

RK11/RK05

DYNAMIC TEST

MD-11-DZRKL-D

EP-DZRKL-D-DL-B

COPYRIGHT © 1976

FICHE 1 OF 1

DEC 1976

digital

MADE IN U.S.A.

B01

IDENTIFICATION

SEQ 0001

PRODUCT CODE:	MAINDEC-11-DZRKL-D-D
PRODUCT NAME:	RK11/RK05 DYNAMIC TEST
DATE CREATED:	DECEMBER, 1976
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	JIM KAPADIA
REVISED BY:	PERVEZ ZAKI TOM SAWYER CHUCK HESS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1975, 1976 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	EQUIPMENT
2.2	PRELIMINARY PROGRAMS
2.3	EXECUTION TIME
3.0	STARTING ADDRESSES
4.0	PROGRAM CONTROL MODES AND OPERATOR ACTION
4.1	PAPER TAPE
4.2	RKDP DUMP MODE
4.3	RKDP CHAIN MODE
4.4	ACT11
5.0	DRIVE SELECTION
6.0	SWITCH OPTIONS
7.0	PROGRAM DESCRIPTION
7.1	PERMISSIBLE USER PROGRAM MODIFICATIONS
8.0	SEEK TIMER AND GRAPHS
9.0	FUNCTION SELECTION PROGRAM
10.0	ERROR INFORMATION
11.0	UNEXPECTED TIMEOUTS
12.0	COMMONLY USED SUBROUTINES
13.0	SAMPLE GRAPH AND TIMER OUTPUTS

1.0 ABSTRACT

THE RK11/RK05 DYNAMIC TEST AIMS AT

1. DEMONSTRATING THE ELECTROMECHANICAL INTEGRITY OF THE DRIVE.
2. CHECKING THE LINEAR POSITIONER CONTROL AND SPEED CONTROL
3. VERIFYING THE INTEGRITY 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 OR RKV11 CONTROLLER
- D. 1-8 RK05 OR RK05F DRIVES (DRIVE TYPES MAY BE MIXED)

2.2 PRELIMINARY PROGRAMS

RK11 LOGIC TEST I (MAINDEC-11-DZRKJ)
RK11 LOGIC TEST 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 SWITCHES DOWN

210 FOR FUNCTION SELECTING PROGRAM (CONVERSATIONAL MODE).

4.0 PROGRAM CONTROL MODES & OPERATOR ACTION

PAPER TAPE LOADING
RKDP DUMP MODE
RKDP CHAIN MODE
ACT11

4.1 PAPER TAPE LOADING

- 4.1.1 LOAD PROGRAM INTO MEMORY USING STANDARD PROCEDURE FOR 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

RK11 DYNAMIC TEST
MAINDEC-11-DZRKL-D

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:

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

4.3 RKDP CHAIN MODE

THE PROGRAM IS CHAIN LOADED FROM RKDP PACK ON DRIVE 'N'. AFTER IDENTIFYING ITSELF, THE FOLLOWING MESSAGE APPEARS:

'DRIVE 'N' NOT TESTED'

DRIVE 'N' WILL NOT BE TESTED SINCE THE RKDP PACK 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 LOOPING 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 APPEARS:

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 OCCURRING.

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.

SEQ 0008

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-DSF, 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', 'PTRND1' 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 PRECEDED 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 'STPS' TO 177514 AND LOCATION 'STPB' 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. 'PATO' 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 'PTRND1' AND 'PTRND2' WITH THE PATTERN YOU WANT.
TO WRITE 2 DIFFERENT PATTERNS (IN ALTERNATING SECTORS):
CHANGE 'PAT1', 'PAT2' AND 'PAT3' AS DESCRIBED ABOVE.
FILL 'PTRND1' AND 'PTRND2' 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.

B.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 ADDRESS (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 PERTAINING TO THE ERROR ARE ALSO GIVEN. RKDS, RKER...RKBA INDICATE THE CONTENTS OF THE CORRESPONDING REGISTERS AT THE TIME OF ERROR.

EVERY ERROR MESSAGE CONTAINS A PC. THIS PC INDICATES THE POSITION IN PROGRAM WHERE THE ERROR CALL IS LOCATED. THE ERROR MESSAGE, BECAUSE OF PRACTICAL CONSIDERATIONS IS MADE SHORT AND MEANINGFUL. THE USER IS 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 EXPLANATION 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 (STRPAD) 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

B02

SEQ 0014

CYLS:0-16
FRWRD

REVRSE

100 3135 100 3135

CYLS:0-32
FRWRD

REVRSE

100 3795 100 3795

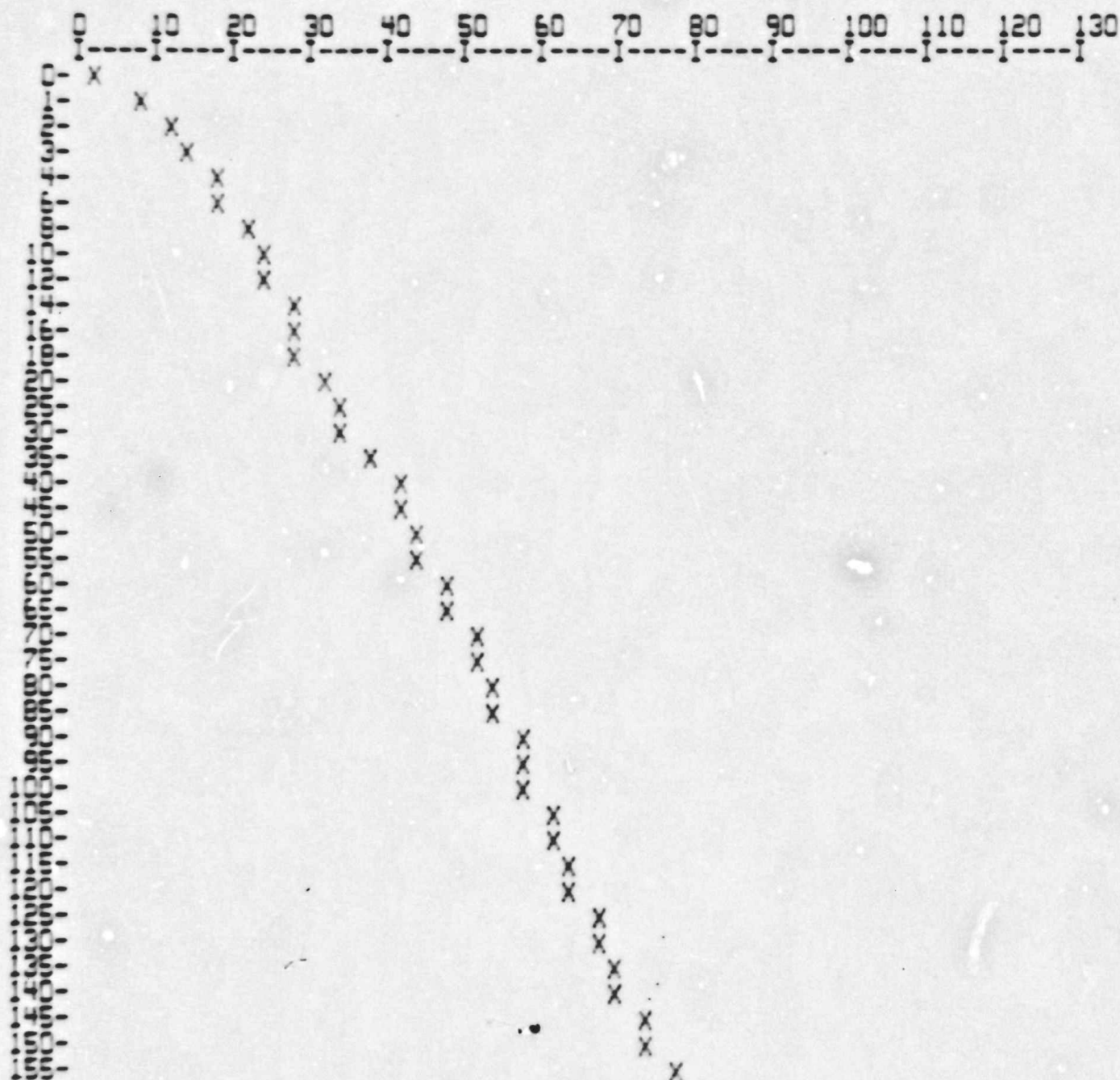
CYLS:0-100
FRWRD

REVRSE

100 5775 100 5775

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

SAMPLE OUTPUT



MAINDEC-11-DZAKL-D MACY11 27(1006) 04-OCT-76 14:26
 DZAKLD.P11 31-AUG-76 15:35 TABLE OF CONTENTS

SEQ 0016

```

1  OPERATIONAL SWITCH SETTINGS
2  BASIC DEFINITIONS
3  TRAP CATCHER
4  STARTING ADDRESS(ES)
5  ACT11 HOOKS
6  COMMON TAGS
7  ERROR POINTER TABLE
8  INITIALIZE THE COMMON TAGS
9  TYPE PROGRAM NAME
10 GET VALUE FOR SOFTWARE SWITCH REGISTER
11 T1 CHECK INNER LIMIT SWITCH & ELECTROMECHANICAL INTEGRITY
12 T2 FORMAT THE DISK
13 T3 READ FORMAT OF THE DISK
14 T4 SEEK PATTERNS: 0-312-0-311-... USING IMPLIED SEEK
15 T5 PERFORM CONVERGING-DIVERGING (IMPLIED) SEEKS
16 T6 WRITE PATTERNS ON THE DISK
17 T7 READ SOFTWARE COMPARE, WRITE CHECK OF THE PATTERNS
18 T10 WRITE, WRITE CHECK ON CYLINDERS 127, 128
19 T11 SEEK FUNCTION TIMER
20 T12 END OF PROGRAM
21 END OF PASS ROUTINE
22 ESCR15
23 ESCR13
24 ESCR20
25 ESCR25
26 ERR2
27 ERR1
28 GCYL
29 DRV.RESET - DRIVE RESET ROUTINE
30 RESDON - WAIT FOR DRIVE RESET TO BE DONE
31 CON.RESET - CONTROL RESET ROUTINE
32 CON.RDY - WAIT FOR CONTROL READY
33 TST.RWS - WAIT FOR R/W/S RDY
34 TEST ABORT ROUTINE
35 SCOPE HANDLER ROUTINE
36 ERROR HANDLER ROUTINE
37 CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
38 TYPE ROUTINE
39 INTEGER MULTIPLY ROUTINE
40 TTY INPUT ROUTINE
41 READ AN OCTAL NUMBER FROM THE TTY
42 BINARY TO OCTAL (ASCII) AND TYPE
43 TYPDSS - TYPE DECIMAL, LEADING ZEROES SUPPRESSED
44 TYPE NUMERICAL ASCII STRING SUPPRESS LEADING ZEROS
45 SAVE AND RESTORE RD-RS ROUTINES
46 DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
47 TRAP DECODER
48 TRAP TABLE
49 POWER DOWN AND UP ROUTINES
50 FUNCTION SELECTION PROGRAM
  
```


E02

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 1
DZRKL.D.F11 31-AUG-76 15:35

SEQ 0017

```

: TITLE MAINDEC-11-DZRKL-D
: *COPYRIGHT (C) 1974, 1976
: *DIGITAL EQUIPMENT CORP.
: *MAYNARD, MASS. 01754
: *
: *PROGRAM BY JIM KAPADIA
: *
: *THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
: *PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
: *
: *JANUARY 1975
: *
: *REVISED MARCH 1976 BY TOM SAWYER
: *REVISED BY CHUCK HESS, AUGUST, 1976

```

.SBTTL OPERATIONAL SWITCH SETTINGS

SWITCH	USE
15	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUTS
12	CYCLE ON ERROR TO PREVIOUS 'SCOPE'
10	BELL ON ERROR
9	LOOP ON ERROR
8	SELECT TEST TYPED IN BY USER
6	EXECUTE THE SEEK TIMER (TEST 11)
5	TYPE THE SEEK TIMER GRAPHS (TEST 11)
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 RK11 REGISTERS ON ERROR

```

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

```

F02

MAINDEC-11-DZAKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 2
 DZAKLD.P11 31-AUG-76 15:35 BASIC DEFINITIONS

SEQ 0018

.SBTTL BASIC DEFINITIONS

:*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***

001100

STACK= 1100

.EQUIV EMT,ERROR ::BASIC DEFINITION OF ERROR CALL

.EQUIV IOT,SCOPE ::BASIC DEFINITION OF SCOPE CALL

:*MISCELLANEOUS DEFINITIONS

000011

HT= 11

::CODE FOR HORIZONTAL TAB

000012

LF= 12

::CODE FOR LINE FEED

000015

CR= 15

::CODE FOR CARRIAGE RETURN

000200

CRLF= 200

::CODE FOR CARRIAGE RETURN-LINE FEED

177776

PS= 177776

::PROCESSOR STATUS WORD

.EQUIV PS,PSW

177774

STKLMT= 177774

::STACK LIMIT REGISTER

177772

PIRQ= 177772

::PROGRAM INTERRUPT REQUEST REGISTER

177570

DSWR= 177570

::HARDWARE SWITCH REGISTER

177570

DDISP= 177570

::HARDWARE DISPLAY REGISTER

:*GENERAL PURPOSE REGISTER DEFINITIONS

000000

R0= %0

::GENERAL REGISTER

000001

R1= %1

::GENERAL REGISTER

000002

R2= %2

::GENERAL REGISTER

000003

R3= %3

::GENERAL REGISTER

000004

R4= %4

::GENERAL REGISTER

000005

R5= %5

::GENERAL REGISTER

000006

R6= %6

::GENERAL REGISTER

000007

R7= %7

::GENERAL REGISTER

000006

SP= %6

::STACK POINTER

000007

PC= %7

::PROGRAM COUNTER

:*PRIORITY LEVEL DEFINITIONS

000000

PR0= 0

::PRIORITY LEVEL 0

000040

PR1= 40

::PRIORITY LEVEL 1

000100

PR2= 100

::PRIORITY LEVEL 2

000140

PR3= 140

::PRIORITY LEVEL 3

000200

PR4= 200

::PRIORITY LEVEL 4

000240

PR5= 240

::PRIORITY LEVEL 5

000300

PR6= 300

::PRIORITY LEVEL 6

000340

PR7= 340

::PRIORITY LEVEL 7

:*"SWITCH REGISTER" SWITCH DEFINITIONS

100000

SW15= 100000

040000

SW14= 40000

020000

SW13= 20000

010000

SW12= 10000

004000

SW11= 4000

002000

SW10= 2000

001000

SW09= 1000

000400

SW08= 400

000200

SW07= 200

000100

SW06= 100

000040

SW05= 40

000020

SW04= 20

000010

SW03= 10

000004

SW02= 4

G02

SEQ 0019

```

100      000002      SW01= 2
101      000001      SW00= 1
102      .EQUIV      SW09,SW9
103      .EQUIV      SW08,SW8
104      .EQUIV      SW07,SW7
105      .EQUIV      SW06,SW6
106      .EQUIV      SW05,SW5
107      .EQUIV      SW04,SW4
108      .EQUIV      SW03,SW3
109      .EQUIV      SW02,SW2
110      .EQUIV      SW01,SW1
111      .EQUIV      SW00,SW0

```

```

112      100000      : *DATA BIT DEFINITIONS (BIT00 TO BIT15)
113      040000      BIT15= 100000
114      020000      BIT14= 40000
115      010000      BIT13= 20000
116      004000      BIT12= 10000
117      002000      BIT11= 4000
118      001000      BIT10= 2000
119      000400      BIT09= 1000
120      000200      BIT08= 400
121      000100      BIT07= 200
122      000040      BIT06= 100
123      000020      BIT05= 40
124      000010      BIT04= 20
125      000004      BIT03= 10
126      000002      BIT02= 4
127      000001      BIT01= 2
128      .EQUIV      BIT00,BIT0
129      .EQUIV      BIT09,BIT9
130      .EQUIV      BIT08,BIT8
131      .EQUIV      BIT07,BIT7
132      .EQUIV      BIT06,BIT6
133      .EQUIV      BIT05,BIT5
134      .EQUIV      BIT04,BIT4
135      .EQUIV      BIT03,BIT3
136      .EQUIV      BIT02,BIT2
137      .EQUIV      BIT01,BIT1
138      .EQUIV      BIT00,BIT0

```

```

139      000004      : *BASIC "CPU" TRAP VECTOR ADDRESSES
140      000010      ERRVEC= 4      :: TIME OUT AND OTHER ERRORS
141      000014      RESVEC= 10     :: RESERVED AND ILLEGAL INSTRUCTIONS
142      000014      TBITVEC= 14    :: "T" BIT
143      000014      TRTVEC= 14     :: TRACE TRAP
144      000014      BPTVEC= 14     :: BREAKPOINT TRAP (BPT)
145      000020      IOTVEC= 20     :: INPUT/OUTPUT TRAP (IOT) **SCOPE**
146      000024      PWRVEC= 24     :: POWER FAIL
147      000030      EMTVEC= 30     :: EMULATOR TRAP (EMT) **ERROR**
148      000034      TRAPVEC= 34    :: "TRAP" TRAP
149      000060      TKVEC= 60      :: TTY KEYBOARD VECTOR
150      000064      TPVEC= 64      :: TTY PRINTER VECTOR
151      000240      PIRQVEC= 240   :: PROGRAM INTERRUPT REQUEST VECTOR

```

.SBTTL TRAP CATCHER

H02

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 4
 DZRKLD.F11 31-AUG-76 15:35 TRAP CATCHER

SEQ 0020

```

156
157      000000      . = 0
158      ; *ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
159      ; *SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
160      ; *LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
161      000174      000174      . = 174
162      000174      000000      DISPREG: .WORD 0      ;: SOFTWARE DISPLAY REGISTER
163      000176      000000      SWREG: .WORD 0      ;: SOFTWARE SWITCH REGISTER
164      .SBTTL      STARTING ADDRESS(ES)
165      000200      000137      002462      JMP      @#START ;: JUMP TO STARTING ADDRESS OF PROGRAM
166
167      000210      000210      . = 210
168      000210      105237      001216      INCB      FFUNC      ;: SET FLAG INDICATING SELECTION OF
169      000214      000137      002462      JMP      @#START      ;: FUNCTION PROGRAM.
170      .SBTTL      ACT11 HOOKS
171
172      ;: *****
173      ;: HOOKS REQUIRED BY ACT11
174      000220      000046      $SVPC=.      ;: SAVE PC
175      000046      015254      $ENDAD      ;: 1) SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
176      000052      000052      . = 52      ;: 2) SET LOC.52 TO ZERO
177      000052      000000      .WORD 0      ;: RESTORE PC
178      000220
179
180

```


.SBTTL COMMON TAGS

```

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

```

. = 1100

```

$CMTAG:      .WORD      0
$SPASS:      .BYTE      0
$STSTNM:     .BYTE      0
$SERFLG:     .BYTE      0
$SICNT:      .WORD      0
$SLPADR:     .WORD      0
$SLPERR:     .WORD      0
$SERTTL:     .WORD      0
$SITEMB:     .BYTE      0
$SERMAX:     .BYTE      1
$SERRPC:     .WORD      0
$SGDADR:     .WORD      0
$SBDADR:     .WORD      0
$SGDDAT:     .WORD      0
$SBDAT:      .WORD      0

```

```
SAUTOB: .BYTE 00
SINTAG: .BYTE 00
```

```
SWR:      .WORD      DSWR
DISPLAY:   .WORD      DDISP
```

```

$TKS: 177560
$TKB: 177562
$TPS: 177564
$TPB: 177566
$NULL: .BYTE 0
$FILLS: .BYTE 2
$FILLC: .BYTE 12
$TPFLG: .BYTE 0
$REGAD: .WORD 0

```

```
$REG0: .WORD 0
$REG1: .WORD 0
$REG2: .WORD 0
$REG3: .WORD 0
$REG4: .WORD 0
$REG5: .WORD 0
$REG6: .WORD 0
$REG7: .WORD 0
$REG10: .WORD 0
```

```
$ESCAPE:0
$BELL: .ASCIZ <207><377><377>
$QUES: .ASCIZ /?/
$CRLF: .ASCIZ <15>
$LF: .ASCIZ <12>
```

```

:; START OF COMMON TAGS
:; CONTAINS PASS COUNT
:; CONTAINS THE TEST NUMBER
:; CONTAINS ERROR FLAG
:; CONTAINS SUBTEST ITERATION COUNT
:; CONTAINS SCOPE LOOP ADDRESS
:; CONTAINS SCOPE RETURN FOR ERRORS
:; CONTAINS TOTAL ERRORS DETECTED
:; CONTAINS ITEM CONTROL BYTE
:; CONTAINS MAX. ERRORS PER TEST
:; CONTAINS PC OF LAST ERROR INSTRUCTION
:; CONTAINS ADDRESS OF 'GOOD' DATA
:; CONTAINS ADDRESS OF 'BAD' DATA
:; CONTAINS 'GOOD' DATA
:; CONTAINS 'BAD' DATA
:; RESERVED--NOT TO BE USED

```

```

::AUTOMATIC MODE INDICATOR
::INTERRUPT MODE INDICATOR

```

```

: ADDRESS OF SWITCH REGISTER
: ADDRESS OF DISPLAY REGISTER
: TTY KBD STATUS
: TTY KBD BUFFER
: TTY PRINTER STATUS REG. ADDRESS
: TTY PRINTER BUFFER REG. ADDRESS
: CONTAINS NULL CHARACTER FOR FILLS
: CONTAINS # OF FILLER CHARACTERS REQUIRED
: INSERT FILL CHARS. AFTER A "LINE FEED"
: "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
: CONTAINS THE ADDRESS FROM
: WHICH ($REG0) WAS OBTAINED
: CONTAINS (($REGAD)+0)
: CONTAINS (($REGAD)+2)
: CONTAINS (($REGAD)+4)
: CONTAINS (($REGAD)+6)
: CONTAINS (($REGAD)+10)
: CONTAINS (($REGAD)+12)
: CONTAINS (($REGAD)+14)
: CONTAINS (($REGAD)+16)
: CONTAINS (($REGAD)+20)
: ESCAPE ON ERROR ADDRESS
: CODE FOR BELL
: QUESTION MARK
: CARRIAGE RETURN
: LINE FEED

```

J02

MAINDEC-11-DZAKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 6
 DZAKLD.P11 31-AUG-76 15:35 COMMON TAGS

SEQ 0022

; 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 (" ")

; TAGS AND GENERAL DATA AREA

001216 000000
 001220 000000

FFUNC: .WORD 0
 XXDPMO: .WORD 0

; FLAG SET, TO INDICATE ENTRY INTO FUNCTION PROGRAM
 ; IF PROGRAM LOADED BY XXDP, THE
 ; LOWER BYTE HAS THE DRIVE NUMBER
 ; AND THE UPPER BYTE CONTAINS THE RK05 'XXDP' CODE
 ; FLAG, SET TO INDICATE THAT A
 ; PARTICULAR TEST WAS SELECTED BY USER (SW 8)

001222 000

LUPSW: .BYTE 0

001223 000

DRVON: .BYTE 0

; CONTAINS NUMBER OF DRIVES THAT HAVE
 ; BEEN ALREADY CHECKED

001224 000

DRIVS: .BYTE 0

; CONTAINS TOTAL # OF DRIVES PRESENT

001226

.EVEN

001226 000000

DRVPT: 0

; CONTAINS POINTER TO INDICATOR STARTING
 ; WHICH CHECKING SHOULD BE DONE FOR NEXT
 ; AVAILABLE DRIVE
 ; CONTAINS THE ADDRESS OF THE DRIVE
 ; BEING TESTED

001230 000000

DRIVAD: 0

001232 000000

DRIVO: 000000

; THESE ARE FLAGS TO INDICATE
 ; THAT A PARTICULAR DRIVE IS
 ; PRESENT. BIT 0 IS SET TO
 ; INDICATE THAT. BITS 13, 14, 15
 ; CONTAIN THE LOGICAL DRIVE
 ; ADDRESS

001234 020000

DRIV1: 020000

001236 040000

DRIV2: 040000

001240 060000

DRIV3: 060000

001242 100000

DRIV4: 100000

001244 120000

DRIV5: 120000

001246 140000

DRIV6: 140000

001250 160000

DRIV7: 160000

001252 000000

RETRY1: 0

; GENERAL REGISTERS

001254 000000

RETRY2: 0

001256 000000

RETRY3: 0

001260 000000

INADR: 0

; CONTAINS INNER ADDRESS

001262 000000

OUTADR: 0

; CONTAINS OUTER ADDRESS

001264 000000

TIMER: 0

001266 000015

BUFR: .BLKW 13.

; GENERAL BUFFERS

001320 000015

BUFR1: .BLKW 13.

001352 000015

BUFR2: .BLKW 13.

L02

MAINDEC-11-DZAKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 8
DZAKLD.P11 31-AUG-76 15:35 COMMON TAGS

SEQ 0024

349 001464 177412
350 001466 177416
351
352 001470 000200
353
354
355
356
357
358
359
360 001472 000220
361
362
363
364
365
366
367
368 001474 000000
369 001476 000000
370 001500 000000
371 001502 000000
372
373 001504 000000
374 001506 000000
375 001510 000000
376 001512 000000
377 001514 000000
378 001516 000000
379 001520 000000
380 001522 000000
381
382
383
384
385
386
387
388
389
390 001524 000000
391 001526 000000
392 001530 013140
393 001532 000000
394 001534 000000
395 001536 000000
396 001540 000000
397
398
399 001542 014500
400 001544 000040
401 001546 013240
402 001550 000140
403 001552 001000
404 001554 002000

RKDA: 177412
RKDB: 177416

RKPRI: 200

RKVEC: 220

INDX1: 0
INDX2: 0
INDX3: 0
INDX4: 0

ERCNT1: 0
ERCNT2: 0
ERCNT3: 0
ERCNT4: 0
ERCNT5: 0
ERCNT6: 0
ERCNT7: 0
ERCNT8: 0

:CONTAINS THE CPU LEVEL (4) AT WHICH
:RK11 NORMALLY INTERRUPTS. THIS WORD
:SHOULD BE CHANGED IF RK11 IS DESIGNATED
:A BR LEVEL OTHER THAN 5.EXP: IF IT
:IS CHANGED TO 6, THE CPU LEVEL WOULD
:BE 1 LESS (5) & HENCE THIS WORD
:SHOULD BE 240 (BIT POSITIONS ARE
:IDENTICAL TO THE PRIORITY BITS IN PSW)
:CONTAINS THE NORMAL VECTOR ADDRESS
:TO WHICH THE RK11 INTERRUPTS. IF THE
:VECTOR ADDRESS HAS BEEN CHANGED, MODIFY
:THIS WORD.

:GENERAL INDEX REGISTERS

:GENERAL REGISTERS
:GENERAL REGISTERS
:GENERAL REGISTERS

:*THE FOLLOWING TABLE CONTAINS THE CYLINDERS BETWEEN WHICH THE SEEKS WILL BE
:*TIMED. THEY HAVE BEEN SELECTED TO GIVE SOME TYPICAL SEEKS TIMES FOR THE
:*3 SEEK SPEEDS. IF FOR ANY REASON YOU WANT TO TIME SEEKS BETWEEN ANY
:*OTHER SET OF CYLINDERS, MAKE CHANGES IN THE CORRESPONDING SEEK CYLINDER
:*ADDRESSES.

:*OUTER CYLINDER ADDRESS, FROM WHERE SEEK WILL BE DONE
SOAD: 0 CYLINDER 0

0	0
13140	179
0	0
0	0
0	0
0	0

:*INNER ADDRESS, TO WHICH SEEK WILL BE DONE
SIAD: 14500 CYLINDER 202, LAST

40	1
13240	181
140	3
1000	16
2000	32

M02

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 9
 DZRKLD.P11 31-AUG-76 15:35 COMMON TAGS

SEQ 0025

405 001556 006200 6200 ; " 100

: FOLLOWING POINTERS ARE USED TO TRANSFER CONTROL TO THE
 : TEST SELECTED BY USING SW 8. IF ANY MORE TESTS ARE
 : ADDED TO THIS PROGRAM ADDITIONAL POINTERS SHOULD BE INSERTED.

PT1: TST1+2
 PT2: TST2+2
 PT3: TST3+2
 PT4: TST4+2
 PT5: TST5+2
 PT6: TST6+2
 PT7: TST7+2
 PT10: TST10+2
 PT11: TST11+2

: MESSAGES & ASCII STRINGS

MSG1: .ASCIZ <15><12>/SIN/

MSG2: .ASCIZ <15><12>/SKE/

MSG3: .ASCIZ <15><12>/TEST # ABORTED:/

MSG4: .ASCIZ <15><12>/PROG ABORTED/

MSG5: .ASCIZ <15><12>/READ HDRS OK FROM CYLB ABOVE/

MSG6: .ASCIZ /EXPCTD HDR= /

MSG7: .ASCIZ / PC= /

MSG10: .ASCIZ <15><12>/CNTRL RDY DIDN'T SET/

MSG11: .ASCIZ /SECTR EXPC P-HDR RECV P-HDR/

N02

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 10
DZRKLD.P11 31-AUG-76 15:35 COMMON TAGS

SEQ 0026

461						
462	002027	015	051012	053457	MSG12: .ASCIZ	<15><12>"R/W/S RDY NOT SET"
463	002034	051457	051040	054504		
464	002042	047040	052117	051440		
465	002050	052105	000			
466						
467	002053	040	052040	054522	MSG13: .ASCIZ	/ TRY #:/
468	002060	021440	000072			
469						
470						
471	002064	005015	051104	053111	MSG14: .ASCIZ	<15><12>/DRIVE /
472	002072	020105	000			
473						
474	002075	040	020040		BLNK13: .ASCII	/ / /
475	002100	040			BLNK10: .ASCII	/ / /
476	002101	040			BLNKS9: .ASCII	/ / /
477	002102	040			BLNKS8: .ASCII	/ / /
478	002103	040			BLNKS7: .ASCII	/ / /
479	002104	040			BLNKS6: .ASCII	/ / /
480	002105	040			BLNKS5: .ASCII	/ / /
481	002106	040			BLNKS4: .ASCII	/ / /
482	002107	040			BLNKS3: .ASCII	/ / /
483	002110	040			BLNKS2: .ASCII	/ / /
484	002111	040	000		BLNKS1: .ASCIZ	/ / /
485						
486		002114				
487	002114	000000			FDRIVE: 0	EVEN
488	002116	000000			FDRVE1: 0	
489	002120	000000			DRHOLD: 0	

B03

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 11
 DZRKLD.F11 31-AUG-76 15:35 ERROR POINTER TABLE

SEQ 0027

.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

002122

\$ERRTB:

:ERROR ITEMS TABLE

:ITEM

1

002122 024250
 002124 025414
 002126 026250
 002130 000000

EM1 :CNTRL RDY DIDN'T SET AFTER SEEK
 DH1 :PC RKCS RKER RKDS RKDA
 DT1 :\$ERRPC \$REGO \$REG1 \$REG2 \$REG3
 0

:ITEM

2

002132 024307
 002134 025414
 002136 026250
 002140 000000

EM2 :SIN ON SEEK
 DH1 :PC RKCS RKER RKDS RKDA
 DT1 :\$ERRPC \$REGO \$REG1 \$REG2 \$REG3
 0

:ITEM

3

002142 024323
 002144 025414
 002146 026250
 002150 000000

EM3 :DRE ON SEEK
 DH1 :PC RKCS RKER RKDS RKDA
 DT1 :\$ERRPC \$REGO \$REG1 \$REG2 \$REG3
 0

:ITEM

4

002152 024337
 002154 025414
 002156 026250
 002160 000000

EM4 :ERR ON SEEK
 DH1 :PC RKCS RKER RKDS RKDA
 DT1 :\$ERRPC \$REGO \$REG1 \$REG2 \$REG3
 0

:ITEM

5

002162 024355
 002164 025414
 002166 026250
 002170 000000

EM5 :DRU ON SEEK; PUT DRIVE ON 'LOAD' BACK TO 'RUN'
 DH1 :PC RKCS RKER RKDS RKDA
 DT1 :\$ERRPC \$REGO \$REG1 \$REG2 \$REG3
 0

C03

MAINDEC-11-DZRLD-0 MACY11 27(1006) 04-OCT-76 14:26 PAGE 12
 DZRLD.F11 31-AUG-76 15:35 ERROR POINTER TABLE

SEQ 0028

```

:ITEM 6
      EM6      :R/W/S RDY NOT SET AFTER SEEK
      DH1      :PC      RKCS      RKER      RKDS      RKDA
      DT1      :$ERRPC $REG0      $REG1      $REG2      $REG3
      0

:ITEM 7
      EM7      :SIN ON WRITE FMT
      DH7      :PC      RKCS      RKER      RKDS      RKDA      CYLINDER
      DT7      :$ERRPC $REG0      $REG1      $REG2      $REG3      $REG4
      0

:ITEM 10
      EM10     :'ERR' ON DOING WRITE FMT
      DH7      :PC      RKCS      RKER      RKDS      RKDA      CYLINDER
      DT7      :$ERRPC $REG0      $REG1      $REG2      $REG3      $REG4
      0

:ITEM 11
      EM11     :SIN ON READ FMT
      DH11     :PC      RKCS      RKER      RKDS      RKDA:
      DT11     :DRV#    CYL      SUR      SEC
      :$ERRPC $REG0      $REG1      $REG2
      :$REG4 $REG5      $REG6      $REG7
      0

:ITEM 12
      EM12     :'ERR' ON READ FMT
      DH7      :PC      RKCS      RKER      RKDS      RKDA      CYLINDER
      DT7      :$ERRPC $REG0      $REG1      REG2      $REG3      $REG4
      0

:ITEM 13
      EM13     :WRONG HEADERS FROM 'SEC #
      DH13     :SECTOR #      HEADER RECVD
      0
      ESR13    :USE THIS SUBROUTINE FOR TYPING OUT ERROR DATA

:ITEM 14
      EM14     :ERROR ON IMPLIED SEEK FROM CYLA TO CYLB
      DH14     :PC      CYLA      CYLB      RKER      RKDS      TRY#
      DT7      :$ERRPC $REG0      $REG1      $REG2      $REG3      $REG4
      0

:ITEM 15
      MS15     :READ WRONG HDRS FROM CYLB ABOV
      DH13     :SEC#      HEADER RECVD
      0
  
```

```

46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
00
01
  
```

```

002172 024424
002174 025414
002176 026250
002200 000000

002202 024461
002204 025512
002206 026264
002210 000000

002212 024500
002214 025512
002216 026264
002220 000000

002222 024531
002224 025567
002226 026302
002230 000000

002232 024547
002234 025512
002236 026264
002240 000000

002242 024571
002244 025664
002246 000000
002250 015402

002252 024620
002254 025703
002256 026264
002260 000000

002262 024673
002264 025664
002266 000000
  
```


D03

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 13
 DZRKLD.P11 31-AUG-76 15:35 ERROR POINTER TABLE

SEQ 0029

```

602 002270 015310 ESR15 ;GO TO "ESR15" FOR TYPING OUT
603
604 ;ITEM 16
605
606 002272 024735 EM16 ;READ WRONG FIRST WORD FROM SECTOR 0, 'CYLB' (ON IMPLIED SEEK FROM CYLA
607 002274 025762 DH16 ;PC CYLA CYLB EXPCT RECVD TRY#
608 002276 026264 DT7 ;$ERRPC $REG0 $REG1 $REG2 $REG3 $REG4
609 002300 000000 0
610
611 ;ITEM 17
612
613 002302 025042 MS17 ;READ FIRST WORD FROM SECTOR 1, 'CYLB' ABOVE
614 002304 026037 DH17 ;PC CYLB EXPCT RECVD
615 002306 026324 DT17 ;$ERRPC $REG0 $REG1 $REG2
616 002310 000000 0
617
618 ;ITEM 20
619
620 002312 025110 EM20 ;READ WRONG HEADER ON IMPLIED SEEK FROM 'CYLA' TO 'CYLB'
621 002314 025664 DH13 ;SECTOR # HEADER RECVD
622 002316 000000 0
623 002320 015456 ESR20 ;USE THIS SUBROUTINE FOR TYPING OUT ERROR DATA
624
625 ;ITEM 21
626
627 002322 025176 EM21 ;EROR ON DOING WRITE ON DSK
628 002324 025414 DH1 ;PC RKCS RKER RKDS RKDA
629 002326 026250 DT1 ;$ERRPC $REG0 $REG1 $REG2 $REG3
630 002330 000000 0
631
632 ;ITEM 22
633
634 002332 025232 EM22 ;SIN ON DOING WRITE
635 002334 025414 DH1 ;PC RKCS RKER RKDS RKDA
636 002336 026250 DT1 ;$ERRPC $REG0 $REG1 $REG2 $REG3
637 002340 000000 0
638
639 ;ITEM 23
640
641 002342 025255 EM23 ;HE ON DOING READ
642 002344 025567 DH11 ;PC RKCS RKER RKDS RKDA:
643 ;DRV# CYL SUR SEC
644 002346 026302 DT11 ;$ERRPC $REG0 $REG1 $REG2 $REG3
645 ;$REG4 $REG5 $REG6 $REG7
646 002350 000000 0
647
648 ;ITEM 24
649
650 002352 025276 EM24 ;CSE ON READ
651 002354 026075 DH24 ;PC TRY# RKCS RKER RKDS RKDA:
652 ;DRV# CYL SUR SEC
653 002356 026336 DT24 ;$ERRPC $REG10 $REG0 $REG1 $REG2
654 ;$REG4 $REG5 $REG6 $REG7
655 002360 000000 0
  
```

E03

MAINDEC-11-DZKLD-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 14
 DZKLD.P11 31-AUG-76 15:35 ERROR POINTER TABLE

SEG 0030

```

658      ;ITEM 25
659      EM25 :DATA ERROR ON READ FROM DISK ADDRESS
660      DH25 :WORD# EXPCT RECVD CYL SUR SEC
661      0
662      ESR25 :USE THIS ROUTINE FOR ERROR REPORTING
663
664      ;ITEM 26
665      EM26 :HE ON WRT CHK
666      DH11 :PC RKCS RKER RKDS RKDA:
667      :DRV# CYL SUR SEC
668      DT11 :$ERRPC $REG0 $REG1 $REG2
669      0 :$REG4 $REG5 $REG6 $REG7
670
671      ;ITEM 27
672      EM27 :WRT CHK EROR
673      DH24 :PC TRY# RKCS RKER RKDS RKDA:
674      :DRV# CYL SUR SEC
675      DT24 :$ERRPC $REG10 $REG0 $REG1 $REG2
676      0 :$REG4 $REG5 $REG6 $REG7
677
678      ;ITEM 30
679      EM30 :ERROR
680      DH1 :PC RKCS RKER RKDS RKDA
681      DT1 :$ERRPC $REG0 $REG1 $REG2 $REG3
682      0
683
684      002362 025312
685      002364 026202
686      002366 000000
687      002370 015544
688
689      002372 025354
690      002374 025567
691
692      002376 026302
693      002400 000000
694
695      002402 025372
696      002404 026075
697
698      002406 026336
699      002410 000000
700
701      002412 025407
702      002414 025414
703      002416 026250
704      002420 000000
705
706      000000
707      000000
708      000000
709      000000
710      000000
711      000000
712      000000
713      000000
714      000000
715      000000
716      000000
717      000000
718      000000
719      000000
720      000000
721      000000
722      000000
723      000000
724      000000
725      000000
726      000000
727      000000
728      000000
729      000000
730      000000
731      000000
732      000000
733      000000
734      000000
735      000000
736      000000
737      000000
738      000000
739      000000
740      000000
741      000000
742      000000
743      000000
744      000000
745      000000
746      000000
747      000000
748      000000
749      000000
750      000000
751      000000
752      000000
753      000000
754      000000
755      000000
756      000000
757      000000
758      000000
759      000000
760      000000
761      000000
762      000000
763      000000
764      000000
765      000000
766      000000
767      000000
768      000000
769      000000
770      000000
771      000000
772      000000
773      000000
774      000000
775      000000
776      000000
777      000000
778      000000
779      000000
780      000000
781      000000
782      000000
783      000000
784      000000
785      000000
786      000000
787      000000
788      000000
789      000000
790      000000
791      000000
792      000000
793      000000
794      000000
795      000000
796      000000
797      000000
798      000000
799      000000
800      000000

```


F03

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 15
 DZRKLD.P11 31-AUG-76 15:35 ERROR POINTER TABLE

SEQ 0031

```

695
696
697
698
699
700 002422 011600
701 002424 005740
702 002426 022626
703 002430 104401 002436
704 002434 000407
705
706 002454
707 002454 010046
708 002456 104402
709 002460 000000
710
711 002462 000005
712
713
714 002464 012706 001100
715 002470 005026
716 002472 022706 001140
717 002476 001374
718 002500 012706 001100
719
720 002504 012737 016732 000020
721 002512 012737 000340 000022
722 002520 012737 017106 000030
723 002526 012737 000340 000032
724 002534 012737 022616 000034
725 002542 012737 000340 000036
726 002550 012737 022724 000024
727 002556 012737 000340 000026
728 002564 005037 001204
729 002570 112737 000001 001115
730 002576 012737 002576 001106
731 002604 012737 002604 001110
732
733
734 002612 013746 000004
735 002616 012737 002652 000004
736 002624 012737 177570 001140
737 002632 012737 177570 001142
738 002640 022777 177777 176272
739 002646 001012
740
741 002650 000403
742 002652 012716 002660
743 002656 000002
744 002660 012737 000176 001140
745 002666 012737 000174 001142
746 002674 012637 000004
747
748 002700 004737 020536
749
750

;THIS IS THE HANDLER FOR UNEXPECTED TIME OUT. PRESSING CONTINUE WILL
;RESTART THE PROGRAM.

BADTMO: MOV (SP),RO ;SAVE PC WHERE TIME OUT OCCURED
        TST -(RO)
        CMP (SP)+,(SP)+ ;RESTORE STACK POINTER
        TYPE 65$ ;TYPE ASCIZ STRING
        BR 64$ ;GET OVER THE ASCIZ
;65$: .ASCIZ <15><12>/TIMOUT:PC=/
64$:
        MOV RO,-(SP) ;SET UP FOR TYPING OUT PC
        TYPOC ;GO TYPE OUT OCTAL PC
        HALT

START: RESET ;CLEAR THE BUS
;SBTTL INITIALIZE THE COMMON TAGS
;:CLEAR THE COMMON TAGS ($CMTAG) AREA
        MOV #CMTAG,R6 ;:FIRST LOCATION TO BE CLEARED
        CLR (R6)+ ;:CLEAR MEMORY LOCATION
        CMP #SWR,R6 ;:DONE?
        BNE -6 ;:LOOP BACK IF NO
        MOV #STACK,SP ;:SETUP THE STACK POINTER
;:INITIALIZE A FEW VECTORS
        MOV #SCOPE,3#IOTVEC ;:IOT VECTOR FOR SCOPE ROUTINE
        MOV #340,3#IOTVEC+2 ;:LEVEL 7
        MOV #ERROR,3#EMTVEC ;:EMT VECTOR FOR ERROR ROUTINE
        MOV #340,3#EMTVEC+2 ;:LEVEL 7
        MOV #STRAP,3#TRAPVEC ;:TRAP VECTOR FOR TRAP CALLS
        MOV #340,3#TRAPVEC+2 ;:LEVEL 7
        MOV #SPWRDN,3#PWRVEC ;:POWER FAILURE VECTOR
        MOV #340,3#PWRVEC+2 ;:LEVEL 7
        CLR $ESCAPE ;:CLEAR THE ESCAPE ON ERROR ADDRESS
        MOVB #1,$ERMAX ;:ALLOW ONE ERROR PER TEST
        MOV #,$SLPADR ;:INITIALIZE THE LOOP ADDRESS FOR SCOPE
        MOV #,$SLPERR ;:SETUP THE ERROR LOOP ADDRESS
;:SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
;:EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
        MOV 3#ERRVEC,-(SP) ;:SAVE ERROR VECTOR
        MOV #64$,3#ERRVEC ;:SET UP ERROR VECTOR
        MOV #DSWR,SWR ;:SETUP FOR A HARDWARE SWICH REGISTER
        MOV #DDISP,DISPLAY ;:AND A HARDWARE DISPLAY REGISTER
        CMP #-1,3#SWR ;:TRY TO REFERENCE HARDWARE SWR
        BNE 66$ ;:BRANCH IF NO TIMEOUT TRAP OCCURRED
        ;:AND THE HARDWARE SWR IS NOT = -1
        BR 65$ ;:BRANCH IF NO TIMEOUT
64$: MOV #65$,(SP) ;:SET UP FOR TRAP RETURN
65$: MOV #SWREG,SWR ;:POINT TO SOFTWARE SWR
        MOV #DISPREG,DISPLAY
66$: MOV (SP)+,3#ERRVEC ;:RESTORE ERROR VECTOR

        JSR PC,$TKINT ;INITIALIZE THE TTY HANDLER
;SBTTL TYPE PROGRAM NAME
;:TYPE THE NAME OF THE PROGRAM IF FIRST PASS

```

G03

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 16
 DZRKLD.P11 31-AUG-76 15:35 TYPE PROGRAM NAME

SEQ 0032

```

751 002704 005227 177777 INC #1 ;:FIRST TIME?
752 002710 001043 BNE 67$ ;:BRANCH IF NO
753 002712 104401 002750 TYPE 68$ ;:TYPE ASCIZ STRING
754 .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
755 002716 005737 000042 TST 0#42 ;:ARE WE RUNNING UNDER XXDP/ACT?
756 002722 001006 BNE 69$ ;:BRANCH IF YES
757 002724 023727 001140 000176 CMP SWR,#SWREG ;:SOFTWARE SWITCH REG SELECTED?
758 002732 001005 BNE 70$ ;:BRANCH IF NO
759 002734 104406 GTSWR ;:GET SOFT-SWR SETTINGS
760 002736 000403 BR 70$
761 002740 112737 000001 001134 69$: MOVB #1,$AUTOB ;:SET AUTO-MODE INDICATOR
762 002746 70$:
763 002746 000424 BR 67$ ;:GET OVER THE ASCIZ
764 ;:68$: .ASCIZ <CRLF>/RK11 DYNAMIC TEST/<15><12>/MAINDEC-11-DZRKL-D/<CRLF>
765 003020 67$:
766 003020 105737 001216 START1: TSTB FFUNC ;:FUNCTION PROGRAM SELECTED?
767 003024 001404 BEQ 7$ ;:NO
768 003026 105037 001216 CLRB FFUNC ;:YES, CLEAR THE FLAG
769 003032 000137 023106 JMP 0#FUNBEG ;:GO TO 'FUNCTION SELECTION PROGRAM'
770 003036 012700 001220 7$: MOV #XXDPMD,RO ;:CLEAR FLAGS FROM
771 003042 105020 5$: CLRB (RO)+ ;:'XXDPMD' TO 'DRIVAD'
772 003044 020027 001232 CMP RO,#DRIVAD+2
773 003050 001374 BNE 5$
774 003052 012701 177770 MOV #-10,R1
775 003056 042720 000003 6$: BIC #3,(R0)+ ;:CLEAR BIT 0'S IN 'DRIVE
776 003062 005201 INC R1 ;:PRESENT' FLAGS.
777 003064 001374 BNE 6$
778
779 ;:THE FOLLOWING CODE FINDS OUT THE PROGRAM CONTROL MODE:
780 ;:PAPER TAPE (MANUAL), ACT11, RKDP CHAIN OR DUMP
781
782 003066 122737 000002 000041 CMPB #2,41 ;:LOADED FROM AN RK05 ?
783 003074 001160 BNE ST2 ;:BR IF NOT
784 003076 013737 000040 001220 MOV 40,XXDPMD ;:GET DEVICE INDICATOR AND DRIVE ADDRESS OF
785 ;:LOADING RK05
786 003104 122737 000010 001220 CMPB #10,XXDPMD ;:DRIVE ADDRESS 7 OR LESS ?
787 003112 101002 BHI 2$ ;:BR IF YES
788 003114 105037 001220 CLRB XXDPMD ;:DRIVE ZERO LOADED THE PROGRAM
789 003120 005737 000042 2$: TST 42 ;:CHAIN MODE OR ACT11 AUTO ACCEPT ?
790 003124 001424 BEQ 3$ ;:BR IF NEITHER
791 003126 104401 003134 TYPE 65$ ;:TYPE ASCIZ STRING
792 003132 000413 BR 64$ ;:GET OVER THE ASCIZ
793 ;:65$: .ASCIZ <15><12>/NOT TESTING DRIVE /
794 64$:
795 003162 005046 CLR -(SP) ;:CLEAR WORD ON STACK
796 003164 113716 001220 MOVB XXDPMD,(SP) ;:GET DRIVE ADDRESS
797 003170 104403 TYP0S ;:TYPE THE ADDRESS
798 003172 001 .BYTE 1 ;:ONLY 1 CHARACTER
799 003173 000 .BYTE 0 ;:SUPPRESS LEADING ZEROS
800 003174 000520 BR ST2 ;:GET NUMBER OF DRIVES
801 003176 005227 177777 3$: INC #-1 ;:FIRST TIME THROUGH HERE ?
802 003202 001115 BNE ST2 ;:BR IF NOT FIRST TIME
803 003204 104401 003212 TYPE 67$ ;:TYPE ASCIZ STRING
804 003210 000411 BR 66$ ;:GET OVER THE ASCIZ
805 ;:67$: .ASCIZ <15><12>/TO TEST DRIVE /
806 66$:

```


H03

MAINDEC-11-DZKRL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 17
 DZKRLD.F11 31-AUG-76 15:35 GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0033

```

807 003234 005046          CLR      -(SP)          :CLEAR WORD ON THE STACK
808 003236 113716 001220  MOV      XXDPMO,(SP)      :GET DRIVE ADDRESS
809 003242 104403          TYP0S          :TYPE THE DRIVE ADDRESS
810 003244      001      .BYTE      1          :ONLY 1 CHARACTER
811 003245      000      .BYTE      0          :SUPRESS LEADING ZEROS
812 003246 104401 003254  TYPE      69$          :TYPE ASCIZ STRING
813 003252 000431          BR      68$          :GET OVER THE ASCIZ
814          :69$: .ASCIZ / HALT PROGRAM, REMOVE RKDP PACK AND REPLACE IT/<15><12>
815          68$:
816 003336 104401 003344  TYPE      71$          :TYPE ASCIZ STRING
817 003342 000435          BR      70$          :GET OVER THE ASCIZ
818          :71$: .ASCIZ /WITH A WORK PACK, CLEAR LOCATION 40, AND RESTART PROGRAM/
819          70$:
820
821 003436 012737 002422 000004 ST2:  MOV      #BADTMO,ERRVEC :SET TIMEOUT VECTOR FOR
822                                     :UNEXPECTED TIME OUT
823
824                                     :THIS CODE FINDS WHICH DRIVES ARE PRESENT & PRINTS OUT THE DRIVE
825                                     :DRIVE NUMBERS THAT WERE FOUND ON LINE.
826
827
828 003444 104401 003452          TYPE      65$          :TYPE ASCIZ STRING
829 003450 000411          BR      64$          :GET OVER THE ASCIZ
830          :65$: .ASCIZ <15><12>/DRIVES PRESENT/
831          64$:
832 003474 105037 001224  CLRB      DRVS          :INITIALIZE NO. OF DRVS PRESENT
833 003500 005001          CLR      R1
834 003502 012702 001232  MOV      #DRIVO,R2
835 003506 005003          CLR      R3          :INITIALIZE COUNT TO 0
836 003510 005737 001220  TST      XXDPMO      :LOADED FROM AN RKDS ?
837 003514 001403          BEQ      6$          :BR IF NOT
838 003516 120337 001220  CMPB      R3,XXDPMO      :CHECKING THE LOAD DRIVE ?
839 003522 001411          BEQ      2$          :BR IF YES
840 003524 010177 175734  MOV      R1,DRKDA      :ADRES A DRIVE
841 003530 105777 175716  TSTB      DRKDS          :IS IT PRESENT?
842 003534 100004          BPL      2$          :NO, BRANCH
843 003536 105237 001224  INCB      DRVS          :INCREMENT TOTAL # OF DRVS
844 003542 052712 000001  BIS      #1,(R2)      :SET FLAG INDICATING THIS DRV PRSNT
845 003546 005722          TST      (R2)+
846 003550 005203          INC      R3          :INCREMENT COUNT
847 003552 062701 020000  ADD      #20000,R1      :ADRES THE NXT DRV
848                                     :CHKD ALL 8 DRIVES?
849 003556 001354          BNE      1$          :IF NOT, GO CHK IF NEXT DRV PRSNT
850
851 003560 004737 024164  JSR      PC,SIZEF      :FIND WHICH ARE FS
852 003564 105737 001224  TSTB      DRVS          :WERE ANY DRIVES FOUND?
853 003570 001010          BNE      3$          :YES, BRANCH
854 003572 104401 003600  TYPE      67$          :TYPE ASCIZ STRING
855 003576 000403          BR      66$          :GET OVER THE ASCIZ
856          :67$: .ASCIZ / NONE/
857          66$:
858 003606 000137 015172  JMP      $EOP          :IF NONE WERE FOUND, GO
859                                     :TO THE END OF PROGRAM
860 003612 005002          CLR      R2          :DRIVE NUMBER
861 003614 012700 001232  MOV      #DRIVO,R0      :TABLE OF AVAIL DRIVES
862 003620 105710          TSTB      (R0)          :DRIVE HERE?

