976
 4977  020256                           $MULT:
 4978  020256  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 4979  020260  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 4980  020262  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 4981  020264  005046                          CLR     -(SP)           ;;CLEAR THE SIGN KEY
 4982  020266  016601  000012                  MOV     12(SP),R1       ;;GET THE MULTIPLICAND
 4983  020272  100002                          BPL     1$              ;;BR IF PLUS
 4984  020274  005216                          INC     (SP)            ;;SET THE SIGN KEY
 4985  020276  005401                          NEG     R1              ;;MAKE THE MULTIPLICAND POSTIVE
 4986  020300  016602  000014          1$:     MOV     14(SP),R2       ;;GET THE MULTIPLIER
 4987  020304  100002                          BPL     2$              ;;BR IF PLUS
 4988  020306  005316                          DEC     (SP)            ;;UPDATE THE SIGN KEY
 4989  020310  005402                          NEG     R2              ;;MAKE THE MULTIPLIER POSTIVE
 4990  020312  012746  000021          2$:     MOV     #17.,-(SP)      ;;SET THE LOOP COUNT
 4991  020316  005000                          CLR     R0              ;;SETUP FOR THE MULTIPLY LOOP
 4992  020320  103001                  3$:     BCC     4$              ;;DON'T ADD IF MULTIPLICAND = 0
 4993  020322  060200                          ADD     R2,R0
 4994  020324  006000                  4$:     ROR     R0              ;;POSITION THE PARITIAL PRODUCT AND
 4995  020326  006001                          ROR     R1              ;;THE MULTIPLICAND
 4996  020330  005316                          DEC     (SP)            ;;HAS ALL BITS OF THE MULTIPLICAND BEEN DONE?
 4997  020332  001372                          BNE     3$              ;;BR IF NO
 4998  020334  022616                          CMP     (SP)+,(SP)      ;;SHOULD PRODUCT BE NEGATIVE?
 4999  020336  001403                          BEQ     5$              ;;GO TO EXIT IF NO
 5000  020340  005400                          NEG     R0              ;;YES--SO MAKE IT SO
 5001  020342  005401                          NEG     R1
 5002  020344  005600                          SBC     R0
 5003  020346  005726                  5$:     TST     (SP)+           ;;CLEAR SIGN INFO. OFF OF STACK
 5004  020350  010066  000012                  MOV     R0,12(SP)       ;;PUT THE PRODUCT ON THE STACK (MSB'S)
 5005  020354  010166  000010                  MOV     R1,10(SP)       ;;LSB'S
 5006  020360  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 5007  020362  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 5008  020364  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 5009  020366  000207                          RTS     PC
 5010
 5011                                   .SBTTL  TTY INPUT ROUTINE
 5012
 5013                                   ;;**************************************************************************
 5014                                   .ENABL  LSB
 5015
 5016                                   ;;**************************************************************************
 5017                                   ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
 5018                                   ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
 5019                                   ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
 5020                                   ;*WHEN OPERATING IN TTY FLAG MODE.
 5021  020370  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
 5022  020376  001074                          BNE     15$             ;;BRANCH IF NO
 5023  020400  105777  160540                  TSTB    @$TKS           ;;CHAR THERE?
 5024  020404  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
 5025  020406  117746  160534                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
 5026  020412  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
 5027  020416  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
 5028  020422  001062                          BNE     15$             ;;NO, RETURN TO USER
 5029  020424  123727  001134  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
 5030  020432  001456                          BEQ     15$             ;;BRANCH IF YES
```

```
5031
5032  020434  104400  021115            TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (↑G)
5033  020440  104400  021122  $GTSWR:   TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
5034  020444  013746  000176            MOV     $WREG,-(SP)     ;;SAVE $WREG FOR TYPEOUT
5035  020450  104401                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
5036  020452  104400  021133            TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
5037  020456  005046            19$:    CLR     -(SP)           ;;CLEAR COUNTER
5038  020460  005046                    CLR     -(SP)           ;;THE NEW SWR
5039  020462  105777  160456    7$:     TSTB    @$TKS           ;;CHAR THERE?
5040  020466  100375                    BPL     7$              ;;IF NOT TRY AGAIN
5041
5042  020470  117746  160452            MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
5043  020474  042716  177600            BIC     #↑C177,(SP)     ;;MAKE IT 7-BIT ASCII
5044
5045
5046
5047  020500  021627  000025    9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
5048  020504  001005                    BNE     10$             ;;BRANCH IF NOT
5049  020506  104400  021110            TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (↑U)
5050  020512  062706  000006    20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
5051  020516  000757                    BR      19$             ;;LET'S TRY IT AGAIN
5052
5053
5054  020520  021627  000015    10$:    CMP     (SP),#15        ;;IS IT A <CR>?
5055  020524  001022                    BNE     16$             ;;BRANCH IF NO
5056  020526  005766  000004            TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
5057  020532  001403                    BEQ     11$             ;;BRANCH IF YES
5058  020534  016677  000002  160376    MOV     2(SP),@SWR      ;;SAVE NEW SWR
5059  020542  062706  000006    11$:    ADD     #6,SP           ;;CLEAR UP STACK
5060  020546  104400  001213    14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
5061  020552  123727  001135  000001    CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
5062  020560  001003                    BNE     15$             ;;BRANCH IF NOT
5063  020562  012777  000100  160354    MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
5064  020570  000002    15$:    RTI                             ;;RETURN
5065  020572  004737  020206    16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
5066  020576  021627  000060            CMP     (SP),#60        ;;CHAR < 0?
5067  020602  002420                    BLT     18$             ;;BRANCH IF YES
5068  020604  021627  000067            CMP     (SP),#67        ;;CHAR > 7?
5069  020610  003015                    BGT     18$             ;;BRANCH IF YES
5070  020612  042726  000060            BIC     #60,(SP)+       ;;STRIP-OFF ASCII
5071  020616  005766  000002            TST     2(SP)           ;;IS THIS THE FIRST CHAR
5072  020622  001403                    BEQ     17$             ;;BRANCH IF YES
5073  020624  006316                    ASL     (SP)            ;;NO, SHIFT PRESENT
5074  020626  006316                    ASL     (SP)            ;;    CHAR OVER TO MAKE
5075  020630  006316                    ASL     (SP)            ;;    ROOM FOR NEW ONE.
5076  020632  005266  000002    17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
5077  020636  056616  177776            BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
5078  020642  000707                    BR      7$              ;;GET THE NEXT ONE
5079  020644  104400  001212    18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
5080  020650  000720                    BR      20$             ;;SIMULATE CONTROL-U
5081                            .DSABL  LSB
5082
5083
5084                            ;;*****************************************************************
5085                            ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
5086                            ;*CALL:
```

```
5087                                    ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
5088                                    ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
5089                                    ;*                              ;;WITH PARITY BIT STRIPPED OFF
5090                                    ;
5091
5092  020652  011646                    $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
5093  020654  016666  000004  000002            MOV     4(SP),2(SP)     ;;SAVE THE PS
5094  020662  105777  160256            1$:     TSTB    @$TKS           ;;WAIT FOR
5095  020666  100375                            BPL     1$              ;;A CHARACTER
5096  020670  117766  160252  000004            MOVB    @$TKB,4(SP)     ;;READ THE TTY
5097  020676  042766  177600  000004            BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
5098  020704  026627  000004  000023            CMP     4(SP),#23       ;;IS IT A CONTROL-S?
5099  020712  001013                            BNE     3$              ;;BRANCH IF NO
5100  020714  105777  160224            2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
5101  020720  100375                            BPL     2$              ;;LOOP UNTIL ITS THERE
5102  020722  117746  160220                    MOVB    @$TKB,-(SP)     ;;GET CHARACTER
5103  020726  042716  177600                    BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
5104  020732  022627  000021                    CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
5105  020736  001366                            BNE     2$              ;;IF NOT DISCARD IT
5106  020740  000750                            BR      1$              ;;YES, RESUME
5107  020742  026627  000004  000140  3$:      CMP     4(SP),#140      ;;IS IT UPPER CASE?
5108  020750  002407                            BLT     4$              ;;BRANCH IF YES
5109  020752  026627  000004  000175            CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
5110  020760  003003                            BGT     4$              ;;BRANCH IF YES
5111  020762  042766  000040  000004            BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
5112  020770  000002                    4$:     RTI                     ;;GO BACK TO USER
5113                                    ;;**************************************************************
5114                                    ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
5115                                    ;*CALL:
5116                                    ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
5117                                    ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
5118                                    ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
5119
5120  020772  010346                    $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
5121  020774  012703  021100            1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
5122  021000  022703  021110            2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
5123  021004  101405                            BLOS    4$              ;;BR IF YES
5124  021006  104407                            RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
5125  021010  112613                            MOVB    (SP)+,(R3)      ;;GET CHARACTER
5126  021012  122713  000177            10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
5127  021016  001003                            BNE     3$              ;;SKIP IF NOT
5128  021020  104400  001212            4$:     TYPE    $QUES           ;;TYPE A '?'
5129  021024  000763                            BR      1$              ;;CLEAR THE BUFFER AND LOOP
5130  021026  111337  021076            3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
5131  021032  104400  021076                    TYPE    ,9$
5132  021036  122723  000015                    CMPB    #15,(R3)+       ;;CHECK FOR RETURN
5133  021042  001356                            BNE     2$              ;;LOOP IF NOT RETURN
5134  021044  105063  177777                    CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
5135  021050  104400  001214                    TYPE    $LF             ;;TYPE A LINE FEED
5136  021054  012603                            MOV     (SP)+,R3        ;;RESTORE R3
5137  021056  011646                            MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
5138  021060  016666  000004  000002            MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
5139  021066  012766  021100  000004            MOV     #$TTYIN,4(SP)
5140  021074  000002                            RTI                     ;;RETURN
5141  021076     000                    9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
5142  021077     000                            .BYTE   0               ;;TERMINATOR
```

