

RK11/RK05-F-J

BASIC LOGIC TEST 2
MD-11-DZRKK-C

EP-DZRKK-C-DL-A

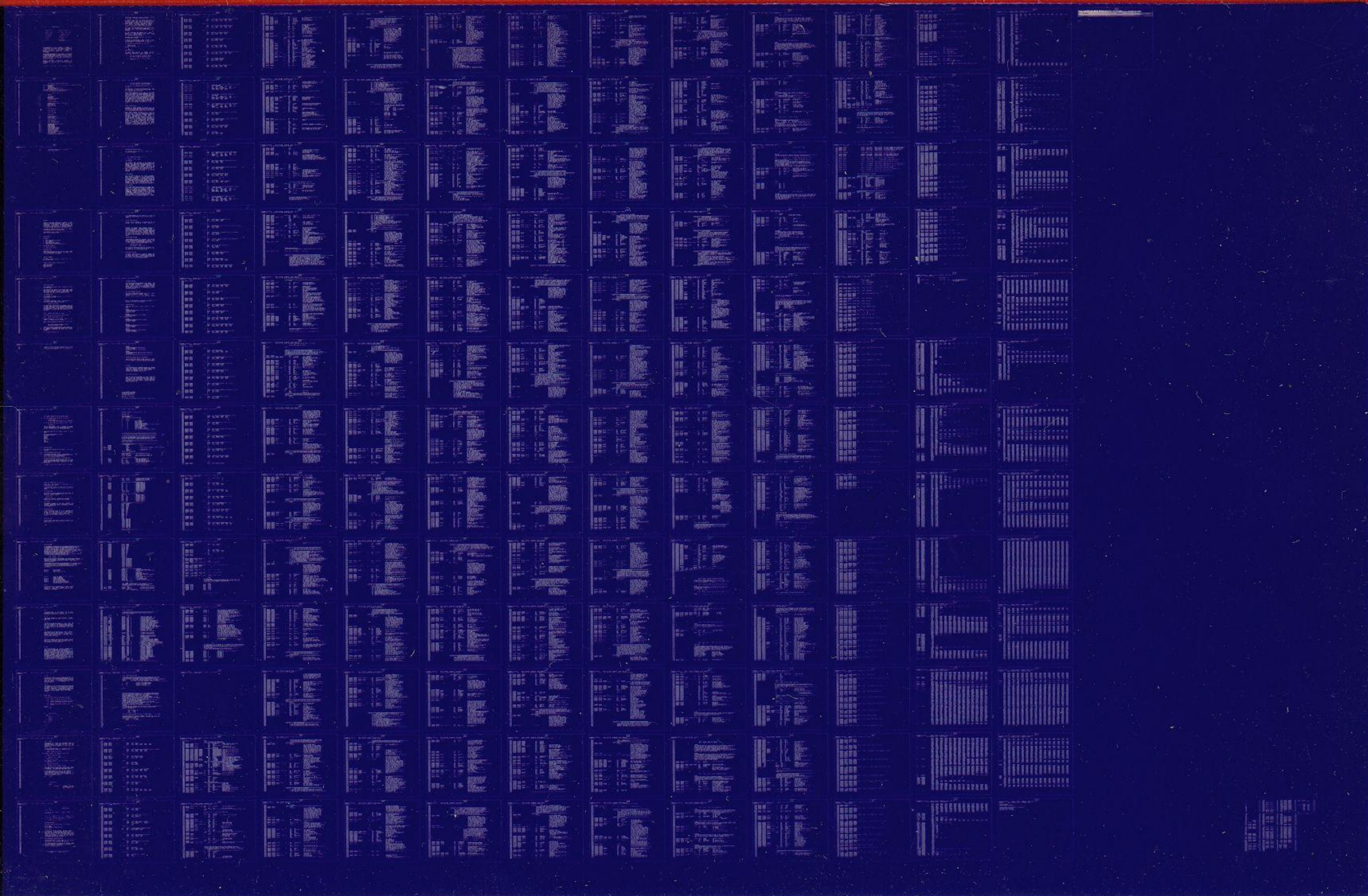
OCT 1976

COPYRIGHT ©1976

digital

FICHE 1 OF 1

Made in U.S.A.



B01

MAINDEC-11-DZRKK-C
RK11 BASIC LOGIC TEST-II

MACY11 27(732) 16-SEP-76 16:00 PAGE 2

.PEN %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZRKK-C-D
PRODUCT NAME:	RK11 BASIC LOGIC TEST-II
DATE CREATED:	FEBRUARY 23, 1976
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	JIM KAPADIA
REVISED BY:	PERVEZ ZAKI TOM SAWYER MARCH 1976

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

QUICK LOOK-UP OPERATING INSTRUCTIONS

FOR A QUICK REFERENCE, LOOK UP THE FOLLOWING SECTIONS:

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

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

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	EQUIPMENT
2.2	PRELIMINARY PROGRAMS
2.3	EXECUTION TIME
3.0	STARTING ADDRESS
4.0	PROGRAM CONTROL MODES & 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	DRIVE-LESS TEST
7.0	SWITCH OPTIONS
8.0	SCOPE LOOPS
9.0	PROGRAM STRUCTURE
9.1	SET-UP PHASE
9.2	DRIVE DEPENDENT CONTROLLER TESTS
10.0	ERROR REPORTING
11.0	ERROR INTERPRETATION
12.0	HANDLERS AND COMMON ROUTINES
12.1	TRAP HANDLER
12.2	SCOPE HANDLER
12.3	ERROR HANDLER
12.4	CONTROL RESET ROUTINE
12.5	CONTROL READY ROUTINE
12.6	DRIVE RESET ROUTINE
12.7	TIME DELAY ROUTINE
12.8	WAIT FOR INTERRUPT ROUTINE
12.9	OTHER ROUTINES
	TTY HANDLER (I/O), ERROR TYPEOUT ROUTINE
	POWER DOWN/POWER UP ROUTINE

MAINDEC-11-DZKCC-C
DZKCC.P11

MACY11 27(732) 16-SEP-76 16:00 PAGE 4

DO1

110
111

13.0
14.0

UNEXPECTED TIMEOUTS & RK11 INTERRUPTS
QUICK VERIFYING MODE

1.0 ABSTRACT

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

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

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

2.0 REQUIREMENTS

2.1 EQUIPMENT

- A. PDP11 WITH CONSOLE TELETYPE.
- B. 8K OF MEMORY
- C. RK11 CONTROLLER (C OR D)
- D. RK05 DRIVE/S OR SIMULATOR/S
- E. DISK PLATTER - NEED NOT BE FORMATTED.

2.2 PRELIMINARY PROGRAMS

RK11 BASIC LOGIC TESTS-I

2.3 EXECUTION TIME

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

3.0 STARTING ADDRESS

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

4.0 PROGRAM CONTROL MODES & OPERATOR ACTION

PAPER TAPE LOADING
RKDP DUMP MODE
RKDP CHAIN MODE
ACT11

4.1 PAPER TAPE LOADING

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

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

4.1.3 LOAD ADDRESS 200

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

PRESS START.

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

DRIVES TO BE TESTED?

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

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

THE DRIVES DO NOT HAVE TO BE IN LOGICAL ORDER.

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

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

THUS A NORMAL SEQUENCE WITH DRIVES 0,1 RK11D WOULD BE

RK11 LOGIC TESTS-II MAINDEC-11-DZRKK-C
DRIVES TO BE TESTED? 0,1<CR>

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

"IU" DELETES THE ENTIRE LINE

GO1

MAINDEC-11-DZRKK-C
DZRKKC.P11

MACY11 27(732) 16-SEP-76 16:00 PAGE 7

224
225
226

4.1.7 IF REPLY TO ANY OF THE ABOVE QUESTION IS IN A WRONG
FORMAT (EX: 012<CR>:0.8<CR>: 0.A<CR>: M<CR> ETC) IT
IS AUTOMATICALLY REJECTED, A "???" IS PRINTED OUT;

224
225
226

THE CORRECT ANSWER CAN NOW BE RETYPED AGAIN.

4.1.8 THE DRIVE NUMBER BEING TESTED OUT IS PRINTED:

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

IF THE DRIVE IS AN RK-OSF, AN F IS APPENDED

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

END PASS # X

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

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

RK11 LOGIC TESTS MAINDEC-11-DZRKK-C
DRIVES TO BE TSTED?

0,1<CR>

DRIVE 0

DRIVE 1

END PASS # 1

0

DRIVE 1

END PASS # 2

...

...

4.2 RKDP DUMP MODE

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

4.2.2 START AS NORMALLY USING SA 200

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

REPLACE DR0 RKDP-PAK BY OTHER, TYP CR WHEN DONE

IF DRIVE 0 HAS TO BE TESTED THE RKDP PACK ON THAT DRIVE SHOULD BE REPLACED BY ANOTHER PACK. THE DRIVE SHOULD BE PUT ON 'WRT ENABL' (BECAUSE RKDP WRITE PROTECTS DRIVE 0)

IF DRIVE 0 IS NOT TO BE CHECKED THEN THE MESSAGE

227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282

283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332

SHOULD BE IGNORED AND A CR TYPED IN.

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

IT IS IMPORTANT TO NOTE THAT RKDP PACK ON DRIVE 0 SHOULD BE REPLACED OR THAT DRIVE SHOULD NOT BE TESTED.

4.3 RKDP CHAIN MODE

THE PROGRAM IS CHAIN-LOADED FROM THE RKDP PACK ON DRIVE 0. AFTER THE PROGRAM IDENTIFIES ITSELF THE FOLLOWING PRINTOUT OCCURS.

DRO NOT TSTD

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

4.4 ACT11 MODE

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

5.0 DRIVE SELECTION

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

6.0 DRIVE-LESS TEST

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

7.0 SWITCH OPT ONS

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E. AN 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE 'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' WHENEVER THE PROGRAM ENTERS THE SCOPE ROUTINE OR BEGINS A NEW TEST. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

'SWR = NNNNNN NEW ='

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

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

SW<15>=1 HALT ON ERROR
SW<14>=1 LOOP ON TEST
SW<13>=1 INHIBIT ERROR PRINTOUTS

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

7.1 SW<15>

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

7.2 SW<14>

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

7.3 SW <13>

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

7.4 SW <12>

THIS SWITCH ALLOWS THE 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 8.0 FOR DIFFERENT SCOPE LOOPS AVAILABLE.

7.5 SW <11>

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

7.6 SW <10>

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

7.7 SW <09>

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

7.8 SW <08>

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

7.9 SW<06>

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

8.0 SCOPE LOOPS

THERE ARE THREE KINDS OF SCOPE LOOPS AVAILABLE

1. SW14: LOOPING IS DONE FOR THE ENTIRE SUB-TEST
2. SW12: LOOPING IS DONE FROM THE POINT OF ERROR BACK TO THE PREVIOUS 'SCOPE' STATEMENT.
3. SW09: PROVIDE THE TIGHTEST POSSIBLE SCOPE LOOP SEE SEC. 7.7

EXAMPLE:

TST1: SCOPE

INITIALIZATION

ERROR 1
ERROR 2
ERROR 3
ERROR 4
:

TST2: SCOPE

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

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

1. ERROR 2 OCCURS, SW 14 SET.

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

2. ERROR 2 OCCURS, SW 12 SET.

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

3. ERROR 2,3; SW 14 SET.

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

4. ERROR 2,3; SW 12 SET.

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

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

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

TST1: SCOPE

:

INITIALIZATION

:

ERROR 1

:

MOV #15,\$LPERR

;'SLPERR' CONTAINS
;THE ADDRESS TO LOOP
;BACK ON ERROR- SW 9

487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537

538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588

```

1$:      :      -----
          :      I
          :      N REPETITIONS
          :      I
TST2:    SCOPE  -----

```

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

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

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

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

9.0 PROGRAM STRUCTURE

THERE ARE THREE DISTINCT PARTS OF THE PROGRAM.

SET-UP PHASE DRIVE-DEPENDENT CONTROLLER TESTS

9.1 SET-UP PHASE

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

9.2 DRIVE DEPENDENT CONTROLLER TESTS

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

JUST BEFORE ENTERING THIS SECTION THE PROGRAM FINDS OUT WHICH DRIVE IS TO BE CHECKED. IF IN RKDP CHAIN MODE, DRIVE 0 IF PRESENT, IS SKIPPED AND THE NEXT AVAILABLE DRIVE IS SELECTED.

THE DRIVE NUMBER BEING TESTED IS PRINTED OUT:
DRIVE N ;N=0,1,2...7

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

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

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

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

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

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

THUS ONE PASS OF THE PROGRAM INVOLVES DOING

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

10.0 ERROR REPORTING

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

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

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

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

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

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

11.0 ERROR INTERPRETATION

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

EVERY ERROR MESSAGE CONTAINS A PC. THIS PC INDICATES THE POSITION IN PROGRAM WHERE THE ERROR CALL IS LOCATED. THE ERROR MESSAGE, BECAUSE OF PRACTICAL CONSIDERATIONS IS MADE SHORT AND MEANINGFUL. THE USER IS ADVTD 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 IN PINPOINTING THE FAULT. A BRIEF EXPLANATION OF WHAT IS BEING CHECKED IN THE SUBTEST IS GIVEN AT THE BEGINNING OF EVERY SUBTEST. ALL THE NUMBERS GIVEN WITH ERROR MESSAGES ARE IN OCTAL.

12.0 HANDLERS AND COMMON ROUTINES

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

A. AS A SUBROUTINE THROUGH 'JSR' CALL

B. THROUGH A 'TRAP' HANDLER

12.1 TRAP HANDLER

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

12.2 SCOPE HANDLER

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

12.3 ERROR HANDLER

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

687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
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

IF AN ERROR MESSAGE IS TO BE TYPED OUT AN EXIT IS
MADE TO THE ERROR MESSAGE TYPEOUT ROUTINE LOCATED AT
'SERRTYP'.

12.4 CONTROL RESET ROUTINE

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

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

12.5 CONTROL READY ROUTINE

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

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

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

12.6 DRIVE RESET ROUTINE

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

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
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783

12.7 TIME DELAY ROUTINE

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

12.8 WAIT FOR INTERRUPT ROUTINE

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

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

12.9 OTHER ROUTINES

THERE ARE OTHER COMMONLY USED ROUTINES AS LISTED BELOW.

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

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

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

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

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

784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839

840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895

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

SPWRUP:
ROUTINE FOR HANDLING POWER UP AFTER A POWER FAIL.
LOCATED AT SPWRUP
CALLED WHEN POWER RETURNS AFTER HAVING GONE DOWN.

13.0 UNEXPECTED TIMEOUTS AND RK11 INTERRUPTS

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

CONTINUE.

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

14.0 QUICK VERIFYIN' MODE

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

.TITLE MAINDEC-11-DZRKK-C
; *COPYRIGHT (C) 1974, 1976
; *DIGITAL EQUIPMENT CORP.
; *MAYNARD, MASS. 01754
; *
; *PROGRAM BY JIM KAPADIA
; *
; *THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC

H02

MAINDEC-11-DZRKK-C
DZRKKC.P11

MACY11 27(732) 16-SEP-76 16:00 PAGE 21

896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951

000000
000046
000052
000000
000000
001100
000011
000012
000015
000200
177776
177774

```

;PACKAGE (MAINDEC-11-DZOAC-CO),MAR 21, 1976.
;
;JANUARY 1975
;
;REVISED MARCH,1976
;BY TOM SAWYER
.SBTTL OPERATIONAL SWITCH SETTINGS
;
;      SWITCH      USE
;      -----
;      15          HALT ON ERROR
;      14          LOOP ON TEST
;      13          INHIBIT ERROR TYPEOUTS
;      12          CYCLE ON ERROR TO PREVIOUS 'SCOPE' STATEMENT
;      11          INHIBIT ITERATIONS
;      10          TESTING ON SIMULATOR
;      9           LOOP ON ERROR
;      8           LOOP ON TEST IN SWR<7:0>
;      6           DROP THE DRIVE IF MORE THAN 5 ERRORS
;
;*****
;YOU ARE ADVISED TO READ THE DOCUMENT BEFORE USING THIS PROGRAM.
;ON GETTING AN ERROR REFER TO THE LISTINGS AT THE PC POINTED
;OUT IN THE ERROR MESSAGE. ADJACENT ERROR MESSAGES IF FOLLOWED
;CAREFULLY COULD LEAD TO AN EASY PINPOINTING OF THE FAULT
;*****
.SBTTL ACT11 HOOKS
;*****
;HOOKS REQUIRED BY ACT11
;      $SVPC=.      ;SAVE PC
;      .=46         ;;1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
;      $ENDAD       ;;2)SET LOC.52 TO ZERO
;      .=52         ;; RESTORE PC
;      .WORD 0
;      .=$SVPC
.SBTTL BASIC DEFINITIONS
;
;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK= 1100
.EQUIV EMT,ERROR    ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE    ;;BASIC DEFINITION OF SCOPE CALL
;
;*MISCELLANEOUS DEFINITIONS
HT= 11              ;;CODE FOR HORIZONTAL TAB
LF= 12              ;;CODE FOR LINE FEED
CR= 15              ;;CODE FOR CARRIAGE RETURN
CRLF= 200           ;;CODE FOR CARRIAGE RETURN-LINE FEED
PS= 177776          ;;PROCESSOR STATUS WORD
.EQUIV PS,PSW
STKLMT= 177774      ;;STACK LIMIT REGISTER

```

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 22
 DZRKKC.P11 BASIC DEFINITIONS

```

952      177772      PIRQ= 177772      ;;PROGRAM INTERRUPT REQUEST REGISTER
953      177570      DSWR= 177570      ;;HARDWARE SWITCH REGISTER
954      177570      DDISP= 177570      ;;HARDWARE DISPLAY REGISTER
955
956      ;*GENERAL PURPOSE REGISTER DEFINITIONS
957      000000      R0= %0      ;;GENERAL REGISTER
958      000001      R1= %1      ;;GENERAL REGISTER
959      000002      R2= %2      ;;GENERAL REGISTER
960      000003      R3= %3      ;;GENERAL REGISTER
961      000004      R4= %4      ;;GENERAL REGISTER
962      000005      R5= %5      ;;GENERAL REGISTER
963      000006      R6= %6      ;;GENERAL REGISTER
964      000007      R7= %7      ;;GENERAL REGISTER
965      .EQUIV R6,SP      ;;STACK POINTER
966      .EQUIV R7,PC      ;;PROGRAM COUNTER
967
968      ;*PRIORITY LEVEL DEFINITIONS
969      000000      PR0= 0      ;;PRIORITY LEVEL 0
970      000040      PR1= 40      ;;PRIORITY LEVEL 1
971      000100      PR2= 100      ;;PRIORITY LEVEL 2
972      000140      PR3= 140      ;;PRIORITY LEVEL 3
973      000200      PR4= 200      ;;PRIORITY LEVEL 4
974      000240      PR5= 240      ;;PRIORITY LEVEL 5
975      000300      PR6= 300      ;;PRIORITY LEVEL 6
976      000340      PR7= 340      ;;PRIORITY LEVEL 7
977
978      ;*"SWITCH REGISTER" SWITCH DEFINITIONS
979      100000      SW15= 100000
980      040000      SW14= 40000
981      020000      SW13= 20000
982      010000      SW12= 10000
983      004000      SW11= 4000
984      002000      SW10= 2000
985      001000      SW09= 1000
986      000400      SW08= 400
987      000200      SW07= 200
988      000100      SW06= 100
989      000040      SW05= 40
990      000020      SW04= 20
991      000010      SW03= 10
992      000004      SW02= 4
993      000002      SW01= 2
994      000001      SW00= 1
995      .EQUIV SW09,SW9
996      .EQUIV SW08,SW8
997      .EQUIV SW07,SW7
998      .EQUIV SW06,SW6
999      .EQUIV SW05,SW5
1000      .EQUIV SW04,SW4
1001      .EQUIV SW03,SW3
1002      .EQUIV SW02,SW2
1003      .EQUIV SW01,SW1
1004      .EQUIV SW00,SW0
1005
1006      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
1007      100000      BIT15= 100000

```

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 23
 DZRKKC.P11 BASIC DEFINITIONS

1008	040000	BIT14=	40000
1009	020000	BIT13=	20000
1010	010000	BIT12=	10000
1011	004000	BIT11=	4000
1012	002000	BIT10=	2000
1013	001000	BIT09=	1000
1014	000400	BIT08=	400
1015	000200	BIT07=	200
1016	000100	BIT06=	100
1017	000040	BIT05=	40
1018	000020	BIT04=	20
1019	000010	BIT03=	10
1020	000004	BIT02=	4
1021	000002	BIT01=	2
1022	000001	BIT00=	1
1023		.EQUIV	BIT09,BIT9
1024		.EQUIV	BIT08,BIT8
1025		.EQUIV	BIT07,BIT7
1026		.EQUIV	BIT06,BIT6
1027		.EQUIV	BIT05,BIT5
1028		.EQUIV	BIT04,BIT4
1029		.EQUIV	BIT03,BIT3
1030		.EQUIV	BIT02,BIT2
1031		.EQUIV	BIT01,BIT1
1032		.EQUIV	BIT00,BIT0
1033			
1034		;*BASIC "CPU" TRAP VECTOR ADDRESSES	
1035	000004	ERRVEC= 4	::TIME OUT AND OTHER ERRORS
1036	000010	RESVEC= 10	::RESERVED AND ILLEGAL INSTRUCTIONS
1037	000014	TBITVEC=14	::"T" BIT
1038	000014	TRTVEC= 14	::TRACE TRAP
1039	000014	BPTVEC= 14	::BREAKPOINT TRAP (BPT)
1040	000020	IOTVEC= 20	::INPUT/OUTPUT TRAP (IOT) **SCOPE**
1041	000024	PWRVEC= 24	::POWER FAIL
1042	000030	EMTVEC= 30	::EMULATOR TRAP (EMT) **ERROR**
1043	000034	TRAPVEC=34	::"TRAP" TRAP
1044	000060	TKVEC= 60	::TTY KEYBOARD VECTOR
1045	000064	TPVEC= 64	::TTY PRINTER VECTOR
1046	000240	PIRQVEC=240	::PROGRAM INTERRUPT REQUEST VECTOR
1047		.SBTTL	TRAP CATCHER
1048			
1049	000000	.=0	
1050		;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"	
1051		;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS	
1052		;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS	
1053	000174	.=174	
1054	000174	DISPREG: .WORD 0	::SOFTWARE DISPLAY REGISTER
1055	000176	SWREG: .WORD 0	::SOFTWARE SWITCH REGISTER
1056		.SBTTL	STARTING ADDRESS(ES)
1057	000200	JMP	2*START ;;JUMP TO STARTING ADDRESS OF PROGRAM

K02

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 24
 DZRKKC.P11 COMMON TAGS

```

1058      .SBTTL  COMMON TAGS
1059
1060      ;*****
1061      ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
1062      ;*USED IN THE PROGRAM.
1063
1064      001100      .=1100
1065      001100      $CNTAG:      .WORD      0      ; START OF COMMON TAGS
1066      001100      $PASS:      .WORD      0      ; CONTAINS PASS COUNT
1067      001102      $STNM:      .BYTE      0      ; CONTAINS THE TEST NUMBER
1068      001103      $ERFLG:      .BYTE      0      ; CONTAINS ERROR FLAG
1069      001104      $ICNT:      .WORD      0      ; CONTAINS SUBTEST ITERATION COUNT
1070      001106      $LPADR:      .WORD      0      ; CONTAINS SCOPE LOOP ADDRESS
1071      001110      $LPERR:      .WORD      0      ; CONTAINS SCOPE RETURN FOR ERRORS
1072      001112      $ERTTL:      .WORD      0      ; CONTAINS TOTAL ERRORS DETECTED
1073      001114      $ITEMB:      .BYTE      0      ; CONTAINS ITEM CONTROL BYTE
1074      001115      $EX:      .BYTE      1      ; CONTAINS MAX. ERRORS PER TEST
1075      001116      $ERRPC:      .WORD      0      ; CONTAINS PC OF LAST ERROR INSTRUCTION
1076      001120      $GDADR:      .WORD      0      ; CONTAINS ADDRESS OF 'GOOD' DATA
1077      001122      $BDADR:      .WORD      0      ; CONTAINS ADDRESS OF 'BAD' DATA
1078      001124      $GDDAT:      .WORD      0      ; CONTAINS 'GOOD' DATA
1079      001126      $BDDAT:      .WORD      0      ; CONTAINS 'BAD' DATA
1080      001130      .WORD      0      ; RESERVED--NOT TO BE USED
1081      001132      .WORD      0
1082      001134      $AUTOB:      .BYTE      0      ; AUTOMATIC MODE INDICATOR
1083      001135      $INTAG:      .BYTE      0      ; INTERRUPT MODE INDICATOR
1084      001136      .WORD      0
1085      001140      $SWR:      .WORD      DSWR      ; ADDRESS OF SWITCH REGISTER
1086      001142      $DISPLAY:      .WORD      DDISP      ; ADDRESS OF DISPLAY REGISTER
1087      001144      $TKS:      177560      ; TTY KBD STATUS
1088      001146      $TKB:      177562      ; TTY KBD BUFFER
1089      001150      $TPS:      177564      ; TTY PRINTER STATUS REG. ADDRESS
1090      001152      $TPB:      177566      ; TTY PRINTER BUFFER REG. ADDRESS
1091      001154      $NULL:      .BYTE      0      ; CONTAINS NULL CHARACTER FOR FILLS
1092      001155      $FILLS:      .BYTE      2      ; CONTAINS # OF FILLER CHARACTERS REQUIRED
1093      001156      $FILLC:      .BYTE      12      ; INSERT FILL CHARS. AFTER A "LINE FEED"
1094      001157      $TPFLG:      .BYTE      0      ; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
1095      001160      $REGAD:      .WORD      0      ; CONTAINS THE ADDRESS FROM
1096      ; WHICH ($REGO) WAS OBTAINED
1097      001162      $REGO:      .WORD      0      ; CONTAINS (($REGAD)+0)
1098      001164      $REG1:      .WORD      0      ; CONTAINS (($REGAD)+2)
1099      001166      $REG2:      .WORD      0      ; CONTAINS (($REGAD)+4)
1100      001170      $REG3:      .WORD      0      ; CONTAINS (($REGAD)+6)
1101      001172      $REG4:      .WORD      0      ; CONTAINS (($REGAD)+10)
1102      001174      $REG5:      .WORD      0      ; CONTAINS (($REGAD)+12)
1103      001176      $REG6:      .WORD      0      ; CONTAINS (($REGAD)+14)
1104      001200      $REG7:      .WORD      0      ; CONTAINS (($REGAD)+16)
1105      001202      $REG10:      .WORD      0      ; CONTAINS (($REGAD)+20)
1106      001204      $REG11:      .WORD      0      ; CONTAINS (($REGAD)+22)
1107      001206      $TIMES:      0      ; MAX. NUMBER OF ITERATIONS
1108      001210      $ESCAPE:      0      ; ESCAPE ON ERROR ADDRESS
1109      001212      $QUES:      .ASCII      '?'      ; QUESTION MARK
1110      001213      $CRLF:      .ASCII      <15>      ; CARRIAGE RETURN
1111      001214      $LF:      .ASCII      <12>      ; LINE FEED
1112      ;*****

```

.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

\$ERRTB:

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

```

      TIME OUT ON RK11 REG
      PC      REG
      ***** 177400
  
```

NOTE THAT ***** WOULD BE THE ACTUAL PC WHERE 'ERROR 1' IS LOCATED.

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

;ERROR ITEMS TABLE

Line	Address	Value	Item	Comment
1169			1	
1170				
1171	001216	024764	EM12	;'SIN' IS SET
1172	001220	031711	DH44	;PC RKCS RKER RKDS RKDA
1173	001222	031324	DT20	;\$ERRPC \$REG0 \$REG1 \$REG2 \$REG3
1174	001224	000000	0	
1175				
1176			2	
1177				
1178	001226	026635	EM70	;CNTRL RDY DIDN'T SET ON READ/FMT FROM DISK ADDRESS
1179	001230	031475	DH14	;PC RKCS RKER RKWC
1180	001232	031344	DT26	;\$ERRPC \$REG0 \$REG1 \$REG2
1181	001234	000000	0	
1182				
1183			3	
1184				
1185	001236	025011	EM16	;RKDA WRONG AFTER SSE
1186	001240	031431	DH4	;PC EXPCT RECVD
1187	001242	031314	DT2	;\$ERRPC \$REG0 \$REG1
1188	001244	000000	0	
1189				
1190			4	
1191				
1192	001246	025040	EM21	;RKDS ERROR
1193	001250	031575	DH34	;PC RKDS
1194	001252	031306	DT1	;\$ERRPC \$REG0
1195	001254	000000	0	
1196				
1197			5	
1198				
1199	001256	025052	EM30	;'DPL' BIT SET, CHECK DRIVE POWER
1200	001260	031537	DH30	;PC RKCS RKER RKDS
1201	001262	031344	DT26	;\$ERRPC \$REG0 \$REG1 \$REG2
1202	001264	000000	0	
1203				
1204			6	
1205				
1206	001266	025062	EM31	;'DRU' BIT SET, CHECK DRIVE
1207	001270	031537	DH30	;PC RKCS RKER RKDS
1208	001272	031344	DT26	;\$ERRPC \$REG0 \$REG1 \$REG2
1209	001274	000000	0	
1210				
1211			7	
1212				
1213	001276	025072	EM32	;'RKDS' BIT NOT SET
1214	001300	031575	DH34	;PC RKDS
1215	001302	031306	DT1	;\$ERRPC \$REG0
1216	001304	000000	0	
1217				
1218			10	
1219				
1220	001306	025113	EM33	;'DRY' NOT SET
1221	001310	031711	DH44	;PC RKCS RKER RKDS RKDA
1222	001312	031324	DT20	;\$ERRPC \$REG0 \$REG1 \$REG2 \$REG3
1223	001314	000000	0	
1224				

1225			; ITEM	11	
1226					
1227	001316	025133		EM34	; 'SOK' DID NOT SET
1228	001320	031575		DH34	; PC RKDS
1229	001322	031306		DT1	; \$ERRPC \$REGO
1230	001324	000000		0	
1231					
1232			; ITEM	12	
1233					
1234	001326	025152		EM35	; 'SEC COUNTR' DIDN'T COUNT TO 0
1235	001330	031613		DH35	; PC SEC-CNTR
1236	001332	031306		DT1	; \$ERRPC \$REGO
1237	001334	000000		0	
1238					
1239			; ITEM	13	
1240					
1241	001336	025205		EM36	; 'SEC COUNTR' DIDN'T INCREMENT
1242	001340	031633		DH36	; PC PRSNT-COUNT NXT-COUNT
1243	001342	031314		DT2	; \$ERRPC \$REGO \$REG1
1244	001344	000000		0	
1245					
1246			; ITEM	14	
1247					
1248	001346	025235		EM37	; 'SECTOR COUNTER' INCREMENTED WRONG
1249	001350	031431		DH4	; PC EXPCTD RECVD
1250	001352	031314		DT2	; \$ERRPC \$REGO \$REG1
1251	001354	000000		0	
1252					
1253			; ITEM	15	
1254					
1255	001356	025271		EM40	; DIDN'T GET SC=SA FOR THIS SECTOR
1256	001360	031663		DH40	; PC SECTOR RKDS
1257	001362	031314		DT2	; \$ERRPC \$REGO \$REG1
1258	001364	000000		0	
1259					
1260			; ITEM	16	
1261					
1262	001366	025331		EM41	; ERROR-'R/W/S RDY' SHOULD BE SET
1263	001370	031575		DH34	; PC RKDS
1264	001372	031306		DT1	; \$ERRPC \$REGO
1265	001374	000000		0	
1266					
1267			; ITEM	17	
1268					
1269	001376	024777		EM13	; RKBA ERROR
1270	001400	031431		DH4	; PC EXPCT RECVD
1271	001402	031314		DT2	; \$ERRPC \$REGO \$REG1
1272	001404	000000		0	
1273					
1274			; ITEM	20	
1275					
1276	001406	025366		EM43	; UNEXPECTED RK11 INTERRUPT
1277	001410	031532		DH21	; PC
1278	001412	031340		DT21	; \$ERRPC
1279	001414	000000		0	
1280					

1281			: ITEM	21	
1282					
1283	001416	025420		EM44	: 'CNTRL RDY' DIDN'T SET AFTER SEEK OR DRIVE RESET
1284	001420	031711		DH44	: PC RKCS RKER RKDS RKDA
1285	001422	031324		DT20	: \$ERRPC \$REG0 \$REG1 \$REG2 \$REG3.
1286	001424	000000		0	
1287					
1288			: ITEM	22	
1289					
1290	001426	025474		EM45	: 'ERR' OR 'HE' SET ON SEEK OR DRIVE RESET
1291	001430	031711		DH44	: PC RKCS RKER RKDS RKDA
1292	001432	031324		DT20	: \$ERRPC \$REG0 \$REG1 \$REG2 \$REG3
1293	001434	000000		0	
1294					
1295			: ITEM	23	
1296					
1297	001436	025542		EM46	: RKER BIT, ON SEEK OR DRIVE RESET
1298	001440	031537		DH30	: PC RKCS RKER RKDS
1299	001442	031344		DT26	: \$ERRPC \$REG0 \$REG1 \$REG2
1300	001444	000000		0	
1301					
1302			: ITEM	24	
1303					
1304	001446	025600		EM47	: RKCS CHANGED AFTER FUNCTION WAS DONE
1305	001450	031431		DH4	: PC EXPCT RECVD
1306	001452	031314		DT2	: \$ERRPC \$REG0 \$REG1
1307	001454	000000		0	
1308					
1309			: ITEM	25	
1310					
1311	001456	025642		EM50	: 'R/W/S RDY' DID NOT CLEAR
1312	001460	031537		DH30	: PC RKCS RKER RKDS
1313	001462	031344		DT26	: \$ERRPC \$REG0 \$REG1 \$REG2
1314	001464	000000		0	
1315					
1316			: ITEM	26	
1317					
1318	001466	025671		EM51	: 'R/W/S RDY' DIDN'T SET AFTER SEEK OR DRIVE RESET
1319	001470	031711		DH44	: PC RKCS RKER RKDS RKDA
1320	001472	031324		DT20	: \$ERRPC \$REG0 \$REG1 \$REG2 \$REG3
1321	001474	000000		0	
1322					
1323			: ITEM	27	
1324					
1325	001476	025744		EM52	: RKDA CHANGED AFTER SEEK
1326	001500	031431		DH4	: PC EXPCTD REGVD
1327	001502	031314		DT2	: \$ERRPC \$REG0 \$REG1
1328	001504	000000		0	
1329					
1330			: ITEM	30	
1331					
1332	001506	025771		EM53	: 'CNTRL RDY' DIDN'T CLEAR AS GO WAS SET
1333	001510	031537		DH30	: PC RKCS RKER RKDS
1334	001512	031344		DT26	: \$ERRPC \$REG0 \$REG1 \$REG2
1335	001514	000000		0	
1336					

54

1337			:ITEM	31	
1338					
1339	001516	026034		EM54	: 'CNTRL RDY' DIDN'T SET ON DOING WRITE/FMT STARTING
1340					: FROM <DSK-ADRES>
1341	001520	031756		DH54	: PC RKCS RKER RKDS RKDA
1342					: DRV# CYL <DSK-ADRES> SUR SECTR
1343	001522	031356		DT54	: \$ERRPC \$REG0 \$REG1 \$REG2 \$REG3
1344					: \$REG4 \$REG5 \$REG6 \$REG7
1345	001524	000000		0	
1346					
1347			:ITEM	32	
1348					
1349	001526	026126		EM55	: 'HE' OR 'ERR' ON WRITE/FMT STARTING FROM
1350					: <DSK-ADRES>
1351	001530	031756		DH54	: PC RKCS RKER RKDS RKDA
1352					: DRV# CYL <DSK-ADRES> SUR SECTR
1353	001532	031356		DT54	: \$ERRPC \$REG0 \$REG1 \$REG2 \$REG3
1354					: \$REG4 \$REG5 \$REG6 \$REG7
1355	001534	000000		0	
1356					
1357			:ITEM	33	
1358					
1359	001536	026205		EM56	: RKDA INCREMENTED WRONG ON WRITE OR WRITE FORMAT
1360	001540	032065		DH56	: PC EXPCT: DRV# CYL SUR SECTR
1361					: RECVD: DRV# CYL SUR SECTR
1362	001542	031356		DT54	: \$ERRPC \$REG0 \$REG1 \$REG2 \$REG3
1363					: \$REG4 \$REG5 \$REG6 \$REG7
1364	001544	000000		0	
1365					
1366			:ITEM	34	
1367					
1368	001546	026244		EM57	: RKWC DIDN'T OVERFLOW ON WRITE OR WRITE FORMAT
1369	001550	031457		DH5	: PC RECVD
1370	001552	031306		DT1	: \$ERRPC \$REG0
1371	001554	000000		0	
1372					
1373			:ITEM	35	
1374					
1375	001556	026302		EM60	: RKBA INCREMENTED WRONG ON WRITE OR WRITE FORMAT
1376	001560	031431		DH4	: PC EXPCT RECVD
1377	001562	031314		DT2	: \$ERRPC \$REG0 \$REG1
1378	001564	000000		0	
1379					
1380			:ITEM	36	
1381					
1382	001566	026341		EM61	: RKER SET, ON WRITE/READ/FORMAT
1383	001570	031537		DH30	: PC RKCS RKER RKDS
1384	001572	031344		DT26	: \$ERRPC \$REG0 \$REG1 \$REG2
1385	001574	000000		0	
1386					
1387			:ITEM	37	
1388					
1389	001576	026376		EM62	: RKDB ERROR
1390	001600	031431		DH4	: PC EXPCT RECVD
1391	001602	031314		DT2	: \$ERRPC \$REG0 \$REG1
1392	001604	000000		0	