```

MAINDEC-11-DZRL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 18
 DZRLD.P11 31-AUG-76 15:35 GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0034

```

863 003622 001414 BEQ 4$ ;NO
864 003624 104401 TYPE
865 003626 001213 $CRLF
866 003630 010246 MOV R2,-(SP) ;PUSH NO ON THE STACK
867 003632 104403 TYPOS ;TO TYPE OCTAL NO.
868 003634 001 .BYTE 1 ;TYPE 1 DIGIT, SUPPRESS LDG 0'S
869 003635 000 .BYTE 0
870 003636 032710 000002 BIT #2,(R0) ;IS IT RK05F?
871 003642 001404 BEQ 4$ ;NO
872 003644 104401 003652 TYPE ,69$ ;TYPE ASCIZ STRING
873 003650 000401 BR ,68$ ;GET OVER THE ASCIZ
874 .:69$: .ASCIZ /F/
875 003654 68$:
876 003654 005202 4$: INC R2 ;POINT TO NEXT DRIVE #
877 003656 005720 TST (R0)+ ;NEXT DRIVE IN TABLE
878 003660 020027 001251 CMP R0,#DRIV7+1 ;ALL DONE?
879 003664 002755 BLT 5$ ;NO, CHECK REST
880
881 ;FIND OUT THE FIRST (FROM 0-7) DRIVE THAT IS PRESENT. PUT THE ADDRESS
882 ;OF THAT DRIVE IN 'DRIVAD'. INDICATE THAT DRIVE # WILL
883 ;BE TESTED.
884
885 003666 012737 001232 001226 ST3: MOV #DRIVO,DRVPTR
886 003674 105037 001223 CLR DRVDON
887 003700 005037 001230 CLR DRIVAD
888 003704 105037 001102 NXTDRV: CLR $TSTNM ;RESET TEST NUMBER TO 1
889 003710 005037 001112 CLR $ERTL ;CLEAR ERROR COUNT FOR THIS DRIVE
890 003714 013701 001226 MOV DRVPTR,R1
891 003720 032721 000001 1$: BIT #1,(R1)+ ;IS THIS DRIVE PRESENT?
892 003724 001005 BNE 2$ ;YES, BRANCH
893 003726 020127 001252 4$: CMP R1,#DRIV7+2 ;CHECKED THE WHOLE LIST?
894 003732 001372 BNE 1$ ;NO
895 003734 000137 015172 JMP $EOP ;YES, EXIT
896 003740 010137 001226 2$: MOV R1,DRVPTR ;NO, GO AHEAD
897 003744 014104 MOV -(R1),R4 ;GET DRIVE NO. TO BE TESTED
898 003746 005037 002116 CLR FDRVE1
899 003752 005037 002114 CLR FDRIVE ;SHOWS F IF -1
900 003756 032704 000002 BIT #2,R4 ;RK-05F?
901 003762 001410 BEQ 7$ ;NO
902 003764 005237 002116 INC FDRVE1 ;SHOWS F
903 003770 032704 020000 BIT #20000,R4 ;EVEN DRIVE?
904 003774 001003 BNE 7$ ;NO
905 003776 012737 177777 002114 7$: MOV #-1,FDRIVE ;RK05F AND EVEN
906 004004 042704 000003 BIC #3,R4
907 004010 010437 001230 MOV R4,DRIVAD ;SET UP DRIVE ADRES
908 004014 104401 002064 TYPE ,MSG14
909 004020 000241 CLC
910 004022 006104 ROL R4 ;TYPE OUT THE DRIVE NO.
911 004024 006104 ROL R4
912 004026 006104 ROL R4
913 004030 006104 ROL R4
914 004032 010446 MOV R4,-(SP)
915 004034 104403 TYPOS
916 004036 001 .BYTE 1
917 004037 000 .BYTE 0
918

```


J03

MAINDEC-11-DZAKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 19
 DZAKLD.P11 31-AUG-76 15:35 GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0035

```

919 004040 005737 002116      TST   FDRVE1      ;RK-05F?
920 004044 001404      BEQ    6$         ;NO
921 004046 104401 004054      TYPE   65$        ;TYPE ASCIZ STRING
922 004052 000401      BR     64$        ;GET OVER THE ASCIZ
923      ;:65$: .ASCIZ /F/
924 004056      64$:
925 004056      6$:
926      ;IF SW 8 IS SET THEN FIND OUT WHICH TEST NUMBER IS
927      ;SELECTED AND JUMP TO THAT TEST.
928
929 004056 105037 001222      CLRB   LUPSW      ;CLEAR FLAG INDICATING SW8 SET
930 004062 032777 000400 175050 BIT    #SW8,DSWR   ;SW 8 SET?
931 004070 001445      BEQ    TST1      ;NO, BRANCH
932 004072
933 004072 104401 004100 5$:   TYPE   67$        ;TYPE ASCIZ STRING
934 004076 000410      BR     66$        ;GET OVER THE ASCIZ
935      ;:67$: .ASCIZ <15><12>/OCTAL TEST#?/
936      66$:
937 004120
938 004120 104412      RDOCT
939 004124 001762      MOV    (SP)+,RO
940 004126 020027 000011      BEQ    5$
941 004132 003357      CMP     RO,#11      ;CHECK TYPED IN TEST #
942 004134 110037 001102      BGT    5$      ;IS LEGAL, IF NOT ASK
943 004140 005300      MOVB   RO,$TSTNM
944 004142 006300      DEC     RO      ;FORM POINTERS FOR THE TEST #
945 004144 016037 001560 001106 MOV    PT1(RO),$LPADR ;ADJUST POINTERS FOR SCOPE
946 004152 013737 001106 001110 MOV    $LPADR,$LPERR ;LOOP, ETC.
947 004160 105237 001222      INCB   LUPSW   ;SET FLAG INDICATING TEST #
948      ;SELECTED
949 004164 000177 174716      JMP     @SLPADR ;GO TO THE TEST SELECTED
950
951
952
953
954
955
956
957 004170 005000
958 004172 005001
959 004174 005201
960 004176 001376
961 004200 105200
962 004202 001374
963
964
965
966
967
968
969
970
971
972
973
974 004204 000004

```

;ON RECOVERY FROM POWER FALIURE RETURN HERE

```

PWRFL: CLR    RO
        CLR    R1
1$:     INC     R1
        BNE    1$
        INCB   RO
        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.
*****
†TST1: SCOPE

```

K03

 MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 20
 DZRKLD.P11 31-AUG-76 15:35 T1

CHECK INNER LIMIT SWITCH & ELECTROMECHANICAL INTEGRITY

SEQ 0036

975	004206	005000				CLR	R0	: INITIALIZE COUNT
976	004210	005001				CLR	R1	: INITIALIZE COUNT FOR # OF SEEKS
977	004212	005002				CLR	R2	: CONTAINS SEEK ADRES
978	004214	012737	004246	001110		MOV	#20\$, \$LPERR	: SET RETURN ADRES FOR LUPING
979								: ON ERROR
980	004222	012703	001266			MOV	#BUFR, R3	: INITIALIZE POINTER TO THE TABLE
981	004226	012704	177767			MOV	#-11, R4	: ALLOW ONLY 9 CNTRL-RDY, SIN, OR R/W/S RDY ERORS
982	004232	012705	177770			MOV	#-10, R5	: ALLOW ONLY 8 DRU+DRE+ERR+DRY ERRORS
983	004236	000402				BR	2\$	
984								
985	004240	005703			1\$:	TST	R3	: WAS THERE ANY ERROR?
986	004242	001403				BEQ	3\$	: NO, BRANCH
987	004244	005003			2\$:	CLR	R3	: CLR ERROR FLAG
988	004246	104415			20\$:	CON.RESET		: GO DO CNTRL RESET. SUB ROUTINE
989								: AT 'CNT.RST'
990	004250	104416				DRV.RESET		: GO TO 'DRV.RST' & DO DRV RESET
991								
992	004252	013777	001230	175204	3\$:	MOV	DRIVAD, DRKDA	: ADRES THE DRIVE
993	004260	050277	175200			BIS	R2, DRKDA	: SET SEEK ADRES
994	004264	105777	175162			TSTB	DRKDS	: DRIVE RDY?
995	004270	100406				BMI	21\$	: YES
996	004272	004737	016106			JSR	PC, GT4RG	: NO, GET RKCS, ER, DS, DA
997	004276	104030				ERROR	30	: DRIVE RDY BIT IS NOT SET
998								: IN RKDS
999	004300	005203				INC	R3	: SET ERROR FLAG
1000	004302	005205				INC	R5	: ALLOW ONLY 5 ERORS, IF MORE
1001	004304	001515				BEQ	18\$	: ABORT
1002								
1003	004306	012777	000011	175142	21\$:	MOV	#11, DRKCS	: GO, SEEK
1004	004314	005200			4\$:	INC	R0	: WAIT FOR CNTRL RDY
1005	004316	001007				BNE	5\$	: WAITED LONG?
1006								: IF YES, ERROR
1007	004320	004737	016106			JSR	PC, GT4RG	: GO, GET RKCS, ER, DS, DA
1008	004324	104001				ERROR	1	: CNTRL RDY DIDN'T SET AFTER
1009								: SEEK WAS DONE TO CYLINDER
1010								: SHOWN IN RKDA. GO TO 'RK11 LOGIC TESTS'
1011	004326	005203				INC	R3	: SET ERROR FLAG
1012	004330	005204				INC	R4	: EXIT THIS TEST IF THERE R 5 OR MORE ERRORS
1013	004332	001502				BEQ	18\$	
1014	004334	000403				BR	6\$	
1015	004336	105777	175114		5\$:	TSTB	DRKCS	: DID CNTRL RDY SET?
1016	004342	100364				BPL	4\$	: IF NOT WAIT FOR IT
1017								
1018	004344	005000			6\$:	CLR	R0	: INITIALIZE COUNT
1019	004346	032777	000100	175076		BIT	#100, DRKDS	: R/W/S RDY SET?
1020	004354	001010				BNE	7\$	: YES
1021	004356	005200				INC	R0	: WAIT FOR R/W/S RDY
1022	004360	001372				BNE	6\$+2	
1023	004362	004737	016106			JSR	PC, GT4RG	: GET RKCS, ER, DS, DA
1024	004366	104006				ERROR	6	: R/W/S RDY DID NOT SET WHEN SEEK
1025								: WAS DONE TO CYLINDER INDICATED IN RKDA
1026	004370	005203				INC	R3	: SET ERROR FLAG
1027	004372	005204				INC	R4	: IF MAXM EROR COUNT, ABORT
1028	004374	001461				BEQ	18\$	
1029	004376	032777	001000	175046	7\$:	BIT	#1000, DRKDS	: SIN ERROR?
1030	004404	001406				BEQ	8\$	: NO, BRANCH

L03

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 21
 DZRKLD.P11 31-AUG-76 15:35 T1 CHECK INNER LIMIT SWITCH & ELECTROMECHANICAL INTEGRITY

SEQ 0037

1031	004406	004737	016106		JSR	PC,GT4RG	:GO, GET RKCS, ER, DS, DA
1032	004412	104002			ERROR	2	:SIN ERROR, ON DOING SEEK TO
1033							:CYL AS SHOWN IN RKDA
1034	004414	005203			INC	R3	:SET ERROR FLAG
1035	004416	005204			INC	R4	:IF MAXM EROR COUNT REACHED,
1036	004420	001447			BEQ	18\$	:ABORT THE TEST
1037	004422	005777	175026	8\$:	TST	ARKER	:DRE ERROR?
1038	004426	100006			BPL	10\$	:NO, BRANCH
1039							
1040	004430	004737	016106		JSR	PC,GT4RG	:GO, GET RKCS, ER, DS, DA
1041	004434	104003			ERROR	3	:DRE ON DOING SEEK TO CYLINDER
1042							:AS SHOWN IN RKDA
1043	004436	005203			INC	R3	:SET ERROR FLAG
1044	004440	005205			INC	R5	:IF MAXM EROR COUNT REACHED,
1045	004442	001767			BEQ	8\$	:ABORT THE TEST
1046							
1047	004444	005777	175006	10\$:	TST	ARKCS	: 'ERR' BIT IN RKCS SET?
1048	004450	100006			BPL	12\$	:NO, BRANCH
1049	004452	004737	016106		JSR	PC,GT4RG	:GO, GET RKCS, ER, DS, DA
1050	004456	104004			ERROR	4	: 'ERR' IN RKCS SET, ON DOING SEEK
1051							:TO CYL AS SHOWN IN RKDA. NOTE
1052							:WHICH BIT IN RKER SET?
1053	004460	005203			INC	R3	:SET ERROR FLAG
1054	004462	005205			INC	R5	:IF MAXM EROR COUNT REACHED,
1055	004464	001425			BEQ	18\$	:ABORT THE TEST
1056							
1057	004466	032777	002000	174756	12\$:	BIT	#2000,ARKDS
1058	004474	001406			BEQ	15\$	:DRU SET?
1059	004476	004737	016106		JSR	PC,GT4RG	:NO, BRANCH
1060	004502	104005			ERROR	5	:GO, GET RKCS, ER, DS, DA
1061							:DRU SET, THIS IS AN IRRECOVERABLE
1062							:ERROR. HENCE PUT THE DRIVE ON
1063							:LOAD, BACK TO RUN. DRU ERROR
1064							:SHOULD BE CLEARED, IF IT IS NOT
1065							:1) THE HEAD POSITION TRANSDUCER LAMP
1066							:IS INOPERATIVE
1067							:2) OR ERASE OR WRT CURRENT PRESENT
1068	004504	005203			INC	R3	:WITHOUT 'WRT GATE'
1069	004506	005205			INC	R5	:SET EROR FLAG
1070	004510	001413			BEQ	18\$	:ALLOW ONLY 5 ERRORS
1071							:IF MORE THAN 5
1072							:GO TO THE END OF THE PROGRAM
1073	004512	005702		15\$:	TST	R2	:WAS SEEKING TO 0 OR 312?
1074	004514	001402			BEQ	16\$	:TO 0, BRANCH
1075							:TO 312,
1076	004516	005002			CLR	R2	:SEEK NXT TIME TO 0
1077	004520	000647			BR	1\$	:GO BAK & SK TO 0
1078							
1079	004522	012702	014500	16\$:	MOV	#14500,R2	:SEEK NXT TIME TO 312
1080							
1081	004526	005201			INC	R1	:DONE SEEKS 200 TIMES?
1082	004530	022701	000200		CMP	#200,R1	
1083							
1084	004534	001241			BNE	1\$	:IF NOT, GO BAK
1085	004536	000404			BR	TST2	::EXIT
1086							

MO3

MAINDEC-11-DZRLD-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 22
 DZRLD.P11 31-AUG-76 15:35 T1 CHECK INNER LIMIT SWITCH & ELECTROMECHANICAL INTEGRITY

SEQ 0038

```

1087
1088 004540 104401 001640 18$: TYPE MSG4
1089 004544 000137 015150 JMP TST12
1090
1091 *****
1092 :*TEST 2 FORMAT THE DISK
1093 :*THIS PROGRAM ASSUMES AN UNFORMATTED DISK AND ITS
1094 :*FORMATTING IS DONE IN THIS TEST. A SECTOR IS FORMATTED
1095 :*AT A TIME. THE FIRST OWRD OF EVERY SECTOR IS WRITTEN
1096 :*TO BE A PSUEDO-HEADER CONTAINING THE DRIVE #, CYLINDER
1097 :*#, SURFACE AND SECTOR #. THE FOLLOWING IS CHECKED
1098 :*1. 'SIN' IF 'SIN' OCCURS, A DRIVE RESET IS DONE
1099 :*AND THE SAME SECTOR IS FORMATTED AGAIN. THREE
1100 :*RETRIES ARE DONE BEFORE AN ERROR MESSAGE IS PRINTED.
1101 :*2. 'ERR' ON FINDING THAT THE 'ERR' BIT SET, RKR
1102 :*SCANNED TO FIND OUT WHAT CAUSED IT AND THE
1103 :*ERROR IS REPORTED.
1104 *****
1105 TST2: SCOPE
1106 MOV DRIVAD,DRHOLD ;SAVE DRIVE NUMBER
1107 TST FDRIVE ;SEE IF EVEN RK-05F DRIVE
1108 BNE 11$ ;YES
1109 TST FDRVE1 ;ODD RK-05F?
1110 BNE TST3 ;DO NOT FORMAT IF ODD RK-05F
1111
1112 11$: MOV #-626,R2 ;203 CYLINDERS, (406 TRAKS)
1113 MOV #-14,R3 ;12 SECTORS
1114 MOV #-5,R1 ;ALLOW ONLY 5 'SIN' ERRORS
1115 MOV #-5,R5 ;ALLOW ONLY 5 'ERR'S
1116 MOV DRIVAD,R4 ;STORE ADRES OF DRIVE.
1117 4$: CON.RESET
1118 DRV.RESET ;GO TO 'DR-RST' & DO DRIVE RESET
1119 1$: CLR R0 ;KEEP COUNT OF 'SIN' ERORS
1120 ;ALLOW 3 RETRIES ON SIN
1121 TST @RKCS ;ERR?
1122 BPL 3$ ;NO
1123
1124 CON.RESET ;GO TO 'CN-RST' & DO CONTROL RESET
1125
1126 3$: CLR -(SP)
1127 MOV #12$,-(SP)
1128 RTI ;SET PRIORITY TO ZERO
1129 12$: MOV R4,OUTBUF ;WRITE THIS WORD
1130 MOV @OUTBUF,@RKBA ;FROM THIS ADRES
1131 MOV R4,@RKDA ;ON THIS DISK SECTOR
1132 MOV #-1,@RKWC ;WRITE 1 WORD
1133 MOV #4$, $LPERR ;SET RETURN ADDRESS FOR
1134 ;LUPING ON ERROR
1135
1136 MOV #2003,@RKCS ;GO WRT FMT
1137
1138 CON.RDY
1139 BIT #1000,@RKDS ;WAIT FOR CONTROL READY
1140 BEQ 6$ ;WAS THERE A SIN?
1141 JSR PC,GT5RG ;NO, SKIP DOING DRV RESET
1142 ERROR 7 ;GO, GET RKCS, ER, DS, DA, CYLINDER
;SIN ERROR ON TRYING TO
  
```


Line	Address	Offset	Value	Label	Operation	Comment
1143						;WRT FMT ON CYLINDER AS
1144						;INDICATED IN RKDA. 3 RETRIES
1145						;ARE DONE
1146						;NOTE THAT BEFORE
1147	004726	104415			CON.RESET	;RETRYING A DRIVE RESET WAS DONE
1148	004730	104416			DRV.RESET	;GO TO 'DR-RST' & DO DRV RESET
1149	004732	005200			INC R0	;INCRMNT SIN COUNT
1150	004734	022700	000003		CMP #3,R0	;ALLOW 3 RETRIES WERE THERE 3?
1151	004740	001332			BNE 1\$+2	;IF NOT, GO & RETRY
1152						
1153	004742	005201			INC R1	;ALLOW ONLY 12 SIN ERRORS
1154						;IF MORE THAN 5 EXIT THIS TEST
1155	004744	001436			BEQ 9\$	;IF MORE THAN 5 EXIT THIS TEST
1156	004746	005777	174504	6\$:	TST 3RKCS	;DID 'ERR' BIT SET IN RKCS?
1157	004752	100005			BPL 7\$	;NO, BRANCH
1158	004754	004737	016062		JSR PC,GT5RG	;GO, GET RKCS, ER, DS,DA,CYL
1159	004760	104010			ERROR 10	; 'ERR' OCCURED WHILE DOING
1160						;WRT FMT ON SECTOR, CYLINDER
1161						;AS INDICATED IN RKDA.
1162	004762	005205			INC R5	;ALLOW ONLY 5 'ERR'S. IF
1163	004764	001426			BEQ 9\$	;MORE THAN 5 EXIT THIS TEST
1164	004766	005204		7\$:	INC R4	;INCRMNT DISK ADRES TO NXT SCTR
1165	004770	005203			INC R3	;ALL 12 SECTORS DONE?
1166	004772	001314			BNE 1\$	;IF NOT, GO BAK & FMT NXT SCTR
1167						;IF YES
1168	004774	012703	177764		MOV #-14,R3	;RESET COUNT FOR 12 SECTORS
1169	005000	042704	000017		BIC #17,R4	;CLR OUT SEC BITS
1170						
1171	005004	062704	000020	8\$:	ADD #20,R4	;ADRES THE NXT TRAK TO B FMTED
1172	005010	005202			INC R2	;ALL TRAKS FMTED?
1173	005012	001304			BNE 1\$	;IF NOT GO BAK B FMT NXT CYL, SUR 0
1174	005014	005237	002114		INC FDRIVE	;EVEN RKCSF?
1175	005020	001004			BNE 10\$	;NO
1176	005022	062737	020000	001230	ADD #20000,DRIVAD	;FORMAT ODD DRIVE OF F
1177	005030	000661			BR 11\$	
1178	005032	013737	002120	001230	MOV DRHOLD,DRIVAD	;RESTORE DRIVE ADDR
1179	005040	000402		10\$:	BR TST3	;EXIT
1180						
1181	005042	004737	016716	9\$:	JSR PC,ABRT	
1182						
1183						*****
1184						;*TEST 3 READ FORMAT OF THE DISK
1185						;* IN THIS TEST, THE HEADERS FROM ALL THE SECTORS ARE READ
1186						;* & CHECKED IF THEY ARE CORRECT. THE FOLLOWING IS THE
1187						;* TEST SEQUENCE.
1188						;* 1. READ 12 SECTORS (HDRS ONLY) AT A TIME
1189						;* 2. IF THERE IS A 'SIN' ERROR RETRY ONCE MORE, IF SAW AGAIN
1190						;* REPORT ERROR & READ HEADER FROM NEXT CYLINDER
1191						;* 3. IF THERE IS 'ERR' IN RKCS, DO A CONTROL RESET, REPORT
1192						;* ERROR & READ HEADER FROM NEXT CYLINDER. IF THERE ARE
1193						;* MORE THAN 5 ERRORS OF THIS KIND, THIS TEST WILL BE EXITED
1194						;* 4. THE 12 HEADERS ARE CHECKED. IF THEY ARE CORRECT THE
1195						;* NEXT CYLINDER IS READ.
1196						;* IF THEY ARE NOT CORRECT, A RETRY IS DONE; IF AGAIN CORRECT
1197						;* HEADERS ARE NOT RECIEVED, AN ERROR IS REPORTED. THE
1198						;* SECTOR #'S GIVING THE BAD HEADERS, & THE BAD HEADERS ARE

B04

MAINDEC-11-DZKRL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 24
 DZKRLD.F11 31-AUG-76 15:35 T3 READ FORMAT OF THE DISK

SEQ 0040

```

1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254

```

005046	000004					ST3: SCOPE	
005050	012737	177773	001504			MOV	#-5,ERCNT1
005056	012737	177766	001506			MOV	#-12,ERCNT2
005064	012737	177773	001510			MOV	#-5,ERCNT3
005072	013705	001230				MOV	DRIVAD,R5
005076	012737	177152	001476			MOV	#-626,INDX2
005104	104415				4\$:	CON.RESET	
005106	104416					DRV.RESET	
005110	005037	001254			1\$:	CLR	RETRY2
005114	005037	001252			2\$:	CLR	RETRY1
005120	012777	026362	174334		3\$:	MOV	#OUTBUF,ARKBA
005126	010577	174332				MOV	R5,ARKDA
005132	012777	177764	174320			MOV	#-14,ARKWC
005140	012737	005104	001110			MOV	#4\$, \$LPERR
005146	012777	002005	174302			MOV	#2005,ARKCS
005154	104421					CON.RDY	
005156	032777	001000	174266		5\$:	BIT	#1000,ARKDS
005164	001420					BEG	6\$
005166	004737	016140				JSR	PC,GETINF
005172	104011					ERROR	11
005174	104415					CON.RESET	
005176	104416					DRV.RESET	
005200	005237	001252				INC	RETRY1
005204	022737	000002	001252			CMP	#2,RETRY1
005212	001342					BNE	3\$
005214	005237	001510				INC	ERCNT3
005220	001002					BNE	6\$
005222	000137	005532				JMP	16\$
005226	005777	174224			6\$:	TST	ARKCS
005232	100010					BPL	7\$
005234	004737	016062				JSR	PC,GT5RG
005240	104012					ERROR	12
005242	104415					CON.RESET	

** STORED.
 ** 5. IF INHIBIT TYPEOUT SWITCH IS NOT SET, THE FIRST WORDS OF
 ** THE 12 SECTORS (PSUEDO-HEADERS) ARE READ. IN A PREVIOUS
 ** 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 RECEIVED ARE TYPED OUT. THIS WOULD
 ** WRONG, HEADER WAS READ WRONG, ETC.
 ** 6. THE NEXT CYLINDER IN LINE IS READ. ORDER OF READING IS
 ** CYLO,SURO CYLO,SUR1 CYL312,SUR1

 :ST3: SCOPE
 :ALLOW ONLY 5 ERRORS (OF BAD HEADER
 :KIND FROM 5 CYLINDERS)
 :ALLOW ONLY 12 'ERR'S
 :ALLOW ONLY 5 ERRORS
 :SET DRIVE # CYL ADRES=0
 :313 CYLS (626 TRAKS) TO B READ
 :GO DO CONTROL RESET
 :GO DO DRIVE RESET
 :ALLOW 2 RETRIES IF HDRS READ WRONG
 :ALLOW 2 RETRIES FOR 'SINS'
 :RD HDRS INTO LOC STARTING AT THIS
 :FROM THIS DSK ADRES
 :12 HDRS TO BE READ
 :SET RETURN ADRES FOR LUPING ON ERROR
 :GO, RD FMT OF THIS CYLINDER
 :WAIT FOR CNTRL RDY TO SET
 : 'SIN' ERROR?
 :NO, BRANCH
 : 'SIN' OCCURED WHEN DOING RD FMT
 :FROM CYL SHOWN IN RKDA. IT
 :DO CNTRL RESET
 :GO, DO DRIVE RESET
 :ALLOW ONLY 2 RETRIES FOR THIS ERROR
 :IF TRIED 2 TIMES REPORT
 :ERROR, OTHERWISE GO BAK & RETRY
 :WAS TRIED TWICE, BUT 'SIN'.
 :ALLOW 5 ERRORS AT MOST
 : 'ERR' IN RKCS?
 :NO, BRANCH
 :GO, GET RKCS, ER,DS,DA,CYLNRD
 : 'ERR' SET WHILE DOING RD FMT
 :FROM CYL SHOWN IN RKDA
 :GO DO CNTRL RESET

Line	Address	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464
------	---------	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 26
DZRKLD.P11 31-AUG-76 15:35 T3 READ FORMAT OF THE DISK

ADDRESS	INSTR	OP1	OP2	OP3	OP4	COMMENT
1311						:TYPE: SEC #, EXPC P-HDR, RECVD P-HDR
1312						:TYPE OUT
1313	005402	104401				
1314	005404	001771				
1315	005406	012701	001266			:SEC #'S ARE STORED HERE
1316						
1317	005412	104401				:TYPE CR, LF
1318	005414	001213				
1319						
1320	005416	011102				
1321	005420	012703	026362			:PSUEDO-HEADERS WHICH WERE :READ ARE STORED HERE
1322						
1323						
1324	005424	005702				:IS THIS SEC # CORRESPONDING TO THE
1325	005426	001403				:ONE IN ERROR
1326	005430	005302				:R2 CONTAINS THE SEC #
1327	005432	005723				
1328	005434	000773				
1329						
1330	005436	011146				:GO TYPEOUT SEC # GIVING
1331	005440	104403				:MISCOMPARISON OF HEADERS
1332	005442	002				
1333	005443	000				:SUPRES LDG 0'S
1334						
1335	005444	104401				:TYPE 2 BLANKS
1336	005446	002105				
1337						
1338	005450	010546				:GO TYPE EXPCD PSUEDO HEADER
1339	005452	051116				
1340	005454	104402				
1341						
1342	005456	104401				:TYPE 2 BLNKS
1343	005460	002103				
1344						
1345	005462	011346				:GO TYPE PSUEDO-HEADER RECVD
1346	005464	104402				
1347						
1348	005466	005721				:TYPED OUT ALL SEC #'S IN ERROR.
1349	005470	021127	177777			:IF NOT GO BAK & TYPE NXT
1350	005474	001346				
1351						
1352	005476	104401	001733			:TYPE OUT PC
1353	005502	012746	005320			
1354	005506	104402				
1355	005510	104401	001213			:TYPE ROUTINE ENDS HERE
1356						
1357						
1358						:FIND OUT NXT TRAK TO B READ
1359						:FORMATTED
1360						
1361	005514	062705	000020			:SET ADRES FOR SUR D, NXT CYL IN LINE
1362	005520	005237	001476			:READ ALL 313 CYLINDERS (626 TRAKS)?
1363	005524	001404				:EXIT
1364	005526	000137	005110			:IF NOT, GO BAK & READ NXT
1365						
1366	005532	004737	016716			

E04

MAINDEC-11-DZAKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 27
 DZAKLD.F11 31-AUG-76 15:35 T3 READ FORMAT OF THE DISK

SEQ 0043

 : *TEST 4 SEEK PATTERNS: 0-312-0-311-..., USING IMPLIED SEEK

: ****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 INCORRECT, 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.

†ST4:

SCOPE
 CON.RESET
 DRV.RESET

:GO DO CONTROL RESET
 :GO DO DRIVE RESET

CLR R4

:FLAG, CLR IF DOING IMPLIED
 :SEEK IN FROM 0 TO 'INADR'
 :=1, IF GOING FROM 'INADR'
 :OUT TO CYL 0
 :313 SEEK PATTERNS

MOV #-313,INDX2
 MOV #-14,R0
 MOV R0,ERCNT1
 MOV R0,ERCNT2
 MOV R0,ERCNT3

:ALLOW ONLY 12 ERRORS
 :OF THESE KINDS

MOV #14500,INADR

: 'INADR' CONTAINS THE INNER
 :CYL TO WHICH IMPLIED SEEK WILL
 :BE DONE

005536 000004
 005540 104415
 005542 104416
 005544 005004

005546 012737 177465 001476
 005554 012700 177764
 005560 010037 001504
 005564 010037 001506
 005570 010037 001510
 005574 012737 014500 001260

F04

MAINDEC-11-DZAKL-D
DZAKLD.F11

31-AUG-76

MACY11 27(1006)

04-OCT-76

14:26 PAGE 28

T4

SEEK PATTERNS: 0-312-0-311-..., USING IMPLIED SEEK

SEQ 0044

1423	005602	005704		1\$:	TST	R4	:GOING IN OR OUT?
1424	005604	001005			BNE	2\$	:GOING OUT, BRANCH
1425	005606	013705	001260		MOV	INADR,R5	:SET CYL ADRES BITS FOR GOING IN
1426	005612	053705	001230		BIS	DRIVAD,R5	:FORM DISK ADRES FOR INNER
1427	005616	000402			BR	3\$	:CYLINDER
1428	005620	013705	001230	2\$:	MOV	DRIVAD,R5	:FORM DISK ADRES FOR OUTER
1429							:CYLINDER - 0
1430							:ALLOW 2 TRIES WHEN
1431	005624	012737	177776	001254	3\$:	MOV	#-2,RETRY2
1432	005632	012737	177776	001252	13\$:	MOV	#-2,RETRY1
1433	005640	012737	177777	001256	4\$:	MOV	#-1,RETRY3
1434	005646	000404			BR	5\$	
1435	005650	104415		6\$:	CON.RESET		
1436	005652	104416			DRV.RESET		
1437	005654	004737	006452		JSR	PC,SBR1	:REPOSITION HEADS TO PRE-ERROR CYL
1438							
1439	005660	012777	177777	173572	5\$:	MOV	#-1,ARKWC
1440	005666	010577	173572		MOV	R5,ARKDA	:READ 1 WORD
1441	005672	012777	026362	173562	MOV	#OUTBUF,ARKBA	:FROM THIS CYLINDER, SEC 0
1442	005700	012737	005650	001110	MOV	#6\$,SLPERR	:INTO THIS BUS ADRES
1443							:SET RETURN ADRES FOR LUPING
1444							:ON 'ERROR'
1445	005706	012777	000005	173542	MOV	#5,ARKCS	:GO, READ
1446							
1447	005714	104421			CON.RDY		:WAIT FOR CNTRL RDY
1448							
1449	005716	032777	001000	173526	BIT	#1000,ARKDS	:SIN?
1450	005724	001434			BEQ	8\$	:NO, BRANCH
1451							:YES, THERE WAS A SIN
1452	005726	004737	016322		JSR	PC,ERR1	:GO GET, CYLS BETW'N WHICH SK WAS TRIED
1453	005732	017737	173514	001170	MOV	ARKDS,\$REG3	
1454	005740	017737	173510	001166	MOV	ARKER,\$REG2	
1455	005746	104420	001602		TYPMSG	MSG1	
1456	005752	013737	001256	001172	MOV	RETRY3,\$REG4	:SAVE TRY # ON 'SIN'
1457	005760	062737	000002	001172	ADD	#2,\$REG4	
1458	005766	104014			ERROR	14	:AN IMPLIED SEEK WAS TRIED
1459							:FROM 'CYLA' TO 'CYLB' (INDICATED
1460							:IN EROR MESSAGE), 'SIN' OCCURRED.
1461							:2 TRIES ARE DONE BEFORE
1462							:ABORTING
1463	005770	005737	001256		TST	RETRY3	:DONE RETRIES
1464	005774	001403			BEQ	7\$	:YES, BRANCH
1465	005776	005237	001256		INC	RETRY3	:GO DO 2ND TRY
1466	006002	000722			BR	6\$	
1467							
1468	006004	005237	001504	7\$:	INC	ERCNT1	:ALLOW LESS THAN 12 ERORS OF THIS TYPE
1469	006010	001103			BNE	19\$	:IF MORE SKIP THIS TEST
1470	006012	000137	006540		JMP	EXT4	:EXIT THIS TEST
1471							
1472	006016	032777	010000	173430	8\$:	BIT	#10000,ARKER
1473	006024	001506			BEQ	20\$	:SKE?
1474	006026	004737	016322		JSR	PC,ERR1	:GO GET 2 CYL NOS. BETWEEN WHICH
1475	006032	017737	173416	001166	MOV	ARKER,\$REG2	:IMPLIED SEEK WAS DONE
1476	006040	017737	173406	001170	MOV	ARKDS,\$REG3	
1477	006046	013737	001252	001172	MOV	RETRY1,\$REG4	:GET TRY # ON 'SKE'
1478	006054	062737	000003	001172	ADD	#3,\$REG4	

G04

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 29
 DZRKLD.F11 31-AUG-76 15:35 T4 SEEK PATTERNS: 0-312-0-311-..., USING IMPLIED SEEK

SEQ 0045

1479	006062	104420	001610		TYPMSG MSG2	:GO PRINT 'SKE'
1480	006066	104014			ERROR 14	:IMPLIED SEEK WAS TRIED FROM
1481						: 'CYLA' TO 'CYLB' (INDICATED
1482						: IN EROR MESSAGE). 'SKE' OCCURRED.
1483						: 2 TRIES ARE DONE.
1484	006070	104415			CON.RESET	:DO CONTROL RESET
1485						
1486	006072	004737	006476	9\$:	JSR PC, SBR2	:GO READ 12 HEADERS FROM
1487						: THIS CYLINDER & COMPARE
1488						: THEM. NOTE RS CONTAINS THE
1489						: DISK ADRES THAT WILL BE USED.
1490						
1491	006076	012777	000015	173352	MOV #15, BRKCS	:GO DO DRIVE RESET
1492						: WHILE THE DRIVE IS DOING RESET
1493						: THE HDRS THAT WERE READ
1494						: ABOVE ARE CHECKED, PRINTED
1495						
1496	006104	005737	001500		TST INDX3	: WAS THERE A MISCOMPARISON
1497						: IN ANY HEADER?
1498	006110	001006			BNE 10\$	: IF INDX3>0, THERE WAS.
1499						: NO, THERE WASN'T. HDRS OK
1500	006112	005237	001252		INC RETRY1	: ONLY 2 TRIES FOR SKE
1501	006116	001414			BEQ 12\$	: BRANCH IF THIS WAS A 2ND TRY
1502	006120	104420	001657		TYPMSG ,MSG5	: TYPE OUT THAT HDRS WERE READ
1503						: CORRECTLY. THIS WAS TRY # 1
1504						
1505	006124	000405			BR 11\$	
1506						
1507						: HDRS WERE READ INCORRECT.
1508	006126	005237	001252	10\$:	INC RETRY1	: ALLOW 2 TRIES FOR SKE
1509	006132	001411			BEQ 14\$	: BRANCH, IF THIS WAS 2ND TRY
1510						
1511						: THERE WAS SKE ON DOING IMPLIED
1512	006134	104417	000015		MESSAGE ,15	: SEEK TO 'CYL B'. THEN HDRS WERE
1513						: READ FROM CYL B, WRONG HDRS
1514						: RECVD
1515						
1516	006140	104423		11\$:	RESDON	: WAIT FOR PREVIOUS DRIVE RESET
1517						: TO BE DONE
1518	006142	004737	006452		JSR PC, SBR1	: GO, REPOSITION HEADS
1519	006146	000634			BR 4\$	
1520						
1521						: 2ND TRY, SKE THIS TIME ALSO. BUT
1522						: READ HDRS CORRECTLY FROM
1523						: CYLINDER THAT GAVE SKE
1524						: NOTE THIS WAS THE 2ND TRY
1525	006150	104420	001657	12\$:	TYPMSG ,MSG5	: TYPE OUT THAT HDRS WERE
1526						: READ CORRECTLY.
1527						
1528	006154	000402			BR 16\$	
1529						
1530	006156	104417	000015	14\$:	MESSAGE ,15	: 2ND TRY, SKE THIS TIME ALSO.
1531						: READ HDRS FROM CYL THAT
1532						: GAVE SKE, THEY WERE INCORRECT.
1533						
1534	006162	104423		16\$:	RESDON	: WAIT FOR PREVIOUS DRIVE RESET

H04

MAINDEC-11-DZRLD-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 30
 DZRLD.P11 31-AUG-76 15:35 T4 SEEK PATTERNS: 0-312-0-311-..., USING IMPLIED SEEK

SEQ 0046

```

1535                                     ;TO BE DONE
1536
1537 006164 005237 001506                INC    ERCNT2      ;ALLOW ONLY LESS THAN 10 ERRORS OF
1538                                     ;THIS TYPE (SKE)
1539 006170 001002                        BNE     17$
1540 006172 000137 006540                JMP     EXT4      ;EXIT THIS TEST IF MORE
1541
1542 006176 005704                17$:    TST     R4          ;WENT LAST TIME IN OR OUT?
1543 006200 001007                        BNE     19$          ;OUT
1544                                     ;IN
1545 006202 005703                        TST     R3          ;WERE HDRS CORRECT?
1546 006204 001005                        BNE     19$          ;NO
1547                                     ;YES
1548
1549 006206 005204                18$:    INC     R4
1550 006210 004737 006452                JSR     PC,SBR1      ;GO POSITION HEADS BAK ON INNER
1551                                     ;CYL
1552 006214 000137 005602                JMP     1$          ;GO BAK & SEEK OUT NOW
1553
1554 006220 005237 001476                19$:    INC     INDX2      ;ALL SEEK PATTERNS DONE?
1555 006224 001547                        BEQ     TST5          ;EXIT
1556
1557 006226 162737 000040 001260          SUB     #40,INADR    ;SET ADDRESS FOR THE NXT
1558                                     ;INNER CYLINDER
1559 006234 005004                        CLR     R4          ;INDICATE THAT NOW SEEK IS GOING
1560                                     ;TO BE IN
1561 006236 000137 005602                JMP     1$          ;GO BAK & SEEK IN TO 'INADR'
1562
1563 006242                20$:           ;IF THERE WAS NO SIN OR SKE
1564                                     ;ENTER HERE
1565
1566 006242 012737 005624 001110          MOV     #3$,$LPERR    ;SET RETURN ADRES FOR LUPING
1567                                     ;ON ERROR
1568 006250 020537 026362                CMP     R5,OUTBUF    ;CORRECT PSUEDO-HEADER READ?
1569 006254 001471                        BEQ     24$          ;YES, BRANCH
1570 006256 013737 001254 001172          MOV     RETRY2,$REG4  ;GET TRY #
1571 006264 062737 000003 001172          ADD     #3,$REG4
1572 006272 004737 016322                JSR     PC,ERR1      ;GO GET CYL #'S BETW'N
1573                                     ;WHICH IMPLIED SEEK (READ)
1574                                     ;WAS DONE
1575 006276 010537 001166                MOV     R5,$REG2      ;GET EXPCTD PSUEDO-HDR
1576 006302 013737 026362 001170          MOV     OUTBUF,$REG3 ;GET PSUEDO-HDR RECVD
1577 006310 104016                        ERROR  16          ;IMPLIED SEEK FROM CYLA TO CYLB WAS DONE.
1578                                     ;READ PSEUDO-HEADER OF SEC 0,
1579                                     ;CYLB (IN EROR MESSAGE), BUT
1580                                     ;THE WRONG PSEUDO-HEADER WAS
1581                                     ;RECEIVED
1582 006312 005237 001254                INC     RETRY2
1583 006316 001402                        BEQ     21$
1584 006320 000137 005632                JMP     13$
1585
1586
1587 006324 004737 006476                21$:    JSR     PC,SBR2      ;GO READ HEADERS (12) FROM
1588                                     ;THIS CYLINDER, & CHECK THEM.
1589                                     ;IF MISCOMPARISON INDX3 WILL
1590                                     ;BE > 0.