```
5143  021100  000010                  $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
5144  021110  052536  005015    000   $CNTLU: .ASCIZ  /↑U/<15><12>            ;;CONTROL "U"
5145  021115     136  006507  000012  $CNTLG: .ASCIZ  /↑G/<15><12>            ;;CONTROL "G"
5146  021122  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
5147  021130  020075     000
5148  021133     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
5149  021140  036440  000040
5150
5151                                   .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
5152
5153                                   ;;**********************************************************************
5154                                   ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
5155                                   ;*CHANGE IT TO BINARY.
5156                                   ;*CALL:
5157                                   ;*      RDOCT                           ;;READ AN OCTAL NUMBER
5158                                   ;*      RETURN HERE                     ;;LOW ORDER BITS ARE ON TOP OF THE STACK
5159                                   ;*                                      ;;HIGH ORDER BITS ARE IN $HIOCT
5160
5161  021144  011646                  $RDOCT: MOV     (SP),-(SP)              ;;PROVIDE SPACE FOR THE
5162  021146  016666  000004  000002          MOV     4(SP),2(SP)             ;;INPUT NUMBER
5163  021154  010046                          MOV     R0,-(SP)                ;;PUSH R0 ON STACK
5164  021156  010146                          MOV     R1,-(SP)                ;;PUSH R1 ON STACK
5165  021160  010246                          MOV     R2,-(SP)                ;;PUSH R2 ON STACK
5166  021162  104410                  1$:     RDLIN                           ;;READ AN ASCIZ LINE
5167  021164  012600                          MOV     (SP)+,R0                ;;GET ADDRESS OF 1ST CHARACTER
5168  021166  005001                          CLR     R1                      ;;CLEAR DATA WORD
5169  021170  005002                          CLR     R2
5170  021172  112046                  2$:     MOVB    (R0)+,-(SP)             ;;PICKUP THIS CHARACTER
5171  021174  001412                          BEQ     3$                      ;;IF ZERO GET OUT
5172  021176  006301                          ASL     R1                      ;;*2
5173  021200  006102                          ROL     R2
5174  021202  006301                          ASL     R1                      ;;*4
5175  021204  006102                          ROL     R2
5176  021206  006301                          ASL     R1                      ;;*8
5177  021210  006102                          ROL     R2
5178  021212  042716  177770                BIC     #↑C7,(SP)               ;;STRIP THE ASCII JUNK
5179  021216  062601                          ADD     (SP)+,R1                ;;ADD IN THIS DIGIT
5180  021220  000764                          BR      2$                      ;;LOOP
5181  021222  005726                  3$:     TST     (SP)+                   ;;CLEAN TERMINATOR FROM STACK
5182  021224  010166  000012                  MOV     R1,12(SP)               ;;SAVE THE RESULT
5183  021230  010237  021244                  MOV     R2,$HIOCT
5184  021234  012602                          MOV     (SP)+,R2                ;;POP STACK INTO R2
5185  021236  012601                          MOV     (SP)+,R1                ;;POP STACK INTO R1
5186  021240  012600                          MOV     (SP)+,R0                ;;POP STACK INTO R0
5187  021242  000002                          RTI                             ;;RETURN
5188  021244  000000                  $HIOCT: .WORD   0                       ;;HIGH ORDER BITS GO HERE
5189
5190                                   .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
5191
5192                                   ;;**********************************************************************
5193                                   ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
5194                                   ;*OCTAL (ASCII) NUMBER AND TYPE IT.
5195                                   ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
5196                                   ;*CALL:
5197                                   ;*      MOV     NUM,-(SP)               ;;NUMBER TO BE TYPED
5198                                   ;*      TYPOS                           ;;CALL FOR TYPEOUT
```

```
5199                                    ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
5200                                    ;*      .BYTE   M               ;;M=1 OR 0
5201                                    ;*                              ;;1=TYPE LEADING ZEROS
5202                                    ;*                              ;;0=SUPPRESS LEADING ZEROS
5203                                    ;*
5204                                    ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
5205                                    ;*$TYPOS OR $TYPOC
5206                                    ;*CALL:
5207                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
5208                                    ;*      TYPON                   ;;CALL FOR TYPEOUT
5209                                    ;*
5210                                    ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
5211                                    ;*CALL:
5212                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
5213                                    ;*      TYPOC                   ;;CALL FOR TYPEOUT
5214
5215  021246  017646  000000            $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
5216  021252  116637  000001  021471            MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
5217  021260  112637  021473                    MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
5218  021264  062716  000002                    ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
5219  021270  000406                            BR      $TYPON
5220  021272  112737  000001  021471  $TYPOC:  MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
5221  021300  112737  000006  021473          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
5222  021306  112737  000005  021470  $TYPON:  MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
5223  021314  010346                            MOV     R3,-(SP)        ;;SAVE R3
5224  021316  010446                            MOV     R4,-(SP)        ;;SAVE R4
5225  021320  010546                            MOV     R5,-(SP)        ;;SAVE R5
5226  021322  113704  021473                    MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
5227  021326  005404                            NEG     R4
5228  021330  062704  000006                    ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
5229  021334  110437  021472                    MOVB    R4,$OMODE       ;;SAVE IT FOR USE
5230  021340  113704  021471                    MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
5231  021344  016605  000012                    MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
5232  021350  005003                            CLR     R3              ;;CLEAR THE OUTPUT WORD
5233  021352  006105                    1$:     ROL     R5              ;;ROTATE MSB INTO "C"
5234  021354  000404                            BR      3$              ;;GO DO MSB
5235  021356  006105                    2$:     ROL     R5              ;;FORM THIS DIGIT
5236  021360  006105                            ROL     R5
5237  021362  006105                            ROL     R5
5238  021364  010503                            MOV     R5,R3
5239  021366  006103                    3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
5240  021370  105337  021472                    DECB    $OMODE          ;;TYPE THIS DIGIT?
5241  021374  100016                            BPL     7$              ;;BR IF NO
5242  021376  042703  177770                    BIC     #177770,R3      ;;GET RID OF JUNK
5243  021402  001002                            BNE     4$              ;;TEST FOR 0
5244  021404  005704                            TST     R4              ;;SUPPRESS THIS 0?
5245  021406  001403                            BEQ     5$              ;;BR IF YES
5246  021410  005204                    4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
5247  021412  052703  000060                    BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
5248  021416  052703  000040            5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
5249  021422  110337  021466                    MOVB    R3,8$           ;;SAVE FOR TYPING
5250  021426  104400  021466                    TYPE    ,8$             ;;GO TYPE THIS DIGIT
5251  021432  105337  021470            7$:     DECB    $OCNT           ;;COUNT BY 1
5252  021436  003347                            BGT     2$              ;;BR IF MORE TO DO
5253  021440  002402                            BLT     6$              ;;BR IF DONE
5254  021442  005204                            INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
```

```
5255  021444  000744                        BR      2$              ;;GO DO THE LAST DIGIT
5256  021446  012605              6$:       MOV     (SP)+,R5        ;;RESTORE R5
5257  021450  012604                        MOV     (SP)+,R4        ;;RESTORE R4
5258  021452  012603                        MOV     (SP)+,R3        ;;RESTORE R3
5259  021454  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
5260  021462  012616                        MOV     (SP)+,(SP)
5261  021464  000002                        RTI                     ;;RETURN
5262  021466     000              8$:       .BYTE   0               ;;STORAGE FOR ASCII DIGIT
5263  021467     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
5264  021470     000              $OCNT:    .BYTE   0               ;;OCTAL DIGIT COUNTER
5265  021471     000              $OFILL:   .BYTE   0               ;;ZERO FILL SWITCH
5266  021472  000000              $OMODE:   .WORD   0               ;;NUMBER OF DIGITS TO TYPE
5267
5268
5269                                        .SBTTL  TYPDSS - TYPE DECIMAL, LEADING ZEROES SUPPRESSED
5270                              ;TYPDSS
5271                              ;ROUTINE FOR TYPING OUT DECIMAL NUMBERS, LEADING 0'S ARE SUPPRESSED
5272                              ;THE NUMBER IS LEFT JUSTIFIED. NOTE THE 16 BIT BINARY NUMBER SHOULD
5273                              ;BE POSITIVE (BIT 15= 0).
5274                              ;CALL:  MOV     NUMBER,-(SP)    ;PUT BINARY NUMBER ON STACK
5275                              ;       TYPDSS                  ;GO TYPE DECIMAL
5276
5277  021474  016637  000004  021534  TYPDES: MOV     4(SP),1$                ;GET THE NUMBER
5278  021502  012746  021534                 MOV     #1$,-(SP)               ;PUT PTR ON THE STACK
5279  021506  004737  021674                 JSR     PC,@#$DB2D              ;GO CONVERT BINARY NO. TO
5280                                                                           ;ASCII STRING
5281  021512  004737  021540                 JSR     PC,@#$SUPRS             ;GO TYPE OUT DECIMAL STRING
5282                                                                           ;SUPRESING LEADING 0'S
5283  021516  016666  000002  000004        MOV     2(SP),4(SP)             ;ADJUST RETURN
5284  021524  011666  000002                 MOV     (SP),2(SP)              ;ADJUST RETURN ADRES
5285  021530  005726                         TST     (SP)+                   ;POP STACK
5286  021532  000002                         RTI                             ;RETURN
5287
5288  021534  000000  000000          1$:    .WORD   0,0
5289
5290
5291                                        .SBTTL  TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS
5292
5293                              ;;*****************************************************************
5294                              ;*THIS ROUTINE IS USED TO TYPE AN ASCIZ NUMBER SUPPRESSING THE
5295                              ;*LEADING NUMBERS.
5296                              ;*CALL
5297                              ;*      MOV     #NUMADR,-(SP)   ;;FIRST ADDRESS OF ASCIZ STRING
5298                              ;*      JSR     PC,@#$SUPRS
5299
5300
5301  021540  010046          $SUPRS: MOV     R0,-(SP)        ;;SAVE R0
5302  021542  016600  000004          MOV     4(SP),R0        ;;PICKUP THE POINTER
5303  021546  105710          1$:     TSTB    (R0)            ;;TERMINATEOR?
5304  021550  001403                  BEQ     2$              ;;BR IF YES
5305  021552  122720  000060          CMPB    #'0,(R0)+       ;;IS THIS AN ASCII "0" ?
5306  021556  001773                  BEQ     1$              ;;BR IF YES
5307  021560  005300          2$:     DEC     R0              ;;BACKUP BY "1"
5308  021562  010037  021570          MOV     R0,3$           ;;SAVE FOR TYPING
5309  021566  104400                  TYPE                    ;;GO TYPE
5310  021570  000000          3$:     .WORD   0               ;;ASCIZ POINTER GOES HERE
```