1393									
1394			: ITEM	40					
1395									
1396	001606	026410		EM63	: RKDA INCREMENTED WRONG ON READ OR READ FORMAT				
1397	001610	032065		DH56	: PC EXPCT: DRV# CYL SUR SECTR				
1398					: RECVD: DRV# CYL SUR SECTR				
1399	001612	031356		DT54	: \$ERRPC \$REG0 \$REG1 \$REG2 \$REG3				
1400					: \$REG4 \$REG5 \$REG6 \$REG7				
1401	001614	000000		0					
1402									
1403			: ITEM	41					
1404									
1405	001616	026454		EM64	: RKWC DID NOT OVERFLOW ON READ OR READ FORMAT				
1406	001620	032172		DH64	: PC RKWC RKDA				
1407	001622	031314		DT2	: \$ERRPC \$REG0 \$REG1				
1408	001624	000000		0					
1409									
1410			: ITEM	42					
1411									
1412	001626	026517		EM65	: RKBA INCREMENTED WRONG ON READ OR READ FORMAT				
1413	001630	031431		DH4	: PC EXPCT RECVD				
1414	001632	031314		DT2	: \$ERRPC \$REG0 \$REG1				
1415	001634	000000		0					
1416									
1417			: ITEM	43					
1418									
1419	001636	026563		EM66	: INCORRECT HEADER FROM 'SECTOR'				
1420	001640	032216		DH66	: PC SECTR EXPCT RECVD				
1421	001642	031344		DT26	: \$ERRPC \$REG0 \$REG1 \$REG2.				
1422	001644	000000		0					
1423									
1424			: ITEM	44					
1425									
1426	001646	026622		EM67	: DATA ERROR				
1427	001650	032254		DH67	: PC EXPCT RECVD DSK-ADRES				
1428	001652	031344		DT26	: \$ERRPC \$REG0 \$REG1 \$REG2				
1429	001654	000000		0					
1430									
1431			: ITEM	45					
1432									
1433	001656	026635		EM70	: 'CNTRL RDY' DIDN'T SET ON DOING READ/FMT STARTING				
1434					: FROM <DSK-ADRES>				
1435	001660	031756		DH54	: PC RKCS RKER RKDS RKDA				
1436					: DRV# CYL <DSK-ADRES> SUR SECTR				
1437	001662	031356		DT54	: \$ERRPC \$REG0 \$REG1 \$REG2 \$REG3				
1438					: \$REG4 \$REG5 \$REG6 \$REG7				
1439	001664	000000		0					
1440									
1441			: ITEM	46					
1442									
1443	001666	026726		EM71	: 'HE' OR 'ERR' BIT SET ON READ/FMT STARTING				
1444					: FROM <DSK-ADRES>				
1445	001670	031756		DH54	: PC RKCS RKER RKDS RKDA				
1446					: DRV# CYL <DSK-ADRES> SUR SECTR				
1447	001672	031356		DT54	: \$ERRPC \$REG0 \$REG1 \$REG2 \$REG3				
1448					: \$REG4 \$REG5 \$REG6 \$REG7				

E03

MAINDEC-11-DZK-K-C MACY11 27(732) 16-SEP-76 16:00 PAGE 31
 DZK-KC.P11 ERROR POINTER TABLE

1449	001674	000000	0	
1450				
1451			; ITEM	47
1452				
1453	001676	027004	EM72	; WRONG DRIVE ID IN RKDS AFTER SEEK
1454	001700	031431	DH4	; PC EXPCT RECVD
1455	001702	031314	DT2	; SERRPC SREG0 SREG1
1456	001704	000000	0	
1457				
1458			; ITEM	50
1459				
1460	001706	027046	EM73	; HARDWARE POLL. DRIVE ID BITS(13-15) SHOULD BE CLEAR
1461	001710	031575	DH34	; PC RKDS
1462	001712	031314	DT2	; SERRPC SREG0
1463	001714	000000	0	
1464				
1465			; ITEM	51
1466				
1467	001716	027120	EM74	; HARDWARE POLL. INTERRUPTING DRIVE # NOT PRESENT
1468	001720	032314	DH74	; PC DRIVE #
1469	001722	031306	DT1	; SERRPC SREG0
1470	001724	000000	0	
1471				
1472			; ITEM	52
1473				
1474	001726	027170	EM75	; 'DRIVE #' DID NOT INTERRUPT DURING HARDWARE POLL
1475	001730	032314	DH74	; PC DRIVE #
1476	001732	031306	DT1	; SERRPC SREG0
1477	001734	000000	0	
1478				
1479			; ITEM	53
1480				
1481	001736	027240	EM76	; SCP DID NOT SET AFTER WAS DONE
1482	001740	032314	DH117	; PC RKCS
1483	001742	031306	DT1	; SERRPC SREG0
1484	001744	000000	0	
1485				
1486			; ITEM	54
1487				
1488	001746	027303	EM77	; RKDA CHANGED AFTER 'DRIVE RESET'
1489	001750	031431	DH4	; PC EXPCT RECVD
1490	001752	031314	DT2	; SERRPC SREG0 SREG1
1491	001754	000000	0	
1492				
1493			; ITEM	55
1494				
1495	001756	027340	EM100	; DATA ERROR AT WORD#
1496	001760	032335	DH100	; PC WORD# EXPCT RECVD
1497	001762	031344	DT26	; SERRPC SREG0 SREG1 SREG2
1498	001764	000000	0	
1499				
1500			; ITEM	56
1501				
1502	001766	027363	EM101	; CNTRL RDY DID NOT SET AFTER READ CHECK
1503	001770	031711	DH44	; PC RKCS RKER RKDS RKDA
1504	001772	031324	DT20	; SERRPC SREG0 SREG1 SREG2 SREG3

F03

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 32
 DZRKKC.P11 ERROR POINTER TABLE

1505	001774	000000	0	
1506				
1507			: ITEM	57
1508				
1509	001776	027425	EM102	: 'ERR' OF 'HE' SET ON READ CHECK
1510	002000	031537	DH30	: PC RKCS RKER RKDS
1511	002002	031344	DT26	: \$ERRPC \$REG0 \$REG1 \$REG2
1512	002004	000000	0	
1513				
1514			: ITEM	60
1515				
1516	002006	027451	EM103	: 'CSE' ON READ CHECK
1517	002010	032372	DH103	: PC RKER
1518	002012	031306	DT1	: \$ERRPC \$REG0
1519	002014	000000	0	
1520				
1521			: ITEM	61
1522				
1523	002016	027467	EM104	: RKWC DID NOT OVERFLOW ON READ CHECK OR WRITE CHECK
1524	002020	032406	DH104	: PC RECVD RKCS
1525	002022	031314	DT2	: \$ERRPC \$REG0 \$REG1
1526	002024	000000	0	
1527				
1528			: ITEM	62
1529				
1530	002026	027540	EM105	: RKDA INCREMENTED WRONG ON READ CHECK
1531	002030	031431	DH4	: PC EXPCT RECVD
1532	002032	031314	DT2	: \$ERRPC \$REG0 \$REG1
1533	002034	000000	0	
1534				
1535			: ITEM	63
1536				
1537	002036	027576	EM106	: RKBA CHANGED AFTER READ CHECK
1538	002040	031431	DH4	: PC EXPCT RECVD
1539	002042	031314	DT2	: \$ERRPC \$REG0 \$REG1
1540	002044	000000	0	
1541				
1542			: ITEM	64
1543				
1544	002046	027627	EM107	: MEMORY WORD CHANGED AFTER READ CHECK
1545	002050	032432	DH107	: PC LOC EXPCT RECVD
1546	002052	031344	DT26	: \$ERRPC \$REG0 \$REG1 \$REG2
1547	002054	000000	0	
1548				
1549			: ITEM	65
1550				
1551	002056	027670	EM110	: CNTRL RDY DID NOT SET AFTER WRITE CHECK
1552	002060	031711	DH44	: PC RKCS RKER RKDS RKDA
1553	002062	031324	DT20	: \$ERRPC \$REG0 \$REG1 \$REG2 \$REG3
1554	002064	000000	0	
1555				
1556			: ITEM	66
1557				
1558	002066	027733	EM111	: HE OR ERR BIT SET AFTER DOING WRITE CHECK
1559	002070	031537	DH30	: PC RKCS RKER RKDS
1560	002072	031344	DT26	: \$ERRPC \$REG0 \$REG1 \$REG2

G03

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 33
 DZRKKC.P11 ERROR POINTER TABLE

1561	002074	000000	0	
1562				
1563			:ITEM	67
1564				
1565	002076	027760	EM112	;WRITE CHECK ERROR
1566	002100	031537	DH30	;PC RKCS RKER RKDS
1567	002102	031344	DT26	;SERRPC \$REG0 \$REG1 \$REG2
1568	002104	000000	0	
1569				
1570			:ITEM	70
1571				
1572	002106	030001	EM113	;RKDA INCREMENTED WRONG ON WRITE CHECK
1573	002110	031431	DH4	;PC EXPCT RECVD
1574	002112	031314	DT2	;SERRPC \$REG0 \$REG1
1575	002114	000000	0	
1576				
1577			:ITEM	71
1578				
1579	002116	030040	EM114	;RKBA INCREMENTED WRONG ON WRITE CHECK
1580	002120	031431	DH4	;PC EXPCT RECVD
1581	002122	031314	DT2	;SERRPC \$REG0 \$REG1
1582	002124	000000	0	
1583				
1584			:ITEM	72
1585				
1586	002126	030077	EM115	;RKBA INCREMENTED WITH IBA SET
1587	002130	031431	DH4	;PC EXPCT RECVD
1588	002132	031314	DT2	;SERRPC \$REG0 \$REG1
1589	002134	000000	0	
1590				
1591			:ITEM	73
1592				
1593	002136	030133	EM116	;WRONG MEMORY LOCATION CHANGED WITH IBA SET
1594	002140	032335	DH100	;PC WORD# EXPCT RECVD
1595	002142	031344	DT26	;SERRPC \$REG0 \$REG1 \$REG2
1596	002144	000000	0	
1597				
1598			:ITEM	74
1599				
1600	002146	030206	EM117	;RK11 DID NOT INTERRUPT WHEN IDE WAS SET
1601	002150	032470	DH117	;PC RKCS
1602	002152	031306	DT1	;SERRPC \$REG0
1603	002154	000000	0	
1604				
1605			:ITEM	75
1606				
1607	002156	030253	EM120	;RK11 DID NOT INTERRUPT AFTER SEEK WAS INITIATED
1608	002160	032470	DH117	;PC RKCS
1609	002162	031306	DT1	;SERRPC \$REG0
1610	002164	000000	0	
1611				
1612			:ITEM	76
1613				
1614	002166	030326	EM121	;SCP SET BEFORE SEEK COMPLETED
1615	002170	032470	DH117	;PC RKCS
1616	002172	031306	DT1	;SERRPC \$REG0

H03

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 34
 DZRKKC.P11 ERROR POINTER TABLE

1617	002174	000000	0	
1618				
1619			; ITEM	77
1620				
1621	002176	030364	EM122	; RK11 DID NOT INTERRUPT AFTER SEEK COMPLETED
1622	002200	031537	DH30	; PC RKCS RKER RKDS
1623	002202	031344	DT26	; \$ERRPC \$REG0 \$REG1 \$REG2
1624	002204	000000	0	
1625				
1626			; ITEM	100
1627				
1628	002206	030433	EM123	; CNTRL RESET DID NOT CLEAR 'SCP' BIT
1629	002210	032470	DH117	; PC RKCS
1630	002212	031306	DT1	; \$ERRPC \$REG0
1631	002214	000000	0	
1632				
1633			; ITEM	101
1634				
1635	002216	030472	EM124	; RK11 DID NOT INTERRUPT AFTER READ WAS DONE
1636	002220	032470	DH117	; PC RKCS
1637	002222	031306	DT1	; \$ERRPC \$REG0
1638	002224	000000	0	
1639				
1640			; ITEM	102
1641				
1642	002226	030534	EM125	; CNTRL RESET DID NOT CLEAR REGISTER
1643	002230	031402	DH2	; PC REGADD RECVD
1644	002232	031314	DT2	; \$ERRPC \$REG0 \$REG1
1645	002234	000000	0	
1646				
1647			; ITEM	103
1648				
1649	002236	030573	EM126	; RK11 DID NOT INTERRUPT AT CPU LEVEL
1650	002240	032504	DH126	; PC LEVEL RKCS
1651	002242	031314	DT2	; \$ERRPC \$REG0 \$REG1
1652	002244	000000	0	
1653				
1654			; ITEM	104
1655				
1656	002246	030634	EM127	; RK11 INTERRUPTED AT WRONG CPU LEVEL
1657	002250	032504	DH126	; PC LEVEL RKCS
1658	002252	031314	DT2	; \$ERRPC \$REG0 \$REG1
1659	002254	000000	0	
1660				
1661			; ITEM	105
1662				
1663	002256	030676	EM130	; 'ERR BIT' DID NOT SET IN RKER
1664	002260	032532	DH130	; PC RKCS RKER ERR BIT
1665	002262	031344	DT26	; \$ERRPC \$REG0 \$REG1 \$REG2
1666	002264	000000	0	
1667				
1668				
1669			; ITEM	106
1670				
1671	002266	030733	EM131	; HE OR ERR DID NOT SET
1672	002270	032571	DH131	; PC RKCS RKER

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 35
 DZRKKC.P11 ERROR POINTER TABLE

1673	002272	031314	DT2	:SERRPC \$REG0	\$REG1
1674	002274	000000	0		
1675					
1676			:ITEM	107	
1677					
1678	002276	030760	EM132	:RKEP ERROR	
1679	002300	031431	DH4	:PC EXPCT RECVD	
1680	002302	031314	DT2	:SERRPC \$REG0	\$REG1
1681	002304	000000	0		
1682					
1683			:ITEM	110	
1684					
1685	002306	030772	EM133	:NXC BIT DID NOT SET	
1686	002310	032617	DH133	:PC RKCS RKEP RKDA	
1687	002312	031344	DT26	:PC \$REG0 \$REG1	\$REG2
1688	002314	000000	0		
1689					
1690			:ITEM	111	
1691					
1692	002316	031015	EM134	:RK11 DIDN'T INTERRUPT ON SOFT ERROR	
1693	002320	032571	DH131	:PC RKCS RKEP	
1694	002322	031314	DT2	:SERRPC \$REG0	\$REG1
1695	002324	000000	0		
1696					
1697			:ITEM	112	
1698					
1699	002326	031056	EM135	:MEX BITS INCREMENTED WRONG IN RKCS	
1700	002330	031431	DH4	:PC EXPCTD RECVD	
1701	002332	031314	DT2	:SERRPC \$REG0	\$REG1
1702	002334	000000	0		
1703					
1704			:ITEM	113	
1705					
1706	002336	027670	EM110	:CNTRL RDY DID NOT SET AFTER WRT CHK	
1707	002340	031475	DH14	:PC RKCS RKEP RKWC	
1708	002342	031344	DT26	:SERRPC \$REG0	\$REG1 \$REG2
1709	002344	000000	0		
1710					
1711			:ITEM	114	
1712					
1713	002346	031113	EM137	: 'WPS' NOT CLEAR	
1714	002350	031711	DH44	:PC RKCS RKEP RKDS RKDA	
1715	002352	031324	DT20	:SERRPC \$REG0	\$REG1 \$REG2 \$REG3
1716	002354	000000	0		
1717					
1718			:ITEM	115	
1719					
1720	002356	031131	EM140	:DATA ERROR ON TRANSFER FROM DISK TO TTY	
1721	002360	032655	DH140	:PC EXPCT RECVD RKBA RKCS	
1722	002362	031324	DT20	:SERRPC \$REG0	\$REG1 \$REG2 \$REG3
1723	002364	000000	0		
1724					
1725			:ITEM	116	
1726					
1727					
1728	002366	031200	EM141	: 'DRIVE #' PRESENT, BUT NOT SPECIFIED	

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 36
 DZRKKC.P11 ERROR POINTER TABLE

1729	002370	032314				DH74	:PC	DRIVE #
1730	002372	031306				DT1	:SERRPC	\$REGO
1731	002374	000000				0		
1732								
1733			:ITEM	117				
1734								
1735	002376	024752				EM11	:RKWC	ERROR
1736	002400	031431				DH4	:PC	EXPCT
1737	002402	031314				DT2	:SERRPC	\$REGO
1738	002404	000000				0		RECVD
1739			:ITEM	120				\$REG1
1740	002406	031244				EM142		
1741	002410	000000				0		
1742								
1743								
1744	002412	005015	051104	053111	MSG1:	.ASCIZ	<15><12>/DRIVE	PRESNT/
1745	002420	020105	051120	051505				
1746	002426	052116	000					
1747		002432				.EVEN		
1748	002432	005015	047516	042516	MSG2:	.ASCIZ	<15><12>/NONE/	
1749	002440	000						
1750								
1751	002441	015	041412	052116	MSG3:	.ASCIZ	<15><12>/CNT RDY	DIDN'T SET/
1752	002446	051040	054504	042040				
1753	002454	042111	023516	020124				
1754	002462	042523	000124					
1755								
1756	002466	005015	051104	053111	MSG4:	.ASCIZ	<15><12>/DRIVE /	
1757	002474	020105	000					
1758								
1759	002477	015	040412	046114	MSG5:	.ASCII	<15><12>/ALL DRVS/	
1760	002504	042040	053122	123				
1761								
1762	002511	040	051104	050117	MSG6:	.ASCIZ	/ DROPD/<15><12>	
1763	002516	006504	000012					
1764						.EVEN		
1765								
1766						:RK11	REGISTERS	
1767						:IF	FOR ANY REASON THE REGISTER ADDRESSES ARE DIFFERENT FROM THESE	
1768						:(GIVEN BELOW),	THE CONTENTS OF THE APPROPRIATE POINTERS SHOULD BE	
1769						:MODIFIED	SO THAT THE CORRECT ADDRESS IS USED.	
1770						:		
1771						.EVEN		
1772	002522	177400			RKDS:	177400		
1773	002524	177402			RKER:	177402		
1774	002526	177404			RKCS:	177404		
1775	002530	177406			RKWC:	177406		
1776	002532	177410			RKBA:	177410		
1777	002534	177412			RKDA:	177412		
1778	002536	177416			RKDB:	177416		
1779								
1780								
1781					:TAGS	AND GENERAL DATA AREA		
1782					:			
1783					:			
1784					:			

K03

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 37
 DZRKKC.P11 ERROR POINTER TABLE

1785	002540	000000	SIMUL: 0	; FLAG TO BE SET TO 1 WHEN ON SIMULATOR
1786	002542	000000	FTITLE: 0	; FLAG FOR PRINTING PROGRAM TITLE
1787	002544	000000	DRIVAD: 0	; CONTAINS ADDRESS OF THE DRIVE UNDER TEST
1788	002546	000000	DRVDON: 0	; CONTAINS THE NUMBER OF DRIVES CHECKED.
1789				; IT IS INCREMENTED EACH TIME THE TESTS FOR
1790				; A DRIVE IS COMPLETED.
1791	002550	000000	DRVPTR: 0	; CONTAINS THE POINTER TO THE DRIVE FLAG (DRIVED
1792				; -DRIVE?) OF THE DRIVE TO BE CHECKED NEXT.
1793	002552	000000	INDX1: 0	; GENERAL INDEX FOR KEEPING COUNT
1794	002554	000000	INDX2: 0	; GENERAL INDEX
1795	002556	000000	COUNT: 0	; GENERAL COUNT REGISTER
1796	002560	000000	COUNT1: 0	; COUNT REGISTER USED FOR 'DRESET' SUBROUTINE
1797	002562	000000	TIMER: 0	; TIMER REGISTER
1798	002564	000000	EFLG1: 0	; SET TO INDICATE A PARTICULAR
1799				; ERROR CONDITION
1800				
1801	002566	000100	SEEK0: 100	; CONTAINS ADDRESS OF CYLINDER 2
1802	002570	001000	SEEK1: 1000	; CONTAINS ADDRESS OF CYLINDER 20
1803	002572	014500	SEEK2: 14500	; CONTAINS ADDRESS OF CYLINDER 312
1804	002574	000200	RKPRI: 200	; CONTAINS THE CPU LEVEL AT WHICH
1805				; RK11 NORMALLY INTERRUPTS. THIS WORD
1806				; SHOULD BE CHANGED IF RK11 IS DESIGNATED
1807				; A BR LEVEL OTHER THAN 5. E.G. IF IT IS CHANGED
1808				; TO 6, THIS WORD SHOULD BE CHANGED TO 240.
1809	002576	000220	RKVEC: 220	; CONTAINS THE NORMAL VECTOR ADDRESS TO WHICH
1810				; RK11 INTERRUPTS. IF THIS IS NOT SO, CHANGE
1811				; THIS WORD TO CONTAIN MODIFIED VECTOR ADDRESS.
1812	002600	000000	DDPCH: 0	; FLAG- SET WHEN IN DDP CHAIN MODE
1813	002602	000000	DRIVS: 0	; CONTAINS THE NUMBER OF DRIVES PRESENT
1814	002604	000000	FFLAG: 0	
1815	002606	000000	ODDEVN: 0	; USED TO DETERMINE WHICH OF RK-05F DRIVES ACTIVE
1816				; 0 IF EVEN DRIVE
1817				; -1 IF ODD DRIVE
1818				
1819				
1820				
1821				
1822				; THE FLAGS BELOW (BIT 0) ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE
1823				; IS PRESENT AND IS TO BE TESTED. BIT 12, IF SET, INDICATES THAT THE DRIVE
1824				; WAS DROPPED AFTER MAXIMUM ALLOWABLE NUMBER OF ERRORS OCCURED ON THAT
1825				; DRIVE (SW 6 SET).
1826				; IF MORE THAN 5 ERRORS OCCUR IN THE HARDWARE POLLING TEST (LAST)
1827				; THEN ALL DRIVES ARE DROPPED. BUT BIT 12 IS NOT SET.
1828				
1829	002610	000000	DRIVO: 0	; FLAG SET TO 1 WHEN DRIVE 0 PRESENT
1830	002612	000000	DRIV1: 0	; FOR DRIVE 1
1831	002614	000000	DRIV2: 0	; FOR DRIVE 2
1832	002616	000000	DRIV3: 0	; FOR DRIVE 3
1833	002620	000000	DRIV4: 0	; FOR DRIVE 4
1834	002622	000000	DRIV5: 0	; FOR DRIVE 5
1835	002624	000000	DRIV6: 0	; FOR DRIVE 6
1836	002626	000000	DRIV7: 0	; FOR DRIVE 7
1837				
1838	002630	000000	T56FLG: 0	
1839	002632	000000	PHYDRV: 0	
1840	002634	000000	SIZYET: 0	

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 38
DZRKKC.P11 ERROR POINTER TABLE

L03

1841
1842

MO3

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGF 39
 DZRKKC.P11 ERROR POINTER TABLE

1843	002636				START:
1844					.SBTTL INITIALIZE THE COMMON TAGS
1845					::CLEAR THE COMMON TAGS (\$CMTAG) AREA
1846	002636	012706	001100		MOV \$CMTAG,R6 ;:FIRST LOCATION TO BE CLEARED
1847	002642	005026			CLR (R6)+ ;:CLEAR MEMORY LOCATION
1848	002644	022706	001140		CMP \$SWR,R6 ;:DONE?
1849	002650	001374			BNE -6 ;:LOOP BACK IF NO
1850	002652	012706	001100		MOV \$STACK,SP ;:SETUP THE STACK POINTER
1851					::INITIALIZE A FEW VECTORS
1852	002656	012737	021552	000020	MOV \$SCOPE,\$IOTVEC ;:IOT VECTOR FOR SCOPE ROUTINE
1853	002664	012737	000340	000022	MOV \$340,\$IOTVEC+2 ;:LEVEL 7
1854	002672	012737	022024	000030	MOV \$ERROR,\$EMTVEC ;:EMT VECTOR FOR ERROR ROUTINE
1855	002700	012737	000340	000032	MOV \$340,\$EMTVEC+2 ;:LEVEL 7
1856	002706	012737	024274	000034	MOV \$TRAP,\$TRAPVEC ;:TRAP VECTOR FOR TRAP CALLS
1857	002714	012737	000340	000036	MOV \$340,\$TRAPVEC+2 ;:LEVEL 7
1858	002722	012737	024360	000024	MOV \$SPWRDN,\$PWRVEC ;:POWER FAILURE VECTOR
1859	002730	012737	000340	000026	MOV \$340,\$PWRVEC+2 ;:LEVEL 7
1860	002736	005037	001206		CLR \$TIMES ;:INITIALIZE NUMBER OF ITERATIONS
1861	002742	005037	001210		CLR \$ESCAPE ;:CLEAR THE ESCAPE ON ERROR ADDRESS
1862	002746	112737	000001	001115	MOVB \$1,\$ERMAX ;:ALLOW ONE ERROR PER TEST
1863	002754	012737	002754	001106	MOV \$.,\$LPADR ;:INITIALIZE THE LOOP ADDRESS FOR SCOPE
1864	002762	012737	002762	001110	MOV \$.,\$LPERR ;:SETUP THE ERROR LOOP ADDRESS
1865					::SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1866					::EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1867	002770	013746	000004		MOV \$ERRVEC,-(SP) ;:SAVE ERROR VECTOR
1868	002774	012737	003030	000004	MOV \$64,\$ERRVEC ;:SET UP ERROR VECTOR
1869	003002	012737	177570	001140	MOV \$DSWR,\$SWR ;:SETUP FOR A HARDWARE SWICH REGISTER
1870	003010	012737	177570	001142	MOV \$DDISP,\$DISPLAY ;:AND A HARDWARE DISPLAY REGISTER
1871	003016	022777	177777	176114	CMP #-1,\$SWR ;:TRY TO REFERENCE HARDWARE SWR
1872	003024	001012			BNE 66\$;:BRANCH IF NO TIMEOUT TRAP OCCURRED
1873					::AND THE HARDWARE SWR IS NOT = -1
1874	003026	000403			BR 65\$;:BRANCH IF NO TIMEOUT
1875	003030	012716	003036		64\$: MOV \$65\$,(SP) ;:SET UP FOR TRAP RETURN
1876	003034	000002			
1877	003036	012737	000176	001140	65\$: MOV \$SWREG,\$SWR ;:POINT TO SOFTWARE SWR
1878	003044	012737	000174	001142	MOV \$DISPREG,\$DISPLAY
1879	003052	012637	000004		66\$: MOV (SP)+,\$ERRVEC ;:RESTORE ERROR VECTOR
1880					
1881	003056	000005			RESET
1882	003060	012700	002600		MOV \$DDPCH,\$R0
1883	003064	012701	177765		MOV #-13,\$R1
1884	003070	005020			1\$: CLR (R0)+
1885	003072	005201			INC \$R1
1886	003074	001375			BNE 1\$
1887					.SBTTL TYPE PROGRAM NAME
1888					::TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1889	003076	005227	177777		INC #-1 ;:FIRST TIME?
1890	003102	001045			BNE 67\$;:BRANCH IF NO
1891	003104	104400	003142		TYPE 68\$;:TYPE ASCIZ STRING
1892					.SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
1893	003110	005737	000042		TST \$42 ;:ARE WE RUNNING UNDER XXDP/ACT?
1894	003114	001006			BNE 69\$;:BRANCH IF YES
1895	003116	023727	001140	000176	CMP \$SWR,\$SWREG ;:SOFTWARE SWITCH REG SELECTED?
1896	003124	001005			BNE 70\$;:BRANCH IF NO
1897	003126	104405			GTSWR ;:GET SOFT-SWR SETTINGS
1898	003130	000403			BR 70\$

N03

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 40
DZRKKC.P11 GET VALUE FOR SOFTWARE SWITCH REGISTER

```

1899 003132 112737 000001 001134 69$: MOV B #1,$AUTOB ;;SET AUTO-MODE INDICATOR
1900 003140 70$: BR 67$ ;;GET OVER THE ASCIZ
1901 003140 000426 ;;68$: .ASCIZ <CRLF><15><12>/RK11 LOGIC TESTS-II MAINDEC-11-DZRKK-C/<CRLF>
1902 003216 67$:
1903
1904
1905
1906
1907 003216 012700 002602
1908 003222 012701 177767
1909 003226 005020
1910 003230 005201
1911 003232 001375
1912
1913 003234 005737 000042
1914 003240 001005
1915 003242 123727 000041 000002
1916 003250 001412
1917 003252 000466
1918 003254 123727 000041 000002 1$: CMPB @#41,#2
1919 003262 001443 BEQ 2$ ;;RK05 DDP DUMP MODE
1920 003264 105737 000041 BEQ 3$ ;;PAPER TAPE LOADER
1921 003270 001002 TSTB @#41
1922 003272 000137 003752 BNE 2$ ;;RK05 DDP CHAIN MODE
1923 003276 2$: JMP ST3 ;;NOT UNDER ACT11
1924 003276 104400 003304
1925 003302 000430
1926
1927 003364
1928 003364 104407
1929 003366 005726
1930
1931 003370 000417
1932 003372
1933 003372 104400 003400
1934 003376 000410
1935
1936 003420
1937 003420 012737 000001 002600
1938 003426 000551
1939
1940
1941
1942
1943
1944
1945
1946
1947 003430 012700 002602
1948 003434 012701 177767
1949 003440 005020
1950 003442 005201
1951 003444 001375
1952 003446 104400 003454
1953 003452 000414
1954

```

69\$: MOV B #1,\$AUTOB ;;SET AUTO-MODE INDICATOR
 70\$: BR 67\$;;GET OVER THE ASCIZ
 ;;68\$: .ASCIZ <CRLF><15><12>/RK11 LOGIC TESTS-II MAINDEC-11-DZRKK-C/<CRLF>
 67\$:
 ;FIND OUT IF ACT11, DDP- CHAIN OR DUMP MODE
 START1: MOV #DRIVS,R0
 MOV #-11,R1 ;CLEAR OUT DRIVE TABLE AREA
 10\$: CLR (R0)+
 INC R1
 BNE 10\$
 TST @#42
 BNE 1\$
 CMPB @#41,#2
 BEQ 2\$
 BR ST2
 1\$: CMPB @#41,#2
 BEQ 3\$
 TSTB @#41
 BNE 2\$
 JMP ST3
 2\$:
 TYPE ,65\$
 BR ,64\$
 ;;65\$: .ASCIZ <15><12>/REPLACE DRO DDP-PAK BY OTHER,TYP CR WHEN DONE/
 64\$: RDCHR ;READ CHARACTER FROM KEYBOARD
 TST (6)+ ;UPDATE THE STACK POINTER
 BR ST2 ;DDP DUMP MODE
 3\$: TYPE ,67\$
 BR ,66\$
 ;;67\$: .ASCIZ <15><12>/DRO NOT TSTD/
 66\$: MOV #1,DDPCH ;REPORT THAT DRIVE 0 WILL NOT BE TESTED
 BR ST3 ;ASCERTAIN NO OF DRIVES
 ;FIND OUT FROM USER WHICH DRIVES (LOGICAL ADDRESSES) ARE TO BE
 ;TESTED (DRIVS TO B TSTD?). IN REPLY THE USER SHOULD TYPE IN THE
 ;LOGICAL ADDRESSES SEPERATED BY COMMAS. THUS IF 2 DRIVES 0,1 ARE PRESENT:
 ; 'DRIVS TO B TSTD'
 ; '0,1<CR>' A CAR. RET. SHOULD BE TYPED TO TERMINATE THE LIST.
 ST2: MOV #DRIVS,R0
 MOV #-11,R1
 13\$: CLR (R0)+
 INC R1
 BNE 13\$
 TYPE ,65\$
 BR ,64\$
 ;;65\$: .ASCIZ <15><12>/DRIVS TO BE TSTD?/<15><12>

1955	003504		64\$:	RDLIN		
1956	003504	104410		MOV	(SP)+,R0	:GET STARTING ADRES OF ASCII STRING
1957	003506	012600		MOV	#-10,R1	:SET UP COUNT
1958	003510	012701	177770	MOV	(R0)+,R2	:GET ASCII CHARACTER
1959	003514	112002		BIC	#177400,R2	:MASK UNWANTED BITS
1960	003516	042702	177400	MOV	#DRIVO,R3	
1961	003522	012703	002610	MOV	#-10,R4	
1962	003526	012704	177770	MOV	#60,R5	
1963	003532	012705	000060	MOV	R5,R2	
1964	003536	020502		CMP	R5,R2	:WAS THE TYPED IN CHARACTER
1965						:A NUMBER BETWEEN 0-7?
1966	003540	001414		BEQ	3\$	:YES, BRANCH
1967	003542	005205		INC	R5	:NO, INCREMENT
1968	003544	005723		TST	(R3)+	:INCREMENT POINTER TO DRV FLAG
1969	003546	005204		INC	R4	:CHARACTER THAT WAS INPUT
1970	003550	001372		BNE	2\$	:SHOULD BE 0-7, IF ANY OTHER
1971						:TYPE ?? & AGAIN ASK FOR
1972						:DRIVES TO BE TSTD?
1973	003552	005702		TST	R2	:IS IT A TERMINATOR?
1974	003554	001461		BEQ	6\$	:YES, EXIT. NO DRIVES INDICATED.
1975	003556		4\$:			
1976	003556	104400	003564	TYPE	,67\$	::TYPE ASCIZ STRING
1977	003562	000402		BR	66\$	::GET OVER THE ASCIZ
1978			66\$:	.ASCIZ	???	
1979	003570					
1980	003570	000717		BR	ST2	:GO, AGAIN ASK QUESTION
1981	003572	005713		TST	DR3	:SEE IF ALL READY SELECTED
1982	003574	001370		BNE	4\$	:ERROR IF SELECTED ALL READY
1983	003576	005213		INC	DR3	:SET UP FLAG FOR THE DRIVE
1984	003600	005237	002602	INC	DRIVS	:INCREMENT TOTAL NO OF DRIVES PRESENT
1985	003604	111002		MOV	DR0,R2	:GET NEXT CHAR
1986	003606	042702	177400	BIC	#177400,R2	:CHARACTER ONLY
1987	003612	022702	000106	CMP	#'F,R2	:IS IT F?
1988	003616	001026		BNE	8\$	:NO, GO ON
1989	003620	052713	100000	BIS	#BIT15,DR3	:SET BIT 15 TO SHOW RK05F
1990	003624	032705	000001	BIT	#BIT0,R5	:EVEN DRIVE?
1991	003630	001407		BEQ	9\$	:EVEN DRIVE SO BRANCH
1992	003632	005763	177776	TST	-2(R3)	:CHECK EVEN DRIVE
1993	003636	001347		BNE	4\$	:EVEN ALL READY SELECTED
1994	003640	012763	100001	MOV	#BIT15:BIT0,-2(R3)	:SELECT EVEN DRIVE
1995	003646	000406		BR	10\$	:CONTINUE
1996	003650	005763	000002	TST	2(R3)	:CHECK ODD DRIVE
1997	003654	001340		BNE	4\$	:ERROR IF SELECTED BEFORE
1998	003656	012763	100001	MOV	#BIT15:BIT0,2(R3)	:SELECT ODD DRIVE
1999	003664	005237	002602	INC	DRIVS	:COUNT DRIVES SELECTED
2000	003670	105720		TSTB	(R0)+	:POINT TO NEXT CHAR
2001	003672	000744		BR	11\$	:CHECK FOR COMMA
2002	003674	022702	000054	CMP	#54,R2	:IS IT A 'COMMA'?
2003	003700	001403		BEQ	5\$	:YES, GO PROCESS NXT WORD
2004	003702	005702		TST	R2	:NO, IS IT A TERMINATOR?
2005	003704	001324		BNE	4\$	:IF NOT, SOMETHING WRONG
2006						:GO ASK QUESTION AGAIN
2007	003706	000404		BR	6\$	:EXIT, IF A TERMINATOR
2008	003710	105720		TSTB	(R0)+	:INCREMENT PTR TO NXT BYTE
2009						:IN INPUT BUFFER
2010	003712	005201		INC	R1	:THERE SHOULD BE NO MORE THAN