```


MAINDEC-11-DZAKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 31
 DZAKLD.P11 31-AUG-76 15:35 T4 SEEK PATTERNS: 0-312-0-311-..., USING IMPLIED SEEK

SEQ 0047

1591	006330	005737	001500		TST	INDX3	
1592	006334	001003			BNE	22\$	
1593	006336	104420	001657		TYPMSG	,MSG5	:WRONG PSUEDO-HDR WAS READ
1594							:BUT WHEN HDRS WERE READ
1595							:FROM THE SAME CYLINDER, THEY
1596	006342	000402			BR	23\$	:WERE CORRECT
1597							
1598	006344	104417	000015	22\$:	MESSAGE	,15	:WRONG PSUEDO-HDR WAS READ
1599							:FROM 'CYLB' (IN ERROR MESSAGE).
1600							:THEN HEADERS WERE READ FROM THE
1601							:SAME CYLINDER. THEY WERE ALSO
1602							:WRONG.
1603	006350	010500		23\$:	MOV	R5,R0	:NOW READ THE PSUEDO-HEADER
1604	006352	005200			INC	R0	:FROM THE NEXT SECTOR (1)
1605	006354	010077	173104		MOV	R0,ARKDA	:SAME CYLINDER
1606	006360	012777	026362	173074	MOV	#OUTBUF,ARKBA	
1607	006366	012777	177777	173064	MOV	#-1,ARKWC	
1608	006374	012777	000005	173054	MOV	#5,ARKCS	
1609	006402	104421			CON.RDY		
1610	006404	010537	001162		MOV	R5,\$REGO	
1611	006410	004737	016360		JSR	PC,GCYL	:GO GET CYL # 8 STORE IT IN \$REGO
1612	006414	010037	001164		MOV	R0,\$REG1	:GET EXPCT PSUEDO-HDR FROM SEC 1
1613	006420	013737	026362	001166	MOV	OUTBUF,\$REG2	
1614	006426	104417	000017		MESSAGE	,17	:PSUEDO-HEADER FROM SEC 1, CYLB
1615							: (IN MESSAGE) WAS READ. THE EXPCTD
1616							: & RECVD DATA WORDS ARE REPORTED.
1617	006432	005237	001510		INC	ERCNT3	:ALLOW ONLY LESS THAN 10 ERRORS
1618							:OF THIS TYPE (WRONG PS-HDRS)
1619	006436	001440			BEG	EXT4	
1620							
1621	006440	005704		24\$:	TST	R4	:SEEKED IN OR OUT LAST TIME?
1622	006442	001266			BNE	19\$	:IF OUT, GO SEEK NXT INNER CYL
1623							:IF IN, GO SEEK BAK TO 0
1624	006444	005204			INC	R4	:INDICATE THAT SEEK OUT (0)
1625	006446	000137	005602		JMP	1\$	:WILL BE DONE NOW
1626							
1627							
1628							:THIS ROUTINE IS USED IN THIS TEST ONLY.
1629							:R4=0 INDICATES SEEK BEING DONE FROM
1630							:CYL 0 TO INNER CYL.
1631							:R4=1 INDICATES SEEK BEING DONE FROM
1632							:INNER CYL TO 0. THIS ROUTINE POSITIONS
1633							:THE HEADS ON 'INADR' CYL IF R4=1
1634							
1635	006452	005704		SBR1:	TST	R4	
1636	006454	001407			BEG	1\$	
1637	006456	013777	001260	173000	MOV	INADR,ARKDA	
1638	006464	012777	000011	172764	MOV	#11,ARKCS	
1639	006472	104422			TST.RWS		
1640	006474	000207		1\$:	RTS	PC	
1641							
1642							:THIS ROUTINE IS USED IN THIS TEST
1643							:ONLY. IT READS 12 HEADERS FROM CYLINDER
1644							:WHOSE ADRES IS IN R5. THEN IT CHECKS
1645							:IF THE EXPECTED HEADER IS RECEIVED.
1646							:IF IT IS NOT, INDX3 IS INCREMENTED INDICATING

J04

MAINDEC-11-DZAKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 32
 DZAKLD.P11 31-AUG-76 15:35 T4 SEEK PATTERNS: 0-312-0-311-..., USING IMPLIED SEEK

SEQ 0048

; THE ERROR

```

1647
1648
1649 006476 012700 177764 SBR2: MOV #-14,R0
1650 006502 012701 026362 MOV #OUTBUF,R1
1651 006506 010077 172746 MOV R0,ARKWC ;READ 12 HDRS
1652 006512 010177 172744 MOV R1,ARKBA ;INTO THIS ADRES
1653 006516 010577 172742 MOV R5,ARKDA ;FROM THIS CYLINDER
1654 006522 012777 002005 172726 MOV #2005,ARKCS ;RD FMT, GO
1655 006530 104421 CON.RDY
1656
1657 006532 004737 007204 JSR PC,CHKHDRS ;GO CHECK IF CORRECT HEADERS WERE READ
1658
1659 006536 000207 RTS PC ;EXIT
1660
1661 006540 004737 016716 EXT4: JSR PC,ABRT
1662
1663 ;*****
1664 ;*TEST 5 PERFORM CONVERGING-DIVERGING (IMPLIED) SEEKS
1665 ;*THIS TEST PERFORMS A CONVERGING-DIVERGING SEEK PATTERN
1666 ;*USING IMPLIED SEEK (READ FORMAT). THE SEEK SEQUENCE IS:
1667 ;*0-312-1-311-2-310-3-307-----310-2-311-1-312
1668 ;*ALL READ FORMATS ARE DONE FROM SURFACE 0.
1669 ;*THE CYLINDER ADDRESSES, BETWEEN WHICH THE IMPLIED SEEK IS
1670 ;*PERFORMED, ARE CONTAINED IN 'OUTADR' & 'INADR'. IF 'SIN' OCCURS
1671 ;*AN ERROR IS REPORTED AND A RETRY IS DONE. ON READING INCORRECT
1672 ;*HEADERS AN ERROR IS REPORTED AND A RETRY IS DONE. NOTE THAT IF
1673 ;*ALL THE 12 HEADERS ARE INCORRECT, IT COULD MEAN THAT THE HEADS
1674 ;*COULD NOT POSITION CORRECTLY. THIS WOULD BE CONFIRMED IF IN
1675 ;*PREVIOUS TESTS BAD HEADERS WERE NOT RECIEVED FROM THE SAME
1676 ;*CYLINDER. IF THAT CYLINDER GAVE BAD HEADERS IN ALL THE PREVIOUS
1677 ;*TESTS THE PROBLEM COULD BE DIFFERENT.
1678 ;*MAXIMUM 12 ERRORS OF ANY KIND ARE ALLOWED.
1679 ;*IF MORE THAN 12 ERRORS OCCUR THE TEST IS ABORTED.
1680 ;*****
1681 006544 000004 TSTS: SCOPE
1682 006546 104415 CON.RESET ;GO,DO CONTROL RESET
1683 006550 104416 DRV.RESET ;GO,DO DRIVE RESET
1684
1685 006552 005004 CLR R4 ;(R4)=0 SEEKING FROM 'OUTADR' TO 'INADR'
1686 ;(R4)=1 SEEKING FROM 'INADR' TO 'OUTADR'
1687
1688 006554 012737 177466 001476 MOV #-312,INDX2 ;SET COUNT FOR DOING 312 TIMES
1689 006562 012700 177764 MOV #-14,R0
1690 006566 010037 001504 MOV R0,ERCNT1 ;ALLOW ONLY 12 ERRORS
1691 006572 010037 001506 MOV R0,ERCNT2
1692
1693 006576 005037 001262 CLR OUTADR ;INITIALIZE 'OUTADR' TO 0
1694 006602 012737 014500 001260 MOV #14500,INADR ;INITIALIZE 'INADR' TO 312
1695
1696 006610 005704 1$: TST R4 ;GOING IN OR OUT?
1697 006612 001005 BNE Z$ ;GOING OUT,BRANCH
1698 006614 013705 001260 MOV INADR,R5 ;SET CYL ADRES BITS FOR GOING IN
1699 006620 053705 001230 BIS DRIVAD,R5 ;FORM DISK ADRES FOR INNER CYLINDER
1700 006624 000404 BR Z$
1701
1702 006626 013705 001262 2$: MOV OUTADR,R5 ;SET CYL ADRES BITS FOR GOING OUT

```


K04

 MAINDEC-11-DZRLD-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 33
 DZRLD.P11 31-AUG-76 15:35 TS PERFORM CONVERGING-DIVERGING (IMPLIED) SEEKS

SEQ 0049

1703	006632	053705	001230		BIS	DRIVAD,R5	;FORM DISK ADRES FOR GOING OUT
1704							
1705	006636	005037	001254		3\$: CLR	RETRY2	;ALLOW 2 RETRIES
1706	006642	012737	177777	001252	4\$: MOV	#-1,RETRY1	;WHEN ERRORS OCCUR
1707	006650	000404			BR	7\$	
1708	006652	104415			5\$: CON.RESET		
1709	006654	104416			DRV.RESET		
1710	006656	004737	007150		JSR	PC,SBR3	;GO REPOSITION HEADS
1711							
1712	006662	012777	177764	172570	7\$: MOV	#-14,ARKWC	;READ ALL HDRS FROM THIS CYLINDER
1713	006670	010577	172570		MOV	R5,ARKDA	;FROM THIS CYL, SEC 0
1714	006674	012777	026362	172560	MOV	#OUTBUF,ARKBA	;INTO THIS BUS ADRES
1715	006702	012737	006652	001110	MOV	#5\$,SLPERR	;SET RETURN ADRES FOR LOOPING ON ERROR
1716							
1717	006710	012777	002005	172540	MOV	#2005,ARKCS	;READ FORMAT,GO
1718							
1719	006716	104421			CON.RDY		;WAIT FOR CONTRL RDY
1720							
1721	006720	032777	001000	172524	BIT	#1000,ARKDS	;SIN?
1722	006726	001443			BEQ	8\$	;NO, BRANCH
1723	006730	017737	172520	001166	MOV	ARKER,\$REG2	;SAVE RKER
1724	006736	017737	172510	001170	MOV	ARKDS,\$REG3	;SAVE RKDS
1725	006744	013737	001252	001172	MOV	RETRY1,\$REG4	;GET RETRY #
1726	006752	062737	000002	001172	ADD	#2,\$REG4	
1727	006760	004737	016244		JSR	PC,ERR2	;GET CYL #'S BELOW 'N WHICH
1728							;SEEK WAS TRIED
1729	006764	104420	001602		TYPMSG	MSG1	;TYPE 'SIN'
1730	006770	104014			ERROR	14	
1731							; 'SIN' OCCURRED ON DOING IMPLIED
1732							;SEEK FROM 'CYLA' TO 'CYLB' (IN
1733							;EROR MESSAGE).
1734	006772	005737	001252		TST	RETRY1	;DONE 2 TRIES?
1735	006776	001403			BEQ	6\$	;YES, BRANCH
1736	007000	005237	001252		INC	RETRY1	;NO, RETRY
1737	007004	000722			BR	5\$	
1738	007006	104415			6\$: CON.RESET		
1739	007010	104416			DRV.RESET		
1740							
1741							
1742	007012	005237	001504		INC	ERCNT1	;ALLOW LESS THAN 12 ERORS OF THE
1743	007016	001527			BEQ	EXT5	;ABOVE KIND
1744							;IF MORE SKIP THIS TEST
1745							;SIN OCCURED WHEN GOING TO CYL (IN
1746							;R5). A DRIVE RESET HAS BEEN DONE,
1747							;NOW TRY POSITIONING HEADS ON
1748							;THAT CYL.
1749	007020	010577	172440		MOV	R5,ARKDA	;SET CYL ADRES
1750	007024	012777	000011	172424	MOV	#11,ARKCS	;SEEK,GO
1751	007032	104422			TST.RWS		
1752	007034	000424			BR	11\$	
1753							;IF NO SIN, ENTER HERE
1754	007036	004737	007204		8\$: JSR	PC,CHKHDRS	;GO CHECK IF CORRECT HEADERS WERE READ
1755							
1756	007042	012737	006642	001110	MOV	#4\$,SLPERR	;SET LUP ADDRESS
1757	007050	005737	001500		TST	INDX3	;WAS THERE A BAD HDR?
1758	007054	001414			BEQ	11\$	;NO, BRANCH

L04

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 34
 DZRKLD.P11 31-AUG-76 15:35 75 PERFORM CONVERGING-DIVERGING (IMPLIED) SEEKS

SEQ 0050

```

1759 007056 104020          ERROR 20          ;WRONG HEADERS WERE READ FROM
1760                                     ;'SEC #'S, ON DOING AN IMPLIED
1761                                     ;SEEK (RDFMT) FROM 'CYLA' TO 'CYLB'
1762 007060 005237 001254    INC  RETRY2       ;ALLOW 2 TRIES
1763 007064 022737 000002 001254    CMP  #2,RETRY2
1764 007072 001263          BNE  4$           ;GO TRY 2ND TIME
1765 007074 005237 001506    INC  ERCNT2       ;ALLOW ONLY 12 ERRORS
1766 007100 001002          BNE  11$
1767 007102 000137 007276    JMP  EXT5        ;IF MORE, EXIT THIS TEST
1768
1769 007106 005704          11$: TST  R4        ;GOING WHICH WAY?
1770 007110 001006          BNE  12$         ;'INADR' TO 'OUTADR', BRANCH
1771                                     ;'OUTADR' TO 'INADR'
1772 007112 005204          INC  R4           ;INDICATE THAT NXT TIME GOING
1773                                     ;FROM 'INADR' TO 'OUTADR'
1774 007114 062737 000040 001262    ADD  #40,OUTADR ;INCREMENT CYLINDER ADRES
1775 007122 000137 006610    JMP  1$         ;GO BAK & DO IMPLIED SEEK
1776                                     ;FROM 'INADR' TO 'OUTADR'
1777
1778 007126 005004          12$: CLR  R4        ;INDICATE THAT NXT TIME GOING
1779                                     ;FROM 'OUTADR' TO 'INADR'
1780 007130 162737 000040 001260    SUB  #40,INADR ;DECREMENT CYLINDER ADRES
1781 007136 005237 001476    INC  INDX2       ;DONE ALL 312 FORWARD-BACKWARD
1782                                     ;SEEK PATTERNS
1783 007142 001457          BEQ  TST6        ;;IF YES, EXIT
1784
1785 007144 000137 006610    JMP  1$         ;IF NOT, GO BAK & DO IMPLIED
1786                                     ;SEEK FROM 'OUTADR' TO 'INADR'
1787
1788                                     ;THIS SUBROUTINE IS ENTERED AFTER A 'SIN' ERROR OCCURED ON GOING FROM
1789                                     ;'CYLA' TO 'CYLB' AS INDICATED IN THE ERROR MESSAGE. BEFORE RETRYING THE
1790                                     ;HEADS HAVE TO BE POSITIONED BACK TO 'CYLA'. NOTE THAT A DRIVE RESET
1791                                     ;WAS DONE TO CLEAR SIN.
1792                                     ;R4=0, INDICATES SEEK IS BEING DONE FROM 'OUTADR' CYLINDER TO 'INADR' CYLINDER.
1793                                     ;R4=1, INDICATES THAT SEEK IS BEING DONE FROM 'INADR' TO 'OUTADR'.
1794 007150 005704          SBR3: TST  R4        ;GOING WHICH WAY?
1795 007152 001404          BEQ  1$           ;IF FROM 'OUTADR' TO 'INADR', BRANCH.
1796 007154 013777 001260 172302    MOV  INADR,ARKDA
1797 007162 000403          BR   2$
1798 007164 013777 001262 172272 1$: MOV  OUTADR,ARKDA
1799 007172 012777 000011 172256 2$: MOV  #11,ARKCS
1800 007200 104422          TST.RWS
1801 007202 000207          RTS  PC
1802
1803                                     ;CHKHDRS
1804                                     ;THIS ROUTINE CHECKS THAT THE HEADERS READ PREVIOUSLY WERE CORRECT.
1805                                     ;IF NOT, THE BAD HEADERS AND THE SECTOR #'S FROM WHICH THEY WERE READ
1806                                     ;ARE STORED.
1807                                     ;ON ENTRY:
1808                                     ;R5 CONTAINS DISK ADDRESS FROM WHERE HEADERS WERE READ.
1809                                     ;OUTBUF - 12 HEADERS READ PREVIOUSLY ARE STORED STARTING 'OUTBUF'.
1810                                     ;ON EXIT:
1811                                     ;INDX3=0, IF THE HEADERS WERE CORRECT
1812                                     ;INDX3=1, IF THE HEADERS WERE INCORRECT
1813                                     ;BUFR - SECTOR #'S GIVING BAD HEADERS ARE STORED STARTING AT 'BUFR'.
1814

```


M04

MAINDEC-11-DZAKL-D
DZAKLD.P11MACY11 27(1006)
31-AUG-76 15:3504-OCT-76 14:26 PAGE 35
TS PERFORM CONVERGING-DIVERGING (IMPLIED) SEEKS

SEQ 0051

;BUFR1 - BAD HEADERS FOR THE ABOVE SECTORS ARE STORED STARTING 'BUFR1'.
;THE BAD SECTOR TABLE IS TERMINATED BY A '177777' WORD.

```

CHKHDRS: CLR      RD      ;INITIALIZE FOR 14 HDRS
           MOV      #OUTBUF,R1 ;INITIALIZE PTR TO HDRS RECVD
           MOV      #BUFR,R2  ;INITIALIZE PTR TO SECTOR TABLE
           MOV      #BUFR1,R3 ;INITIALIZE PTR TO BAD HDR TABLE
           MOV      R5,$GDADR
           BIC      #160037,$GDADR ;GET EXPCTD HEADER
           CLR      INDX3      ;CLR FLG INDICATING BAD HDRS

9$:        CMP      $GDADR,(R1) ;HEADER OK?
           BEQ      10$        ;YES, BRANCH
           MOV      (R1),(R3)+ ;SAVE BAD HDR
           MOV      R0,(R2)+   ;SAVE BAD SECTR #

           MOV      #177777,(R2) ;PUT TERMINATR ON SECTR TABLE
           INC      INDX3      ;SET FLG INDICATING BAD HDR
           TST      (R1)+      ;INCRMT PTR TO NXT HDR
           INC      R0         ;ALL HDRS CHKD?
           CMP      #14,R0
           BNE      9$         ;IF NOT, LUP BAK
           RTS      PC

EXT5:      JSR      PC,ABRT

```

;TEST 6 WRITE PATTERNS ON THE DISK

;IN THIS TEST DIFFERENT PATTERNS ARE WRITTEN ON THE ENTIRE DISK. 768
;WORDS (3 SECTORS) ARE WRITTEN AT A TIME. TWO 768 WORDS LONG
;BUFFERS HAVE BEEN USED IN THIS TEST. WHILE ONE BUFFER IS
;USED TO WRITE ON THE DISK, THE OTHER BUFFER GETS FILLED UP WITH
;PATTERNS TO BE USED NEXT! THIS OVERLAPPING IS DONE TO SAVE TIME.

;THREE PATTERN-GENERATOR SUB-ROUTINES ARE USED:
;1. PTGEN0 2. PTGEN1 3. PTGEN2 4. PTGEN3
;THESE THREE ROUTINES ARE CALLED IN A CYCLIC ORDER:
;PTGEN0-PTGEN1-PTGEN2-PTGEN3-PTGEN0-PTGEN1.....
;THE FOLLOWING ORDER IS MAINTAINED IN WRITING BLOCKS (EACH
;3 SECTORS LONG) USING ONE OF THE FOUR PATTERN GENERATORS:

*CYL	SECTORS	ROUTINE	*CYL	SECTORS	ROUTINE	*CYL	SECTORS	ROUTINE	*CYL	SECTORS	ROUTINE			
* 0	0-2	PTGEN0	0	0	6-10	PTGEN1	0	0	3-5	PTGEN2	0	0	11-13	PTGEN3
* 0	1	PTGEN0	0	1	6-10	PTGEN1	0	1	3-5	PTGEN2	0	1	11-13	PTGEN3
* 1	0-2	PTGEN0	1	0	6-10	PTGEN1	1	0	3-5	PTGEN2	1	0	11-13	PTGEN3
* 1	1	PTGEN0	1	1	6-10	PTGEN1	1	1	3-5	PTGEN2	1	1	11-13	PTGEN3
* 2	0-2	PTGEN0	2	0	6-10	PTGEN1	2	0	3-5	PTGEN2	2	0	11-13	PTGEN3

;ETC, ETC.....
;THE ABOVE SHOWN STAGGERING (OF SECTORS TO BE WRITTEN) IS DONE TO
;SAVE ONE REV (40MS) EVERY TIME A BLOCK (3 SECTORS) IS WRITTEN.
;IF YOU WANT TO USE BUFFERS STARTING AT SOME OTHER MEMORY ADDRESS
;MAKE THE FOLLOWING CHANGES:
;CHANGE 'PBUF0' TO STARTING ADDRESS OF THE FIRST 768 WORDS LONG BUFFER.
;CHANGE 'PBUF1' TO STARTING ADDRESS OF SECOND 768 WORDS LONG BUFFER.

NO4

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 36
 DZRKLD.F11 31-AUG-76 15:35 T6 WRITE PATTERNS ON THE DISK

SEQ 0052

```

1871
1872
1873
1874
1875
1876
1877
1878
1879 007302 000004
1880 007304 012737 177764 001504
1881 007312 012737 177764 001506
1882 007320 012737 177152 001474
1883 007326 005037 001410
1884 007332 005037 001412
1885
1886
1887
1888 007336 013737 001230 001432
1889 007344 012737 001414 001424
1890
1891 007352 004737 007770
1892 007356 017737 172042 001430
1893 007364 004777 172040
1894
1895 007370 005005
1896 007372 005737 001410
1897 007376 100407
1898 007400 013737 001406 001434
1899
1900 007406 052737 000200 001412
1901
1902
1903 007414 000406
1904
1905 007416 013737 001404 001434
1906
1907
1908 007424 052737 000200 001410
1909
1910 007432 012737 007440 001110
1911 007440 013777 001432 172016
1912 007446 013777 001434 172006
1913 007454 012777 176400 171776
1914 007462 012777 000003 171766
1915
1916 007470 013737 001424 001426
1917 007476 023727 001424 001422
1918 007504 001004
1919 007506 012737 001414 001426
1920 007514 000403
1921 007516 062737 000002 001426
1922 007524 004737 007770
1923 007530 017737 171672 001430
1924 007536 004777 171666
1925
1926 007542 104421
  
```

```

; *IF YOU WANT TO WRITE YOUR OWN PATTERNS USING PATTERN GENERATOR 'PTGENO'
; *CHANGE 'PTRNO1' AND 'PTRNO2' TO THE PATTERNS YOU WANT.

; *TO WRITE THE SAME TWO (OR ONE) PATTERNS ON THE ENTIRE DISK CHANGE
; *LOCATION 'PAT1' TO 'PTGENO' THE STARTING ADDRESS OF PAT-GENERATOR 0
; *LOCATION 'PAT2' TO 'PTGENO' THE STARTING ADDRESS OF PAT-GENERATOR 0
; *LOCATION 'PAT3' TO 'PTGENO' THE STARTING ADDRESS OF PAT-GENERATOR 0
; *****
TST6: SCOPE
      MOV     #-14,ERCNT1
      MOV     #-14,ERCNT2
      MOV     #-626,INDX1
      CLR     BUFLG0
      CLR     BUFLG1
      ; SET COUNT FOR 313X2 TRACKS
      ; CLR FLAG FOR BUFR 0
      ; CLR FLAG FOR BUFR 1
      ; BIT 7 OF ABOVE FLAGS WHEN SET
      ; INDICATES, THAT BUFR TO BE USED
      ; FOR WRITING ON DSK
      MOV     DRIVAD,DSKADR
      MOV     #PATO,PRSPAT
      ; GET DRIVE #, DISK ADRES
      ; INITIALIZE PTR TO THE FIRST
      ; PATRN GENERATOR

      JSR     PC,GETBUF
      MOV     @PRSPAT,PGSUBR
      JSR     PC,@PGSUBR
      ; GO GENERATE PATRNS FOR
      ; 3 SECTOR (400X3)
      ; INITIALIZE COUNT FOR 4 BLOCKS
      ; FIND OUT WHICH BUFR TO USE
      ; FOR WRITING ON DSK
      CLR     R5
      TST     BUFLG0
      BMI     3$
      MOV     PBUF1,BUSADR
      ; USE 'IOBUF1' FOR TRANSFER
      ; OR THE ONE INDICATED BY THE USER
      BIS     #BIT7,BUFLG1
      ; SET FLAG TO INDICATE THAT
      ; WRITING ON DSK WILL B DONE FROM
      ; THIS BUFR (BUF 1)

      BR      13$

      MOV     PBUFO,BUSADR
      ; USE 'IOBUFO' FOR TRANSFER
      ; OR THE ONE INDICATED BY THE USER

      BIS     #BIT7,BUFLG0
      ; INDICATE THAT 'IOBUFO' WILL
      ; B USED FOR WRITING ON DISK

      MOV     #4$,SLPERR
      MOV     DSKADR,@RKDA
      MOV     BUSADR,@RKBA
      MOV     #-1400,@RKWC
      MOV     #3,@RKCS
      ; SET RKDA

      ; WRITE THE 4 SECTORS ON
      ; DISK
      MOV     PRSPAT,NXTPAT
      CMP     PRSPAT,#PAT3
      BNE     5$
      ; WHILE THE PATRNS R BEING WRITTEN
      ; GO GENERATE THE NXT PATRNS
      ; TO B WRITTEN
      MOV     #PATO,NXTPAT
      BR      6$
      ; KEEP GENERATING PATRNS IN THIS
      ; WAY "PATO"--"PAT1"--"PAT2"--"PAT3"--"PATO"--

      ADD     #2,NXTPAT
      JSR     PC,GETBUF
      MOV     @NXTPAT,PGSUBR
      JSR     PC,@PGSUBR
      ; GO GENERATE THESE PATRNS.
      ; (3 X 400) WORDS
  
```

CON.RDY

B05

MAINDEC-11-DZRLD-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 37
 DZRLD.P11 31-AUG-76 15:35 T6 WRITE PATTERNS ON THE DISK

SEQ 0053

```

1927 007544 032777 140000 171704 BIT #140000,DRKCS :ANY ERROR?
1928 007552 001411 BEQ 12$ :GET RKCS,ER,DS,DA
1929 007554 004737 JSR PC,GT4RG
1930 007560 104021 ERROR 21 :ERROR ON DOING WRITE
1931 007562 104415 CON.RESET :CLEAR IT
1932 007564 005237 001504 INC ERCNT1 :ALLOW 12 ERRORS AT MOST
1933 007570 001002 BNE 12$
1934 007572 000137 010356 JMP EXT6 :IF MORE, EXIT
1935 007576 032777 001000 171646 12$: BIT #BIT9,DRKDS :SIN, ON DOING WRITE?
1936 007604 001412 BEQ 7$
1937 007606 004737 016106 JSR PC,GT4RG
1938 007612 104022 ERROR 22 :SIN ERROR ON DOING WRITE
1939 007614 104415 CON.RESET
1940 007616 104416 DRV.RESET
1941 007620 005237 001506 INC ERCNT2 :ALLOW 12 ERRORS AT MOST
1942 007624 001002 BNE 7$
1943 007626 000137 010356 JMP EXT6
1944 :FIGURE OUT WHICH BUFFER IS
1945 :AVAILABLE FOR USE
1946 007632 105737 001410 7$: TSTB BUFLGO :WAS PREVIOUS DSK-WRITE DONE
1947 007636 100003 BPL 8$ :USING BUFR 0?
1948 007640 005037 001410 CLR BUFLGO :YES, CLR FLAG INDICATING IT'S
1949 :AVAILABLE NOW
1950 007644 000402 BR 9$
1951 007646 005037 001412 8$: CLR BUFLG1 :CLR FLAG INDICATING BUFR1
1952 :IS AVAILABLE NOW
1953 007652 013737 001426 001424 9$: MOV NXTPAT,PRSPAT :PRSPAT'S PATRNS WILL BE USED
1954 :ON NEXT WRITE
1955 007660 010500 MOV R5,R0 :FORM SEC # TO BE USED NXT TIME
1956 007662 116000 010352 MOVB SECPTR(R0),R0 :GET SEC #
1957 007666 042737 000017 001432 10$: BIC #17,DSKADR :MASK SECTOR BITS FROM DSK-ADRES
1958 007674 050037 001432 BIS R0,DSKADR :FORM NXT DSK-ADRES
1959 007700 005205 INC R5 :DONE WITH 12 SECTRS (3 BLOCKS)?
1960 007702 022705 000004 CMP #4,R5
1961 007706 001231 BNE 2$ :ON THIS SURFACE? IF NOT, GO
1962 :DO NXT 4 SECTRS
1963 007710 032737 000001 001474 BIT #BIT0,INDX1 :WHICH SURFACE WAS DONE, 0 OR 1?
1964 007716 001415 BEQ 11$ :IF 0, GO DO SURFACE 1
1965 007720 005237 001474 INC INDX1 :COUNT # OF TRACKS
1966 007724 001002 BNE +6
1967 007726 000137 010362 JMP TST7 :EXIT IF DONE
1968 007732 042737 000020 001432 BIC #20,DSKADR :GO TO SUR 0, NEXT CYLINDER
1969 007740 062737 000040 001432 ADD #40,DSKADR
1970 007746 000137 007370 JMP 1$ :GO BACK AND DO WRITE
1971 007752 005237 001474 11$: INC INDX1 :COUNT # OF TRACKS
1972 007756 052737 000020 001432 BIS #20,DSKADR :SET BIT FOR SUR 1
1973 007764 000137 007370 JMP 1$ :GO, WRITE PATRNS ON SURFACE 1
1974 :*GET BUF#
1975 :*THIS ROUTINE FINDS OUT WHICH BUFFER (IOBUF0, IOBUF1) OR ONE OF
1976 :*THE TWO BUFFERS SELECTED BY THE USER) TO USE
1977 :*FOR GENERATING PATTERNS. THEN IT SETS UP A FLAG INDICATING THAT
1978 :*BUFFER HAS TO BE FILLED UP. THE STARTING ADDRESS OF THE
1979 :*BUFFER IS STORED IN 'BUF #'.
1980
1981 007770 005737 001410 GETBUF: TST BUFLGO :BUFR 0 AVAILABLE FOR USE?
1982 007774 100007 BPL 1$ :YES, BRANCH

```


C05

MAINDEC-11-DZRL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 38
 DZRLD.F11 31-AUG-76 15:35 T6 WRITE PATTERNS ON THE DISK

SEQ 0054

```

1983 007776 052737 100000 001412 BIS #BIT15,BUFLG1 ;NO, USE BUFR 1. INDICATE 50.
1984 010004 013737 001406 001446 MOV PBUF1,BUFNO ;SAVE STARTING ADRES OF BUFR1
1985 010012 000207 RTS PC
1986 010014 052737 100000 001410 1$: BIS #BIT15,BUFLG0 ;INDICATED, USING BUF 0
1987 010022 013737 001404 001446 MOV PBUF0,BUFNO ;SAVE STARTING ADRES OF BUFR 0
1988 010030 000207 RTS PC ;RETURN
1989
1990 ;*PTGEN0
1991 ;*THIS ROUTINE GENERATES A 768 (3X256) WORDS LONG
1992 ;*BUFFER CONTAINING THE FOLLOWING PATTERNS
1993 ;*FIRST BLOCK OF 256 WORDS: 125252
1994 ;*SECOND BLOCK OF 256 WORDS: 052525 (COMPLEMENT OF ABOVE)
1995 ;*THIRD BLOCK OF 256 WORDS: 010421
1996 ;*YOU CAN USE ANY OTHER PATTERN/S (& ITS COMPLEMENT)
1997 ;*MAKING THE CHANGES IN THE 2 LOCATIONS SHOWN BELOW.
1998
1999 PTGEN0: MOV BUFNO,R0 ;GET STARTING ADRES OF BUFR
2000 MOV PTRNO1,R1 ;GET PATRN TO BE GENERATED
2001 MOV #-400,R2 ;IN THE FIRST 400 WORD BLOCK
2002
2003 1$: MOV R1,(R0)+ ;GENERATE THE FIRST BLOCK
2004 INC R2 ;WITH 'PAT01' PATRN
2005 BNE 1$ ;ALL DONE?
2006
2007 MOV #-400,R2
2008 COM R1 ;COMPLEMENT 'PAT01' PATRN
2009 2$: MOV R1,(R0)+ ;GENERATE 2ND BLOCK WITH
2010 INC R2 ;'PAT01'S COMPLEMENT PATRN
2011 BNE 2$ ;ALL DONE?
2012 MOV #-400,R2
2013 MOV PTRNO2,R1 ;GET PATRN TO BE GENERATED
2014 ;FOR 3RD BLOCK
2015 3$: MOV R1,(R0)+ ;GENERATE 3RD BLOCK USING
2016 INC R2 ;'PAT02' PATRN
2017 BNE 3$ ;ALL DONE?
2018
2019 RTS PC ;RETURN
2020
2021 PTRNO1: 125252 ;CHANGE THESE LOCATIONS IF
2022 PTRNO2: 010421 ;YOU WANT ANY OTHER PATTERNS
2023
2024 ;*PTGEN1
2025 ;*THIS ROUTINE GENERATES A 768 (3X256) WORDS
2026 ;*LONG BUFFER CONTAINING THE FOLLOWING PATTERNS:
2027
2028 ;*FIRST BLOCK-256 WORDS 000001 FILL 1'S
2029 ;* 000003
2030 ;* 177777
2031 ;*
2032 ;*SECOND BLOCK 177776 FILL 0'S
2033 ;* 177774
2034 ;* 000000
2035 ;*
2036 ;*THIRD BLOCK 000001 FLOAT A 1

```


005

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 39
DZRKLD.P11 31-AUG-76 15:35

T6 WRITE PATTERNS ON THE DISK

SEQ 0055

2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094

010114	012703	000001
010120	012704	177776
010124	013700	001446
010130	012702	177760
010134	010301	
010136	010120	
010140	000261	
010142	006101	
010144	103374	
010146	005202	
010150	001371	
010152	012702	177760
010156	010401	
010160	010120	
010162	000241	
010164	006101	
010166	103774	
010170	005202	
010172	001371	
010174	012702	177760
010200	010301	
010202	010120	
010204	006301	
010206	103375	
010210	005202	
010212	001372	
010214	000207	

```

**                                000020
**                                100000
**
;* 'BUFNO' CONTAINS THE STARTING ADDRESS OF THE
;* BUFFER.

PTGEN1: MOV    #1,R3              ; INITIALIZE PATRNS
        MOV    #177776,R4
        MOV    BUFNO,R0          ; GET STARTING ADRES OF BUFR
        MOV    #-20,R2          ; SET COUNT
        MOV    R3,R1            ; INITIALIZE PATRN
        MOV    R1,(R0)+         ; GENERATE THE FIRST
        SEC                                ; BLOCK USING "FILL 1'S"
        ROL    R1
        BCC    2$
        INC    R2
        BNE    1$
        ; ALL DONE?

        MOV    #-20,R2
        MOV    R4,R1            ; INITIALIZE PATRN
        MOV    R1,(R0)+         ; GENERATE 2ND BLOCK
        CLC                                ; USING "FILL 0'S"
        ROL    R1
        BCS    4$
        INC    R2
        BNE    3$
        ; DONE?

        MOV    #-20,R2
        MOV    R3,R1            ; SET COUNT
        MOV    R1,(R0)+         ; INITIALIZE PATRN
        ASL    R1                ; GENERATE THE 3RD BLOCK
        BCC    6$
        INC    R2
        BNE    5$
        RTS    PC               ; RETURN
        ; DONE?

;* 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
;*                                :

```

E05

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 40
DZRKLD.P11 31-AUG-76 15:35

T6 WRITE PATTERNS ON THE DISK

SEQ 0056

```

2095      ;*                               177777
2096
2097      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2098
2099      PTGEN2: CLR      R1                ;INITIALIZE PATRN
2100
2101      010216 005001
2102      010220 013700 001446
2103      010224 010120
2104      010226 105201
2105      010230 001375
2106
2107      010232 005001
2108      010234 010120
2109      010236 062701 000400
2110      010242 103374
2111      010244 005001
2112      010246 010120
2113      010250 062701 000401
2114      010254 103374
2115      010256 000207
2116
2117      ;PTGEN3
2118      ;*THIS ROUTINE GENERATES A 768 (3X256) WORDS LONG
2119      ;*BUFFER CONTAINING THE FOLLOWING PATTERNS:
2120
2121      ;*FIRST BLOCK OF 256 WORDS:      167356 (COMPLEMENT OF 010421)
2122
2123      ;*SECOND BLOCK      177776 FLOAT A 0
2124      ;*      177775
2125      ;*      077777
2126
2127      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2128      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2129      ;*      177400
2130
2131      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2132
2133      PTGEN3: MOV      BUFNO,R0
2134      010260 013700 001446
2135      010264 012702 177400
2136      010270 013701 010112
2137      010274 005101
2138      010276 010120
2139      010300 005202
2140      010302 001375
2141
2142      ;*      177760
2143      010304 012702 177776
2144      010310 000261
2145      010312 012701
2146      010316 010120
2147      010320 006101
2148      010322 103775
2149      010324 005202
2150
2151      ;*      177777
2152
2153      ;*      077777
2154
2155      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2156      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2157      ;*      177400
2158
2159      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2160
2161      PTGEN3: MOV      BUFNO,R0
2162      MOV      #-400,R2
2163      MOV      PTRNO2,R1
2164      COM      R1
2165      ;*      177776
2166      ;*      177775
2167      ;*      077777
2168
2169      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2170      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2171      ;*      177400
2172
2173      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2174
2175      PTGEN3: MOV      BUFNO,R0
2176      MOV      #-400,R2
2177      MOV      PTRNO2,R1
2178      COM      R1
2179      ;*      177776
2180      ;*      177775
2181      ;*      077777
2182
2183      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2184      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2185      ;*      177400
2186
2187      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2188
2189      PTGEN3: MOV      BUFNO,R0
2190      MOV      #-400,R2
2191      MOV      PTRNO2,R1
2192      COM      R1
2193      ;*      177776
2194      ;*      177775
2195      ;*      077777
2196
2197      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2198      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2199      ;*      177400
2200
2201      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2202
2203      PTGEN3: MOV      BUFNO,R0
2204      MOV      #-400,R2
2205      MOV      PTRNO2,R1
2206      COM      R1
2207      ;*      177776
2208      ;*      177775
2209      ;*      077777
2210
2211      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2212      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2213      ;*      177400
2214
2215      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2216
2217      PTGEN3: MOV      BUFNO,R0
2218      MOV      #-400,R2
2219      MOV      PTRNO2,R1
2220      COM      R1
2221      ;*      177776
2222      ;*      177775
2223      ;*      077777
2224
2225      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2226      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2227      ;*      177400
2228
2229      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2230
2231      PTGEN3: MOV      BUFNO,R0
2232      MOV      #-400,R2
2233      MOV      PTRNO2,R1
2234      COM      R1
2235      ;*      177776
2236      ;*      177775
2237      ;*      077777
2238
2239      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2240      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2241      ;*      177400
2242
2243      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2244
2245      PTGEN3: MOV      BUFNO,R0
2246      MOV      #-400,R2
2247      MOV      PTRNO2,R1
2248      COM      R1
2249      ;*      177776
2250      ;*      177775
2251      ;*      077777
2252
2253      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2254      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2255      ;*      177400
2256
2257      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2258
2259      PTGEN3: MOV      BUFNO,R0
2260      MOV      #-400,R2
2261      MOV      PTRNO2,R1
2262      COM      R1
2263      ;*      177776
2264      ;*      177775
2265      ;*      077777
2266
2267      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2268      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2269      ;*      177400
2270
2271      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2272
2273      PTGEN3: MOV      BUFNO,R0
2274      MOV      #-400,R2
2275      MOV      PTRNO2,R1
2276      COM      R1
2277      ;*      177776
2278      ;*      177775
2279      ;*      077777
2280
2281      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2282      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2283      ;*      177400
2284
2285      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2286
2287      PTGEN3: MOV      BUFNO,R0
2288      MOV      #-400,R2
2289      MOV      PTRNO2,R1
2290      COM      R1
2291      ;*      177776
2292      ;*      177775
2293      ;*      077777
2294
2295      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2296      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2297      ;*      177400
2298
2299      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2300
2301      PTGEN3: MOV      BUFNO,R0
2302      MOV      #-400,R2
2303      MOV      PTRNO2,R1
2304      COM      R1
2305      ;*      177776
2306      ;*      177775
2307      ;*      077777
2308
2309      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2310      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2311      ;*      177400
2312
2313      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2314
2315      PTGEN3: MOV      BUFNO,R0
2316      MOV      #-400,R2
2317      MOV      PTRNO2,R1
2318      COM      R1
2319      ;*      177776
2320      ;*      177775
2321      ;*      077777
2322
2323      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2324      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2325      ;*      177400
2326
2327      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2328
2329      PTGEN3: MOV      BUFNO,R0
2330      MOV      #-400,R2
2331      MOV      PTRNO2,R1
2332      COM      R1
2333      ;*      177776
2334      ;*      177775
2335      ;*      077777
2336
2337      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2338      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2339      ;*      177400
2340
2341      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2342
2343      PTGEN3: MOV      BUFNO,R0
2344      MOV      #-400,R2
2345      MOV      PTRNO2,R1
2346      COM      R1
2347      ;*      177776
2348      ;*      177775
2349      ;*      077777
2350
2351      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2352      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2353      ;*      177400
2354
2355      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2356
2357      PTGEN3: MOV      BUFNO,R0
2358      MOV      #-400,R2
2359      MOV      PTRNO2,R1
2360      COM      R1
2361      ;*      177776
2362      ;*      177775
2363      ;*      077777
2364
2365      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2366      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2367      ;*      177400
2368
2369      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2370
2371      PTGEN3: MOV      BUFNO,R0
2372      MOV      #-400,R2
2373      MOV      PTRNO2,R1
2374      COM      R1
2375      ;*      177776
2376      ;*      177775
2377      ;*      077777
2378
2379      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2380      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2381      ;*      177400
2382
2383      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2384
2385      PTGEN3: MOV      BUFNO,R0
2386      MOV      #-400,R2
2387      MOV      PTRNO2,R1
2388      COM      R1
2389      ;*      177776
2390      ;*      177775
2391      ;*      077777
2392
2393      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2394      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2395      ;*      177400
2396
2397      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2398
2399      PTGEN3: MOV      BUFNO,R0
2400      MOV      #-400,R2
2401      MOV      PTRNO2,R1
2402      COM      R1
2403      ;*      177776
2404      ;*      177775
2405      ;*      077777
2406
2407      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2408      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2409      ;*      177400
2410
2411      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2412
2413      PTGEN3: MOV      BUFNO,R0
2414      MOV      #-400,R2
2415      MOV      PTRNO2,R1
2416      COM      R1
2417      ;*      177776
2418      ;*      177775
2419      ;*      077777
2420
2421      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2422      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2423      ;*      177400
2424
2425      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2426
2427      PTGEN3: MOV      BUFNO,R0
2428      MOV      #-400,R2
2429      MOV      PTRNO2,R1
2430      COM      R1
2431      ;*      177776
2432      ;*      177775
2433      ;*      077777
2434
2435      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2436      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2437      ;*      177400
2438
2439      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2440
2441      PTGEN3: MOV      BUFNO,R0
2442      MOV      #-400,R2
2443      MOV      PTRNO2,R1
2444      COM      R1
2445      ;*      177776
2446      ;*      177775
2447      ;*      077777
2448
2449      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2450      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2451      ;*      177400
2452
2453      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2454
2455      PTGEN3: MOV      BUFNO,R0
2456      MOV      #-400,R2
2457      MOV      PTRNO2,R1
2458      COM      R1
2459      ;*      177776
2460      ;*      177775
2461      ;*      077777
2462
2463      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2464      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2465      ;*      177400
2466
2467      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2468
2469      PTGEN3: MOV      BUFNO,R0
2470      MOV      #-400,R2
2471      MOV      PTRNO2,R1
2472      COM      R1
2473      ;*      177776
2474      ;*      177775
2475      ;*      077777
2476
2477      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2478      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2479      ;*      177400
2480
2481      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2482
2483      PTGEN3: MOV      BUFNO,R0
2484      MOV      #-400,R2
2485      MOV      PTRNO2,R1
2486      COM      R1
2487      ;*      177776
2488      ;*      177775
2489      ;*      077777
2490
2491      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2492      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2493      ;*      177400
2494
2495      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2496
2497      PTGEN3: MOV      BUFNO,R0
2498      MOV      #-400,R2
2499      MOV      PTRNO2,R1
2500      COM      R1
2501      ;*      177776
2502      ;*      177775
2503      ;*      077777
2504
2505      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2506      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2507      ;*      177400
2508
2509      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2510
2511      PTGEN3: MOV      BUFNO,R0
2512      MOV      #-400,R2
2513      MOV      PTRNO2,R1
2514      COM      R1
2515      ;*      177776
2516      ;*      177775
2517      ;*      077777
2518
2519      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2520      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2521      ;*      177400
2522
2523      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2524
2525      PTGEN3: MOV      BUFNO,R0
2526      MOV      #-400,R2
2527      MOV      PTRNO2,R1
2528      COM      R1
2529      ;*      177776
2530      ;*      177775
2531      ;*      077777
2532
2533      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2534      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2535      ;*      177400
2536
2537      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2538
2539      PTGEN3: MOV      BUFNO,R0
2540      MOV      #-400,R2
2541      MOV      PTRNO2,R1
2542      COM      R1
2543      ;*      177776
2544      ;*      177775
2545      ;*      077777
2546
2547      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2548      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2549      ;*      177400
2550
2551      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2552
2553      PTGEN3: MOV      BUFNO,R0
2554      MOV      #-400,R2
2555      MOV      PTRNO2,R1
2556      COM      R1
2557      ;*      177776
2558      ;*      177775
2559      ;*      077777
2560
2561      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2562      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2563      ;*      177400
2564
2565      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2566
2567      PTGEN3: MOV      BUFNO,R0
2568      MOV      #-400,R2
2569      MOV      PTRNO2,R1
2570      COM      R1
2571      ;*      177776
2572      ;*      177775
2573      ;*      077777
2574
2575      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2576      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2577      ;*      177400
2578
2579      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2580
2581      PTGEN3: MOV      BUFNO,R0
2582      MOV      #-400,R2
2583      MOV      PTRNO2,R1
2584      COM      R1
2585      ;*      177776
2586      ;*      177775
2587      ;*      077777
2588
2589      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2590      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2591      ;*      177400
2592
2593      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2594
2595      PTGEN3: MOV      BUFNO,R0
2596      MOV      #-400,R2
2597      MOV      PTRNO2,R1
2598      COM      R1
2599      ;*      177776
2600      ;*      177775
2601      ;*      077777
2602
2603      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2604      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2605      ;*      177400
2606
2607      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2608
2609      PTGEN3: MOV      BUFNO,R0
2610      MOV      #-400,R2
2611      MOV      PTRNO2,R1
2612      COM      R1
2613      ;*      177776
2614      ;*      177775
2615      ;*      077777
2616
2617      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2618      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2619      ;*      177400
2620
2621      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2622
2623      PTGEN3: MOV      BUFNO,R0
2624      MOV      #-400,R2
2625      MOV      PTRNO2,R1
2626      COM      R1
2627      ;*      177776
2628      ;*      177775
2629      ;*      077777
2630
2631      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2632      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2633      ;*      177400
2634
2635      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2636
2637      PTGEN3: MOV      BUFNO,R0
2638      MOV      #-400,R2
2639      MOV      PTRNO2,R1
2640      COM      R1
2641      ;*      177776
2642      ;*      177775
2643      ;*      077777
2644
2645      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2646      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2647      ;*      177400
2648
2649      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2650
2651      PTGEN3: MOV      BUFNO,R0
2652      MOV      #-400,R2
2653      MOV      PTRNO2,R1
2654      COM      R1
2655      ;*      177776
2656      ;*      177775
2657      ;*      077777
2658
2659      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2660      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2661      ;*      177400
2662
2663      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2664
2665      PTGEN3: MOV      BUFNO,R0
2666      MOV      #-400,R2
2667      MOV      PTRNO2,R1
2668      COM      R1
2669      ;*      177776
2670      ;*      177775
2671      ;*      077777
2672
2673      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2674      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2675      ;*      177400
2676
2677      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2678
2679      PTGEN3: MOV      BUFNO,R0
2680      MOV      #-400,R2
2681      MOV      PTRNO2,R1
2682      COM      R1
2683      ;*      177776
2684      ;*      177775
2685      ;*      077777
2686
2687      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2688      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2689      ;*      177400
2690
2691      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2692
2693      PTGEN3: MOV      BUFNO,R0
2694      MOV      #-400,R2
2695      MOV      PTRNO2,R1
2696      COM      R1
2697      ;*      177776
2698      ;*      177775
2699      ;*      077777
2700
2701      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2702      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2703      ;*      177400
2704
2705      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2706
2707      PTGEN3: MOV      BUFNO,R0
2708      MOV      #-400,R2
2709      MOV      PTRNO2,R1
2710      COM      R1
2711      ;*      177776
2712      ;*      177775
2713      ;*      077777
2714
2715      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2716      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2717      ;*      177400
2718
2719      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2720
2721      PTGEN3: MOV      BUFNO,R0
2722      MOV      #-400,R2
2723      MOV      PTRNO2,R1
2724      COM      R1
2725      ;*      177776
2726      ;*      177775
2727      ;*      077777
2728
2729      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2730      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2731      ;*      177400
2732
2733      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2734
2735      PTGEN3: MOV      BUFNO,R0
2736      MOV      #-400,R2
2737      MOV      PTRNO2,R1
2738      COM      R1
2739      ;*      177776
2740      ;*      177775
2741      ;*      077777
2742
2743      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2744      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2745      ;*      177400
2746
2747      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2748
2749      PTGEN3: MOV      BUFNO,R0
2750      MOV      #-400,R2
2751      MOV      PTRNO2,R1
2752      COM      R1
2753      ;*      177776
2754      ;*      177775
2755      ;*      077777
2756
2757      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2758      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2759      ;*      177400
2760
2761      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2762
2763      PTGEN3: MOV      BUFNO,R0
2764      MOV      #-400,R2
2765      MOV      PTRNO2,R1
2766      COM      R1
2767      ;*      177776
2768      ;*      177775
2769      ;*      077777
2770
2771      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2772      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2773      ;*      177400
2774
2775      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2776
2777      PTGEN3: MOV      BUFNO,R0
2778      MOV      #-400,R2
2779      MOV      PTRNO2,R1
2780      COM      R1
2781      ;*      177776
2782      ;*      177775
2783      ;*      077777
2784
2785      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2786      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2787      ;*      177400
2788
2789      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2790
2791      PTGEN3: MOV      BUFNO,R0
2792      MOV      #-400,R2
2793      MOV      PTRNO2,R1
2794      COM      R1
2795      ;*      177776
2796      ;*      177775
2797      ;*      077777
2798
2799      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2800      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2801      ;*      177400
2802
2803      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2804
2805      PTGEN3: MOV      BUFNO,R0
2806      MOV      #-400,R2
2807      MOV      PTRNO2,R1
2808      COM      R1
2809      ;*      177776
2810      ;*      177775
2811      ;*      077777
2812
2813      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2814      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2815      ;*      177400
2816
2817      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2818
2819      PTGEN3: MOV      BUFNO,R0
2820      MOV      #-400,R2
2821      MOV      PTRNO2,R1
2822      COM      R1
2823      ;*      177776
2824      ;*      177775
2825      ;*      077777
2826
2827      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2828      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2829      ;*      177400
2830
2831      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2832
2833      PTGEN3: MOV      BUFNO,R0
2834      MOV      #-400,R2
2835      MOV      PTRNO2,R1
2836      COM      R1
2837      ;*      177776
2838      ;*      177775
2839      ;*      077777
2840
2841      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2842      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2843      ;*      177400
2844
2845      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2846
2847      PTGEN3: MOV      BUFNO,R0
2848      MOV      #-400,R2
2849      MOV      PTRNO2,R1
2850      COM      R1
2851      ;*      177776
2852      ;*      177775
2853      ;*      077777
2854
2855      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2856      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2857      ;*      177400
2858
2859      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2860
2861      PTGEN3: MOV      BUFNO,R0
2862      MOV      #-400,R2
2863      MOV      PTRNO2,R1
2864      COM      R1
2865      ;*      177776
2866      ;*      177775
2867      ;*      077777
2868
2869      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2870      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2871      ;*      177400
2872
2873      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2874
2875      PTGEN3: MOV      BUFNO,R0
2876      MOV      #-400,R2
2877      MOV      PTRNO2,R1
2878      COM      R1
2879      ;*      177776
2880      ;*      177775
2881      ;*      077777
2882
2883      ;*THIRD BLOCK      000377 COUNT UP HIGHER BYTE 0-377
2884      ;*      000776 COUNT DOWN LOWER BYTE 377-0
2885      ;*      177400
2886
2887      ;*'BUFNO' CONTAINS THE STARTING ADDRESS OF THE BUFFER.
2888
2889      PTGEN3: MOV      BUFNO,R0
2890      MOV      #-400,R2
2891      MOV      PTRNO2,R1
2892      COM      R1
2893      ;*      1777
```