```
                                        MOV     (SP)+,R0        ;;RESTORE R0
                                        MOV     (SP)+,(SP)      ;;RESTORE THE STACK
                                        RTS     PC              ;;RETURN

                                .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

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

021600                          $SAVREG:
021600  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
021602  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
021604  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
021606  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
021610  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
021612  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
021614  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
021620  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
021624  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
021630  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
021634  000002                          RTI

                                ;*RESTORE R0-R5
                                ;*CALL:
                                ;*      RESREG
021636                          $RESREG:
021636  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
021642  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
021646  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
021652  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
021656  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
021660  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
021662  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
021664  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
021666  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
021670  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
021672  000002                          RTI

                                .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE

                                ;;**************************************************************************
                                ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
                                ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
                                ;*POSITIVE.
```

```
                                        ;*CALL
                                        ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
                                        ;*      JSR     PC,@#$DB2D
                                        ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
                                                                        ;;IS ON THE STACK


021674  104412                  $DB2D:  SAVREG                          ;;SAVE REGISTERS
021676  016602  000002                  MOV     2(SP),R2                ;;PICKUP THE DATA POINTER
021702  012700  022054                  MOV     #$DECVL,R0              ;;GET ADDRESS OF "$DECVL" STRING
021706  010066  000002                  MOV     R0,2(SP)                ;;PUT ADDRESS OF ASCIZ STRING ON STACK
021712  012201                          MOV     (R2)+,R1                ;;PICKUP THE BINARY NUMBER
021714  012202                          MOV     (R2)+,R2
021716  012737  000012  021772          MOV     #10.,4$                 ;;SET UP TO DO 10 CONVERSIONS
021724  012704  022004                  MOV     #$TNPWR,R4              ;;ADDRESS OF TEN POWER
021730  012705  022006                  MOV     #$TNPWR+2,R5
021734  005003                  1$:     CLR     R3                      ;;CLEAR PARTIAL
021736  161401                  2$:     SUB     (R4),R1                 ;;SUBTRACT TEN POWER
021740  005602                          SBC     R2
021742  161502                          SUB     (R5),R2
021744  002402                          BLT     3$                      ;;BR IF TEN POWER TO LARGE
021746  005203                          INC     R3                      ;;ADD 1 TO PARTIAL
021750  000772                          BR      2$                      ;;LOOP
021752  062401                  3$:     ADD     (R4)+,R1                ;;RESTORE SUBTRACTED VALUE
021754  005502                          ADC     R2
021756  062402                          ADD     (R4)+,R2
021760  022525                          CMP     (R5)+,(R5)+             ;;MOVE TO NEXT TEN POWER
021762  052703  000060                  BIS     #'0,R3                  ;;CHANGE PARTIAL TO ASCII
021766  110320                          MOVB    R3,(R0)+                ;;SAVE IT
021770  005327                          DEC     (PC)+                   ;;DONE?
021772  000000                  4$:     .WORD   0
021774  001357                          BNE     1$                      ;;BR IF NO
021776  105020                          CLRB    (R0)+                   ;;TERMINATOR
022000  104413                          RESREG                          ;;RESTORE REGISTERS
022002  000207                          RTS     PC                      ;;RETURN
022004  145000                  $TNPWR: 145000                          ;;1.0E09
022006  035632                          35632
022010  160400                          160400                          ;;1.0E08
022012  002765                          2765
022014  113200                          113200                          ;;1.0E07
022016  000230                          230
022020  041100                          041100                          ;;1.0E06
022022  000017                          17
022024  103240                          103240                          ;;1.0E05
022026  000001                          1
022030  023420                          23420                           ;;1.0E04
022032  000000                          0
022034  001750                          1750                            ;;1.0E03
022036  000000                          0
022040  000144                          144                             ;;1.0E02
022042  000000                          0
022044  000012                          12                              ;;1.0E01
022046  000000                          0
022050  000001                          1                               ;;1.0E00
022052  000000                          0
022054  000014                  $DECVL: .BLKB   12.                     ;;RESERVE STORAGE FOR ASCIZ STRING
```

```
                                        .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.

022070  010046              $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
022072  016600  000002              MOV     2(SP),R0        ;;GET TRAP ADDRESS
022076  005740                      TST     -(R0)           ;;BACKUP BY 2
022100  111000                      MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
022102  006300                      ASL     R0              ;;POSITION FOR INDEXING
022104  016000  022112              MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
022110  000200                      RTS     R0              ;;GO TO ROUTINE

                                        .SBTTL  TRAP TABLE

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

                                        ;       ROUTINE
                                        ;       -------
022112                          $TRPAD:
022112  020036                          $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
022114  021272                          $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
022116  021246                          $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
022120  021306                          $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
022122  017612                          $TYPDS  ;;CALL=TYPDS    TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)

022124  020440                          $GTSWR  ;;CALL=GTSWR    TRAP+5(104405)  GET SOFT-SWR SETTING

022126  020370                          $CKSWR  ;;CALL=CKSWR    TRAP+6(104406)  TEST FOR CHANGE IN SOFT-SWR
022130  020652                          $RDCHR  ;;CALL=RDCHR    TRAP+7(104407)  TTY TYPEIN CHARACTER ROUTINE
022132  020772                          $RDLIN  ;;CALL=RDLIN    TRAP+10(104410) TTY TYPEIN STRING ROUTINE
022134  021144                          $RDOCT  ;;CALL=RDOCT    TRAP+11(104411) READ AN OCTAL NUMBER FROM TTY
022136  021600                          $SAVREG ;;CALL=SAVREG   TRAP+12(104412) SAVE R0-R5 ROUTINE
022140  021636                          $RESREG ;;CALL=RESREG   TRAP+13(104413) RESTORE R0-R5 ROUTINE

022142  016374                          CN.RST  ;;CALL=CON.RESET        TRAP+14(104414) CONTROL RESET ROUTINE

022144  016262                          DR.RST  ;;CALL=DRV.RESET        TRAP+15(104415) DRIVE RESET ROUTINE

022146  015636                          MSGE    ;;CALL=MESAGE   TRAP+16(104416) MESSAGE HANDLER

022150  015700                          TY.MSG  ;;CALL=TYPMSG   TRAP+17(104417) MESSAGE TYPEOUT ROUTINE

022152  016412                          CN.RDY  ;;CALL=CON.RDY  TRAP+20(104420) WAIT FOR CONTROL READY

022154  016510                          TSTRWS  ;;CALL=TST.RWS  TRAP+21(104421) TEST R/W/S RDY SET

022156  016306                          RES.DO  ;;CALL=RESDON   TRAP+22(104422) DRIVE RESET DONE?
```

```
5479  022160  021474                          TYPDES  ;;CALL=TYPDSS   TRAP+23(104423) TYPE DECIMAL, SUPRES LDG 0'S
5480
5481
5482                                  .SBTTL  POWER DOWN AND UP ROUTINES
5483
5484                                  ;;*****************************************************************
5485                                  ;POWER DOWN ROUTINE
5486  022162  012737  022326  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
5487  022170  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
5488  022176  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
5489  022200  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
5490  022202  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
5491  022204  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
5492  022206  010446                         MOV     R4,-(SP)        ;;PUSH R4 ON STACK
5493  022210  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
5494  022212  017746  156722                 MOV     @$WR,-(SP)      ;;PUSH @$WR ON STACK
5495  022216  010637  022332                 MOV     SP,$SAVR6       ;;SAVE SP
5496  022222  012737  022234  000024         MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
5497  022230  000000                         HALT
5498  022232  000776                         BR      .-2             ;;HANG UP
5499
5500                                  ;;*****************************************************************
5501                                  ;POWER UP ROUTINE
5502  022234  012737  022326  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
5503  022242  013706  022332                 MOV     $SAVR6,SP       ;;GET SP
5504  022246  005037  022332                 CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
5505  022252  005237  022332          1$:    INC     $SAVR6          ;;WAIT FOR THE INC
5506  022256  001375                         BNE     1$              ;;OF  WORD
5507  022260  012677  156654                 MOV     (SP)+,@$WR      ;;POP STACK INTO @$WR
5508  022264  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
5509  022266  012604                         MOV     (SP)+,R4        ;;POP STACK INTO R4
5510  022270  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
5511  022272  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
5512  022274  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
5513  022276  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
5514  022300  012737  022162  000024         MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
5515  022306  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;;PRIO:7
5516  022314  104400                         TYPE                    ;REPORT THE POWER FAILURE
5517  022316  022334                 $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
5518  022320  012716                         MOV     (PC)+,(SP)      ;;RESTART AT PWRFL
5519  022322  004026                 $PWRAD: .WORD   PWRFL            ;;RESTART ADDRESS
5520  022324  000002                         RTI
5521  022326  000000                 $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
5522  022330  000776                         BR      .-2             ;;  BEFORE THE POWER DOWN WAS COMPLETE
5523  022332  000000                 $SAVR6: 0                       ;;PUT THE SP HERE
5524  022334  005015  047520  042527 $POWER: .ASCIZ  <15><12>"POWER"
5525  022342  000122
5526                                          .EVEN
5527
5528
```

```
                        .SBTTL  FUNCTION SELECTION PROGRAM

                ;THIS IS THE FUNCTION SELECTION PROGRAM.
                ;ON ENTERING THIS SUB-PROGRAM THE FIRST QUESTION ASKED IS
                ;       FUNCTION?       IN REPLY TYPE IN ONE OF THE FOLLOWING COMMANDS.
                ;COMMANDS:      CR - CONTROL RESET
                ;               DR - DRIVE RESET
                ;               SK - SEEK
                ;               WR - WRITE
                ;               RD - READ
                ;               WC - WRITE CHECK
                ;               RC - READ CHECK
                ;TERMINATE EVERY COMMAND WITH <CR>. FURTHER QUESTIONS (RKBA?   RKDA?
                ;#WORDS? ) WILL BE ASKED. TYPE IN THE BUS ADDRESS (OCTAL), DISK ADDRESS
                ;(OCTAL), AND NUMBER OF WORDS TO BE TRANSFERRED (OCTAL). IF A NON-EXISTENT
                ;CYLINDER OR A NON-EXISTENT SECTOR IS TYPED IN, THE QUESTION (RKDA?) WILL
                ;BE ASKED AGAIN.