2011	003714	001277			BNE	15		: 8 DRIVES, HENCE IF MORE
2012	003716	000717			BR	45		: THAN 8 DIFFERENT NOS. TYPED IN, ERROR!
2013								: GO AGAIN ASK THE QUESTION
2014								
2015	003720	005037	002634	65:	CLR	SIZYET		: NO SIZING NEEDED
2016	003724	032777	002000	175206	BIT	#SW10,JSWR		: TESTING ON SIMULATOR?
2017	003732	001003			BNE	75		: YES, BRANCH
2018	003734	005037	002540		CLR	SIMUL		: NO, CLR FLAG
2019	003740	000502			BR	ST4		
2020								
2021	003742	012737	000001	002540	75:	MOV	#1,SIMUL	: SET FLAG TO INDICATE SIMULATOR
2022	003750	000476			BR	ST4		
2023								
2024								
2025								
2026								
2027	003752	012737	177777	002634	ST3:	MOV	#-1,SIZYET	: CHECK FOR RKDSF LATER
2028	003760	012737	004132	000004		MOV	#55,JS4	: SET UP ADRES FOR TIME-OUT VECTOR
2029	003766	005777	176530			TST	ARKDS	: REFERENCE RKDS
2030	003772	005777	176536			TST	ARKDA	: REFERENCE RKDA
2031	003776	012737	004224	000004		MOV	#BADTMO,JS4	
2032	004004	104400				TYPE		
2033	004006	002412				MSG1		
2034	004010	012700	177770			MOV	#-10,R0	: INITIALIZE COUNT FOR THE 8 DRIVES
2035	004014	005037	002602			CLR	DRIVS	: INITIALIZE # OF DRIVES PRESENT TO 0
2036	004020	005001				CLR	R1	: INITIALIZE ADDRESS TO DRIVE 0
2037	004022	005004				CLR	R4	
2038	004024	012702	002610			MOV	#DRIVO,R2	
2039	004030	010177	176500	15:		MOV	R1,ARKDA	: ADDRESS THE DRIVE
2040	004034	020177	176474			CMP	R1,ARKDA	: CHECK, WAS IT ADDRESSED?
2041	004040	001405				BEQ	35	: YES
2042	004042	012703	004046			MOV	#25,R3	
2043	004046	004737	020440	25:		JSR	PC,TERM	: WHILE CHECKING NUMBER OF DRIVE
2044								: UNDER NON-MANUAL MODE :-
2045								: RKDA HAD TO BE ADRESED BUT
2046								: IT WAS FOUND THAT THE DRIVE NO
2047								: THAT WAS WRITTEN COULD NOT BE READ BACK
2048								: CORRECTLY.
2049								
2050	004052	000413				BR	45	
2051	004054	032777	000200	176440	35:	BIT	#200,ARKDS	: CHECK IF 'DRY' BIT IS SET. IF SET DRIVE IS
2052								: PRESENT
2053	004062	001407				BEQ	45	
2054	004064	104400				TYPE		
2055	004066	001213				\$CRLF		
2056	004070	005237	002602			INC	DRIVS	: IF PRESENT, INCREMENT # OF DRIVES
2057	004074	005212				INC	(R2)	: SET UP FLAG INDICATING THIS DRIVE PRESENT
2058	004076	010446				MOV	R4,-(SP)	
2059	004100	104401				TYPOC		
2060	004102	005722			45:	TST	(R2)+	: SHIFT POINTER TO NXT DRIVE INDICATOR
2061	004104	062701	020000			ADD	#20000,R1	: SET UP ADDRESS FOR THE NEXT DRIVE
2062	004110	005204				INC	R4	: HAVE U CHECKED FOR ALL 9 DRIVES
2063	004112	005200				INC	R0	
2064	004114	001345				BNE	15	
2065	004116	005737	002602			TST	DRIVS	
2066	004122	001011				BNE	ST4	


```

2067 004124 104400
2068 004126 002432
2069 004130 000406
2070
2071 004132 011603
2072 004134 022626
2073 004136 062703 177776
2074 004142 004737 020440
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084 004146 005037 002630
2085 004152 005737 002602
2086 004156 001004
2087 004160 004737 021354
2088 004164 000137 020264
2089 004170 012737 002610 002550
2090 004176 005037 002546
2091
2092 004202 005037 002544
2093
2094 004206 012737 004224 000004
2095
2096 004214 012777 004270 176354
2097
2098 004222 000465
2099
2100
2101
2102
2103
2104
2105 004224 011600
2106 004226 005740
2107 004230 022626
2108 004232 104400 004240
2109 004236 000407
2110
2111 004256
2112 004256 010046
2113 004260 104401
2114 004262 000000
2115 004264 000137 002636
2116
2117
2118
2119
2120
2121
2122

TYPE MSG2
BR ST4
5$: MOV (SP),R3
CMP (SP)+,(SP)+
ADD #-2,R3
JSR PC,TYERM
;GO CHECK THE DRIVE INDEPENDENT
;CONTROLLER LOGIC
;GET PC WHERE TIMEOUT OCCURED
;RESTORE STACK
;GO TYPE ERROR MESSAGE
;WHILE CHECKING FOR THE NUMBER OF
;DRIVES IN NON-MANUAL MODE:-
;RKDS AND RKDA HAD TO BE REFERENCED, TIMEOUT
;OCCURED ON REFERENCING.PC IN THE ERROR
;MESSAGE INDICATES WHERE THE TIMEOUT OCCURED.
:
ST4: CLR T56FLG
TST DRVS
BNE 1$
JSR PC,WATIME
JMP $EOP
1$: MOV #DRIVO,DRVPTR
CLR DRVDON
;INITIALIZE THE NO. OF DRIVES
;THAT HAVE BEEN CHECKED
;INITIALIZE DRIVE ADDRESS TO
;THE FIRST DRIVE
MOV #BADTMO,2#4
;SET TIME OUT VECTOR FOR UNEXPECTED
;TIME OUTS
MOV #BADINT,2#KVEC
;SET UP RK11 INTERRUPT VECTOR FOR
;UNEXPECTED INTERRUPTS FROM RK11
BR TST1
;GO TO TEST 1

;THIS ROUTINE HANDLES UNEXPECTED TIME OUTS
BADTMO: MOV (SP),R0 ;SAVE PC WHERE TIME OUT OCCURED
TST -(R0)
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 R0,-(SP) ;SET UP FOR TYPING OUT PC
TYPOC ;GO TYPE OUT OCTAL PC
HALT
JMP 2#START

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

```

```

2123
2124 004270 011600      BADINT: MOV      (SP),RO      ;SAVE PC WHERE INTERRUPT OCCURED
2125 004272 005740      TST      -(RO)
2126 004274 032777 020000 174636  BIT      #20000,ASWR  ;INHIBIT ERROR TYPEOUT"
2127 004302 001014      BNE      1$      ;YES, DON'T TYPE OUT
2128 004304 104400      TYPE
2129 004306 001213      $CRLF
2130 004310 104400      TYPE
2131 004312 025366      EM43      ;TYPE 'UNEXPE'ED RK11 INTERRUPT'
2132                                ;TYPE ' AT PC='
2133 004314 104400 004322  TYPE      65$      ;TYPE ASCII STRING
2134 004320 000403      BR      64$      ;GET OVER THE ASCII
2135                                ;:65$: .ASCIIZ /,PC=/
2136 004330                                64$:
2137 004330 010046      MOV      RO,-(SP)      ;SET UP FOR TYPING OUT PC
2138 004332 104401      TYPOC      ;GO TYPE OCTAL PC WHERE BAD
2139                                ;INTERLUPT OCCURED
2140 004334 032777 001000 174576 1$: BIT      #1000,ASWR  ;LOOP ON ERROR?
2141 004342 001403      BEQ      2$      ;NO, BRANCH
2142 004344 022626      CMP      (SP)+,(SP)+      ;YES, REPOSITION STACK
2143 004346 000177 174534      JMP      ASLPADR      ;GO TO THE STARTING ADDRESS OF
2144                                ;THE TEST THAT GAVE UNEXPECTED INTERRUPT
2145 004352 032777 040000 174560 2$: BIT      #40000,ASWR ;LOOP ON TEST?
2146 004360 001401      BEQ      3$      ;NO, BRANCH
2147 004362 000002      RTI
2148 004364 000000      3$: HALT      ;YES, LOOP. GO BACK WHEN U INTERRUPTED FROM.
2149                                ;UNEXPECTED INTERRUPT OCCURED AS
2150                                ;INDICATED IN THE TYPE OUT.U CAN LOOP
2151                                ;ON ERROR, TEST OR INHIBIT TYPEOUT BY
2152 004366 000137 002636      JMP      ASSTART      ;SETTING APPROPRIATE SWITCHES.
2153                                ;GO BACK TO THE START OF THE
2154                                ;PROGRAM. THUS PRESSING CONTINUE
2155                                ;AFTER THE ABOVE HALT WILL
2156                                ;RESTART THE PROGRAM
2157
2158
2159      ;RESTART AFTER POWER FAIL
2160      ;THE PROGRAM WOULD RESTART HERE IF POWER CAME BACK AFTER A FALIURE.
2161
2162 004372 004737 021354  PFSTRT: JSR      PC,WATIME      ;KILL TIME
2163
2164
2165
2166      ;*****
2167      ;*TEST 1      CHECK THAT THE DRIVES THAT ARE NOT SPECIFIED ARE NOT FOUND TO BE PRESENT
2168      ;*THIS TEST CHECKS THAT THE DRIVES THAT ARE NOT SPECIFIED
2169      ;*(IN RESPONSE TO "DRIVS TO BE TSTD") ARE NOT FOUND TO BE PRESENT.
2170      ;*EVERY DRIVE FROM 0 TO 7 IS ADDRESSED. IF A PARTICULAR DRIVE
2171      ;*GIVES 'DRY' (IN RKDS), IT IS CHECKED THAT THIS DRIVE
2172      ;*WAS SPECIFIED BY THE USER, IF IT WAS NOT AN ERROR IS
2173      ;*REPORTED, GIVING THE DRIVE NUMBER. IT IS LIKELY THAT THE USER
2174      ;*MAY HAVE FORGOTTEN TO PUT THE DRIVE (THAT IS NOT SPECIFIED) ON
2175      ;*'LOAD'. IF THIS IS THE CASE THEN PUT THIS DRIVE ON 'LOAD'.
2176      ;*IF THIS IS NOT THE CASE, THERE IS A GENUINE ERROR. (TWO DIFFERENT
2177      ;*DRIVE ADDRESSES MAY BE RESULTING IN THE SELECTION OF THE SAME
2178      ;*PHYSICAL DRIVE.)

```

```

2179          ;*****
2180 004376 000004          1ST: SCOPE
2181
2182 004400 012700 002610      MOV    #DRIVO,R0      ;INITIALIZE POINTER
2183 004404 005001          CLR    R1              ;INITIALIZE DRIVE ADRES 0
2184 004406 005002          CLR    R2              ;INITIALIZE DRIVE # 0
2185
2186 004410 010177 176120      1$: MOV    R1,RKDA      ;ADRES THE DRIVE
2187 004414 105777 176102      TSTB   RKDS          ;DRIVE READY?
2188 004420 100005          BPL     2$              ;NO, THIS DRIVE NOT PRESENT
2189
2190 004422 305710          TST     2R0             ;YES, THIS DRIVE SELECTED
2191
2192 004424 001026          BNE     3$              ;WAS THIS DRIVE SPECIFIED BY
2193
2194
2195
2196 004426 010237 001162      MOV    R2,$REGO      ;THE USER?
2197 004432 104116          ERROR  116             ;YES, OK
2198
2199
2200
2201
2202
2203
2204
2205 004434 005710          2$: TST     2R0          ;NO, THIS DRIVE # WAS NOT SPECIFIED
2206 004436 001421          BEQ     3$              ;BY THE USER, BUT STILL IS GIVING
2207
2208 004440 004737 020406      JSR     PC,GT4RG      ;'DRY' WHEN ADRESED. REPORT EROR.
2209 004444 104010          ERROR  10              ;GET DRIVE #
2210
2211
2212
2213
2214
2215 004446 005010          CLR     2R0             ;THIS DRIVE # WAS NOT SPECIFIED BY
2216
2217 004450 010003          MOV    R0,R3            ;THE USER, BUT WHEN ADRESED GAVE
2218 004452 162703 002610      SUB     #DRIVO,R3      ;'DRY'. CHECK THAT THIS DRIVE # IF
2219 004456 042703 000003      BIC     #3,R3          ;PHYSICALLY PRESENT IS ON 'LOAD'. IF
2220 004462 062703 002610      ADD     #DRIVO,R3      ;THIS IS NOT THE CASE, THEN ONE DRIVE
2221 004466 042723 100000      BIC     #100000,(R3)+  ;MAY BE GETTING SELECTED BY TWO DIFFERENT
2222 004472 042713 100000      BIC     #100000,(R3)  ;LOGICAL ADDRESSES.
2223 004476 005337 002602      DEC     DRIVS        ;CHECK THAT THIS DRIVE WAS NOT INDICATED
2224
2225 004502 005202          3$: INC     R2          ;IF IT WAS, & IT IS NOT FOUND TO BE
2226 004504 005720          TST     (R0)+          ;PRESENT (DRY CLEAR), REPORT ERROR.
2227 004506 062701 020000      ADD     #20000,R1      ;GET RKCS, ER, DS, DA
2228 004512 001336          BNE     1$              ;DRIVE # (AS IN RKDA) WAS INDICATED BY
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
2263
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
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334

```

```

2235                                     ;DRIVE BEING CURRENTLY CHECKED.
2236                                     ;
2237 004514      NUDRV:
2238
2239
2240                                     ;*****
2241                                     ;*TEST 2      FIND OUT NEXT DRIVE TO BE CHECKED
2242                                     ;THIS CODE FINDS OUT THE NEXT DRIVE THAT IS PRESENT AND THEN  SETS UP
2243                                     ;THE ADDRESS IN DRIVAD (BITS 13,14,15).  THUS THROUGHOUT THE FOLLOWING TESTS
2244                                     ;THE DRIVE TESTED IS THE DRIVE WHOOSE ADDRESS IS IN 'DRIVAD'.
2245                                     ;*****
2246 004514  000004      TST2:  SCOPE
2247 004516  012737      MOV      #1,$TIMES      ;DO 1 ITERATION
2248 004524  012737      MOV      #2,$TSTNM      ;RESET POINTER TO THIS TEST
2249                                     ;NO. CHANGE THIS (2) IN CASE THE
2250                                     ;TEST NO. CHANGES
2251 004532  005037      CLR      $ERTTL          ;CLEAR TOTAL ERROR COUNT
2252 004536  005737      TST      DRVS           ;R THERE ANY DRIVES PRESENT?
2253 004542  001002      BNE      +6             ;YES, BRANCH
2254 004544  000137      JMP      $EOP          ;NO, JMP TO THE END
2255 004550  013701      MOV      DRVPTR,R1      ;GET THAT POINTER TO THE NEXT
2256                                     ;DRIVE FLAG
2257 004554  032721      2$:      BIT      #BIT0,(R1)+ ;IS THIS DRIVE PRESENT?
2258 004560  001005      BNE      1$            ;YES
2259 004562  062737      6$:      ADD      #20000,DRIVAD ;FORM NXT DRIVE ADRES
2260 004570  001371      BNE      2$
2261 004572  000764      BR      4$
2262 004574  005737      1$:      TST      DDPCH          ;DDP CHAIN MODE?
2263 004600  001403      BEQ      3$            ;NO, BRANCH
2264 004602  005737      TST      DRIVAD         ;YES, IS THIS DRIVE 0?
2265 004606  001765      BEQ      6$            ;IF YES, DON'T TEST DRIVE 0
2266 004610  010137      3$:      MOV      R1,DRVPTR    ;STORE POINTER TO THE NEXT
2267                                     ;DRIVE FLAG
2268 004614  104400      TYPE      MSG4
2269 004620  013746      MOV      DRIVAD,-(R6)    ;GET THE DRIVE ADDRESS
2270 004624  004737      JSR      PC,SHFRT      ;GO SHIFT IT TO THE RIGHT
2271 004630  005037      CLR      FFLAG
2272 004634  011600      MOV      (R6),R0        ;DRIVE NUMBER
2273 004636  104402      TYPOS          ;GO TYPE THE OCTAL # FOR THE
2274                                     ;DRIVE THAT IS BEING CHECKED
2275 004640      001      000
2276 004642  006300      ASL      R0              ;INDEX TO TABLE
2277 004644  005760      TST      DRIVO(R0)      ;SEE IF F
2278 004650  100006      BPL      5$            ;NO
2279 004652  104400      TYPE      65$         ;TYPE ASCIZ STRING
2280 004656  000401      BR      64$           ;GET OVER THE ASCIZ
2281                                     ;65$: .ASCIZ /F/
2282 004662      64$:
2283 004662  005237      INC      FFLAG          ;SET F FLAG
2284 004666  104400      5$:      TYPE
2285 004670  001213      $CRLF          ;TYPE CR, LF
2286                                     ;*****
2287                                     ;*TEST 3      CHECK THAT DRIVE IS SUPPLIED WITH POWER-DPL BIT
2288                                     ;*****
2289 004672  000004      TST3:  SCOPE
2290 004674  104412      CNT.RESET      ;GO, DO CONTROL RESET

```

H04

MAINDEC-11-DZRKK-C
DZRKKC.P11 T3MACY11 27(732) 16-SEP-76 16:00 PAGE 47
CHECK THAT DRIVE IS SUPPLIED WITH POWER-DPL BIT

```

2291
2292
2293
2294
2295
2296
2297
2298
2299
2300 004676 013700 002522      MOV      RKDS,RO
2301 004702 013777 002544 175624  MOV      DRIVAD,ARKDA
2302 004710 005710      TST      AR0
2303 004712 001003      BNE      1$
2304 004714 011037 001162      MOV      AR0,$REGO
2305 004720 104004      ERROR    4
2306
2307 004722 012777 000015 175576 1$:  MOV      #15,ARKCS
2308
2309 004730 005001      CLR      R1
2310 004732 032710 010000      2$:  BIT      #10000,AR0
2311 004736 001003      BNE      3$
2312 004740 005201      INC      R1
2313 004742 001373      BNE      2$
2314 004744 000403      BR       4$-2
2315 004746 004737 020414      3$:  JSR      PC,GT3RG
2316 004752 104005      ERROR    5
2317
2318
2319 004754 005001      CLR      R1
2320 004756 032710 000100      4$:  BIT      #100,AR0
2321 004762 001010      BNE      TST4
2322 004764 104416 000011      DELAY    11
2323 004770 005201      INC      R1
2324 004772 001371      BNE      4$
2325 004774 017737 175522 001162      MOV      ARKDS,$REGO
2326 005002 104016      ERROR    16
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336 005004 000004
2337 005006 104412
2338
2339
2340
2341
2342
2343
2344
2345
2346

```

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

; ADDRESS THE DRIVE UNDER TEST
 ; CHECK IF ANY BIT OF RKDS IS SET?
 ; IF SET, BRANCH
 ; GET RKDS
 ; RKDS ERROR! RKDS IF ADDRESSED
 ; CORRECTLY SHOULD BE NON-ZERO
 ; ISSUE A DRV RESET, IF DRIVE
 ; POWER IS LO, DPL WILL SET

; IS 'DPL' BIT SET?
 ; DPL IS SET, BRANCH
 ; WAIT FOR SOME TIME TO
 ; SEE IF DPL WOULD SET
 ; OK, DPL NOT SET
 ; GO, GET RKCS, ER, DS
 ; DPL BIT OF RKDS IS SET, CHECK DRIVE POWER

; DID R/W/S RDY BIT SET?
 ; YES, EXIT
 ; TIME DELAY
 ; WAIT FOR R/W/S RDY

; GET RKDS
 ; R/W/S RDY DID NOT SET AFTER
 ; DRIVE RESET. DRIVE RESET WAS DONE
 ; TO CHECK 'DPL' BIT. THIS TEST
 ; IS NOT FOR CHECKING DRIVE RESET.
 ; U MIGHT WANT TO USE THE TEST PROVIDED
 ; FOR CHECKING DRIVE RESET.

; *****
 ; *TEST 4 CHECK THAT 'DRIVE UNSAFE' IS CLEAR, 'MDEN' IS SET, 'WPS' IS CLEAR
 ; *****
 TST4: SCOPE
 CNT.RESET

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

MAINDEC-11-DZRKK-C
 DZRKKC.P11 T4

 MACY11 27(732) 16-SEP-76 16:00 PAGE 48
 CHECK THAT 'DRIVE UNSAFE' IS CLEAR, 'HDEN' IS SET, 'WPS' IS CLEAR

```

2347 005010 013777 002544 175516      MOV    DRIVAD,ARKDA    ;SET DRIVE ADDRESS
2348 005016 017700 175500                MOV    ARKDS,R0      ;GET RKDS
2349 005022 032700 002000                BIT    #2000,R0      ;IS 'DRU' BIT OF RKDS SET?
2350 005026 001403                BEQ    1$              ;NO
2351 005030 004737 020414                JSR    PC,GT3RG      ;GO, GET RKCS, ER, DS
2352 005034 104006                ERROR   6              ;'DRU' BIT OF RKDS IS SET, CHECK
2353                                ;DRIV BY PUTTING RUN/LOAD SW TO LOAD
2354                                ;THEN BACK TO RUN
2355 005036 032700 004000      1$:      BIT    #4000,R0      ;IS 'HDEN' BIT SET?
2356 005042 001004                BNE    2$              ;YES, BRANCH
2357 005044 017737 175452 001162        MOV    ARKDS,$REGO    ;GET RKDS
2358 005052 104007                ERROR   7              ;ERROR, 'RKDS' BIT IS NOT SET
2359
2360 005054 032777 000040 175440 2$:      BIT    #40,ARKDS    ;IS 'WPS' CLEAR?
2361 005062 001403                BEQ    TST5            ;YES, EXIT
2362 005064 004737 020406                JSR    PC,GT4RG      ;GET RKCS, ER, DS, DA
2363 005070 104114                ERROR   114           ;'WPS'-WRITE PROTECT STATUS- BIT OF
2364                                ;OF RKDS SHOULD BE CLEAR, IF THIS DRIVE
2365                                ;IS WRITE ENABLED. CHECK & SEE IF THIS
2366                                ;DRIVE IS WRITE ENABLED, IF IT IS NOT.
2367                                ;WRITE ENABLE IT.
2368
2369
2370
2371
2372
2373 005072 000004      - TST5:      SCOPE
2374 005074 104412                CNT.PESET
2375                                ;GO, DO CONTROL RESET
2376                                ;THIS IS A CALL FOR THE 'CNTRL-
2377                                ;RESET' ROUTINE. A CONTROL RESET IS
2378                                ;ISSUED AND AFTER A CERTAIN TIME
2379                                ;IF THE 'CNTRL RDY' DOES NOT SET
2380                                ;AN ERROR IS REPORTED. NOTE THAT
2381                                ;THE PC IN ERROR MESSAGE IS THE
2382                                ;PC WHERE 'CNT.RESET' IS LOCATED.
2383                                ;THIS IS A VERY BASIC ERR & IF IT
2384                                ;OCCURS GO BACK TO TEST 10
2385 005076 013777 002544 175430      MOV    DRIVAD,ARKDA    ;ADDRS THE DRIVE
2386 005104 105777 175412                TSTB   ARKDS          ;IS 'DRY' SET?
2387 005110 100403                BMI    TST6            ;YES, OK
2388 005112 004737 020406                JSR    PC,GT4RG      ;GO, GET RKCS, ER, DS, DA
2389 005116 104010                ERROR   10           ;'DRY' NOT SET
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402

```

 ;*TEST 5 CHECK THAT 'DRIVE READY' IS SET IN RKDS

 ;* THIS TEST CHECKS THAT WITHIN A CERTAIN TIME
 ;* 'SOK' BIT CAN SET, IF IT DOES NOT AN ERROR IS REPORTED

 ;*TEST 6 CHECK THAT 'SOK' BIT CAN SET

 ;* THIS TEST CHECKS THAT WITHIN A CERTAIN TIME
 ;* 'SOK' BIT CAN SET, IF IT DOES NOT AN ERROR IS REPORTED

```

2395 005120 000004      TST6:      SCOPE
2396 005122 013777 002544 175404      MOV    DRIVAD,ARKDA    ;ADDRS THE DRIVE
2397 005130 005001                CLR    R1              ;INITIALIZE COUNT FOR TIMING WAIT LOOP
2398 005132 032777 000400 175362 1$:      BIT    #400,ARKDS    ;IS SOK SET?
2399 005140 001006                BNE    TST7            ;EXIT
2400 005142 005201                INC    R1              ;NO, WAIT
2401 005144 001372                BNE    1$              ;WAITED LONG?
2402 005146 017737 175350 001162        MOV    ARKDS,$REGO    ;GET RKDS

```


J04

MAINDEC-11-DZRKK-C
DZRKKC.P11 T6MACY11 27(732) 16-SEP-76 16:00 PAGE 49
CHECK THAT 'SOK' BIT CAN SET

2403 005154 104011

ERROR 11

;WAITED LONG BUT 'SEC OK' BIT DID NOT
;SET2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458005156 000004
005160 104412005162 013777 002544 175344
005170 013700 002522
005174 005037 002552
005200 005005
005202 012704 177764
005206 012703 000001
005212 005037 002554
005216 005237 002552
005222 001440
005224 005237 002554
005230 001441
005232 011001
005234 032701 000400
005240 001771
005242 021001
005244 001362
005246 042701 177760
005252 001357

```

*****
;TEST 7 CHECK THAT 'SECTOR COUNTER' CAN COUNT FROM 0-13
; * THIS TEST CHECKS THAT THE SECTOR COUNTER CAN COUNT FROM
; * 0-13
; * 1) FIRST, FOR INITIALIZING PURPOSES THERE IS A TIMED LOOP
; * DURING WHICH SECTOR COUNTER SHOULD COUNT DOWN TO 0. IF THIS
; * IS NOT DONE AN ERROR IS REPORTED
; * 2) AFTER A COUNT OF 0 IS REACHED, THE PROGRAM WAITS
; * FOR A CERTAIN TIME, DURING WHICH THE SEC COUNTER
; * IS SAMPLED. IF THE COUNTER DOES NOT CHANGE WITHIN THIS
; * TIME PERIOD AN ERROR IS REPORTED.
; * 3) UPON FINDING THAT THE COUNTER HAS CHANGED, IT IS CHECKED
; * IF IT INCREMENTED CORRECTLY. IF IT DID NOT AN ERROR IS REPORTED
; * 4) IF IT INCREMENTED CORRECTLY, THE PROGRAM AGAIN WAITS IN A
; * LOOP TILL THE COUNTER CHANGES. (STEPS 2,3,4 ARE REPEATED
; * TILL THE COUNTER COUNTS UP TO 13)
*****

```

```

*****
;ST7: SCOPE
      CNT.RESET

```

```

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

```

```

MOV    DRIVAD,DRKDA
MOV    RKDS,R0
CLR    INDX1
CLR    R5
MOV    #-14,R4
MOV    #1,R3
1$:    CLR    INDX2
INC    INDX1
BEQ    6$
2$:    INC    INDX2
BEQ    7$
MOV    DR0,R1
BIT    #400,R1
BEQ    2$
CMP    DR0,R1
1$:    BNE    1$
BIC    #177760,R1
1$:    BNE    1$

```

```

;INITIALIZE
;'COUNT' - TO TIME 'ERROR 35'
;INITIALIZE 'COUNT' - TO TIME
;'ERROR 36' (WAIT LOOP)
;INITIALIZE 'COUNT' - FOR THE 12 SECTORS.
;R3 CONTAINS THE 'NEXT' COUNT OF SEC-CNTR
;R1 CONTAINS THE 'PREVIOUS' COUNT OF SEC-CNTR
;R2 CONTAINS THE 'PRESENT' COUNT OF SEC-CNTR
;INITIALIZE 'COUNT' - TO TIME
;(WAIT LOOP) 'ERROR 34'
;KEEP TIMING FOR 'ERROR 35'
;BRANCH & REPORT ERROR IF WAITED LONG
;KEEP TIMING FOR 'ERROR 34'
;BRANCH & REPORT ERROR IF WAITED LONG
;GET RKDS
;IS 'SOK' SET?
;NO, WAIT FOR IT TO SET
;MAKE SURE THAT 2 CONSECUTIVE
;READINGS OF SEC-CNTR ARE SAME
;YES, MASK OUT NON-SEC CNTR BITS
;IS IT SECTOR 0, IF NOT LOOP BACK &

```

K04

MAINDEC-11-DZRKK-C
DZRKKC.P11 T7MACY11 27(732) 16-SEP-76 16:00 PAGE 50
CHECK THAT 'SECTOR COUNTER' CAN COUNT FROM 0-13

```

2459
2460 005254 005204
2461 005256 001447
2462 005260 005205
2463 005262 001431
2464 005264 011002
2465 005266 032702 000400
2466 005272 001772
2467 005274 021002
2468 005276 001370
2469 005300 042702 177760
2470 005304 020201
2471 005306 001764
2472 005310 020203
2473 005312 001023
2474
2475 005314 005203
2476 005316 005201
2477 005320 005005
2478 005322 000754
2479
2480 005324 010137 001162
2481 005330 104012
2482
2483 005332 000421
2484
2485 005334 017737 175162 001162
2486 005342 104011
2487
2488 005344 000414
2489
2490 005346 010237 001162
2491 005352 010337 001164
2492 005356 104013
2493
2494
2495 005360 000406
2496
2497 005362 010337 001162
2498 005366 010237 001164
2499 005372 104014
2500
2501
2502 005374 000747
2503
2504
2505
2506
2507
2508
2509
2510 005376 000004
2511 005400 104412
2512
2513
2514

```

```

3$: INC R4
   BEQ TST10
4$: INC R5
   BEQ 8$
   MOV 2R0,R2
   BIT 400,R2
   BEQ 4$
   CMP 2R0,R2
   BNE 4$
   BIC 177760,R2
   CMP R2,R1
   SEQ 4$
   CMP R2,R3
   BNE 9$

5$: INC R3
   INC R1
   CLR R5
   BR 3$

6$: MOV R1,$REG0
   ERROR 12
   BR TST10

7$: MOV 2RKDS,$REG0
   ERROR 11
   BR TST10

8$: MOV R2,$REG0
   MOV R3,$REG1
   ERROR 13
   BR TST10

9$: MOV R3,$REG0
   MOV R2,$REG1
   ERROR 14
   BR 5$
;

;*****
;*TEST 10 CHECK THAT SC=SA CAN BE GENERATED
;* THIS TEST CHECKS THAT SC=SA CAN BE GENERATED FOR
;* EVERY SECTOR
;*****
TST10: SCOPE
        CNT.RESET

```

```

;WAIT FOR SECTOR 0
;KEEP TRACK OF SECTORS CHECKED
;EXIT IF ALL SECTORS CHKD
;KEEP TIMING FOR 'ERROR 36'
;BR & REPORT ERROR IF WAITED LONG
;GET RKDS
;IS SOK SET?
;NO, WAIT FOR SOK
;MAKE SURE THAT 2 CONSECUTIVE
;READINGS OF SEC-CNTR ARE SAME
;MASK NON-SEC-CNTR BITS
;HAS SEC CNTR INCREMENTED?
;NO, WAIT FOR IT TO CHANGE
;YES, DID IT INCREMENT CORRECTLY?
;NO - REPORT ERROR

;INCREMENT "NEXT COUNT"
;INCREMENT "PREVIOUS COUNT"
;INITIALIZE AGAIN FOR TIMING 'ERROR 36'
;GO & CHECK THE NEXT SECTOR COUNT

;GET 'SEC CNTR'
;WAITED LONG, BUT SECTOR COUNTER
;DID NOT COUNT TO 0
;;EXIT

;GET RKDS
;WAITED LONG, BUT 'SOK' BIT DID
;NOT SET
;;EXIT

;GET SEC CNTR (PRESENT COUNT)
;GET "NEXT COUNT"
;WAITED LONG, BUT THE SECTOR
;COUNTER DID NOT INCREMENT FROM
;THE PRESENT COUNT TO THE NEXT COUNT
;;EXIT

;GET 'NEXT COUNT' (SEC CNTR SHOULD BE THIS)
;GET PRESENT COUNT (WHAT SEC CNTR WAS)
;SEC CNTR INCREMENTED WRONG, DID
;NOT INCREMENT FROM PRESENT COUNT
;TO NEXT COUNT

```

```

;GO, DO CONTROL RESET
;THIS IS A CALL FOR THE 'CNTRL-
;RESET' ROUTINE. A CONTROL RESET IS
;ISSUED AND AFTER A CERTAIN TIME

```



```

2515                                     ;IF THE 'CNTRL RDY' DOES NOT SET
2516                                     ;AN ERROR IS REPORTED. NOTE THAT
2517                                     ;THE PC IN ERROR MESSAGE IS THE
2518                                     ;PC WHERE 'CNT.RESET' IS LOCATED.
2519                                     ;THIS IS A VERY BASIC ERR & IF IT
2520                                     ;OCCURS GO BACK TO TEST 10
2521 005402 013704 002544                 MOV    DRIVAD,R4
2522 005406 013700 002527                 MOV    RKDS,R0
2523 005412 012703 177764                 MOV    #-14,R3
2524 005416 010477 175112                 1$:    MOV    R4,DRKDA
2525 005422 005005                       CLR    R5
2526 005424 005205                       2$:    INC    R5
2527 005426 001410                       BEQ    3$
2528 005430 011001                       MOV    DR0,R1
2529 005432 032701 000020                 BIT    #20,R1
2530 005436 001772                       BEQ    2$
2531 005440 005204                       4$:    INC    R4
2532 005442 005203                       INC    R3
2533 005444 001364                       BNE    1$
2534 005446 000406                       BR     TST11
2535                                     ;YES, EXIT
2536 005450 110437 001162                 3$:    MOVB   R4,$REG0
2537 005454 010137 001164                 MOV    R1,$REG1
2538 005460 104015                       ERROR  15
2539                                     ;GET SECTOR ADDRESS
2540 005462 000766                       BR     4$
2541                                     ;GET RKDS
2542                                     ;COULD NOT GET SC=SA FOR THIS
2543                                     ;'SECTOR ADDRESS'
2544                                     ;GO CHK FOR THE REST
2545 *****
2546 ;*TEST 11 CHECK THAT 'R/W/S RDY' IS SET & 'SIN' IS CLEAR
2547 *****
2548 †TST11: SCOPE
2549 CNT.RESET
2550 MOV    DRIVAD,DRKDA
2551 1$:    MOV    DRKDS,R0
2552 BIT    #100,R0
2553 BNE    2$
2554 INC    R1
2555 BNE    1$
2556 MOV    R0,$REG0
2557 ERROR  16
2558 2$:    BIT    #1000,R0
2559 BEQ    TST12
2560 JSR    PC,GT4RG
2561 ERROR  1
2562                                     ;GO, DO CONTROL RESET
2563                                     ;ADDRESS THE DRIVE
2564                                     ;GET RKDS
2565                                     ;IS R/W/S RDY SET?
2566                                     ;YES, BRANCH
2567                                     ;WAITED LONG ENOUGH?
2568                                     ;IF NOT LUP BAK & WAIT
2569                                     ;GET RKDS
2570                                     ;R/W/S RDY SHOULD BE SET
2571                                     ;IS SIN CLEAR?
2572                                     ;YES, EXIT
2573                                     ;GET RKCS,ER,DS,DA
2574                                     ;'SIN' SHOULD HAVE BEEN CLEAR
2575                                     ;IT WAS NOT CLEAR
2576                                     ;NEXT TEST IS GOING TO CHECK
2577                                     ;DRIVE RESET, SIN SHOULD BE
2578                                     ;CLEARED THEN. IT WILL BE CHECKED
2579                                     ;THERE.
2580 *****
2581 ;*TEST 12 CHECK 'DRIVE RESET'
2582 ;*THIS TEST CHECKS THE VERY BASIC DRIVE RESET LOGIC.
2583 ;*SINCE THE HEADS ARE AT CYLINDER 0 (GOING INTO THIS
2584 ;*TEST) DRIVE RESET RETRACTS THEM BACK BEYOND CYLINDER 0.

```

MO4

```

2571                                     ;*AFTER WHICH THEY ARE PUSHED FORWARD TO CYLINDER 0 AGAIN.
2572                                     ;*IN THE LATER PART OF THIS PROGRAM THERE IS A DRIVE RESET
2573                                     ;*TEST WHICH DOES THE RESET FROM LAST CYLINDER.
2574                                     ;*****
2575 005540 000004                                TST12: SCOPE
2576 005542 104412                                CNT.RESET
2577
2578                                     ;GO, DO CONTROL RESET
2579                                     ;THIS IS A CALL FOR THE 'CNTRL-
2580                                     ;RESET' ROUTINE. A CONTROL RESET IS
2581                                     ;ISSUED AND AFTER A CERTAIN TIME
2582                                     ;IF THE 'CNTRL RDY' DOES NOT SET
2583                                     ;AN ERROR IS REPORTED. NOTE THAT
2584                                     ;THE PC IN ERROR MESSAGE IS THE
2585                                     ;PC WHERE 'CNT.RESET' IS LOCATED.
2586 005544 013700 002526                                MOV    RKCS,R0
2587 005550 005004                                CLR    R4
2588 005552 013777 002544 174754                    MOV    DRIVAD,ARKDA
2589 005560 012710 000015                                MOV    #15,R0
2590 005564 104411                                CHKCRDY
2591
2592 005566 104021                                ERROR 21
2593
2594                                     ;INITIALIZ COUNT - TO TIME ERROR
2595                                     ;ADDRESS THE DRIVE
2596 005570 032777 000100 174724 2$:                BIT    #100,ARKDS
2597 005576 001005                                BNE    3$
2598 005600 005204                                INC    R4
2599 005602 001372                                BNE    2$
2600 005604 004737 020406                                JSR    PC,GT4RG
2601 005610 104026                                ERROR 26
2602
2603                                     ;DID R/W/S RDY SET?
2604 005612 032777 001000 174702 3$:                BIT    #1000,ARKDS
2605 005620 001403                                BEQ    5$
2606 005622 004737 020406                                JSR    PC,GT4RG
2607 005626 104001                                ERROR 1
2608
2609                                     ;DID SIN SET?
2610 005630 032710 140000 5$:                        BIT    #140000,R0
2611 005634 001403                                BEQ    4$
2612 005636 004737 020406                                JSR    PC,GT4RG
2613 005642 104022                                ERROR 22
2614
2615                                     ;NO, BRANCH
2616 005644 022710 000214 4$:                        CMP    #214,R0
2617 005650 001406                                BEQ    TST13
2618 005652 012737 000214 001162                    MOV    #214,$REG0
2619 005660 011037 001164                                MOV    R0,$REG1
2620 005664 104024                                ERROR 24
2621
2622                                     ;*****
2623                                     ;*TEST 13 CHECK 'SEEK' TO CYLINDER 0
2624                                     ;* THIS TEST CHECKS THE SEEK LOGIC DOING SEEK TO CYLINDER 0.
2625                                     ;* NOTE THAT SINCE THE HEADS ARE ALREADY ON CYLINDER 0, NO
2626                                     ;* HEAD MOVEMENT IS INVOLVED AND THE STRESS IS ON THE BASIC SEEK

```

```

2627                                     ;* LOGIC.
2628                                     ;*****
2629 005566 000004 1ST13: SCOPE
2630 005670 104412 CNT.RESET
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640 005672 104420 TST.SIN
2641
2642 005674 013700 002526 MOV RKCS,RO
2643 005700 013777 002544 174626 MOV DRIVAD,DRKDA
2644
2645 005706 012710 000011 MOV #11,DR0
2646 005712 104411 CHKCRDY
2647
2648 005714 104021 ERROR 21
2649
2650
2651
2652 005716 005005 2$: CLR RS
2653 005720 032777 000100 174574 BIT #100,DRKDS
2654 005726 001005 BNE 3$
2655 005730 005205 INC RS
2656 005732 001372 BNE 2$+2
2657 005734 004737 020406 JSR PC,GT4RG
2658 005740 104026 ERROR 26
2659 005742 032777 001000 174552 3$: BIT #1000,DRKDS
2660 005750 001403 BEQ 6$
2661 005752 004737 020406 JSR PC,GT4RG
2662 005756 104001 ERROR 1
2663
2664
2665
2666
2667 005760 032710 140000 6$: BIT #140000,DR0
2668 005764 001403 BEQ 4$
2669
2670 005766 004737 020406 JSR PC,GT4RG
2671 005772 104022 ERROR 22
2672
2673 005774 005777 174524 4$: TST DRKER
2674 006000 001403 BEQ 5$
2675 006002 004737 020414 JSR PC,GT3RG
2676 006006 104023 ERROR 23
2677
2678 006010 022710 000210 5$: CMP #210,DR0
2679 006014 001406 BEQ TST14
2680 006016 012737 000210 001162 MOV #210,$REG0
2681 006024 011037 001164 MOV DR0,$REG1
2682 006030 104024 ERROR 24

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

;ADDRESS THE DRIVE

;'SEEK' GO
;GO CHECK IF CONTROL RDY IS SET
;IF SO, SKIP THE EROR MESSAGE.
;'CNTRL RDY' DID NOT SET AFTER SENDING
;CYL ADDR TO THE DRIVE, 'ADD ACK'
;SHOULD HAVE COME BACK FROM THE
;DRIVE, THEREUPON SETTING 'CNTRL RDY'

;DID R/W/S RDY BIT SET?
;YES, BRANCH
;WAITED LONG ENOUGH?
;IF NOT, LUP BAK & WAIT
;GO, GET RKCS, ER, DS, DA
;R/W/S RDY DID NOT SET AFTER SEEK
;DID SIN SET?
;NO, BRANCH
;GO, GET RKCS, ER, DS, DA
;SIN SET ON DOING SEEK
;TO CYL 0 NOTE THIS IS THE
;FIRST TIME THE HEADS HAVE
;BEEN MOVED

;WAS 'ERR' OR 'HE' BIT SET?

;GO, GET RKCS, ER, DS, DA
;'ERR' OR 'HE' BIT SET WHILE DOING 'SEEK'

;WAS ANY BIT IN RKER SET?
;NO
;GO, GET RKCS, ER, DS
;RKER SHOWS AN ERROR BIT, CHECK

;DOES RKCS STILL CONTAIN 'SEEK' FUNCTION
;;YES, EXIT
;GET EXPCTD RKCS
;GET RKCS RECVD
;NO, RKCS SHOULD BE STILL CONTAINING

```

MAINDEC-11-DZRAK-C
DZRAK.C.P11 T13

MACY11.27(732) 16-SEP-76 16:00 PAGE 54
CHECK 'SEEK' TO CYLINDER 0

: 'SEEK' FUNCTION ERROR - IF IT CHANGED

2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738

006032 000004
006034 104412

006036 104420

006040 004737 021116
006044 104026

006046 005005
006050 013777 002544 174456
006056 052777 000100 174450
006064 013701 002522
006070 012777 000011 174430
006076 032711 000100

006102 001405
006104 005205
006106 100373
006110 004737 020414
006114 104025

006116 004737 021050
006122 104016

: *TEST 14 CHECK R/W/S RDY IS CLEAR WHEN HEADS ARE IN MOTION
: *THIS TEST CHECKS THAT R/W/S DOES GET CLEARED
: *WHEN THE HEADS ARE IN MOTION. SINCE 'MOVE L' ON
: *M7700 (RK05) GENERATES THIS SIGNAL. ABSENCE OF
: *R/W/S RDY-CLEAR COULD MEAN A FAULT ON M7702
: *WHERE 'MOVE L' IS GENERATED.
: *NOTE THIS IS THE FIRST TIME HEADS ARE MADE TO MOVE BY SEEKING
: *TO CYLINDER 2.