F05

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 41
 DZRKLD.P11 31-AUG-76 15:35 T6 WRITE PATTERNS ON THE DISK

SEQ 0057

```

2151 010326 001370      BNE      7$      ;ALL DONE?
2152
2153
2154 010330 012701 000377      MOV      #377,R1      ;3RD BLOCK
2155 010334 010102      MOV      R1,R2      ;GENERATE 3RD BLOCK USING
2156 010336 010120      MOV      R1,(R0)+    ;'COUNT DOWN LOWER BYTE'
2157 010340 060201      ADD      R2,R1      ;'COUNT UP HIGHER BYTE'
2158 010342 022701 177777      CMP      #-1,R1
2159 010346 001373      BNE      9$      ;ALL DONE?
2160 010350 000207      RTS      PC      ;RETURN
2161
2162
2163
2164 010352      006
2165 010353      003
2166 010354      011
2167 010355      000
2168
2169 010356 004737 016716      EXT6:   JSR      PC,ABRT
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206

```

;SECTOR POINTER TABLE. DATA TRANSFERS ARE DONE IN BLOCKS (3 SECTORS
 ;EACH) IN THE CYCLIC ORDER: 0-2, 6-10, 3-5, 11-13, 0-2, AND SO ON.

```

SECPTR: .BYTE 6
        .BYTE 3
        .BYTE 11
        .BYTE 0

```

EXT6: JSR PC,ABRT

```

;*****
;TEST 7      READ, SOFTWARE COMPARE, WRITE CHECK OF THE PATTERNS

```

```

;****TEST 6 (WRITING PATTERNS ON DISK) SHOULD BE DONE BEFORE DOING****
;****THIS TEST.****

```

```

;*IN THIS TEST THE PATTERNS THAT WERE WRITTEN BEFORE (IN THE
;*PREVIOUS TEST) ARE READ BACK AND SOFTWARE COMPARED.
;*WHILE THE SOFTWARE COMPARISON IS TAKING PLACE, AN OVERLAPPING
;*WRITE CHECK' IS DONE FOR THE SAME BLOCK. THE READING
;*BACK OF EACH BLOCK (4 SECTORS) IS DONE IN THE SAME
;*MANNER AS DESCRIBED IN THE BEGINNING OF PREVIOUS TEST.
;*OVERLAPPING OPERATIONS AND STAGGERING OF BLOCKS ARE DONE IN
;*A SIMILIAR MANNER.
;*THE FOLLOWING IS A DESCRIPTION OF HOW ERRORS ARE HANDLED.
;*IF A 'SIN' OR 'HE' OCCURS IT IS REPORTED AND THAT BLOCK
;*IS SKIPPED. IN THIS STAGE OF TESTING THESE ERRORS SHOULD NOT
;*NORMALLY OCCUR.
;*IF A CHECKSUM ERROR OCCURS, CONTROL IS TRANSFERRED TO
;*'CSEERROR'. FIRST THE CSE IS REPORTED. THE SECTOR GIVING
;*CSE IS READ AGAIN. IN ALL 3 TRIES ARE DONE. IF STILL
;*THE ERROR PRESISTS THAT SECTOR IS ABORTED AND THE REST
;*OF THE SECTORS ARE READ AND CHECKED FOR CSE. IF
;*AGAIN SOME OTHER SECTOR GIVES CSE, REPORTING AND RETRIES
;*ARE DONE AGAIN.
;*NEXT SOFTWARE COMPARISON IS DONE OF THE DATA THAT WAS
;*READ BACK. ON GETTING A DATA MISCOMPARISON
;*THE RELEVANT INFO(GOOD DATA, BAD DATA, ADDRESS ETC.) IS
;*STORED. AFTER THE WHOLE BLOCK HAS BEEN CHECKED, THE DATA
;*ERROR/S IS/ARE REPORTED. THEN THE BLOCK IS READ AGAIN. IN ALL
;*THREE TRIES ARE DONE.
;*WHILE THE DATA COMPARISON (SOFTWARE) IS GOING ON THE 'WRITE
;*CHECK' IS ALSO IN PROGRESS. IF A WRITE CHECK ERROR OCCURS THE
;*CONTROL IS TRANSFERRED TO 'WCEROR'. WRITE CHECK OF THE SECTOR
;*THAT GAVE WCE IS DONE AGAIN. IN ALL THREE TRIES ARE DONE.

```

G05

MAINDEC-11-DZRLD-D
DZRLD.P11MACY11 27(1006) 04-OCT-76 14:26 PAGE 42
31-AUG-76 15:35

T7 READ, SOFTWARE COMPARE, WRITE CHECK OF THE PATTERNS

SEQ 0058

2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262

010362 000004
010364 012737 177764 001510
010372 012737 177773 001512
010400 012737 177742 001514
010406 012737 177742 001516
010414 012737 177764 001520
010422 012737 177742 001522
010430 012737 177152 001474
010436 005037 001476

010442 012737 001414 001424
010450 013737 001230 001450

010456 005037 001504
010462 005037 001506
010466 012737 177775 001252
010474 012737 177776 001254
010502 012737 000003 001256

010510 013737 001450 001440
010516 013737 001406 001442

010524 012737 176400 001444

010532 013737 001450 001432
010540 012737 176400 001436
010546 013737 001404 001434

010554 000404

010556 104415
010560 104416
010562 000401

010564 104415

010566 013777 001432 170670
010574 013777 001436 170656
010602 013777 001434 170652

010610 012777 000405 170640

010616 013737 001406 001446
010624 017737 170574 001430
010632 004777 170572

TST7: SCOPE
MOV #-14,ERCNT3 ;ALLOW 12 ERRORS AT THE MOST
MOV #-5,ERCNT4 ;ALLOW 5 ERRORS AT THE MOST
MOV #-36,ERCNT5 ;ALLOW 10 ERRORS AT THE MOST
MOV #-36,ERCNT6 ;ALLOW 10 ERRORS AT THE MOST
MOV #-14,ERCNT7 ;ALLOW 12 ERRORS AT THE MOST
MOV #-36,ERCNT8 ;ALLOW 10 ERRORS AT THE MOST
MOV #-626,INDX1 ;SET COUNT FOR 626 TRACKS
CLR INDX2 ;CLR COUNT FOR 4 BLOCKS ON A TRACK

MOV #PATO,PRSPAT ;INITLZE PTR TO PATRN GENRTR
MOV DRIVAD,ADRES ;INITLZE DRV#,ADRES

BEGIN: CLR ERCNT1 ;IF > 0, MEANS THAT RETRIES
CLR ERCNT2 ;DONE AFTER CSE OR CSE CHKD
MOV #-3,RETRY1 ;RETRY COUNT FOR CSE
MOV #-2,RETRY2 ;RETRY COUNT FOR SFTWRE MISCMP'N
MOV #3,RETRY3 ;RETRY COUNT FOR WCE

MOV ADRES,WDSKAD ;DISK ADRES TO WRT CHK WITH
MOV PBUF1,WBUSAD ;USE THIS BUFR 1 TO WRT CHK
;OR THE BUFR INDICATED BY THE USER
MOV #-1400,WWRDCN ;WRT CHK 1 BLOCK=3SECS=1400 WRDS

READ: MOV ADRES,DSKADR ;DISK ADRES TO READ FROM
MOV #-1400,WRCNT ;1 BLOCK = 3 SECTORS = 1400 WRDS
MOV PBUFO,BUSADR ;USE 'IOBUFO' TO READ INTO
;OR THE BUFR INDICATED BY THE USER

BR RDAGAIN

LUPSIN: CON.RESET
DRV.RESET
BR RDAGAIN

LUPHE: CON.RESET

RDAGAIN: MOV DSKADR,ARKDA ;READ FROM THIS DSK-ADRES
MOV WRCNT,ARKWC ;THIS # OF WORDS
MOV BUSADR,ARKBA ;INTO THIS BUFR

MOV #405,ARKCS ;READ,SSE,GO

MOV PBUF1,BUFNO ;SET UP STARTING ADRES
MOV PRSPAT,PGSUBR
JSR PC,PGSUBR ;GO GENERATE A BUFFER

H05

MAINDEC-11-DZRLD-D
DZRLD.F11

31-AUG-76

MACY11 27(1006)
15:3504-OCT-76
T7

14:26 PAGE 43

READ, SOFTWARE COMPARE, WRITE CHECK OF THE PATTERNS

SEQ 0059

; OF 1400 WORDS USING THIS
; PATRN GENRATR; DONE WITH PATRN GENRTNG,
; WAIT FOR CNT RDY TO SET
; (FROM PREVIOUS READ); CNT RDY SET
; HARD ERROR?

; HARD ERROR

; ALLOW 12 ERRORS AT MOST

; IF MORE, EXIT

; SIN SET?
; NO

; SIN ON READ

; ALLOW 5 ERRORS AT MOST

; IF MORE, EXIT

; CHECKING CSE FOR 1ST TIME
; NO BRANCH
; INDICATE THAT CSE HAS BEEN
; CHECKED ONCE

; CHECK SUM EROR?

; GET THE RETRY #
; SAVE IT FOR TYPEOUT
; CSE

; ALLOW 10 ERRORS AT MOST

; IF MORE, EXIT
; GO, SERVICE CSE; CLR FLAG INDICATING SOFTWARE
; COMPARE DONE; WRT CHK DONE BEFORE OR
; IT'S 1ST TIME?
; IF DONE, BRANCH OTHERWISE DO IT
; WRT CHK FROM THIS DSK-ADRES
; THIS # FO WORDS2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318

010636 104421

CON.RDY

010640 032777 040000 170610

BIT #BIT14,ARKCS
BEQ NOHE

010646 001416

010650 012737 010564 001110

MOV #LUPHE,\$LPERR
JSR PC,GETINF

010656 004737 016140

ERROR 23

010662 104023

CON.RESET

010664 104415

INC ERCNT3

010666 005237 001510

BNE 1\$

010672 001002

JMP EXT7

010674 000137 012114

JMP FINISH

010700 000137 011436

BIT #BIT9,ARKDS

010704 032777 001000 170540

BEQ NOSIN

010712 001417

010714 012737 010556 001110

MOV #LUPSIN,\$LPERR
JSR PC,GETINF

010722 004737 016140

ERROR 11

010726 104011

CON.RESET

010730 104415

DRV.RESET

010732 104416

INC ERCNT4

010734 005237 001512

BNE 1\$

010740 001002

JMP EXT7

010742 000137 012114

JMP FINISH

010746 000137 011436

1\$:

010752 005737 001504

TST ERCNT1

010756 001031

BNE WRTCHK

010760 005237 001504

INC ERCNT1

010764 032777 000002 170462

BIT #BIT1,ARKER

010772 001423

BEQ WRTCHK

010774 012737 010456 001110

MOV #BEGIN,\$LPERR

011002 004737 016140

JSR PC,GETINF

011006 013737 001252 001202

MOV RETRY1,\$REG10

011014 062737 000004 001202

ADD #4,\$REG10

011022 104024

ERROR 24

011024 005237 001514

INC ERCNT5

011030 001002

BNE 1\$

011032 000137 012114

JMP EXT7

011036 000137 011600

JMP CSEROR

1\$:

011042 005037 001506

WRTCHK: CLR ERCNT2

011046 022737 000003 001256

CMP #3,RETRY3

011054 001016

BNE SFTCMP

011056 013777 001440 170400

WCAGAIN: MOV WDSKAD,ARKDA

011064 013777 001444 170366

MOV WWRDCN,ARKWC

MAINDEC-11-DZRLD-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 44
 DZRLD.P11 31-AUG-76 15:35 T7 READ, SOFTWARE COMPARE, WRITE CHECK OF THE PATTERNS

SEQ 0060

```

2319 011072 013777 001442 170362      MOV      WBUSAD,ARKBA      ;WITH THIS BUFFER
2320 011100 012777 000407 170350      MOV      #407,ARKCS      ;WRT CHK,GO,SSE
2321 011106 005337 001256              DEC      RETRY3          ;INDICATE WRT CHK DONE
2322 011112 005737 001506      SFTCMP: TST      ERCNT2      ;SOFTWARE CMPARE DONE ONCE BEFORE?
2323 011116 001060              BNE      WCREPT          ;IF SFTWARE CMPARE HAS BEEN DONE
2324 011120 005237 001506      INC      ERCNT2          ;ONCE DON'T DO IT AGAIN, OTHERWISE,
2325                                ;DO IT. INDICATE IT IS DONE.
2326                                ;MORE THAN ONCE BEFORE.
2327                                ;IF THIS IS 1ST TIME THRU &
2328                                ;WRT CHK WAS DONE ONCE BEFORE
2329                                ;DO SOFTWARE COMPARISON OF
2330                                ;THE DATA THAT WAS READ FROM
2331                                ;THE DISK
2332
2333 011124 012702 001266      MOV      #BUFR,R2      ;INITLZE PTR TO BUFR STORING
2334                                ;ADRES OF BAD DATA
2335 011130 012703 001320      MOV      #BUFR1,R3      ;STORE EXPCTD DATA STARTING HERE
2336 011134 012704 001352      MOV      #BUFR2,R4      ;STORE RECVD (BAD) DATA
2337                                ;STARTING HERE
2338
2339 011140 005037 001500              CLR      INDX3          ;CLR FLAG INDICATING MISCMPRE
2340
2341 011144 013700 001404      COMPAR: MOV      PBUFO,R0      ;INITLZE PTR TO 'RECVD DATA' BUFR
2342 011150 012737 177764 001502      MOV      #-14,INDX4      ;STORE AND REPORT 12 OR LESS DATA ERRORS
2343 011156 013701 001406      MOV      PBUF1,R1      ;INITLZE PTR TO 'EXPCTD DATA' BUFR
2344 011162 012737 176400 001260      MOV      #-1400,INADR      ;SET COUNT
2345      CMPAGAN: CMP      (R0),(R1)      ;CORRECT WORD READ FROM DISK?
2346      BEQ      1$
2347      JMP      MISCMPE
2348      1$: TST      (R0)+      ;BRANCH IF MISCMPE ERROR
2349      TST      (R1)+      ;INCRMNT PTRS
2350      INC      INADR      ;TO NXT WORDS
2351      BNE      CMPAGAN      ;DONE WITH CMPRISON?
2352                                ;IF NOT, COMPARE THE REST
2353
2354 011212 005737 001500      TST      INDX3          ;WAS THERE A BAD DATA WORD
2355                                ;(EVEN AFTR RETRYING)
2356      BEQ      WCREPT          ;NO, BRANCH
2357
2358 011220 012737 010532 001110      REPMSC: MOV      #READ,$LPERR      ;DATA ERROR
2359 011226 104025              ERROR      25      ;ALLOW 10 EPRORS AT MOST
2360 011230 005237 001516      INC      ERCNT6
2361 011234 001002              BNE      1$
2362 011236 000137 012114      JMP      EXT7
2363 011242 005737 001254      1$: TST      RETRY2
2364 011246 001404      BEQ      WCREPT
2365 011250 005237 001254      INC      RETRY2
2366 011254 000137 010532      JMP      READ
2367
2368 011260 104421      WCREPT: CON.RDY      ;WAIT FOR CNTRL RDY FROM
2369                                ;PREVIOUS WRT CHK
2370 011262 022737 177776 001254      CMP      #-2,RETRY2      ;IF THERE WAS A RETRY AFTER MISC
2371                                ;-OMPARISON, DO WRT CHK AGAIN
2372
2373 011270 001417      BEQ      ERWCHK
2374 011272 000401      BR      LUPWCE+2

```


J05

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 45
 DZRKLD.P11 31-AUG-76 15:35 T7 READ, SOFTWARE COMPARE, WRITE CHECK OF THE PATTERNS

SEQ 0061

```

2375 011274 104415 LUPWCE: CON.RESET
2376 011276 013777 001440 170160 MOV WDSKAD,ARKDA ;WRT CHK WITH THIS DSK-ADRES
2377 011304 013777 001444 170146 MOV WWRDCN,ARKWC ;THIS # OF WORDS
2378 011312 013777 001442 170142 MOV WBUSAD,ARKBA ;THIS BUS ADRES
2379 011320 012777 000407 170130 MOV #407,ARKCS
2380
2381 011326 104421
2382 011330 012737 011274 001110 ERWCHK: CON.RDY
2383 011336 032777 040000 170106 MOV #LUPWCE,$LPERR
2384 011344 001410 BIT #BIT14,ARKDS ;HARD ERROR?(FROM WRT CHK)
2385 BEQ XHE ;NO,BRANCH
2386 011346 004737 016140 JSR PC,GETINF
2387 011352 104026 ERROR 26 ;HE ON WRT CHK
2388 011354 005237 001520 INC ERCNT7 ;ALLOW 12 ERRORS AT MOST
2389 011360 001002 BNE XHE
2390 011362 000137 012114 JMP EXT7 ;IF MORE, EXIT
2391
2392 011366 032777 000001 170060 XHE: BIT #BIT0,ARKER ;WRITE CHECK ERROR?
2393 011374 001420 BEQ FINISH
2394 011376 004737 016140 JSR PC,GETINF
2395 011402 012737 000003 001202 MOV #3,$REG10 ;GET TRY #
2396 011410 163737 001256 001202 SUB RETRY3,$REG10 ;SAVE IT FOR TYPEOUT
2397 011416 104027 ERROR 27 ;WRT CHK ERROR
2398 011420 005237 001522 INC ERCNT8 ;ALLOW 10 ERRORS AT MOST
2399 011424 001002 BNE 1$
2400 011426 000137 012114 JMP EXT7 ;IF MORE, EXIT
2401 011432 000137 012006 1$: JMP WCEROR
2402
2403
2404
2405 ;THERE WAS NO WCE. DONE
2406 ;WITH ALL CHECKING FOR SOFT
2407 ;ERRORS,ETC. MODIFY PARAMETERS
2408 ;TO CHECK NXT BLOCK ON
2409 ;THE DISK
2410 011436 022737 001422 001424 FINISH: CMP #PAT3,PRSPAT ;FIND OUT THE NXT PATRN GENRATR
2411 011444 001404 BEQ 1$ ;TO USE FOR GENRATING THE
2412 011446 062737 000002 001424 ADD #2,PRSPAT ;BUFR,STORE POINTER TO
2413 011454 000403 BR 2$ ;GENRATR ROUTINE IN 'PRSPAT'
2414 011456 012737 001414 001424 1$: MOV #PATO,PRSPAT ;NOTE THER R 4 PAT-GENRATRS
2415 011464 013701 001476 2$: MOV INDX2,R1 ;PATO-PAT1-PAT2-PAT3-----PATO-
2416 011470 116101 010352 MOVB SECPTR(R1),R1 ;FORM SECTR # TO BE USED NEXT
2417 011474 042737 000017 001450 3$: BIC #17,ADRES ;GET SECTR #
2418 011502 050137 001450 BIS R1,ADRES ;MASK SECTR BITS
2419 ;FORM THE NEW DISK-ADRES
2420 ;FORM THE NXT BLOCK TO BE
2421 ;CHECKED.
2422 011506 005237 001476 INC INDX2 ;DONE ALL 4 BLOCKS(3 SECS
2423 011512 022737 000004 001476 CMP #4,INDX2 ;EACH) ON THIS TRACK?
2424 011520 001025 BNE GOBAK ;IF NOT, GO BAK & DO THE
2425 ;NXT BLOCK ON THIS TRACK
2426
2427
2428 011522 005037 001476 DONTRK: CLR INDX2 ;REINITLZE COUNT FOR
2429 011526 032737 000001 001474 BIT #BIT0,INDX1 ;4 BLOCKS ON THIS TRACK
2430 ;WHICH SUR TO DO NXT? 0 OR 1

```

K05

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 46
 DZRKLD.P11 31-AUG-76 15:35 T7 READ, SOFTWARE COMPARE, WRITE CHECK OF THE PATTERNS

SEQ 0062

```

2431 011534 001407          BEQ      DOSUR1          ;BRANCH, DO SUR 1 NXT
2432
2433 011536 062737 000040 001450      ADD      #40,ADRES
2434
2435 011544 042737 000020 001450      BIC      #20,ADRES          ;CLR SUR BIT, DO SUR 0 NXT
2436 011552 000403          BR      DONE
2437
2438 011554 052737 000020 001450  DOSUR1: BIS      #20, ADRES          ;SET SUR BIT, DO SUR 1 NXT
2439
2440 011562 005237 001474          DONE:  INC      INDX1          ;DONE WITH ALL 626 TRACKS?
2441 011566 001002          BNE      GOBAK
2442 011570 000137 012120          JMP      TST10
2443
2444 011574 000137 010456          GOBAK: JMP      BEGIN          ;IF NOT, GO BAK & CHECK
2445                                     ;THE NXT TRACK
2446
2447                                     ;CSEROR
2448
2449                                     ;THIS IS THE ENTRY POINT
2450 011600          CSEROR:          ;FOR SERVICE ROUTINE FOR CHECK
2451                                     ;SUM EROR. CONTROL IS XFERED
2452                                     ;HERE ONCE CSE OCCURS
2453
2452 011600 017700 167660          MOV      DRKDA,R0          ;GET RKDA AFTER CSE
2453 011604 010001          MOV      R0,R1          ;SAVE RKDA
2454 011606 032700 000017          BIT      #17,R0          ;FORM THE ADRES OF THE SECTR
2455 011612 001002          BNE      1$          ;WHICH GAVE CSE
2456 011614 162700 000004          SUB      #4,R0
2457 011620 005300          1$: DEC      R0          ;R0 CONTAINS DSK-ADRES WHERE
2458                                     ;CSE OCURED
2459 011622 020037 001432          CMP      R0,DSKADR          ;DID A PREVIOUS RETRY (IF ANY)
2460                                     ;GIVE CSE ON SAME ADRES
2461 011626 001021          BNE      2$          ;NO, THIS IS A FRESH CSE, BRANCH.
2462                                     ;IF THIS WAS A CSE ON A
2463 011630 005237 001252          INC      RETRY1          ;RETRY INCMNT RETRY COUNT.
2464                                     ;BRANCH IF 3 RETRIES HAVEN'T
2465 011634 001021          BNE      3$          ;BEEN DONE
2466                                     ;GO REPORT CSE
2467                                     ;IF CSE WAS IN THE LAST SECTR OF
2468                                     ;THE 3 SEC BLOCK, GO TO 'WRTCHK'
2469                                     ;OTHERWISE CHECK THE REST OF
2470                                     ;THE SECTORS FOR CSE.
2471
2471 011636 010102          MOV      R1,R2
2472 011640 042702 177760          BIC      #177760,R2
2473 011644 001002          BNE      8$
2474 011646 000137 011042          JMP      WRTCHK
2475 011652 162702 000003          8$: SUB      #3,R2
2476 011656 100372          BPL      7$
2477
2478 011660 010100          6$: MOV      R1,R0
2479 011662 012737 177775 001252      MOV      #-3,RETRY1
2480 011670 000403          BR      3$
2481
2482 011672 012737 177776 001252  2$: MOV      #-2,RETRY1          ;ALLOW 2 MORE TRIES FOR
2483                                     ;THE CSE ON SAME DISK-ADRES
2484
2485 011700 010037 001432          3$: MOV      R0,DSKADR          ;SAVE DSK-ADRES FOR DOING
2486                                     ;THE RETRY-READ

```


L05

MAINDEC-11-DZAKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 47

ZAKLD.P11 31-AUG-76 15:35

T7

READ, SOFTWARE COMPARE, WRITE CHECK OF THE PATTERNS

SEQ 0063

```

2487 011704 163700 001450      SUB      ADRES,RO      ;MODIFY THE
2488 011710 005200              INC      RO      ;BUS ADRES & WORD COUNT
2489 011712 005300              DEC      RO      ;TO BE USE ON RETRY-
2490 011714 001407              BEQ      5$      ;READ
2491 011716 062737 001000 001434  ADD      #1000,BUSADR
2492 011724 062737 000400 001436  ADD      #400,WRDCNT
2493 011732 000767              BR       4$
2494
2495 011734 104415              5$:  CON.RESET      ;CLR THE CSE IN RKER
2496 011736 000137 010566      JMP      RDAGAIN      ;GO BACK, READ AGAIN
2497
2498
2499
2500
2501      ;MISCMP      ;THIS IS THE ENTRY POINT
2502      ;FOR SERVICE ROUTINE, FOR
2503      ;DATA ERROR (MISCOMPARISON)
2504      ;ON SOFTWARE COMPARISON
2505      ;OF DATA READ FROM THE DISK BLOCK
2506
2507
2508 011742 104421      MISCMP: CON.RDY      ;WAIT FOR CNTRL RDY FROM
2509
2510 011744 032777 040000 167504      BIT      #BIT14,ZRKCS      ;PREVIOUS WRT CHK
2511 011752 001401              BEQ      1$
2512 011754 104415              CON.RESET
2513
2514 011756 010022              1$:  MOV      RO,(R2)+      ;CLR WCE IN RKER
2515
2516 011760 012123              MOV      (R1)+,(R3)+      ;BUT STILL DATA ERROR ENTER HERE
2517 011762 012024              MOV      (R0)+,(R4)+      ;STORE MEM ADRES WHERE
2518 011764 005237 001500      INC      INDX3      ;DATA ERROR OCCURED
2519
2520 011770 005237 001502              INC      INDX4      ;SAVE EXPCTD DATA
2521 011774 001402              BEQ      4$      ;SAVE DATA RECVD (BAD)
2522
2523 011776 000137 011170              JMP      CMPAGAN      ;INDICATE MISCMPRISON
2524
2525 012002 000137 011220              4$:  JMP      REPMSC      ;STORE ONLY 12(DEC) ERRORS
2526
2527
2528
2529
2530
2531
2532
2533 012006 005737 001256      WCEROR: TST      RETRY3      ;THIS THE ENTRY POINT FOR
2534 012012 003035              BGT      CLRERR      ;ERROR SERVICE ON GETTING A
2535
2536
2537 012014 012737 000003 001256      MOV      #3,RETRY3      ;WRITE CHECK ERROR.
2538
2539 012022 017700 167436              MOV      ZRKDA,RO      ;DONE 3 TRIES?
2540 012026 010002              MOV      RO,R2      ;IF NOT SKIP, OTHERWISE REPORT
2541 012030 042702 177760              BIC      #177760,R2      ;W C EROR
2542 012034 001002              3$:  BNE      4$      ;SET CONT FOR RETRIES

```

```

;IF WCE WAS IN THE LAST SECTOR
;OF THE BLOCK, NO MORE SECS
;TO CHECK, GO TO 'FINISH'
;IF IT WASN'T LAST SECTOR OF THE

```

M05

MAINDEC-11-DZRKL-D
DZRKLD.P11MACY11 27(1006)
31-AUG-76 15:3504-OCT-76 14:26 PAGE 48
T7 READ, SOFTWARE COMPARE, WRITE CHECK OF THE PATTERNS

SEQ 0064

2543	012036	000137	011436		JMP	FINISH		:BLOCK, THEN CHECK REMAINING
2544	012042	162702	000003	4\$:	SUB	#3,R2		:SECTORS, (STARTING FROM THE
2545	012046	100372			BPL	3\$		:SEC AFTER THE ONE GIVING WCE)
2546	012050	010001		1\$:	MOV	RO,R1		:SAVE DISK ADRES
2547	012052	163700	001440		SUB	WDSKAD,RO		
2548	012056	010137	001440		MOV	R1,WDSKAD		:GET SAVED DISK ADRES
2549	012062	005200			INC	RO		
2550	012064	005300		2\$:	DEC	RO		
2551	012066	001407			BEQ	CLRERR		
2552	012070	062737	001000 001442		ADD	#1000,WBUSAD		:FORM THE NEW BUS ADRES
2553	012076	062737	000400 001444		ADD	#400,WWRDCN		:FORM THE NEW WORD COUNT
2554	012104	000767			BR	2\$		
2555	012106	104415		CLRERR:	CON.RESET			
2556	012110	000137	011056		JMP	WCAGAIN		
2557								
2558	012114	004737	016716	EXT7:	JSR	PC,ABRT		

```

*****
;TEST 10 WRITE, WRITE CHECK ON CYLINDERS 127, 128
;THIS TEST WRITES 12 UNIQUE PATTERNS (ONE FOR EACH
;SECTOR) ON CYLINDERS 127 AND 128, THEN WRITE
;CHECK IS DONE TO SEE IF THEY WERE WRITTEN
;CORRECTLY. IT SHOULD BE NOTED THAT THERE IS
;CHANGE IN 'WRITE' CURRENT AT THIS CYLINDER.
;PATTERNS ARE RELOCATED ON THE CYLINDERS AFTER EACH
;WRITE/WRITE CHECK CYCLE. THUS THE FIRST TIME
;PATTERN 'SP1' IS WRITTEN ON SECTOR 0, PATTERN 'SP2' ON
;SECTOR 2, ETC. AFTER THIS WRITE/WRITE CHECK CYCLE
;IS OVER PATTERN 'SP2' IS WRITTEN ON SECTOR 0, PATTERN
;: 'SP3' ON SECTOR 1 AND SO ON. THIS WRITE/WRITE
;CHECK CYCLE IS REPEATED 12 TIMES, THUS
;THE LAST WRITE/WRITE CHECK IS DONE USING
;PATTERN 'SP12' ON SECTOR 0, PATTERN 'SP1' IS
;WRITTEN ON SECTOR 1, PATTERN 'SP2' ON SECTOR 2,
;ETC. IF YOU WANT TO WRITE ANY OTHER PATTERNS
;FILL IN THE PATTERNS YOU WANT IN LOCATIONS
;'SP1', 'SP2', ...ETC.

```

2581								
2582	012120	000004			TST10:	SCOPE		
2583								
2584	012122	012737	177764 001504		MOV	#-14,ERCNT1		:ALLOW 12 ERRORS AT MOST
2585	012130	012737	177764 001506		MOV	#-14,ERCNT2		:ALLOW 12 ERRORS AT MOST
2586	012136	012737	177723 001510		MOV	#-55,ERCNT3		:ALLOW 15 ERRORS AT MOST
2587	012144	012702	012626		MOV	#SP1,R2		:INITIALIZE POINTER TO PATRN
2588	012150	010201		DOWRT:	MOV	R2,R1		
2589	012152	012700	007740		MOV	#7740,RO		:SET UP CYL ADRES BITS (127)
2590	012156	053700	001230		BIS	DRIVAD,RO		:SET UP DRIVE # BITS
2591	012162	005003		WRL01:	CLR	R3		
2592	012164	104415		WRERR:	CON.RESET			
2593								
2594	012166	010077	167272	WRL0:	MOV	RO,ARKDA		:ADRES THE DRIVE
2595	012172	012777	177400 167260		MOV	#-400,ARKWC		:WRITE 1 SECTOR
2596	012200	010177	167256		MOV	R1,ARKBA		:USE THIS PATTERN
2597	012204	012777	004003 167244		MOV	#4003,ARKCS		:WRITE, GO
2598	012212	104421			CON.RDY			

N05

MAINDEC-11-DZRLD-D MACY11 27(1006) 04-OCT-76, 14:26 PAGE 49
DZRLD.P11 31-AUG-76 15:35 T10 WRITE, WRITE CHECK ON CYLINDERS 127, 128

SEQ 0065

2599	012214	032777	140000	167234	BIT	#140000,ARKCS	;ANY ERROR?
2600	012222	001414			BEQ	4\$	
2601	012224	012737	012164	001110	MOV	#WRERR,\$LPERR	;SET ADRES FOR LOOPING ON ERROR
2602	012232	004737	016106		JSR	PC,GT4RG	;GET TKCS, ER, DS, DA
2603	012236	104021			ERROR	21	;ERROR OCCURRED ON DOING A WRITE
2604	012240	104415			CON.RESET		;CLEAR THE ERROR
2605	012242	005237	001504		INC	ERCNT1	;ALLOW 12 ERRORS ONLY
2606	012246	001002			BNE	4\$	
2607	012250	000137	012656		JMP	EXT10	
2608							
2609	012254	005200			4\$: INC	R0	
2610	012256	005203			INC	R3	;KEEP COUNT
2611	012260	022701	012654		CMP	#SP12,R1	;USE PATTERNS IN A CYCLIC FASHION
2612	012264	001002			BNE	3\$	
2613	012266	012701	012624		MOV	#SP1-2,R1	
2614	012272	005721			3\$: TST	(R1)+	;INCREMENT POINTERS TO NEXT PATTERN
2615	012274	020327	000014		CMP	R3,#14	;DONE SURFACE 0?
2616	012300	002732			BLT	WRL0	;NO
2617	012302	001005			BNE	2\$	;YES, IF CHANGING HEADS
2618	012304	010201			MOV	R2,R1	
2619	012306	042700	000017		BIC	#17,R0	;SET UP CORRECT ADDRESS ETC.
2620	012312	052700	000020		BIS	#20,R0	
2621							
2622	012316	020327	000030		2\$: CMP	R3,#30	;DONE WITH WRITING SURFACE 1?
2623	012322	001321			BNE	WRL0	;NO, BRANCH
2624							
2625	012324	032700	007700		WRHI: BIT	#7700,R0	;DONE WITH BOTH CYLINDERS - 127 & 128?
2626	012330	001405			BEQ	DOWCHK	;YES
2627	012332	012700	010000		MOV	#10000,R0	;NO, DO CYLINDER 128
2628	012336	053700	001230		BIS	DRIVAD,R0	
2629	012342	000707			BR	WRL01	;GO BACK
2630							;CYLINDERS 127 AND 128 HAVE BEEN
2631							
2632							
2633	012344	010201			DOWCHK: MOV	R2,R1	;WRITTEN, NOW DO WRITE CHECK
2634	012346	012737	177775	001252	MOV	#-3,RETRY1	;INITIALIZE POINTER TO FIRST PATTERN
2635	012354	012700	010000		MOV	#10000,R0	;RETRY COUNT
2636	012360	053700	001230		BIS	DRIVAD,R0	;DO CYLINDER 128 FIRST
2637	012364	005003			WCHI1: CLR	R3	
2638	012366	010077	167072		WCERR: MOV	R0,ARKDA	;ADRES THE DRIVE
2639							
2640	012372	012777	177400	167060	WCHI: MOV	#-400,ARKWC	;WRITE CHECK 1 SECTOR
2641	012400	010177	167056		MOV	R1,ARKBA	;WITH THIS PATTERN
2642	012404	012777	004007	167044	MOV	#4007,ARKCS	;WRITE CHECK, GO
2643	012412	104421			CON.RDY		
2644							
2645	012414	032777	040000	167030	BIT	#40000,ARKDS	;HE?
2646	012422	001406			BEQ	1\$	;NO
2647	012424	004737	016140		JSR	PC,GETINF	
2648	012430	104026			ERROR	26	;HE ON DOING WRT CHK
2649	012432	005237	001506		INC	ERCNT2	;ALLOW 12 ERRORS ONLY
2650	012436	001507			BEQ	EXT10	;IF MORE, EXIT
2651	012440	032777	000001	167006	1\$: BIT	#BIT0,ARKER	;WCE?
2652	012446	001425			BEQ	4\$	;NO
2653	012450	012737	012366	001110	MOV	#WCERR,\$LPERR	;SET ADRES FOR LOOPING ON ERROR
2654	012456	004737	016140		JSR	PC,GETINF	;GET INFO ON ERROR

MAINDEC-11-DZRL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 50
DZRLD.P11 31-AUG-76 15:35 T10 WRITE, WRITE CHECK ON CYLINDERS 127, 128

SEQ 0066

```

2655 012462 013737 001252 001202      MOV      RETRY1,$REG10
2656 012470 062737 000004 001202      ADD      #4,$REG10      :WCE ON DOING WRITE CHECK, WITH
2657 012476 104027          ERROR      27      :PATTERN STORED IN R1
2658 012500 005237 001252          INC      RETRY1      :DO 3 TIMES IN ALL
2659 012504 001330          BNE      WCERR
2660 012506 005237 001510          INC      ERCNT3      :ALLOW 15 ERRORS ONLY
2661 012512 001461          BEQ      EXT10      :IF MORE, EXIT
2662 012514 012737 177775 001252      MOV      #-3,RETRY1
2663
2664 012522 005200          4$: INC      R0      :KEEP TRACK OF DISK-ADRES
2665 012524 005203          INC      R3      :AND COUNT
2666 012526 022701 012654          CMP      #SP12,R1      :USE PATTERNS IN CYCLIC
2667 012532 001002          BNE      3$
2668 012534 012701 012624          MOV      #SP1-2,R1      :FASHION
2669 012540 005721          3$: TST      (R1)+      :INCREMENT POINTER TO NEXT PATTERN
2670 012542 020327 000014          CMP      R3,#14      :DONE SURFACE 0?
2671 012546 002711          BLT      WCHI      :NO
2672 012550 001005          BNE      2$
2673 012552 010201          MOV      R2,R1      :IF CHANGING HEADS (0-1), SET CORRECT
2674 012554 042700 000017          BIC      #17,R0      :ADRES BITS
2675 012560 052700 000020          BIS      #20,R0
2676
2677 012564 020327 000030          2$: CMP      R3,#30      :DONE WRITE CHECKING SURFACE 1?
2678 012570 001300          BNE      WCHI      :NO, GO BACK
2679
2680 012572 032700 007700          WOLO: BIT      #7700,R0      :DONE BOTH CYLINDERS - 127, 128?
2681 012576 001005          BNE      REPEAT      :YES, BRANCH
2682 012600 012700 007740          MOV      #7740,R0      :DO CYLINDER 127 NOW
2683 012604 053700 001230          BIS      DRIVAD,R0
2684 012610 000665          BR      WCHI1
2685
2686
2687 012612 005722          REPEAT: TST      (R2)+      :RELOCATE THE PATTERNS ON THE
2688 012614 020227 012656          CMP      R2,#SP12+2      :CYLINDERS AND DO IT AGAIN
2689 012620 001420          BEQ      TST11      :EXIT
2690 012622 000137 012150          JMP      DOWAT      :THIS TEXT)
2691
2692
2693
2694
2695
2696
2697 012626 177777          SP1: .WORD      177777      :IF YOU WANT TO WRITE ANY
2698 012630 052525          SP2: .WORD      052525      :OTHER PATTERNS, CHANGE THESE
2699 012632 111111          SP3: .WORD      111111      :12 LOCATIONS TO ANY PATTERN
2700 012634 010421          SP4: .WORD      010421      :YOU WANT.
2701 012636 102041          SP5: .WORD      102041
2702 012640 010101          SP6: .WORD      010101
2703 012642 040201          SP7: .WORD      040201
2704 012644 000401          SP8: .WORD      000401
2705 012646 031463          SP9: .WORD      031463
2706 012650 070707          SP10: .WORD     070707
2707 012652 007417          SP11: .WORD     007417
2708 012654 041020          SP12: .WORD     041020
2709
2710 012656 004737 016716          EXT10: JSR      PC,ABRT

```


C06

MAINDEC-11-DZRLD-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 51
 DZRLD.F11 31-AUG-76 15:35 T10 WRITE, WRITE CHECK ON CYLINDERS 127, 128

SEQ 0067

711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766

 *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 FORWARD 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

012662	000004			TST11: SCOPE	
012664	032777	000100	166246	BIT #SW6, 3SWR	:INHIBIT TIMER?
012672	001002			BNE +6	
012674	000137	014032		JMP PLTGRPH	
012700	104415			CON.RESET	
012702	104416			DRV.RESET	
012704	012737	177771	001252	MOV #-7, RETRY1	:COUNT FOR 7 DIFRNT SEEK TIMES :TO BE RECORDED
012712	104401	012720		TYPE 65\$	:TYPE ASCIZ STRING
012716	000424			BR 64\$	:GET OVER THE ASCIZ
012770				65\$: .ASCIZ <15><12>/SEEK	TIME SCALE FACTOR=0.01 MILI SECS/
012770	104401	012776		64\$: TYPE 67\$	:TYPE ASCIZ STRING
012774	000421			BR 66\$	:GET OVER THE ASCIZ
013040				67\$: .ASCIZ <15><12><12>/#	OF SEEK # OF SEEK/
013040	104401	013046		66\$: TYPE 69\$	:TYPE ASCIZ STRING
013044	000421			BR 68\$	:GET OVER THE ASCIZ
013110				69\$: .ASCIZ <15><12>/SEEKS	TIME SEEKS TIME/<15><12>
013110	012737	001542	001260	MOV #SIAD, INADR	:INITLZE PTR TO INNER ADRES
013116	012737	001524	001262	MOV #SOAD, OUTADR	:INITLZE PTR TO OUTER ADRES
013124	017777	166132	166332	REPTIM: MOV 3OUTADR, 3RKDA	:POSITION HEADS TO OUTER CYLINDER :BEFORE STARTING TO TIME
013132	053777	001230	166324	BIS DRIVAD, 3RKDA	:SET DRIVE # BITS
013140	012777	000011	166310	MOV #11, 3RKCS	:SEEK, GO
013146	104421			CON.RDY	:WAIT FOR CNTRL RDY
013150	104422			TST.RWS	:WAIT FOR R/W/S RDY
013152	005037	001474		CLR INDX1	:INDX1 = 0, GOING IN; OTHERWISE OUT
013156	012704	026742		MOV #BUFR10, R4	:STORE FWRD SEEK TIMES IN THIS BUFR
013162	012705	027342		MOV #BUFR11, R5	:STORE REVRSE "
013166	012737	177634	001476	MOV #-144, INDX2	:SET COUNT FOR # OF SEEKS
013174	013702	001464		BEGSK: MOV RKDA, R2	