                ;IN CASE OF SEEK FUNCTION, SEEK IS DONE BETWEEN A SET OF CYLINDERS GIVEN
                ;BY THE USER  (CYL1? , CYL2?). IN REPLY TO (CYL1?, CYL2?) TYPE IN THE OCTAL
                ;CYLINDER NUMBERS BETWEEN WHICH THE SEEK IS TO BE DONE. SET SWITCH
                ;REGISTER BITS <15-13> TO THE DRIVE # ON WHICH SEEK IS TO BE DONE.

                ;IN CASE OF A WRITE FUNCTION IF SW 0 IS SET TO 1 THE PROGRAM WILL ASK
                ;THE USER FOR THE PATTERN TO BE WRITTEN:
                ;       PATRN?125252<CR>
                ;THE USER SHOULD TYPE IN THE (OCTAL) PATTERN HE WANTS TO WRITE.
                ;NOTE THAT THE NUMBER OF WORDS TRANSFERRED SHOULD BE WITHIN THE
                ;BOUNDS OF THE SYSTEM.

                ;IN CASE OF A WRITE CHECK FUNCTION IF SW0 IS SET TO 1 THE PROGRAM
                ;WILL ASK THE USER FOR THE PATTERN TO BE WRITE CHECKED:
                ;       PATRN?125252<CR>
                ;THE USER SHOULD TYPE IN THE (OCTAL) PATTERN TO BE WRITE CHECKED.

                ;LOCATIONS "IOBUF0" ONWARDS  ARE RESERVED FOR DATA BUFFERS. YOU CAN USE
                ;THESE LOCATIONS FOR DOING DATA TRANSFERS IN THIS PROGRAM (OR YOU CAN
                ;USE ANY OTHER BUFFER FOR DATA TRANSFER AS LONG AS IT DOES NOT OVERLAY
                ;THE PROGRAM).

                ;THE SAME FUNCTION IS PERFORMED AGAIN AND AGAIN, THUS PROVIDING
                ;A VERY GOOD SCOPE LOOP. IF YOU WANT TO GIVE A NEW FUNCTION PUT SW 3 UP.
                ;THE QUESTION (FUNCTION?) WILL BE ASKED AND YOU CAN START ALL OVER AGAIN.
                ;IF ON EXECUTING A FUNCTION AN ERROR OCCURS, IT IS REPOTRED , WITH
                ; RELEVANT REGISTER CONTENTS GIVEN.

                                                ;R2 CONTAINS RKCS CONTENTS
                                                ;R3 CONTAINS RKDA CONTENTS
                                                ;R4,R5 CONTAIN THE CYLINDER #S BETWEEN
                                                ;WHICH SEEK IS  TO BE DONE.
022344          FUNBEG:
022344  104400  022352          TYPE    ,65$            ;;TYPE ASCIZ STRING
022350  000407                  BR      64$             ;;GET OVER THE ASCIZ
                ;;65$:  .ASCIZ  <15><12>/FUNCTION? /
022370          64$:
022370  104410                  RDLIN
```

```
5585  022372  012600                        MOV     (SP)+,R0
5586  022374  112001                        MOVB    (R0)+,R1
5587  022376  120127  000127                CMPB    R1,#'W
5588  022402  001026                        BNE     2$
5589  022404  121027  000122                CMPB    (R0),#'R
5590  022410  001010                        BNE     1$
5591  022412  004737  023236                JSR     PC,CHKSW0       ;CHECK SW 0 SET?
5592  022416  012702  004003                MOV     #4003,R2
5593  022422  000536                        BR      NXTDA
5594
5595  022424  012702  000003        9$:     MOV     #3,R2
5596  022430  000521                        BR      NXTBA
5597  022432  121027  000103        1$:     CMPB    (R0),#'C
5598  022436  001342                        BNE     FUNBEG
5599  022440  004737  023236                JSR     PC,CHKSW0       ;CHECK SW 0 SET?
5600  022444  012702  004007                MOV     #4007,R2
5601  022450  000523                        BR      NXTDA
5602
5603  022452  012702  000007                MOV     #7,R2
5604  022456  000506                        BR      NXTBA
5605
5606  022460  120127  000122        2$:     CMPB    R1,#'R
5607  022464  001014                        BNE     3$
5608  022466  121027  000104                CMPB    (R0),#'D
5609  022472  001003                        BNE     8$
5610  022474  012702  000005                MOV     #5,R2
5611  022500  000475                        BR      NXTBA
5612  022502  121027  000103        8$:     CMPB    (R0),#'C
5613  022506  001316                        BNE     FUNBEG
5614  022510  012702  000013                MOV     #13,R2
5615  022514  000501                        BR      NXTDA
5616
5617  022516  120127  000104        3$:     CMPB    R1,#'D
5618  022522  001006                        BNE     4$
5619  022524  121027  000122                CMPB    (R0),#'R
5620  022530  001305                        BNE     FUNBEG
5621  022532  012702  000015                MOV     #15,R2
5622  022536  000470                        BR      NXTDA
5623
5624  022540  120127  000103        4$:     CMPB    R1,#'C
5625  022544  001006                        BNE     5$
5626  022546  121027  000122                CMPB    (R0),#'R
5627  022552  001274                        BNE     FUNBEG
5628  022554  012702  000001                MOV     #1,R2
5629  022560  000533                        BR      EXEC
5630
5631  022562  120127  000123        5$:     CMPB    R1,#'S
5632  022566  001266                        BNE     FUNBEG
5633  022570  121027  000113                CMPB    (R0),#'K
5634  022574  001263                        BNE     FUNBEG
5635  022576  012702  000011                MOV     #11,R2
5636
5637  022602                        6$:
5638  022602  104400  022610                TYPE    ,67$            ;;TYPE ASCIZ STRING
5639  022606  000404                        BR      66$             ;;GET OVER THE ASCIZ
5640                                ;;67$:  .ASCIZ  /CYL1? /
```

```
 5641  022620                          66$:
 5642  022620  004737  023204                  JSR     PC,INPT
 5643  022624  000766                          BR      6$
 5644  022626  010004                          MOV     R0,R4
 5645
 5646  022630                          7$:
 5647  022630  104400  022636                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 5648  022634  000404                          BR      68$             ;;GET OVER THE ASCIZ
 5649                                  ;;69$:  .ASCIZ  /CYL2? /
 5650  022646                          68$:
 5651  022646  004737  023204                  JSR     PC,INPT
 5652  022652  000766                          BR      7$
 5653  022654  010005                          MOV     R0,R5
 5654  022656  017700  156256                  MOV     @SWR,R0         ;GET DRIVE # FROM SW REG<15-13>
 5655  022662  042700  017777                  BIC     #17777,R0               ;CLR UNWANTED BITS
 5656  022666  050004                          BIS     R0,R4                   ;SET DRIVE # IN DSK ADRES
 5657  022670  050005                          BIS     R0,R5
 5658  022672  000466                          BR      EXEC
 5659
 5660
 5661  022674                          NXTBA:
 5662  022674  104400  022702                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 5663  022700  000404                          BR      64$             ;;GET OVER THE ASCIZ
 5664                                  ;;65$:  .ASCIZ  /RKBA? /
 5665  022712                          64$:
 5666  022712  104411                          RDOCT
 5667  022714  012637  001774                  MOV     (SP)+,INDX2
 5668
 5669  022720                          NXTDA:
 5670  022720  104400  022726                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 5671  022724  000404                          BR      64$             ;;GET OVER THE ASCIZ
 5672                                  ;;65$:  .ASCIZ  /RKDA? /
 5673  022736                          64$:
 5674  022736  104411                          RDOCT
 5675  022740  012600                          MOV     (SP)+,R0
 5676  022742  010001                          MOV     R0,R1
 5677  022744  006201                          ASR     R1
 5678  022746  006201                          ASR     R1
 5679  022750  006201                          ASR     R1
 5680  022752  006201                          ASR     R1
 5681  022754  006201                          ASR     R1
 5682  022756  042701  177400                  BIC     #177400,R1
 5683  022762  020127  000312                  CMP     R1,#312
 5684  022766  003354                          BGT     NXTDA
 5685  022770  010001                          MOV     R0,R1
 5686  022772  042701  177760                  BIC     #177760,R1
 5687  022776  020127  000013                  CMP     R1,#13
 5688  023002  003346                          BGT     NXTDA
 5689  023004  010003                          MOV     R0,R3
 5690  023006  022702  000015                  CMP     #15,R2
 5691  023012  001416                          BEQ     EXEC
 5692
 5693
 5694  023014                          NXTWC:
 5695  023014  104400  023022                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 5696  023020  000405                          BR      64$             ;;GET OVER THE ASCIZ
```

```
5697                                        ;;65$:  .ASCIZ  /#WORDS? /
5698  023034                                64$:
5699  023034  104411                              RDOCT
5700  023036  005416                              NEG     (SP)
5701  023040  012637  002000                      MOV     (SP)+,INDX4
5702  023044  000401                              BR      EXEC
5703
5704  023046  104414                    EXEC1:    CON.RESET                       ;CLR EROR, CONTROL RESET
5705  023050  022702  000011            EXEC:     CMP     #11,R2                  ;SEEK FUNCTION?
5706  023054  001005                              BNE     2$                      ;NO
5707  023056  020403                              CMP     R4,R3                   ;IF SEEK, INSERT THE RIGHT
5708  023060  001402                              BEQ     3$                  ;CYLINDER ADDRESS
5709  023062  010403                              MOV     R4,R3
5710  023064  000401                              BR      2$
5711  023066  010503                    3$:       MOV     R5,R3
5712  023070  013777  001774  156662    2$:       MOV     INDX2,@RKBA
5713  023076  010377  156660                      MOV     R3,@RKDA
5714  023102  013777  002000  156646              MOV     INDX4,@RKWC
5715  023110  010277  156640                      MOV     R2,@RKCS
5716  023114  105777  156634            1$:       TSTB    @RKCS               ;WAIT FOR CNTROL RDY
5717  023120  100375                              BPL     1$
5718  023122  022702  000001                      CMP     #1,R2               ;IF IT'S CON RESET FUNCTION
5719  023126  001401                              BEQ     4$                  ;DONT WAIT FOR R/W/S RDY
5720  023130  104422                              RESDON                      ;R/W/S RDY?
5721
5722  023132  032777  140000  156614    4$:       BIT     #140000,@RKCS       ;ERROR?
5723  023140  001006                              BNE     FUNERR              ;YES
5724  023142  032777  000010  155770    CHSW:     BIT     #SW3,@SWR           ;TERMINATE THIS FUNCTION OR REPEAT?
5725  023150  001737                              BEQ     EXEC                ;REPEAT
5726  023152  000137  022344                      JMP     FUNBEG              ;TERMINATE
5727
5728
5729  023156  012737  023046  001110    FUNERR:   MOV     #EXEC1,$LPERR       ;SET UP FOR LUPING
5730  023164  012737  023046  001106              MOV     #EXEC1,$LPADR
5731  023172  004737  015754                      JSR     PC,GT4RG
5732  023176  104030                              ERROR   30                  ;REPORT ERROR
5733  023200  104414                              CON.RESET                   ;CLR ERROR
5734  023202  000757                              BR      CHSW
5735
5736  023204  104411                    INPT:     RDOCT
5737  023206  012600                              MOV     (SP)+,R0
5738  023210  020027  000312                      CMP     R0,#312
5739  023214  003007                              BGT     1$
5740  023216  006300                              ASL     R0
5741  023220  006300                              ASL     R0
5742  023222  006300                              ASL     R0
5743  023224  006300                              ASL     R0
5744  023226  006300                              ASL     R0
5745  023230  062716  000002                      ADD     #2,(SP)
5746  023234  000207                    1$:       RTS     PC
5747
5748  023236  032777  000001  155674    CHKSWO:   BIT     #SW0,@SWR           ;WRITE A PATTERN GIVEN BY USER?
5749  023244  001416                              BEQ     1$                  ;NO
5750                                                                        ;YES, ASK FOR PATTERN
5751  023246  104400  023254                      TYPE    ,65$                ;;TYPE ASCIZ STRING
5752  023252  000404                              BR      64$                 ;;GET OVER THE ASCIZ
```