ST14: SCOPE
CNT.RESET

: GO, DO CONTROL RESET
: THIS IS A CALL FOR THE 'CNTRL-
: RESET' ROUTINE. A CONTROL RESET IS
: ISSUED AND AFTER A CERTAIN TIME
: IF THE 'CNTRL RDY' DOES NOT SET
: AN ERROR IS REPORTED. NOTE THAT
: THE PC IN ERROR MESSAGE IS THE
: PC WHERE 'CNT.RESET' IS LOCATED.
: THIS IS A VERY BASIC ERR & IF IT
: OCCURS GO BACK TO TEST 10
: GO CHECK IF SIN IS SET
: IF SET DO DRV-RESET TO CLR IT
: MAKE SURE HEADS R ON CYL 0
: R/W/S RDY DIDN'T SET
: AFTER THE ABOVE DRV RESET

TST.SIN
JSR PC,DRESET
ERROR 26
CLR R5
MOV DRIVAD,ARKDA
BIS #100,ARKDA
MOV ARKDS,R1
MOV #11,ARKCS
BIT #100,ARI
BEQ 25
INC R5
BPL 15
JSR PC,GT3RG
ERROR 25

: SEEK CYLINDER 2

: SEEK GO
: DID R/W/S RDY CLR?
: YES. BRANCH

: R/W/S RDY WAS NOT CLEAR WHEN HEADS
: WERE SEEKING TO CYLINDER 2

25: JSR PC,TSTRWS
ERROR 16

: GO, WAIT FOR R/W/S RDY TO SET
: R/W/S RDY DID NOT SET AFTER SEEK
: WAS TRIED TO CYLINDER 2 (ABOVE).
: NOTE THIS WAS THE FIRST TIME A SEEK
: WAS TRIED TO A CYLINDER OTHER THAN
: 0.

: *TEST 15 CHECK 'WRITE' FORMAT FUNCTION-CYLINDER 0, SECTOR C
: *THIS TEST CHECKS THE LOGIC INVOLVED IN THE WRITE FMT
: *FUNCTION. ON ISSUING A WRT FMT, THE FOLLOWING IS CHECKED
: *1) CNTRL RDY WAS CLEARED AS GO WAS SET.

C05

 MAINDEC-11-DZKCC-C
 DZKCC.P11 T15

 MACY11 27(732) 16-SEP-76 16:00 PAGE 55
 CHECK 'WRITE' FORMAT FUNCTION-CYLINDER 0, SECTOR 0

```

2739      ;*2) CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION OF FUNCTION
2740      ;*3) IF 'HE' OR 'ERR' BIT SET?
2741      ;*4) IF RKDA INCREMENTED CORRECTLY FROM 0 TO 1?
2742      ;*5) IF RKWC OVERFLOWED CORRECTLY TO 0?
2743      ;*6) IF RKBA INCREMENTED CORRECTLY BY 2?
2744      ;*7) IF ANY BIT IN RKER SET?
2745      ;*8) IF THE 'WRT FMT' FUNCTION BITS ARE STILL IN THE RKCS?
2746      ;*NOTE THAT ONE WORD '125252' WAS WRITTEN ON SECTOR
2747      ;*0 & IT WILL BE CHECKED IN THE NEXT TESTS.
2748      ;*****
2749      006124 000004
2750      006126 104412
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760      006130 104420
2761
2762      006132 012703 032724
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776      006136 012700 000001
2777
2778      006142 010023
2779      006144 010013
2780      006146 005423
2781      006150 005200
2782      006152 022700 000200
2783      006156 001371
2784      006160 005023
2785      006162 012713 125252
2786
2787      006166 012703 032724
2788      006172 013701 002526
2789      006176 013702 002532
2790      006202 010312
2791      006204 012777 177400 174316
2792      006212 013777 002544 174314
2793      006220 012711 002003
2794
  
```

```

          TST15: SCOPE
                  CNT.RESET
;GO, DO CONTROL RESET
;THIS IS A CALL FOR THE 'CNTRL-
;RESET' ROUTINE. A CONTROL RESET IS
;ISSUED AND AFTER A CERTAIN TIME
;IF THE 'CNTRL RDY' DOES NOT SET
;AN ERROR IS REPORTED. NOTE THAT
;THE PC IN ERROR MESSAGE IS THE
;PC WHERE 'CNT.RESET' IS LOCATED.
;THIS IS A VERY BASIC ERR & IF IT
;OCCURS GO BACK TO TEST 10
;GO CHECK IF SIN IS SET
;IF SET, DO DRIVE RESET TO CLR IT

;THIS CODE SETS UP A 256 WORD BUFFER
;WHICH WILL BE USED TO WRITE 1 SECTOR
;ON THE DISK
;1ST WORD      000001
;2ND WORD      177777 2'S COMPLEMENT
;3RD WORD      000002 OF ABOVE
;4TH WORD      177776

;253RD WORD    000177
;254TH WORD    177601
;255TH WORD    000000
;256TH WORD    125252

          MOV     #1,R0
;SET COUNT

          9$:    MOV     R0,(R3)+
          MOV     R0,(R3)+
          NEG     (R3)+
          INC     R0
          CMP     #200,R0
          BNE     9$
;DONE?

          CLR     (R3)+
          MOV     #125252,R3
;SET 255TH WORD TO 0
;SET 256TH WORD

          MOV     #OUTBUF,R3
;RESET POINTER TO OUTBUF

          MOV     RKCS,R1
          MOV     RKBA,R2
          MOV     R3,R2
;FROM HERE-SET UP CURRENT ADDRESS
          MOV     #-400,RKWC
;SET UP WORD COUNT 400 WORDS
          MOV     DRIVAD,RKDA
;SET UP DISK ADDRS, SECTOR 0, CYLINDER 0
          MOV     #2003,R1
;WRITE FORMAT, GO
  
```

005

MAINDEC-11-DZRKK-C
DZRKKC.P11 T15

MACY11 27(732) 16-SEP-76 14:00 PAGE 56
CHECK 'WRITE' FORMAT FUNCTION-CYLINDER 0, SECTOR 0

2795	006224	105711		1\$:	TSTB	DR1		: WAS 'CNTRL RDY' CLEARED AS GO WAS SET?
2796	006226	100003			BPL	2\$		: YES, BRANCH
2797	006230	004737	020414		JSR	PC,GT3RG		: GO, GET RKCS, ER, DS
2798	006234	104030			ERROR	30		: 'CNTRL RDY' DIDN'T CLEAR AS GO
2799								: WAS SET TO 'WRITE FORMAT'
2800	006236	005000		2\$:	CLR	RO		: WAS 'CNTRL RDY' SET ON COMPLETION OF WRITE?
2801	006240	105711			TSTB	DR1		: YES, BRANCH
2802	006242	100411			BMI	3\$		: NO, HAVE U WAITED LONG ENOUGH?
2803	006244	005200			INC	RO		: IF NOT, LOOP BACK & WAIT
2804	006246	001374			BNE	2\$+2		: IF YES, REPORT ERROR
2805								: GO, GET RKCS, ER, DS, DA
2806	006250	004737	020406		JSR	PC,GT4RG		
2807	006254	013737	002544	001202	MOV	DR1VAD,\$REG10		
2808	006262	104415			BRKDA4			: GO TO 'BD44' & BREAK CONTENTS OF
2809								: \$REG10 INTO DR 8, CYL SUR, SEC BITS
2810	006264	104031			ERROR	31		: 'CNTRL RDY' DIDN'T SET ON COMPLETION
2811								: OF WRITE FORMAT
2812								: WRT FMT WAS DONE STARTING AT (DSK-ADRES)
2813								: INDICATED IN EROR MSGE.
2814	006266	004737	020646	3\$:	JSR	PC,CHKHE		: GO CHECK IF 'HE' OR 'ERR' BIT SET.
2815								: IF YES, SAVE RKCS, ER, DS, DA.
2816								: RETURN HERE IF ERROR.
2817	006272	104032			ERROR	32		: 'HE' OR 'ERR' BIT SET WHILE DOING
2818								: A WRITE FORMAT
2819								: WRT FMT WAS DONE STARTING AT (DSK-ADRES)
2820								: INDICATED IN EROR MSGE.
2821	006274	004737	020674	4\$:	JSR	PC,CHKDA		: GO CHECK IF RKDA INCREMENTED CORRECTLY
2822								: IF NOT, RETURN HERE.
2823	006300	104033			ERROR	33		: RKDA SHOULD HAVE INCREMENTED BY
2824								: 1 SECTOR, IT DID NOT
2825	006302	004737	020730	5\$:	JSR	PC,CHKWC		: CHECK IF WORD COUNT OVERFLOWED, IF
2826								: NOT RETURN HERE.
2827	006306	104034			ERROR	34		: RKWC DID NOT OVERFLOW TO 0, AFTER
2828								: XFER ON WRITE FORMAT
2829	006310	022712	033724	6\$:	CMP	#OUTBUF+1000,DR2		: DID RKBA INCREMENT CORRECTLY?
2830	006314	001406			BEQ	7\$		: YES, BRANCH
2831	006316	012737	033724	001162	MOV	#OUTBUF+1000,\$REG0		: GET EXPCTD RKBA
2832	006324	011237	001164		MOV	DR2,\$REG1		: GET ACTUAL RKBA
2833	006330	104035			ERROR	35		: RKBA DIDN'T INCREMENT BY 1000 AFTER
2834								: WRITE FORMAT OF 400 WORDS
2835	006332	004737	020754	7\$:	JSR	PC,CHKER		: CHECK IF ANY BIT IN RKER SET.
2836								: IF YES RETURN HERE.
2837	006336	104036			ERROR	36		: RKER BIT SET ON DOING 1 WORD
2838								: WRITE FORMAT
2839	006340	022711	002202	8\$:	CMP	#2202,DR1		: DOES RKCS STILL HAVE 'WRT FMT' BITS?
2840	006344	001406			BEQ	TST16		: YES, EXIT
2841	006346	012737	002202	001162	MOV	#2202,\$REG0		: GET EXPCTD RKCS
2842	006354	011137	001164		MOV	DR1,\$REG1		: GET ACTUAL RKCS
2843	006360	104024			ERROR	24		: RKCS DIDN'T CONTAIN 'WRT FMT' BITS
2844								: AFTER THE FUNCTION WAS COMPLETED

; *TEST 16 CHECK 'READ FORMAT' FUNCTION-CYLINDER 0, SECTOR 0
; *THIS TEST CHECKS THE LOGIC INVOLVED IN THE WRITE FMT
; *FUNCTION. ON ISSUING A WRT FMT, THE FOLLOWING IS CHECKED
; *1) CNTRL RDY WAS CLEARED AS GO WAS SET.

2845
2846
2847
2848
2849
2850

E05

MAINDEC-11-DZRKK-C
DZRKKC.P11 T16MACY11 27(732) 16-SEP-76 16:00 PAGE 57
CHECK 'READ FORMAT' FUNCTION-CYLINDER 0, SECTOR 0

```

2851      ;*2) CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION OF FUNCTION
2852      ;*3) IF 'HE' OR 'ERR' BIT SET?
2853      ;*4) IF RKDA INCREMENTED CORRECTLY FROM 0 TO 1?
2854      ;*5) IF RKWC OVERFLOWED CORRECTLY TO 0?
2855      ;*6) IF RKBA INCREMENTED CORRECTLY BY 2?
2856      ;*7) IF ANY BIT IN RKER SET?
2857      ;*8) IF THE CORRECT HEADER WAS RECEIVED?
2858      ;*9) FOR RK11C, AFTER RD FMT RKDB CONTAINS THE CHECKSUM
2859      ;*FOR THAT SECTOR. (125252 IN THIS CASE, BECAUSE THE
2860      ;*FIRST WORD IN SEC 0 WAS WRITTEN AS 125252 IN
2861      ;*THE PREVIOUS TEST)
2862      ;*10) FOR RK11D, AFTER RD FMT RKDB SHOULD CONTAIN
2863      ;*A ZERO
2864      ;*11) IF THE RD FMT FUNCTION BITS ARE STILL IN
2865      ;*THE RKCS?
2866      ;*****
2867      006362 000004
2868      006364 005000
2869      006366 104412
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879      006370 104420
2880
2881      006372 013701 002526
2882      006376 013702 002532
2883      006402 012703 032724
2884      006406 010312
2885
2886      006410 012777 177777 174112
2887      006416 013777 002544 174110
2888      006424 012711 002005
2889
2890      006430 105711
2891      006432 100003
2892      006434 004737 020414
2893      006440 104030
2894
2895      006442 005000
2896      006444 105711
2897
2898      006446 100411
2899      006450 005200
2900      006452 001374
2901
2902      006454 004737 020406
2903      006460 013737 002544 001202
2904      006466 104415
2905
2906      006470 104045

```

```

TST16:  SCOPE
        CLR    RO
        CNT.RESET

TST.SIN

        MOV    RKCS,R1
        MOV    RKBA,R2
        MOV    #OUTBUF,R3
        MOV    R3,R2
        MOV    #-1,RKWC
        MOV    DRIVAD,RKDA
        MOV    #2005,R1

1$:     TSTB    R1
        BPL     2$
        JSR     PC,GT3RG
        ERROR   30

2$:     CLR     RO
        TSTB    R1
        BMI     3$
        INC     RO
        BNE     2$+2
        JSR     PC,GT4RG
        MOV     DRIVAD,$REGIO
        BRKDA4

        ERROR   45

```

```

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

;SETUP ADRS WHERE HEADER WORD IS TO BE
;ERRED
;SET UP WORD COUNT
;SET UP DISK ADRS, SECTOR 0, CYLINDER 0
;READ FORMAT, GO

;WAS 'CNTRL RDY' CLEARED AS GO WAS SET?
;YES, BRANCH
;GO, GET RKCS, RKER
;CNTRL RDY DIDN'T CLEAR AS GO WAS
;SET TO 'READ FORMAT'

;WAS 'CNTRL RDY' SET ON COMPLETION OF
;TRANSFER
;YES, BRANCH
;NO, HAVE U WAITED LONG ENOUGH?
;IF NOT, LOOP BACK & WAIT
;IF YES, REPORT ERROR
;GO, GET RKCS, ER, DS,DA

;GO TO 'BD44' & BREAK CONTENTS OF
;$REGIO INTO DR #,CYL,SUR,SEC BITS
;'CNTRL RDY' DIDN'T SET ON COMPLETION

```

```

;*****
;*TEST 17      CHECK 'READ' FUNCTION-CYLINDER 0,SECTOR 0
;*THIS IS THE FIRST TIME A PURE READ IS PERFORMED IN THIS
;*TEST SEQUENCE.  THE FOLLOWING IS CHECKED
;*1) CNTRL RDY CLEARS AS GO IS SET
;*2) CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
;*OF FUNCTION
;*3) IF 'HE' OR 'ERR' BIT SET?
;*4) IF RKDA INCREMENTED CORRECTLY?
;*5) IF RKWC OVERFLOWED TO 0?

```


Address	Offset	Value	Label	Operation	Comments
2963					*6) IF RKBA INCREMENTED CORRECTLY?
2964					*7) IF ANY RKER BIT SET?
2965					*8) IF THE CORRECT PSEUDO-HEADER (FIRST WORD) WAS
2966					*READ FROM SECTOR 0
2967					*9) IF THE 'READ' FUNCTION BITS ARE STILL IN RKCS
2968					*****
2969	006610	000004	TST17:	SCOPE	
2970	006612	104412		CNT.RESET	GO, DO CONTROL RESET
2971					THIS IS A CALL FOR THE 'CNTRL-
2972					RESET' ROUTINE. A CONTROL RESET IS
2973					ISSUED AND AFTER A CERTAIN TIME
2974					IF THE 'CNTRL RDY' DOES NOT SET
2975					AN ERROR IS REPORTED. NOTE THAT
2976					THE PC IN ERROR MESSAGE IS THE
2977					PC WHERE 'CNT.RESET' IS LOCATED.
2978					THIS IS A VERY BASIC ERR & IF IT
2979					OCCURS GO BACK TO TEST 10
2980	006614	104420		TST.SIN	GO CHECK IF SIN IS SET
2981					IF SET, DO DRIVE RESET TO CLR IT
2982	006616	013701		MOV RKCS,R1	
2983	006622	005000		CLR R0	
2984	006624	013702		MOV RKBA,R2	
2985	006630	012703		MOV #OUTBUF,R3	
2986	006634	010312		MOV R3,R2	SET UP ADDRS WHERE DATA WORD IS
2987					TO BE X-FERRED
2988	006636	012777	177400	MOV #400,RKWC	SET UP WORD COUNT
2989	006644	013777	002544	MOV DRIVAD,RKDA	SET UP DISK ADRS, SECTOR 0, CYLINDER 0
2990	006652	012711	000005	MOV #5,R1	READ, GO
2991					
2992	006656	105711		1\$: TSTB R1	WAS 'CNTRL RDY' CLEARED AS GO WAS SET?
2993	006660	100003		BPL 2\$	YES, BRANCH
2994	006662	004737	020414	JSR PC,GT3RG	GO, GET RKCS, ER
2995	006666	104030		ERROR 30	CNTRL RDY DID NOT CLEAR AS GO
2996					WAS SET TO 'READ'
2997	006670	005000		2\$: CLR R0	
2998	006672	105711		TSTB R1	WAS CNTRL RDY SET ON COMPLETION
2999					OF TRANSFER?
3000	006674	100411		BMI 3\$	YES, BRANCH
3001	006676	005200		INC R0	NO, HAVE U WAITED LONG ENOUGH?
3002	006700	001374		BNE 2\$+2	IF NOT, LOOP BACK & WAIT
3003					IF YES, REPORT ERROR
3004	006702	004737	020406	JSR PC,GT4RG	GO, GET RKCS, ER, DS,DA
3005	006706	013737	002544	MOV DRIVAD,\$REG10	
3006	006714	104415		BRKDA4	GO TO 'BDA4' & BREAK CONTENTS OF
3007					\$REG10 INTO DR #CYL SUR,SEC BITS
3008	006716	104045		ERROR 45	CNTRL RDY DID NOT SET ON
3009					COMPLETION OF READ
3010					READ WAS DONE STARTING AT <DSK-ADRES>
3011					INDICATED IN EROR MESGE
3012					
3013	006720	004737	020646	3\$: JSR PC,CHKHE	CHECK IF 'ERR' OR 'HE' BIT IS SET
3014					IF YES, RETURN HERE.
3015	006724	104046		ERROR 46	'HE' OR 'ERR' BIT SET WHILE
3016					DOING A READ.
3017					READ WAS DONE STARTING AT <DSK-ADRES>
3018					INDICATED IN EROR MESGE

Address	Op Code	Op 1	Op 2	Op 3	Op 4	Op 5	Op 6	Op 7	Op 8	Op 9	Op 10	Op 11	Op 12	Op 13	Op 14	Op 15	Op 16	Op 17	Op 18	Op 19	Op 20	Op 21	Op 22	Op 23	Op 24	Op 25	Op 26	Op 27	Op 28	Op 29	Op 30	Op 31	Op 32	Op 33	Op 34	Op 35	Op 36	Op 37	Op 38	Op 39	Op 40	Op 41	Op 42	Op 43	Op 44	Op 45	Op 46	Op 47	Op 48	Op 49	Op 50	Op 51	Op 52	Op 53	Op 54	Op 55	Op 56	Op 57	Op 58	Op 59	Op 60	Op 61	Op 62	Op 63	Op 64	Op 65	Op 66	Op 67	Op 68	Op 69	Op 70	Op 71	Op 72	Op 73	Op 74	Op 75	Op 76	Op 77	Op 78	Op 79	Op 80	Op 81	Op 82	Op 83	Op 84	Op 85	Op 86	Op 87	Op 88	Op 89	Op 90	Op 91	Op 92	Op 93	Op 94	Op 95	Op 96	Op 97	Op 98	Op 99	Op 100	Op 101	Op 102	Op 103	Op 104	Op 105	Op 106	Op 107	Op 108	Op 109	Op 110	Op 111	Op 112	Op 113	Op 114	Op 115	Op 116	Op 117	Op 118	Op 119	Op 120	Op 121	Op 122	Op 123	Op 124	Op 125	Op 126	Op 127	Op 128	Op 129	Op 130	Op 131	Op 132	Op 133	Op 134	Op 135	Op 136	Op 137	Op 138	Op 139	Op 140	Op 141	Op 142	Op 143	Op 144	Op 145	Op 146	Op 147	Op 148	Op 149	Op 150	Op 151	Op 152	Op 153	Op 154	Op 155	Op 156	Op 157	Op 158	Op 159	Op 160	Op 161	Op 162	Op 163	Op 164	Op 165	Op 166	Op 167	Op 168	Op 169	Op 170	Op 171	Op 172	Op 173	Op 174	Op 175	Op 176	Op 177	Op 178	Op 179	Op 180	Op 181	Op 182	Op 183	Op 184	Op 185	Op 186	Op 187	Op 188	Op 189	Op 190	Op 191	Op 192	Op 193	Op 194	Op 195	Op 196	Op 197	Op 198	Op 199	Op 200	Op 201	Op 202	Op 203	Op 204	Op 205	Op 206	Op 207	Op 208	Op 209	Op 210	Op 211	Op 212	Op 213	Op 214	Op 215	Op 216	Op 217	Op 218	Op 219	Op 220	Op 221	Op 222	Op 223	Op 224	Op 225	Op 226	Op 227	Op 228	Op 229	Op 230	Op 231	Op 232	Op 233	Op 234	Op 235	Op 236	Op 237	Op 238	Op 239	Op 240	Op 241	Op 242	Op 243	Op 244	Op 245	Op 246	Op 247	Op 248	Op 249	Op 250	Op 251	Op 252	Op 253	Op 254	Op 255	Op 256	Op 257	Op 258	Op 259	Op 260	Op 261	Op 262	Op 263	Op 264	Op 265	Op 266	Op 267	Op 268	Op 269	Op 270	Op 271	Op 272	Op 273	Op 274	Op 275	Op 276	Op 277	Op 278	Op 279	Op 280	Op 281	Op 282	Op 283	Op 284	Op 285	Op 286	Op 287	Op 288	Op 289	Op 290	Op 291	Op 292	Op 293	Op 294	Op 295	Op 296	Op 297	Op 298	Op 299	Op 300	Op 301	Op 302	Op 303	Op 304	Op 305	Op 306	Op 307	Op 308	Op 309	Op 310	Op 311	Op 312	Op 313	Op 314	Op 315	Op 316	Op 317	Op 318	Op 319	Op 320	Op 321	Op 322	Op 323	Op 324	Op 325	Op 326	Op 327	Op 328	Op 329	Op 330	Op 331	Op 332	Op 333	Op 334	Op 335	Op 336	Op 337	Op 338	Op 339	Op 340	Op 341	Op 342	Op 343	Op 344	Op 345	Op 346	Op 347	Op 348	Op 349	Op 350	Op 351	Op 352	Op 353	Op 354	Op 355	Op 356	Op 357	Op 358	Op 359	Op 360	Op 361	Op 362	Op 363	Op 364	Op 365	Op 366	Op 367	Op 368	Op 369	Op 370	Op 371	Op 372	Op 373	Op 374	Op 375	Op 376	Op 377	Op 378	Op 379	Op 380</
---------	---------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	----------

MAINDEC-11-DZRKK-C
DZRKKC.P11 T17

MACY11 27(732) 16-SEP-76 16:00 PAGE 61
CHECK 'READ' FUNCTION-CYLINDER 0, SECTOR 0

```

3075 007102 013737 002536 001164      MOV    RKDB,$REG1      ;GET ADRES OF RKDB
3076 007110 017737 173422 001164      MOV    @RKDB,$REG1    ;GET CONTENTS OF RKDB
3077 007116 104102      ERROR    102      ;CONTROL RESET DIDN'T CLR RKDB
3078
3079      ;*****
3080      ;*TEST 20      CHECK 'WRITE FORMAT' -CYLINDER 0, SECTOR 0-13
3081      ;*THIS TEST GOES ONE STEP FURTHER & PERFORMS A WRT
3082      ;*FMT ON CYLINDER 0 & CHECKS THE FOLLOWING
3083      ;*1) IF CNTRL RDY SET WITHIN A CERTAIN TIME ON COMPLETION
3084      ;*OF THE FUNCTION
3085      ;*2) IF 'HE' OR 'ERR' BIT SET?
3086      ;*3) IF THE RKDA INCREMENTS CORRECTLY?
3087      ;*4) IF THE RKDB IS CLEAR?
3088      ;*WRT FMT IS DONE ONE SECTOR AT A TIME
3089      ;*THE FIRST WORD OF EVERY SECTOR IS WRITTEN AS A
3090      ;*PSUEDO-HEADER CONSISTING OF DRIVE #, CYLINDER #, SURFACE
3091      ;*8 SECTOR #. THIS WILL BE READ & CHECKED IN THE FOLLOWING TEST.
3092      ;*****
3093 007120 000004      TST20:  SCOPE
3094 007122 013703 002526      MOV    RKCS,R3
3095 007126 012702 177764      MOV    #-14,R2      ;SET UP COUNT FOR 12 SECTORS
3096 007132 013704 002534      MOV    RKDA,R4
3097 007136 013701 002544      MOV    DRIVAD,R1    ;GET DRIVE ADDRESS
3098 007142 010105      MOV    R1,R5      ;STORE IT
3099 007144 005205      INC     R5
3100 007146 012737 007154 001110      MOV    #1$, $LPERR    ;SET RETURN ADRES FOR LUPING
3101      ;ON      ERROR (SW 9)
3102 007154 104412      1$:      CNT.RESET      ;GO, DO CONTROL RESET
3103      ;THIS IS A CALL FOR THE 'CNTRL-
3104      ;RESET' ROUTINE. A CONTROL RESET IS
3105      ;ISSUED AND AFTER A CERTAIN TIME
3106      ;IF THE 'CNTRL RDY' DOES NOT SET
3107      ;AN ERROR IS REPORTED. NOTE THAT
3108      ;THE PC IN ERROR MESSAGE IS THE
3109      ;PC WHERE 'CNT.RESET' IS LOCATED.
3110      ;THIS IS A VERY BASIC ERR & IF IT
3111      ;OCCURS GO BACK TO TEST 10
3112 007156 104420      TST.SIN      ;GO CHECK IF SIN IS SET
3113      ;IF SET, DO DRIVE RESET TO CLR IT
3114 007160 005000      CLR     RO
3115 007162 010137 032724      MOV    R1,OUTBUF    ;THIS WORD TO BE X-FERRED. FIRST
3116      ;WORD OF EACH SECTOR WILL BE THE
3117      ;ACTUAL DRIVE-ADDRS CONSISTING OF
3118      ;DRIVE NO, CYL ADDR, SURFACE
3119      ;SECTOR NO.
3120 007166 012777 032724 173336      MOV    @OUTBUF,@RKBA    ;ADRS FROM WHICH DATA WORD IS TO
3121      ;X-FERRED
3122 007174 012777 177777 173326      MOV    #-1,@RKWC      ;SET UP WORD COUNT
3123 007202 010114      MOV    R1,@R4      ;ADRS THE DRIVE, CYL 0, & CORRECT SECTOR
3124 007204 012713 002003      MOV    #2003,@R3    ;WRITE FORMAT, GO
3125
3126 007210 105777 173312      2$:      TSTB    @RKCS      ;DID 'CNTRL RDY' SET?
3127 007214 100410      BMI     3$      ;YES, BRANCH
3128 007216 005200      INC     RO      ;NO, HAVE U WAITED LONG?
3129 007220 001373      BNE     2$      ;IF NOT, LOOP BACK & WAIT
3130      ;IF YES, REPORT ERROR

```

MAINDEC-11-DZRKK-C
DZRKKC.P11 T20

MACY11 27(732) 16-SEP-76 16:00 PAGE 62
CHECK 'WRITE FORMAT' -CYLINDER 0, SECTOR 0-13

3131	007222	004737	020406		JSR	PC,GT4RG		:GO GET RKCS, ER, DS, DA
3132	007226	010137	001202		MOV	R1,\$REG10		:GET DISK ADRES (UNIT,CYL,SUR,SEC) TO WHICH
3133								:WRITE FORMAT WAS DONE
3134	007232	104415			BRKDA4			:GO TO 'BDA4' & BREAK CONTENTS OF
3135								:\$REG10 INTO DR #,CYL,SUR,SEC BITS
3136	007234	104031			ERROR	31		: 'CNTRL RDY' DID NOT SET ON COMPLETION
3137								: OF 'WRITE FORMAT'
3138								: WRT FMT WAS DONE STARTING AT <DSK-ADRES>
3139								: INDICATED IN EROR MSGE.
3140	007236	004737	020640	3\$:	JSR	PC,CHKHE1		:CHECK IF 'ERR' OR 'HE' BIT IS SET,
3141								: IF YES RETURN HERE.
3142	007242	104032			ERROR	32		: 'HE' OR 'ERR' BIT SET WHILE DOING
3143								: WRITE FORMAT ON CYLINDER 0,
3144								: SECTOR IN ERROR IS AS SHOWN IN
3145								: DISK-ADRES BITS 0-3
3146								: WRT FMT WAS DONE STARTING AT <DSK-ADRES>
3147								: INDICATED IN EROR MSGE.
3148								
3149	007244	004737	020702	4\$:	JSR	PC,CHKDA1		:CHECK IF RKDA INCREMENTED CORRECTLY?
3150								
3151	007250	104033			ERROR	33		:RKDA DID NOT INCREMENT CORRECT
3152								: AFTER 1 WORD 'WRITE FORMAT' ON
3153								: CYLINDER 0, SECTOR IN ERROR IS 1
3154								: LESS THAN THAT SHOWN IN EXPCTD RKDA
3155	007252	005777	173260	5\$:	TST	\$RKDB		:CHECK THAT RKDB DOES CONTAIN A 0
3156								: AFTER WRT BECAUSE LAST WORD WRITTEN
3157								: WAS SERIALLY SHIFTED OUT TO THE DISK
3158	007256	001406			BEQ	6\$		: YES, BRANCH
3159	007260	005037	001162		CLR	\$REG0		: THIS IS WHAT RKDB SHOULD CONTAIN
3160	007264	017737	173246	001164	MOV	\$RKDB,\$REG1		: GET RKDB
3161	007272	104037			ERROR	37		: RKDB SHOULD BE 0 AFTER WRT SINCE THE
3162								: LAST WORD WRITTEN WAS SERIALLY SHIFTED
3163								: OUT OF RKDB
3164	007274	005201		6\$:	INC	R1		: INCREMENT DRIVE ADRES TO NXT SECTOR
3165	007276	005205			INC	R5		
3166	007300	122705	000014		CMPB	#14,R5		: R U GOING TO CHECK THE LAST SECTOR?
3167	007304	001002			BNE	+.6		: IF NOT, BRANCH
3168	007306	062705	000004		ADD	#4,R5		: IF YES, INCREMENT R5 CORRECTLY TO 'EXPCTD RKDA'
3169								: AFTER HAVING CHECKED THE LAST SECTOR
3170	007312	005202			INC	R2		: HAVE U FORMATTED ALL 12 SECTORS?
3171	007314	001317			BNE	1\$		: IF NOT, BRANCH BACK & LOOP
3172								: IF YES, EXIT
3173								
3174								
3175								
3176								
3177								
3178								
3179								
3180								
3181								
3182								
3183								
3184								
3185								
3186								

: *TEST 21 CHECK 'READ FORMAT'-CYLINDER 0, SECTOR 0-13
: *THIS TEST PERFORMS A RD FMT ON THE 12 SECTORS OF CYLINDER 0
: *THE FOLLOWING IS CHECKED
: *1) IF CNTRL RDY SET WITHIN A CERTAIN TIME ON COMPLETION
: *OF THE FUNCTION
: *2) IF 'HE' OR 'ERR' BIT SET?
: *3) IF THE RKDA INCREMENTS CORRECTLY?
: *4) RKDA INCREMENTED CORRECTLY BY 30 (OCTAL)
: *5) RKWC OVERFLOWED TO 0 FROM -14 (OCTAL)
: *6) CORRECT HEADER WAS RECEIVED FROM ALL 12 SECTORS.
: *7) RKCS STILL CONTAINS THE 'RD FMT' FUNCTION BITS.
: *IF THERE IS A READ ERROR IN THIS TEST OR ANY

K05

MAINDEC-11-DZRKK-C
DZRKKC.P11 T21MACY11 27(732) 16-SEP-76 16:00 PAGE 63
CHECK 'READ FORMAT'-CYLINDER 0, SECTOR 0-13

```

3187 ;*OTHER TESTS THE USER SHOULD MAKE SURE THAT
3188 ;*IT IS AN IRRECOVERABLE ERROR AND NOT A TRANSIENT
3189 ;*ONE. THIS CAN BE DONE BY LOOPING ON THE TEST
3190 ;*IN QUESTION. USUALLY A TRANSIENT ERROR
3191 ;*DISAPPEARS ON RETRIES, WHEREAS A LOGIC ERROR DOES NOT.
3192 ;*****
3193 007316 000004 TST21: SCOPE
3194 007320 005005 CLR R5
3195 007322 104412 CNT.RESET
3196
3197 ;GO, DO CONTROL RESET
3198 ;THIS IS A CALL FOR THE 'CNTRL-
3199 ;RESET' ROUTINE. A CONTROL RESET IS
3200 ;ISSUED AND AFTER A CERTAIN TIME
3201 ;IF THE 'CNTRL RDY' DOES NOT SET
3202 ;AN ERROR IS REPORTED. NOTE THAT
3203 ;THE PC IN ERROR MESSAGE IS THE
3204 ;PC WHERE 'CNT.RESET' IS LOCATED.
3205 007324 104420 TST.SIN ;THIS IS A VERY BASIC ERR & IF IT
3206 ;OCCURS GO BACK TO TEST 10
3207 007326 013701 002526 MOV RKCS,R1 ;GO CHECK IF SIN IS SET
3208 007332 012700 177764 MOV #-14,R0 ;IS SET, DO DRIVE RESET TO CLR IT
3209 007336 013702 002534 MOV RKDA,R2 ;SET UP COUNT FOR 12 SECTORS
3210 007342 013712 002544 MOV DRIVAD,R2 ;ADDRESS THE DRIVE
3211 007346 012704 032724 MOV #OUTBUF,R4
3212 007352 010477 173154 MOV R4,ARKBA ;ADRS TO WHICH X-FER DATA FROM DSK
3213 007356 012777 177764 173144 MOV #-14,ARKWC ;SET UP WORD COUNT FOR 12 HEADERS TO BREAD
3214 007364 012777 002005 173134 MOV #2005,ARKCS ;READ FORMAT, GO
3215
3216 007372 105777 173130 1$: TSTB ARKCS ;DID CNTRL RDY SET ON COMPLETION?
3217 007376 100411 BMI 2$ ;YES, BRANCH
3218 007400 005205 INC R5 ;NO, WAIT FOR IT TO SET
3219 007402 001373 BNE 1$ ;IF WAITED LONG ENOUGH REPORT
3220 ;ERROR, OTHERWISE LOOP BACK & WAIT
3221 007404 004737 020406 JSR PC,GT4RG ;GO, GET RKCS, ER, DS,DA
3222 007410 013737 002544 001202 MOV DRIVAD,$REG10
3223 007416 104415 BRKDA4
3224
3225 007420 104045 ERROR 45 ;GO TO 'BD4' & BREAK CONTENTS OF
3226 ;$REG10 INTO DR#,CYL,SUR,SEC BITS
3227 ;CNTRL RDY DID NOT SET ON COMPLETION
3228 ;OF READ FORMAT-OF CYLINDER 0,
3229 ;SECTORS 0-13
3230 007422 004737 020646 2$: JSR PC,CHKHE ;READ FMT WAS DONE STARTING AT <DSK-ADRES>
3231 ;INDICATED IN EROR MESGE
3232 007426 104046 ERROR 46 ;CHECK IF 'ERR' OR 'HE' BIT IS SET,
3233 ;IF YES RETURN HERE.
3234 ;'ERR' OR 'HE' BIT SET ON DOING
3235 ;READ FMT-OF CYLINDER 0, SEC 0-13
3236 007430 013705 002544 3$: MOV DRIVAD,R5 ;READ FMT WAS DONE STARTING AT <DSK-ADRES>
3237 007434 062705 000020 ADD #20,R5 ;INDICATED IN EROR MESGE
3238 ;RKDA SHOULD HAVE INCREMENTD TO (R2)
3239 007440 004737 020702 JSR PC,CHKDA1 ;CHECK IF RKDA INCREMENTED CORRECTLY,
3240 ;IF NOT, RETURN HERE.
3241 007444 104040 ERROR 40 ;RKDA DID NOT INCREMENT BY 12
3242 ;AFTER A 'RD FMT' OF 12 HEADERS OF