006

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 52
 DZRKLD.P11 31-AUG-76 15:35 T11 SEEK FUNCTION TIMER

SEQ 0068

```

2767 013200 005737 001474      TST      INDX1      :GOING FWRD OR REVRSE?
2768 013204 001005                BNE      1$        :REVRSE, BRANCH
2769
2770 013206 017712 166046      MOV      @INADR,@R2    :FWRD, SET INNER CYL ADRES
2771 013212 053712 001230      BIS      DRIVAD,@R2   :SET DRIVE # BITS
2772 013216 000404                BR      2$
2773
2774 013220 017712 166036      1$: MOV      @OUTADR,@R2  :SET OUTER CYL ADRES
2775 013224 053712 001230      BIS      DRIVAD,@R2   :SET DRIVE # BITS
2776 013230 004737 014722      2$: JSR      PC,@TIMSEK :GO, TIME THE SEEK FROM CYLINDER
2777                                     :0 TO THE ABOVE CYL. RETURN WITH
2778                                     :R3 CONTAINING THE TIME (MS, SCALE
2779                                     :FACTOR= 0.01) REQUIRED FOR THE SEEK.
2780
2780 013234 005737 001474      TST      INDX1
2781 013240 001004                BNE      3$
2782 013242 010324                MOV      R3,(R4)+    :STORE TIME TAKEN FOR FWRD SEEK
2783 013244 005237 001474      INC      INDX1      :SET FLAG FOR DOING REVRSE SEEK
2784 013250 000751                BR      BEGSK      :GO DO IT
2785
2786 013252 010325                3$: MOV      R3,(R5)+    :STORE TIME TAKEN FOR REVRSE SEEK
2787 013254 005037 001474      CLR      INDX1      :CLR FLG FOR DOING FWRD SEEK
2788
2789 013260 005237 001476      INC      INDX2
2790 013264 001343                BNE      BEGSK      :RECORDED 144 SEEK TIMES
2791                                     :IF NOT, GO BAK
2792
2793                                     :AT THIS POINT 100 SEEKS HAVE BEEK PERFORMED BETWEEN TWO
2794                                     :CYLINDERS (FORWARD & REVERSE DIRECTION). FORWARD SEEK
2795                                     :TIMES ARE STORED IN TABLE STARTING AT 'BUFR10' REVERSE
2796                                     :STARTING AT 'BUFR11'. THE FOLLOWING CODE FINDS OUT THE
2797                                     :NUMBERS OF TIMES A PARTICULAR 'SEEK TIME' WAS OBTAINED.
2798                                     :EXAMPLE: OUT OF 100 TIMES THE SEEK WAS DONE BETWEEN
2799                                     :CYLINDER 0 & LAST.
2800                                     :70 TIMES IT TOOK 95 MILI SECS
2801                                     :20 TIMES IT TOOK 85 MILI SECS
2802                                     :10 TIMES IT TOOK 100 MILI SECS
2803                                     :THIS INDICATES HOW CONSISTENT WAS THE SEEKING TIME.
2804                                     :NOTE THAT ONLY 5 DIFFERENT "SEEK TIMES" WILL BE TYPED
2805                                     :OUT.
2806
2807                                     ;SORTING ROUTINE
2808
2808 013266 012705 177776      SORT: MOV      #-2,R5      :COUNT FOR FWRD, REVRSE
2809 013272 012737 026742 001162  MOV      @BUFR10,$REG0 :INTLZE PTR TO 'SEEK TIME'
2810 013300 012737 026744 001164  MOV      @BUFR10+2,$REG1
2811 013306 012737 026442 001166  MOV      @BUFR4,$REG2
2812 013314 012737 026522 001170  MOV      @BUFR5,$REG3 :INTLZE PTR TO '# OF TIMES'
2813
2814 013322 013700 001162      1$: MOV      $REG0,R0      :PTR T 'SEEK TIME'
2815 013326 013701 001164      MOV      $REG1,R1
2816 013332 012702 177635      MOV      #-143,R2    :COUNT FOR 143 ITEMS TO SORT
2817 013336 005003                CLR      R3
2818
2819 013340 021011                2$: CMP      (R0),(R1)    :SORT THE ITEMS & PUT THEM
2820 013342 003404                BLE      3$          :IN DESCENDING ORDER
2821 013344 011004                MOV      (R0),R4    :LARGER ITEMS AT TOP OF LIST,
2822 013346 011110                MOV      (R1),(R0)    :SMALLER AT THE BOTTOM

```


E06

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 53
 DZRKLD.P11 31-AUG-76 15:35 T11 SEEK FUNCTION TIMER

SEQ 0069

```

2823 013350 010411      MOV      R4,(R1)
2824 013352 005203      INC      R3
2825 013354 005720      3$:    TST      (R0)+
2826 013356 005721      TST      (R1)+
2827 013360 005202      INC      R2
2828 013362 001366      BNE      2$
2829 013364 005703      TST      R3
2830 013366 001355      BNE      1$      ;SORTED ALL ITEMS?
                        ;IF NOT LOOP BACK
2831
2832 013370 013700 001162      MOV      $REG0,R0      ;PTR TO 'SEEK TIME'
2833 013374 013701 001166      MOV      $REG2,R1      ;SAVE 'SEEK TIME' HERE
2834 013400 013702 001170      MOV      $REG3,R2      ;SAVE '# OF TIMES' HERE
2835 013404 010204      MOV      R2,R4
2836 013406 005024      CLR      (R4)+
2837 013410 005024      CLR      (R4)+
2838 013412 005024      CLR      (R4)+
2839 013414 005024      CLR      (R4)+
2840 013416 005024      CLR      (R4)+
2841 013420 012703 177773      MOV      #-5,R3      ;SAVE ONLY 5 DIFRNT 'SEEK TIMES'
2842
2843 013424 011011      MOV      (R0),(R1)      ;FIND OUT THE '# OF TIMES'
2844 013426 012703 177634      MOV      #-144,R3      ;EACH 'SEEK TIME' WAS
2845 013432 022011      4$:    CMP      (R0)+,(R1)
2846 013434 001411      BEQ      5$      ;OBTAINED
2847
2848 013436 005721      TST      (R1)+
2849 013440 016011 177776      MOV      -2(R0),(R1)      ;SAVE 'SEEK TIME'
2850 013444 005722      TST      (R2)+
2851 013446 012712 000001      MOV      #1,(R2)      ;KEEP '# OF TIMES'
2852 013452 005203      INC      R3
2853 013454 001404      BEQ      6$
2854 013456 000765      BR      4$
2855
2856 013460 005212      5$:    INC      (R2)      ;INCRMNT '# OF TIMES'
2857 013462 005203      INC      R3      ;ALL DONE?
2858 013464 001362      BNE      4$      ;IF NOT, GO BAK
2859
2860 013466 005205      6$:    INC      R5      ;SORTED BOTH FRWRD, REVRSE
2861 013470 001415      BEQ      GOTYPE      ;'SEEK TIMES', IF YES GO TYPE
2862
2863 013472 012737 027342 001162      MOV      #BUFR11,$REG0      ;IF NOT, INITLZE PTR TO 'SEEK TIME'
2864 013500 012737 027344 001164      MOV      #BUFR11+2,$REG1
2865 013506 012737 026602 001166      MOV      #BUFR6,$REG2      ;SAVE 'SEEK TIME'
2866 013514 012737 026662 001170      MOV      #BUFR7,$REG3      ;SAVE '# OF TIMES' HERE
2867
2868 013522 000677      BR      1$      ;GO BAK & DO SORTING FOR REVRSE SEEK TIMES
2869
2870
2871
2872 013524 104401      GOTYPE: TYPE
2873 013526 001213      $CRLF
2874 013530 104401 013536      TYPE      65$
2875 013534 000403      BR      64$      ;:TYPE ASCIZ STRING
2876
2877 013544      ;:65$: .ASCIZ /CYLS:/
2878      64$:

```

F06

MAINDEC-11-DZRLK-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 54
 DZRLKLD.P11 31-AUG-76 15:35 T11 SEEK FUNCTION TIMER

SEQ 0070

```

2879 013544 017700 165512      MOV      @OUTADR,RO      ;GET OUTER CYL #
2880 013550 006200      ASR      RO
2881 013552 006200      ASR      RO
2882 013554 006200      ASR      RO
2883 013556 006200      ASR      RO
2884 013560 006200      ASR      RO
2885 013562 010046      MOV      RO,-(SP)
2886 013564 104424      TYPDSS
2887 013566 104401 013574      TYPE      67$      ;TYPE IT OUT IN DECIMAL
2888 013572 000401      BR        66$      ;TYPE ASCIZ STRING
2889                      ;:67$: .ASCIZ  /- /      ;GET OVER THE ASCIZ
2890 013576                      66$:
2891 013576 017701 165456      MOV      @INADR,R1      ;GET INNER CYL #
2892 013602 006201      ASR      R1
2893 013604 006201      ASR      R1
2894 013606 006201      ASR      R1
2895 013610 006201      ASR      R1
2896 013612 006201      ASR      R1
2897 013614 010146      MOV      R1,-(SP)
2898 013616 104424      TYPDSS
2899 013620 104401 013626      TYPE      69$      ;TYPE IT OUT IN DECIMAL
2900 013624 000405      BR        68$      ;TYPE ASCIZ STRING
2901                      ;:69$: .ASCIZ <15><12>/ FRWRD/      ;GET OVER THE ASCIZ
2902 013640                      68$:
2903 013640 104401 002101      TYPE      ,BLNKS9
2904 013644 104401 013652      TYPE      71$      ;TYPE ASCIZ STRING
2905 013650 000404      BR        70$      ;GET OVER THE ASCIZ
2906                      ;:71$: .ASCIZ /REVRSE/
2907 013662                      70$:
2908
2909                      ;TYPE OUT THE '# OF SEEKS' & 'SEEK
2910                      ;TIME' OBTAINED FOR EACH OF THOSE
2911                      ;SEEKS
2912 013662 005000      TYPTIM: CLR      RO
2913 013664 005005      CLR      R5
2914 013666 104401      1$: TYPE
2915 013670 001213      $CRLF
2916 013672 016046 026522      MOV      BUFR5(RO),-(SP) ;GET '# OF SEEKS', IF NONE (0)
2917 013676 001424      BEQ      3$      ;SKIP TYPING (FRWRD SEEK)
2918 013700 104405      TYPDS      ;GO TYPE OUT DECIMAL '# OF SEEKS'
2919 013702 104401      TYPE
2920 013704 002110      BLNKS2
2921 013706 016046 026442      MOV      BUFR4(RO),-(SP) ;GET 'SEEK TIME' FOR EACH OF
2922 013712 104405      TYPDS      ;OF THAT '# OF SEEKS'. 'GO
2923                      ;TYPE OUT IN DECIMAL
2924
2925 013714 016046 026662      2$: MOV      BUFR7(RO),-(SP) ;GET '# OF SEEKS', IF NONE (0)
2926 013720 001416      BEQ      4$      ;SKIP TYPING (REVRSE SEEK)
2927 013722 005705      TST      R5
2928 013724 001402      BEQ      6$
2929 013726 104401 002075      TYPE      ,BLNK13
2930 013732 104405      TYPDS      ;TYPE OUT IN DECIMAL
2931 013734 104401      TYPE
2932 013736 002110      BLNKS2
2933 013740 016046 026602      MOV      BUFR6(RO),-(SP) ;GET 'SEEK TIME' & TYPE IT
2934 013744 104405      TYPDS      ;OUT IN DECIMAL

```


G06

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 55
 DZRKLD.P11 31-AUG-76 15:35 T11 SEEK FUNCTION TIMER

SEQ 0071

```

2935 013746 000406 BR 5$
2936
2937 013750 005726 3$: TST (SP)+ ;POP STACK
2938 013752 005205 INC R5
2939 013754 000757 BR 2$
2940
2941 013756 005726 4$: TST (SP)+ ;POP STACK
2942 013760 005705 TST R5
2943 013762 001004 BNE TIMDON
2944
2945 013764 005720 5$: TST (R0)+ ;INCREMENT PTR TO TABLES
2946 013766 020027 000012 CMP R0,#12 ;ALL DONE?
2947 013772 001335 BNE 1$ ;IF NOT GO BAK
2948
2949 013774 062737 000002 001260 TIMDON: ADD #2,INADR ;INCRMNT POINTER TO NEXT
2950 014002 062737 000002 001262 ADD #2,OUTADR ;INNER & OUTER ADRES
2951 014010 005237 001252 INC RETRY1 ;ALL DONE?
2952 014014 001406 BEQ PLTGRAPH
2953 014016 032777 000100 165114 BIT #SW6,SWR ;INHIBIT TIMER? FURTHER ?
2954 014024 001402 BEQ PLTGRAPH ;YES, BRANCH
2955 014026 000137 013124 JMP REPTIM ;GO, BACK AND TIME REST
2956 ;OF SEEKS
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967

```

;PLOT GRAPH OF 'SEEK TIME' V/S 'CYLIDERS SEEKED'

;PERFORM SEEK FROM CYLINDER 0 TO EVERY OTHER CYLINDER AND TIME IT.
 ;0 0.0 1.----0 312. NOTE 'SECTOR COUNTER' IS USED AS A READ
 ;TIME CLOCK TO TIME THERE SEEKS. AFTER OBTAINING THE SEEK TIMES A
 ;GRAPH IS PLOTTED OF 'SEEK TIME' V/S 'CYLINDER'.

```

2968 014032 032777 000040 165100 PLTGRPH: ;TIME THE SEEKS
2969 014040 001002 BNE BIT #SW5,SWR ;SKIP THE GRAPH?
2970 014042 000137 015150 JMP TST12 ;YES, BRANCH
2971 014046 104415 CON.RESET
2972 014050 104416 DRV.RESET
2973 014052 012737 177465 001500 MOV #-313,INDX3 ;PERFORM 313 SEEKS 0-0,0-1,0-312
2974 014060 012704 026742 MOV #BUF10,R4 ;STORE 'SEEK TIME' HERE
2975 014064 005037 001260 CLR INADR ;CLR CYL ADRES BITS
2976
2977 014070 013777 001260 165366 1$: MOV INADR,ARKDA ;ADRES THE RIGHT CYLINDER
2978 014076 053777 001230 165360 BIS DRIVAD,ARKDA ;ADRES THE RIGHT DRIVE
2979
2980 014104 004737 014722 JSR PC,TIMSEK ;GO TIME THE SEEK FROM CYL 0
2981 ;TO THE ABOVE CYL. RETURN WITH
2982 ;R3 CONTAINING 'SEEK TIME' IN MS
2983 ;SCALE FACTOR OF 0.01
2984 014110 010324 MOV R3,(R4)+ ;STORE 'SEEK TIME'
2985 014112 042777 017777 165344 BIC #17777,ARKDA ;SEEK BACK TO CYL 0 FOR
2986 014120 012777 000011 165330 MOV #11,ARKCS ;TIMING NXT CYL SEEK
2987 014126 104421 CON.RDY ;WAIT FOR CNTRL RDY?
2988 014130 104422 TST.RWS ;WAIT FOR R/W/S RDY
2989 014132 062737 000040 001260 ADD #40,INADR ;FORM NXT CYL ADRES
2990 014140 005237 001500 INC INDX3

```

H06

MAINDEC-11-DZRLD-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 56
 DZRLD.P11 31-AUG-76 15:35 T11 SEEK FUNCTION TIMER

SEQ 0072

```

2991 014144 001351      BNE      1$
2992
2993      ;PLOT A GRAPH USING 'SEEK TIMES' RECORDED BEFORE
2994
2995 014146
2996 014146 104401 014154  PLOT:
2997 014152 000422      TYPE      65$      ;:TYPE ASCIZ STRING
2998      ;:65$: .ASCIZ <15><12><12><12>/X AXIS - SEEK TIME - MILI SECS/
2999      64$:
3000 014220 104401 014226      TYPE      67$      ;:TYPE ASCIZ STRING
3001 014224 000423      BR      66$      ;:GET OVER THE ASCIZ
3002      ;:67$: .ASCIZ <15><12>/Y AXIS - CYLINDER SEEKED FROM 0/<15><12><12>
3003      66$:
3004
3005 014274 104401      TYPE
3006 014276 002103      BLNKS7
3007 014300 005000      CLR      RO
3008 014302 010046      1$: MOV      RO,-(SP)      ;:TYPE OUT THE TIME UNITS
3009 014304 104424      TYPDSS      ;:(MILI SECS) FOR THE X-AXIS
3010 014306 005700      TST      RO      ;:LIKE THIS:
3011 014310 001411      BEQ      2$      ;:0 20 30 40.....
3012 014312 022700 000144      CMP      #144,RO
3013 014316 003010      BGT      4$
3014 014320 022700 000170      CMP      #170,RO
3015 014324 002412      BLT      5$
3016 014326 104401      TYPE
3017 014330 002110      BLNKS2
3018 014332 000404      BR      3$
3019 014334 104401      2$: TYPE
3020 014336 002111      BLNKS1
3021 014340 104401      4$: TYPE
3022 014342 002107      BLNKS3
3023 014344 062700 000012      3$: ADD      #12,RO
3024 014350 000754      BR      1$
3025
3026 014352 104401      5$: TYPE
3027 014354 001213      $CRLF
3028 014356 104401      TYPE
3029 014360 002103      BLNKS7
3030
3031 014362 012700 177763  PLT1: MOV      #-15,RO      ;:TYPE OUT THE X-AXIS MARKERS
3032 014366      1$:
3033 014366 104401 014374      TYPE      65$      ;:TYPE ASCIZ STRING
3034 014372 000403      BR      64$      ;:GET OVER THE ASCIZ
3035      ;:65$: .ASCIZ /I-----/
3036      64$:
3037 014402 005200      INC      RO
3038 014404 001370      BNE      1$      ;:I-----I-----I-----
3039
3040      ;IF SW 4 IS SET THEN TYPE THE COMPLETE GRAPH. IF NOT TYPE THE SMALL GRAPH.
3041
3042 014406 032777 000020 164524  BIT      #SW4,2SWR      ;:TYPE COMPLETE GRAPH?
3043 014414 001054      BNE      CMPGRP      ;:YES BRANCH
3044      ;:IF NOT, TYPE SMALL GRAPH
3045
3046

```


MAINDEC-11-DZRLD-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 57
 DZRLD.P11 31-AUG-76 15:35 T11 SEEK FUNCTION TIMER

SEQ 0073

```

3047
3048 014416 005000
3049 014420 032777 000040 164512 SMGRP: CLR RO
3050 014426 001445 1$: BIT #SW5,DSWR ;SKIP REST OF GRAPH?
3051 014430 104401 BEQ 5$ ;YES
3052 014432 001213 TYPE ;IN THIS GRAPH SEEK TIMES ARE
3053 $CRLF ;PLOTTED ONLY FOR SELECTED
3054 ;CYLINDERS (NOT ALL) SHOWN BELOW:
3055 ;0,1,2,3,4, 6,8,10,12,14,16,18,20,
3056 ;25,30,35,....,190,195,200, 203
3057 014434 010046 MOV RO,-(SP) ;TYPE THE MARKERS
3058 014436 104405 TYPDS
3059 014440 104401 014446 TYPE ;TYPE ASCIZ STRING
3060 014444 000401 BR 65$ ;GET OVER THE ASCIZ
3061 65$: .ASCIZ /- /
3062 014450 010001 MOV RO,R1 ;FORM THE ADRES OF 'SEEK TIME'
3063 014452 006301 ASL R1
3064 014454 016103 026742 MOV BUFR10(R1),R3 ;GET THE SEEK TIME
3065 014460 004737 014662 JSR PC,PLTPT ;GO PLOT IT
3066 014464 022700 000004 CMP #4,R0 ;PLOTTED UPTO CYL 4?
3067 014470 003402 BLE 2$ ;YES
3068 014472 005200 INC RO
3069 014474 000751 BR 1$
3070 014476 022700 000024 2$: CMP #24,R0 ;PLOTTED UPTO CYL 20?
3071 014502 003403 BLE 3$
3072 014504 062700 000002 ADD #2,R0
3073 014510 000743 BR 1$
3074 014512 022700 000310 3$: CMP #310,R0 ;PLOTTED UPTO CYL 200?
3075 014516 003403 BLE 4$
3076 014520 062700 000005 ADD #5,R0
3077 014524 000735 BR 1$
3078 014526 022700 000312 4$: CMP #312,R0 ;PLOTTED ALL CYLS?
3079 014532 001403 BEQ 5$
3080 014534 062700 000002 ADD #2,R0
3081 014540 000727 BR 1$
3082 014542 000137 015150 5$: JMP TST12
3083
3084
3085 ;IF SW 4 IS SET THE COMPLETE GRAPH IS PRINTED OUT. IT GIVES TIMES FOR
3086 ;SEEKS FROM CYLINDER 0 TO ALL OTHER CYLINDERS (0,1,2,3...202).
3087
3088 014546 005000 CMPGRP: CLR RO ;INITLZE COUNT
3089 014550 012701 177773 MOV #-5,R1 ;INITLZE COUNT FOR Y-AXIS MARKER
3090 014554 012702 026742 MOV #BUFR10,R2 ;INITLZE PTR TO SEEK TIMES
3091 014560 104401 TYPE
3092 014562 001213 $CRLF
3093 014564 000412 BR 3$
3094
3095 014566 032777 000040 164344 2$: BIT #SW5,DSWR ;SKIP REST OF GRAPH?
3096 014574 001002 BNE +6
3097 014576 000137 015150 JMP TST12
3098 014602 005201 INC R1 ;TYPE OUT Y-AXIS MARKER 'CYL #'
3099 014604 001005 BNE 4$ ;IF REQUIRED
3100 014606 012701 177773 MOV #-5,R1
3101 014612 010046 3$: MOV RO,-(SP) ;TYPE 'CYL #' ON Y-AXIS
3102 014614 104405 TYPDS ; (IN DECIMAL)

```

J06

MAINDEC-11-DZKRL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 58
 DZKRLD.P11 31-AUG-76 15:35 T11 SEEK FUNCTION TIMER

SEQ 0074

3103 014616 000402
 3104 014620 104401
 3105 014622 002104
 3106 014624
 3107 014624 104401 014632
 3108 014630 000401
 3109
 3110 014634
 3111
 3112 014634 012203
 3113 014636 004737 014662
 3114
 3115
 3116 014642 104401
 3117 014644 001213
 3118 014646 005200
 3119 014650 022700 000312
 3120 014654 001344
 3121 014656 000137 015150
 3122
 3123
 3124
 3125
 3126
 3127
 3128
 3129
 3130
 3131
 3132
 3133
 3134
 3135 014662 162703 000310
 3136 014666 002403
 3137
 3138 014670 104401
 3139 014672 002111
 3140 014674 000772
 3141 014676 062703 000144
 3142 014702 002402
 3143 014704 104401
 3144 014706 002111
 3145
 3146 014710
 3147 014710 104401 014716
 3148 014714 000401
 3149
 3150 014720
 3151 014720 000207
 3152
 3153
 3154
 3155
 3156
 3157
 3158

BR 5\$
 4\$: TYPE
 BLNKS6
 5\$:
 TYPE 65\$
 BR 64\$
 65\$: .ASCIZ /-/
 64\$:
 MOV (R2)+,R3
 JSR PC,PLTPT ;GO PLOT THE POINT
 ;GET SEEK TIME
 TYPE
 \$CRLF
 INC R0
 CMP #312,R0 ;ALL DONE?
 BNE 2\$;IF NOT, GO BAK
 6\$: JMP TST12
 ;PLTPT
 ;THE ROUTINE IS ENTERED WITH R3 CONTAINING HORIZONTAL AXIS
 ;COORDINATE- SEEK TIME
 ;PLOT THE ACTUAL TIME ON THE GRAPH. IN KEEPING WITH NORMAL
 ;CONVENTION A NUMBER LESS THAN HALF THE CELL WIDTH IS
 ;CONSIDERED AS FALLING UNDER THE PREVIOUS CELL, A NUMBER
 ;GREATER THAN OR EQUAL TO HALF THE CELL WIDTH FALLS UNDER THE NEXT CELL
 ;EX: IF SEEK TIME IS 11.5 MS, IT'S BETW'N 10 & 12, BUT > 11
 ;HENCE IT WILL BE PLOTTED AS 12 IF SEEK TIME IS 10.8 MS,
 ;IT'S BETW'N 10 & 12, BUT < 11 HENCE IT WILL BE PLOTTED
 ;AS 10.0 MS
 PLTPT: SUB #310,R3 ;FIND OUT HOW MANY BLANKS TO
 BLT 7\$;INSERT TO PLOT THE POINT
 ;NOTE THE FIRST CELL = 0 MS
 TYPE
 BLNKS1
 BR PLTPT
 7\$: ADD #144,R3
 BLT 8\$
 TYPE
 BLNKS1
 8\$:
 TYPE 65\$
 BR 64\$
 65\$: .ASCIZ /X/
 64\$:
 RTS PC
 ;TIMSEK
 ;THIS ROUTINE FINDS OUT THE TIME REQUIRED TO SEEK TO THE CYLINDER
 ;INDICATED IN RKDA.
 ;***CAUTION*** SECTOR COUNTER IS USED AS A REAL TIME CLOCK TO KEEP TIME.
 ;ENTRY: JSR PC,TIMSEK
 ; RKDA CONTAINS THE CYLINDER TO BE SEEKED FROM THE PRESENT POSITION.

K06

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 59
 DZRKLD.P11 31-AUG-76 15:35 T11 SEEK FUNCTION TIMER

SEQ 0075

;RETURN: R3 CONTAINS THE SEEK TIME IN MILI SECS. SCALE FACTOR = 0.01

3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214

014722 010246
014724 005003
014726 013701 001452
014732 011102
014734 032702 000400
014740 001774
014742 032702 000010
014746 001771
014750 011102
014752 032702 000400
014756 001774
014760 021102
014762 001372
014764 032702 000017
014770 001367
014772 012777 000011 164456
015000 104421
015002 011102
015004 032702 000400
015010 001774
015012 020211
015014 001372
015016 032702 000100
015022 001025
015024 032702 000017
015030 001764
015032 011102
015034 032702 000400
015040 001774
015042 020211
015044 001372
015046 032702 000100
015052 001005
015054 032702 000017
015060 001364
015062 005203
015064 000746
015066 032702 000017
015072 001001
015074 005203

TIMSEK: MOV R2, -(SP)
 CLR R3
 MOV RKDS, R1
 1\$: MOV @R1, R2
 BIT #400, R2
 BEQ 1\$
 BIT #BIT3, R2
 BEQ 1\$
 2\$: MOV @R1, R2
 BIT #400, R2
 BEQ 2\$
 CMP @R1, R2
 JNE 2\$
 BIT #17, R2
 BNE 2\$
 MOV #11, @RKCS
 CON. RDY
 3\$: MOV @R1, R2
 BIT #400, R2
 BEQ 3\$
 CMP R2, @R1
 BNE 3\$
 BIT #100, R2
 BNE SKDON
 BIT #17, R2
 BEQ 3\$
 4\$: MOV @R1, R2
 BIT #400, R2
 BEQ 4\$
 CMP R2, @R1
 BNE 4\$
 BIT #100, R2
 BNE 5\$
 BIT #17, R2
 BNE 4\$
 INC R3
 BR 3\$
 5\$: BIT #17, R2
 BNE SKDON
 INC R3

;R3 WILL COUNT REVOLUTIONS OF
 ;DISK (FROM INDEX MARK TO INDEX MARK)
 ;40 MILI SECS FOR EACH REV

;WAIT FOR SOK

;WAIT FOR SECTOR 10 SO THAT
 ;U CAN START WAITING FOR
 ;INDEX, SEC 0

;WAIT FOR SEC OK

;WAIT FOR SEC 0, INDEX MARK
 ;AS SOON AS IT IS SEC 0, ISSUE
 ;A SEEK & START TIMING

;ISSUE A SEEK, START TIMING
 ;THE SEC COUNTER
 ;WAIT FOR CNTRL RDY

;GET RKDS
 ;WAIT FOR SOK

;INFO CORRECT?
 ;NO
 ;R/W/S RDY SET?
 ;IF YES, BRANCH
 ;WAIT FOR SEC CNTR TO MOVE
 ;FROM 0 TO 1

;WAIT FOR SOK

;R/W/S RDY SET, SEEK DONE?
 ;YES, BRANCH
 ;IF NOT KEEP TRACK OF SEC
 ;COUNTER. INCREMENT R3 AT
 ;EVERY INDEX MARK, EVERY
 ;40 MILI SECS
 ;GO BAK, KEEP TIME

;CHECK, IS IT INDEX MARK -SEC 0
 ;IF NOT, SKIP
 ;IF YES, INCREMENT COUNT

;SEEK DONE, SAVE RKDS-SEC COUNTER.

L06

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 60
 DZRKLD.P11 31-AUG-76 15:35 T11 SEEK FUNCTION TIMER

SEQ 0076

```

3215 015076
3216 015076 012746 000014
3217 015102 010346
3218 015104 004737 020414
3219 015110 012616
3220 015112 012603
3221 015114 042702 177760
3222 015120 060203
3223
3224 015122 012746 000512
3225 015126 010346
3226 015130 004737 020414
3227 015134 012616
3228 015136 012603
3229 015140 062703 000245
3230
3231
3232
3233
3234
3235
3236 015144 012602
3237 015146 000207
3238
3239
3240
3241
3242
3243
3244
3245 015150 000004
3246 015152 005237 001223
3247 015156 123737 001223 001224
3248 015164 001402
3249 015166 000137 003704
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260 015172
3261 015172 000004
3262 015174 005037 001102
3263 015200 005237 001100
3264 015204 042737 100000 001100
3265 015212 005327
3266 015214 000001
3267 015216 003022
3268 015220 012737
3269 015222 000001
3270 015224 015214

SKDON:
MOV #14,-(SP)          ;;PUT THE MULTIPLIER ON THE STACK
MOV R3,-(SP)           ;;PUT THE MULTIPLICAND ON THE STACK
JSR PC,@#SMULT         ;;CALL THE MULTIPLY ROUTINE
MOV (SP)+,(SP)         ;;DISREGARD THE MSB'S
MOV (SP)+,R3           ;;GET THE LSB'S OF THE PRODUCT
BIC #177760,R2         ;SEEK. TOTAL TIME=(IN DECIMAL)
ADD R2,R3              ;[(R3)X12+SEC COUNTER]X330X0.01
                        ;NOTE THERE IS A SCALE FACTOR
MOV #512,-(SP)         ;;PUT THE MULTIPLIER ON THE STACK
MOV R3,-(SP)           ;;PUT THE MULTIPLICAND ON THE STACK
JSR PC,@#SMULT         ;;CALL THE MULTIPLY ROUTINE
MOV (SP)+,(SP)         ;;DISREGARD THE MSB'S
MOV (SP)+,R3           ;;GET THE LSB'S OF THE PRODUCT
ADD #245,R3            ;ASSUMPTION THAT EACH SECTOR
                        ;TAKES 3.3 MILI SECS. IF THE
                        ;DISK SPEED IS VERY MUCH DIFRNT
                        ;FROM THE SPEC SPEED OF
                        ;1500 RPM (40 MS/REV), THEN
                        ;SEC COUNTER WOULD NOT BE AN
                        ;ACCURATE TIME CLOCK.
MOV (SP)+,R2          ;POP R2 BAK
RTS PC                 ;RETURN

;*****
;*TEST 12      END OF PROGRAM
;*THIS IS NOT A TEST BUT IS JUST A LINKAGE
;*PROVIDED TO TEST ALL THE DRIVES.
;*****
;ST12: SCOPE
;INCB DRVDON
BTEOP: CMPB DRVDON,DRIVS
BEQ .+6
JMP NXTDRV

.SBTTL END OF PASS ROUTINE

;*****
;*INCREMENT THE PASS NUMBER ($PASS)
;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
;*IF THERES A MONITOR GO TO IT
;*IF THERE ISN'T JUMP TO ST3

$EOP:
SCOPE
CLR $STNM              ;;ZERO THE TEST NUMBER
INC $PASS              ;;INCREMENT THE PASS NUMBER
BIC #100000,$PASS      ;;DON'T ALLOW A NEG. NUMBER
DEC (PC)+              ;;LOOP?
$EOPCT: .WORD 1
BGT $DOAGN             ;;YES
MOV (PC)+,@(PC)+       ;;RESTORE COUNTER
$ENDCT: .WORD 1
$EOPCT

```


MO6

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 61
DZRKLD.P11 31-AUG-76 15:35 END OF PASS ROUTINE

SEQ 0077

3271	015226	104401	015273	TYPE	\$ENDMG	::TYPE "END PASS #"
3272	015232	013746	001100	MOV	\$PASS, -(SP)	::SAVE \$PASS FOR TYPEOUT
3273	015236	104405		TYPDS		::GO TYPE--DECIMAL ASCII WITH SIGN
3274	015240	104401	015270	TYPE	\$ENULL	::TYPE A NULL CHARACTER
3275	015244	013700	000042	\$GET42: MOV	\$#42, R0	::GET MONITOR ADDRESS
3276	015250	001405		BEQ	\$DOAGN	::BRANCH IF NO MONITOR
3277	015252	000005		RESET		::CLEAR THE WORLD
3278	015254	004710		\$ENDAD: JSR	PC, (R0)	::GO TO MONITOR
3279	015256	000240		NOP		::SAVE ROOM
3280	015260	000240		NOF		::FOR
3281	015262	000240		NOP		::ACT11
3282	015264			\$DOAGN:		
3283	015264	000137		JMP	\$ (PC)+	::RETURN
3284	015266	003666		\$RTNAD: .WORD	ST3	
3285	015270	377	000	\$ENULL: .BYTE	-1, -1, 0	::NULL CHARACTER STRING
3286	015273	015	042412 042116	\$ENDMG: .ASCIZ	<15><12>/END PASS #/	
3287	015300	050040	051501 020123			
3288	015306	000043				
3289						

N06

SEQ 0078

;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 RS 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 'BUFRI'.

;THE PRINTOUT LOOKS LIKE:

;SEC# HDR RECVD
;AA BBBBBA AA=BAD SEC # BBBBBA=BAD HEADER
;EXPCTD HDR=XXXXXX TRY#= Y

ESR15: MOV R1,-(SP) ;:PUSH R1 ON STACK
MOV R2,-(SP) ;:PUSH R2 ON STACK
MOV #BUFR,R1 ;:SEC #'S STORED HERE PREVIOUSLY
MOV #BUFRI,R2 ;:BAD HDRS STORED HERE PRVSLY
1\$: MOV (R1)+,-(SP)
TYP0S ;:GO TYPE OUT BAD SEC # (OCTAL)
.BYTE 2 ;:ONLY 2 DIGITS
.BYTE 0 ;:SUPRES LDG 0'S
TYPE ;:TYPE 3 BLNKS
BLNKS3
MOV (R2)+,-(SP) ;:GO TYPE OUT BAD HEADER
TYP0C
TYPE
BLNKS4
TYPE
\$CRLF
CMP #177777,(R1) ;:ALL BAD SEC #'S TYPD OUT?
BNE 1\$;:IF NOT GO BAK
TYPE
MSG6
MOV R5,-(SP) ;:TYPE OUT EXPCTD HEADER FOR
BIC #160037,(SP) ;:THAT CYLINDER
TYP0C
MOV (SP)+,R2 ;:POP STACK INTO R2
MOV (SP)+,R1 ;:POP STACK INTO R1
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 'BUFRI'. RS CONTAINS THE EXPECTED

3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310 015310 010146
3311 015310 010246
3312 015314 012701 001266
3313 015320 012702 001320
3314 015324 012146
3315 015326 104403
3316 015330 002
3317 015331 000
3318 015332 104401
3319 015334 002107
3320 015336 012246
3321 015340 104402
3322 015342 104401
3323 015344 002106
3324 015346 104401
3325 015350 001213
3326 015352 022711 177777
3327 015356 001362
3328
3329 015360 104401
3330 015362 001716
3331 015364 010546
3332 015366 042716 160037
3333 015372 104402
3334
3335 015374 012602
3336 015376 012601
3337 015400 000207
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400

B07

MAINDEC-11-DZKRL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 63
 DZKRLD.F11 31-AUG-76 15:35 ESR13

SEQ 0079

;HEADER FOR THAT CYLINDER. THE TYPEOUT LOOKS LIKE

```
;SEC# HDR RCVD
;AA      BBBB88      AA=BAD SEC #
;EXPCTD HDR=XXXXXX  BBBB88=BAD HEADER
;TRY#    Y          SUR=Z
```

```
ESR13: JSR      PC,ESR15
        TYPE     65$      ;;TYPE ASCIZ STRING
        BR       64$      ;;GET OVER THE ASCIZ
```

```
;;65$: .ASCIZ  / SUR= /
64$:   CLR      -(SP)
        BIT     #20,R5      ;SUR 0 OR 11?
        BEO     1$
        INC     (SP)
1$:    TYPOC
```

```
        TYPE     MSG13
        MOV      RETRY2,-(SP)
        INC     (SP)
        TYPOC
        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.
```

```
ESR20: JSR      PC,ESR13      ;GO TYPE OUT SEC #'S, BAD HDRS
        JSR      PC,ERR2      ;GET CYL #'S BETWN WHICH SEEK
                                ;WAS TRIED
        TYPE     65$      ;;TYPE ASCIZ STRING
        BR       64$      ;;GET OVER THE ASCIZ
```

```
;;65$: .ASCIZ  / CYLA= /
64$:   MOV      $REGD,-(SP)    ;GO TYPE CYL # FROM WHERE
        TYPOS    ;SEEK BEGAN
        .BYTE    3           ;TYPE 3 DIGITS
        .BYTE    0           ;SUPRES LDG 0'S
        TYPE     67$      ;;TYPE ASCIZ STRING
        BR       66$      ;;GET OVER THE ASCIZ
```

```
;;67$: .ASCIZ  / CYLB= /
66$:   MOV      $REG1,-(SP)    ;TYPE CYL # TO WHICH SEEK
        TYPOS    ;WAS DONE
        .BYTE    3           ;TYPE 3 DIGITS
        .BYTE    0           ;SUPRES LDG 0'S
        RTS      PC          ;RETURN
```

.SBTTL ESR25

```
015402 004737 015310
015406 104401 015414
015412 000404
015424 005046
015426 032705 000020
015430 001401
015434 005216
015436 104402
015440 104401 002053
015444 013746 001254
015450 005216
015454 104402
015458 000207
015456 004737 015402
015462 004737 016244
015466 104401 015474
015472 000404
015504
015504 013746 001162
015510 104403
015512 003
015513 000
015514 104401 015522
015520 000404
015532
015532 013746 001164
015536 104403
015540 003
015541 000
015542 000207
```

C07

 MAINDEC-11-DZAKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 64
 DZAKLD.P11 31-AUG-76 15:35 ESR25

SEQ 0080

3402	015544	010205		ESR25:	MOV	R2,R5	:SAVE ADRES OF TERMINATOR	
3403	015546	012702	001266		MOV	#BUFR,R2	:INITLZE PTR TO TABLE STORING	
3404							:ADRES OF BAD DATA	
3405	015552	012703	001320		MOV	#BUFR1,R3	:INITLZE PTR TO 'EXPCD' DATA	
3406	015556	012704	001352		MOV	#BUFR2,R4	:INITLZE PTR TO 'RECVD' DATA	
3407								
3408	015562	032777	020000	163350	1\$:	BIT	#SW13,3SWR	:INHIBIT TYPE OUT?
3409	015570	001076			BNE	4\$	:YES, EXIT	
3410	015572	104401			TYPE		:TYPE CR,LF	
3411	015574	001213			\$CRLF			
3412								
3413	015576	163712	001404		SUB	PBUFO,(R2)	:GET WORD # IN BUFR (0,1,2...)	
3414	015602	006212			ASR	(R2)		
3415	015604	011246			MOV	(R2),-(SP)	:WHICH WAS BAD. NOTE YOU	
3416							:CAN HAVE THE ACTUAL MEMORY	
3417							:ADRES BY ADDING 'IOBUFO'	
3418							:TO THIS	
3419	015606	104403			TYPOS		:GO TYPE WORD # THAT WAS BAD	
3420	015610	004			.BYTE	4		
3421	015611	000			.BYTE	0		
3422	015612	104401			TYPE			
3423	015614	002107			BLNKS3		:2 BLANKS	
3424								
3425	015616	012346			MOV	(R3)+,-(SP)	:GET EXPCD DATA	
3426	015620	104402			TYPOC		:GO TYPE IT	
3427	015622	104401			TYPE			
3428	015624	002110			BLNKS2			
3429	015626	012446			MOV	(R4)+,-(SP)	:GET RECVD DATA (BAD)	
3430	015630	104402			TYPOC		:GO TYPE IT	
3431	015632	104401			TYPE			
3432	015634	002110			BLNKS2			
3433								
3434	015636	012700	000400		MOV	#400,R0	:GET THE DISK ADRES FROM	
3435	015642	021200		2\$:	CMP	(R2),R0	:WHICH THIS (BAD) DATA WAS	
3436	015644	002405			BLT	3\$	:READ	
3437	015646	062700	000400		ADD	#400,R0		
3438	015650	022700	002400		CMP	#2400,R0		
3439	015656	001371			BNE	2\$		
3440								
3441	015660	000300		3\$:	SWAB	R0		
3442	015662	005300			DEC	R0		
3443	015664	063700	001450		ADD	ADRES,R0	:R0 CONTAINS THE DISK	
3444							:ADRES FROM WHICH THE (BAD)	
3445							:DATA WAS READ	
3446	015670	010037	001170		MOV	R0,\$REG3		
3447								
3448	015674	004737	016144		JSR	PC,BRKDA	:GO BREAK ABOVE DISK ADRES	
3449							:INTO CYL#, SUR#, SEC#	
3450								
3451	015700	013746	001174		MOV	\$REG5,-(SP)	:GET THE CYL#	
3452	015704	104403			TYPOS		:TYPE IT	
3453	015706	003			.BYTE	3	:ONLY 3 DIGITS	
3454	015707	000			.BYTE	0	:NO LEADING 0'S	
3455	015710	104401			TYPE			
3456	015712	002107			BLNKS3			

D07

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 65
 DZRKL.D.P11 31-AUG-76 15:35 ESR25

SEQ 0081

```

015714 013746 001176      MOV    $REG6,-(SP)      ;GET SUR #
015720 104403      TYPOS      ;TYPE
015722 001        .BYTE    1      ;1 DIGIT ONLY
015723 000        .BYTE    0
015724 104401      TYPE
015726 002106      BLNKS4
015730 013746 001200      MOV    $REG7,-(SP)      ;GET SEC#
015734 104403      TYPOS      ;TYPE
015736 002        .BYTE    2      ;2 DIGITS
015737 000        .BYTE    0
015740 005722      TST     (R2)+      ;INCREMNT PTR
015742 020205      CMP     R2,R5      ;TYPED OUT ALL BAD DATA
015744 001306      BNE     1$          ;INFO?
015746 104401      TYPE          ;IF NOT LUP BAK
015750 002053      MSG13
015752 013746 001254      MOV    RETRY2,-(SP)      ;TRY #:
015756 062716 000003      ADD     #3,(SP)      ;GET RETRY COUNT
015762 104403      TYPOS      ;FORM THE RETRY NO.
015764 001        .BYTE    1      ;TYPE IT OUT
015765 000        .BYTE    0
015766 000207      4$:    RTS     PC      ;IF YES, RETURN
                          ;MESSAGE HANDLER
                          ;THE MESSAGE HANDLER IS USED FOR TYPING OUT MESSAGES & DATA
                          ;RELATED TO THE MESSAGE. IF SW13 IS SET, THE TYPEOUT IS
                          ;INHIBITED. THE CALL IS:
                          ;      MESSAGE,XX
                          ;XXX IS THE MESSAGE NUMBER & PROVIDES AN INDEX TO THE
                          ;'ERROR ITEMS TABLE' WHERE THAT MESSAGE ITEM
                          ;IS LOCATED.
                          ;THE MESSAGE ITEM CONTAINS:
                          ;      MS:    POINTER TO THE ASCII MESSAGE
                          ;      DH:    POINTER TO THE DATA HEADER
                          ;      DT:    POINTER TO THE DATA
                          ;      0      TERMINATOR
                          ;IF 'DT' IS 0 THE DATA IS TO BE PRINTED USING THE SUBROUTINE
                          ;INDICATED IN PLACE OF THE TERMINATOR
015770 032777 020000 163142 MSGE:  BIT     #SW13,SWR      ;INHIBIT TYPEOUT?
015776 001012      BNE     1$          ;IF YES, EXIT
016000 011637 001116      MOV    (SP),SERRPC      ;GET ADRES OF 'MESSAGE' CALL
016004 162737 000002 001116      SUB     #2,SERRPC      ;STORE IT
016012 117637 000000 001114      MOVB   2(SP),SITEMB      ;GET MESSAGE # (INDEX TO ITEM TABLE)
016020 004737 017362      JSR     PC,2$ERRTYP      ;GO TO 'ERRTYP' & TYPE OUT
016024 062716 000002      1$:    ADD     #2,(SP)      ;INFO
016030 000002      RTI          ;ADJUST RETURN ADDRES
                          ;EXIT
                          ;THIS ROUTINE IS USED FOR TYPING OUT ASCII MESSAGES. BEFORE
                          ;THE MESSAGE IS TYPED SW13 IS CHECKED & IF SET THE
                          ;TYPEOUT IS INHIBITED & AN EXIT IS MADE.
                          ;THE CALL FOR THIS ROUTINE IS "TYPMSG", AN ENCODED

```

E07

MAINDEC-11-DZAKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 66
 DZAKLD.P11 31-AUG-76 15:35 ESR25

SEQ 0082

```

3514      ;TRAP INSTRUCTION.
3515      ;THE POINTER TO THE ASCII MESSAGE TO BE TYPED IS LOCATED IN THE
3516      ;WORD FOLLOWING THE "TYPMSG" CALL.
3517
3518 016032 032777 020000 163100 TY.MSG: BIT      #SW13,DSWR      ;INHIBIT TYPEOUT?
3519 016040 001005                BNE      2$                ;YES, EXIT
3520 016042 017637 000000 016052      MOV      2(SP),1$      ;GET POINTER TO ASCII MESSAGE
3521 016050 104401                TYPE                     ;GO TYPE ASCII STRING
3522 016052 000000      1$:      0
3523 016054 062716 000002      2$:      ADD      #2,(SP)      ;ADJUST RETURN ADRES, SKIP OVER
3524                                ;POINTER ON RETURN
3525 016060 000002                                RTI          ;EXIT
3526
3527
3528
3529
3530
3531
3532
3533      ;GT5RG
3534      ;THIS ROUTINE EXTRACTS THE CYLINDER # FROM RKDA AND STORES IT
3535      ;IN $REG4. THEN TRANSFERS RKCS, ER, DS, DA TO $REG0, $REG1, $REG2, $REG3
3536 016062 017746 163376      GT5RG: MOV      2RKDA,-(SP)      ;PUSH RKDA ONTO STACK
3537 016066 042716 160037      BIC      #160037,(SP)      ;MASK OUT NON-CYLINDER BITS
3538 016072 006316      ASL      (SP)      ;SHIFT 8 BITS OF CYL ADRES TO LO BYTE
3539 016074 006316      ASL      (SP)
3540 016076 006316      ASL      (SP)
3541 016100 000316      SWAB      (SP)
3542 016102 112637 001172      MOVB      (SP)+,$REG4      ;UP STACK
3543
3544
3545
3546
3547
3548
3549
3550      ;GT4RG
3551      ;THIS ROUTINE TRANSFERS THE CONTENTS OF RKCS, RKER, RKDS
3552      ;RKDA TO $REG0, $REG1, $REG2, $REG3 RESPECTIVELY. $REG'S
3553      ;ARE USED FOR TYPING OUT THERE CONTENTS AT THE TIME OF ERROR
3554 016106 017737 163344 001162      GT4RG: MOV      2RKCS,$REG0      ;GET RKCS
3555 016114 017737 163334 001164      MOV      2RKER,$REG1      ;RKER
3556 016122 017737 163324 001166      MOV      2RKDS,$REG2      ;RKDS
3557 016130 017737 163330 001170      MOV      2RKDA,$REG3      ;RKDA
3558 016136 000207      RTS      PC      ;EXIT FROM THIS ROUTINE
3559
3560
3561
3562
3563
3564
3565
3566      ;GETINF
3567      ;THIS ROUTINE SAVES THE CONTENTS OF RKCS IN $REG0
3568      ;RKER IN $REG1, RKDS IN $REG2. THEN IT BREAKS RKDA
3569      ;INTO DRIVE NO, CYLINDER #, SURFACE AND SECTOR #.
3570      ;AND SAVES THEM IN $REG4, $REG5, $REG6, $REG7.
3571 016140 004737 016106      GETINF: JSR      PC,GT4RG
3572 016144 010046      BRKDA: MOV      R0,-(SP)
3573 016146 010146      MOV      R1,-(SP)
3574 016150 010246      MOV      R2,-(SP)
3575 016152 012700 001202      MOV      #2,$REG7+2,R0
3576 016156 013701 001170      MOV      $REG3,R1
3577 016162 010102      MOV      R1,R2
3578 016164 042702 177760      BIC      #177760,R2
3579 016170 010240      MOV      R2,-(R0)
3580 016172 006201      ASR      R1

```


F07

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 67
 DZRKLD.P11 31-AUG-76 15:35 ESR25

SEQ 0083

3570	016174	006201		ASR	R1
3571	016176	006201		ASR	R1
3572	016200	006201		ASR	R1
3573	016202	010102		MOV	R1,R2
3574	016204	042702	177776	BIC	#177776,R2
3575	016210	010240		MOV	R2,-(R0)
3576	016212	006201		ASR	R1
3577	016214	010102		MOV	R1,R2
3578	016216	042702	177400	BIC	#177400,R2
3579	016222	010240		MOV	R2,-(R0)
3580	016224	000301		SWAB	R1
3581	016226	042701	177770	BIC	#177770,R1
3582	016232	010140		MOV	R1,-(R0)
3583	016234	012602		MOV	(SP)+,R2
3584	016236	012601		MOV	(SP)+,R1
3585	016240	012600		MOV	(SP)+,R0
3586	016242	000207		RTS	PC

.SBTTL ERR2

```

:ERR2
:THIS ROUTINE GETS THE CYLINDER NUMBERS BETWEEN WHICH (IMPLIED) SEEK
:WAS DONE. (R4)=0 INDICATES SEEK FROM 'OUTADR' TO 'INADR'
:(R4)=1 INDICATES SEEK TO 'OUTADR'. ON EXIT $REG0 CONTAINS CYL #
:FROM WHICH SEEK WAS INITIATED, $REG1 CONTAINS CYL # TO WHICH SEEK WAS DONE
ERR2:  MOV    INADR,$REG0      ;GET CYL ADRES
        JSR    PC,GCYL        ;GO GET CYL# FROM IT
        MOV    $REG0,$REG1    ;SAVE
        MOV    OUTADR,$REG0   ;GET CYL ADRES
        JSR    PC,GCYL        ;GO GET CYL # FROM IT
        TST    R4             ;GOING WHICH WAY?
        BEQ    IS             ;'OUTADR' TO 'INADR', BRANCH
        MOV    $REG0,-(SP)     ;EXCHANG CYL" TO GET
        MOV    $REG1,$REG0     ;CORRECT 'TO' & 'FROM' CYLS
        MOV    (SP)+,$REG1
        IS:    RTS            PC      ;RETURN

```

.SBTTL ERR1

```

:ERR1
:THIS SUBROUTINE FINDS OUT THE CYLINDER NOS. BETWEEN WHICH THE SEEK
:IS DONE. THE CYLINDER # WHERE THE HEADS WHERE PRIOR TO MOVING, IS
:DEPOSITED IN $REG0. THE CYLINDER # WHERE THE HEADS SHOULD BE AFTER
:MOVEMENT, IS DEPOSITED IN $REG1. R4 INDICATES WHICH DIRECTION THE
:HEADS WERE MOVING, IN OR OUT. R5 CONTAINS THE
:DISK ADDRESS (IN OR OUT AS THE CASE MAY BE).