```
5753                                        ;;65$:  .ASCIZ  /PATRN?/
5754  023264                                64$:
5755  023264  104411                              RDOCT
5756  023266  012637  030620                      MOV     (SP)+,IOBUF1    ;SAVE THE PATTERN
5757  023272  012737  030620  001774              MOV     #IOBUF1,INDX2
5758  023300  000207                              RTS     PC
5759  023302  062716  000006              1$:     ADD     #6,(SP)         ;SKIP NXT 2 INST ON RETURN
5760  023306  000207                              RTS     PC
5761
5762  023310  104415              FCHECK: DRV.RESET
5763  023312  104414                      CON.RESET
5764  023314  013737  001526  002416              MOV     DRIVAD,DRHOLD           ;SAVE DRIVE ADRR
5765  023322  032737  020000  001526              BIT     #20000,DRIVAD           ;SEE IF ODD
5766  023330  001404                              BEQ     1$
5767  023332  042737  020000  001526              BIC     #20000,DRIVAD           ;MAKE EVEN
5768  023340  000403                              BR      2$
5769  023342  052737  020000  001526      1$:     BIS     #20000,DRIVAD           ;MAKE ODD
5770  023350  013777  001526  156404      2$:     MOV     DRIVAD,@RKDA    ;DRIVE ADDR
5771  023356  012777  000011  156370              MOV     #11,@RKCS               ;DRIVE SEEK
5772  023364  104420                              CON.RDY
5773  023366  013777  002416  156366              MOV     DRHOLD,@RKDA            ;OTHER DRIVE
5774  023374  104420                              CON.RDY
5775  023376  032777  000100  156344              BIT     #100,@RKDS              ;HEAEDS IN MOTIONN?
5776  023404  001001                              BNE     3$                      ;NO SO RK-05J
5777  023406  005725                              TST     (R5)+                   ;YES RK-05F
5778  023410  013737  002416  001526      3$:     MOV     DRHOLD,DRIVAD           ;RESTORE ADDR
5779  023416  104415                      DRV.RESET
5780  023420  000205                              RTS     R5
5781  023422  005037  001526              SIZEF:  CLR     DRIVAD          ;START AT DRO
5782  023426  012700  001530                      MOV     #DRIVO,R0               ;TABLE OF AVAIL DRIVES
5783  023432  105710                      4$:     TSTB    (R0)                    ;THIS DRIVE HERE?
5784  023434  001413                              BEQ     2$                      ;NO
5785  023436  105760  000002                      TSTB    2(R0)                   ;COMPLEMENT HERE?
5786  023442  001410                              BEQ     2$                      ;NO
5787  023444  004537  023310                      JSR     R5,FCHECK               ;SEE IF F MODEL
5788  023450  000405                              BR      2$                      ;J MODEL
5789  023452  052710  000002                      BIS     #2,(R0)         ;SET SIGN FOR F
5790  023456  052760  000002  000002              BIS     #2,2(R0)                ;BOTH DRIVES
5791  023464  005720                      2$:     TST     (R0)+
5792  023466  005720                              TST     (R0)+                   ;NEXT PAIR OF DRIVES
5793  023470  062737  040000  001526              ADD     #40000,DRIVAD           ;NEXT ACTUL ADDR
5794  023476  022700  001547                      CMP     #DRIV7+1,R0             ;CHECKED ALL?
5795  023502  003353                              BGT     4$                      ;NOT YET
5796  023504  000207                              RTS     PC
5797
5798                                      ;ERROR MESSAGES
5799
5800
5801  023506  047103  051124  020114  EM1:    .ASCIZ  /CNTRL RDY DIDN'T SET AFTR SEEK/
5802  023514  042122  020131  044504
5803  023522  047104  052047  051440
5804  023530  052105  040440  052106
5805  023536  020122  042523  045505
5806  023544     000
5807
5808  023545     123  047111  047440  EM2:    .ASCIZ  /SIN ON SEEK/
```

```
5809  023552  020116  042523  045505
5810  023560     000
5811
5812  023561     104  042522  047440  EM3:    .ASCIZ  /DRE ON SEEK/
5813  023566  020116  042523  045505
5814  023574     000
5815
5816  023575     047  051105  023522  EM4:    .ASCIZ  /'ERR' ON SEEK/
5817  023602  047440  020116  042523
5818  023610  045505     000
5819
5820  023613     104  052522  047440  EM5:    .ASCIZ  /DRU ON SEEK, PUT DRV ON 'LOAD' & 'RUN'/
5821  023620  020116  042523  045505
5822  023626  020054  052520  020124
5823  023634  051104  020126  047117
5824  023642  023440  047514  042101
5825  023650  020047  020046  051047
5826  023656  047125  000047
5827
5828  023662  027522  027527  020123  EM6:    .ASCIZ  "R/W/S RDY NOT SET AFTER SEEK"
5829  023670  042122  020131  047516
5830  023676  020124  042523  020124
5831  023704  043101  042524  020122
5832  023712  042523  045505     000
5833
5834  023717     123  047111  047440  EM7:    .ASCIZ  /SIN ON WRT FMT/
5835  023724  020116  051127  020124
5836  023732  046506  000124
5837
5838  023736  042447  051122  020047  EM10:   .ASCIZ  /'ERR' ON DOING WRITE FMT/
5839  023744  047117  042040  044517
5840  023752  043516  053440  044522
5841  023760  042524  043040  052115
5842  023766     000
5843  023767     123  047111  047440  EM11:   .ASCIZ  "SIN ON RD/FMT"
5844  023774  020116  042122  043057
5845  024002  052115     000
5846  024005     047  051105  023522  EM12:   .ASCIZ  /'ERR' ON READ FMT/
5847  024012  047440  020116  042522
5848  024020  042101  043040  052115
5849  024026     000
5850  024027     127  047522  043516  EM13:   .ASCIZ  /WRONG HDRS FROM 'SEC#'/
5851  024034  044040  051104  020123
5852  024042  051106  046517  023440
5853  024050  042523  021503  000047
5854
5855  024056  051105  051117  047440  EM14:   .ASCIZ  /EROR ON IMPLIED SEEK FROM 'CYLA' TO 'CYLB'/
5856  024064  020116  046511  046120
5857  024072  042511  020104  042523
5858  024100  045505  043040  047522
5859  024106  020115  041447  046131
5860  024114  023501  052040  020117
5861  024122  041447  046131  023502
5862  024130     000
5863
5864  024131     122  040505  020104  MS15:   .ASCIZ  /READ WRONG HDRS FROM 'CYLB' ABOVE/
```

```
5865  024136  051127  047117  020107
5866  024144  042110  051522  043040
5867  024152  047522  020115  041447
5868  024160  046131  023502  040440
5869  024166  047502  042526     000
5870
5871  024173     122  040505  020104  EM16:   .ASCIZ  /READ WRONG, 1ST WRD FROM SEC 0, 'CYLB' (ON IMPLIED SEEK FROM 'CYLA')/
5872  024200  051127  047117  026107
5873  024206  030440  052123  053440
5874  024214  042122  043040  047522
5875  024222  020115  042523  020103
5876  024230  026060  023440  054503
5877  024236  041114  020047  047450
5878  024244  020116  046511  046120
5879  024252  042511  020104  042523
5880  024260  045505  043040  047522
5881  024266  020115  041447  046131
5882  024274  023501  000051
5883
5884  024300  042522  042101  030440  MS17:   .ASCIZ  /READ 1ST WRD FROM SEC 1, 'CYLB' ABOVE/
5885  024306  052123  053440  042122
5886  024314  043040  047522  020115
5887  024322  042523  020103  026061
5888  024330  023440  054503  041114
5889  024336  020047  041101  053117
5890  024344  000105
5891
5892  024346  042522  042101  053440  EM20:   .ASCIZ  /READ WRONG HDRS ON IMPLIED SEEK FROM 'CYLA' TO 'CYLB'/
5893  024354  047522  043516  044040
5894  024362  051104  020123  047117
5895  024370  044440  050115  044514
5896  024376  042105  051440  042505
5897  024404  020113  051106  046517
5898  024412  023440  054503  040514
5899  024420  020047  047524  023440
5900  024426  054503  041114  000047
5901
5902  024434  051105  051117  047440  EM21:   .ASCIZ  /EROR ON DOING WRITE ON DISK/
5903  024442  020116  047504  047111
5904  024450  020107  051127  052111
5905  024456  020105  047117  042040
5906  024464  051511  000113
5907
5908  024470  044523  020116  047117  EM22:   .ASCIZ  /SIN ON DOING WRITE/
5909  024476  042040  044517  043516
5910  024504  053440  044522  042524
5911  024512     000
5912
5913  024513     110  020105  047117  EM23:   .ASCIZ  /HE ON DOING READ/
5914  024520  042040  044517  043516
5915  024526  051040  040505  000104
5916
5917  024534  051503  020105  047117  EM24:   .ASCIZ  /CSE ON READ/
5918  024542  051040  040505  000104
5919
5920  024550  040504  040524  042440  EM25:   .ASCIZ  /DATA EROR ON READ FROM DISK ADRES/
```