```

L05

MAINDEC-11-DZRKK-C
DZRKKC.P11 T21MACY11 27(732) 16-SEP-76 16:00 PAGE 64
CHECK 'READ FORMAT'-CYLINDER 0, SECTOR 0-13

```

3243                                     ;CYLINDER 0, SECTORS 0-13
3244                                     ;RKBA SHOULD INCREMENT BY 24 BYTES
3245                                     ;AT THE END OF X-FER
3246 007446 022777 032754 173056 4$: CMP #OUTBUF+30,ARKBA ;DID RKBA INCREMENT CORRECTLY?
3247 007454 001407 BEQ 5$ ;YES, BRANCH
3248 007456 012737 032754 001162 MOV #OUTBUF+30,$REG0 ;GET EXPCTD RKBA
3249 007464 017737 173042 001164 MOV ARKBA,$REG1 ;GET ACTUAL RKBA
3250 007472 104042 ERROR 42 ;RKBA DID NOT INCREMENT CORRECTLY
3251                                     ;AFTER READ FORMAT OF 12 HEADERS
3252 007474 004737 020730 5$: JSR PC,CHKWC ;GO CHECK IF RKWC OVERFLOWED TO 0
3253                                     ;IF NOT RETURN HERE.
3254 007500 104041 ERROR 41 ;RKWC DID NOT OVERFLOW TO 0
3255                                     ;AFTER 'RD FMT' OF 12 HEADERS
3256                                     ;OF CYLINDER 0
3257 007502 005724 6$: TST (R4)+ ;WAS THE CORRECT HEADER RECIEVED?
3258 007504 001413 BEQ 7$ ;YES, BRANCH
3259 007506 010037 001162 MOV RD,$REG0 ;GET SECTOR FOR WHICH THE HEADER
3260 007512 062737 000014 001162 ADD #14,$REG0 ;COULD NOT BE READ CORRECT
3261 007520 005037 001164 CLR $REG1 ;EXPCTD HEADER-0, FOR CYL 0
3262 007524 014437 001166 MOV -(R4),$REG2 ;GET WRONG HEADER RECVD
3263 007530 104043 ERROR 43 ;HEADER WAS NOT READ RIGHT FOR
3264                                     ;SECTOR (AS IN ER MSGE), & CYL 0
3265 007532 005724 TST (R4)+ ;WAS THE CORRECT HEADER RECVD?
3266 007534 005200 7$: INC RD ;YES, HAVE U CHECKED FOR ALL 12 SECTORS?
3267 007536 001361 BNE 6$ ;IF NOT, LOOP BACK & CHK HDR FRM NXT SECTR
3268
3269 007540 004737 020754 JSR PC,CHKER ;CHECK IF ANY BIT IN RKER IS SET,
3270                                     ;IF YES, RETURN HERE.
3271 007544 104036 ERROR 36 ;RKER BIT SET ON DOING RD FMT
3272                                     ;OF CYL 0, SECTORS 0-13
3273 007546 022711 002204 8$: CMP #2204,AR1 ;DOES RKCS STILL CONTAIN FUNCTION BITS?
3274 007552 001406 BEQ TST22 ;YES, EXIT
3275 007554 012737 002204 001162 MOV #2204,$REG0 ;GET EXPCTD RKCS
3276 007562 011137 001164 MOV AR1,$REG1 ;GET ACTUAL RKCS
3277 007566 104024 ERROR 24 ;RKCS DID NOT CONTAIN 'RD FMT'
3278                                     ;FUNCTION BITS ON COMPETION OF
3279                                     ;THE FUNCTION
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298

```

```

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

```


MAINDEC-11-DZRKK-C
DZRKKC.P11 T22MACY11 27(732) 16-SEP-76 16:00 PAGE 65
CHECK 'READ', CYLINDER 0, SECTORS 0 TO 13

```

3299 ;*IS READ BECAUSE THE SIMULATOR CAN STORE ONLY 1 SECTOR (256 WORDS).
3300 ;*HENCE ONLY THE DATA WRITTEN LAST CAN BE READ BACK.
3301 ;*****
3302 007570 000004 TST2: SCOPE
3303 007572 012737 007644 001110 MOV #1$, $LPERR ;SET RETURN ADRES FOR LUPING
3304 ;ON ERROR (SW 9)
3305 007600 013703 002526 MOV RKCS, R3
3306 007604 013701 002544 MOV DRIVAD, R1
3307 007610 010105 MOV R1, R5
3308 007612 012704 032724 MOV #OUTBUF, R4
3309 007616 005737 002540 TST SIMUL ; TESTING ON SIMULATOR?
3310 007622 001405 BEQ 9$ ; NO, BRANCH
3311 ; IF TESTING ON SIMULATOR READ
3312 ; SECTOR 13 ONLY
3313 007624 052701 000013 BIS #13, R1 ; SET BITS FOR SEC 13
3314 007630 052705 000020 BIS #20, R5 ; RKDA SHOULD INCRMNT TO THIS AFTER READ
3315 007634 000403 BR 1$
3316 007636 012702 177764 9$: MOV #-14, R2 ; SET COUNT FOR 12 SECTORS
3317 007642 005205 INC R5 ; RKDA SHOULD INCREMENT TO
3318 ; THIS AFTER 1 SECTOR READ
3319 007644 104412 1$: CNT.RESET ; GO, DO CONTROL RESET
3320 ; THIS IS A CALL FOR THE 'CNTRL-
3321 ; RESET' ROUTINE. A CONTROL RESET IS
3322 ; ISSUED AND AFTER A CERTAIN TIME
3323 ; IF THE 'CNTRL RDY' DOES NOT SET
3324 ; AN ERROR IS REPORTED. NOTE THAT
3325 ; THE PC IN ERROR MESSAGE IS THE
3326 ; PC WHERE 'CNT.RESET' IS LOCATED.
3327 ; THIS IS A VERY BASIC ERR & IF IT
3328 ; OCCURS GO BACK TO TEST 10
3329 007646 104420 TST.SIN ; GO CHECK IF SIN IS SET
3330 ; IF SET, DO DRIVE RESET TO CLR IT
3331 007650 010177 172660 MOV R1, @RKDA ; ADDRESS THE DRIVE
3332 007654 010477 172652 MOV R4, @RKBA ; ADRES TO WHICH X-FER DATA FROM DISK
3333 007660 012777 177400 172642 MOV #-400, @RKWC ; SETUP WORD COUNT
3334 007666 012713 000005 MOV #5, @R3 ; READ, GO
3335
3336 007572 005000 2$: CLR R0 ; DID CNTRL RDY SET ON COMPETION?
3337 007674 105713 TSTB @R3 ; YES, BRANCH
3338 007676 100410 BMI 3$ ; NO, WAIT FOR IT TO SET
3339 007700 005200 INC R0 ; IF WAITED LONG ENOUGH, REPORT
3340 007702 001374 BNE 2$ ; ERROR, OTHERWISE LOOP BAK & WAIT
3341
3342 007704 004737 020406 JSR PC, GT4RG ; GO, GET RKCS, ER, DS, DA
3343 007710 010137 001202 MOV R1, $REGIO ; GET SECTOR ADDRESS WHERE ERROR OCCURED
3344 007714 104415 BRKDA4 ; GO TO 'BDAY' & BREAK CONTENTS OF
3345 ; $REGIO INTO DR #, CYL SUR, SEC BITS
3346 007716 104045 ERROR 45 ; CNTRL RDY DID NOT SET ON COMPLETION
3347 ; OF READ OF CYLINDER 0, SECTOR
3348 ; AS SHOWN IN <DSK-ADRES>
3349 ; READ WAS DONE STARTING AT <DSK-ADRES>
3350 ; INDICATED IN EROR MESGE
3351 007720 004737 020640 3$: JSR PC, CHKHE1 ; CHECK IF 'ERR' OR 'HE' BIT IS SET,
3352 ; IF YES RETURN HERE.
3353 007724 104046 ERROR 46 ; HE OR ERR BIT SET
3354 ; ON 'READ' OF CYLINDER 0, SECTOR

```

3355									;AS SHOWN IN <DSK-ADRES>
3356									;READ WAS DONE STARTING AT <DSK-ADRES>
3357									;INDICATED IN EROR MESGE
3358	007726	020114		4\$:	CMP	R1,(R4)			;WAS THE DATA WORD RECVD, CORRECT?
3359									;THE FIRST DATA WORD OF EACH SECTOR
3360									;IS AN ADRS WORD COMRISING OF DRIVE NO,
3361									;CYLINDER ADRS, SUR, SECTOR ADRS
3362	007730	001407			BEQ	5\$			
3363	007732	010137	001162		MOV	R1,\$REGO			;GET EXPCTD DATA WORD FROM DISK
3364	007736	011437	001164		MOV	(R4),\$REG1			;GET THE DATA WORD RECVD
3365	007742	010137	001166		MOV	R1,\$REG2			;GET DISK ADRS
3366	007746	104044			ERROR	44			;DID NOT RECIEVE CORRECT DATA WORD ON
3367									;READ, OF CYLINDER 0, SECTOR AS SHOWN IN 'DSK
3368									;ADRES' OF EXPCTD DATA WORD
3369	007750	004737	020702	5\$:	JSR	PC,CHKDA1			;CHECK IF RKDA INCREMENTED CORRECTLY,
3370									;IF NOT RETURN HERE.
3371	007754	104040			ERROR	40			;RKDA DID NOT INCREMENT CORRECTLY
3372									;AFTER READ OF 1 WORD, FROM CYL 0
3373									;SEC IN ERROR IS 1 LESS THAN THAT
3374									;SHOWN IN EXPCTD RKDA
3375									
3376									;AS A RESULT OF 'WRT FMT' IN A PREVIOUS TEST
3377									;FIRST WORD OF EVERY SECTOR IS NON-
3378									;ZERO (PSUEDO-HDR), REST 377 WORDS
3379									;ARE ALL O'S.
3380									;CHECK IF THE REST OF THE 377
3381									;WORDS ARE ALL O'S
3382	007756	012737	177775	002564	MOV	#-3,EFLG1			;ALLOW ONLY 3 ERRORS
3383	007764	012700	032726		MOV	#OUTBUF+2,R0			;INITIALIZE PTR TO 2ND WRD IN BUFR
3384	007770	012737	177401	002556	MOV	#-377,COUNT			;CHECK 377 WORDS IN THE BUFFER
3385	007776	005710		11\$:	TST	R0			;IS THIS WRD 0?
3386	010000	001005			BNE	12\$			;NO, ERROR
3387	010002	005720			TST	(R0)+			;INCRMNT PTR TO NXT WRD
3388	010004	005237	002556		INC	COUNT			;CHKD ALL 377 WRDS'
3389	010010	001372			BNE	11\$			
3390	010012	000412			BR	7\$			;YES, BRANCH
3391	010014	005037	001162	12\$:	CLR	\$REGO			;GET EXPCTD WORD
3392	010020	012037	001164		MOV	(R0)+,\$REG1			;GET WORD RECVD
3393	010024	010137	001166		MOV	R1,\$REG2			;GET DISK ADRS, ERROR IN THIS
3394									;SECTOR
3395	010030	104044			ERROR	44			;DATA ERROR, THE LAST 377 WORDS
3396									;READ FROM EACH SECTOR SHOULD BE 0
3397									;IN A PREVIOUS TEST, FIRST WORD OF
3398									;EVERY SEC (CYL 0) WAS WRITTEN AS A
3399									;PSUEDO-HDR, REST OF THE WORDS IN THE
3400									;SECTR ARE AUTOMATICALLY WRITTEN AS
3401									;O'S. THIS ERROR MAY MEAN THAT IT
3402									;DIDN'T HAPPEN SO
3403	010032	005237	002564		INC	EFLG1			;ALLOW ONLY 3 DATA ERORS OF THIS KIND
3404	010036	001357			BNE	11\$			
3405									
3406									
3407	010040	005737	002540	7\$:	TST	SIMUL			;TESTING ON SIMULATOR?
3408	010044	001011			BNE	10\$			;YES BRANCH
3409									;IF NOT TESTING ON SIMULATOR GO AHEAD
3410									; & READ ALL 12 SECTORS ON CYL 0

MAINDEC-11-DZK-K-C
 DZK-K-C.P11 T22

 MACY11 27(732) 16-SEP-76 16:00 PAGE 67
 CHECK 'READ' CYLINDER 0. SECTORS 0 TO 13

3411	010046	005201			INC	R1		: INCREMENT DRIV-ADRES TO NXT SECTOR
3412	010050	005205			INC	R5		: INCREMENT 'EXPCTD DRIV-ADRES'
3413	010052	122705	000014		CMPB	#14,R5		: R U GOING TO READ THE LAST SECTOR?
3414	010056	001002			BNE	.+6		: IF NOT, BRANCH
3415	010060	062705	000004		ADD	#4,R5		: IF YES, INCREMENT 'EXPCTD RKDA'
3416								: CORRECTLY
3417	010064	005202			INC	R2		: HAVE U READ ALL 12 SECTORS?
3418	010066	001266			BNE	1\$		: IF NOT LOOP BACK & READ THE
3419								: NXT SECTOR
3420	010070	022713	000204	10\$:	CMP	#204,R3		: DOES RKCS, STILL HAVE THE 'READ' FUNCTION
3421	010074	001406			BEQ	8\$		: YES, BRANCH
3422	010076	012737	000204	001162	MOV	#204,\$REG0		: GET EXPCTD RKCS
3423	010104	011337	001164		MOV	R3,\$REG1		: GET RKCS RECVD
3424	010110	104024			ERROR	24		: RKCS SHOULD STILL CONTAIN THE 'READ'
3425								: FUNCTION BITS
3426	010112	104412		9\$:	CNT.RESET			: GO DO CONTROL RESET
3427								: THIS IS A CALL FOR THE 'CNTRL-
3428								: RESET' ROUTINE. A CONTROL RESET IS
3429								: ISSUED AND AFTER A CERTAIN TIME
3430								: IF THE 'CNTRL RDY' DOES NOT SET
3431								: AN ERROR IS REPORTED. NOTE THAT
3432								: THE PC IN ERROR MESSAGE IS THE
3433								: PC WHERE 'CNT.RESET' IS LOCATED.
3434								: THIS IS A VERY BASIC ERR & IF IT
3435								: OCCURS GO BACK TO TEST 10
3436	010114	005777	172416		TST	RKDB		: DID CNTRL RESET CLEAR RKDB?
3437	010120	001407			BEQ	TST23		: YES, EXIT
3438	010122	013737	002536	001162	MOV	RKDB,\$REG0		: GET ADRES OF RKDB
3439	010130	017737	172402	001164	MOV	RKDB,\$REG1		: GET CONTENTS OF RKDB
3440	010136	104102			ERROR	102		: CONTROL RESET DID NOT
3441								: CLEAR RKDB

:*****

```

: *TEST 23      CHECK 'WRITE FORMAT' OF THE DISK
: *THIS TEST WRITE FORMATS THE ENTIRE DISK.  THE FIRST
: *WORD OF EVERY SECTOR IS WRITTEN TO BE A PSUEDO-HEADER
: *CONSISTING OF THE DRIVE #, CYLINDER #, SURFACE & SECTOR #.
: *1 SECTOR IS WRITTEN AT A TIME.  THE WRITING IS DONE
: *IN THIS ORDER: CYL 0-SUR 0; CYL 0-SUR 1;  CYL 1-SUR 0
: *CYL 1-SUR 1; CYL 2-SUR 0; CYL 2-SUR 1----- CYL 312-SUR 1.
: *IMPORTANCE OF THIS TEST SHOULD BE REALIZED, THIS IS
: *THE FIRST TIME EACH & EVERY SECTOR ON THE DISK IS
: *ACCESSED & WRITTEN ON.  THIS IS THE FIRST TIME RKDA
: *IS BEING MADE TO INCREMENT OVER THE ENTIRE DISK (FROM
: *000000 TO 014520) IF A 'SIN' OCCURS AT ANY POINT
: *A DRIVE RESET IS DONE BEFORE DOING WRT FMT FOR THE NEXT
: *SECTOR.  ANY OTHER ERROR IS CLEARED THROUGH A CONTROL RESET.
: *THE FOLLOWING CHECKING IS DONE AFTER WRITING EACH
: *CYLINDER.
: *1. CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
: *OF THE FUNCTION.
: *2. IF 'SIN' OCCURRED?
: *3. IF 'HE' OR 'ERR' BIT SET?
: *4. IF RKDA INCREMENTED CORRECTLY, INCLUDING BOUNDARY
: *CONDITIONS (SECTOR COUNTER BITS OVERFLOWING INTO SURFACE.

```

MAINDEC-11-DZRKC-C
DZRKC.P11 T23

MACY11.27(732) 16-SEP-76 16:00 PAGE 68
CHECK 'WRITE FORMAT' OF THE DISK

```

3467      : *SURFACE BIT OVERFLOWING INTO CYLINDER BITS) AT THE END
3468      : *OF THIS POINTERS ARE INCREMENTED ADJUSTED, ETC.
3469      : *8 'WRT FMT' ON THE NEXT SECTOR IS DONE.
3470      : *****
3471 010140 000004      TST23: SCOPE
3472 010142 012737 000001 001206      MOV      #1, $TIMES
3473 010150 012737 010200 001110      MOV      #1$, $LPERR
3474      : DO 1 ITERATION
3475 010156 005003      CLR      R3
3476      : SET RETURN ADRES FOR LUPING
3477 010160 012704 177465      MOV      #-313, R4
3478 010164 012702 177764      MOV      #-14, R2
3479 010170 013701 002544      MOV      $DRIVAD, R1
3480 010174 010105      MOV      R1, R5
3481 010176 005205      INC      R5
3482 010200 104412      1$: CNT.RESET
3483      : GO, DO CONTROL RESET
3484      : THIS IS A CALL FOR THE 'CNTRL-
3485      : RESET' ROUTINE. A CONTROL RESET IS
3486      : ISSUED AND AFTER A CERTAIN TIME
3487      : IF THE 'CNTRL RDY' DOES NOT SET
3488      : AN ERROR IS REPORTED. NOTE THAT
3489      : THE PC IN ERROR MESSAGE IS THE
3490      : PC WHERE 'CNT.RESET' IS LOCATED.
3491      : THIS IS A VERY BASIC ERR & IF IT
3492      : OCCURS GO BACK TO TEST 10
3493      : GO CHECK IF SIN IS SET
3494      : IF SET, DO DRIVE RESET TO CLR IT
3495 010204 005037 002556      7$: CLR      COUNT
3496 010210 010137 032724      MOV      R1, $OUTBUF
3497      : THIS WORD TO BE WRITTEN. THE FIRST
3498      : WORD OF EACH SECTOR WILL BE THE ACTUAL
3499      : DISK-ADRES, CONSISTING OF THE DRIVE NO.
3500 010214 012777 032724 172310      MOV      $OUTBUF, $RKBA
3501 010222 012777 177777 172300      MOV      #-1, $RKWC
3502 010230 010177 172300      MOV      R1, $RKDA
3503      : SET UP WORD COUNT
3504 010234 012777 002003 172264      MOV      #2003, $RKCS
3505      : ADRES THE DRIVE, WITH CORRECT CYL
3506      : & SECTOR ADRES
3507      : WRITE FORMAT, GO
3508 010242 105777 172260      2$: TSTB    $RKCS
3509 010246 100411      BMI      3$
3510 010250 005237 002556      INC      COUNT
3511 010254 001372      BNE      2$
3512      : DID CNTRL RDY SET
3513      : YES, BRANCH
3514      : NO, HAVE U WAITED LONG ENOUGH?
3515      : IF NOT, LOOP BACK & WAIT
3516      : IF YES, REPORT ERROR
3517 010256 004737 020406      JSR      PC, $T4RG
3518 010262 010137 001202      MOV      R1, $REG10
3519 010266 104415      BRKDA4
3520      : GO, GET RKCS, ER, DS, DA
3521      : GET DISK ADRES, WHERE ERROR OCCURED
3522      : GO TO 'BD4' & BREAK CONTENTS OF
3523      : $REG10 INTO DR # CYL, SUR, SEC BITS
3524      : CNTRL RDY DID NOT SET ON COMPLETION
3525      : OF 'WRITE FORMAT' ON SECTOR AS
3526      : SHOWN IN <DSK-ADRES>
3527      : WRT FMT WAS DONE STARTING AT <DSK-ADRES>
3528      : INDICATED IN EROR MSGE.
3529      : DID SIN BIT SET?
3530      : NO, BRANCH
3531      : GO, GET RKCS, ER, DS
3532 010272 032777 001000 172222 3$: BIT      #1000, $RKDS
3533 010300 001405      BEQ      4$
3534 010302 004737 020444      JSR      PC, $T3RG
3535 010306 010137 001170      MOV      R1, $REG3
3536      : GET, DISK-ADRES WHERE ERROR OCCURED

```

MAINDEC-11-DZRKK-C
DZRKKC.P11 T23

MACY11.27(732) 16-SEP-76 16:00 PAGE 69
CHECK 'WRITE FORMAT' OF THE DISK

3523	010312	104001		ERROR	1		; SIN SET WHILE DOING WRT FMT
3524							; TO DISK-ADRES (AS IN \$REG3)
3525							
3526	010314	004737	020640	4\$:	JSR	PC,CHKHE1	; CHECK IF 'ERR' OR 'HE' BIT IS SET
3527							; IF YES, RETURN HERE.
3528	010320	104032		ERROR	32		; HE OR ERR SET WHILE DOING WRITE
3529							; FORMAT ON SECTOR AS INDICATED IN
3530							; <DSK-ADRES>
3531							; WRT FMT WAS DONE STARTING AT <DSK-ADRES>
3532							; INDICATED IN EROR MSGE.
3533	010322	004737	020702	5\$:	JSR	PC,CHKDA1	; CHECK IF RKDA INCREMENTED CORRECTLY,
3534							; IF NOT, RETURN HERE.
3535	010326	104033		ERROR	33		; RKDA DID NOT INCREMENT CORRECTLY
3536							; AFTER 'WRITE FORMAT' WAS DONE
3537							; TO THE SECTOR PREVIOUS TO THAT
3538							; INDICATED IN 'EXPCTD' RKDA
3539	010330	005201		6\$:	INC	R1	; INCREMENT TO THE NXT SECTOR
3540	010332	005205			INC	R5	; INCREMENT R5, TO WHAT RKDA WILL INCREMENT
3541	010334	022702	177776		CMP	#-2,R2	; R U GOING TO FORMAT THE LAST SECTOR
3542							; IN THE CYLINDER ?
3543	010340	001002			BNE	#+6	; IF NOT, BRANCH
3544	010342	062705	000004		ADD	#4,R5	; INCREMENT R5 CORRECTLY TO 'EXPCTD RKDA'
3545	010346	005202			INC	R2	; HAVE U FORMATTED ALL 12 SECTORS
3546							; ON THIS CYLINDER
3547	010350	001313			BNE	1\$	; IF NOT, LOOP BACK & FORMAT THE
3548							; NEXT SECTOR
3549							; YES
3550	010352	012702	177764		MOV	#-14,R2	; RESET THE COUNT FOR 12 SECTORS
3551	010356	042701	000037		BIC	#37,R1	; CLEAR THE SEC ADRES BITS
3552	010362	005703			TST	R3	; SURFACE 1?
3553	010364	001006			BNE	8\$	; YES, BRANCH
3554	010366	005203			INC	R3	; NO, SET FLAG
3555	010370	062701	000020		ADD	#20,R1	; INCREMENT TO THE NXT SURFACE
3556	010374	010105			MOV	R1,R5	; THIS IS WHAT RKDA SHOULD
3557	010376	005205			INC	R5	; INCREMENT TO.
3558	010400	000677			BR	1\$	; GO, DO NXT SURFACE
3559	010402	062701	000040	8\$:	ADD	#40,R1	; INCREMENT TO NXT CYL
3560	010406	010105			MOV	R1,R5	; POSITION FOR
3561	010410	005205			INC	R5	; EXPCTD RKDA
3562	010412	005003			CLR	R3	
3563	010414	005204			INC	R4	; HAVE U FORMATTED ALL 203 CYLINDERS
3564	010416	001270			BNE	1\$	; IF NOT, LOOP BACK & FORMAT THE
3565							; NEXT CYLINDER
3566							
3567							
3568							
3569							
3570							
3571							
3572							
3573							
3574							
3575							
3576							
3577							
3578							

```

*****
; *TEST 24      CHECK 'READ FORMAT' FOR THE ENTIRE DISK
; *THIS TEST READ FORMATS THE ENTIRE DISK, WHICH WAS WRT
; *FORMATTED IN THE PREVIOUS TEST.  THE FOLLOWING CHECKING
; *IS DONE
; *1. CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
; *OF FUNCTION
; *2. IF 'SIN' OCCURRED?
; *3. IF 'HE' OR 'ERR' OCCURRED?
; *4. RKDA INCREMENTED CORRECTLY.

```

E06

MAINDEC-11-DZKk-C
DZKkC.P11 T24MACY11 27(732) 16-SEP-76 16:00 PAGE 70
CHECK 'READ FORMAT' FOR THE ENTIRE DISK

```

3579                                     ;*5. IF THE CORRECT HEADER WAS READ.
3580                                     ;*6. IF RKWC OVERFLOWED CORRECTLY.
3581                                     ;*12 SECTORS (1 CYLINDER) ARE READ AT A TIME. IF 'SIN'
3582                                     ;*OCCURS A DRIVE RESET IS DONE BEFORE READING THE NEXT
3583                                     ;*SECTOR. READING IS DONE IN THIS ORDER CYL 0-SUR 0;
3584                                     ;*CYL 0-SUR 1; CYL 1-SUR 0; CYL 1-SUR 1; CYL 2-SUR 0;
3585                                     ;*CYL 2-SUR 1;-----CYL 312-SUR 1. IF TESTING ON SIMULATOR, ONLY
3586                                     ;*THE LAST CYLINDER (312), LAST SECTOR (13), SURFACE 1 IS READ.
3587                                     ;*****
3588 010420 000004                                †ST24: SCOPE
3589 010422 012737 000001 001206                MOV     #1,$TIMES           ;DO 1 ITERATION
3590 010430 012737 010514 001110                MOV     #1,$SLPERR        ;SET RETURN ADRES FOR LUPING
3591                                     ;ON ERROR (SW 9)
3592 010436 005037 002552                        CLR     INDX1          ;INDX1=0, SURFACE 0 BEING READ
3593                                     ;INDX1=1, SURFACE 1 BEING READ
3594 010442 013701 002544                        MOV     DRIVAD,R1         ;GET DRIVE ADRES
3595 010446 010102                                MOV     R1,R2
3596 010450 005737 002540                        TST     SIMUL             ;TESTING ON SIMULATOR?
3597 010454 001410                                BEQ     12$               ;NO, BRANCH
3598 010456 052701 014533                        BIS     #14533,R1         ;SET BITS FOR CYL 312, SEC 13, SUR 1
3599                                     ;ON SIMULATOR, CHECK ONLY CYL 312,
3600                                     ;SECTOR 13, SURFACE 1
3601 010462 052702 014540                        BIS     #14540,R2         ;RKDA SHOULD INCRMNT TO THIS AFTR
3602                                     ;RD FMT OF 1 SECTOR
3603 010466 012737 177777 002564                MOV     #-1,EFLG1        ;SET COUNT FOR READING HDR
3604                                     ;FROM 1 SECTOR ONLY
3605 010474 000407                                BR       1$
3606 010476 012705 177465 12$: MOV     #-313,R5           ;SET UP COUNT FOR 203 CYLINDERS
3607 010502 012737 177764 002564                MOV     #-14,EFLG1      ;SET COUNT FOR 12 HDRS TO BE
3608                                     ;READ FROM EACH CYLINDER
3609 010510 062702 000020                        ADD     #20,R2           ;THIS IS WHAT RKDA SHOULD INCREMENT
3610                                     ;BY, AFTER 'RD FMT' OF EACH CYLINDER
3611 010514 104412 1$: CNT.RESET                GO, DO CONTROL RESET
3612                                     ;THIS IS A CALL FOR THE 'CNTRL-
3613                                     ;RESET' ROUTINE. A CONTROL RESET IS
3614                                     ;ISSUED AND AFTER A CERTAIN TIME
3615                                     ;IF THE 'CNTRL RDY' DOES NOT SET
3616                                     ;AN ERROR IS REPORTED. NOTE THAT
3617                                     ;THE PC IN ERROR MESSAGE IS THE
3618                                     ;PC WHERE 'CNT.RESET' IS LOCATED.
3619                                     ;THIS IS A VERY BASIC ERR & IF IT
3620                                     ;OCCURS GO BACK TO TEST 10
3621
3622 010516 104420                                TST.SIN                ;CHECK IF SIN IS SET
3623                                     ;IF SET DO DRV-RESET TO CLR IT
3624
3625 010520 012703 032724                                MOV     #OUTBUF,R3       ;STORE ADRES OF BUFFER
3626 010524 005037 002554 11$: CLR     INDX2
3627 010530 010377 171776                                MOV     R3,ARKBA        ;ADRES TO WHICH DATA IS TO BE X-FERRED
3628                                     ;FROM THE DISK
3629 010534 013777 002564 171766                MOV     EFLG1,ARKWC     ;SET UP WORD COUNT FOR 12 HEADERS
3630                                     ;TO BE READ OFF EACH CYLINDER
3631                                     ;(ONLY ! FOR SIMULATOR)
3632 010542 010177 171766                                MOV     R1,ARKDA        ;ADRES THE DRIVE WITH CORRECT
3633                                     ;CYLINDER & SECTOR ADRES
3634 010546 012777 002005 171752                MOV     #2005,ARKCS    ;READ FORMAT, GO

```

MAINDEC-11-DZKK-C
DZKKC.P11 T24

MACY11 27(732) 16-SEP-76 16:00 PAGE 71
CHECK 'READ FORMAT' FOR THE ENTIRE DISK

3635											
3636	010554	105777	171746	2\$:	TSTB	ARKCS					:DID CNTRL RDY SET?
3637	010560	100411			BMI	3\$					:YES, BRANCH
3638	010562	005237	002554		INC	INDX2					:NO, HAVE U WAITED LONG ENOUGH?
3639	010566	001372			BNE	2\$					:IF NOT, LOOP BACK & WAIT FOR IT
3640											:IF YES, REPORT ERROR
3641	010570	004737	020406		JSR	PC,GT4RG					:GO, GET RKCS, ER, DS, DA
3642	010574	010137	001202		MOV	R1,\$REG10					:GET DRIV-ADRES STARTING WHICH
3643											: 'READ FORMAT' WAS DONE
3644	010600	104415			BRKDA4						:GO TO 'BD4' & BREAK CONTENTS OF
3645											: \$REG10 INTO DR #,CYL,SUR,SEC BITS
3646	010602	104045			ERROR	45					:CNTRL RDY DID NOT SET AFTER
3647											: READ FORMAT. 'RKDA' IN EROR MSGE
3648											:GIVES THE CONTENTS OF RKDA AT THE
3649											: TIME OF ERROR.
3650											: READ FMT WAS DONE STARTING AT <DSK-ADRES>
3651											: INDICATED IN EROR MSGE.
3652											
3653	010604	032777	001000	171710	3\$:	BIT	#1000,ARKDS				:DID 'SIN' SET?
3654	010612	001405				BEQ	4\$				:NO, BRANCH
3655	010614	004737	020414			JSR	PC,GT3RG				:GO, GET RKCS, ER, DS
3656	010620	010137	001170			MOV	R1,\$REG3				:GET DISK-ADRES WHERE 'SIN'
3657											: OCCURED
3658	010624	104001				ERROR	1				:SIN ERROR ON DOING RD FMT
3659											: TO CYL INDICATED IN \$REG3
3660											
3661	010626	004737	020640		4\$:	JSR	PC,CHKHE1				:CHECK IF 'ERR' OR 'HE' BIT IS SET,
3662											: IF YES, RETURN HERE.
3663	010632	104046				ERROR	46				:HE OR ERR WHILE DOING A READ
3664											:FORMAT. 'RKDA' IN EROR MSGE GIVES
3665											:THE CONTENTS OF RKDA AT THE TIME OF ERROR
3666											:READ FMT WAS DONE STARTING AT <DSK-ADRES>
3667											: INDICATED IN EROR MESGE
3668	010634	020277	171674		5\$:	CMP	R2,ARKDA				:DID RKDA INCREMENT CORRECTLY BY 12 SEC
3669	010640	001410				BEQ	6\$				
3670	010642	010237	001202			MOV	R2,\$REG10				:GET EXPCTD RKDA
3671	010646	104414				BRKDA0					:GO TO 'BD4' & BREAK CONTENTS OF
3672											: \$REG10 INTO DR #,CYL,SUR,SEC BITS
3673	010650	017737	171660	001202		MOV	ARKDA,\$REG10				:GET RECVD RKDA
3674	010656	104415				BRKDA4					:GO TO 'BD4' & BREAK CONTENTS OF
3675											: \$REG10 INTO DR #,CYL,SUR,SEC BITS
3676	010660	104040				ERROR	40				:RKDA DID NOT INCREMENT BY 12 SECTORS
3677											:AFTER RD FMT WAS DONE. ADRES
3678											:OF CYLINDER IN ERROR CAN BE OBTAINED
3679											:FROM 'EXPCTD' RKDA
3680	010662	013700	002564		6\$:	MOV	EFLG1,R0				:SET UP COUNT FOR 12 HEADERS TO B CHKD
3681											: (ONLY 1, IF SIMULATOR)
3682	010666	010104				MOV	R1,R4				:GET DRIV-ADRES FROM WHERE RDMT WAS DONE
3683	010670	042704	160037			BIC	#160037,R4				:GET THE CYLINDER ADRES ONLY. (HEADER)
3684	010674	020413			7\$:	CMP	R4,(R3)				:IS THE RECVD HEADER SAME AS EXPCTD?
3685	010676	001412				BEQ	8\$				
3686	010700	010437	001164			MOV	R4,\$REG1				:GET EXPCTD HEADER WORD
3687	010704	011337	001166			MOV	(R3),\$REG2				:GET HEADER WORD RECVD
3688	010710	010037	001162			MOV	R0,\$REG0				
3689	010714	062737	000014	001162		ADD	#14,\$REG0				:GET THE SECTOR (OCTAL NO) WHICH DID
3690											:NOT GIVE THE CORRECT HEADER

	ERROR	43
8\$:	TST	(R3)+
	INC	R0
	BNE	7\$
	JSR	PC,CHKWC
	ERROR	41
9\$:	TST	SIMUL
	BNE	TST25
	TST	INDX1
	BNE	10\$
	INC	INDX1
	ADD	#20,R1
	MOV	R1,R2
	ADD	#20,R2
10\$:	JMP	1\$
	CLR	INDX1
	BIC	#37,R1
	ADD	#40,R1
	MOV	R1,R2
	ADD	#20,R2
	INC	R5
	BEQ	TST25
	JMP	1\$

```

;DID NOT RECIEVE THE CORRECT HEADER
;WORD FROM 'SECTOR' AS INDICATED
;(NOTE SECTOR # IS OCTAL)
;INCREMENT POINTER TO THE NXT WORD
;IN MEMORY WHERE THE RECVD HDR IS STORED
;HAVE U CHECKED ALL 12 HEADERS?
;IF NOT, LOOP BACK & CHK THE NXT.
;YES, ALL HEADERS FOR THIS CYLINDER
;CHECKED.
;CHECK IF RKWC OVERFLOWED TO 0, IF
;NOT RETURN HERE.
;RKWC DID NOT OVERFLOW AFTER DOING
;RDFMT OF 12 SECTORS ON THE CYLINDER
;NOTE THAT 'RKDA' IS THE INCREMENTED
;RKDA AFTER THE RDFMT
;TSTING ON SIMULATOR?
;IF YES, EXIT
;NO
;DOING SURFACE 1
;YES, BRANCH
;NO
;INCREMENT DRIV ADRES TO THE NXT SURFACE

;THIS IS WHAT RKDA SHOULD INCREMENT
;TO, AFTER READ FMT OF THE CYLINDER
;GO RD FMT THE NXT SURFACE

;CLR SEC, SURFACE BITS
;INCREMENT TO NXT CYL
;THIS IS WHAT RKDA SHOULD BE
;AFTER RD FMT OF CYLINDER
;HAVE U DONE ALL CYLINDERS?
;EXIT
;IF NOT, LOOP BACK & READ FMT FROM
;THE NXT CYLINDER

```

```

;*****
;*TEST 25      CHECK 'READ' OF THE ENTIRE DISK
;*READ OF THE ENTIRE DISK (ONE WORD PER SECTOR) IS DONE
;*IN THIS TEST.  IN A PREVIOUS TEST THE FIRST WORD OF
;*EVERY SECTOR WAS WRITTEN LIKE A PSUEDO-HEADER (DRIVE #,
;*CYLINDER #, SURFACE & SECTOR #).  THESE PSUEDO HEADERS
;*WILL BE READ & CHECKED IN THIS TEST, PROVING THAT ANY
;*SECTOR CAN BE ACCESSED AND READ.
;*THE FOLLOWING CHECKING IS DONE
;*1. CNTRL RDY SETS WITHIN A CERTAIN TIME ON COMPLETION
;*OF FUNCTION.
;*2. IF 'SIN' OCCURRED?
;*3. IF 'HE' OR 'ERR' OCCURRED?
;*4. THE CORRECT FIRST WORD FROM EVERY SECTOR
;*WAS RECEIVED.  THIS WORD REFLECTS THE ABSOLUTE
;*DISK ADDRESS (DRV #, CYL #, SUR, SEC#) OF THAT SECTOR.
;*5. IF RKDB CONTAINED THE CORRECT WORD.
;*IF 'SIN' OCCURS DRIVE RESET IS DONE BEFORE READING
;*THE NEXT SECTOR.  READ IS DONE IN THIS ORDER SEC 0-11