```

3620	016322	010537	001162	ERR1:	MOV	R5,\$REG0	
3621	016326	004737	016360		JSR	PC,GCYL	;GO GET CYL #
3622	016332	005704			TST	R4	;WAS GOING IN OR OUT?
3623	016334	001006			BNE	IS	;OUT
3624	016336	013737	001162	001164	MOV	\$REG0,\$REG1	
3625	016344	005037	001162		CLR	\$REG0	

G07

MAINDEC-11-DZRLD-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 68
 DZRLD.P11 31-AUG-76 15:35 ERR1

SEQ 0084

```

3626 016350 000207
3627 016352 005037 001164
3628 016356 000207
3629
3630
3631
3632
3633
3634
3635 016360 010046
3636 016362 013700 001162
3637 016366 042700 160037
3638
3639 016372 006200
3640 016374 006200
3641 016376 006200
3642 016400 006200
3643 016402 006200
3644 016404 010037 001162
3645 016410 012600
3646 016412 000207
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658 016414 005037 001174
3659 016420 013777 001230 163036
3660 016426 012777 000015 163022
3661 016434 104421
3662 016436 000402
3663 016440 005037 001174
3664 016444 032777 000100 163000
3665 016452 001024
3666 016454 012746 177770
3667 016460 005216
3668 016462 001376
3669 016464 005726
3670 016466 005237 001174
3671 016472 001364
3672 016474 032777 020000 162436
3673 016502 001010
3674 016504 104420
3675 016506 002027
3676 016510 104420 001733
3677 016514 011646
3678 016516 162716 000002
3679 016522 104402
3680 016524 000002
3681

      RTS      PC
      CLR      $REG1
      RTS      PC

      .SBTTL GCYL
      :GCYL
      :THIS ROUTINE EXTRACTS THE CYLINDER NO. FROM THE DISK ADDRESS
      :CONTAINED IN '$REG0' AND THEN STORES IT BACK IN '$REG0'
      GCYL: MOV      RO, -(SP)      ;PUSH RO ONTO STACK
            MOV      $REG0, RO
            BIC      #160037, RO    ;MASK OUT DRV # BITS &
            ;SUR, SEC BITS IF PRESENT
            ASR      RO
            ASR      RO
            ASR      RO
            ASR      RO
            ASR      RO
            MOV      RO, $REG0      ;STORE CYL # IN $REG0
            MOV      (SP)+, RO      ;POP RO FROM STACK
            RTS      PC
            ;EXIT

      .SBTTL DRV.RESET - DRIVE RESET ROUTINE
      .SBTTL RESDON - WAIT FOR DRIVE RESET TO BE DONE
      :DR.RST
      :THIS ROUTINE DOES A DRIVE RESET ON THE DRIVE WHOOSE ADDRESS IS IN
      :RKDA. MULTIPLE RETURN ADDRESSES FOR THIS ROUTINE ARE PROVIDED.
      :IF THERE IS NO ERROR (R/W/S RDY SETS WITHIN CERTAIN TIME) THEN
      :A NORMAL EXIT IS MADE. IF R/W/S RDY DOES NOT SET ERROR IS REPORTED.

      DR.RST: CLR      $REG5      ;INITIALIZE THE COUNT
            MOV      DRIVAD, $RKDA
            MOV      #15, $RKCS    ;DRIVE RESET, GO
            CON.RDY
            BR        RES.DO+4
      RES.DO: CLR      $REG5
      1$: BIT      #100, $RKDS      ;DID R/W/S RDY SET?
            BNE      2$
            MOV      #-10, -(SP)    ;PUSH COUNT ON SP
            INC      (SP)           ;COUNT IT DOW
            BNE      -2
            TST      (SP)+
            INC      $REG5
            BNE      1$
            BIT      #SW13, $SWR
            BNE      2$
            TYPMSG    MSG12
            TYPMSG    MSG7
            MOV      (SP), -(SP)
            SUB      #2, (SP)
            TYP0C
      2$: RTI
  
```


H07

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 69
 DZRKLD.P11 31-AUG-76 15:35 RESDON - WAIT FOR DRIVE RESET TO BE DONE

SEQ 0085

3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737

```

      .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 983 MS FOR 11/20
:175 MS FOR 11/45 WITH BIPOLAR MEMORY.
CN.RST: MOV      #1,DRKCS      ;ISSUE A CONTROL RESET
        MOV      #-300,$REG3   ;SET UP COUNT
        BR       CN.RDY+4      ;SKIP OVER CN.RDY
CN.RDY: CLR      $REG3
1$:     TSTB     DRKCS          ;DID CNTRL-RDY SET?
        BMI      2$           ;YES, EXIT
        INC      $REG3        ;WAITED LONG?
        BNE      1$           ;IF NOT, GO BAK & WAIT
        TYPMSG  MSG10
        TYPE     65$          ;;TYPE ASCIZ STRING
        BR       64$          ;;GET OVER THE ASCIZ
;;65$: .ASCIZ  <15><12>/PC=/
64$:   MOV      (SP),-(SP)
        SUB      #2,(SP)
        TYPOC
        TYPE     67$          ;;GO TYPE PC IN THE MAIN PROGRAM,
        BR       66$          ;;WHERE ERROR OCCURRED
        BR       66$          ;;TYPE ASCIZ STRING
        BR       66$          ;;GET OVER THE ASCIZ
;;67$: .ASCIZ  / RKCS=/
66$:   MOV      DRKCS,-(SP)    ;GET RKCS
        TYPOC                ;GO TYPE IT
2$:    RTI                  ;RETURN FROM THIS
                        ;ROUTINE TO THE MAIN

```

```

016526 012777 000001 162722
016534 012737 177500 001170
016542 000402
016544 005037 001170
016550 105777 162702
016554 100431
016556 005237 001170
016562 001372
016564 104420
016566 001742
016570 104401 016576
016574 000403
016604
016604 011646
016606 162716 000002
016612 104402
016614 104401 016622
016620 000404
016632
016632 017746 162620
016636 104402
016640 000002

```

MAINDEC-11-DZAKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 70
 DZAKLD.P11 31-AUG-76 15:35 CON.RDY - WAIT FOR CONTROL READY

SEQ 0086

;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

3738
 3739
 3740
 3741
 3742
 3743
 3744
 3745
 3746
 3747
 3748 016642 005037 001264
 3749 016646 032777 000100 162576
 3750 016654 001017
 3751 016656 005237 001264
 3752 016662 001371
 3753 016664 032777 020000 162246
 3754 016672 001010
 3755 016674 104420 002027
 3756 016700 104420 001733
 3757 016704 011646
 3758 016706 162716 000002
 3759 016712 104402
 3760 016714 000002

TSTRWS: CLR TIMER
 1\$: BIT #100,2RKDS
 BNE 2\$
 INC TIMER
 BNE 1\$
 BIT #BIT13,2SWR
 BNE 2\$
 TYPMSG ,MSG12
 TYPMSG ,MSG7
 MOV (SP),-(SP)
 SUB #2,(SP)
 TYP0C
 2\$: RTI

.SBTTL TEST ABORT ROUTINE

:ABRT

3761
 3762
 3763
 3764
 3765 016716 104401 001616
 3766 016722 113746 001102
 3767 016726 104402
 3768 016730 000207

ABRT: TYPE MSG3
 MOV \$TSTNM,-(SP)
 TYP0C
 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

3785 016732
 3786 016732 104407
 3787 016734 032777 000400 162176
 3788 016742 001053
 3789
 3790 016744 032777 040000 162166
 3791 016752 001047
 3792
 3793 016754 000416

\$SCOPE: CKSWR ;:TEST FOR CHANGE IN SOFT-SWR
 BIT #SW8,2SWR ;:WAS SW8 USED TO SELECT
 BNE \$OVER ;:A TEST? IF YES, SKIP OVER
 ;:THE REST, U ARE LOOPING ON
 1\$: BIT #BIT14,2SWR ;:LOOP ON PRESENT TEST?
 BNE \$OVER ;:YES IF SW14=1
 :*****START OF CODE FOR THE XOR TESTER*****
 \$XTSTR: BR 6\$;:IF RUNNING ON THE "XOR" TESTER CHANGE

J07

MAINDEC-11-DZRLD-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 71
 DZRLD.P11 31-AUG-76 15:35 SCOPE HANDLER ROUTINE

SEQ 0087

```

3794
3795 016756 013746 000004      MOV    2*ERRVEC, -(SP)    ;: THIS INSTRUCTION TO A "NOP" (NOP=240)
3796 016762 012737 017002 000004      MOV    5$, 2*ERRVEC    ;: SAVE THE CONTENTS OF THE ERROR VECTOR
3797 016770 005737 177060      TST     2*177060      ;: SET FOR TIMEOUT
3798 016774 012637 000004      MOV     (SP)+, 2*ERRVEC ;: TIME OUT ON XOR?
3799 017000 000421      BR      $SVLAD        ;: RESTORE THE ERROR VECTOR
3800 017002 022626      5$:    CMP     (SP)+, (SP)+    ;: GO TO THE NEXT TEST
3801 017004 012637 000004      MOV     (SP)+, 2*ERRVEC ;: CLEAR THE STACK AFTER A TIME OUT
3802 017010 000407      BR      7$          ;: RESTORE THE ERROR VECTOR
3803 017012      6$:;####END OF CODE FOR THE XOR TESTER#### ;: LOOP ON THE PRESENT TEST
3804 017012 105737 001103      2$:    TSTB    $ERFLG        ;: HAS AN ERROR OCCURRED?
3805 017016 001412      BEQ     $SVLAD        ;: BR IF NO
3806 017020 032777 001000 162112      BIT     #BIT09, 2$SWR ;: LOOP ON ERROR?
3807 017026 001404      BEQ     4$          ;: BR IF NO
3808 017030 013737 001110 001106      7$:    MOV     $LPERR, $LPADR ;: SET LOOP ADDRESS TO LAST SCOPE
3809 017036 000415      BR      $OVER
3810 017040 105037 001103      4$:    CLRB    $ERFLG        ;: ZERO THE ERROR FLAG
3811 017044 105237 001102      $SVLAD: INCB    $TSTNM       ;: COUNT TEST NUMBERS
3812 017050 011637 001106      MOV     (SP), $LPADR    ;: SAVE SCOPE LOOP ADDRESS
3813 017054 011637 001110      MOV     (SP), $LPERR    ;: SAVE ERROR LOOP ADDRESS
3814 017060 005037 001204      CLR     $ESCAPE      ;: CLEAR THE ESCAPE FROM ERROR ADDRESS
3815 017064 112737 000001 001115      MOVB   #1, $ERMAX     ;: ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3816 017072 013777 001102 162042      $OVER: MOV     $TSTNM, 2$DISPLAY ;: DISPLAY TEST NUMBER
3817 017100 013716 001106      MOV     $LPADR, (SP) ;: FUDGE RETURN ADDRESS
3818 017104 000002      RTI
3819
3820 ;:*****
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849

```

.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 PREVIOUS 'SCOPE' STATEMENT
;:GO TO ERRTP ON ERROR
;:NOT FROM SYSMAC

```

```

3833 017106 104407      $ERROR: CKSWR ;:CHECK FOR SOFTWARE SWITCH REGISTER REQUEST
3834 017110 105237 001103      7$:    INCB    $ERFLG        ;: SET THE ERROR FLAG
3835 017114 001775      BEQ     7$          ;: DON'T LET THE FLAG GO TO ZERO
3836 017116 013777 001102 162016      MOV     $TSTNM, 2$DISPLAY
3837 017124 032777 002000 162006      BIT     #SW10, 2$SWR
3838 017132 001402      BEQ     1$          ;:
3839 017134 104401 001206      TYPE    $BELL
3840 017140 005237 001112      1$:    INC     $ERTTL
3841 017144 011637 001116      MOV     (SP), $ERRPC
3842
3843 017150 032777 000004 161762      BIT     #SW2, 2$SWR ;: DROP THE DRIVE?
3844 017156 001404      BEQ     5$          ;: SW NOT SET, SKIP
3845 017160 023727 001112 000006      CMP     $ERTTL, #6    ;: MORE THAN 6 ERRORS ON THIS DRIVE?
3846 017166 101040      BHI     6$          ;: YES, DROP THE DRIVE
3847
3848 017170 162737 000002 001116      5$:    SUB     #2, $ERRPC
3849 017176 117737 161714 001114      MOVB   2$ERRPC, $ITEMB

```

K07

MAINDEC-11-DZAKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 72
 DZAKLD.P11 31-AUG-76 15:35 ERROR HANDLER ROUTINE

SEQ 0088

```

3850 017204 032777 020000 161726      BIT      #SW13,DSWR
3851 017212 001004      BNE      2$
3852 017214 004737 017362      JSR      PC,DSERRTYP
3853 017220 104401 001213      TYPE     $CRLF
3854 017224 005777 161710      2$:      TST      DSWR
3855 017230 100002      BPL      3$
3856 017232 000000      HALT
3857 017234 104407      CKSWR      ;CHECK FOR SOFTWARE SWITCH REGISTER REQUEST
3858 017236 032777 010000 161674 3$:      BIT      #SW12,DSWR
3859 017244 001402      BEQ      +6
3860 017246 013716 001106      MOV      $LPADR,(SP)
3861 017252 032777 001000 161660      BIT      #SW09,DSWR
3862 017260 001402      BEQ      4$
3863 017262 013716 001110      MOV      $LPERR,(SP)
3864 017266 000002      4$:      RTI
3865
3866 017270 013746 001226      6$:      MOV      DRVPTR,-(SP) ;GET POINTER TO DRIVE #
3867 017274 162716 000002      SUB      #2,(SP)
3868 017300 042736 000377      BIC      #377,D(SP)+ ;CLEAR THE DRIVE PRESENT FLAG
3869 017304 104401 002064      TYPE     MSG14
3870 017310 013746 001230      MOV      DRIVAD,-(SP)
3871 017314 000241      CLC
3872 017316 006116      ROL      (SP) ;GET THE DRIVE #
3873 017320 006116      ROL      (SP)
3874 017322 006116      ROL      (SP)
3875 017324 006116      ROL      (SP)
3876 017326 104402      TYPOC
3877 017330 104401 017336      TYPE     65$ ;TYPE IT OUT
3878 017334 000405      BR      64$ ;TYPE ASCIZ STRING
3879 ;65$: .ASCIZ / DROPPED/ ;GET OVER THE ASCIZ
3880 64$:
3881 017350 105337 001224      DECB     DRIVS ;DECRMNT # OF DRIVS PRESENT
3882 017354 022626      CMP      (SP)+,(SP)+ ;RESTORE STACK
3883 017356 000137 015156      JMP      BTEOP ;EXIT
3884
3885 017362      ERRTYP:
3886 017362 104401 001213      TYPE     $CRLF ;"CARRIAGE RETURN" & LINE FEED"
3887 017366 010046      MOV      RO,-(SP) ;SAVE RO
3888 017370 005000      CLR      RO ;PICKUP THE ITEM INDEX
3889 017372 153700 001114      BISB     D$ITEMB,RO
3890 017376 001011      BNE      1$ ;IF ITEM NUMBER IS ZERO, JUST
3891
3892
3893 017400 013746 001116      MOV      $ERRPC,-(SP) ;TYPE THE PC OF THE ERROR
3894 ;SAVE $ERRPC FOR TYPEOUT
3895 017404 104402      TYPOC ;ERROR ADDRESS
3896 017406 104401      TYPE ;GO TYPE--OCTAL ASCII(ALL DIGITS)
3897 017410 001733      MSG7
3898 017412 013746 001116      MOV      $ERRPC,-(SP)
3899 017416 104402      TYPOC
3900 017420 000440      BR      6$
3901 017422 005300      1$:      DEC      RO ;GET OUT
3902 017424 006300      ASL      RO ;ADJUST THE INDEX SO THAT IT WILL
3903 017426 006300      ASL      RO ;WORK FOR THE ERROR TABLE
3904 017430 006300      ASL      RO
3905 017432 062700 002122      ADD      $ERRTB,RO ;FORM TABLE POINTER

```


L07

SEQ 0089

```

3906 017436 012037 017446      MOV      (RO)+,2$      ;PICKUP "ERROR MESSAGE" POINTER
3907 017442 001404      BEQ      3$          ;SKIP TYPEOUT IF NOT POINTER
3908 017444 104401      TYPE      ;TYPE THE "ERROR MESSAGE"
3909 017446 000000      2$:      .WORD      0      ;"CARRIAGE RETURN" & LINE FEED"
3910 017450 104401 001213      TYPE      ,SCRLF      ;PICKUP "DATA HEADER" POINTER
3911 017454 032777 004000 161456 3$:      BIT      $SW11,$SWR      ;DUMP OUT ALL RK REGISTERS
3912 017462 001042      BNE      10$         ;YES, BRANCH
3913 017464 012037 017474      MOV      (RO)+,4$      ;PICKUP "DATA HEADER" POINTER
3914 017470 001412      BEQ      5$          ;SKIP TYPEOUT IF 0
3915 017472 104401      TYPE      ;TYPE THE "DATA HEADER"
3916 017474 000000      4$:      .WORD      0      ;"DATA HEADER" POINTER GOES HERE
3917 017476 104401 001213      TYPE      ,SCRLF      ;"CARRIAGE RETURN" & LINE FEED"
3918 017502 062700 000002      ADD      #2,RO      ;FORM POINTER TO TERMINATOR
3919 017506 005710      TST      (RO)          ;IS THE TERMINATOR 0?
3920 017510 001017      BNE      9$          ;IF NOT, BRANCH
3921 017512 162700 000002      SUB      #2,RO      ;YES, IT IS 0. REPOINT TO "DATA"
3922                                ;GO TYPE OUT DATA AS USUAL
3923 017516 011000      5$:      MOV      (RO),RO      ;PICKUP "DATA TABLE" POINTER
3924 017520 001004      BNE      7$          ;GO TYPE THE DATA
3925 017522 012600      6$:      MOV      (SP)+,RO      ;RESTORE RO
3926 017524 104401 001213      TYPE      ,SCRLF      ;"CARRIAGE RETURN" & LINE FEED"
3927 017530 000207      RTS      PC          ;RETURN
3928 017532      7$:
3929 017532 013046      MOV      2(RO)+,-(SP)      ;SAVE 2(RO)+ FOR TYPEOUT
3930 017534 104402      TYPLOC      ;GO TYPE--OCTAL ASCII(ALL DIGITS)
3931 017536 005710      TST      (RO)          ;IS THERE ANOTHER NUMBER?
3932 017540 001770      BEQ      6$          ;BR IF NO
3933 017542 104401 002110      TYPE      ,BLNKS2
3934 017546 000771      BR      7$
3935 017550 004770 000000      9$:      JSR      PC,2(RO)      ;GO TO THE SPECIAL ERROR
3936                                ;DATA HANDLING SUBROUTINE
3937                                ;NOTE THAT THIS ROUTINE IS
3938                                ;THE ONE INDICATED IN THE
3939                                ;LAST WORD OF AN ERROR
3940                                ;ITEM IN THE ERROR TABLE
3941                                ; (STARTING AT $ERRTB)
3942 017554 104401      TYPE
3943 017556 001733      MSG7
3944 017560 013746 001116      MOV      $ERRPC,-(SP)
3945 017564 104402      TYPLOC
3946 017566 000755      BR      6$          ;GO BACK TO THE EXIT POINT
3947                                ;FOR 'ERRTYP'
3948
3949 017570 004737 017576      10$:      JSR      PC,DMPREG
3950 017574 000752      BR      6$
3951
3952
3953
3954      :DMPREG
3955      :DUMPS OUT ALL RK REGISTERS WHEN SW 11 IS SET
3956
3957 017576      DMPREG:
3958 017576 104401 017604      TYPE      65$      ;:TYPE ASCIZ STRING
3959 017602 000441      BR      64$      ;:GET OVER THE ASCIZ
3960
3961      ;:65$: .ASCIZ <15><12>/ PC      RKDS      RKER      RKCS      RKWC      RKBA      RKDA      RKDB/<
      64$:

```

M07

MAINDEC-11-DZRL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 74
 DZRLD.F11 31-AUG-76 15:35 ERROR HANDLER ROUTINE

SEQ 0090

3962	017706	013746	001116	MOV	\$ERRPC,-(SP)
3963	017712	104402		TYPDC	
3964	017714	104401	002110	TYPE	BLNKS2
3965	017720	010046		MOV	RO,-(SP)
3966	017722	012700	001452	MOV	#RKDS,RO
3967	017726	013046		1\$: MOV	2(RO)+,-(SP)
3968	017730	104402		TYPDC	
3969	017732	104401	002110	TYPE	BLNKS2
3970	017736	020027	001466	CMP	RO,#RKDB
3971	017742	003771		BLE	1\$
3972	017744	012600		MOV	(SP)+,RO
3973	017746	000207		RTS	PC

.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

 ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 ;SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 ;NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 ;BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 ;REPLACED WITH SPACES.

;CALL:

;* MOV NUM,-(SP) ;:PUT THE BINARY NUMBER ON THE STACK
 ;* TYPDS ;:GO TO THE ROUTINE

\$TYPDS:

3991	017750			MOV	RO,-(SP)	;;PUSH RO ON STACK
3992	017750	010046		MOV	R1,-(SP)	;;PUSH R1 ON STACK
3993	017752	010146		MOV	R2,-(SP)	;;PUSH R2 ON STACK
3994	017754	010246		MOV	R3,-(SP)	;;PUSH R3 ON STACK
3995	017756	010346		MOV	R5,-(SP)	;;PUSH R5 ON STACK
3996	017760	010546		MOV	#20200,-(SP)	;;SET BLANK SWITCH AND SIGN
3997	017762	012746	020200	MOV	20(SP),R5	;;GET THE INPUT NUMBER
3998	017766	016605	000020	BPL	1\$	;;BR IF INPUT IS POS.
3999	017772	100004		NEG	R5	;;MAKE THE BINARY NUMBER POS.
4000	017774	005405		MOVB	#'-,1(SP)	;;MAKE THE ASCII NUMBER NEG.
4001	017776	112766	000055 000001	1\$: CLR	RO	;;ZERO THE CONSTANTS INDEX
4002	020004	005000		MOV	#\$DBLK,R3	;;SETUP THE OUTPUT POINTER
4003	020006	012703	020164	MOVB	#',(R3)+	;;SET THE FIRST CHARACTER TO A BLANK
4004	020012	112723	000040	2\$: CLR	R2	;;CLEAR THE BCD NUMBER
4005	020016	005002		MOV	\$DTBL(RO),R1	;;GET THE CONSTANT
4006	020020	016001	020154	3\$: SUB	R1,R5	;;FORM THIS BCD DIGIT
4007	020024	160105		BLT	4\$	;;BR IF DONE
4008	020026	002402		INC	R2	;;INCREASE THE BCD DIGIT BY 1
4009	020030	005202		BR	3\$	
4010	020032	000774		4\$: ADD	R1,R5	;;ADD BACK THE CONSTANT
4011	020034	060105		TST	R2	;;CHECK IF BCD DIGIT=0
4012	020036	005702		BNE	5\$	;;FALL THROUGH IF ?
4013	020040	001002		TSTB	(SP)	;;STILL DOING LEADING 0'S?
4014	020042	105716		BMI	7\$	;;BR IF YES
4015	020044	100407		ASLB	(SP)	;;MSD?
4016	020046	106316		BCC	6\$	;;BR IF NO
4017	020050	103003				

N07

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 75
DZRKLD.P11 31-AUG-76 15:35 CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

SEQ 0091

4018	020052	116663	000001	177777		MOV	1(SP),-1(R3)	::YES--SET THE SIGN
4019	020050	052702	000060		6\$:	BIS	#'0,R2	::MAKE THE BCD DIGIT ASCII
4020	020064	052702	000040		7\$:	BIS	#' ,R2	::MAKE IT A SPACE IF NOT ALREADY A DIGIT
4021	020070	110223				MOV	R2,(R3)+	::PUT THIS CHARACTER IN THE OUTPUT BUFFER
4022	020072	005720				TST	(R0)+	::JUST INCREMENTING
4023	020074	020027	000010			CMP	R0,#10	::CHECK THE TABLE INDEX
4024	020100	002746				BLT	2\$	::GO DO THE NEXT DIGIT
4025	020102	003002				BGT	8\$	::GO TO EXIT
4026	020104	010502				MOV	R5,R2	::GET THE LSD
4027	020106	000764				BR	6\$	::GO CHANGE TO ASCII
4028	020110	105726			8\$:	TSTB	(SP)+	::WAS THE LSD THE FIRST NON-ZERO?
4029	020112	100003				BPL	9\$	::BR IF NO
4030	020114	116663	177777	177776		MOV	-1(SP),-2(R3)	::YES--SET THE SIGN FOR TYPING
4031	020122	105013			9\$:	CLRB	(R3)	::SET THE TERMINATOR
4032	020124	012605				MOV	(SP)+,R5	::POP STACK INTO R5
4033	020126	012603				MOV	(SP)+,R3	::POP STACK INTO R3
4034	020130	012602				MOV	(SP)+,R2	::POP STACK INTO R2
4035	020132	012601				MOV	(SP)+,R1	::POP STACK INTO R1
4036	020134	012600				MOV	(SP)+,R0	::POP STACK INTO R0
4037	020136	104401	020164			TYPE	\$DBLK	::NOW TYPE THE NUMBER
4038	020142	016666	000002	000004		MOV	2(SP),4(SP)	::ADJUST THE STACK
4039	020150	012616				MOV	(SP)+,(SP)	
4040	020152	000002				RTI		::RETURN TO USER

\$DTBL: 10000.

1000.

100.

10.

\$DBLK: .BLKW 4

.SBTTL TYPE ROUTINE

```

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

```

4064	020174	105737	001157		\$TYPE:	TSTB	\$TPFLG	::IS THERE A TERMINAL?
4065	020200	100002				BPL	1\$	::BR IF YES
4066	020202	000000				HALT		::HALT HERE IF NO TERMINAL
4067	020204	000407				BR	3\$	::LEAVE
4068	020206	010046			1\$:	MOV	R0,-(SP)	::SAVE R0
4069	020210	017600	000002			MOV	32(SP),R0	::GET ADDRESS OF ASCIIZ STRING
4070	020214	112046			2\$:	MOV	(R0)+,-(SP)	::PUSH CHARACTER TO BE TYPED ONTO STACK
4071	020216	001005				BNE	4\$	::BR IF IT ISN'T THE TERMINATOR
4072	020220	005726				TST	(SP)+	::IF TERMINATOR POP IT OFF THE STACK
4073	020222	012600			60\$:	MOV	(SP)+,R0	::RESTORE R0

MAINDEC-11-DZRLD-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 76
DZRLD.F11 31-AUG-76 15:35 TYPE ROUTINE

SEQ 0092

```

4074 020224 062716 000002 3$: ADD #2,(SP) ;;ADJUST RETURN PC
4075 020230 000002 ;;RETURN
4076 020232 122716 000011 4$: CMPB #HT,(SP) ;;BRANCH IF <HT>
4077 020236 001430 BEQ #S ;;BRANCH IF NOT <CR>
4078 020240 122716 000200 CMPB #CRLF,(SP) ;;BRANCH IF NOT <CRLF>
4079 020244 001006 BNE #S ;;POP <CR><LF> EQUIV
4080 020246 005726 TST (SP)+ ;;TYPE A CR AND LF
4081 020250 104401 TYPE
4082 020252 001213 SCRLF
4083 020254 105037 020410 CLRB $CHARCNT ;;CLEAR CHARACTER COUNT
4084 020260 000755 BR #S ;;GET NEXT CHARACTER
4085 020262 004737 020344 5$: JSR PC,$TYPEC ;;GO TYPE THIS CHARACTER
4086 020266 123726 001156 6$: CMPB $FILLC,(SP)+ ;;IS IT TIME FOR FILLER CHARS.?
4087 020272 001350 BNE #S ;;IF NO GO GET NEXT CHAR.
4088 020274 013746 001154 MOV $NULL,-(SP) ;;GET # OF FILLER CHARS. NEEDED
4089 020300 105366 000001 7$: DECB 1(SP) ;;AND THE NULL CHAR.
4090 020304 002770 BLT #S ;;DOES A NULL NEED TO BE TYPED?
4091 020306 004737 020344 JSR PC,$TYPEC ;;BR IF NO--GO POP THE NULL OFF OF STACK
4092 020312 105337 020410 DECB $CHARCNT ;;GO TYPE A NULL
4093 020316 000770 BR #S ;;DO NOT COUNT AS A COUNT
4094 020316 000770 BR #S ;;LOOP

```

;HORIZONTAL TAB PROCESSOR

```

4098 020320 112716 000040 8$: MOVB #',(SP) ;;REPLACE TAB WITH SPACE
4099 020324 004737 020344 9$: JSR PC,$TYPEC ;;TYPE A SPACE
4100 020330 132737 000007 020410 BITB #7,$CHARCNT ;;BRANCH IF NOT AT
4101 020336 001372 BNE #S ;;TAB STOP
4102 020340 005726 TST (SP)+ ;;POP SPACE OFF STACK
4103 020342 000724 BR #S ;;GET NEXT CHARACTER
4104 020344 105777 160600 $TYPEC: TSTB $STPB ;;WAIT UNTIL PRINTER IS READY
4105 020350 100375 BPL $TYPEC
4106 020352 116677 000002 160572 MOVB 2(SP),$STPB ;;LOAD CHAR TO BE TYPED INTO DATA REG.
4107 020360 122766 000015 000002 CMPB #CR,2(SP) ;;IS CHARACTER A CARRIAGE RETURN?
4108 020366 001003 BNE #S ;;BRANCH IF NO
4109 020370 105037 020410 CLRB $CHARCNT ;;YES--CLEAR CHARACTER COUNT
4110 020374 000406 BR $TYPEX ;;EXIT
4111 020376 122766 000012 000002 1$: CMPB #LF,2(SP) ;;IS CHARACTER A LINE FEED?
4112 020404 001402 BEQ $TYPEX ;;BRANCH IF YES
4113 020406 105227 INCB (PC)+ ;;COUNT THE CHARACTER
4114 020410 000000 $CHARCNT: .WORD 0 ;;CHARACTER COUNT STORAGE
4115 020412 000207 $TYPEX: RTS PC

```

.SBTTL INTEGER MULTIPLY ROUTINE

```

*****
CALL
MOV MULTIPLIER,-(SP)
MOV MULTIPLICAND,-(SP)
JSR PC,$MULT
RETURN ;;PRODUCT IS ON THE STACK

STACK PRODUCT
-----

```


C08

MAINDEC-11-DZKRL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 77
 DZKRLD.F11 31-AUG-76 15:35 INTEGER MULTIPLY ROUTINE

SEQ 0093

```

4130      ;*      TOP      LSB'S
4131      ;*      +2      MSB'S
4132
4133      $MULT:
4134      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
4135      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
4136      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
4137      CLR      -(SP)        ;;CLEAR THE SIGN KEY
4138      MOV      12(SP),R1     ;;GET THE MULTIPLICAND
4139      BPL      1$           ;;BR IF PLUS
4140      INC      (SP)         ;;SET THE SIGN KEY
4141      NEG      R1           ;;MAKE THE MULTIPLICAND POSTIVE
4142      MOV      14(SP),R2     ;;GET THE MULTIPLIER
4143      BPL      2$           ;;BR IF PLUS
4144      DEC      (SP)         ;;UPDATE THE SIGN KEY
4145      NEG      R2           ;;MAKE THE MULTIPLIER POSTIVE
4146      MOV      17,-(SP)     ;;SET THE LOOP COUNT
4147      CLR      R0           ;;SETUP FOR THE MULTIPLY LOOP
4148      BCC      4$           ;;DON'T ADD IF MULTIPLICAND = 0
4149      ADD      R2,R0
4150      ROR      R0           ;;POSITION THE PARITIAL PRODUCT AND
4151      ROR      R1           ;;THE MULTIPLICAND
4152      DEC      (SP)         ;;HAS ALL BITS OF THE MULTIPLICAND BEEN DONE?
4153      BNE      3$           ;;BR IF NO
4154      CMP      (SP)+,(SP)   ;;SHOULD PRODUCT BE NEGATIVE?
4155      BEQ      5$           ;;GO TO EXIT IF NO
4156      NEG      R0           ;;YES--SO MAKE IT SO
4157      NEG      R1
4158      SBC      R0
4159      TST      (SP)+        ;;CLEAR SIGN INFO. OFF OF STACK
4160      MOV      R0,12(SP)     ;;PUT THE PRODUCT ON THE STACK (MSB'S)
4161      MOV      R1,10(SP)     ;;LSB'S
4162      MOV      (SP)+,R2     ;;POP STACK INTO R2
4163      MOV      (SP)+,R1     ;;POP STACK INTO R1
4164      MOV      (SP)+,R0     ;;POP STACK INTO R0
4165      RTS      PC
4166
4167      .SBTTL  TTY INPUT ROUTINE
4168
4169      ;*****
4170      .ENABL  LSB
4171      STKCNT: .WORD  0      ;;NUMBER OF ITEMS IN QUEUE
4172      STKQIN: .WORD  0      ;;INPUT POINTER
4173      STKQOUT: .WORD  0     ;;OUTPUT POINTER
4174      STKQSR: .BLKB  1     ;;TTY KEYBOARD QUEUE
4175      STKQEND=.
4176      .EVEN
4177
4178      ;*TK INITIALIZE ROUTINE
4179      ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
4180      ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
4181
4182      ;*CALL:
4183      ;*      JSR      PC,STKINT
4184      ;*      RETURN
4185

```

000012

000014

000021

000012

000010

MAINDEC-11-DZRL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 78
 DZRLD.P11 31-AUG-76 15:35 TTY INPUT ROUTINE

SEQ 0094

```

4186 020536 005037 020526 STKINT: CLR STKCNT ;; CLEAR COUNT OF ITEMS IN QUEUE
4187 020542 012737 020534 020530 MOV #STKQSR, STKQIN ;; MOVE THE STARTING ADDRESS OF THE
4188 020550 013737 020530 020532 MOV STKQIN, STKQOUT ;; QUEUE INTO THE INPUT & OUTPUT POINTERS.
4189 020556 012737 020606 000060 MOV #STKSRV, STKVEC ;; INITIALIZE THE KEYBOARD VECTOR
4190 020564 012737 000200 000062 MOV #200, STKVEC+2 ;; "BR" LEVEL 4
4191 020572 005777 160350 TST STKB ;; CLEAR DONE FLAG
4192 020576 012777 000100 160340 MOV #100, STKS ;; ENABLE TTY KEYBOARD INTERRUPT
4193 020604 000207 RTS PC ;; RETURN TO CALLER

```

```

;; *TK SERVICE ROUTINE
;; *THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
;; *BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
;; *IT IN THE QUEUE.

```

```

4200 020606 117746 160334 STKSRV: MOVB STKB, -(SP) ;; PICKUP THE CHARACTER
4201 020612 042716 177600 BIC #C177, (SP) ;; STRIP THE JUNK
4202 020616 021627 000007 1$: CMP (SP), #7 ;; IS IT A CONTROL G?
4203 020622 001004 BNE 2$ ;; BRANCH IF NO
4204 020624 022737 000176 001140 CMP #SWREG, SWR ;; IS SOFT-SWR SELECTED?
4205 020632 001500 BEQ 6$ ;; GO TO SWR CHANGE
4206
4207 020634 022737 000001 020526 2$: CMP #1, STKCNT ;; IS THE QUEUE FULL?
4208 020642 001004 BNE 3$ ;; BRANCH IF NO
4209 020644 104401 001206 TYPE $BELL ;; RING THE TTY BELL
4210 020650 005726 TST (SP)+ ;; CLEAN CHARACTER OFF OF STACK
4211 020652 000451 BR 5$ ;; EXIT
4212 020654 021627 000023 3$: CMP (SP), #23 ;; IS IT A CONTROL-S?
4213 020660 001021 BNE 32$ ;; BRANCH IF NO
4214 020662 005077 160256 CLR STKS ;; DISABLE TTY KEYBOARD INTERRUPTS
4215 020666 005726 TST (SP)+ ;; CLEAN CHAR OFF STACK
4216 020670 105777 160250 31$: TSTB STKS ;; WAIT FOR A CHAR
4217 020674 100375 BPL 31$ ;; LOOP UNTIL ITS THERE
4218 020676 117746 160244 MOVB STKB, -(SP) ;; GET THE CHARACTER
4219 020702 042716 177600 BIC #C177, (SP) ;; MAKE IT 7-BIT ASCII
4220 020706 022627 000021 CMP (SP)+, #21 ;; IS IT A CONTROL-Q?
4221 020712 001366 BNE 31$ ;; BRANCH IF NO
4222 020714 012777 000100 160222 MOV #100, STKS ;; REENABLE TTY KEYBOARD INTERRUPTS
4223 020722 000002 RTI ;; RETURN
4224 020724 005237 020526 32$: INC STKCNT ;; COUNT THIS CHARACTER
4225 020730 021627 000140 CMP (SP), #140 ;; IS IT UPPER CASE?
4226 020734 002405 BLT 4$ ;; BRANCH IF YES
4227 020736 021627 000175 CMP (SP), #175 ;; IS IT A SPECIAL CHAR?
4228 020742 003002 BGT 4$ ;; BRANCH IF YES
4229 020744 042716 000040 BIC #40, (SP) ;; MAKE IT UPPER CASE
4230 020750 112677 177554 4$: MOVB (SP)+, STKQIN ;; AND PUT IT IN QUEUE
4231 020754 005237 020530 INC STKQIN ;; UPDATE THE POINTER
4232 020760 023727 020530 020535 CMP STKQIN, STKQEND ;; GO OFF THE END?
4233 020766 001003 BNE 5$ ;; BRANCH IF NO
4234 020770 012737 020534 020530 MOV #STKQSR, STKQIN ;; RESET THE POINTER
4235 020776 000002 5$: RTI ;; RETURN

```

```

*****
;; *SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
;; *ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
;; *SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP

```


E08

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 79
 DZRKLD.P11 31-AUG-76 15:35 TTY INPUT ROUTINE

SEQ 0095

```

4242      *CALL WHEN OPERATING IN TTY INTERRUPT MODE.
4243      $CKSWR: CMP      #SWREG,SWP      ;;IS THE SOFT-SWR SELECTED
4244             BNE      15$             ;;EXIT IF NOT
4245             TSTB     @STKS           ;;IS A CHAR WAITING?
4246             BPL      15$             ;;IF NOT, EXIT
4247             MOVB     @STKB,-(SP)      ;;YES
4248             BIC      #1C177,(SP)     ;;MAKE IT 7-BIT ASCII
4249             CMP      (SP),#7         ;;IS IT A CONTROL-G?
4250             BNE      25$             ;;IF NOT, PUT IT IN THE TTY QUEUE
4251             ;;AND EXIT