```
 5921  024556  047522  020122  047117
 5922  024564  051040  040505  020104
 5923  024572  051106  046517  042040
 5924  024600  051511  020113  042101
 5925  024606  042522  000123
 5926
 5927  024612  042510  047440  020116  EM26:   .ASCIZ  /HE ON WRT CHK/
 5928  024620  051127  020124  044103
 5929  024626  000113
 5930
 5931  024630  051127  020124  044103  EM27:   .ASCIZ  /WRT CHK EROR/
 5932  024636  020113  051105  051117
 5933  024644     000
 5934
 5935  024645     105  047522  000122  EM30:   .ASCIZ  /EROR/
 5936
 5937                                  ;DATA HEADERS
 5938
 5939
 5940  024652  020040  041520  020040  DH1:    .ASCIZ  /  PC     RKCS    RKER    RKDS    RKDA/
 5941  024660  020040  051040  041513
 5942  024666  020123  020040  051040
 5943  024674  042513  020122  020040
 5944  024702  051040  042113  020123
 5945  024710  020040  051040  042113
 5946  024716  000101
 5947
 5948  024720  042523  045505  021440  DH6:    .ASCIZ  /SEEK #  SEEKING  ERRORS/
 5949  024726  020040  042523  045505
 5950  024734  047111  020107  042440
 5951  024742  051122  051117  000123
 5952
 5953  024750  020040  041520  020040  DH7:    .ASCIZ  /  PC     RKCS    RKER    RKDS    RKDA    CYL/
 5954  024756  020040  051040  041513
 5955  024764  020123  020040  051040
 5956  024772  042513  020122  020040
 5957  025000  051040  042113  020123
 5958  025006  020040  051040  042113
 5959  025014  020101  020040  041440
 5960  025022  046131     000
 5961
 5962  025025     040  050040  020103  DH11:   .ASCIZI/  PC    RKCS    RKER    RKDS   RKDA: DR  CYL     SUR     SEC/
 5963  025032  020040  051040  041513
 5964  025040  020123  020040  051040
 5965  025046  042513  020122  020040
 5966  025054  051040  042113  020123
 5967  025062  020040  045522  040504
 5968  025070  020072  051104  020040
 5969  025076  054503  020114  020040
 5970  025104  020040  052523  020122
 5971  025112  020040  020040  042523
 5972  025120  000103
 5973  025122  042523  021503  020040  DH13:   .ASCIZ  /SEC#  HDR RCVD/
 5974  025130  042110  020122  041522
 5975  025136  042126     000
 5976
```

```
 5977  025141     040  050040  020103  DH14:   .ASCIZ  /  PC     CYLA    CYLB     RKER   RKDS     TRY#/
 5978  025146  020040  020040  054503
 5979  025154  040514  020040  020040
 5980  025162  054503  041114  020040
 5981  025170  020040  051040  042513
 5982  025176  020122  020040  045522
 5983  025204  051504  020040  020040
 5984  025212  052040  054522  000043
 5985
 5986  025220  020040  041520  020040  DH16:   .ASCIZ  /  PC     CYLA    CYLB     EXPCT   RECVD   TRY#/
 5987  025226  020040  041440  046131
 5988  025234  020101  020040  054503
 5989  025242  041114  020040  020040
 5990  025250  054105  041520  020124
 5991  025256  020040  042522  053103
 5992  025264  020104  020040  051124
 5993  025272  021531     000
 5994
 5995  025275     040  050040  020103  DH17:   .ASCIZ  /  PC     CYLB    EXPCT   RECVD/
 5996  025302  020040  020040  054503
 5997  025310  041114  020040  042440
 5998  025316  050130  052103  020040
 5999  025324  051040  041505  042126
 6000  025332     000
 6001
 6002  025333     040  050040  020103  DH24:   .ASCIZ/  PC     TRY#    RKCS     RKER     RKDS   RKDA: DR  CYL      SUR       SEC/
 6003  025340  020040  020040  051124
 6004  025346  021531  020040  051040
 6005  025354  041513  020123  020040
 6006  025362  051040  042513  020122
 6007  025370  020040  051040  042113
 6008  025376  020123  051040  042113
 6009  025404  035101  042040  020122
 6010  025412  041440  046131  020040
 6011  025420  020040  051440  051125
 6012  025426  020040  020040  020040
 6013  025434  042523  000103
 6014
 6015  025440  047527  042122  020043  DH25:   .ASCIZ  /WORD#  EXPCT   RECVD   CYL  SUR  SEC/
 6016  025446  042440  050130  052103
 6017  025454  020040  051040  041505
 6018  025462  042126  020040  041440
 6019  025470  046131  020040  052523
 6020  025476  020122  051440  041505
 6021  025504     000
 6022
 6023
 6024                                  ;ERROR DATA POINTERS
 6025
 6026
 6027          025506                  .EVEN
 6028
 6029  025506  001116  001162  001164  DT1:    .WORD   $ERRPC,$REG0,$REG1,$REG2,$REG3,0
 6030  025514  001166  001170  000000
 6031
 6032  025522  001116  001162  001164  DT7:    .WORD   $ERRPC,$REG0,$REG1,$REG2,$REG3,$REG4,0
```

```
 6033  025530  001166  001170  001172
 6034  025536  000000
 6035
 6036  025540  001116  001162  001164  DT11:   .WORD   $ERRPC,$REG0,$REG1,$REG2,$REG4,$REG5,$REG6,$REG7,0
 6037  025546  001166  001172  001174
 6038  025554  001176  001200  000000
 6039
 6040  025562  001116  001162  001164  DT17:   .WORD   $ERRPC,$REG0,$REG1,$REG2,0
 6041  025570  001166  000000
 6042
 6043  025574  001116  001202  001162  DT24:   .WORD   $ERRPC,$REG10,$REG0,$REG1,$REG2,$REG4,$REG5,$REG6,$REG7,0
 6044  025602  001164  001166  001172
 6045  025610  001174  001176  001200
 6046  025616  000000
 6047
 6048                                  ;DATA BUFFERS
 6049
 6050  025620                          IOBUF0:
 6051  025620  000030                  OUTBUF: .BLKW   24.      ;IOBUF0 AND IOBUF1 ARE
 6052  025700  000030                  BUFR4:  .BLKW   24.      ;TWO - 768 WORDS LONG BUFFERS
 6053  025760  000030                  BUFR5:  .BLKW   24.      ;NORMALLY USED FOR DATA TRANSFERS
 6054  026040  000030                  BUFR6:  .BLKW   24.      ;TO AND FROM DISK
 6055  026120  000030                  BUFR7:  .BLKW   24.
 6056  026200  000200                  BUFR10: .BLKW   128.
 6057  026600  000200                  BUFR11: .BLKW   128.
 6058  027200  000200                  BUFR12: .BLKW   128.
 6059  027600  000200                  BUFR13: .BLKW   128.
 6060  030200  000210                  BUFR14: .BLKW   136.
 6061
 6062  030620  001400                  IOBUF1: .BLKW   768.
 6063
 6064
 6065
 6066          000001                          .END
```

```
ABRT    016564          2024    2209    2504    2682    3012    3401    3552    4610#
ADRES   001746          1387#   3069*   3078    3083    3261*   3262*   3276*   3278*   3281*   3330    4289
BADTMO  002420          1547#   1665
BEGIN   010314          3072#   3145    3287
BEGSK   013C32          3609#   3627    3633
BIT0  = 000001           998#   2806    3235    3273    3494
BIT00 = 000001           988#    998
BIT01 = 000002           987#    997
BIT02 = 000004           986#    996
BIT03 = 000010           985#    995
BIT04 = 000020           984#    994
BIT05 = 000040           983#    993
BIT06 = 000100           982#    992
BIT07 = 000200           981#    991
BIT08 = 000400           980#    990
BIT09 = 001000           979#    989    4651
BIT1  = 000002           997#   3143
BIT10 = 002000           978#
BIT11 = 004000           977#
BIT12 = 010000           976#
BIT13 = 020000           975#   4598
BIT14 = 040000           974#   3114    3226    3353    4635
BIT15 = 100000           973#   2826    2829
BIT2  = 000004           996#
BIT3  = 000010           995#   4013
BIT4  = 000020           994#
BIT5  = 000040           993#
BIT6  = 000100           992#
BIT7  = 000200           991#   2743    2751
BIT8  = 000400           990#
BIT9  = 001000           989#   2778    3125
BLNKS1  002407          1535#   3863    3982    3987
BLNKS2  002406          1534#   3763    3775    3860    4274    4278    4777    4808    4813
BLNKS3  002405          1533#   3865    4164    4269    4301
BLNKS4  002404          1532#   4168    4309
BLNKS5  002403          1531#   2179
BLNKS6  002402          1530#   3948
BLNKS7  002401          1529#   2186    3849    3872
BLNKS8  002400          1528#
BLNKS9  002377          1527#   3746
BLNK10  002376          1526#
BLNK13  002373          1525#   3772
GPTVEC= 000014          1005#
BRKDA   016012          4293    4406#
BTEOP   015024          4092#   4727
BUFLG0  001706          1354#   2726*   2739    2751*   2789    2791*   2824    2829*
BUFLG1  001710          1355#   2727*   2743*   2794*   2826*
BUFNO   001744          1385#   2827*   2830*   2841    2891    2944    2978    3103*
BUFR    001564          1340#   1823    2158    2663    3179    4157    4249
BUFR1   001616          1341#   2664    3181    4158    4251
BUFR10  026200          3604    3652    3653    3817    3907    3933    6056#
BUFR11  026600          3605    3706    3707    6057#
BUFR12  027200          6058#
BUFR13  027600          6059#
BUFR14  030200          6060#
BUFR2   001650          1342#   3182    4252
```