```


H06

MAINDEC-11-DZRKK-C
DZRKKC.P11 T25MACY11 27(732) 16-SEP-76 16:00 PAGE 73
CHECK 'READ' OF THE ENTIRE DISK

```

3747      : *CYL 0 SUR 0 -> SEC 0-11 CYL 0 SUR 1 -> SEC 0-11 CYL 1.....
3748      : *IF TESTING ON SIMULATOR ONLY LAST CYLINDER (312), LAST
3749      : *SECTOR (13), SURFACE 1 IS READ.
3750      : *****
3751 011030 000004      †T25: SCOPE
3752 011032 012737 000001 001206      MOV      #1,$TIMES      : DO 1 ITERATION
3753 011040 012737 011104 001110      MOV      #1,$SLPERR      : SET RETURN ADRES FOR
3754      :                               : LOOPING ON ERROR (SW9)
3755 011046 012703 032724      MOV      #OUTBUF,R3
3756 011052 005004      CLR      R4      : FLAG, CLEAR WHEN READING SURFACE 0
3757      :                               : SET WHEN READING SURFACE 1
3758 011054 013701 002544      MOV      DRIVAD,R1      : GET DRIVE ADDRESS
3759 011060 005737 002540      TST      SIMUL      : TESTING ON SIMULATOR?
3760 011064 001403      BEQ      10$      : IF NOT BRANCH
3761 011066 052701 014533      BIS      #14533,R1      : SET ADRES BITS FOR LAST CYL (312)
3762 011072 000404      BR      1$      : LAST SECTOR (13), SURFACE 1
3763 011074 012700 177764      10$: MOV      #-14,R0      : SET COUNT FOR 12 SECTORS
3764 011100 012705 177465      MOV      #-313,R5      : SET UP COUNT FOR 203 CYLINDERS
3765
3766 011104 104412      1$: CNT.RESET      : GO, DO CONTROL RESET
3767      : THIS IS A CALL FOR THE 'CNTRL-
3768      : RESET' ROUTINE. A CONTROL RESET IS
3769      : ISSUED AND AFTER A CERTAIN TIME
3770      : IF THE 'CNTRL RDY' DOES NOT SET
3771      : AN ERROR IS REPORTED. NOTE THAT
3772      : THE PC IN ERROR MESSAGE IS THE
3773      : PC WHERE 'CNT.RESET' IS LOCATED.
3774      : THIS IS A VERY BASIC ERR & IF IT
3775      : OCCURS GO BACK TO TEST 10
3776 011106 104420      TST.SIN      : GO CHECK SIN, IF SET DO
3777      : DRIVE RESET TO CLR IT
3778 011110 005037 002552      8$: CLR      INDX1
3779 011114 010377 171412      MOV      R3,ARKBA      : ADRES TO WHICH DATA IS TO B X-FERRED
3780      : FROM THE DISK
3781 011120 012777 177777 171402      MOV      #-1,ARKWC      : SET UP WORD COUNT
3782 011126 010177 171402      MOV      R1,ARKDA      : ADRES THE DRIVE WITH CORRECT
3783      : CYLINDER & SECTOR ADRES
3784 011132 012777 000005 171366      MOV      #5,ARKCS      : READ, GO
3785
3786 011140 105777 171362      2$: TSTB      ARKCS
3787 011144 100411      BMI      3$
3788 011146 005237 002552      INC      INDX1
3789 011152 001372      BNE      2$
3790
3791 011154 004737 020406      JSR      PC,GT4RG
3792 011160 010137 001202      MOV      R1,$REG10
3793 011164 104415      BRKDA4
3794
3795 011166 104045      ERROR      45
3796
3797
3798
3799
3800
3801 011170 032777 001000 171324 3$: BIT      #1000,ARKDS      : DID 'SIN' SET?
3802 011176 001405      BEQ      4$      : NO, BRANCH

```

MAINDEC-11-DZRKK-C
DZRKKC.P11 T25

MACY11 27(732) 16-SEP-76 16:00 PAGE 74
CHECK 'READ' OF THE ENTIRE DISK

3803	011200	004737	020414		JSR	PC,GT3RG	:GO GET RKCS, ER, DS
3804	011204	010137	001170		MOV	R1,\$REG3	:GET DISK-ADRES WHERE SIN OCCURED3
3805	011210	104001			ERROR	1	:SIN' ERROR ON DOING READ FROM
3806							:DISK-ADRES INDICATED IN \$REG3
3807	011212	004737	020640	4\$:	JSR	PC,CHKHE1	:CHECK IF 'ERR' OR 'HE' BIT IS SET,
3808							:IF YES, RETURN HERE.
3809	011216	104046			ERROR	46	: 'HE' OR 'ERR' ON DOING A READ OF
3810							: 1 WORD FROM ADRES AS INDICATED
3811							: IN <DISK-ADRES>
3812							: 'RKDA' IN EROR MSGE GIVES THE
3813							: CONTENTS OF RKDA AT THE TIME OF EROR
3814	011220	020113		5\$:	CMP	R1,(R3)	: WAS THE CORRECT DATA WORD RECVD?
3815	011222	001407			BEG	6\$	
3816	011224	010137	001162		MOV	R1,\$REG0	: GET EXPCTD DATA WORD
3817	011230	011337	001164		MOV	(R3), \$REG1	: GET DATA WORD RECVD
3818	011234	010137	001166		MOV	R1,\$REG2	: GET DISK-ADRES
3819	011240	104044			ERROR	44	: DID NOT RECIEVE THE CORRECT
3820							: DATA WORD FROM DISK ON DOING
3821							: 1 WORD READ FROM 'DISK-ADRES'
3822							: AS INDICATED BY 'EXPCTD' DATA WORD
3823							: NOTE THAT IN A PREVIOUS TEST THE
3824							: FIRST WORD OF EACH SECTOR IS UNIQUELY
3825							: WRITTEN WITH A WORD GIVING THE
3826							: ABSOLUTE ADDRESS OF THAT SECTOR IN
3827							: TERMS OF, DRIV #, CYL ADRES, SUR, SEC ADRES.
3828	011242	020177	171270	6\$:	CMP	R1,\$RKDB	: DOES RKDB CONTAIN CORRECT WORD
3829	011246	001406			BEG	7\$	: YES, BRANCH
3830	011250	010137	001162		MOV	R1,\$REG0	: NO, GET EXPCTD RKDB
3831	011254	017737	171256 001164		MOV	\$RKDB,\$REG1	: GET RKDB RECVD
3832	011262	104037			ERROR	37	: RKDB ERROR ON READ.
3833							: FOR RK11C, AFTER A READ RKDB
3834							: CONTAINS CHECKSUM FOR THE SECTOR
3835							: READ.
3836							: WHEREAS FOR RK11D, AFTER READ
3837							: RKDB CONTAINS THE LAST WORD
3838							: READ FROM THAT SECTOR &
3839							: X-FERRED TO MEMORY
3840	011264	005737	002540	7\$:	TST	SIMUL	: TESTING ON SIMULATOR?
3841	011270	001022			BNE	TST26	: IF YES, EXIT
3842	011272	005201			INC	R1	: INCREMENT TO ADRES NEXT SECTOR
3843	011274	005200			INC	R0	: HAVE U CHKD ALL 12 SECTORS?
3844	011276	001302			BNE	1\$	: IF NOT, LUP BAK & CHK THE NXT
3845							: IF YES...
3846	011300	012700	177764		MOV	#-14,R0	: RESET THE COUNT FOR 12 SECTORS
3847	011304	042701	000037		BIC	#37,R1	: CLEAR SECTOR, SURFACE BITS
3848	011310	005704			TST	R4	: DOING SURFACE 1?
3849	011312	001004			BNE	9\$	: YES, BRANCH
3850	011314	005204			INC	R4	: NO
3851	011316	062701	000020		ADD	#20,R1	: INCREMENT THE ADRES TO NXT SURFACE
3852	011322	000670			BR	1\$	: GO READ SURFACE 1
3853	011324	005004		9\$:	CLR	R4	
3854	011326	062701	000040		ADD	#40,R1	: INCREMENT TO NXT CYL
3855	011332	005205			INC	R5	: HAVE U CHKD ALL 203 CYLINDERS
3856	011334	001263			BNE	1\$	: IF NOT, LOOP BACK & CHK THE NXT CYLINDER
3857							: YES
3858							


```

3859
3860
3861
3862
3863
3864
3865
3866
3867
3868 011336 000004
3869 011340 012737 000005 001206
3870 011346 012703 002566
3871
3872 011352 005037 002552
3873
3874 011356 013700 002526
3875 011362 013701 002522
3876 011366 013702 002524
3877 011372 012737 011400 001110
3878
3879 011400 000240
3880 011402 104412
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890 011404 104420
3891
3892
3893
3894 011406 013704 002544
3895 011412 051304
3896 011414 010477 171114
3897 011420 012710 000011
3898
3899 011424 104411
3900
3901 011426 104021
3902
3903
3904
3905 011430 005005
3906 011432 032711 000100
3907 011436 001005
3908 011440 005205
3909 011442 001373
3910 011444 004737 020406
3911 011450 104026
3912
3913 011452 032711 001000
3914 011456 001403

*****
;TEST 26 CHECK 'SEEK' FUNCTION, WITH DIFFERENT VELOCITY MODES
; * THIS TEST CHECKS SEEK IN DIFFERENT VELOCITY MODES (DIFF < 3,
; * 3 < DIFF < 31, DIFF > 31). FOR THESE 3 BASIC VELOCITIES SEEK IS DONE BOTH
; * IN FWD AND REV DIRECTION TO CHECK THE ADDER & DIFFERENCE LOGIC. IF
; * WHILE DOING A SEEK 'SIN' OCCURS, A DRIVE RESET IS DONE TO INITIALIZE
; * THE POSITIONING LOGIC
*****
TST26: SCOPE
MOV #5, $TIMES ;DO 5 ITERATIONS
MOV #SEEK0, R3 ;INITIALIZE POINTER TO THE FIRST
;SEEK ADDRESS
CLR INDX1 ;INDX1, WHEN 0 INDICATES SEEK IN FWD DIRECTION
; WHEN 1 INDICATES SEEK IN REV DIRECTION
MOV RKCS, R0
MOV RKDS, R1
MOV RKER, R2
MOV #1$, $LPERR ;SET RETURN ADRES FOR LLPING ON
;EROR (SW 9)
1$: NOP
2$: CNT.RESET
;GO, DO CONTROL RESET
;THIS IS A CALL FOR THE 'CNTRL-
;RESET' ROUTINE. A CONTROL RESET IS
;ISSUED AND AFTER A CERTAIN TIME
;IF THE 'CNTRL RDY' DOES NOT SET
;AN ERROR IS REPORTED. NOTE THAT
;THE PC IN ERROR MESSAGE IS THE
;PC WHERE 'CNT.RESET' IS LOCATED.
;THIS IS A VERY BASIC ERR & IF IT
;OCCURS GO BACK TO TEST 10
;GO, CHECK IF SIN IS SET, IF SET
;DO DRV-RESET TO CLEAR IT

TST.SIN

MOV DRIVAD, R4 ;GET DRIV-ADRES
BIS (R3), R4 ;SET CYLINDER BITS
MOV R4, $AKDA ;ADDRS THE DRIVE
MOV #11, $RO ;SET 'SEEK', 'GO'

CHKCRDY
ERROR 21
;GO CHECK IF CONTROL RDY IS SET
;IF SO, SKIP THE EROR MESSAGE.
;'CNTRL RDY' DID NOT SET AFTER
;SENDING CYL ADD TO THE DRIV, 'ADD ACK'
;FROM DRIVE SHLD HAVE COME BACK
;THEREUPON SETTING 'CNTRL RDY'

4$: CLR R5
5$: BIT #100, $R1 ;DID R/W/S RDY SET?
BNE 6$ ;YES, BRANCH
INC R5 ;NO, WAIT
BNE 5$ ;WAITED LONG?
JSR PC, GT4RG ;GO, GET RKCS, ER, DS, DA
ERROR 26 ;R/W/S RDY DID NOT SET ON
;COMPLETION OF SEEK
6$: BIT #1000, $R1 ;DID SIN SET?
BEQ 7$ ;NO, BRANCH

```

K06

MAINDEC-11-DZRKK-C
DZRKKC.P11 T26MACY11 27(732) 16-SEP-76 16:00 PAGE 76
CHECK 'SEEK' FUNCTION, WITH DIFFERENT VELOCITY MODES

3915	011460	004737	020406		JSR	PC,GT4RG		:GO, GET RKCS, ER, DS, DA
3916	011464	104001			ERROR	1		:SIN SET, ON DOING SEEK
3917	011466	032710	140000	7\$:	BIT	#140000,2RD		:DID 'HE' OR 'ERR' SET?
3918	011472	001403			BEQ	8\$		:YES
3919	011474	004737	020406		JSR	PC,GT4RG		:GO, GET RKCS, ER, DS, DA
3920	011500	104022			ERROR	22		:ERR OF 'HE' BIT SET WHEN
3921								:SEEKING TO CYL AS INDICATED
3922								:IN RKDA
3923								
3924	011502	022710	000210	8\$:	CMP	#210,2RD		:DOES RKCS STILL CONTAIN THE 'SEEK' FNCTION
3925	011506	001406			BEQ	9\$		:YES - EXIT
3926	011510	011537	001164		MOV	2RD,\$REG1		:NO, GET RKCS RECVD
3927	011514	012737	000210	001162	MOV	#210,\$REG0		:GET EXPCTD RKCS
3928	011522	104024			ERROR	24		:RKCS SHOULD CONTAIN THE 'SEEK' BITS
3929								:IF NOT, ERROR
3930								
3931	011524	020477	171004	9\$:	CMP	R4,2RKDA		:DID RKDA CHANGE?
3932	011530	001406			BEQ	10\$		:NO
3933	011532	010437	001162		MOV	R4,\$REG0		:YES, GET EXPCTD?
3934	011536	017737	170772	001164	MOV	2RKDA,\$REG1		:GET RKDA
3935	011544	104027			ERROR	27		:RKDA CHANGED AFTER DOING SEEK
3936								
3937	011546	010477	170762	10\$:	MOV	R4,2RKDA		:ADRES THE DRIVE, SEC 0
3938	011552	012777	032724	170752	MOV	#OUTBUF,2RKBA		:READ ONE HEADER INTO THIS
3939	011560	012777	177777	170742	MOV	#-1,2RKWC		:BUS ADRES
3940	011566	012710	002005		MOV	#2005,2RD		:GO, READ FORMAT
3941	011572	104413			CNT.RDY			:WAIT FOR CNTRL RDY
3942	011574	021337	032724		CMP	(R3),OUTBUF		:WAS THE CORRECT READE4R READ (FROM
3943	011600	001410			BEQ	11\$		:CYLINDER TO WHICH SEEK WAS DONE BEFORE)
3944	011602	005037	001162		CLR	\$REG0		:STORE SEC # FROME WHERE HDR WAS RD (0)
3945	011606	011337	001164		MOV	(R3),\$REG1		:GET EXPCTD HEADER
3946	011612	013737	032724	001166	MOV	OUTBUF,\$REG2		:GET HDR RECVD
3947	011620	104043			ERROR	43		:WRONG HDR WAS RECVD FROM CYLINDER (ADRES
3948								:IN ER MSGE). NOTE THAT A PURE SEEK WAS
3949								:DONE TO THIS CYL BEFORE READING HDR
3950								:USING READ FORMAT
3951	011622	005737	002552	11\$:	TST	INDX1		:SEEK IN REVRSE DIRECTION?
3952	011626	001007			BNE	12\$		:YES, BRANCH
3953	011630	005723			TST	(R3)+		:NO, INCREMENT PTR TO NXT SEEK ADRES
3954	011632	022703	002574		CMP	#SEEK2+2,R3		:DONE WITH ALL SKS IN FWD DIR?
3955	011636	001260			BNE	1\$		:NO, GO & DO NXT ONE
3956	011640	005237	002552		INC	INDX1		:SET FLAG INDICATING SK IN REVRSE
3957	011644	005743			TST	-(R3)		
3958	011646	005743		12\$:	TST	-(R3)		:POSITION PTR TO NXT SK IN REV
3959	011650	022703	002564		CMP	#SEEK0-2,R3		:DONE WITH ALL?
3960	011654	001251			BNE	1\$		:IF NOT, DO NXT ONE

```

*****
;TEST 27      CHECK DRIVE RESET FROM LAST CYLINDER
;THE HEADS ARE POSITIONED ON THE LAST CYLINDER (DOING
;AN IMPLIED SEEK-READ). THEN A DRIVE RESET IS ISSUED.
;IT'S CHECKED IF THE HEADS WERE BROUGHT BACK TO 0 BY
;DOING A 1 WORD READ & CHECKING THAT THE CORRECT WORD
;WAS RECEIVED. IF TESTING ON SIMULATOR THIS TEST IS SKIPPED.

```

MAINDEC-11-DZRKC-C
DZRKC.P11 T27

MACY11 27(732) 16-SEP-76 16:00 PAGE 77
CHECK DRIVE RESET FROM LAST CYLINDER

```

3971
3972 011656 000004
3973 011660 012737 000005 001206
3974 011666 005737 002540
3975 011672 001124
3976 011674 013701 002526
3977 011700 104412
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987 011702 005000
3988 011704 012703 032724
3989 011710 013704 002544
3990 011714 010405
3991 011716 052705 014500
3992 011722 010577 170606
3993 011726 012777 177777 170574
3994 011734 010377 170572
3995
3996 011740 012711 000005
3997
3998 011744 005000
3999 011746 104413
4000
4001
4002
4003
4004
4005
4006 011750 020513
4007 011752 001407
4008 011754 010537 001162
4009 011760 011337 001164
4010 011764 010537 001166
4011 011770 104044
4012
4013
4014
4015
4016 011772 012711 000015
4017 011776 104413
4018
4019
4020
4021
4022
4023
4024 012000 005000
4025 012002 032777 000100 170512
4026 012010 001011

*****
1$T27: SCOPE
MOV #5,$TIMES ;;DO 5 ITERATIONS
TST SIMUL ;R U ON A SIMULATOR?
BNE TST30 ;;YES, EXIT
MOV RKCS,R1
CNT.RESET ;GO, DO CONTROL RESET
;THIS IS A CALL FOR THE 'CNTRL-
;RESET' ROUTINE. A CONTROL RESET IS
;ISSUED AND AFTER A CERTAIN TIME
;IF THE 'CNTRL RDY' DOES NOT SET
;AN ERROR IS REPORTED. NOTE THAT
;THE PC IN ERROR MESSAGE IS THE
;PC WHERE 'CNT.RESET' IS LOCATED.
;THIS IS A VERY BASIC ERR & IF IT
;OCCURS GO BACK TO TEST 10

CLR R0
MOV #OUTBUF,R3 ;ADRES WHERE DATA WILL BE READ INTO
MOV DRIVAD,R4
MOV R4,R5
BIS #14500,R5 ;SET CYL ADRES=312 (OCTAL)
MOV R5,DRKDA ;ADRES THE DRIVE, LAST CYLINDER
MOV #-1,DRKWC ;READ 1 WORD
MOV R3,DRKBA ;INTO THIS MEMORY ADRES

MOV #5,DR1 ;READ, GO

CLR R0
1$: CNT.RDY ;THIS IS A CALL FOR CN.RDY ROUTINE
;WHICH WAITS FOR CNTRL RDY TO SET.
;A RETURN IS MADE AFTER CNTRL RDY
;SETS. IF WITHIN A CERTAIN TIME
;CNTRL RDY DOESN'T SET AN ERROR
;MESSAGE IS GIVEN. WAITING TIME
;883 MS FOR 11/20, 175 MS FOR 11/45
;WAS THE CORRECT WORD READ?
;YES, SEEK TO 312 WAS DONE CORRECTLY.2
;GET EXPCTD WORD
;GET WORD RECVD
;GET DSK-ADRES FROM WHERE WORD WAS READ
;DID NOT READ BACK CORRECT WORD FROM
;LAST CYL, SEC 0. IF TEST 45 & 46
;WERE SUCCESSFULLY DONE THIS
;ERROR MEANS THAT IMPLIED SEEK
;TO CYL 312 COULD NOT BE DONE
;DRIVE RESET, GO
;THIS IS A CALL FOR CN.RDY ROUTINE
;WHICH WAITS FOR CNTRL RDY TO SET.
;A RETURN IS MADE AFTER CNTRL RDY
;SETS. IF WITHIN A CERTAIN TIME
;CNTRL RDY DOESN'T SET AN ERROR
;MESSAGE IS GIVEN. WAITING TIME
;883 MS FOR 11/20, 175 MS FOR 11/45

2$: CMP R5,DR3
BEQ 3$
MOV R5,$REG0
MOV DR3,$REG1
MOV R5,$REG2
ERROR 44

3$: MOV #15,DR1
CNT.RDY

4$: CLR R0
BIT #100,DRKDS ;DID R/W/S RDY SET?
BNE 5$ ;YES, BRANCH

```

4027	012012	012702	177763		MOV	#-15,R2	; IF U R ON A SLOWER MACHINE
4028	012016	005202			INC	R2	& DO NOT NEED SUCH A LARGE MACHINE
4029	012020	001376			BNE	.-2	: TIME LOOP, CHANGE THESE 3
4030							: INSTRUCTIONS TO 'NOP' THE
4031							: LOOP TIME WILL BE REDUCED
4032							: TO 1100 MS
4033							
4034							: THE TOTAL TIME FOR THE ABOVE
4035							: LOOPS (W/O PUTTING 'NOP'S) IS
4036							: 5304 MS FOR 11/20 AND
4037							: 1061 MS FOR 11/45 WITH MOS
4038							: OR BIPOLAR MEMORY
4039	012022	005200			INC	R0	: WAITED LONG?
4040	012024	001366			BNE	4\$	: IF NOT, LUP BAK & WAIT
4041							: IF YES, ERROR
4042	012026	004737	020406		JSR	PC,GT4RG	: GET RKCS,ER,DS,DA
4043	012032	104026			ERROR	26	: R/W/S RDY DID NOT SET AFTER
4044							: DOING DRIVE RESET
4045	012034	032711	140000	5\$:	BIT	#140000,DR1	: DID HE OR ERR BIT SET?
4046	012040	001403			BEQ	6\$	: IF NOT, BRANCH
4047							
4048	012042	004737	020406		JSR	PC,GT4RG	: GET RKCS,ER,DS,DA FOR ERROR MESSAGE
4049	012046	104022			ERROR	22	: HE OR ERR BIT SET ON DOING DRIVE
4050							: RESET FROM LAST CYLINDER
4051	012050	005205		6\$:	INC	R5	: POSITION R5 TO EXPCTD RKDA
4052	012052	020577	170456		CMP	R5,DRKDA	: DID THE CYL ADRES BITS IN RKDA GET CHANGED?
4053	012056	001406			BEQ	7\$	: NO, BRANCH
4054	012060	010537	001162		MOV	R5,\$REGD	: GET EXPCTD RKDA
4055	012064	017737	170444	001164	MOV	DRKDA,\$REG1	: GET RKDA RECVD
4056	012072	104054			ERROR	54	: CYLINDER ADRES BITS IN RKDA
4057							: GOT CHANGED AFTER
4058							: DRIVE RESET, FROM LAST CYLINDER
4059	012074	012777	177777	170426	7\$:	MOV	#-1,DRKW C
4060	012102	010377	170424		MOV	R3,DRKBA	: READ 1 WORD
4061	012106	010477	170422		MOV	R4,DRKDA	: INTO THIS ADRES
4062							: FROM THIS DSK ADRES-CYL 0, SEC 0
4063	012112	012711	000005		MOV	#5,DR1	: ;READ, GO
4064							
4065	012116	005000			CLR	R0	
4066	012120	104413		8\$:	CNT.RDY		: THIS IS A CALL FOR CN.RDY ROUTINE
4067							: WHICH WAITS FOR CNTRL RDY TO SET.
4068							: A RETURN IS MADE AFTER CNTRL RDY
4069							: SETS. IF WITHIN A CERTAIN TIME
4070							: CNTRL RDY DOESN'T SET AN ERROR
4071							: MESSAGE IS GIVEN. WAITING TIME
4072							: 883 MS FOR 11/20, 175 MS FOR 11/45
4073	012122	020413		9\$:	CMP	R4,DR3	: WAS THE CORRECT WORD READ?
4074	012124	001407			BEQ	TST30	: :YES, EXIT
4075	012126	010437	001162		MOV	R4,\$REGD	: GET EXPCTD WORD
4076	012132	011337	001164		MOV	DR3,\$REG1	: GET WORD RECVD
4077	012136	010437	001166		MOV	R4,\$REG2	: GET DISK ADRES WHERE ERROR OCCURED
4078	012142	104044			ERROR	44	: DID NOT READ CORRECT WORD FROM
4079							: CYL 0, SEC 0. IF TEST 45 & 46
4080							: WERE SUCCESSFULLY DONE THIS
4081							: ERROR COULD MEAN THAT DRIVE-RESET
4082							: DID NOT BRING HEADS BACK TO 0.

4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138

012144 000004
012146 104412

012150 104420
012152 013704 002526

012156 012700 032724
012162 012701 177401
012166 012702 177400
012172 012703 177400

012176 0.0320
012200 005202
012202 060103
012204 010320
012206 005202
012210 001374

012212 012777 177400 170310
012220 012777 032724 170304
012226 013777 002544 170300

012234 012714 000003

```

*****
;TEST 30      'WRITE' - 256 WORD BLOCK ON SECTOR 0, CYLINDER 0
;THE TEST BELOW SHOULD BE CONSIDERED AS A SET UP PHASE FOR
;THE FOLLOWING TEST. IT WRITES A BLOCK OF 256 WORDS IN
;SECTOR 0, CYLINDER 0 WITH A SPECIFIC PATTERN AND THIS WRITTEN
;BLOCK WILL BE MADE USE OF IN THE NEXT TEST TO CHECK
;OUT 'WRITE-CHECK' AND 'READ CHECK' FUNCTIONS.
*****
TST30:  SCOPE
        CNT.RESET
;GO, DO CONTROL RESET
;THIS IS A CALL FOR THE 'CNTRL-
;RESET' ROUTINE. A CONTROL RESET IS
;ISSUED AND AFTER A CERTAIN TIME
;IF THE 'CNTRL RDY' DOES NOT SET
;AN ERROR IS REPORTED. NOTE THAT
;THE PC IN ERROR MESSAGE IS THE
;PC WHERE 'CNT.RESET' IS LOCATED.
;THIS IS A VERY BASIC ERR& IF IT
;OCCURS GO BACK TO TEST 10
;CHECK IF SIN IS SET, IF SET
;DO DRIVE RESET TO CLEAR IT

        TST.SIN
        MOV     RKCS,R4

;THE FOLLOWING CODE IS FOR SETTING
;UP THE I/O BUFFER IN MEMORY (STARTING AT
;OUTBUF), WITH A PARTICULAR 256 WORD PATTERN.
;STARTING FROM THE FIRST WORD IN THE BUFFER
;THE LO BYTE WILL BE A COUNT PATTERN
;FROM 0 TO 255 (DECIMAL), WHEREAS THE
;HI-BYTE WILL BE THE COMPLEMENT OF LO BYTE,
;A DECREASING COUNT PATTERN FROM 255 TO 0.
;I.E. THE BUFFER WILL LOOK LIKE:
;OUTBUF      (1 111 111 1 00 000 000)
;OUTBUF+2    (1 111 111 0 00 000 001)
;LAST WORD   (0 000 000 0 11 111 111)

        MOV     #OUTBUF,R0
        MOV     #177401,R1
        MOV     #-400,R2
        MOV     #177400,R3
;PATTERN GENERATING NUMBER
;SET UP COUNT FOR 256 WORDS
;SET UP THE FIRST PATTERN TO B WRITTEN

        MOV     R3,(R0)+
        INC     R2
        IS:    ADD     R1,R3
        MOV     R3,(R0)+
        INC     R2
        BNE     IS
;SET UP FIRST WORD IN I/O BUFFER
;INCREMENT COUNT
;SET UP NEXT WORD PATTERN
;WRITE IT IN NXT I/O BUFFER WORD
;HAVE U WRITTEN ALL 256 WORDS
;IF NOT GO & WRITE NEXT PATTERN

        MOV     #-400,DRKWC
        MOV     #OUTBUF,DRKBA
        MOV     DRIVAD,DRKDA
;WRITE 256 WORDS
;STARTING FROM THIS BUS ADRES
;TO THIS DISK ADRES. CYL 0, SEC 0

        MOV     #3,DR4
;WRITE, GO

```

Line	Address	Offset	Label	Instruction	Register	Comment
4139	012240	105714	25:	TSTB	3R4	: WAS CNTRL RDY CLEARED AS GO WAS SET?
4140	012242	100003		BPL	35-2	: YES, BRANCH
4141	012244	004737	020414	JSR	PC,GT3RG	: GET RKCS, ER, DS
4142	012250	104030		ERROR	30	: CNTRL RDY DID NOT CLEAR AS GO WAS SET
4143						: TO 'WRITE'
4144						
4145	012252	005002		CLR	R2	
4146	012254	105777	170246	TSTB	3RKCS	: DID CNTRL RDY SET?
4147	012260	100411		BMI	45	: YES, BRANCH
4148	012262	005202		INC	R2	: WAITED LONG ENOUGH?
4149	012264	001373		BNE	35	: IF NOT, LUP BAK & WAIT
4150						: IF YES, ERROR
4151	012266	004737	020406	JSR	PC,GT4RG	: GO, GET RKCD, ER, DS, DA
4152	012272	013737	002544	MOV	DRIVAD,\$REGIO	: GET THE STARTING ADRES
4153	012300	104415		BRKDA4		: BREAK CONTENTS OF \$REGIO INTO
4154						: DRV #, CYL, SUR, SEC #
4155	012302	104031		ERROR	31	: CNTRL RDY DID NOT SET ON COMPLETION
4156						: OF WRITE OF 256 WORDS ON CYL 0, SEC 0
4157						: 'RKDA' IN EROR MSGE GIVES THE
4158						: CONTENTS OF RKDA AT THE TIME OF EROR
4159						: WRITE WAS DONE STARTING AT <DSK-ADRES>
4160						: INDICATED IN EROR MSGE
4161	012304	004737	020646	JSR	PC,CHKHE	: CHECK IF 'ERR' OR 'HE' BIT IS SET,
4162						: IF YES, RETURN HERE
4163	012310	104032		ERROR	32	: HE OR ERR BIT SET ON DOING WRITE OF
4164						: 256 WORDS ON CYL 0, SEC 0
4165						: WRITE WAS DONE STARTING AT <DSK-ADRES>
4166						: INDICATED IN EROR MSGE
4167						: 'RKDA' IN EROR MSGE GIVES THE
4168						: CONTENTS OF RKDA AT THE TIME OF EROR
4169	012312	020077	170214	CMP	RO,3RKBA	: DID RKBA INCREMENT CORRECTLY?
4170	012316	001406		BEQ	65	: YES, BRANCH
4171	012320	010037	001162	MOV	RO,\$REGO	: GET EXPCTD RKBA
4172	012324	017737	170202	MOV	3RKBA,\$REG1	: GET RKBA RECVD
4173	012332	104035		ERROR	35	: RKBA DID NOT INCREMENT CORRECTLY
4174						: (BY 1000 OCTAL BYTES) AFTER WRITE
4175						: OF 400 (OCTAL) WORDS ON SEC 0, CYL 0
4176	012334	004737	020730	JSR	PC,CHKWC	: CHECK IF RKWC OVERFLOWED TO 0,
4177						: IF NOT RETURN HERE.
4178	012340	104034		ERROR	34	: RKWC DID NOT OVERFLOW, AFTER A
4179						: WRITE OF 256 WORDS ON CYL 0, SEC 0
4180	012342	004737	020674	JSR	PC,CHKDA	: CHECK IF RKDA INCREMENTED CORRECTLY,
4181						: IF NO, RETURN HERE
4182	012346	104033		ERROR	33	: RKDA DID NOT INCREMENT BY 1 AFTER
4183						: A WRITE OF 256 WORDS IN CYL 0, SEC 0
4184	012350	004737	020754	JSR	PC,CHKER	: CHECK IF ANY BIT RKER IS SET
4185						: IF YES RETURN HERE.
4186	012354	104036		ERROR	36	: RKER BIT SET ON DOING WRITE ON
4187						: CYLINDER 0, SECTOR 0
4188	012356	022714	000202	CMP	#202,3R4	: DOES RKCS STILL CONTAIN THE WRITE BITS?
4189	012362	001406		BEQ	TST31	: YES, EXIT
4190	012364	012737	000202	MOV	#202,\$REGO	: GET EXPECTED RKCS
4191	012372	011437	001164	MOV	3R4,\$REG1	: GET RKCS RECVD
4192	012376	104024		ERROR	24	: RKCS DID NOT CONTAIN THE 'WRITE'
4193						: BITS AFTER THE FUNCTION WAS DONE.
4194						

C07

MAINDEC-11-DZKk-C
DZKkC.P11 T30MACY11 27(732) 16-SEP-76 16:00 PAGE 81
'WRITE' - 256 WORD BLOCK ON SECTOR 0, CYLINDER 04195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250012400 000004
012402 104412

012404 104420

012406 012700 177400
012412 012701 032724

012416 005021

012420 005200

012422 001375

012424 005000

012426 012777 177400 170074

012434 012777 032724 170070

012442 013777 002544 170064

012450 012777 000005 170050

012456 105777 170044

012462 100411

012464 005200

012466 001373

012470 004737 020406

012474 013737 002544 001202

012502 104415

012504 104045

012506 032777 001000 170006

012514 001033

012516 012701 177400

012522 012702 177777

TEST 31 CHECK THAT WRITE WAS DONE CORRECTLY
 *THIS TEST CHECKS IF THE 'WRITE' OF 256 WORDS DONE IN PREVIOUS
 *TEST IS GOOD. THE SEQUENCE OF OPERATIONS IS AS FOLLOWING:
 *1) DO A READ OF 256 WORDS FROM SECTOR 0, CYLINDER 0
 * INTO A BUFFER STARTING AT 'OUTBUF'
 *2) COMPARE & CHECK THE DATA THAT IS READ (STARTING AT 'OUTBUF')
 * WITH THE DATA THAT WAS GENERATED PREVIOUSLY
 *3) REPORT AN ERROR IF THE DATA READ BACK FROM DISK DOES
 * NOT COMPARE WITH DATA THAT WAS SUPPOSE TO HAVE BEEN WRITTEN

TST31: SCOPE
CNT.RESET

GO, DO CONTROL RESET
 THIS IS A CALL FOR THE 'CNTRL-
 RESET' ROUTINE. A CONTROL RESET IS
 ISSUED AND AFTER A CERTAIN TIME
 IF THE 'CNTRL RDY' DOES NOT SET
 AN ERROR IS REPORTED. NOTE THAT
 THE PC IN ERROR MESSAGE IS THE
 PC WHERE 'CNT.RESET' IS LOCATED.
 THIS IS A VERY BASIC ERR& IF IT
 OCCURS GO BACK TO TEST 10
 CHECK IF SIN IS SET, IF SET
 DO DRIVE RESET TO CLEAR IT
 SET COUNT FOR 400 WORDS
 TO BE CLEARED IN THE BUFFER
 CLR THE 400 WORD BUFFER
 STARTING AT 'OUTBUF'

TST.SIN

SS: MOV #-400,R0
 MOV #OUTBUF,R1
 CLR (R1)+
 INC R0
 BNE SS
 CLR R0
 MOV #-400,ARKHC
 MOV #OUTBUF,ARKBA
 MOV DRIVAD,ARKDA

READ 256 WORDS
 INTO THIS ADRES
 STARTING FROM THIS DISK ADRES

MOV #5,ARKCS

READ, GO

IS: TSTB ARKCS
 BMI 2S
 INC R0
 BNE IS

DID CNTRL RDY SET?
 YES, BRANCH
 WAITED LONG ENOUGH?
 IF NOT, LUP BAK & WAIT
 ERROR, IF YES
 GO, GET RKCD, ER, DS, DA
 GET THE STARTING ADRES
 GO TO 'BDAY' & BREAK CONTENTS OF
 \$REGIO INTO DRV #, CYL, SUR, SEC BITS
 CNTRL RDY DID NOT SET AFTER READ
 OF 400 WORDS FROM CYL 0, SEC 0
 'RKDA' IN EROR MSGE GIVES THE
 CONTENTS OF RKDA AT THE TIME OF EROR
 READ WAS DONE STARTING AT (DSK-ADRES)
 INDICATED IN EROR MSGE
 IS SIN SET?
 IF YES, EXIT

JSR PC,GT4RG
 MOV DRIVAD,\$REGIO
 BRKDA4

ERROR 45

2S: BIT #1000,ARKDS
 BNE TST32
 SS: MOV #-400,R1
 MOV #177777,R2

MAINDEC-11-DZRKK-C
 DZRKKC.P11 T31

 MACY11 27(732) 16-SEP-76 16:00 PAGE 82
 CHECK THAT WRITE WAS DONE CORRECTLY

4251	012526	012703	032724	MOV	#OUTBUF,R3	
4252	012532	012705	177773	MOV	#-5,R5	
4253	012536	062702	177401	65: ADD	#177401,R2	
4254	012542	020213		CMP	R2,(R3);	WAS THE READ WORD SAME AS THE WORD
4255						THAT WAS SUPPOSE TO BE WRITTEN
4256	012544	001414		BEG	75	YES, BRANCH
4257						NO ERROR
4258	012546	010137	001162	MOV	R1,\$REGO	GET THE # OF WORD
4259	012552	062737	000401 001162	ADD	#401,\$REGO	THAT IS IN ERROR (EXAMPLE=1,2--376,377,400)
4260	012560	010237	001164	MOV	R2,\$REG1	GET EXPTD WORD (THAT WAS SUPPOSED TO
4261						BE WRITTEN)
4262	012564	011337	001166	MOV	(R3),\$REG2	GET WORD RECVD (THAT WAS READ BAK)
4263	012570	104055		ERROR	55	DID NOT READ BACK WORD THAT WAS SUPPOSED
4264						TO HAVE BEEN WRITTEN PREVIOUSLY. POSITION
4265						OF WORD IN ERROR IS AS INDICATED BY
4266						WORD # (\$REGO), SEC 0, CYL 0
4267	012572	005205		INC	R5	
4268	012574	001403		SEQ	TST32	EXIT
4269	012576	005723		75: TST	(R3)+	INCREMENT POINTER TO NXT WORD (THAT
4270						WAS READ BAK)
4271	012600	005201		INC	R1	HAVE U CHKD ALL 256 WORDS?
4272	012602	001355		BNE	65	IF NOT, LUP BAK & CHK THE NXT WORD
4273						IF YES, EXIT
4274						
4275						*****
4276						TEST 32 CHECK 'READ CHECK' FUNCTION - CYLINDER 0, SECTOR 0
4277						*THIS TEST CHECKS OUT THE BASIC 'READ CHECK' LOGIC, USING THE DATA BLOCK
4278						*'CYLINDER, SECTOR 0) WRITTEN IN A PREVIOUS TEST. HENCE THE TEST WHICH
4279						*WRITES THE DATA BLOCK SHOULD BE DONE PRIOR TO THIS TEST.
4280						*****
4281	012604	000004		TST32: SCOPE		
4282	012606	104412		CNT.RESET		GO, DO CONTROL RESET
4283						THIS IS A CALL FOR THE 'CNTRL-
4284						RESET' ROUTINE. A CONTROL RESET IS
4285						ISSUED AND AFTER A CERTAIN TIME
4286						IF THE 'CNTRL RDY' DOES NOT SET
4287						AN ERROR IS REPORTED. NOTE THAT
4288						THE PC IN ERROR MESSAGE IS THE
4289						PC WHERE 'CNT.RESET' IS LOCATED.
4290						THIS IS A VERY BASIC ERRS IF IT
4291						OCCURS GO BACK TO TEST 10
4292	012610	104420		TST.SIN		CHECK IF SIN IS SET, IF SET
4293						DO DRIVE RESET TO CLEAR IT
4294	012612	013701	002526	MOV	RKCS,R1	
4295	012616	013702	002530	MOV	RKWC,R2	
4296	012622	013703	002534	MOV	RKDA,R3	
4297	012626	013704	002532	MOV	RKBA,R4	
4298	012632	012737	052525 032724	MOV	#52525,OUTBUF	
4299	012640	012712	177400	MOV	#-400,R2	READ CHECK 256 WORDS
4300	012644	013713	002544	MOV	DRIVAD,R3	STARTING FROM CYL 0, SECTOR 0
4301	012650	012714	032724	MOV	#OUTBUF,R4	
4302	012654	012711	000013	MOV	#13,R1	READ CHECK, GO
4303						
4304	012660	105711		15: TSTB	R1	DID CNTRL RDY GET CLEARED AS GO WAS SET?
4305	012662	100003		BPL	25	YES, BRANCH
4306	012664	004737	020414	JSR	PC,GT3RG	GET RKCS, ER, DS