*****
4252      *CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
4253      *ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
4254      *CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
4255      6$: CMPB      $AUTOB,#1          ;;ARE WE RUNNING IN AUTO-MODE?
4256             BEQ      25$             ;;BRANCH IF YES
4257             TST      (SP)+           ;;CLEAR CONTROL-G OFF STACK
4258             JSR      PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
4259             CLR      @STKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
4260             MOVB     #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
4261
4262      SGT$SWR: TYPE      ,$CNTLG       ;;ECHO THE CONTROL-G (1G)
4263             TYPE      ,$MSWR         ;;TYPE CURRENT CONTENTS
4264             MOV      $SWREG,-(SP)    ;;SAVE SWREG FOR TYPEOUT
4265             TYPEOC    ,$MNEW         ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4266             TYPE      ,(SP)         ;;PROMPT FOR NEW SWR
4267             CLR      -(SP)           ;;CLEAR COUNTER
4268             CLR      -(SP)           ;;THE NEW SWR
4269             19$: TSTB     @STKS       ;;CHAR THERE?
4270             BPL      7$             ;;IF NOT TRY AGAIN
4271             MOVB     @STKB,-(SP)    ;;PICK UP CHAR
4272             BIC      #1C177,(SP)    ;;MAKE IT 7-BIT ASCII
4273
4274             9$: CMP      (SP),#25    ;;IS IT A CONTROL-U?
4275             BNE      10$            ;;BRANCH IF NOT
4276             TYPE      ,$CNTLU       ;;YES, ECHO CONTROL-U (1U)
4277             ADD      #6,SP          ;;IGNORE PREVIOUS INPUT
4278             BR       19$            ;;LET'S TRY IT AGAIN
4279
4280             10$: CMP      (SP),#15   ;;IS IT A <CR>?
4281             BNE      16$            ;;BRANCH IF NO
4282             TST      4(SP)          ;;YES, IS IT THE FIRST CHAR?
4283             BEQ      11$            ;;BRANCH IF YES
4284             MOV      2(SP),@SWR     ;;SAVE NEW SWR
4285             ADD      #6,SP          ;;CLEAR UP STACK
4286             TYPE      ,$CRLF        ;;ECHO <CR> AND <LF>
4287             CMPB     $INTAG,#1      ;;RE-ENABLE TTY KBD INTERRUPTS?
4288             BNE      15$            ;;BRANCH IF NOT
4289             MOV      #100,@STKS     ;;RE-ENABLE TTY KBD INTERRUPTS
4290             RTI                     ;;RETURN
4291             JSR      PC,$TYPEOC     ;;ECHO CHAR
4292
4293      11$: ADD      #6,SP
4294      14$: TYPE      ,$CRLF
4295      15$: RTI
4296      16$: JSR      PC,$TYPEOC
4297
4298      17$: TSTB     @STKS
4299      18$: BPL      15$
4300      19$: CLR      -(SP)
4301      20$: ADD      #6,SP
4302      21$: BR       19$
4303
4304      22$: CMP      (SP),#15
4305      23$: BNE      16$
4306      24$: TST      4(SP)
4307      25$: BEQ      11$
4308      26$: MOV      2(SP),@SWR
4309      27$: ADD      #6,SP
4310      28$: TYPE      ,$CRLF
4311      29$: CMPB     $INTAG,#1
4312      30$: BNE      15$
4313      31$: MOV      #100,@STKS
4314      32$: RTI
4315      33$: JSR      PC,$TYPEOC
4316
4317      34$: TSTB     @STKS
4318      35$: BPL      7$
4319      36$: MOVB     @STKB,-(SP)
4320      37$: BIC      #1C177,(SP)
4321
4322      38$: CMPB     $AUTOB,#1
4323      39$: BEQ      25$
4324      40$: TST      (SP)+
4325      41$: JSR      PC,$TKINT
4326      42$: CLR      @STKS
4327      43$: MOVB     #1,$INTAG
4328
4329      44$: TYPE      ,$CNTLG
4330      45$: TYPE      ,$MSWR
4331      46$: MOV      $SWREG,-(SP)
4332      47$: TYPEOC    ,$MNEW
4333      48$: TYPE      ,(SP)
4334      49$: CLR      -(SP)
4335      50$: CLR      -(SP)
4336      51$: TSTB     @STKS
4337      52$: BPL      7$
4338      53$: MOVB     @STKB,-(SP)
4339      54$: BIC      #1C177,(SP)
4340
4341      55$: CMP      (SP),#25
4342      56$: BNE      10$
4343      57$: TYPE      ,$CNTLU
4344      58$: ADD      #6,SP
4345      59$: BR       19$
4346
4347      60$: CMP      (SP),#15
4348      61$: BNE      16$
4349      62$: TST      4(SP)
4350      63$: BEQ      11$
4351      64$: MOV      2(SP),@SWR
4352      65$: ADD      #6,SP
4353      66$: TYPE      ,$CRLF
4354      67$: CMPB     $INTAG,#1
4355      68$: BNE      15$
4356      69$: MOV      #100,@STKS
4357      70$: RTI
4358      71$: JSR      PC,$TYPEOC
4359
4360      72$: TSTB     @STKS
4361      73$: BPL      7$
4362      74$: MOVB     @STKB,-(SP)
4363      75$: BIC      #1C177,(SP)
4364
4365      76$: CMPB     $AUTOB,#1
4366      77$: BEQ      25$
4367      78$: TST      (SP)+
4368      79$: JSR      PC,$TKINT
4369      80$: CLR      @STKS
4370      81$: MOVB     #1,$INTAG
4371
4372      82$: TYPE      ,$CNTLG
4373      83$: TYPE      ,$MSWR
4374      84$: MOV      $SWREG,-(SP)
4375      85$: TYPEOC    ,$MNEW
4376      86$: TYPE      ,(SP)
4377      87$: CLR      -(SP)
4378      88$: CLR      -(SP)
4379      89$: TSTB     @STKS
4380      90$: BPL      7$
4381      91$: MOVB     @STKB,-(SP)
4382      92$: BIC      #1C177,(SP)
4383
4384      93$: CMP      (SP),#25
4385      94$: BNE      10$
4386      95$: TYPE      ,$CNTLU
4387      96$: ADD      #6,SP
4388      97$: BR       19$
4389
4390      98$: CMP      (SP),#15
4391      99$: BNE      16$
4400     100$: TST      4(SP)
4401     101$: BEQ      11$
4402     102$: MOV      2(SP),@SWR
4403     103$: ADD      #6,SP
4404     104$: TYPE      ,$CRLF
4405     105$: CMPB     $INTAG,#1
4406     106$: BNE      15$
4407     107$: MOV      #100,@STKS
4408     108$: RTI
4409     109$: JSR      PC,$TYPEOC
4410
4411     110$: TSTB     @STKS
4412     111$: BPL      7$
4413     112$: MOVB     @STKB,-(SP)
4414     113$: BIC      #1C177,(SP)
4415
4416     114$: CMPB     $AUTOB,#1
4417     115$: BEQ      25$
4418     116$: TST      (SP)+
4419     117$: JSR      PC,$TKINT
4420     118$: CLR      @STKS
4421     119$: MOVB     #1,$INTAG
4422
4423     120$: TYPE      ,$CNTLG
4424     121$: TYPE      ,$MSWR
4425     122$: MOV      $SWREG,-(SP)
4426     123$: TYPEOC    ,$MNEW
4427     124$: TYPE      ,(SP)
4428     125$: CLR      -(SP)
4429     126$: CLR      -(SP)
4430     127$: TSTB     @STKS
4431     128$: BPL      7$
4432     129$: MOVB     @STKB,-(SP)
4433     130$: BIC      #1C177,(SP)
4434
4435     131$: CMP      (SP),#25
4436     132$: BNE      10$
4437     133$: TYPE      ,$CNTLU
4438     134$: ADD      #6,SP
4439     135$: BR       19$
4440
4441     136$: CMP      (SP),#15
4442     137$: BNE      16$
4443     138$: TST      4(SP)
4444     139$: BEQ      11$
4445     140$: MOV      2(SP),@SWR
4446     141$: ADD      #6,SP
4447     142$: TYPE      ,$CRLF
4448     143$: CMPB     $INTAG,#1
4449     144$: BNE      15$
4450     145$: MOV      #100,@STKS
4451     146$: RTI
4452     147$: JSR      PC,$TYPEOC
4453
4454     148$: TSTB     @STKS
4455     149$: BPL      7$
4456     150$: MOVB     @STKB,-(SP)
4457     151$: BIC      #1C177,(SP)
4458
4459     152$: CMPB     $AUTOB,#1
4460     153$: BEQ      25$
4461     154$: TST      (SP)+
4462     155$: JSR      PC,$TKINT
4463     156$: CLR      @STKS
4464     157$: MOVB     #1,$INTAG
4465
4466     158$: TYPE      ,$CNTLG
4467     159$: TYPE      ,$MSWR
4468     160$: MOV      $SWREG,-(SP)
4469     161$: TYPEOC    ,$MNEW
4470     162$: TYPE      ,(SP)
4471     163$: CLR      -(SP)
4472     164$: CLR      -(SP)
4473     165$: TSTB     @STKS
4474     166$: BPL      7$
4475     167$: MOVB     @STKB,-(SP)
4476     168$: BIC      #1C177,(SP)
4477
4478     169$: CMP      (SP),#25
4479     170$: BNE      10$
4480     171$: TYPE      ,$CNTLU
4481     172$: ADD      #6,SP
4482     173$: BR       19$
4483
4484     174$: CMP      (SP),#15
4485     175$: BNE      16$
4486     176$: TST      4(SP)
4487     177$: BEQ      11$
4488     178$: MOV      2(SP),@SWR
4489     179$: ADD      #6,SP
4490     180$: TYPE      ,$CRLF
4491     181$: CMPB     $INTAG,#1
4492     182$: BNE      15$
4493     183$: MOV      #100,@STKS
4494     184$: RTI
4495     185$: JSR      PC,$TYPEOC
4496
4497     186$: TSTB     @STKS
4498     187$: BPL      7$
4499     188$: MOVB     @STKB,-(SP)
4500     189$: BIC      #1C177,(SP)
4501
4502     190$: CMPB     $AUTOB,#1
4503     191$: BEQ      25$
4504     192$: TST      (SP)+
4505     193$: JSR      PC,$TKINT
4506     194$: CLR      @STKS
4507     195$: MOVB     #1,$INTAG
4508
4509     196$: TYPE      ,$CNTLG
4510     197$: TYPE      ,$MSWR
4511     198$: MOV      $SWREG,-(SP)
4512     199$: TYPEOC    ,$MNEW
4513     200$: TYPE      ,(SP)
4514     201$: CLR      -(SP)
4515     202$: CLR      -(SP)
4516     203$: TSTB     @STKS
4517     204$: BPL      7$
4518     205$: MOVB     @STKB,-(SP)
4519     206$: BIC      #1C177,(SP)
4520
4521     207$: CMP      (SP),#25
4522     208$: BNE      10$
4523     209$: TYPE      ,$CNTLU
4524     210$: ADD      #6,SP
4525     211$: BR       19$
4526
4527     212$: CMP      (SP),#15
4528     213$: BNE      16$
4529     214$: TST      4(SP)
4530     215$: BEQ      11$
4531     216$: MOV      2(SP),@SWR
4532     217$: ADD      #6,SP
4533     218$: TYPE      ,$CRLF
4534     219$: CMPB     $INTAG,#1
4535     220$: BNE      15$
4536     221$: MOV      #100,@STKS
4537     222$: RTI
4538     223$: JSR      PC,$TYPEOC
4539
4540     224$: TSTB     @STKS
4541     225$: BPL      7$
4542     226$: MOVB     @STKB,-(SP)
4543     227$: BIC      #1C177,(SP)
4544
4545     228$: CMPB     $AUTOB,#1
4546     229$: BEQ      25$
4547     230$: TST      (SP)+
4548     231$: JSR      PC,$TKINT
4549     232$: CLR      @STKS
4550     233$: MOVB     #1,$INTAG
4551
4552     234$: TYPE      ,$CNTLG
4553     235$: TYPE      ,$MSWR
4554     236$: MOV      $SWREG,-(SP)
4555     237$: TYPEOC    ,$MNEW
4556     238$: TYPE      ,(SP)
4557     239$: CLR      -(SP)
4558     240$: CLR      -(SP)
4559     241$: TSTB     @STKS
4560     242$: BPL      7$
4561     243$: MOVB     @STKB,-(SP)
4562     244$: BIC      #1C177,(SP)
4563
4564     245$: CMP      (SP),#25
4565     246$: BNE      10$
4566     247$: TYPE      ,$CNTLU
4567     248$: ADD      #6,SP
4568     249$: BR       19$
4569
4570     250$: CMP      (SP),#15
4571     251$: BNE      16$
4572     252$: TST      4(SP)
4573     253$: BEQ      11$
4574     254$: MOV      2(SP),@SWR
4575     255$: ADD      #6,SP
4576     256$: TYPE      ,$CRLF
4577     257$: CMPB     $INTAG,#1
4578     258$: BNE      15$
4579     259$: MOV      #100,@STKS
4580     260$: RTI
4581     261$: JSR      PC,$TYPEOC
4582
4583     262$: TSTB     @STKS
4584     263$: BPL      7$
4585     264$: MOVB     @STKB,-(SP)
4586     265$: BIC      #1C177,(SP)
4587
4588     266$: CMPB     $AUTOB,#1
4589     267$: BEQ      25$
4590     268$: TST      (SP)+
4591     269$: JSR      PC,$TKINT
4592     270$: CLR      @STKS
4593     271$: MOVB     #1,$INTAG
4594
4595     272$: TYPE      ,$CNTLG
4596     273$: TYPE      ,$MSWR
4597     274$: MOV      $SWREG,-(SP)
4598     275$: TYPEOC    ,$MNEW
4599     276$: TYPE      ,(SP)
4600     277$: CLR      -(SP)
4601     278$: CLR      -(SP)
4602     279$: TSTB     @STKS
4603     280$: BPL      7$
4604     281$: MOVB     @STKB,-(SP)
4605     282$: BIC      #1C177,(SP)
4606
4607     283$: CMP      (SP),#25
4608     284$: BNE      10$
4609     285$: TYPE      ,$CNTLU
4610     286$: ADD      #6,SP
4611     287$: BR       19$
4612
4613     288$: CMP      (SP),#15
4614     289$: BNE      16$
4615     290$: TST      4(SP)
4616     291$: BEQ      11$
4617     292$: MOV      2(SP),@SWR
4618     293$: ADD      #6,SP
4619     294$: TYPE      ,$CRLF
4620     295$: CMPB     $INTAG,#1
4621     296$: BNE      15$
4622     297$: MOV      #100,@STKS
4623     298$: RTI
4624     299$: JSR      PC,$TYPEOC
4625
4626     300$: TSTB     @STKS
4627     301$: BPL      7$
4628     302$: MOVB     @STKB,-(SP)
4629     303$: BIC      #1C177,(SP)
4630
4631     304$: CMPB     $AUTOB,#1
4632     305$: BEQ      25$
4633     306$: TST      (SP)+
4634     307$: JSR      PC,$TKINT
4635     308$: CLR      @STKS
4636     309$: MOVB     #1,$INTAG
4637
4638     310$: TYPE      ,$CNTLG
4639     311$: TYPE      ,$MSWR
4640     312$: MOV      $SWREG,-(SP)
4641     313$: TYPEOC    ,$MNEW
4642     314$: TYPE      ,(SP)
4643     315$: CLR      -(SP)
4644     316$: CLR      -(SP)
4645     317$: TSTB     @STKS
4646     318$: BPL      7$
4647     319$: MOVB     @STKB,-(SP)
4648     320$: BIC      #1C177,(SP)
4649
4650     321$: CMP      (SP),#25
4651     322$: BNE      10$
4652     323$: TYPE      ,$CNTLU
4653     324$: ADD      #6,SP
4654     325$: BR       19$
4655
4656     326$: CMP      (SP),#15
4657     327$: BNE      16$
4658     328$: TST      4(SP)
4659     329$: BEQ      11$
4660     330$: MOV      2(SP),@SWR
4661     331$: ADD      #6,SP
4662     332$: TYPE      ,$CRLF
4663     333$: CMPB     $INTAG,#1
4664     334$: BNE      15$
4665     335$: MOV      #100,@STKS
4666     336$: RTI
4667     337$: JSR      PC,$TYPEOC
4668
4669     338$: TSTB     @STKS
4670     339$: BPL      7$
4671     340$: MOVB     @STKB,-(SP)
4672     341$: BIC      #1C177,(SP)
4673
4674     342$: CMPB     $AUTOB,#1
4675     343$: BEQ      25$
4676     344$: TST      (SP)+
4677     345$: JSR      PC,$TKINT
4678     346$: CLR      @STKS
4679     347$: MOVB     #1,$INTAG
4680
4681     348$: TYPE      ,$CNTLG
4682     349$: TYPE      ,$MSWR
4683     350$: MOV      $SWREG,-(SP)
4684     351$: TYPEOC    ,$MNEW
4685     352$: TYPE      ,(SP)
4686     353$: CLR      -(SP)
4687     354$: CLR      -(SP)
4688     355$: TSTB     @STKS
4689     356$: BPL      7$
4690     357$: MOVB     @STKB,-(SP)
4691     358$: BIC      #1C177,(SP)
4692
4693     359$: CMP      (SP),#25
4694     360$: BNE      10$
4695     361$: TYPE      ,$CNTLU
4696     362$: ADD      #6,SP
4697     363$: BR       19$
4698
4699     364$: CMP      (SP),#15
4700     365$: BNE      16$
4701     366$: TST      4(SP)
4702     367$: BEQ      11$
4703     368$: MOV      2(SP),@SWR
4704     369$: ADD      #6,SP
4705     370$: TYPE      ,$CRLF
4706     371$: CMPB     $INTAG,#1
4707     372$: BNE      15$
4708     373$: MOV      #100,@STKS
4709     374$: RTI
4710     375$: JSR      PC,$TYPEOC
4711
4712     376$: TSTB     @STKS
4713     377$: BPL      7$
4714     378$: MOVB     @STKB,-(SP)
4715     379$: BIC      #1C177,(SP)
4716
4717     380$: CMPB     $AUTOB,#1
4718     381$: BEQ      25$
4719     382$: TST      (SP)+
4720     383$: JSR      PC,$TKINT
4721     384$: CLR      @STKS
4722     385$: MOVB     #1,$INTAG
4723
4724     386$: TYPE      ,$CNTLG
4725     387$: TYPE      ,$MSWR
4726     388$: MOV      $SWREG,-(SP)
4727     389$: TYPEOC    ,$MNEW
4728     390$: TYPE      ,(SP)
4729     391$: CLR      -(SP)
4730     392$: CLR      -(SP)
4731     393$: TSTB     @STKS
4732     394$: BPL      7$
4733     395$: MOVB     @STKB,-(SP)
4734     396$: BIC      #1C177,(SP)
4735
4736     397$: CMP      (SP),#25
4737     398$: BNE      10$
4738     399$: TYPE      ,$CNTLU
4739     400$: ADD      #6,SP
4740     401$: BR       19$
4741
4742     402$: CMP      (SP),#15
4743     403$: BNE      16$
4744     404$: TST      4(SP)
4745     405$: BEQ      11$
4746     406$: MOV      2(SP),@SWR
4747     407$: ADD      #6,SP
4748     408$: TYPE      ,$CRLF
4749     409$: CMPB     $INTAG,#1
4750     410$: BNE      15$
4751     411$: MOV      #100,@STKS
4752     412$: RTI
4753     413$: JSR      PC,$TYPEOC
4754
4755     414$: TSTB     @STKS
4756     415$: BPL      7$
4757     416$: MOVB     @STKB,-(SP)
4758     417$: BIC      #1C177,(SP)
4759
4760     418$: CMPB     $AUTOB,#1
4761     419$: BEQ      25$
4762     420$: TST      (SP)+
4763     421$: JSR      PC,$TKINT
4764     422$: CLR      @STKS
4765     423$: MOVB     #1,$INTAG
4766
4767     424$: TYPE      ,$CNTLG
4768     425$: TYPE      ,$MSWR
4769     426$: MOV      $SWREG,-(SP)
4770     427$: TYPEOC    ,$MNEW
4771     428$: TYPE      ,(SP)
4772     429$: CLR      -(SP)
4773     430$: CLR      -(SP)
4774     431$: TSTB     @STKS
4775     432$: BPL      7$
4776     433$: MOVB     @STKB,-(SP)
4777     434$: BIC      #1C177,(SP)
4778
4779     435$: CMP      (SP),#25
4780     436$: BNE      10$
4781     437$: TYPE      ,$CNTLU
4782     438$: ADD      #6,SP
4783     439$: BR       19$
4784
4785     440$: CMP      (SP),#15
4786     441$: BNE      16$
4787     442$: TST      4(SP)
4788     443$: BEQ      11$
4789     444$: MOV      2(SP),@SWR
4790     445$: ADD      #6,SP
4791     446$: TYPE      ,$CRLF
4792     447$: CMPB     $INTAG,#1
4793     448$: BNE      15$
4794     449$: MOV      #100,@STKS
4795     450$: RTI
4796     451$: JSR      PC,$TYPEOC
4797
4798     452$: TSTB     @STKS
4799     453$: BPL      7$
4800     454$: MOVB     @STKB,-(SP)
4801     455$: BIC      #1C177,(SP)
4802
4803     456$: CMPB     $AUTOB,#1
4804     457$: BEQ      25$
4805     458$: TST      (SP)+
4806     459$: JSR      PC,$TKINT
4807     460$: CLR      @STKS
4808     461$: MOVB     #1,$INTAG
4809
4810     462$: TYPE      ,$CNTLG
4811     463$: TYPE      ,$MSWR
4812     464$: MOV      $SWREG,-(SP)
4813     465$: TYPEOC    ,$MNEW
4814     466$: TYPE      ,(SP)
4815     467$: CLR      -(SP)
4816     468$: CLR      -(SP)
4817     469$: TSTB     @STKS
4818     470$: BPL      7$
4819     471$: MOVB     @STKB,-(SP)
4820     472$: BIC      #1C177,(SP)
4821
4822     473$: CMP      (SP),#25
4823     474$: BNE      10$
4824     475$: TYPE      ,$CNTLU
4825     476$: ADD      #6,SP
4826     477$: BR       19$
4827
4828     478$: CMP      (SP),#15
4829     479$: BNE      16$
4830     480$: TST      4(SP)
4831     481$: BEQ      11$
4832     482$: MOV      2(SP),@SWR
4833     483$: ADD      #6,SP
4834     484$: TYPE      ,$CRLF
4835     485$: CMPB     $INTAG,#1
4836     486$: BNE      15$
4837     487$: MOV      #100,@STKS
4838     488$: RTI
4839     489$: JSR      PC,$TYPEOC
4840
4841     490$: TSTB     @STKS
4842     491$: BPL      7$
4843     492$: MOVB     @STKB,-(SP)
4844     493$: BIC      #1C177,(SP)
4845
4846     494$: CMPB     $AUTOB,#1
4847     495$: BEQ      25$
4848     496$: TST      (SP)+
4849     497$: JSR      PC,$TKINT
4850     498$: CLR      @STKS
4851     499$: MOVB     #1,$INTAG
4852
4853     500$: TYPE      ,$CNTLG
4854     501$: TYPE      ,$MSWR
4855     502$: MOV      $SWREG,-(SP)
4856     503$: TYPEOC    ,$MNEW
4857     504$: TYPE      ,(SP)
4858     505$: CLR      -(SP)
4859     506$: CLR      -(SP)
4860     507$: TSTB     @STKS
4861     508$: BPL      7$
4862     509$: MOVB     @STKB,-(SP)
4863     510$: BIC      #1C177,(SP)
4864
4865     511$: CMP      (SP),#25
4866     512$: BNE      10$
4867     513$: TYPE      ,$CNTLU
4868     514$: ADD      #6,SP
4869     515$: BR       19$
4870
4871     516$: CMP      (SP),#15
4872     517$: BNE      16$
4873     518$: TST      4(SP)
4874     519$: BEQ      11$
4875     520$: MOV      2(SP),@SWR
4876     521$: ADD      #6,SP
4877     522$: TYPE      ,$CRLF
4878     523$: CMPB     $INTAG,#1
4879     524$: BNE      15$
4880     525$: MOV      #100,@STKS
4881     526$: RTI
4882     527$: JSR      PC,$TYPEOC
4883
4884     528$: TSTB     @STKS
4885     529$: BPL      7$
4886     530$: MOVB     @STKB,-(SP)
4887     531$: BIC      #1C177,(SP)
4888
4889     532$: CMPB     $AUTOB,#1
4890     533$: BEQ      25$
4891     534$: TST      (SP)+
4892     535$: JSR      PC,$TKINT
4893     536$: CLR      @STKS
4894     537$: MOVB     #1,$INTAG
4895
4896     538$: TYPE      ,$CNTLG
4897     539$: TYPE      ,$MSWR
4898     540$: MOV      $SWREG,-(SP)
4899     541$: TYPEOC    ,$MNEW
4900     542$: TYPE      ,(SP)
4901     543$: CLR      -(SP)
4902     544$: CLR      -(SP)
4903     545$: TSTB     @STKS
4904     546$: BPL      7$
4905     547$: MOVB     @STKB,-(SP)
4906     548$: BIC      #1C177,(SP)
4907
4908     549$: CMP      (SP),#25
4909     550$: BNE      10$
4910     551$: TYPE      ,$CNTLU
4911     552$: ADD      #6,SP
4912     553$: BR       19$
4913
4914     554$: CMP      (SP),#15
4915     555$: BNE      16$
4916     556$: TST      4(SP)
4917     557$: BEQ      11$
4918     558$: MOV      2(SP),@SWR
4919     559$: ADD      #6,SP
4920     560$: TYPE      ,$CRLF
4921     561$: CMPB     $INTAG,#1
4922     562$: BNE      15$
4923     563$: MOV      #100,@STKS
4924     564$: RTI
4925     565$: JSR      PC,$TYPEOC
4926
4927     566$: TSTB     @STKS
4928     567$: BPL      7$
4929     568$: MOVB     @STKB,-(SP)
4930     569$: BIC      #1C177,(SP)
4931
4932     570$: CMPB     $AUTOB,#1
4933     571$: BEQ      25$
4934     572$: TST      (SP)+
4935     573$: JSR      PC,$TKINT
4936     574$: CLR      @STKS
4937     575$: MOVB     #1,$INTAG
4938
4939     576$: TYPE      ,$CNTLG
4940     577$: TYPE      ,$MSWR
4941     578$: MOV      $SWREG,-(SP)
4942     579$: TYPEOC    ,$MNEW
4943     580$: TYPE      ,(SP)
4944     581$: CLR      -(SP)
4945     582$: CLR      -(SP)
4946     583$: TSTB     @STKS
4947     584$: BPL      7$
4948     585$: MOVB     @STKB,-(SP)
4949     586$: BIC      #1C177,(SP)
4950
4951     587$: CMP      (SP),#25
4952     588$: BNE      10$
4953     589$: TYPE      ,$CNTLU
4954     590$: ADD      #6,SP
4955     591$: BR       19$
4956
4957     592$: CMP      (SP),#15
4958     593$: BNE      16$
4959     594$: TST      4(SP)
4960     595$: BEQ      11$
4961     596$: MOV      2(SP),@SWR
4962     597$: ADD      #6,SP
4963     598$: TYPE      ,$CRLF
4964     599$: CMPB     $INTAG,#1
4965     600$: BNE      15$
4966     601$: MOV      #100,@STKS
4967     602$: RTI
4968     603$: JSR      PC,$TYPEOC
4969
4970     604$: TSTB     @STKS
4971     605$: BPL      7$
4972     606$: MOVB     @STKB,-(SP)
4973     607$: BIC      #1C177,(SP)
4974
4975     608$: CMPB     $AUTOB,#1
4976     609$: BEQ      25$
4977     610$: TST      (SP)+
4978     611$: JSR      PC,$TKINT
4979     612$: CLR      @STKS
4980     613$: MOVB     #1,$INTAG
4981
4982     614$: TYPE      ,$CNTLG
4983     615$: TYPE      ,$MSWR
4984     616$: MOV      $SWREG,-(
```

F08

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 80
 DZRKLD.P11 31-AUG-76 15:35 TTY INPUT ROUTINE

SEQ 0096

```

4298 021226 021627 000060      CMP      (SP),#60      ;;CHAR < 0?
4299 021232 002420      BLT      18$      ;;BRANCH IF YES
4300 021234 021627 000067      CMP      (SP),#67      ;;CHAR > 7?
4301 021240 003015      BGT      18$      ;;BRANCH IF YES
4302 021242 042726 000060      BIC      #60,(SP)+    ;;STRIP-OFF ASCII
4303 021246 005766 000002      TST      2(SP)      ;;IS THIS THE FIRST CHAR
4304 021252 001403      BEQ      17$      ;;BRANCH IF YES
4305 021254 006316      ASL      (SP)      ;;NO, SHIFT PRESENT
4306 021256 006316      ASL      (SP)      ;;CHAR OVER TO MAKE
4307 021260 006316      ASL      (SP)      ;;ROOM FOR NEW ONE.
4308 021262 005266 000002 17$: INC      2(SP)      ;;KEEP COUNT OF CHAR
4309 021266 056616 177776      BIS      -2(SP),(SP)  ;;SET IN NEW CHAR
4310 021272 000707      BR       7$      ;;GET THE NEXT ONE
4311 021274 104401 001212 18$: TYPE    $QUES      ;;TYPE ?(CR)<LF>
4312 021300 000720      BR       20$      ;;SIMULATE CONTROL-U
      .DSABL  LSB

```

```

*****
*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
*CALL:
*      RDCHR      ;;GET A CHARACTER FROM THE QUEUE
*      RETURN HERE ;;CHARACTER IS ON THE STACK
*                  ;;WITH PARITY BIT STRIPPED OFF

```

```

4320 021302 011646 000004 000002 $RDCHR: MOV      (SP),-(SP)      ;;PUSH DOWN THE PC AND
4321 021304 016666 000004      MOV      4(SP),2(SP)      ;;THE PS
4322 021312 005066 000004      CLR      4(SP)      ;;GET READY FOR A CHARACTER
4323 021316 005046      CLR      -(SP)      ;;PUT NEW PS ON STACK
4324 021320 012746 021326      MOV      #64$,-(SP)      ;;PUT NEW PC ON STACK
4325 021324 000002      RTI      ;;POP NEW PC AND PS
4326 021326 005737 020526 64$: TST      $TKCNT      ;;WAIT ON A CHARACTER
4327 021332 001775      BEQ      1$      ;;
4328 021334 005337 020526 1$: DEC      $TKCNT      ;;DECREMENT THE COUNTER
4329 021340 117766 177166 000004      MOVB    2$TKQOUT,4(SP)  ;;GET ONE CHARACTER
4330 021346 005237 020532      INC      $TKQOUT      ;;UPDATE THE POINTER
4331 021352 023727 020532 020535      CMP      $TKQOUT,$$TKQEND ;;DID IT GO OFF OF THE END?
4332 021360 001003      BNE      2$      ;;BRANCH IF NO
4333 021362 012737 020534 020532      MOV      $$TKQSRST,$$TKQOUT ;;RESET THE POINTER
4334 021370 000002      RTI      ;;RETURN

```

```

*****
*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
*CALL:
*      RDLIN      ;;INPUT A STRING FROM THE TTY
*      RETURN HERE ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
*                  ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

```

```

4340 021372 010346 021626 $RDLIN: MOV      R3,-(SP)      ;;SAVE R3
4341 021374 005046      CLR      -(SP)      ;;CLEAR THE RUBOUT KEY
4342 021376 012703 021636 1$: MOV      $$TTYIN,R3      ;;GET ADDRESS
4343 021402 022703      CMP      $$TTYIN+8.,R3      ;;BUFFER FULL?
4344 021406 101456      BLOS      4$      ;;BR IF YES
4345 021410 104410      RDCHR      ;;GO READ ONE CHARACTER FROM THE TTY
4346 021412 112613      MOVB      (SP)+,(R3)      ;;GET CHARACTER

```


G08

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 81
 DZRKLD.P11 31-AUG-76 15:35 TTY INPUT ROUTINE

SEQ 0097

```

4354 021414 122713 000177      10$: CMPB    #177,(R3)      ;; IS IT A RUBOUT
4355 021420 001022                BNE     5$              ;; BR IF NO
4356 021422 005716                TST     (SP)             ;; IS THIS THE FIRST RUBOUT?
4357 021424 001007                BNE     6$              ;; BR IF NO
4358 021426 112737 000134 021624  MOVB    #'\\,9$      ;; TYPE A BACK SLASH
4359 021434 104401 021624                TYPE    9$
4360 021440 012716 177777                MOV     #-1,(SP)    ;; SET THE RUBOUT KEY
4361 021444 005303                DEC     R3              ;; BACKUP BY ONE
4362 021446 020327 021626      6$:  CMP     R3,#$TTYIN    ;; STACK EMPTY?
4363 021452 103434                BLO     4$              ;; BR IF YES
4364 021454 111337 021624                MOVB    (R3),9$    ;; SETUP TO TYPEOUT THE DELETED CHAR.
4365 021460 104401 021624                TYPE    9$
4366 021464 000746                BR      2$              ;; GO TYPE
4367 021466 005716                TST     (SP)             ;; GO READ ANOTHER CHAR.
4368 021470 001406                BEQ     7$              ;; RUBOUT KEY SET?
4369 021472 112737 000134 021624  MOVB    #'\\,9$      ;; BR IF NO
4370 021500 104401 021624                TYPE    9$      ;; TYPE A BACK SLASH
4371 021504 005016                CLR     (SP)
4372 021506 122713 000025      7$:  CMPB    #25,(R3)    ;; CLEAR THE RUBOUT KEY
4373 021512 001003                BNE     8$              ;; IS CHARACTER A CTRL U?
4374 021514 104401 021636                TYPE    $CNTLU    ;; BR IF NO
4375 021520 000726                BR      1$              ;; TYPE A CONTROL "U"
4376 021522 122713 000022      8$:  CMPB    #22,(R3)    ;; GO START OVER
4377 021526 001011                BNE     3$              ;; IS CHARACTER A "r"?
4378 021530 105013                CLRB    (R3)          ;; BRANCH IF NO
4379 021532 104401 001213                TYPE    $CRLF    ;; CLEAR THE CHARACTER
4380 021536 104401 021626                TYPE    $TTYIN    ;; TYPE A "CR" & "LF"
4381 021542 000717                BR      2$              ;; TYPE THE INPUT STRING
4382 021544 104401 001212      4$:  TYPE    $QUES    ;; GO PICKUP ANOTHER CHARACTER
4383 021550 000712                BR      1$              ;; TYPE A '?'
4384 021552 111337 021624      3$:  MOVB    (R3),9$    ;; CLEAR THE BUFFER AND LOOP
4385 021556 104401 021624                TYPE    9$      ;; ECHO THE CHARACTER
4386 021562 122723 000015                CMPB    #15,(R3)+
4387 021566 001305                BNE     2$              ;; CHECK FOR RETURN
4388 021570 105063 177777                CLRB    -1(R3)    ;; LOOP IF NOT RETURN
4389 021574 104401 001214                TYPE    $LF      ;; CLEAR RETURN (THE 15)
4390 021600 005726                TST     (SP)+
4391 021602 012603                MOV     (SP)+,R3    ;; TYPE A LINE FEED
4392 021604 011646                MOV     (SP),-(SP)  ;; CLEAN RUBOUT KEY FROM THE STACK
4393 021606 016666 000004 000002  MOV     4(SP),2(SP)  ;; RESTORE R3
4394 021614 012766 021626 000004  MOV     #$TTYIN,4(SP)  ;; ADJUST THE STACK AND PUT ADDRESS OF THE
4395 021622 000002                RTI                      ;; FIRST ASCII CHARACTER ON IT
4396 021624 000                9$:  .BYTE    0
4397 021625 000                .BYTE    0
4398 021626 000010                $TTYIN: .BLKB    8
4399 021636 052536 005015 000      $CNTLU: .ASCIZ  /?U<15><12>
4400 021643 136 006507 000012  $CNTLG: .ASCIZ  /?G<15><12>
4401 021650 005015 053523 020122  $MSWR: .ASCIZ  <15><12>/SWR = /
4402 021656 020075 000
4403 021661 040 047040 053505  $MNEW: .ASCIZ  / NEW = /
4404 021666 036440 000040
4405
4406 .SBTTL READ AN OCTAL NUMBER FROM THE TTY
4407
4408 ;*****
4409 ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND

```

H08

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 92
 DZRKLD.P11 31-AUG-76 15:35 READ AN OCTAL NUMBER FROM THE TTY

SEQ 0098

```

4410      ;*CHANGE IT TO BINARY.
4411      ;*CALL:
4412      ;*      RDOCT
4413      ;*      RETURN HERE
4414      ;*
4415      ;*      READ AN OCTAL NUMBER
4416      ;*      LOW ORDER BITS ARE ON TOP OF THE STACK
4417      ;*      HIGH ORDER BITS ARE IN $HIOCT
4418
4419      021672 011646 000004 000002 $RDOCT: MOV      (SP),-(SP)      ;* PROVIDE SPACE FOR THE
4420      021674 016666      MOV      4(SP),2(SP)      ;* INPUT NUMBER
4421      021702 010046      MOV      R0,-(SP)      ;* PUSH R0 ON STACK
4422      021704 010146      MOV      R1,-(SP)      ;* PUSH R1 ON STACK
4423      021706 010246      MOV      R2,-(SP)      ;* PUSH R2 ON STACK
4424      021710 104411 1$: RDLIN      ;* READ AN ASCII LINE
4425      021712 012600      MOV      (SP)+,R0      ;* GET ADDRESS OF 1ST CHARACTER
4426      021714 005001      CLR      R1      ;* CLEAR DATA WORD
4427      021716 005002      CLR      R2
4428      021720 112046 2$: MOVB      (R0)+,-(SP)      ;* PICKUP THIS CHARACTER
4429      021722 001412      BEQ      3$      ;* IF ZERO GET OUT
4430      021724 006301      ASL      R1      ;* *2
4431      021726 006102      ROL      R2
4432      021730 006301      ASL      R1      ;* *4
4433      021732 006102      ROL      R2
4434      021734 006301      ASL      R1      ;* *8
4435      021736 006102      ROL      R2
4436      021740 042716 177770 BIC      #1C7,(SP)      ;* STRIP THE ASCII JUNK
4437      021744 062601      ADD      (SP)+,R1      ;* ADD IN THIS DIGIT
4438      021746 000764      BR      2$      ;* LOOP
4439      021750 005726 3$: TST      (SP)+      ;* CLEAN TERMINATOR FROM STACK
4440      021752 010166      MOV      R1,12(SP)      ;* SAVE THE RESULT
4441      021756 010237      MOV      R2,$HIOCT
4442      021762 012602      MOV      (SP)+,R2      ;* POP STACK INTO R2
4443      021764 012601      MOV      (SP)+,R1      ;* POP STACK INTO R1
4444      021766 012600      MOV      (SP)+,R0      ;* POP STACK INTO R0
4445      021770 000002      RTI      ;* RETURN
4446      021772 000000 $HIOCT: .WORD      0      ;* HIGH ORDER BITS GO HERE
4447
4448      .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
4449
4450      ;******
4451      ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
4452      ;*OCTAL (ASCII) NUMBER AND TYPE IT.
4453      ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
4454      ;*CALL:
4455      ;*      MOV      NUM,-(SP)      ;* NUMBER TO BE TYPED
4456      ;*      TYPOS      ;* CALL FOR TYPEOUT
4457      ;*      .BYTE      N      ;* N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
4458      ;*      .BYTE      M      ;* M=1 OR 0
4459      ;*      ;* 1=TYPE LEADING ZEROS
4460      ;*      ;* 0=SUPPRESS LEADING ZEROS
4461      ;*
4462      ;*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
4463      ;*$TYPOS OR $TYPOC
4464      ;*CALL:
4465      ;*      MOV      NUM,-(SP)      ;* NUMBER TO BE TYPED
4466      ;*      TYPON      ;* CALL FOR TYPEOUT
4467      ;*
4468      ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER

```


MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 93
 DZRKLD.P11 31-AUG-76 15:35 BINARY TO OCTAL (ASCII) AND TYPE

SEQ 0099

```

4466      ;*CALL:
4467      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
4468      ;*      TYPOC      ;;CALL FOR TYPEOUT
4469
4470      021774 017646 000000      $TYPOS: MOV      2(SP),-(SP)      ;;PICKUP THE MODE
4471      022000 116637 000001 022217      MOV      1(SP),$OFILL      ;;LOAD ZERO FILL SWITCH
4472      022006 112637 022221      MOV      (SP)+,$OMODE+1      ;;NUMBER OF DIGITS TO TYPE
4473      022012 062716 000002      ADD      #2,(SP)      ;;ADJUST RETURN ADDRESS
4474      022016 000406      BR      $TYPON
4475      022020 112737 000001 022217      $TYPOC: MOV      #1,$OFILL      ;;SET THE ZERO FILL SWITCH
4476      022026 112737 000006 022221      MOV      #6,$OMODE+1      ;;SET FOR SIX(6) DIGITS
4477      022034 112737 000005 022216      $TYPON: MOV      #5,$OCNT      ;;SET THE ITERATION COUNT
4478      022042 010346      MOV      R3,-(SP)      ;;SAVE R3
4479      022044 010446      MOV      R4,-(SP)      ;;SAVE R4
4480      022046 010546      MOV      R5,-(SP)      ;;SAVE R5
4481      022050 113704 022221      MOV      $OMODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
4482      022054 005404      NEG      R4
4483      022056 062704 000006      ADD      #6,R4      ;;SUBTRACT IT FOR MAX. ALLOWED
4484      022062 110437 022220      MOV      R4,$OMODE      ;;SAVE IT FOR USE
4485      022066 113704 022217      MOV      $OFILL,R4      ;;GET THE ZERO FILL SWITCH
4486      022072 016605 000012      MOV      12(SP),R5      ;;PICKUP THE INPUT NUMBER
4487      022076 005003      CLR      R3      ;;CLEAR THE OUTPUT WORD
4488      022100 006105      1$:      ROL      R5      ;;ROTATE MSB INTO "C"
4489      022102 000404      BR      3$      ;;GO DO MSB
4490      022104 006105      2$:      ROL      R5      ;;FORM THIS DIGIT
4491      022106 006105      ROL      R5
4492      022110 006105      ROL      R5
4493      022112 010503      MOV      R5,R3
4494      022114 006103      3$:      ROL      R3      ;;GET LSB OF THIS DIGIT
4495      022116 105337 022220      DEC      $OMODE      ;;TYPE THIS DIGIT?
4496      022122 100016      BPL      7$      ;;BR IF NO
4497      022124 042703 177770      BIC      #177770,R3      ;;GET RID OF JUNK
4498      022130 001002      BNE      4$      ;;TEST FOR 0
4499      022132 005704      TST      R4      ;;SUPPRESS THIS 0?
4500      022134 001403      BEQ      5$      ;;BR IF YES
4501      022136 005204      4$:      INC      R4      ;;DON'T SUPPRESS ANYMORE 0'S
4502      022140 052703 000060      BIS      #'0,R3      ;;MAKE THIS DIGIT ASCII
4503      022144 052703 000040      5$:      BIS      #' ,R3      ;;MAKE ASCII IF NOT ALREADY
4504      022150 110337 022214      MOV      R3,$$      ;;SAVE FOR TYPING
4505      022154 104401 022214      TYPE      ,R$      ;;GO TYPE THIS DIGIT
4506      022160 105337 022216      7$:      DEC      $OCNT      ;;COUNT BY 1
4507      022164 003347      BGT      2$      ;;BR IF MORE TO DO
4508      022166 002402      BLT      6$      ;;BR IF DONE
4509      022170 005204      INC      R4      ;;INSURE LAST DIGIT ISN'T A BLANK
4510      022172 000744      BR      2$      ;;GO DO THE LAST DIGIT
4511      022174 012605      6$:      MOV      (SP)+,R5      ;;RESTORE R5
4512      022176 012604      MOV      (SP)+,R4      ;;RESTORE R4
4513      022200 012603      MOV      (SP)+,R3      ;;RESTORE R3
4514      022202 016666 000002 000004      MOV      2(SP),4(SP)      ;;SET THE STACK FOR RETURNING
4515      022210 012616      MOV      (SP)+,(SP)
4516      022212 000002      RTI
4517      022214      8$:      .BYTE      0      ;;RETURN
4518      022215      .BYTE      0      ;;STORAGE FOR ASCII DIGIT
4519      022216      .BYTE      0      ;;TERMINATOR FOR TYPE ROUTINE
4520      022217      .BYTE      0      ;;OCTAL DIGIT COUNTER
4521      022220 000000      $OCNT:      .WORD      0      ;;ZERO FILL SWITCH
      $OFILL:      .WORD      0      ;;NUMBER OF DIGITS TO TYPE
      $OMODE:

```

4522
 4523
 4524
 4525
 4526
 4527
 4528
 4529
 4530
 4531
 4532
 4533
 4534
 4535
 4536
 4537
 4538
 4539
 4540
 4541
 4542
 4543
 4544
 4545
 4546
 4547
 4548
 4549
 4550
 4551
 4552
 4553
 4554
 4555
 4556
 4557
 4558
 4559
 4560
 4561
 4562
 4563
 4564
 4565
 4566
 4567
 4568
 4569
 4570
 4571
 4572
 4573
 4574
 4575
 4576
 4577

.SBTTL TYPDSS - TYPE DECIMAL, LEADING ZEROES SUPPRESSED

```

: TYPDSS
: ROUTINE FOR TYPING OUT DECIMAL NUMBERS, LEADING 0'S ARE SUPPRESSED
: THE NUMBER IS LEFT JUSTIFIED. NOTE THE 16 BIT BINARY NUMBER SHOULD
: BE POSITIVE (BIT 15= 0).
: CALL: MOV    NUMBER, -(SP)    ; PUT BINARY NUMBER ON STACK
:        TYPDSS                ; GO TYPE DECIMAL

```

```

022222 016637 000004 022262 TYPDES: MOV    4(SP), 1$    ; GET THE NUMBER
022230 012746 022262        MOV    #1$, -(SP)        ; PUT PTR ON THE STACK
022234 004737 022422        JSR    PC, @#$DB2D        ; GO CONVERT BINARY NO. TO
                                ; ASCII STRING
022240 004737 022266        JSR    PC, @#$SUPRS        ; GO TYPE OUT DECIMAL STRING
                                ; SUPRESING LEADING 0'S
022244 016666 000002 000004        MOV    2(SP), 4(SP)    ; ADJUST RETURN
022252 011666 000002        MOV    (SP), 2(SP)        ; ADJUST RETURN ADRES
022256 005726                TST    (SP)+            ; POP STACK
022260 000002                RTI                      ; RETURN

022262 000000 000000 1$: .WORD 0,0

```

.SBTTL TYPE NUMERICAL ASCII STRING SUPPRESS LEADING ZEROS

```

: *****
: *THIS ROUTINE IS USED TO TYPE AN ASCII NUMBER SUPPRESSING THE
: *LEADING NUMBERS.
: *CALL
: *   MOV    #NUMADR, -(SP)    ;; FIRST ADDRESS OF ASCII STRING
: *   JSR    PC, @#$SUPRS

```

```

022266 010046                $SUPRS: MOV    RO, -(SP)    ;; SAVE RO
022270 016600 000004        MOV    4(SP), RO        ;; PICKUP THE POINTER
022274 105710 1$: TSTB      (RO)                ;; TERMINATE OR?
022276 001403                BEQ    2$                ;; BR IF YES
022300 122720 000060        CMPB    #'0, (RO)+        ;; IS THIS AN ASCII "0" ?
022304 001773                BEQ    1$                ;; BR IF YES
022306 005300 2$: DEC      RO                ;; BACKUP BY "1"
022310 010037 022316        MOV    RO, 3$            ;; SAVE FOR TYPING
022314 104401                TYPE                ;; GO TYPE
022316 000000 3$: .WORD    0                ;; ASCII POINTER GOES HERE
022320 012600                MOV    (SP)+, RO        ;; RESTORE RO
022322 012616                MOV    (SP)+, (SP)        ;; RESTORE THE STACK
022324 000207                RTS    PC                ;; RETURN

```

.SBTTL SAVE AND RESTORE RO-R5 ROUTINES

```

: *****
: *SAVE RO-R5
: *CALL:
: *   SAVREG
: *UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
: *

```


K08

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 85
 DZRKLD.P11 31-AUG-76 15:35 SAVE AND RESTORE R0-R5 ROUTINES

SEQ 0101

```

4578      ;*TOP---(+16)
4579      ;* +2---(+18)
4580      ;* +4---R5
4581      ;* +6---R4
4582      ;* +8---R3
4583      ;*+10---R2
4584      ;*+12---R1
4585      ;*+14---R0
4586
4587      $SAVREG:
4588      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
4589      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
4590      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
4591      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
4592      MOV      R4,-(SP)      ;;PUSH R4 ON STACK
4593      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
4594      MOV      22(SP),-(SP)   ;;SAVE PS OF MAIN FLOW
4595      MOV      22(SP),-(SP)   ;;SAVE PC OF MAIN FLOW
4596      MOV      22(SP),-(SP)   ;;SAVE PS OF CALL
4597      MOV      22(SP),-(SP)   ;;SAVE PC OF CALL
4598      RTI
4599
4600      ;*RESTORE R0-R5
4601      ;*CALL:
4602      ;*      RESREG
4603      $RESREG:
4604      MOV      (SP)+,22(SP)   ;;RESTORE PC OF CALL
4605      MOV      (SP)+,22(SP)   ;;RESTORE PS OF CALL
4606      MOV      (SP)+,22(SP)   ;;RESTORE PC OF MAIN FLOW
4607      MOV      (SP)+,22(SP)   ;;RESTORE PS OF MAIN FLOW
4608      MOV      (SP)+,R5
4609      MOV      (SP)+,R4
4610      MOV      (SP)+,R3
4611      MOV      (SP)+,R2
4612      MOV      (SP)+,R1
4613      MOV      (SP)+,R0
4614      RTI
4615
4616      .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
4617
4618      ;*****
4619      ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
4620      ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
4621      ;*POSITIVE.
4622      ;*CALL
4623      ;*      MOV      #PNTR,-(SP)      ;;POINTER TO LOW WORD OF BINARY NUMBER
4624      ;*      JSR      PC,@#$DB2D
4625      ;*      RETURN
4626      ;*      ;;THE FIRST ADDRESS OF ASCII
4627      ;*      ;;IS ON THE STACK
4628
4629      $DB2D:  SAVREG
4630      MOV      2(SP),R2      ;;SAVE REGISTERS
4631      MOV      #$DECVL,R0    ;;PICKUP THE DATA POINTER
4632      MOV      R0,2(SP)      ;;GET ADDRESS OF "$DECVL" STRING
4633      MOV      (R2)+,R1      ;;PUT ADDRESS OF ASCII STRING ON STACK
4634      ;;PICKUP THE BINARY NUMBER

```

L08

MAINDEC-11-DZRKL-D

MACY11 27(1006)

04-OCT-76 14:26 PAGE 96

DZRKL0.P11

31-AUG-76 15:35

DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE

SEQ 0102