```
BUFR4   025700          3654    3764    6052#
BUFR5   025760          3655    3759    6053#
BUFR6   026040          3708    3776    6054#
BUFR7   026120          3709    3768    6055#
BUSADR  001732          1375#   2741*   2748*   2755    3085*   3098    3334*
CHKHDR  007042          2108    2500    2597    2661#
CHKSWD  023236          5591    5599    5748#
CHSW    023142          5724#   5734
CKSWR = 104406          4631    5458#
CLRERR  011744          3377    3394    3398#
CMPAGA  011026          3191#   3197    3366
CMPGRP  014404          3886    3931#
CN.RDY  016412          4558    4559#   5473
CN.RST  016374          4556#   5465
COMPAR  011002          3187#
CON.RD= 104420          1981    2074    2143    2290    2452    2498    2562    2769    3109    3212    3224    3351    3441
                        3486    3600    3830    4028    4506    5473#   5772    5774
CON.RE= 104414          1831    1960    1967    1990    2061    2082    2097    2146    2249    2278    2327    2525    2551
                        2581    2774    2782    3090    3094    3120    3131    3218    3338    3355    3398    3435    3447
                        3574    3814    5465#   5704    5733    5763
CR    = 000015           913#   4951    4961
CRLF  = 000200           914#   1612    4922    4961
CSEROR  011436          3153    3292#
DDISP = 177570           920#   1059    1580
DH1     024652          1114    1121    1128    1135    1142    1149    1229    1236    1286    5940#
DH11    025025          1170    1243    1268    5962#
DH13    025122          1186    1200    1221    5973#
DH14    025141          1193    5977#
DH16    025220          1207    5986#
DH17    025275          1214    5995#
DH24    025333          1252    1277    6002#
DH25    025440          1261    6015#
DH6     024720          5948#
DH7     024750          1156    1163    1179    5953#
DISPLA  001142          1059#   1580*   1588*   4661*   4681*
DISPRE  000174          1021#   1588
DMPREG  017440          4793    4801#
DONE    011420          3279    3283#
DONTRK  011360          3271#
DOSUR1  011412          3274    3281#
DOWCHK  012202          3469    3476#
DOWRT   012006          3431#   3533
DRHOLD  002416          1540#   1949*   2021    5764*   5773    5778
DRIVAD  001526          1315#   1621    1725*   1750*   1835    1949    1959    2019*   2021*   2059    2269    2271    2542
                        2546    2731    3069    3433    3471    3479    3526    3598    3614    3618    3821    4504    4714
                        5764    5765    5767*   5769*   5770    5778*   5781*   5793*
DRIVS   001523          1307#   1675*   1682*   1691    4092    4725*
DRIV0   001530          1318#   1677    1700    1723    5782
DRIV1   001532          1319#
DRIV2   001534          1320#
DRIV3   001536          1321#
DRIV4   001540          1322#
DRIV5   001542          1323#
DRIV6   001544          1324#
DRIV7   001546          1325#   1717    1731    5794
DRVDON  001522          1305#   1724*   4091*   4092
```

DZRKLC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
DRVPTR  001524          1312#  1723*  1728   1739*  4710
DRV.RE= 104415          1833   1961   1991   2062   2083   2250   2279   2526   2552   2582   2783   3091   3132
                        3575   3815   5467#  5762   5779
DR.RST  016262          4503#  5467
DSKADR  001730          1373#  2731*  2754   2800*  2801*  2811*  2812*  2815*  3083*  3096   3302   3328*
DSWR  = 177570           919#  1058   1579
DT1     025506          1115   1122   1129   1136   1143   1150   1230   1237   1287   6029#
DT11    025540          1172   1245   1270   6036#
DT17    025562          1215   6040#
DT24    025574          1254   1279   6043#
DT7     025522          1157   1164   1180   1194   1208   6032#
EMTVEC= 000030          1008#  1591*  1592*
EM1     023506          1113   5801#
EM10    023736          1162   5838#
EM11    023767          1169   5843#
EM12    024005          1178   5846#
EM13    024027          1185   5850#
EM14    024056          1192   5855#
EM16    024173          1206   5871#
EM2     023545          1120   5808#
EM20    024346          1220   5892#
EM21    024434          1228   5902#
EM22    024470          1235   5908#
EM23    024513          1242   5913#
EM24    024534          1251   5917#
EM25    024550          1260   5920#
EM26    024612          1267   5927#
EM27    024630          1276   5931#
EM3     023561          1127   5812#
EM30    024645          1285   5935#
EM4     023575          1134   5816#
EM5     023613          1141   5820#
EM6     023662          1148   5828#
EM7     023717          1155   5834#
ERCNT1  002002          1424#  2055*  2121*  2258*  2311*  2533*  2585*  2723*  2775*  3072*  3138   3140*  3427*
                        3448*
ERCNT2  002004          1425#  2057*  2099*  2259*  2380*  2534*  2608*  2724*  2784*  3073*  3155*  3168   3171*
                        3428*  3492*
ERCNT3  002006          1426#  2058*  2088*  2260*  2460*  3059*  3121*  3429*  3503*
ERCNT4  002010          1427#  3060*  3133*
ERCNT5  002012          1428#  3061*  3150*
ERCNT6  002014          1429#  3062*  3204*
ERCNT7  002016          1430#  3063*  3231*
ERCNT8  002020          1431#  3064*  3241*
ERRTYP  017224          4350   4697   4729#
ERRVEC= 000004          1001#  1577   1578*  1589*  4640   4641*  4643*  4646*
ERR1    016170          2295   2317   2415   4465#
ERR2    016112          2570   4225   4444#
ERWCHK  011166          3216   3225#
ESR13   015250          1188   4198#  4224
ESR15   015156          1202   4154#  4198
ESR20   015324          1223   4224#
ESR25   015412          1263   4247#
EXEC    023050          5629   5658   5691   5702   5705#  5725
EXEC1   023046          5704#  5729   5730
EXT10   012514          3450   3493   3504   3552#
```

DZRKLC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
EXT4    006376          2313    2383    2462    2504#
EXT5    007134          2586    2610    2682#
EXT6    010214          2777    2786    3012#
EXT7    011752          3123    3135    3152    3206    3233    3243    3401#
FCHECK  023310          5762#   5787
FDRIVE  002412          1538#   1742*   1748*   1950    2017*
FDRVE1  002414          1539#   1741*   1745*   1762    1952
FFUNC   001517          1027*   1298#   1615    1617*
FINISH  011274          3124    3136    3236    3253#   3386
FTITLE  001516          1297#
FUNBEG  022344          1618    5579#   5598    5613    5620    5627    5632    5634    5726
FUNERR  023156          5723    5729#
GCYL    016226          2454    4445    4448    4466    4480#
GETBUF  007626          2734    2765    2824#
GETINF  016006          2078    3118    3129    3146    3229    3237    3490    3497    4405#
GNS   = ****** U        1020    1552    1611    1647    1651    1661    1673    1695    1713    1766    1778    3582    3586
                        3590    3719    3732    3744    3749    3841    3845    3878    3903    3952    3992    4201    4229
                        4237    4568    4576    4723    4804    5450    5451    5452    5453    5454    5456    5458    5459
                        5460    5461    5462    5463    5465    5467    5469    5471    5473    5475    5477    5479    5582
                        5640    5649    5664    5672    5697    5753
GOBAK   011432          3268    3284    3287#
GOTYPE  013362          3704    3715#
GTSWR = 104405          1606    5456#
GT4RG   015754          1839    1850    1866    1874    1883    1892    1902    2772    2780    3445    4393#   4405    5731
GT5RG   015730          1984    2001    2094    4378#
HT    = 000011           911#   4920    4961
INADR   001556          1334#   2262*   2268    2400*   2480    2537*   2541    2623*   2639    3190*   3196*   3593*   3613
                        3734    3792*   3818*   3820    3832*   4444
INDX1   001772          1419#   2725*   2806    2808*   2814*   3065*   3273    3283*   3603*   3610    3623    3626*   3630*
INDX2   001774          1420#   2060*   2205*   2256*   2397*   2531*   2624*   3066*   3259    3265*   3267    3271*   3606*
                        3632*   5667*   5712    5757*
INDX3   001776          1421#   2110    2339    2434    2600    2667*   2675*   3185*   3199    3361*   3816*   3833*
INDX4   002000          1422#   3188*   3363*   5701*   5714
INPT    023204          5642    5651    5736#
IOBUF0  025620          1349    6050#
IOBUF1  030620          1351    5756*   5757    6062#
IOTVEC= 000020          1006#   1567*   1568*
LF    = 000012           912#   4955    4961
LUPHE   010422          3094#   3117
LUPSIN  010414          3090#   3128
LUPSW   001521          1300#   1772*   1790*
LUPWCE  011132          3217    3218#   3225
MESAGE= 104416          2355    2374    2441    2457    5469#
MISCMP  011600          3193    3351#
MSGE    015636          4345#   5469
MSG1    002100          1475#   2298    2572
MSG10   002240          1501#   4565
MSG11   002267          1506#   2157
MSG12   002325          1513#   4520    4600
MSG13   002351          1518#   4209    4321
MSG14   002362          1522#   1751    4713
MSG2    002106          1477#   2322
MSG3    002114          1479#   4610
MSG4    002136          1483#   1931
MSG5    002155          1487#   2345    2368    2436
MSG6    002214          1494#   4175
```