E07

MAINDEC-11-DZRKK-C
DZRKKC.P11 T32MACY11 27(732) 16-SEP-76 16:00 PAGE 83
CHECK 'READ CHECK' FUNCTION - CYLINDER 0, SECTOR 0

```

4307 012670 104030          ERROR 30          :CNTRL RDY DID NOT CLEAR AS GO
4308 012672 104411          2$: CHKCRDY          :GO CHECK IF CONTROL RDY IS SET
4309                                     :IF SO, SKIP THE EROR MESSAGE.
4310                                     :WAS SET TO 'READ CHECK'
4311 012674 104056          ERROR 56          :CNTRL RDY DID NOT SET ON DOING
4312                                     :'READ CHECK' FROM CYL 0, SEC 0
4313 012676 032711 140000          3$: BIT #140000,AR1 :DID 'ERR' OR 'HE' BIT SET?
4314 012702 001403          BEQ 4$          :NO, BRANCH
4315 012704 004737 020414          JSR PC,GT3RG :GO, GET RKCS,ER DS FOR ERROR MESSAGE
4316 012710 104057          ERROR 57          :'ERR' OR 'HE' BIT SET ON DOING
4317                                     :'READ CHECK' ON CYLINDER 0, SEC 0
4318 012712 032777 000002 167604 4$: BIT #2,ARKER :DID 'CSE' BIT SET IN RKER?
4319 012720 001404          SEQ 5$          :NO, BRANCH
4320 012722 017737 167576 001162 :MOV ARKER,$REG0 :GET RKER
4321 012730 104060          ERROR 60          :SOFT ERROR - CSE - ON DOING 'READ
4322                                     :CHECK' ON CYLINDER 0, SECTOR 0
4323                                     :U SHOULD HAVE GOT ERROR 102 ALSC
4324 012732 005712          5$: TST AR2          :DID WORD COUNT OVERFLOW TO 0?
4325 012734 001405          BEQ 6$          :YES, BRANCH
4326 012736 011237 001162          MOV AR2,$REG0 :GET RKWC
4327 012742 011137 001164          MOV AR1,$REG1 :GET RKCS
4328 012746 104061          ERROR 61          :WORD COUNT DID NOT OVERFLOW
4329                                     :ON DOING 'READ CHK' ON CYL 0, SEC 0
4330 012750 013702 002544          6$: MOV DRIVAD,R2 :RKDA SHOULD INCREMENT
4331 012754 005202          INC R2          :TO THIS AFTER 'RD CHK' IS DONE
4332 012756 020213          CMP R2,AR3      :DID RKDA INCREMENT CORRECTLY?
4333 012760 001405          BEQ 7$          :
4334 012762 010237 001162          MOV R2,$REG0 :GET EXPCTD RKDA
4335 012766 011337 001164          MOV AR3,$REG1 :GET RKDA RECVD
4336 012772 104062          ERROR 62          :RKDA DID NOT INCREMENT CORRECTLY
4337                                     : (BY 1) ON DOING 'READ CHK' ON
4338                                     :CYL 0, SEC 0
4339 012774 022714 032724          7$: CMP #OUTBUF,AR4 :DID RKBA GET CHANGED?
4340 013000 001406          BEQ 9$          :NO, BRANCH (RKBA WON'T CHANGE, NO NPR'S)
4341 013002 012737 032724 001162 :MOV #OUTBUF,$REG0 :GET EXPCTD RKBA
4342 013010 011437 001164          MOV AR4,$REG1 :GET RKBA RECVD
4343 013014 104063          ERROR 63          :RKBA CHANGED AFTER DOING 'READ CHK'
4344                                     :ON CYLINDER 0, SECTOR 0. SHOULD
4345                                     :NOT CHANGE, FOR, NO NPR'S.
4346 013016 022737 052525 032724 8$: CMP #52525,OUTBUF :'OUTBUF' SHOULD STILL CONTAIN THE
4347                                     :SAME WORD AS IT DID BEFORE 'RD CHK'
4348                                     :NOTE THAT AT THE BEGINING OF THIS TEST
4349                                     :52525 WAS WRITTEN INTO 'OUTBUF'
4350 013024 001412          BEQ TST33      :YES, EXIT
4351                                     :REPORT ERROR IF 'OUTBUF' CHANGED
4352 013026 012737 032724 001162 :MOV #OUTBUF,$REG0 :GET ADRES OF OUTBUF
4353 013034 012737 052525 001164 :MOV #52525,$REG1 :GET EXPCTD WORD IN 'OUTBUF'
4354 013042 013737 032724 001166 :MOV OUTBUF,$REG2 :GET WORD FOUND IN 'OUTBUF'
4355 013050 104064          ERROR 64          :AS MENTIONED ABOVE, IF 'WRITE' OF
4356                                     :256 WORD DATA BLOCK WAS DONE
4357                                     :CORRECTLY BEFORE, THEN THIS ERROR
4358                                     :COULD MEAN THAT AN NPR WAS DONE
4359                                     :ON 'READ CHECK'.
4360
4361 ;*****
4362 ;*TEST 33          CHECK THE 'WRITE CHECK' FUNCTION - ON CYLINDER 0, SECTOR 0

```

F07

MAINDEC-11-DZRKK-C
DZRKKC.P11 T33MACY11 27(732) 16-SEP-76 16:00 PAGE 84
CHECK THE 'WRITE CHECK' FUNCTION - ON CYLINDER 0, SECTOR 0

```

4363      : *THIS TEST CHECKS OUT THE BASIC 'WRITE CHECK' LOGIC, USING THE 256
4364      : *WORD DATA BLOCK (SECTOR 0, CYLINDER 0) WRITTEN IN A PREVIOUS
4365      : *TEST. THE BUFFER IN MEMORY, USED FOR COMPARISON OF DATA, IS THE
4366      : *ONE STARTING AT 'OUTBUF'. HENCE THE TEST WHICH WRITES THE
4367      : *256 WORD BLOCK ON THE DISK (AS WELL AS CREATING THE 256
4368      : *256 WORD MEMORY BUFFER) SHOULD BE DONE BEFORE THIS TEST.
4369      : *****
4370 013052 000004      TST33: SCOPE
4371 013054 104412      CNT.RESET
4372
4373      : GO, DO CONTROL RESET
4374      : THIS IS A CALL FOR THE 'CNTRL-
4375      : RESET' ROUTINE. A CONTROL RESET IS
4376      : ISSUED AND AFTER A CERTAIN TIME
4377      : IF THE 'CNTRL RDY' DOES NOT SET
4378      : AN ERROR IS REPORTED. NOTE THAT
4379      : THE PC IN ERROR MESSAGE IS THE
4380      : PC WHERE 'CNT.RESET' IS LOCATED.
4381      : THIS IS A VERY BASIC ERR& IF IT
4382      : OCCURS GO BACK TO TEST 10
4383      : CHECK IF SIN IS SET, IF SET
4384      : DO DRIVE RESET TO CLEAR IT
4385
4386      TST.SIN
4387
4388      1$: MOV      RKCS,R1
4389      MOV      #-400,R0
4390      MOV      #OUTBUF,R2
4391      MOV      #177777,R3
4392      ADD      #177401,R3
4393      MOV      R3,(R2)+
4394      INC      R0
4395      BNE      1$
4396      MOV      #-400,DRKWC
4397      MOV      #OUTBUF,DRKBA
4398      MOV      DRIVAD,DRKDA
4399      MOV      #7,DR1
4400      : WRITE CHECK 256 WORDS
4401      : STARTING AT THIS BUS ADRES
4402      : WITH THIS DISK DATA BLOCK (CYL 0, SEC 0)
4403      : WRITE CHECK, GO
4404
4405      2$: CLR      R0
4406      TSTB     DR1
4407      BPL      3$
4408      JSR      PC,GT3RG
4409      ERROR    30
4410      : GIVE SOME TIME
4411      : DID CNTRL RDY CLEAR AS GO WAS SET?
4412      : YES BRANCH
4413      : GET RKCS, ER, DS
4414      : CNTRL RDY DID NOT CLEAR AS GO WAS
4415      : SET TO DO WRITE CHECK
4416      : GO CHECK IF CONTROL RDY IS SET
4417      : IF SO, SKIP THE EROR MESSAGE.
4418      : CNTRL RDY DID NOT SET AFTER
4419      : COMPLETING WRITE CHECK ON
4420      : CYLINDER 0, SECTOR 0
4421      : DID HE OR ERR BIT SET
4422      : NO, BRANCH
4423      : GO GET RKCS ER DS FOR ERROR MESSAGE
4424      : HE OR ERR BIT SET ON DOING WRITE
4425      : CHK ON CYLINDER 0, SEC 0
4426      : DID WCE SET IN RKER?
4427      : NO, BRANCH
4428      : YES GET RKCS, ER, DS
4429      : WCE ON WRITE CHECK OF CYL 0, SEC 0
4430      : NOTE THAT IF A PREVIOUS TEST
4431      : & THEN COMPARED WITH MEMORY BUFFER
4432      : TO SEE IF IT WAS WRITTEN CORRECT WAS

```

MAINDEC-11-DZRKK-C
DZRKKC.P11 T33

MACY11 27(732) 16-SEP-76 16:00 PAGE 85
CHECK THE 'WRITE CHECK' FUNCTION - ON CYLINDER 0, SECTOR 0

[illegible]

H07

MAINDEC-11-DZRRK-C
DZRRKC.P11 T34MACY11 27(732) 16-SEP-76 16:00 PAGE 86
CHECK THAT IBA INHIBITS INCREMENTING OF RKBA

4475	013342	013701	002526		MOV	RKCS,R1	
4476	013346	012700	177400		MOV	#-400,R0	;SET UP COUNT FOR 256 WORDS
4477	013352	012702	032724		MOV	#OUTBUF,R2	
4478	013356	010203			MOV	R2,R3	
4479							
4480	013360	005023		1\$:	CLR	(R3)+	;CLEAR OUT THE 256
4481	013362	005200			INC	R0	;WORD MEMORY BUFFER STARTING
4482	013364	001375			BNE	1\$	;AT 'OUTBUF'
4483	013366	012777	177400	167134	MOV	#-400,ARKWC	;READ BACK 256 WORDS
4484	013374	010277	167132		MOV	R2,ARKBA	;INTO THIS BUS ADRES (IBA WILL B SET)
4485	013400	013777	002544	167126	MOV	DRIVAD,ARKDA	;FROM THIS DSK ADRES (SEC 0, CYL 0)
4486							;NOTE: SEC 0 HAS BEEN WRITTEN IN A
4487							;PREVIOUS TEST WITH A UNIQUE PATTERN
4488	013406	012711	004005		MOV	#4005,ARI	;READ, GO, IBA SET
4489							
4490	013412	005037	002556		CLR	COUNT	
4491	013416	105711		2\$:	TSTB	ARI	;DID CNTRL RDY SET?
4492	013420	100412			BMI	3\$	;YES, BRANCH
4493	013422	005237	002556		INC	COUNT	;WAITED LONG ENOUGH?
4494	013426	001373			BNE	2\$	;IF NOT, LUP BAK & WAIT
4495	013430	004737	020406		JSR	PC,GT4RG	;GO, GET RKCS, ER, DS, DA
4496	013434	013737	002544	001202	MOV	DRIVAD,\$REG10	;GET THE STARTING ADRES
4497	013442	104415			BRKDA4		;BREAK CONTENTS OF \$REG10
4498							;INTO DR #, CYL, SUR, SEC
4499	013444	104045			ERROR	45	;CNTRL RDY DID NOT SET AFTER DOING
4500							;READ
4501	013446	004737	020646	3\$:	JSR	PC,CHKHE	;CHECK IF 'ERR' OR 'HE' BIT IS SET,
4502							;IF YES, RETURN HERE.
4503	013452	104046			ERROR	46	;ERR BIT SET ON DOING READ FROM SEC 0,
4504							;CYL 0 (INDICATED IN <DSK-ADRES>)
4505							; 'RKDA' IN EROR MSGE GIVES THE
4506							;CONTENTS OF RKDA AT THE TIME OF EROR
4507							
4508	013454	020277	167052	4\$:	CMP	R2,ARKBA	;DID RKBA INCREMENT?
4509	013460	001406			BEQ	5\$	;OK IF NOT, BRANCH
4510	013462	010237	001162		MOV	R2,\$REG0	;GET EXPCTD RKBA
4511	013466	017737	167040	001164	MOV	ARKBA,\$REG1	;GET RKBA RECVD
4512	013474	104072			ERROR	72	;RKBA INCREMNTED WHEN IBA BIT WAS
4513							;SET, SHOULD NOT HAVE
4514	013476	032777	001000	167016	5\$:	BIT	#1000,ARKDS
4515	013504	001042			BNE	TST35	;IS \$IN SET?
4516	013506	012700	177400		MOV	#-400,R0	;IF YES, EXIT
4517	013512	022712	000377		CMP	#377,ARI	
4518							;CHECK THAT THE FIRST WORD IN
4519							'OUTBUF' IS 377 (LAST WORD OF SEC 0,
4520	013516	001411			BEQ	6\$	;CYL 0). NOTE THAT READ WAS DONE
4521	013520	012737	000377	001162	MOV	#377,\$REG0	;INTO THIS SAME WRD WITH IBA SET
4522	013526	011237	001164		MOV	(R2),\$REG1	;GET EXPCTD WORD (LAST WORD OF THE BUFFER
4523	013532	013737	002544	001166	MOV	DRIVAD,\$REG2	;GET WORD RECVD (LAST WRD FROM SEC 0)
4524							;DISK ADRES WHERE ERROR OCCURED
4525							(SEC 0, CYL 0 LAST WORD)
4526	013540	104044			ERROR	44	;DATA ERROR
4527							;THE FIRST WORD IN MEM BUFFER (OUTBUF)
4528							;SHOULD BE NON-ZERO & SHOULD CONTAIN
4529							;THE LAST WORD READ BACK FROM SEC 0
4530	013542	005722		6\$:	TST	(R2)+	;CYL 0, THIS DID NOT HAPPEN IF THE ERROR OCCURS
							;INCREMENT POINTER TO THE NXT WORD

MAINDEC-11-DZRKK-C
DZRKKC.P11 T34

MACY11 27(732) 16-SEP-76 16:00 PAGE 87
CHECK THAT IBA INHIBITS INCREMENTING OF RKBA

```

4531 013544 012705 177773      MOV    #-5,R5      ;ALLOW ONLY 5 MESSAGES FOR ERR 116
4532 013550 005200      7$: INC    R0      ;CHKD ALL 256 WORDS IN THE BUFFER?
4533 013552 001417      BEQ     TST35     ;YES, EXIT
4534 013554 005722      TST     (R2)+    ;IS THIS WORD 0?
4535 013556 001774      BEQ     7$      ;YES, LUP BAK & CHK THE NXT WORD?
4536 013560 005037 001164    CLR     $REG1    ;ERROR. GET EXPCTED WORD - 0
4537 013564 014237 001166    MOV     -(R2), $REG2 ;GET WORD THAT WAS FOUND IN THE BUFFER
4538 013570 010004      MOV     R0,R4
4539 013572 062704 000401    ADD     #401,R4
4540 013576 010437 001162    MOV     R4,$REG0
4541                                ;THIS 'WORD #' IN MEMORY BUFFER
4542 013602 104073      ERROR  73      ;SHOULD HAVE BEEN ZERO
4543                                ;THE 256 WORD BUFR (STARTING AT
4544                                ;OUTBUF) WAS CLEARED BEFORE READING
4545                                ;BAK SEC 0 INTO IT. SINCE THE IBA
4546                                ;BIT WAS SET DURING THE READ, ONLY
4547                                ;THE FIRST WORD OF (OUTBUF) SHOULD
4548                                ;HAVE CHANGED. THE REST OF THE WORDS
4549                                ;SHOULD BE STILL 0. IF THIS ERROR
4550                                ;OCCURS, 'WORD #' (OF THE BUFFER) AS
4551                                ;INDICATED IN THE EROR MESSAGE) GOT
4552                                ;CHANGED WHEN READ WAS DONE FROM
4553                                ;THE DISK, INDICATING THAT WITH IBA
4554                                ;SET X-FER WAS NOT DONE INTO THE
4555                                ;SAME MEMORY LOCATION. 'WORD #'
4556                                ;IS OCTAL & SPECIFIES THE POSITION
4557                                ;IN THE BUFFER (FIRST WORD IS 'WORD #' 1)
4558 013604 005205      INC     R5
4559 013606 001401      BEQ     TST35     ;;EXIT
4560 013610 000757      BR      7$
4561                                ;*****
4562                                ;*TEST 35 CHECK THAT RK11 INTERRUPTS WHEN IDE IS SET
4563                                ;*THIS TEST CHECKS IF RK11 INTERRUPTS TO ITS DESIGNATED VECTOR
4564                                ;*ADDRESS WHEN IDE BIT IS SET, WITH CONTROL READY SET & GO CLEAR.
4565                                ;* IT IS NORMALLY 220, UNLESS IT HAS BEEN CHANGED. IF IT HAS BEEN
4566                                ;*CHANGED RK11 WILL INTERRUPT TO 'RKVEC'. NOTE 'RKVEC' HAS
4567                                ;*TO BE SET UP BY THE USER.
4568                                ;*****
4569 013612 000004      TST35: SCOPE
4570 013614 104412      CNT.RESET
4571                                ;GO, DO CONTROL RESET
4572                                ;THIS IS A CALL FOR THE 'CNTRL-
4573                                ;RESET' ROUTINE. A CONTROL RESET IS
4574                                ;ISSUED AND AFTER A CERTAIN TIME
4575                                ;IF THE 'CNTRL RDY' DOES NOT SET
4576                                ;AN ERROR IS REPORTED. NOTE THAT
4577                                ;THE PC IN ERROR MESSAGE IS THE
4578                                ;PC WHERE 'CNT.RESET' IS LOCATED.
4579                                ;THIS IS A VERY BASIC ERRS IF IT
4580                                ;OCCURS GO BACK TO TEST 10
4581                                ;CHECK IF SIN IS SET, IF SET
4582                                ;DO DRIVE RESET TO CLEAR IT
4582 013620 012746 000340      MOV     #340,-(SP)
4583 013624 012746 013632      MOV     #64$,-(SP)
4584 013630 000002      RTI
4585 013632                                64$:
4586 013632 013701 002526      MOV     RKCS,R1

```

[illegible]

MAINDEC-11-DZRKK-C
DZRKKC.P11 T36

MACY11 27(732) 16-SEP-76 16:00 PAGE 90
CHECK THAT WITH IDE SET RK11 INTERRUPTS AFTER INITIATION & COMPLETION OF SEEK

4699	014144	017701	166352		6\$:	MOV	@RKDS,R1	;GET RKDS
4700	014150	042701	017777			BIC	#17777,R1	;MASK NON-ID BITS IN RKDS
4701	014154	020137	002544			CMP	R1,@RIVAD	;CORRECT ID BITS IN RKDS?
4702	014160	001414				BEQ	7\$	;YES, BRANCH
4703								
4704	014162	013746	002544			MOV	@RIVAD,-(SP)	;PUSH DRV ADRES ON THE STACK
4705	014166	004737	020612			JSR	PC,SHFTRT	;GO, SHIFT RIGHT DRV #
4706	014172	012637	001162			MOV	(SP)+,\$REGO	;GET EXPCTD DRV #
4707	014176	010146				MOV	R1,-(SP)	;PUSH ID BITS ON THE STACK
4708	014200	004737	020612			JSR	PC,SHFTRT	;GO SHIFT THEM RIGHT
4709	014204	012637	001164			MOV	(SP)+,\$REG1	;POP THE RECVD ID BITS
4710	014210	104047				ERROR	47	;WRONG ID BITS WERE RECVD IN
4711								;RKDS AFTER SEEK WAS DONE (INTRUPT
4712								;MODE). 'EXPCT' INDICATES THE DRIVE
4713								;# THAT SHOULD HAVE BEEN IN THE
4714								;ID BITS, 'RECVD' INDICATES THE
4715								;DRIVE # THAT WAS RECVD IN THE ID BITS
4716								
4717	014212				7\$:	MOV	#340,-(SP)	
4718	014212	012746	000340			MOV	#64\$,-(SP)	
4719	014216	012746	014224			RTI		
4720	014222	000002			64\$:			
4721	014224					CNT.RESET		;GO DO CONTROL RESET
4722	014224	104412				MOV	@RIVAD,@RKDA	;ADRES THE DRIVE
4723	014226	013777	002544	166300		BIT	#160000,@RKDS	;DID CNTRL RESET CLEAR DRIVE ID BITS?
4724	014234	032777	160000	166260		BEQ	8\$	;YES, BRANCH
4725	014242	001404				MOV	@RKDS,\$REGO	;GET RKDS
4726	014244	017737	166252	001162		ERROR	50	;CONTROL RESET DIDN'T CLEAR THE
4727	014252	104050						;DRIVE ID BITS (13-15) IN RKDS
4728								
4729								
4730					8\$:	CMP	#200,@RO	;WAS SCP BIT CLEARED BY CNTRL RESET?
4731	014254	022710	000200			BEQ	TST37	;YES, EXIT
4732	014260	001403				MOV	@RO,\$REGO	;GET RKCS
4733	014262	011037	001162			ERROR	100	;CNTRL RESET DID NOT CLEAR SCP BIT
4734	014266	104100						
4735								
4736								
4737								
4738								
4739								
4740								
4741								
4742	014270	000004						
4743	014272	104412						
4744								
4745								
4746								
4747								
4748								
4749								
4750								
4751								
4752								
4753	014274	104420						
4754								

M07

MAINDEC-11-DZRKK-C
DZRKKC.P11 T3

MACY11 27(732) 16-SEP-76 16:00 PAGE 91
CHECK THAT WITH IDE SET RK11 INTERRUPTS WHEN READ IS DONE

```

4755
4756 014276 013700 002526      MOV      RKCS,R0
4757 014302 013702 002534      MOV      RKDA,R2
4758 014306 013704 002532      MOV      RKBA,R4
4759 014312 013701 002544      MOV      DRIVAD,R1
4760 014316 052701 000013      BIS      #13,R1      ;SET BITS FOR SEC 13
4761 014322 012777 177600 166200      MOV      #-200,DRKWC      ;READ 200 (OCTAL WORDS)
4762 014330 010112      MOV      R1,DR2      ;FROM THIS DISK ADRES (CYL 0, SEC 13)
4763 014332 012714 032724      MOV      #OUTBUF,DR4      ;INTO THIS BUS ADRES
4764 014336 013705 002576      MOV      RKVEC,R5
4765 014342 012725 014400      MOV      #15,(R5)+      ;SET UP VECTOR ADRES FOR RK11 TO INTRUPT
4766 014346 012715 000340      MOV      #340,(R5)      ;SET PSW ON INTERUPT
4767 014352 012710 000105      MOV      #105,DR0      ;READ, GO, IDE SET
4768 014356 104417 127710      WAT.INT ,127710      ;WAIT FOR RK11 TO INTERRUPT ON
4769                                     ;COMPLETION OF READ
4770                                     ;WAITING TIME= 337 MS FOR 11/20
4771                                     ;67 MS FOR 11/45
4772 014252 012777 004270 166206      MOV      #BADINT,DRKVEC      ;RESTORE UNEXPTED INTERRUPT VECTOR ADRES
4773 014370 011037 001162      MOV      DR0,$REG0      ;GET RKCS
4774 014374 104101      ERROR      101      ;RK11 DID NOT INTERRUPT AFTER READ
4775                                     ;WAS DONE, IDE BIT SET.
4776 014376 000404      BR      1$+10
4777 014400 022626      1$: CMP      (SP)+,(SP)+      ;OK, IF RK11 INTERRUPTED TO THIS
4778                                     ;RESTORE STACK POINTER (FROM RK11 INTERRUPT)
4779 014402 022626      CMP      (SP)+,(SP)+      ;RESTORE STACK POINTER (FROM WAT.INT)
4780 014404 012777 004270 166164      MOV      #BADINT,DRKVEC      ;RESTORE UNEXPECTED RK11 INTERRUPT
4781                                     ;VECTOR ADRES
4782 014412 004737 020754      JSR      PC,CHKER      ;CHECK IF ANY BIT IN RKER IS SET,
4783                                     ;IF YES, RETURN HERE.
4784 014416 104036      ERROR      36      ;RKER SET ON DOING READ FROM SEC 0,
4785                                     ;CYL 13 IN INTERRUPT MODE
4786 014420 062701 000005      4$: ADD      #5,R1      ;RKDA SHOULD HAVE INCREMENTED TO THIS
4787 014424 020112      CMP      R1,DR2      ;DID RKDA INCREMENT CORRECTLY?
4788 014426 001405      BEQ      2$      ;YES BRANCH
4789 014430 010137 001162      MOV      R1,$REG0      ;GET EXPCTD RTDA
4790 014434 011237 001164      MOV      DR2,$REG1      ;GET RKDA RECVD
4791 014440 104040      ERROR      40      ;RKDA INCREMENTED WRONG ON DOING
4792                                     ;A READ ON CYL 0, SEC 13
4793 014442 004737 020730      2$: JSR      PC,CHKWC      ;CHECK THAT RKWC OVERFLOWED TO 0,
4794                                     ;IF NOT RETURN HERE.
4795 014446 104041      ERROR      41      ;RKWC DIDN'T OUFLO AFTER
4796                                     ;A READ OF 200 WORDS
4797
4798 014450      3$:
4799 014450 012746 000340      MOV      #340,-(SP)
4800 014454 012746 014462      MOV      #64$,-(SP)
4801 014460 000002      RTI
4802 014462      64$:
4803 014462 022714 033324      CMP      #OUTBUF+400,DR4 ;DID RKBA INCREMENT CORRECTLY?
4804 014466 001406      BEQ      TST40      ;YES, EXIT
4805 014470 012737 033324 001162      MOV      #OUTBUF+400,$REG0 ;GET EXPCT RKBA
4806 014476 011437 001164      MOV      DR4,$REG1      ;GET RKBA RECVD
4807 014502 104042      ERROR      42      ;RKBA DID NOT INCREMENT CORRECTLY
4808                                     ;AFTER A READ OF 200 WORDS
4809
4810                                     ;*****

```

MAINDEC-11-DZRKK-C
DZRKKC.P11 T40

MACY11 27(732) 16-SEP-76 16:00 PAGE 92
CHECK THAT RK11 INTERRUPTS AT BRS ONLY

4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866

014504 000004
014506 104412

014510 104420

014512 012737 014546 001110

014520 013700 002526
014524 013777 002544 166002
014532 012701 000007
014536 012702 000340
014542 013703 002574

014546 013704 002576
014552 012724 014660
014556 012714 000340
014562 010246
014564 012746 014572
014570 000002
014572
014572 012710 000100
014576 012705 177760
014602 005205
014604 001376
014606 020203
014610 003005

014612 010137 001162
014616 011037 001164
014622 104103

014624 005010
014626 062702 177740
014632 005301

:*TEST 40 CHECK THAT RK11 INTERRUPTS AT BRS ONLY
:*THIS TEST CHECKS THAT RK11 CAN INTERRUPT AT BRS ONLY. IF IT
:*INTERRUPTS AT A LEVEL HIGHER THAN BRS AN ERROR IS INDICATED.
:*IF IT DOES NOT INTERRUPT AT BRS OR LOWER THEN ALSO AN
:*ERROR IS INDICATED. IF FOR SOME REASON THE INTERRUPT
:*LEVEL IS CHANGED FROM BRS, THEN CONTENTS OF RKPRI WILL
:*HAVE TO BE CHANGED ACCORDINGLY AND STILL TEXT WILL
:*CHECK FOR THIS BR LEVEL.

::*****
TST40: SCOPE
CNT.RESET

TST.SIN

MOV #1\$, \$LPERR
MOV RKCS, R0
MOV DRIVAD, DRKDF
MOV #7, R1
MOV #340, R2
MOV RKPRI, R3

1\$: MOV RKVEC, R4
MOV #3\$, (R4)+
MOV #340, (R4)
MOV R2, -(SP)
MOV #4\$, -(SP)
RTI

4\$: MOV #100, DR0
MOV #-20, R5
INC R5
BNE -2
CMP R2, R3
BGT 2\$

MOV R1, \$REG0
MOV DR0, \$REG1
ERROR 103

2\$: CLR DR0
ADD #-40, R2
DEC R1

;GO, DO CONTROL RESET
;THIS IS A CALL FOR THE 'CNTRL
;RESET' ROUTINE. A CONTROL RESET IS
;ISSUED AND AFTER A CERTAIN TIME
;IF THE 'CNTRL R0Y' DOES NOT SET
;AN ERROR IS REPORTED. NOTE THAT
;THE PC IN ERROR MESSAGE IS THE
;PC WHERE 'CNT.RESET' IS LOCATED.
;THIS IS A VERY BASIC ERR& IF IT
;OCCURS GO BACK TO TEST 10
;CHECK IF SIN IS SET, IF SET
;DO DRIVE RESET TO CLEAR IT
;SET RETURN ADRES FOR LUPING
;ON ERROR (SW 9)

;PRIORITY LEVEL 7
;BR LEVEL 7 FOR PSW
;NOTE, IF RK11 INTERRUPT LEVEL IS
;CHANGED FROM 5 TO ANY OTHER LEVEL
;THEN CHANGE CONTENTS OF 'RKPRI'
; ACCORDINGLY

;SET UP ADRES FOR RK11 TO INTERRUPT
;SET UP PSW ON INTERRUPT
;SET PROCESSOR PRIORITY LEVEL AS

;INDICATED BY R2
;SET THE IDE BIT
;WAIT FOR THE RK11 INTERRUPT
;WAITING TIME=78 US FOR 11/20
;13 US FOR 11/45
;WAS THE CPU PRIORITY LEVEL LESS THAN
;THE RK11 LEVEL? IF YES, RK11
;SHOULD HAVE INTERRUPTED. ERROR,
;IF IT DID NOT
;GET CPU BR LEVEL
;GET RKCS
;THOUGH CPU LEVEL WAS LESS THAN
;THE RK11 LEVEL (5), RK11 DID NOT
;INTERRUPT
;CLEAR RKCS
;DECREASE THE PRIORITY LEVEL (FOR
;CPU) BY 1
;CPU WILL B AT THIS LEVEL

MAINDEC-11-DZK-K-C
 DZK-K-C.P11 T40

 MACY11 27(732) 16-SEP-76 16:00 PAGE 93
 CHECK THAT RK11 INTERRUPTS AT BR5 ONLY

```

4967 014634 001344      BNE      1$      ;LUP BAK & CHK FOR THIS BR LEVEL.
4968                                ;DONE WITH CHKING FOR ALL LEVELS.
4969 014636 012777 004270 165732      MOV      #BADINT,ARKVEC ;RESTORE UNEXPECTED RK11 INTERRUPT
4970                                ;VECTOR
4971 014644 012746 000340      MOV      #340,-(SP)
4972 014650 012746 014656      MOV      #643,-(SP)
4973 014654 000002      RTI
4974 014656                                643:
4975 014656 000414      BR        TST41      ;;EXIT, TO NXT TST
4976
4977 014660 022626      3$:      CMP      (SP)+,(SP)+ ;RESTORE STACK POINTER
4978 014662 012777 004270 165706      MOV      #BADINT,ARKVEC ;RESTORE UNEXPECTED RK11 INTERRUPT
4979                                ;VECTOR
4980 014670 020203      CMP      R2,R3      ;IF THIS INTERRUPT OCCURED WHEN
4981 014672 003754      BLE      2$      ;CPU LEVEL WAS LESS THAN THE
4982                                ;RK11 PRIORITY LEVEL (5) THEN IT IS
4983                                ;OK. IF NOT SO, ERROR
4984 014674 010137 001162      MOV      R1,SREG0 ;GET CPU BR LEVEL
4985 014700 011037 001164      MOV      AR0,SREG1 ;GET RKCS
4986 014704 104104      ERROR      104 ;RK11 INTERRUPTED WHEN THE CPU
4987                                ;LEVEL (AS POINTED BY R1) WAS
4988                                ;HIGHER OR SAME AS THE RK11
4989                                ;LEVEL (5)
4990 014706 000746      BR        2$      ;GO BACK & CHK THE NXT LEVEL
4991
4992                                ;*****
4993                                ;*TEST 41 SIMULATE & CHECK 'OVR' ERROR
4994                                ;*THIS TEST SIMULATES OVERRUN ERROR AND CHECKS IF THE OVR
4995                                ;*BIT IN RKER GETS SET. THEN IT IS CLEARED USING CNTRL RESET
4996                                ;*& CHECKED THAT IT WAS CLEARED. OVR CONDITION IS SIMULATED
4997                                ;*BY TRYING TO READ 401(OCTAL) WORDS FROM LAST CYLINDER(312).
4998                                ;*LAST SECTOR (13), SURFACE 1.
4999                                ;*****
5000 014710 000004      TST41: SCOPE
5001 014712 104412      CNT.RESET ;GO, DO CONTROL RESET
5002                                ;THIS IS A CALL FOR THE 'CNTRL-
5003                                ;RESET' ROUTINE. A CONTROL RESET IS
5004                                ;ISSUED AND AFTER A CERTAIN TIME
5005                                ;IF THE 'CNTRL RDY' DOES NOT SET
5006                                ;AN ERROR IS REPORTED. NOTE THAT
5007                                ;THE PC IN ERROR MESSAGE IS THE
5008                                ;PC WHERE 'CNT.RESET' IS LOCATED.
5009                                ;THIS IS A VERY BASIC ERR& IF IT
5010                                ;OCCURS GO BACK TO TEST 10
5011 014714 104420      TST.SIN ;CHECK IF SIN IS SET, IF
5012                                ;SET, DO DRIVE RESET TO CLR IT
5013 014716 013701 002544      MOV      DRIVAD,R1 ;GET ADRES OF DRIVE
5014 014722 052701 014533      BIS      #14533,R1 ;SET BITS FOR LAST CYLINDER (312).
5015                                ;SUR 1, LAST SECTOR (13)
5016 014726 012777 177377 165574      MOV      #-401,ARKWC ;READ 401 WORDS
5017 014734 012777 032724 165570      MOV      #OUTBUF,ARKBA ;INTO THIS MEMORY BUFFER
5018 014742 010177 165566      MOV      R1,ARKDA ;FROM THIS DSK ADRES, LAST CYL,
5019                                ;LAST SEC, SURFACE 1
5020 014746 012777 000005 165552      MOV      #5,ARKCS ;READ, GO
5021
5022 014754 005002      CLR      R2

```

MAINDEC-11-DZRKK-C
DZRKKC.P11 T41MACY11 27(732) 16-SEP-76 16:00 PAGE 94
SIMULATE & CHECK 'OVR' ERROR

```

4923 014756 105777 165544 1$: TSTB 2RKCS ;DID CNTRL RDY SET?
4924 014752 100410 BMI 2$ ;YES, BRANCH
4925 014764 005202 INC R2 ;NO, WAIT FOR IT
4926 014766 001373 BNE 1$ ;IF WAITED LONG, REPORT ERROR MESSAGE BECAUSE
;OVR SHOULD HAVE SET HE CAUSING
;CNTRL RDY TO SET BY NOW
4928
4929 014770 017737 165534 001166 MOV 2RKWC,$REG2 ;GO, GET RKCS, ER
4930 014776 004737 020422 JSR PC,GT2RG ;CNTRL RDY DID NOT SET AFTER DOING
4931 015002 104002 ERROR 2 ;AN OVR READ. HE SHOULD HAVE OCCURED
;SETTING CNTRL RDY (HE BECAUSE OF
;OVR CONDITIONS)
;DID OVR BIT SET IN RKER?
4932
4933
4934
4935 015004 032777 040000 165512 2$: BIT #40000,2RKER ;GET RKCS, ER
4936 015012 001006 BNE 3$ ;THIS BIT (OVR) DID NOT SET.
4937 015014 004737 020422 JSR PC,GT2RG ;OVR ERROR BIT DID NOT SET IN RKER
4938 015020 012737 040000 001166 MOV #40000,$REG2 ;ON SIMULATING OVR CONDITIONS
4939 015026 104105 ERROR 10$ ;DID HE ERR SET WHEN OVR SET IN RKER?
4940
4941 015030 022777 140204 165470 3$: CMP #140204,2RKCS ;YES, BRANCH
4942 015036 001403 BEQ 4$ ;GET RKCS, ER
4943 015040 004737 020422 JSR PC,GT2RG ;HE OR ERR BIT DID NOT SET IN RKCS WHEN
4944 015044 104106 ERROR 10$ ;AN OVR ERROR WAS SIMULATED
;CLEAR OVR, ERR, HE BITS
;GO, DO CONTROL RESET
4945
4946
4947 015046 104412 4$: CNT.RESET ;THIS IS A CALL FOR THE 'CNTRL-
;RESET' ROUTINE. A CONTROL RESET IS
;ISSUED AND AFTER A CERTAIN TIME
;IF THE 'CNTRL RDY' DOES NOT SET
;AN ERROR IS REPORTED. NOTE THAT
;THE PC IN ERROR MESSAGE IS THE
;PC WHERE 'CNT.RESET' IS LOCATED.
;THIS IS A VERY BASIC ERR& IF IT
;OCCURS GO BACK TO TEST 10
;CHECK IF 'OVR' BIT WAS CLEARED BY
;CON.RESET, IF NOT RETURN HERE.
;CNTRL RESET DID NOT CLEAR OVR
;BIT IN RKER
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957 015050 004737 020770 JSR PC,CHKECLR ;CHECK IF 'ERR' & 'HE' BIT GOT CLEARED BY
4958
4959 015054 104102 ERROR 102 ;CON.RESET, IF NOT RETURN HERE.
4960
4961 015056 004737 021014 5$: JSR PC,CHKCCLR ;CNTRL RESET DID NOT CLEAR OVR
4962
4963 015062 104102 ERROR 102 ;BIT IN RKER
4964
4965 015064 004737 021115 6$: JSR PC,DRESET ;CHECK IF 'ERR' & 'HE' BIT GOT CLEARED BY
4966 015070 104026 ERROR 2$ ;CON.RESET, IF NOT RETURN HERE.
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977 015072 000004 ;GO DO DRIVE RESET
4978 015074 104412 CNT.RESET ;R/W'S RDY DIDN'T SET
;AFTER THE ABOVE DRIVE RESET
;*****
;*TEST 42 SIMULATE & CHECK PGE ERROR
;*THIS TEST SIMULATES 'PROGRAMMING ERROR' & CHECKS IF IT IS
;*DETECTED BY PGE BIT IN RKER. THEN A CNTRL RESET IS DONE &
;*IT IS CHECKED IF PGE BIT WAS CLEARED. IT IS ALSO CHECKED IF
;*THE SETTING & CLEARING OF PGE BIT SETS & CLEARS HE, ERR
;*BITS IN RKCS.
;*****
;*ST42: SCOPE
CNT.RESET ;GO, DO CONTROL RESET

```

```

4979
4980
4981
4982
4983
4984
4985
4986
4987
4988 015076 104420 TST.SIN
4989
4990 015100 013701 002524 MOV RKR,R1
4991 015104 013777 002544 165422 MOV DRIVAD,DRKDA
4992
4993 015112 012777 002011 165406 MOV #2011,DRKCS
4994
4995 015120 104413 CNT.RDY
4996
4997
4998
4999
5000
5001 015122 032711 004000 BIT #4000,DR1
5002 015126 001006 BNE 1$
5003 015130 012737 004000 001166 MOV #4000,$REG2
5004 015136 004737 020422 JSR PC,GT2RG
5005 015142 104105 ERROR 10$
5006
5007
5008
5009 015144 022777 142210 165354 1$: CMP #142210,DRKCS
5010 015152 001403 BEQ 2$
5011 015154 004737 020422 JSR PC,GT2RG
5012 015160 104106 ERROR 10$
5013
5014
5015 015162 104412 2$: CNT.RESET
5016
5017
5018
5019
5020
5021
5022
5023
5024
5025 015164 004737 020770 JSR PC,CHKECLR
5026
5027 015170 104102 ERROR 102
5028
5029 015172 004737 021014 3$: JSR PC,CHKCCLR
5030
5031 015176 104102 ERROR 102
5032
5033
5034

```

```

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

```

```

;ADRES THE DRIVE, CYLINDER 0

```

```

;SEEK, GO WITH FMT SET
;THIS IS A PGE SIMULATION
;THIS IS A CALL FOR 'CN.RDY'
;ROUTINE WHICH WAITS FOR CNT
;RDY TO SET. IF CNTRL RDY DOES
;NOT SET WITHIN 883 MS/ 11-20
;(176 MS FOR 11-45 WITH BIPOLAR)
;AN ERROR IS REPORTED
;DID PGE BIT IN RKR SET?
;YES, BRANCH

```

```

;THIS BIT IN RKR (PGE) DID NOT SET
;GO GET RKCS, ER FOR MESSAGE
;PGE BIT DID NOT SET IN RKR
;ON SIMULATION OF PGE CONDITION
;$REG2 CONTAINS THE RKR BIT (PGE)
;THAT SHOULD HAVE SET.
;DID HE & ERR BITS SET?
;YES, BRANCH

```

```

;GO, GET RKCS, ER
;HE OR ERR BIT DID NOT SET WHEN
;PGE SET IN RKR.
;CLEAR PGE, HE, ERR BITS

```

```

;GO, DO CONTROL RESET

```

```

;THIS IS A CALL FOR THE 'CNTRL-
;RESET' ROUTINE. A CONTROL RESET IS
;ISSUED AND AFTER A CERTAIN TIME
;IF THE 'CNTRL RDY' DOES NOT SET
;AN ERROR IS REPORTED. NOTE THAT
;THE PC IN ERROR MESSAGE IS THE
;PC WHERE 'CNT.RESET' IS LOCATED.