```

4634 022442 012202      MOV      (R2)+,R2
4635 022444 012737 000012 022520      MOV      #10.,4$      ;;SET UP TO DO 10 CONVERSIONS
4636 022452 012704 022532      MOV      $TNPWR,R4      ;;ADDRESS OF TEN POWER
4637 022456 012705 022534      MOV      $TNPWR+2,R5
4638 022462 005003      1$:      CLR      R3      ;;CLEAR PARTIAL
4639 022464 161401      2$:      SUB      (R4),R1      ;;SUBTRACT TEN POWER
4640 022466 005602      SBC      R2
4641 022470 161502      SUB      (R5),R2
4642 022472 002402      BLT      3$      ;;BR IF TEN POWER TO LARGE
4643 022474 005203      INC      R3      ;;ADD 1 TO PARTIAL
4644 022476 000772      BR       2$      ;;LOOP
4645 022500 062401      3$:      ADD      (R4)+,R1      ;;RESTORE SUBTRACTED VALUE
4646 022502 005502      ADC      R2
4647 022504 062402      ADD      (R4)+,R2
4648 022506 022525      CMP      (R5)+,(R5)+      ;;MOVE TO NEXT TEN POWER
4649 022510 052703 000060      BIS      #'0,R3      ;;CHANGE PARTIAL TO ASCII
4650 022514 110320      MOVB     R3,(R0)+      ;;SAVE IT
4651 022516 005327      DEC      (PC)+      ;;DONE?
4652 022520 000000      4$:      .WORD    0
4653 022522 001357      BNE      1$      ;;BR IF NO
4654 022524 105020      CLRB     (R0)+      ;;TERMINATOR
4655 022526 104414      RESREG
4656 022530 000207      RTS      PC      ;;RESTORE REGISTERS
4657 022532 145000      $TNPWR: 145000      ;;RETURN
4658 022534 035632      35632      ;;1.0E09
4659 022536 160400      160400      ;;1.0E08
4660 022540 002765      2765      ;;1.0E07
4661 022542 113200      113200      ;;1.0E06
4662 022544 000230      230      ;;1.0E05
4663 022546 041100      041100      ;;1.0E04
4664 022550 000017      17      ;;1.0E03
4665 022552 103240      103240      ;;1.0E02
4666 022554 000001      1      ;;1.0E01
4667 022556 023420      23420      ;;1.0E00
4668 022560 000000      0
4669 022562 001750      1750,
4670 022564 000000      0
4671 022566 000144      144
4672 022570 000000      0
4673 022572 000012      12
4674 022574 000000      0
4675 022576 000001      1
4676 022600 000000      0
4677 022602 000014      $DECVL: .BLKB 12.      ;;RESERVE STORAGE FOR ASCII STRING
4678
4679
4680
4681      .SBTTL  TRAP DECODER
4682
4683      ;;*****
4684      ;;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
4685      ;;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
4686      ;;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
4687      ;;*GO TO THAT ROUTINE.
4688
4689 022616 010046      $TRAP:  MOV      R0,-(SP)      ;;SAVE R0

```


MO8

MAINDEC-11-DZRL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 87
 DZRLD.P11 31-AUG-76 15:35 TRAP DECODER

SEQ 0103

```

4690 022620 016600 000002      MOV      2(SP),RO      ;;GET TRAP ADDRESS
4691 022624 005740              TST      -(RO)          ;;BACKUP BY 2
4692 022626 111000              MOVVB   (RO),RO        ;;GET RIGHT BYTE OF TRAP
4693 022630 006300              ASL      RO              ;;POSITION FOR INDEXING
4694 022632 016000 022652      MOV      $TRPAD(RO),RO    ;;INDEX TO TABLE
4695 022636 000200              RTS      RO              ;;GO TO ROUTINE
4696
4697
4698
4699      ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
4700 022640 011646 000004 000002 $TRAP2: MOV      (SP),-(SP)      ;;MOVE THE PC DOWN
4701 022642 016666              MOV      4(SP),2(SP)      ;;MOVE THE PSW DOWN
4702 022650 000002              RTI                      ;;RESTORE THE PSW
4703
4704      .SBTTL  TRAP TABLE
4705
4706      ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
4707      ;*BY THE "TRAP" INSTRUCTION.
4708
4709      :      ROUTINE
4710      :      -----
4711 022652 022640 $TRPAD: .WORD      $TRAP2
4712 022654 020174      $TYPE      ;;CALL=TYPE      TRAP+1(104401) TTY TYPEOUT ROUTINE
4713 022656 022020      $TYPOC     ;;CALL=TYPOC     TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
4714 022660 021774      $TYPOS     ;;CALL=TYPOS     TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
4715 022662 022034      $TYPON     ;;CALL=TYPON     TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
4716 022664 017750      $TYPDS     ;;CALL=TYPDS     TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
4717
4718 022666 021070      $GTSWR      ;;CALL=GTSWR      TRAP+6(104406) GET SOFT-SWR SETTING
4719
4720 022670 021000      $CKSWR      ;;CALL=CKSWR      TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
4721 022672 021302      $RDCHR      ;;CALL=RDCHR      TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
4722 022674 021372      $RDLIN      ;;CALL=RDLIN      TRAP+11(104411) TTY TYPEIN STRING ROUTINE
4723 022676 021672      $RDOCT      ;;CALL=RDOCT      TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
4724 022700 022326      $SAVREG     ;;CALL=SAVREG     TRAP+13(104413) SAVE R0-R5 ROUTINE
4725 022702 022364      $RESREG     ;;CALL=RESREG     TRAP+14(104414) RESTORE R0-R5 ROUTINE
4726
4727 022704 016526      CN.RST      ;;CALL=CON.RESET    TRAP+15(104415) CONTROL RESET ROUTINE
4728
4729 022706 016414      DR.RST      ;;CALL=DRV.RESET    TRAP+16(104416) DRIVE RESET ROUTINE
4730
4731 022710 015770      MSGE        ;;CALL=MESSAGE    TRAP+17(104417) MESSAGE HANDLER
4732
4733 022712 016032      TY.MSG      ;;CALL=TYPMMSG    TRAP+20(104420) MESSAGE TYPEOUT ROUTINE
4734
4735 022714 016544      CN.RDY      ;;CALL=CON.RDY     TRAP+21(104421) WAIT FOR CONTROL READY
4736
4737 022716 016642      TSTRWS      ;;CALL=TST.RWS     TRAP+22(104422) TEST R/W/S RDY SET
4738
4739 022720 016440      RES.DO      ;;CALL=RESDON     TRAP+23(104423) DRIVE RESET DONE?
4740
4741 022722 022222      TYPDES      ;;CALL=TYPDSS     TRAP+24(104424) TYPE DECIMAL, SUPRES LDG 0'S
4742
4743
4744      .SBTTL  POWER DOWN AND UP ROUTINES
4745

```

N08

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 88
DZRKLD.P11 31-AUG-76 15:35 POWER DOWN AND UP ROUTINES

SEQ 0104

```

4746
4747
4748 022724 012737 023070 000024
4749 022732 012737 000340 000026
4750 022740 010046
4751 022742 010146
4752 022744 010246
4753 022746 010346
4754 022750 010446
4755 022752 010546
4756 022754 017746 156160
4757 022760 010637 023074
4758 022764 012737 022776 000024
4759 022772 000000
4760 022774 000776
4761
4762
4763
4764 022776 012737 023070 000024
4765 023004 013706 023074
4766 023010 005037 023074
4767 023014 005237 023074
4768 023020 001375
4769 023022 012677 156112
4770 023026 012605
4771 023030 012604
4772 023032 012603
4773 023034 012602
4774 023036 012601
4775 023040 012600
4776 023042 012737 022724 000024
4777 023050 012737 000340 000026
4778 023056 104401
4779 023060 023076
4780 023062 012716
4781 023064 004170
4782 023066 000002
4783 023070 000000
4784 023072 000776
4785 023074 000000
4786 023076 005015 047520 042527
4787 023104 000122
4788
4789
4790

*****
: POWER DOWN ROUTINE
$PWRDN: MOV $SILLUP, @PWRVEC ;; SET FOR FAST UP
        MOV #340, @PWRVEC+2 ;; PRIO:7
        MOV R0, -(SP) ;; PUSH R0 ON STACK
        MOV R1, -(SP) ;; PUSH R1 ON STACK
        MOV R2, -(SP) ;; PUSH R2 ON STACK
        MOV R3, -(SP) ;; PUSH R3 ON STACK
        MOV R4, -(SP) ;; PUSH R4 ON STACK
        MOV R5, -(SP) ;; PUSH R5 ON STACK
        MOV @SWR, -(SP) ;; PUSH @SWR ON STACK
        MOV SP, $SAVR6 ;; SAVE SP
        MOV $PWRUP, @PWRVEC ;; SET UP VECTOR
        HALT
        BR -2 ;; HANG UP

*****
: POWER UP ROUTINE
$PWRUP: MOV $SILLUP, @PWRVEC ;; SET FOR FAST DOWN
        MOV $SAVR6, SP ;; GET SP
        CLR $SAVR6 ;; WAIT LOOP FOR THE TTY
        IS: INC $SAVR6 ;; WAIT FOR THE INC
        BNE IS ;; OF WORD
        MOV (SP)+, @SWR ;; POP STACK INTO @SWR
        MOV (SP)+, R5 ;; POP STACK INTO R5
        MOV (SP)+, R4 ;; POP STACK INTO R4
        MOV (SP)+, R3 ;; POP STACK INTO R3
        MOV (SP)+, R2 ;; POP STACK INTO R2
        MOV (SP)+, R1 ;; POP STACK INTO R1
        MOV (SP)+, R0 ;; POP STACK INTO R0
        MOV $PWRDN, @PWRVEC ;; SET UP THE POWER DOWN VECTOR
        MOV #340, @PWRVEC+2 ;; PRIO:7
        TYPE $PWRMG: .WORD $POWER ;; REPORT THE POWER FAILURE
        MOV (PC)+, (SP) ;; POWER FAIL MESSAGE POINTER
        $PWRAD: .WORD PWRFL ;; RESTART AT PWRFL
        RTI ;; RESTART ADDRESS
        $SILLUP: HALT ;; THE POWER UP SEQUENCE WAS STARTED
        BR -2 ;; BEFORE THE POWER DOWN WAS COMPLETE
        $SAVR6: 0 ;; PUT THE SP HERE
        $POWER: .ASCIZ <15><12>"POWER"
                .EVEN

```


MAINDEC-11-DZAKL-D
DZAKLD.P11

MACY11 27(1006)
31-AUG-76 15:35

04-OCT-76 14:26 PAGE 89
FUNCTION SELECTION PROGRAM

SEQ 0105

.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 SW 0 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.

```

FUNBEG:

TYPE 655

BR 648

::655: .ASCIZ <15><12>/FUNCTION? /

648:

RDLIN

```

::TYPE ASCIZ STRING
::GET OVER THE ASCIZ

```

```

023106
023106 104401 023114
023112 000407
023132
023132 104411

```

C09

MAINDEC-11-DZAKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 90
 DZAKLD.P11 31-AUG-76 15:35 FUNCTION SELECTION PROGRAM

SEQ 0106

4847	023134	012600		MOV	(SP)+,R0	
4848	023136	112001		MOVB	(R0)+,R1	
4849	023140	120127	000127	CMPB	R1,#'W	
4850	023144	001026		BNE	2\$	
4851	023146	121027	000122	CMPB	(R0),#'R	
4852	023152	001010		BNE	1\$	
4853	023154	004737	024000	JSR	PC,CHKSWO	;CHECK SW 0 SET?
4854	023160	012702	004003	MOV	#4003,R2	
4855	023164	000536		BR	NXTDA	
4856						
4857	023166	012702	000003	9\$: MOV	#3,R2	
4858	023172	000521		BR	NXTBA	
4859	023174	121027	000103	1\$: CMPB	(R0),#'C	
4860	023200	001342		BNE	FUNBEG	
4861	023202	004737	024000	JSR	PC,CHKSWO	;CHECK SW 0 SET?
4862	023206	012702	004007	MOV	#4007,R2	
4863	023212	000523		BR	NXTDA	
4864						
4865	023214	012702	000007	MOV	#7,R2	
4866	023220	000506		BR	NXTBA	
4867						
4868	023222	120127	000122	2\$: CMPB	R1,#'R	
4869	023226	001014		BNE	3\$	
4870	023230	121027	000104	CMPB	(R0),#'D	
4871	023234	001003		BNE	8\$	
4872	023236	012702	000005	MOV	#5,R2	
4873	023242	000475		BR	NXTBA	
4874	023244	121027	000103	8\$: CMPB	(R0),#'C	
4875	023250	001316		BNE	FUNBEG	
4876	023252	012702	000013	MOV	#13,R2	
4877	023256	000501		BR	NXTDA	
4878						
4879	023260	120127	000104	3\$: CMPB	R1,#'D	
4880	023264	001006		BNE	4\$	
4881	023266	121027	000122	CMPB	(R0),#'R	
4882	023272	001305		BNE	FUNBEG	
4883	023274	012702	000015	MOV	#15,R2	
4884	023300	000470		BR	NXTDA	
4885						
4886	023302	120127	000103	4\$: CMPB	R1,#'C	
4887	023306	001006		BNE	5\$	
4888	023310	121027	000122	CMPB	(R0),#'R	
4889	023314	001274		BNE	FUNBEG	
4890	023316	012702	000001	MOV	#1,R2	
4891	023322	000533		BR	EXEC	
4892						
4893	023324	120127	000123	5\$: CMPB	R1,#'S	
4894	023330	001266		BNE	FUNBEG	
4895	023332	121027	000113	CMPB	(R0),#'K	
4896	023336	001263		BNE	FUNBEG	
4897	023340	012702	000011	MOV	#11,R2	
4898						
4899	023344			6\$: TYPE	#67\$	::TYPE ASCIZ STRING
4900	023344	104401	023352	BR	#6\$	::GET OVER THE ASCIZ
4901	023350	000404		::67\$: .ASCIZ	/CYL1? /	
4902						

MAINDEC-11-DZRLD-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 91
 DZRLD.P11 31-AUG-76 15:35 FUNCTION SELECTION PROGRAM

SEQ 0107

4903	023362			66\$:	JSR	PC, INPT	
4904	023362	004737	023746		BR	6\$	
4905	023366	000766			MOV	RO, R4	
4906	023370	010004					
4907				7\$:	TYPE	69\$	::TYPE ASCIZ STRING
4908	023372				BR	68\$	::GET OVER THE ASCIZ
4909	023372	104401	023400				
4910	023376	000404		::69\$:	.ASCIZ	/CYL2? /	
4911				68\$:	JSR	PC, INPT	
4912	023410				BR	7\$	
4913	023410	004737	023746		MOV	RO, R5	
4914	023414	000766			MOV	35\$R, RO	:GET DRIVE # FROM SW REG<15-13>
4915	023416	010005			BIC	#17777, RO	:CLR UNWANTED BITS
4916	023420	017700	155514		BIS	RO, R4	:SET DRIVE # IN DSK ADRES
4917	023424	042700	017777		BIS	RO, R5	
4918	023430	050004			BR	EXEC	
4919	023432	050005					
4920	023434	000466					
4921				NXTBA:	TYPE	65\$	::TYPE ASCIZ STRING
4922	023436				BR	64\$	::GET OVER THE ASCIZ
4923	023436	104401	023444				
4924	023442	000404		::65\$:	.ASCIZ	/RKBA? /	
4925				64\$:	RDOCT		
4926	023454				MOV	(SP)+, INDX2	
4927	023454	104412					
4928	023456	012637	001476				
4929				NXTDA:	TYPE	65\$	::TYPE ASCIZ STRING
4930	023462				BR	64\$	::GET OVER THE ASCIZ
4931	023462	104401	023470				
4932	023466	000404		::65\$:	.ASCIZ	/RKDA? /	
4933				64\$:	RDOCT		
4934	023500				MOV	(SP)+, RO	
4935	023500	104412			MOV	RO, R1	
4936	023502	012600			MOV	R1	
4937	023504	010001			ASR	R1	
4938	023506	006201			ASR	R1	
4939	023510	006201			ASR	R1	
4940	023512	006201			ASR	R1	
4941	023514	006201			ASR	R1	
4942	023516	006201			BIC	#177400, R1	
4943	023520	042701	177400		CMP	R1, #312	
4944	023524	020127	000312		BGT	NXTDA	
4945	023530	003354			MOV	RO, R1	
4946	023532	010001			BIC	#177760, R1	
4947	023534	042701	177760		CMP	R1, #13	
4948	023540	020127	000013		BGT	NXTDA	
4949	023544	003346			MOV	RO, R3	
4950	023546	010003			CMP	#15, R2	
4951	023550	022702	000015		BEQ	EXEC	
4952	023554	001416					
4953				NXTWC:	TYPE	65\$	::TYPE ASCIZ STRING
4954	023556				BR	64\$	::GET OVER THE ASCIZ
4955	023556	104401	023564				
4956	023562	000405					

E09

MAINDEC-11-DZRLD-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 92
 DZRLD.P11 31-AUG-76 15:35 FUNCTION SELECTION PROGRAM

SEQ 0108

```

4959      ;:65$: .ASCIZ /#WORDS? /
4960      64$:
4961      RDOCT
4962      NEG      (SP)
4963      MOV      (SP)+,INDX4
4964      BR       EXEC
4965
4966      023576 104412
4967      023576 005416
4968      023600 012637 001502
4969      023606 000401
4970
4971      023610 104415
4972      023612 022702 000011
4973      023616 001005
4974      023620 020403
4975      023622 001402
4976      023624 010403
4977      023626 000401
4978      023630 010503
4979      023632 013777 001476 155622
4980      023634 010377 155620
4981      023640 013777 001502 155606
4982      023644 010277 155600
4983      023652 105777 155574
4984      023656 100375
4985      023662 022702 000001
4986      023664 001401
4987      023670 104423
4988
4989      023674 032777 140000 155554
4990      023702 001006
4991      023704 032777 000010 155226
4992      023712 001737
4993      023714 000137 023106
4994
4995      023720 012737 023610 001110
4996      023726 012737 023610 001106
4997      023734 004737 016106
4998      023740 104030
4999      023742 104415
5000      023744 000757
5001
5002      023746 104412
5003      023750 012600
5004      023752 020027 000312
5005      023756 003007
5006      023760 006300
5007      023762 006300
5008      023764 006300
5009      023766 006300
5010      023770 006300
5011      023772 062716 000002
5012      023776 000207
5013
5014      024000 032777 000001 155132
5015      024006 001416
5016      024010 104401 024016
5017      024014 000404

      ;:65$: .ASCIZ /#WORDS? /
      64$:
      RDOCT
      NEG      (SP)
      MOV      (SP)+,INDX4
      BR       EXEC

EXEC1:  CON.RESET
EXEC:   CMP     #11,R2
      BNE      2$
      CMP     R4,R3
      BEQ      3$
      MOV     R4,R3
      BR      2$
      MOV     R5,R3
      MOV     INDX2,ARKBA
      MOV     R3,ARKDA
      MOV     INDX4,ARKWC
      MOV     R2,ARKCS
      TSTB    ARKCS
      BPL     1$
      CMP     #1,R2
      BEQ     4$
      RESDON

      3$:
      MOV     R5,R3
      MOV     INDX2,ARKBA
      MOV     R3,ARKDA
      MOV     INDX4,ARKWC
      MOV     R2,ARKCS
      TSTB    ARKCS
      BPL     1$
      CMP     #1,R2
      BEQ     4$
      RESDON

      1$:
      TSTB    ARKCS
      BPL     1$
      CMP     #1,R2
      BEQ     4$
      RESDON

      4$:
      BIT     #140000,ARKCS
      BNE     FUNERR
      CHSW:   BIT     #SW3,SWR
      BEQ     EXEC
      JMP     FUNBEG

      FUNERR: MOV     #EXEC1,$LPERR
      MOV     #EXEC1,$LPADR
      JSR     PC,GT4RG
      ERROR   30
      CON.RESET
      BR      CHSW

      INPT:   MOV     RDOCT
      MOV     (SP)+,R0
      CMP     R0,#312
      BGT     1$
      ASL     R0
      ASL     R0
      ASL     R0
      ASL     R0
      ASL     R0
      ADD     #2,(SP)
      RTS     PC

      1$:
      BIT     #SW0,SWR
      BEQ     1$
      TYPE    65$
      BR      64$

      ;CLR ERROR, CONTROL RESET
      ;SEEK FUNCTION?
      ;NO
      ;IF SEEK, INSERT THE RIGHT
      ;CYLINDER ADDRESS

      ;WAIT FOR CNTROL RDY
      ;IF IT'S CON RESET FUNCTION
      ;DONT WAIT FOR R/W/S RDY
      ;R/W/S RDY?

      ;ERROR?
      ;YES
      ;TERMINATE THIS FUNCTION OR REPEAT?
      ;REPEAT
      ;TERMINATE

      ;SET UP FOR LUPING
      ;REPORT ERROR
      ;CLR ERROR

      ;WRITE A PATTERN GIVEN BY USER?
      ;NO
      ;YES, ASK FOR PATTERN
      ;TYPE ASCIZ STRING
      ;GET OVER THE ASCIZ

```



```

5015      ;:65$: .ASCIZ /PATRN?/
5016      64$:
5017      RDOCT
5018      MOV      (SP)+,IOBUF1      ;SAVE THE PATTERN
5019      MOV      #IOBUF1,INDX2
5020      RTS      PC
5021      1$:      ADD      #6,(SP)      ;SKIP NXT 2 INST ON RETURN
5022      RTS      PC
5023
5024      FCHECK:  DRV.RESET
5025      CON.RESET
5026      MOV      DRIVAD,DRHOLD      ;SAVE DRIVE ADDR
5027      BIT      #20000,DRIVAD      ;SEE IF ODD
5028      BEQ      1$
5029      BIC      #20000,DRIVAD      ;MAKE EVEN
5030      BR       2$
5031      1$:      BIS      #20000,DRIVAD      ;MAKE ODD
5032      MOV      DRIVAD,ARKDA      ;DRIVE ADDR
5033      2$:      MOV      #11,ARKCS      ;DRIVE SEEK
5034      CON.RDY
5035      MOV      DRHOLD,ARKDA      ;OTHER DRIVE
5036      CON.RDY
5037      BIT      #100,ARKDS      ;HEADS IN MOTION?
5038      BNE      3$      ;NO SO RK-05J
5039      TST      (R5)+      ;YES RK-05F
5040      MOV      DRHOLD,DRIVAD      ;RESTORE ADDR
5041      DRV.RESET
5042      RTS      R5
5043      SIZEF:  CLR      DRIVAD      ;START AT DRO
5044      MOV      #DRIVO,RO      ;TABLE OF AVAIL DRIVES
5045      4$:      TSTB      (RO)      ;THIS DRIVE HERE?
5046      BEQ      2$      ;NO
5047      TSTB      2(RO)      ;COMPLEMENT HERE?
5048      BEQ      2$      ;NO
5049      JSR      R5,FCHECK      ;SEE IF F MODEL
5050      BR       2$      ;J MODEL
5051      BIS      #2,(RO)      ;SET SIGN FOR F
5052      BIS      #2,2(RO)      ;BOTH DRIVES
5053      2$:      TST      (RO)+
5054      TST      (RO)+      ;NEXT PAIR OF DRIVES
5055      ADD      #40000,DRIVAD      ;NEXT ACTUL ADDR
5056      CMP      #DRIV7+1,RO      ;CHECKED ALL?
5057      BGT      4$      ;NOT YET
5058      RTS      PC
5059
5060      ;ERROR MESSAGES
5061
5062
5063      EM1:      .ASCIZ /CNTRL RDY DIDN'T SET AFTR SEEK/
5064
5065
5066
5067
5068
5069
5070      EM2:      .ASCIZ /SIN ON SEEK/

```

G09

MAINDEC-11-DZRL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 94
 DZRLD.P11 31-AUG-76 15:35 FUNCTION SELECTION PROGRAM

SEQ 0110

5071	024314	020116	042523	045505	
5072	024322	000			
5073					
5074	024323	104	042522	047440	EM3: .ASCIZ /DRE ON SEEK/
5075	024330	020116	042523	045505	
5076	024336	000			
5077					
5078	024337	047	051105	023522	EM4: .ASCIZ /'ERR' ON SEEK/
5079	024344	047440	020116	042523	
5080	024352	045505	000		
5081					
5082	024355	104	052522	047440	EM5: .ASCIZ /DRU ON SEEK, PUT DRV ON 'LOAD' & 'RUN' /
5083	024362	020116	042523	045505	
5084	024370	020054	052520	020124	
5085	024376	051104	020126	047117	
5086	024404	023440	047514	042101	
5087	024412	020047	020046	051047	
5088	024420	047125	000047		
5089					
5090	024424	027522	027527	020123	EM6: .ASCIZ "R/W/S RDY NOT SET AFTER SEEK"
5091	024432	042122	020131	047516	
5092	024440	020124	042523	020124	
5093	024446	043101	042524	020122	
5094	024454	042523	045505	000	
5095					
5096	024461	123	047111	047440	EM7: .ASCIZ /SIN ON WRT FMT/
5097	024466	020116	051127	020124	
5098	024474	046506	000124		
5099					
5100	024500	042447	051122	020047	EM10: .ASCIZ /'ERR' ON DOING WRITE FMT/
5101	024506	047117	042040	044517	
5102	024514	043516	053440	044522	
5103	024522	042524	043040	052115	
5104	024530	000			
5105	024531	123	047111	047440	EM11: .ASCIZ "SIN ON RD/FMT"
5106	024536	020116	042122	043057	
5107	024544	052115	000		
5108	024547	047	051105	023522	EM12: .ASCIZ /'ERR' ON READ FMT/
5109	024554	047440	020116	042522	
5110	024562	042101	043040	052115	
5111	024570	000			
5112	024571	127	047522	043516	EM13: .ASCIZ /WRONG HDRS FROM 'SEC#' /
5113	024576	044040	051104	020123	
5114	024604	051106	046517	023440	
5115	024612	042523	021503	000047	
5116					
5117	024620	051105	051117	047440	EM14: .ASCIZ /EROR ON IMPLIED SEEK FROM 'CYLA' TO 'CYLB' /
5118	024626	020116	046511	046120	
5119	024634	042511	020104	042523	
5120	024642	045505	043040	047522	
5121	024650	020115	041447	046131	
5122	024656	023501	052040	020117	
5123	024664	041447	046131	023502	
5124	024672	000			
5125					
5126	024673	122	040505	020104	MS15: .ASCIZ /READ WRONG HDRS FROM 'CYLB' ABOVE/

H09

MAINDEC-11-DZRL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 95
 DZRLD.P11 31-AUG-76 15:35 FUNCTION SELECTION PROGRAM

SEQ 0111

127	024700	051127	047117	020107	
128	024706	042110	051522	043040	
129	024714	047522	020115	041447	
130	024722	046131	023502	040440	
131	024730	047502	042526	000	
132					
133	024735	122	040505	020104	EM16: .ASCIZ /READ WRONG, 1ST WRD FROM SEC 0, 'CYLB' (ON IMPLIED SEEK FROM 'CYLA')/
134	024742	051127	047117	026107	
135	024750	030440	052123	053440	
136	024756	042122	043040	047522	
137	024764	020115	042523	020103	
138	024772	026060	023440	054503	
139	025000	041114	020047	047450	
140	025006	020116	046511	046120	
141	025014	042511	020104	042523	
142	025022	045505	043040	047522	
143	025030	020115	041447	046131	
144	025036	023501	000051		
145					
146	025042	042522	042101	030440	MS17: .ASCIZ /READ 1ST WRD FROM SEC 1, 'CYLB' ABOVE/
147	025050	052123	053440	042122	
148	025056	043040	047522	020115	
149	025064	042523	020103	026061	
150	025072	023440	054503	041114	
151	025100	020047	041101	053117	
152	025106	000105			
153					
154	025110	042522	042101	053440	EM20: .ASCIZ /READ WRONG HDRS ON IMPLIED SEEK FROM 'CYLA' TO 'CYLB' /
155	025116	047522	043516	044040	
156	025124	051104	020123	047117	
157	025132	044440	050115	044514	
158	025140	042105	051440	042505	
159	025146	020113	051106	046517	
160	025154	023440	054503	040514	
161	025162	020047	047524	023440	
162	025170	054503	041114	000047	
163					
164	025176	051105	051117	047440	EM21: .ASCIZ /EROR ON DOING WRITE ON DISK/
165	025204	020116	047504	047111	
166	025212	020107	051127	052111	
167	025220	020105	047117	042040	
168	025226	051511	000113		
169					
170	025232	044523	020116	047117	EM22: .ASCIZ /SIN ON DOING WRITE/
171	025240	042040	044517	043516	
172	025246	053440	044522	042524	
173	025254	000			
174					
175	025255	110	020105	047117	EM23: .ASCIZ /HE ON DOING READ/
176	025262	042040	044517	043516	
177	025270	051040	040505	000104	
178					
179	025276	051503	020105	047117	EM24: .ASCIZ /CSE ON READ/
180	025304	051040	040505	000104	
181					
182	025312	040504	040524	042440	EM25: .ASCIZ /DATA EROR ON READ FROM DISK ADRES/

183	025320	047522	020122	047117
184	025326	051040	040505	020104
185	025334	051106	046517	042040
186	025342	051511	020113	042101
187	025350	042522	000123	
188				
189	025354	042510	047440	020116 EM26: .ASCIZ /HE ON WRT CHK/
190	025362	051127	020124	044103
191	025370	000113		
192				
193	025372	051127	020124	044103 EM27: .ASCIZ /WRT CHK EROR/
194	025400	020113	051105	051117
195	025406	000		
196				
197	025407	105	047522	000122 EM30: .ASCIZ /EROR/
198				
199				;DATA HEADERS
200				
201				
202	025414	020040	041520	020040 DH1: .ASCIZ / PC RKCS RKER RKDS RKDA/
203	025422	020040	051040	041513
204	025430	020123	020040	051040
205	025436	042513	020122	020040
206	025444	051040	042113	020123
207	025452	020040	051040	042113
208	025460	000101		
209				
210	025462	042523	045505	021440 DH6: .ASCIZ /SEEK # SEEKING ERRORS/
211	025470	020040	042523	045505
212	025476	047111	020107	042440
213	025504	051122	051117	000123
214				
215	025512	020040	041520	020040 DH7: .ASCIZ / PC RKCS RKER RKDS RKDA CYL/
216	025520	020040	051040	041513
217	025526	020123	020040	051040
218	025534	042513	020122	020040
219	025542	051040	042113	020123
220	025550	020040	051040	042113
221	025556	020101	020040	041440
222	025564	046131	000	
223				
224	025567	040	050040	020103 DH11: .ASCIIZ/ PC RKCS RKER RKDS RKDA: DR CYL SUR SEC/
225	025574	020040	051040	041513
226	025602	020123	020040	051040
227	025610	042513	020122	020040
228	025616	051040	042113	020123
229	025624	020040	045522	040504
230	025632	020072	051104	020040
231	025640	054503	020114	020040
232	025646	020040	052523	020122
233	025654	020040	020040	042523
234	025662	000103		
235	025664	042523	021503	020040 DH13: .ASCIZ /SEC# HDR RCVD/
236	025672	042110	020122	041522
237	025700	042126	000	
238				

J09

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 97
 DZRKLD.P11 31-AUG-76 15:35 FUNCTION SELECTION PROGRAM

SEQ 0113

5239	025703	040	050040	020103	DH14:	.ASCIZ	/	PC	CYLA	CYLB	RKER	RKDS	TRY#
5240	025710	020040	020040	054503									
5241	025716	040514	020040	020040									
5242	025724	054503	041114	020040									
5243	025732	020040	051040	042513									
5244	025740	020122	020040	045522									
5245	025746	051504	020040	020040									
5246	025754	052040	054522	000043									
5247													
5248	025762	020040	041520	020040	DH16:	.ASCIZ	/	PC	CYLA	CYLB	EXPCT	RECVD	TRY#
5249	025770	020040	041440	046131									
5250	025776	020101	020040	054503									
5251	026004	041114	020040	020040									
5252	026012	054105	041520	020124									
5253	026020	020040	042522	053103									
5254	026026	020104	020040	051124									
5255	026034	021531	000										
5256													
5257	026037	040	050040	020103	DH17:	.ASCIZ	/	PC	CYLB	EXPCT	RECVD		
5258	026044	020040	020040	054503									
5259	026052	041114	020040	042440									
5260	026060	050130	052103	020040									
5261	026066	051040	041505	042126									
5262	026074	000											
5263													
5264	026075	040	050040	020103	DH24:	.ASCIZ	/	PC	TRY#	RKCS	RKER	RKDS	RKDA: DR
5265	026102	020040	020040	051124									CYL
5266	026110	021531	020040	051040									SUR
5267	026116	041513	020123	020040									SEC
5268	026124	051040	042513	020122									
5269	026132	020040	051040	042113									
5270	026140	020123	051040	042113									
5271	026146	035101	042040	020122									
5272	026154	041440	046131	020040									
5273	026162	020040	051440	051125									
5274	026170	020040	020040	020040									
5275	026176	042523	000103										
5276													
5277	026202	047527	042122	020043	DH25:	.ASCIZ	/	WORD#	EXPCT	RECVD	CYL	SUR	SEC
5278	026210	042440	050130	052103									
5279	026216	020040	051040	041505									
5280	026224	042126	020040	041440									
5281	026232	046131	020040	052523									
5282	026240	020122	051440	041505									
5283	026246	000											
5284													
5285													
5286													
5287													
5288													
5289		026250											
5290													
5291	026250	001116	001162	001164	DT1:	.WORD		\$ERRPC, \$REG0, \$REG1, \$REG2, \$REG3, 0					
5292	026256	001166	001170	000000									
5293													
5294	026264	001116	001162	001164	DT7:	.WORD		\$ERRPC, \$REG0, \$REG1, \$REG2, \$REG3, \$REG4, 0					

;ERROR DATA POINTERS

.EVEN

K09

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 98
 DZRKLD.P11 31-AUG-76 15:35 FUNCTION SELECTION PROGRAM

SEQ 0114

```

5295 026272 001166 001170 001172
5296 026300 000000
5297
5298 026302 001116 001162 001164 DT11: .WORD $ERRPC,$REG0,$REG1,$REG2,$REG4,$REG5,$REG6,$REG7,0
5299 026310 001166 001172 001174
5300 026316 001176 001200 000000
5301
5302 026324 001116 001162 001164 DT17: .WORD $ERRPC,$REG0,$REG1,$REG2,0
5303 026332 001166 000000
5304
5305 026336 001116 001202 001162 DT24: .WORD $ERRPC,$REG10,$REG0,$REG1,$REG2,$REG4,$REG5,$REG6,$REG7,0
5306 026344 001164 001166 001172
5307 026352 001174 001176 001200
5308 026360 000000
5309

```

;DATA BUFFERS

```

5310
5311
5312 026362 IOBUF0:
5313 026362 000030 OUTBUF: .BLKW 24. ;IOBUF0 AND IOBUF1 ARE
5314 026442 000030 BUFR4: .BLKW 24. ;TWO - 768 WORDS LONG BUFFERS
5315 026522 000030 BUFR5: .BLKW 24. ;NORMALLY USED FOR DATA TRANSFERS
5316 026602 000030 BUFR6: .BLKW 24. ;TO AND FROM DISK
5317 026662 000030 BUFR7: .BLKW 24.
5318 026742 000200 BUFR10: .BLKW 128.
5319 027342 000200 BUFR11: .BLKW 128.
5320 027742 000200 BUFR12: .BLKW 128.
5321 030342 000200 BUFR13: .BLKW 128.
5322 030742 000210 BUFR14: .BLKW 136.
5323
5324 031362 001400 IOBUF1: .BLKW 768.
5325
5326
5327
5328 000001 .END

```


MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 100
 DZRKLD.F11 31-AUG-76 15:35 SYMBOL TABLE

SEQ 0115

ABRT	016716	BUFR13	030342	DRVDON	001223	EXT10	012656	NOSIN	010752
ADRES	001450	BUFR14	030742	DRVPT	001226	EXT4	006540	NXTBA	023436
BADTMO	002422	BUFR2	001352	DRV.RE=	104416	EXT5	007276	NXTDA	023462
BEGIN	010456	BUFR4	026442	DR.RST	016414	EXT6	010356	NXTDRV	003704
BEGSK	013174	BUFR5	026522	DSKADR	001432	EXT7	012114	NXTPAT	001426
BIT0	= 000001	BUFR6	026602	DSWR	= 177570	FCHECK	024052	NXTWC	023556
BIT00	= 000001	BUFR7	026662	DT1	026250	FDRIVE	002114	OUTADR	001262
BIT01	= 000002	BUSADR	001434	DT11	026302	FDRVE1	002116	OUTBUF	026362
BIT02	= 000004	CHKHDR	007204	DT17	026324	FFUNC	001216	PAT0	001414
BIT03	= 000010	CHKSWD	024000	DT24	026336	FINISH	011436	PAT1	001416
BIT04	= 000020	CHSW	023704	DT7	026264	FUNBEG	023106	PAT2	001420
BIT05	= 000040	CKSWR	= 104407	EMTVEC=	000030	FUNERR	023720	PAT3	001422
BIT06	= 000100	CLRERR	012106	EM1	024250	GCVL	016360	PBUFO	001404
BIT07	= 000200	CMPAGA	011170	EM10	024500	GETBUF	007770	PBUF1	001406
BIT08	= 000400	CMPGRP	014546	EM11	024531	GETINF	016140	PGSUBR	001430
BIT09	= 001000	CN.RDY	016544	EM12	024547	GOBAK	011574	PIRQ	= 177772
BIT1	= 000002	CN.RST	016526	EM13	024571	GOTYPE	013524	PIRQVE=	000240
BIT10	= 002000	COMPAR	011144	EM14	024620	GTSWR	= 104406	PLOT	014146
BIT11	= 004000	CON.RD=	104421	EM16	024735	GT4RG	016106	PLTGRP	014032
BIT12	= 010000	CON.RE=	104415	EM2	024307	GT5RG	016062	PLTPT	014662
BIT13	= 020000	CR	= 000015	EM20	025110	HT	= 000011	PLT1	014362
BIT14	= 040000	CRLF	= 000200	EM21	025176	INADR	001260	PRSPAT	001424
BIT15	= 100000	CSEORR	011600	EM22	025232	INDX1	001474	PRO	= 000000
BIT2	= 000004	DDISP	= 177570	EM23	025255	INDX2	001476	PR1	= 000040
BIT3	= 000010	DH1	025414	EM24	025276	INDX3	001500	PR2	= 000100
BIT4	= 000020	DH11	025567	EM25	025312	INDX4	001502	PR3	= 000140
BIT5	= 000040	DH13	025664	EM26	025354	INPT	023746	PR4	= 000200
BIT6	= 000100	DH14	025703	EM27	025372	IOBUF0	026362	PR5	= 000240
BIT7	= 000200	DH16	025762	EM3	024323	IOBUF1	031362	PR6	= 000300
BIT8	= 000400	DH17	026037	EM30	025407	IOTVEC=	000020	PR7	= 000340
BIT9	= 001000	DH24	026075	EM4	024337	LF	= 000012	PS	= 177776
BLNKS1	002111	DH25	026202	EM5	024355	LUPHE	010564	PSW	= 177776
BLNKS2	002110	DH6	025462	EM6	024424	LUPSIN	010556	PTGEN0	010032
BLNKS3	002107	DH7	025512	EM7	024461	LUPSW	001222	PTGEN1	010114
BLNKS4	002106	DISPLA	001142	ERCNT1	001504	LUPWCE	011274	PTGEN2	010216
BLNKS5	002105	DISPRE	000174	ERCNT2	001506	MESSAGE=	104417	PTGEN3	010260
BLNKS6	002104	DMPREG	017576	ERCNT3	001510	MISCMP	011742	PTRNO1	010110
BLNKS7	002103	DONE	011562	ERCNT4	001512	MSG	015770	PTRNO2	010112
BLNKS8	002102	DONTRK	011522	ERCNT5	001514	MSG1	001602	PT1	001560
BLNKS9	002101	DOSUR1	011554	ERCNT6	001516	MSG10	001742	PT10	001576
BLNK10	002100	DOWCHK	012344	ERCNT7	001520	MSG11	001771	PT11	001600
BLNK13	002075	DOWRT	012150	ERCNT8	001522	MSG12	002027	PT2	001562
BPTVEC=	000014	DRHOLD	002120	ERRTYP	017362	MSG13	002053	PT3	001564
BRKDA	016144	DRIVAD	001230	ERRVEC=	000004	MSG14	002064	PT4	001566
BTEOP	015156	DRIVS	001224	ERR1	016322	MSG2	001610	PT5	001570
BUFLGO	001410	DRIVO	001232	ERR2	016244	MSG3	001616	PT6	001572
BUFLG1	001412	DRIV1	001234	ERWCHK	011330	MSG4	001640	PT7	001574
BUFNO	001446	DRIV2	001236	ESR13	015402	MSG5	001657	PWRFL	004170
BUFR	001266	DRIV3	001240	ESR15	015310	MSG6	001716	PWRVEC=	000024
BUFR1	001320	DRIV4	001242	ESR20	015456	MSG7	001733	RDAGAI	010566
BUFR10	026742	DRIV5	001244	ESR25	015544	MS15	024673	RDCHR	= 104410
BUFR11	027342	DRIV6	001246	EXEC	023612	MS17	025042	RDLIN	= 104411
BUFR12	027742	DRIV7	001250	EXEC1	023610	NOHE	010704	RDOCT	= 104412

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 101
 DZRKLD.P11 31-AUG-76 15:35 SYMBOL TABLE

SEQ 0116

READ 010532	ST2 003436	TYPDES 022222	SEOP 015172	\$REG4 001172
REPEAT 012612	ST3 003666	TYPDS = 104405	SEOPCT 015214	\$REG5 001174
REPMSC 011220	SWR 001140	TYPDSS= 104424	\$ERFLG 001103	\$REG6 001176
REPTIM 013124	SWREG 000176	TYPE = 104401	\$ERMAX 001115	\$REG7 001200
RESDON= 104423	SW0 = 000001	TYPMSG= 104420	\$ERROR 017106	\$RESRE 022364
RESREG= 104414	SW00 = 000001	TYPOC = 104402	\$ERRPC 001116	\$RTNAD 015266
RESVEC= 000010	SW01 = 000002	TYPON = 104404	\$ERRTB 002122	\$SAVRE 022326
RES.DO 016440	SW02 = 000004	TYPOS = 104403	\$ERTTL 001112	\$SAVR6 023074
RETRY1 001252	SW03 = 000010	TYPTIM 013662	\$ESCAP 001204	\$SCOPE 016732
RETRY2 001254	SW04 = 000020	TY.MSG 016032	\$FILLC 001156	\$SETUP= 000117
RETRY3 001256	SW05 = 000040	WBUSAD 001442	\$FILLS 001155	\$STUP = 177777
RKBA 001462	SW06 = 000100	WCAGAI 011056	\$GDADR 001120	\$SUPRS 022266
RKCS 001456	SW07 = 000200	WCEROR 012006	\$GDDAT 001124	\$SVLAD 017044
RKDA 001464	SW08 = 000400	WCERR 012366	\$GET42 015244	\$SVPC = 000220
RKDB 001466	SW09 = 001000	WCHI 012372	\$GTSWR 021070	\$SWR = 163000
RKDS 001452	SW1 = 000002	WCHI1 012364	\$HD = 000000	\$SWRMK= 000000
RKER 001454	SW10 = 002000	WCLO 012572	\$HIOCT 021772	\$TKB 001146
RKPRI 001470	SW11 = 004000	WCREPT 011260	\$ICNT 001104	\$TKCNT 020526
RKVEC 001472	SW12 = 010000	WDSKAD 001440	\$ILLUP 023070	\$TKINT 020536
RKWC 001460	SW13 = 020000	WRDCNT 001436	\$INTAG 001135	\$TKQEN= 020535
R6 =%000006	SW14 = 040000	WRERR 012164	\$ITEMB 001114	\$TKQIN 020530
R7 =%000007	SW15 = 100000	WRHI 012324	\$LF 001214	\$TKQOU 020532
SAVREG= 104413	SW2 = 000004	WRLO 012166	\$LPADR 001106	\$TKQSR 020534
SBR1 006452	SW3 = 000010	WRLO1 012162	\$LPERR 001110	\$TKS 001144
SBR2 006476	SW4 = 000020	WRTCHK 011042	\$MNEW 021661	\$TKSRV 020606
SBR3 007150	SW5 = 000040	WWRDCN 001444	\$MSWR 021650	\$TN = 000013
SECPTR 010352	SW6 = 000100	XHE 011366	\$MUL = 000003	\$TNPWR 022532
SFTCMP 011112	SW7 = 000200	XXDPMO 001220	\$MULT 020414	\$TPB 001152
SIAD 001542	SW8 = 000400	\$AUTOB 001134	\$NULL 001154	\$TPFLG 001157
SIZEF 024164	SW9 = 001000	\$BDADR 001122	\$NWTST= 000001	\$TPS 001150
SKDON 015076	TBITVE= 000014	\$BDDAT 001126	\$OCNT 022216	\$TRAP 022616
SMGRP 014416	TIMDON 013774	\$BELL 001206	\$OMODE 022220	\$TRAP2 022640
SOAD 001524	TIMER 001264	\$CHARC 020410	\$OVER 017072	\$TRP = 000025
SORT 013266	TIMSEK 014722	\$CKSWR 021000	\$PASS 001100	\$TRPAD 022652
SP1 012626	TKVEC = 000060	\$CMTAG 001100	\$POWER 023076	\$TSTNM 001102
SP10 012650	TPVEC = 000064	\$CM1 = 000011	\$PWRAD 023064	\$TTYIN 021626
SP11 012652	TRAPVE= 000034	\$CM2 = 000022	\$PWRDN 022724	\$TYPDS 017750
SP12 012654	TRTVEC= 000014	\$CM3 = 000011	\$PWRMG 023060	\$TYPE 020174
SP2 012630	TSTRWS 016642	\$CNTLG 021643	\$PWRUP 022776	\$TYPEC 020344
SP3 012632	TST.RW= 104422	\$CNTLU 021636	\$QUES 001212	\$TYPEX 020412
SP4 012634	TST1 004204	\$CRLF 001213	\$RDCHR 021302	\$TYPJC 022020
SP5 012636	TST10 012120	\$DBLK 020164	\$RDLIN 021372	\$TYPON 022034
SP6 012640	TST11 012662	\$DB2D 022422	\$RDOCT 021672	\$TYPDS 021774
SP7 012642	TST12 015150	\$DECVL 022602	\$RDSZ = 000010	\$XTSTR 016754
SP8 012644	TST2 004550	\$DOAGN 015264	\$REGAD 001160	\$GET4= 000000
SP9 012646	TST3 005046	\$DTBL 020154	\$REG0 001162	\$OFILL 022217
STACK = 001100	TST4 005536	\$ENDAD 015254	\$REG1 001164	. = 034362
START 002462	TST5 006544	\$ENDCT 015222	\$REG10 001202	
START1 003020	TST6 007302	\$ENDMG 015273	\$REG2 001166	
STKLMT= 177774	TST7 010362	\$ENULL 015270	\$REG3 001170	

. ABS. 034362 000

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

MAINDEC-11-DZRKL-D MACY11 27(1006) 04-OCT-76 14:26 PAGE 102
DZRKLD.P11 31-AUG-76 15:35 SYMBOL TABLE

N09

SEQ 0117

DZRKLD,DZRKLD/LI:ME/NL:MC:MD:OND/SOL/NSQ-DZRKLD.P11
RUN-TIME: 62 43 1 SECONDS
RUN-TIME RATIO: 237/108=2.1
CORE USED: 27K (53 PAGES)