```
MSG7    002231          1498*   2195    4521    4601    4741    4787
MS15    024131          1199    5864*
MS17    024300          1213    5884*
NOHE    010542          3115    3125*
NOSIN   010610          3126    3138*
NXTBA   022674          5596    5604    5611    5661*
NXTDA   022720          5593    5601    5615    5622    5669*   5684    5688
NXTDRV  003524          1726*   4094
NXTPAT  001724          1369*   2759*   2762*   2764*   2766    2796
NXTWC   023014          5694*
OUTADR  001560          1335*   2536*   2545    2617*   2641    3594*   3596    3617    3722    3793*   4447
OUTBUF  025620          1972*   1973    2067    2139    2164    2284    2411    2419    2449    2456    2493    2557    2662
                        6051*
PAT0    001712          1358*   2732    2762    3068    3257
PAT1    001714          1360*
PAT2    001716          1362*
PAT3    001720          1364*   2760    3253
PBUF0   001702          1349*   2748    2830    3085    3187    4259
PBUF1   001704          1351*   2741    2827    3079    3103    3189
PC    =%000007           932*   1690*   1839*   1850*   1866*   1874*   1883*   1892*   1902*   1984*   2001*   2024*   2078*
                        2094*   2108*   2209*   2280*   2295*   2317*   2329*   2361*   2393*   2415*   2430*   2454*   2483*
                        2500*   2502*   2504*   2553*   2570*   2597*   2644*   2680*   2682*   2734*   2736*   2765*   2767*
                        2772*   2780*   2828*   2831*   2862*   2917*   2957*   3003*   3012*   3105*   3118*   3129*   3146*
                        3229*   3237*   3401*   3445*   3490*   3497*   3552*   3619*   3823*   3908*   3956*   3994*   4061*
                        4069*   4080*   4110*   4113*   4123*   4128    4182*   4198*   4213*   4224*   4225*   4243*   4293*
                        4328*   4350*   4397*   4405*   4431*   4445*   4448*   4454*   4466*   4471*   4473*   4491*   4613*
                        4697*   4771*   4779*   4793*   4817*   4929*   4936*   4943*   4957*   4959*   5009*   5065*   5279*
                        5281*   5313*   5396*   5401*   5518    5591*   5599*   5642*   5651*   5731*   5746*   5758*   5760*
                        5796*
PGSUBR  001726          1371*   2735*   2736    2766*   2767    3104*   3105
PIRQ  = 177772           918*
PIRQVE= 000240          1012*
PLOT    014004          3838*
PLTGRP  013670          3573    3795    3797    3811*
PLTPT   014520          3908    3956    3978*   3983
PLT1    014220          3874*
PRSPAT  001722          1366*   2732*   2735    2759    2760    2790*   3068*   3104    3253    3255*   3257*
PR0   = 000000           935*
PR1   = 000040           936*
PR2   = 000100           937*
PR3   = 000140           938*
PR4   = 000200           939*
PR5   = 000240           940*
PR6   = 000300           941*
PR7   = 000340           942*
PS    = 177776           915*    916
PSW   = 177776           916*
PTGEN0  007670          1358    2841*
PTGEN1  007752          1360    2889*
PTGEN2  010054          1362    2942*
PTGEN3  010116          1364    2978*
PTRNG1  007746          2842    2864*
PTRNG2  007750          2855    2865*   2980
PT1     002056          1463*   1788
PT10    002074          1470*
PT11    002076          1471*
```

```
PT2     002060          1464#
PT3     002062          1465#
PT4     002064          1466#
PT5     002066          1467#
PT6     002070          1468#
PT7     002072          1469#
PWRFL   004026          1800#   5519
PWRVEC= 000024          1007#   1571*   1572*   5486*   5487*   5496*   5502*   5514*   5515*
RDAGAI  010424          3088    3092    3096#   3339
RDCHR = 104407          1654    5124    5459#
RDLIN = 104410          5166    5460#   5584
RDCCT = 104411          1780    5461#   5666    5674    5699    5736    5755
READ    010370          3083#   3202    3210
REPEAT  012450          3524    3530#
REPMSC  011056          3202#   3368
REPTIM  012762          3596#   3798
RESDON= 104422          2359    2377    5477#   5720
RESREG= 104413          5400    5463#
RESVEC= 000010          1002#
RES.DO  016306          4507    4508#   5477
RETRY1  001550          1329#   2065*   2084*   2085    2275*   2320    2343*   2351*   2549*   2568    2577    2579*   3074*
                        3147    3306*   3322*   3325*   3477*   3498    3501*   3505*   3577*   3794*
RETRY2  001552          1330#   2064*   2117*   2118    2274*   2413    2425*   2548*   2605*   2606    3075*   3207    3209*
                        3214    4210    4322
RETRY3  001554          1331#   2276*   2299    2306    2308*   3076*   3157    3166*   3239    3376    3380*
RKBA    001760          1399#   1973*   2067*   2139*   2284*   2449*   2495*   2557*   2755*   3098*   3162*   3221*   3439*
                        3484*   5712*
RKCS    001754          1397#   1846*   1858    1890    1964    1979*   1999    2072*   2092    2142*   2144    2288*   2334*
                        2451*   2481*   2497*   2560*   2593*   2642*   2757*   2770    3100*   3114    3164*   3222*   3353
                        3440*   3442    3485*   3599*   3829*   4026*   4393    4505*   4556*   4560    4578    5715*   5716
                        5722    5771*
RKDA    001762          1400#   1679*   1835*   1836*   1974*   2068*   2138*   2283*   2448*   2480*   2496*   2556*   2592*
                        2639*   2641*   2754*   3096*   3160*   3219*   3295    3382    3437*   3481*   3596*   3598*   3609
                        3820*   3821*   3828*   4378    4396    4504*   5713*   5770*   5773*
RKDB    001764          1401#   4814
RKDPCH  001520          1299#   1619    1663*   1737
RKDS    001750          1395#   1680    1837    1862    1872    1900    1982    2076    2292    2296    2319    2564    2567
                        2778    3125    3226    3488    4008    4395    4509    4594    4810    5775
RKER    001752          1396#   1880    2297    2315    2318    2566    3143    3235    3494    4394
RKPRI   001766          1403#
RKVEC   001770          1411#
RKWC    001756          1398#   1975*   2069*   2140*   2282*   2450*   2494*   2555*   2756*   3097*   3161*   3220*   3439*
                        3483*   5714*
R0      =%000000         923#   1547*   1548    1554    1619*   1620*   1621    1624*   1700*   1701    1709    1716    1717
                        1781*   1783    1785    1786*   1787*   1788    1800*   1804*   1818*   1947*   1861*   1864*   1962*
                        1992*   1993    2257*   2258    2259    2260    2446*   2447*   2448    2455    2492*   2494    2532*
                        2533    2534    2661*   2672    2677*   2678    2798*   2799*   2801    2841*   2845*   2851*   2857*
                        2891*   2894*   2903*   2912*   2944*   2945*   2950*   2954*   2978*   2982*   2990*   2999*   3187*
                        3191    3194    3295*   3296    3297    3299*   3300*   3302    3321*   3328    3330*   3331*   3332*
                        3357    3360    3382*   3383    3389    3390*   3392*   3393*   3432*   3433*   3437    3452*   3462*
                        3463*   3468    3470*   3471*   3478*   3479*   3481    3507*   3517*   3518*   3523    3525*   3526*
                        3657*   3662    3664    3665*   3668    3675*   3686    3688    3692    3722*   3723*   3724*   3725*
                        3726*   3727*   3728    3755*   3759    3764    3768    3776    3788    3789    3850*   3851    3853
                        3855    3857    3866*   3874*   3880*   3891*   3899    3905    3909    3911*   3913    3915*   3917
                        3919*   3921    3923*   3931*   3944    3961*   3962    4120*   4123    4280*   4281    4293*   4294
                        4287*   4288*   4289*   4291    4406    4409*   4413*   4420*   4424*   4427*   4430*   4480    4481*
```

F10


```
                    4482*  4484*  4485*  4486*  4487*  4488*  4489   4490*  4731   4732*  4733*  4745*  4746*
                    4747*  4748*  4749*  4750   4757   4762*  4763   4765*  4767*  4769*  4773   4775   4779
                    4809   4810*  4811   4814   4816*  4836   4846*  4850   4866   4867   4880*  4912   4913*
                    4914   4917*  4978   4991*  4993*  4994*  5000*  5002*  5004   5008*  5163   5167*  5170
                    5186*  5301   5302*  5303   5305   5307*  5308   5311*  5333   5358*  5376*  5377   5395*
                    5399*  5434   5435*  5436   5437*  5438*  5439*  5440*  5488   5513*  5585*  5586   5589
                    5597   5608   5612   5619   5626   5633   5644   5653   5654*  5655*  5656   5657   5675*
                    5676   5685   5689   5737*  5738   5740*  5741*  5742*  5743*  5744*  5782*  5783   5785
                    5789*  5790*  5791   5792   5794
R1      =%000001     924#  1623*  1625*  1676*  1679   1686*  1728*  1729   1731   1735   1739   1740   1801*
                    1802*  1819*  1924*  1925   1957*  1996*  2137*  2149*  2158*  2163   2173   2182   2191
                    2192   2493*  2495   2662*  2669   2671   2676   2842*  2845   2850*  2851   2855*  2857
                    2893*  2894   2896*  2902*  2903   2905*  2911*  2912   2913*  2942*  2945   2946*  2949*
                    2950   2951*  2953*  2954   2955*  2980*  2981*  2982   2989*  2990   2991*  2997*  2998
                    2999   3000*  3001   3189*  3191   3195   3259*  3260*  3262   3296*  3314   3321   3359
                    3389*  3391   3431*  3439   3454   3456*  3457   3461*  3476*  3484   3509   3511*  3512
                    3516*  3658*  3662   3665   3666*  3669   3676*  3686*  3688   3691   3692*  3734*  3735*
                    3736*  3737*  3738*  3739*  3740   3905*  3906*  3907   3932*  3941*  3943*  4008*  4009
                    4017   4020   4030   4033   4040   4043   4155   4157*  4159   4171   4181*  4407   4410*
                    4411   4414*  4415*  4416*  4417*  4418   4421*  4422   4425*  4426*  4427   4429*  4837
                    4850*  4851   4855   4879*  4979   4982*  4985*  4995*  5001*  5005   5007*  5164   5168*
                    5172*  5174*  5176*  5179*  5182   5185*  5334   5357*  5378*  5384*  5390*  5489   5512*
                    5586*  5587   5606   5617   5624   5631   5676*  5677*  5678*  5679*  5680*  5681*  5682*
                    5683   5685*  5686*  5687
R2      =%000002     925#  1677*  1683*  1684   1699*  1705   1715*  1820*  1836   1916   1919*  1922*  1955*
                    2015*  2163*  2167   2169*  2663*  2672*  2674*  2843*  2846*  2849*  2852*  2854*  2859*
                    2892*  2898*  2901*  2907*  2910*  2915*  2979*  2983*  2987*  2993*  2998*  3000   3179*
                    3314*  3315*  3318*  3357*  3383*  3384*  3387*  3430*  3431   3461   3476   3516   3530
                    3531   3609*  3613*  3614*  3617*  3618*  3659*  3670*  3677*  3678   3693   3694*  3699*
                    3933*  3955   4006   4009*  4010   4013   4017*  4018   4020   4022   4030*  4031   4033
```