```

```

;THIS IS A VERY BASIC ERR& IF IT
;OCCURS GO BACK TO TEST 10
;CHECK IF 'PGE' BIT GOT CLEARED BY
;CONTROL RESET, IF NOT RETURN HERE.
;CNTRL RESET DID NOT CLEAR

```

```

;PGE BIT IN RKR
;CHECK IF 'ERR' BIT GOT CLEARED BY
;CON.RESET, IF NOT RETURN HERE.
;RKCS BITS HE OR ERR DID NOT
;GET CLEARED BY CNTRL RESET

```

```

;;*****

```

E08

MAINDEC-11-DZRKK-C
DZRKKC.P11 T43MACY11 27(732) 16-SEP-76 16:00 PAGE 96
SIMULATE & CHECK NXM ERROR5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073
5074
5075
5076
5077
5078
5079
5080
5081
5082
5083
5084
5085
5086
5087
5088
5089
5090

:*TEST 43

SIMULATE & CHECK NXM ERROR

;*THIS TEST SIMULATES A NON-EXISTENT MEMORY ERROR (NXM) AND
 ;*CHECKS IF IT IS DETECTED BY NXM BIT OR RKER LOCATION 760000
 ;*IS REFERENCED & IT HAPPENS TO BE A NON EXISTENT LOCATION
 ;*(FOR DIAGNOSTIC PURPOSES LIKE THIS). IT IS ALSO CHECKED
 ;*IF HE & ERR BITS ALSO SET AND ALL 3 BITS CAN BE CLEARED
 ;* BY CONTROL RESET.

;*****
 TST43: SCOPE
 CNT.RESET

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

TST.SIN

CLR R2
 MOV RKCS, R0
 MOV #-1, ARKWC
 MOV #160000, ARKBA
 MOV DRIVAD, ARKDA
 MOV #67, AR0
 TSTB ARKCS
 BMI 2\$
 INC R2
 BNE 1\$
 JSR PC, GT2RG
 MOV ARKWC, \$REG2
 ERROR 113

;WRITE CHECK 1 WORD
 ;AT THIS BUS ADRES
 ;WITH THIS DISK ADRES (CYL 0, SEC 0)
 ;WRT CHK, GO, MEX BITS SET
 ;DID CNTRL RDY SET AS A RESULT OF HE?
 ;YES, BRANCH
 ;WAITED LONG ENOUGH?
 ;IF NOT LUP BAK & WAIT
 ;GET RKCS, ER
 ;GET RKWC
 ;CNTRL RDY DID NOT SET ON DOING
 ;A WRT CHK WITH A NXM LOCATION.
 ;THIS HE SHOULD HAVE SET THE
 ;CNTRL RDY BIT IN RKCS
 ;DID NXM BIT IN RKER SET?
 ;YES, BRANCH
 ;GO GET RKCS, RKER
 ;THIS BIT (NXM) DID NOT SET IN RKER
 ;NXM BIT DID NOT SET IN RKER ON
 ;SIMULATING NXM CONDITION.
 ;DID HE & ERR BIT SET?
 ;YES, BRANCH

1\$:

2\$:

3\$:

4\$:

BIT #2000, ARKER
 BNE 3\$
 JSR PC, GT2RG
 MOV #2000, \$REG2
 ERROR 105
 CMP #140266, AR0
 BEQ 4\$
 JSR PC, GT2RG
 ERROR 106

;GO, GET RKCS, RKER
 ;HE OR ERR BIT DID NOT SET WHEN
 ;NXM ERROR WAS SIMULATED
 ;CLEAR NXM, HE, ERR BITS
 ;GO, DO CONTROL RESET
 ;THIS IS A CALL FOR THE 'CNTRL-
 ;RESET' ROUTINE. A CONTROL RESET IS
 ;ISSUED AND AFTER A CERTAIN TIME
 ;IF THE 'CNTRL RDY' DOES NOT SET
 ;AN ERROR IS REPORTED. NOTE THAT
 ;THE PC IN ERROR MESSAGE IS THE

015200 000004
 015202 104412

015204 104420

015206 005002
 015210 013700 002526
 015214 012777 177777 165306
 015222 012777 160000 165302
 015230 013777 002544 165276
 015236 012710 000067
 015242 105777 165260
 015246 100410
 015250 005202
 015252 001373
 015254 004737 020422
 015260 017737 165244 001166
 015266 104113

015270 032777 002000 165226 2\$:
 015276 001006
 015300 004737 020422
 015304 012737 002000 001166
 015312 104105

015314 022710 140266 3\$:
 015320 001403
 015322 004737 020422
 015326 104106

015330 104412

CNT.RESET


```

5091                                     ;PC WHERE 'CNT.RESET' IS LOCATED.
5092                                     ;THIS IS A VERY BASIC ERR& IF IT
5093                                     ;OCCURS GO BACK TO TEST 10
5094 015332 004737 020770                JSR    PC,CHKECLR      ;CHECK IF 'NXM' BIT GOT C;LEARED BY
5095                                     ;CON.RESET, IF NOT RETURN HERE.
5096 015336 104102                        ERROR 102           ;CNTRL RESET DID NOT CLEAR
5097                                     ;NXM BIT IN RKER
5098 015340 004737 021014                5$: JSR    PC,CHKCCLR   ;CHECK IF 'HE' & 'ERR' BITS GOT CLEAPED
5099                                     ;BY CON.RESET, IF NOT RETURN HERE.
5100 015344 104102                        ERROR 102           ;CNTRL RESET DID NOT CLEAR
5101                                     ;HE OR ERR BIT IN RKCS.
5102 015346 004737 021050                6$: JSR    PC,TSTRWS   ;GO CHECK IF R/W/S RDY IS SET &
5103                                     ;WAIT FOR IT. SKIP ERROR IF IT IS SET
5104 015352 104016                        ERROR 16            ;R/W/S RDY IS NOT SET
5105
5106                                     ;*****
5107                                     ;*TEST 44 SIMULATE & CHECK NXD ERROR
5108                                     ;*THIS TEST SIMULATES NON-EXISTENT DISK ERROR & CHECKS IF
5109                                     ;*IT IS DETECTED BY NXD BIT OF RKER. IF ALL EIGHT ARE PRESENT
5110                                     ;*THEN THIS TEST IS ABORTED FOR SIMULATION CANNOT BE DONE.
5111                                     ;*****
5112 015354 000004                        †ST44: SCOPE
5113 015356 104412                        CNT.RESET
5114
5115                                     ;GO, DO CONTROL RESET
5116                                     ;THIS IS A CALL FOR THE 'CNTRL-
5117                                     ;RESET' ROUTINE. A CONTROL RESET IS
5118                                     ;ISSUED AND AFTER A CERTAIN TIME
5119                                     ;IF THE 'CNTRL RDY' DOES NOT SET
5120                                     ;AN ERROR IS REPORTED. NOTE THAT
5121                                     ;THE PC IN ERROR MESSAGE IS THE
5122                                     ;PC WHERE 'CNT.RESET' IS LOCATED.
5123 015360 104420                        TST.SIN              ;THIS IS A VERY BASIC ERR& IF IT
5124                                     ;OCCURS GO BACK TO TEST 10
5125 015362 013700 002526                MOV     RKCS,R0
5126 015366 012702 160000                MOV     #160000,R2    ;DO DRV RESET TO CLR IT
5127                                     ;ADRES DRIVE 7 TO FIND
5128 015372 010277 165136                1$: MOV     R2,ARKDA    ;IF IT IS PRESENT
5129 015376 104416 000001                DELAY    ,1           ;ADRES DRIVE # POINTED TO BY R2
5130                                     ;TIME DELAY, 7.5 US ON 11/20,
5131 015402 105777 165114                TSTB     ARKDS         ;1.5 US ON 11/45
5132 015406 100004                        BPL      2$           ;IS IT PRESENT?
5133 015410 062702 160000                ADD     #-20000,R2    ;NO, BRANCH
5134                                     ;ADRES THE NXT DRIVE IN THE
5135 015414 001366                        BNE      1$           ;REVERSE ORDER. I.E. 7,6,...
5136                                     ;LUP BAK & TRY TO FIND A DRIVE
5137 015416 000435                        BR       TST45        ;THAT'S NOT PRESENT
5138                                     ;EXIT TO THE NXT TST
5139 015420 012710 000015                2$: MOV     #15,ARO    ;DRIVE RESET, ON A NX DRIVE
5140 015424 104416 000106                DELAY    ,106        ;TIME DELAY, 525 US ON 11/20
5141                                     ;105 US ON 11/45
5142 015430 105777 165070                TSTB     ARKER        ;DID NXD BIT IN RKER SET?
5143 015434 001006                        BNE      3$           ;YES, BRANCH
5144 015436 004737 020422                JSR     PC,GT2RG      ;GET 'RKCS', RKER
5145 015442 012737 000200 001166        MOV     #200,$REG2    ;THIS BIT (NXD) IN RKER DID NOT SET
5146 015450 104105                        ERROR 105          ;NXD BIT DID NOT SET ON TRYING

```

5147
 5148
 5149
 5150
 5151
 5152
 5153
 5154
 5155
 5156
 5157
 5158
 5159
 5160
 5161
 5162
 5163
 5164
 5165
 5166
 5167
 5168
 5169
 5170
 5171
 5172
 5173
 5174
 5175
 5176
 5177
 5178
 5179
 5180
 5181
 5182
 5183
 5184
 5185
 5186
 5187
 5188
 5189
 5190
 5191
 5192
 5193
 5194
 5195
 5196
 5197
 5198
 5199
 5200
 5201
 5202

015452 022710 140214
 015456 001403
 015460 004737 020422
 015464 104106
 015466 104412
 015470 004737 020770
 015474 104102
 015476 004737 021014
 015502 104102
 015504 004737 021050
 015510 104016

3\$: CMP #140214,ARO
 SEQ 4\$
 JSR PC,GT2RG
 ERROR 106
 4\$: CNT.RESET
 JSR PC,CHKECLR
 ERROR 102
 5\$: JSR PC,CHKCCLR
 ERROR 102
 JSR PC,TSTRWS
 ERROR 16

; TO PERFORM A FUNCTION ON A
 ; NON-EXISTENT DRIVE
 ; CHECK THAT THE JUMPER CARD CONTAINING
 ; JUMPERS FOR DRIVES PRESENT IS PROPERLY
 ; CONNECTED
 ; NOTE THAT ON RK11C IF A DRIVE
 ; IS OFFLINE BUT PHYSICALLY PRESENT
 ; (IE. DRY IS CLR FOR THAT DRIVE)
 ; & A FUNCTION IS INITIATED ON THAT
 ; DRIVE NXD WON'T SET, BUT U WILL
 ; GET ONLY A DRE, HE & ERR.
 ; DID HE & ERR SET WHEN NXD SET?
 ; YES BRANCH
 ; HE OR ERR BIT DID NOT SET
 ; WHEN NXD WAS SIMULATED
 ; CLEAR NXD, HE, ERR BITS
 ; GO, DO CONTROL RESET
 ; THIS IS A CALL FOR THE 'CNTRL-
 ; RESET' ROUTINE. A CONTROL RESET IS
 ; ISSUED AND AFTER A CERTAIN TIME
 ; IF THE 'CNTRL RDY' DOES NOT SET
 ; AN ERROR IS REPORTED. NOTE THAT
 ; THE PC IN ERROR MESSAGE IS THE
 ; PC WHERE 'CNT.RESET' IS LOCATED.
 ; THIS IS A VERY BASIC ERR& IF IT
 ; OCCURS GO BACK TO TEST 10
 ; CHECK IF 'NXD' BIT WAS CLEARED BY
 ; CON.RESET. IF NOT, RETURN HERE.
 ; CNTRL RESET DID NOT CLEAR
 ; NXD BIT IN RKER
 ; CHECK IF 'HE' & 'ERR' BITS WERE CLEARED
 ; BY CON.RESET. IF NOT RETURN HERE.
 ; CNTRL RESET DID NOT CLEAR
 ; HE OR ERR BIT IN RKCS
 ; GO CHECK & WAIT FOR R/W/S RDY
 ; TO SET. IF SET SKIP ERROR
 ; R/W/S SHOULD BE SET, IT'S
 ; NOT

;*****
 ;*TEST 45 SIMULATE & CHECK NYC ERROR
 ;*THIS TEST SIMULATES THE NON-EXISTENT CYLINDER ERROR & CHECKS
 ;*IF IT IS DETECTED BY THE NYC BIT OF RKER, HE & ERR BITS
 ;*OF RKCS. IT IS CHECKED IF THEY CAN BE CLEARED BY CONTROL
 ;*RESET

;*****
 ;*ST45: SCOPE

015512 000004
 015514 013700 002526
 015520 012737 177773 002556
 015526 013702 002544
 015532 052702 014540
 015536 012737 015544 001110
 015544 104412

2\$: MOV RKCS,RO
 MOV #-5,COUNT
 MOV DRIVAD,R2
 BIS #14540,R2
 MOV #3\$,SLPERR
 3\$: CNT.RESET

; ALLOW 'ERROR 133' ONLY 5 TIMES
 ; GET ADRES OF DRIVE
 ; SET BITS FOR CYL 313
 ; SET RETURN ADRES FOR
 ; LUPING ON EROR (SW9)
 ; GO, DO CONTROL RESET
 ; THIS IS A CALL FOR THE 'CNTRL-
 ; RESET' ROUTINE. A CONTROL RESET IS

Address	Op Code	Op 1	Op 2	Op 3	Op 4	Op 5	Op 6	Op 7	Op 8	Op 9	Op 10	Op 11	Op 12	Op 13	Op 14	Op 15	Op 16	Op 17	Op 18	Op 19	Op 20	Op 21	Op 22	Op 23	Op 24	Op 25	Op 26	Op 27	Op 28	Op 29	Op 30	Op 31	Op 32	Op 33	Op 34	Op 35	Op 36	Op 37	Op 38	Op 39	Op 40	Op 41	Op 42	Op 43	Op 44	Op 45	Op 46	Op 47	Op 48	Op 49	Op 50	Op 51	Op 52	Op 53	Op 54	Op 55	Op 56	Op 57	Op 58	Op 59	Op 60	Op 61	Op 62	Op 63	Op 64	Op 65	Op 66	Op 67	Op 68	Op 69	Op 70	Op 71	Op 72	Op 73	Op 74	Op 75	Op 76	Op 77	Op 78	Op 79	Op 80	Op 81	Op 82	Op 83	Op 84	Op 85	Op 86	Op 87	Op 88	Op 89	Op 90	Op 91	Op 92	Op 93	Op 94	Op 95	Op 96	Op 97	Op 98	Op 99	Op 100	Op 101	Op 102	Op 103	Op 104	Op 105	Op 106	Op 107	Op 108	Op 109	Op 110	Op 111	Op 112	Op 113	Op 114	Op 115	Op 116	Op 117	Op 118	Op 119	Op 120	Op 121	Op 122	Op 123	Op 124	Op 125	Op 126	Op 127	Op 128	Op 129	Op 130	Op 131	Op 132	Op 133	Op 134	Op 135	Op 136	Op 137	Op 138	Op 139	Op 140	Op 141	Op 142	Op 143	Op 144	Op 145	Op 146	Op 147	Op 148	Op 149	Op 150	Op 151	Op 152	Op 153	Op 154	Op 155	Op 156	Op 157	Op 158	Op 159	Op 160	Op 161	Op 162	Op 163	Op 164	Op 165	Op 166	Op 167	Op 168	Op 169	Op 170	Op 171	Op 172	Op 173	Op 174	Op 175	Op 176	Op 177	Op 178	Op 179	Op 180	Op 181	Op 182	Op 183	Op 184	Op 185	Op 186	Op 187	Op 188	Op 189	Op 190	Op 191	Op 192	Op 193	Op 194	Op 195	Op 196	Op 197	Op 198	Op 199	Op 200	Op 201	Op 202	Op 203	Op 204	Op 205	Op 206	Op 207	Op 208	Op 209	Op 210	Op 211	Op 212	Op 213	Op 214	Op 215	Op 216	Op 217	Op 218	Op 219	Op 220	Op 221	Op 222	Op 223	Op 224	Op 225	Op 226	Op 227	Op 228	Op 229	Op 230	Op 231	Op 232	Op 233	Op 234	Op 235	Op 236	Op 237	Op 238	Op 239	Op 240	Op 241	Op 242	Op 243	Op 244	Op 245	Op 246	Op 247	Op 248	Op 249	Op 250	Op 251	Op 252	Op 253	Op 254	Op 255	Op 256	Op 257	Op 258	Op 259	Op 260	Op 261	Op 262	Op 263	Op 264	Op 265	Op 266	Op 267	Op 268	Op 269	Op 270	Op 271	Op 272	Op 273	Op 274	Op 275	Op 276	Op 277	Op 278	Op 279	Op 280	Op 281	Op 282	Op 283	Op 284	Op 285	Op 286	Op 287	Op 288	Op 289	Op 290	Op 291	Op 292	Op 293	Op 294	Op 295	Op 296	Op 297	Op 298	Op 299	Op 300	Op 301	Op 302	Op 303	Op 304	Op 305	Op 306	Op 307	Op 308	Op 309	Op 310	Op 311	Op 312	Op 313	Op 314	Op 315	Op 316	Op 317	Op 318	Op 319	Op 320	Op 321	Op 322	Op 323	Op 324	Op 325	Op 326	Op 327	Op 328	Op 329	Op 330	Op 331	Op 332	Op 333	Op 334	Op 335	Op 336	Op 337	Op 338	Op 339	Op 340	Op 341	Op 342	Op 343	Op 344	Op 345	Op 346	Op 347	Op 348	Op 349	Op 350	Op 351	Op 352	Op 353	Op 354	Op 355	Op 356	Op 357	Op 358	Op 359	Op 360	Op 361	Op 362	Op 363	Op 364	Op 365	Op 366	Op 367	Op 368	Op 369	Op 370	Op 371	Op 372	Op 373	Op 374	Op 375	Op 376	Op 377	Op 378	Op 379	Op 380</
---------	---------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	----------

MAINDEC-11-DZKK-C
DZKKC.P11 T45

MACY11 27(732) 16-SEP-76 16:00 PAGE 100
SIMULATE & CHECK NXC ERROR

```

5259
5260 015672 004737 020770      JSR      PC,CHKECLR      ;OCCURS GO BACK TO TEST 10
5261                                ;CHECK IF 'NXC' BIT WAS CLEARED BY
5262 015676 104102      ERROR    102      ;CON.RESET. IF NOT, RETURN HERE.
5263                                ;CNTRL RESET DID NOT CLEAR
5264 015700 032710 140000      7$:      BIT      #140000,2R0      ;NXC BIT IN RKER.
5265 015704 001405      BEQ        TST46      ;DID HE & ERR BITS GET CLEARED?
5266 015706 010037 001162      MOV      R0,$REG0      ;YES, EXIT
5267 015712 011037 001164      MOV      2R0,$REG1      ;GET ADRES OF RKCS
5268 015716 104102      ERROR    102      ;GET RKCS CONTENTS
5269                                ;CNTRL RESET DID NOT CLEAR
5270                                ;HE OR ERR BIT IN RKCS
5271
5272      ;*****
5273      ;*TEST 46      SIMULATE & CHECK NXS ERROR
5274      ;*THIS TEST SIMULATES NON-EXISTENT SECTOR ERROR & CHECKS THAT
5275      ;*IT IS DETECTED BY NXS BIT OF RKER. IT IS CHECKED THAT
5276      ;*WHEN NXS SETS HE & ERR OF RKER ALSO SETS, AND ALL THREE
5277      ;*CAN BE CLEARED BY CONTROL RESET.
5278      ;*****
5279 015720 000004      TST46:      SCOPE
5280 015722 104412      CNT.RESET      ;GO, DO CONTROL RESET
5281                                ;THIS IS A CALL FOR THE 'CNTRL-
5282                                ;RESET' ROUTINE. A CONTROL RESET IS
5283                                ;ISSUED AND AFTER A CERTAIN TIME
5284                                ;IF THE 'CNTRL RDY' DOES NOT SET
5285                                ;AN ERROR IS REPORTED. NOTE THAT
5286                                ;THE PC IN ERROR MESSAGE IS THE
5287                                ;PC WHERE 'CNT.RESET' IS LOCATED.
5288                                ;THIS IS A VERY BASIC ERR& IF IT
5289                                ;OCCURS GO BACK TO TEST 10
5290 015724 013700 002526      MOV      RKCS,R0
5291 015730 013777 002544 164576      MOV      DRIVAD,2RKDA      ;GET ADRES OF DRIVE
5292 015736 052777 000014 164570      BIS      #14,2RKDA      ;SET BITS FOR SECTOR 12 (DECIMAL)
5293 015744 012777 177777 164556      MOV      #-1,2RKWC      ;READ 1 WORD
5294 015752 012777 032724 164552      MOV      #CUTBUF,2RKBA      ;INTO THIS BUS ADRES
5295 015764 012710 000005      MOV      #5,2R0      ;READ, GO (FROM NX SECTOR)
5296                                ;THIS IS A CALL FOR 'CN.RDY'
5297                                ;ROUTINE WHICH WAITS FOR CNT
5298                                ;RDY TO SET. IF CNTRL RDY DOES
5299                                ;NOT SET WITHIN 883 MS/ 11-20
5300                                ;(176 MS FOR 11-45 WITH BIPOLAR)
5301                                ;AN ERROR IS REPORTED
5302                                ;NXS ERROR SHOULD OCCUR NOW
5303 015766 017702 164532      MOV      2RKER,R2
5304 015772 032702 000040      BIT      #40,R2      ;DID NXS BIT SET IN RKER?
5305 015776 001006      BNE        1$      ;YES, BRANCH
5306 016000 004737 020422      JSR      PC,GT2RG      ;GO GET RKCS, RKER
5307 016004 012737 000040 001166      MOV      #40,$REG2      ;THIS BIT (NXS) IN RKER DID NOT SET
5308 016012 104105      ERROR    105      ;NXS BIT DID NOT SET ON SIMULATING
5309                                ;NXS ERROR
5310 016014 042702 000040      1$:      BIC      #40,R2      ;MASK NXS BIT
5311 016020 001407      BEQ        2$      ;CHECK IF ANY OTHER
5312                                ;RKER BIT SET
5313 016022 012737 000040 001162      MOV      #40,$REG0      ;GET EXPCTD RKER
5314 016030 017737 164470 001164      MOV      2RKER,$REG1      ;GET RKER RECVD
5315 016036 104107      ERROR    107      ;ONLY 'NXS' SHOULD BE SET

```

```

5315                                     ; IN RKER, ANOTHER RKER BIT
5316                                     ; WAS SET. (NOTE 'NXS' WAS
5317                                     ; SIMULATED)
5318 016040 022710 140204          2$:  CMP      #140204,ARO
5319 016044 001403                BEQ      3$
5320 016046 004737 020422          JSR      PC,GT2RG
5321 016052 104106                ERROR    106
5322                                     ; HE OR ERR BIT DID NOT SET WHEN
5323                                     ; NXS ERROR OCCURED
5324 016054 104412          3$:  CNT.RESET
5325                                     ; GO DO CONTROL RESET
5326                                     ; THIS IS A CALL FOR THE 'CNTRL-
5327                                     ; RESET' ROUTINE. A CONTROL RESET IS
5328                                     ; ISSUED AND AFTER A CERTAIN TIME
5329                                     ; IF THE 'CNTRL RDY' DOES NOT SET
5330                                     ; AN ERROR IS REPORTED. NOTE THAT
5331                                     ; THE PC IN ERROR MESSAGE IS THE
5332                                     ; PC WHERE 'CNT.RESET' IS LOCATED.
5333                                     ; THIS IS A VERY BASIC ERR& IF IT
5334 016056 004737 020770          JSR      PC,CHKECLR
5335                                     ; CHECK IF 'NXS' BIT WAS CLEARED BY
5336 016062 104102                ERROR    102
5337                                     ; CON.RESET. IF NOT, RETUEN HERE.
5338 016064 004737 021014          4$:  JSR      PC,CHKCCLR
5339                                     ; CNTRL RESET DID NOT CLEAR
5340 016070 104102                ERROR    102
5341                                     ; NXS BIT IN RKER
5342                                     ; CHECK IF 'HE' & 'ERR' BITS WERE CLEARED
5343                                     ; BY CON.RESET. IF NOT, RETURN HERE.
5344                                     ; RKCS BITS ERR OR HE WERE NOT
5345                                     ; CLEARED BY CNTRL RESET
5346                                     ;*****
5347                                     ;*TEST 47 SIMULATE & CHECK WCE
5348                                     ;*THIS TEST SIMULATES A WRITE CHECK ERROR AND CHECKS THAT IT
5349                                     ;*IS DETECTED BY WCE BIT OF RKER. FOR COMPARISON IT USES
5350                                     ;*THE 256 WORDS DATA BLOCK WRITTEN ON SECTOR 0, CYLINDER 0
5351                                     ;*IN A PREVIOUS TEST. THIS BLOCK IS COMPARED WITH THE 256 WORDS
5352                                     ;*MEMORY BUFFER STARTING AT 'OUTBUF'. WCE IS SIMULATED BY
5353                                     ;*DROPPING A BIT FROM ONE OF THE WORDS IN THE MEMORY BUFFER.
5354                                     ;*****
5355 016072 000004          TST47:  SCOPE
5356 016074 013700 002526          MOV      RKCS,RO
5357 016100 104412          CNT.RESET
5358                                     ; GO DO CONTROL RESET
5359                                     ; THIS IS A CALL FOR THE 'CNTRL-
5360                                     ; RESET' ROUTINE. A CONTROL RESET IS
5361                                     ; ISSUED AND AFTER A CERTAIN TIME
5362                                     ; IF THE 'CNTRL RDY' DOES NOT SET
5363                                     ; AN ERROR IS REPORTED. NOTE THAT
5364 016102 104420          TST.SIN
5365                                     ; THE PC IN ERROR MESSAGE IS THE
5366                                     ; PC WHERE 'CNT.RESET' IS LOCATED.
5367 016104 012701 032724          MOV      #OUTBUF,R1
5368 016110 012702 177400          MOV      #-400,R2
5369 016114 012703 177777          MOV      #177777,R3
5370                                     ; THIS CODE SETS UP A MEMORY
5371                                     ; BUFFER OF 256 WORDS STARTING
5372                                     ; AT OUTBUF
5373                                     ; FIRST WORD 177400
5374                                     ; SECOND 177001

```

K08

MAINDEC-11-DZRKK-C
DZRKKC.P11 T47MACY11 27(732) 16-SEP-76 16:00 PAGE 102
SIMULATE & CHECK WCE

```

5371 016120 J62703 177401 1$: ADD #177401,R3
5372 016124 010321 MOV R3,(R1)+ ;LAST WORD 000377
5373 016126 005202 INC R2 ;HAVE U GENERATED ALL 256 WORDS?
5374 016130 001373 BNE 1$ ;IF NOT, LUP BAK & GENERATE NXT
5375
5376 016132 012737 170007 032742 MOV #170007,OUTBUF+16 ;WCE WILL B SIMULATED BY DROPPING A
5377 ;BIT IN THE EIGHTH WORD WHICH IS
5378 ;SUPPOSED TO B 174007
5379 016140 012777 177400 164362 MOV #-400,ARKWC ;WRT CHK 400 WORDS
5380 016146 012777 032724 164356 MOV #OUTBUF,ARKBA ;STARTING AT THIS BUS ADRES
5381 016154 013777 002544 164352 MOV DRIVAD,ARKDA ;WITH THIS DISK ADRES, SEC 0, CYL 0
5382 016162 012710 000007 MOV #7,ARO ;WRT CHK, GO
5383
5384 016166 104411 CHKCRDY ;GO CHECK IF CONTROL RDY IS SET
5385 ;IF SO, SKIP THE EROR MESSAGE.
5386 016170 104065 ERROR 65 ;CNTRL RDY DID NOT SET
5387 ;AFTER WRT CHK
5388 016172 032777 000001 164324 3$: BIT #1,ARKER ;DID WCE BIT SET?
5389 016200 001006 BNE 4$
5390 016202 004737 020422 JSR PC,GT2RG ;GO, GET RKCS, RKER
5391 016206 012737 000001 001166 MOV #1,$REG2 ;THIS BIT (WCE) DID NOT SET
5392 016214 104105 ERROR 105 ;WCE DID NOT SET ON SIMULATING
5393 ;WCE CONDITIONS
5394 016216 022710 100206 4$: CMP #100206,ARO ;IS RKCS CORRECT?
5395 016222 001403 BEQ 5$ ;YES, BRANCH
5396 016224 004737 020422 JSR PC,GT2RG ;GO, GET RKCS, RKER
5397 016230 104106 ERROR 106 ;HE OR ERR BIT DID NOT SET WHEN
5398 ;WCE WAS SIMULATED
5399 016232 104412 5$: CNT.RESET ;CNTRL RESET
5400 016234 004737 020770 JSR PC,CHKECLR ;WAS 'WCE' BIT CLEARED?
5401 ;IF NOT, RETURN HERE.
5402 016240 104102 ERROR 102 ;CNTRL RESET DID NOT CLEAR
5403 ;WCE BIT IN RKER
5404 016242 004737 021014 6$: JSR PC,CHKCCLR ;CHECK IF 'ERR' BIT WAS CLEARED. IF
5405 ;NOT RETURN HERE.
5406 016246 104102 ERROR 102 ;CNTRL RESET DID NOT CLEAR
5407 ;RKCS
5408
5409 ;*****
5410 ;*TEST 50 CHECK THAT SSE STOPS ALL CONTROL ACTION ON SOFT ERROR
5411 ;*THIS TEST CHECKS THAT WHEN 'STOP ON SOFT ERROR' BIT IS SET IN
5412 ;*RKCS AND A SOFT ERROR IS ENCOUNTERED ALL CONTROL ACTION WILL
5413 ;*STOP AT THE END OF THE CURRENT SECTOR IF IDE BIT IS CLEAR.
5414 ;*SOFT ERROR IS SIMULATED BY A WCE AS IN THE PREVIOUS
5415 ;*TEST. THE PREVIOUS TEST & THE TEST WHICH WRITES DATA
5416 ;*BLOCK ON CYLINDER 0, SECTOR 0, SHOULD BE DONE PRIOR
5417 ;*TO THIS TEST. A TWO SECTOR 'WRT CHK' WILL BE DONE,
5418 ;*CONTROL ACTION SHOULD STOP AFTER THE FIRST SECTOR DURING
5419 ;*WHICH A SOFT ERROR IS SIMULATED.
5420 ;*****
5421 016250 000004 tst50: SCOPE
5422 016252 104412 CNT.RESET ;GO, DO CONTROL RESET
5423 ;THIS IS A CALL FOR THE 'CNTRL-
5424 ;RESET' ROUTINE. A CONTROL RESET IS
5425 ;ISSUED AND AFTER A CERTAIN TIME
5426 ;IF THE 'CNTRL RDY' DOES NOT SET

```

MAINDEC-11-DZRKK-C
DZRKKC.P11 T50

MACY11 27(732) 16-SEP-76 16:00 PAGE 103
CHECK THAT SSE STOPS ALL CONTROL ACTION ON SOFT ERROR

```

5427
5428
5429
5430
5431
5432 016254 104420 TST.SIN
5433
5434 016256 013700 002526 MOV RKCS,RO
5435 016262 012737 170007 032742 MOV #170007,OUTBUF+16
5436
5437
5438
5439 016270 013701 002544 MOV DRIVAD,R1
5440 016274 012777 177000 164226 MOV #-1000,ARKWC
5441 016302 012777 032724 164222 MOV #OUTBUF,ARKBA
5442 016310 010177 164220 MOV R1,ARKDA
5443 016314 012710 000407 MOV #407,ARO
5444 016320 104411 CHKCRDY
5445
5446 016322 104065 ERROR 65
5447
5448
5449
5450 016324 022777 000001 164172 2$: CMP #1,ARKER
5451
5452 016332 001407 BEQ 3$
5453 016334 012737 000001 001162 MOV #1,$REG0
5454 016342 017737 164156 001164 MOV ARKER,$REG1
5455 016350 104107 ERROR 107
5456
5457
5458
5459 016352 005201 3$: INC R1
5460 016354 020177 164154 CMP R1,ARKDA
5461
5462
5463 016360 001406 BEQ TST51
5464 016362 010137 001162 MOV R1,$REG0
5465 016366 017737 164142 001164 MOV ARKDA,$REG1
5466 016374 104070 ERROR 70
5467
5468
5469
5470
5471
5472
5473
5474
5475
5476
5477
5478
5479
5480
5481
5482

```

AN ERROR IS REPORTED. NOTE THAT
THE PC IN ERROR MESSAGE IS THE
PC WHERE 'CNT.RESET' IS LOCATED.
THIS IS A VERY BASIC ERR& IF IT
OCCURS GO BACK TO TEST 10
CHECK IF SIN IS SET. IF
SET DO DRIVE RESET TO CLR IT

WCE IS SIMULATED BY DROPPING A BIT
IN THE EIGHTH WORD (WHICH IS ACTUALLY
174007). NOTE THAT 256 WORD MEMORY
BUFFER IS CREATED IN THE PREVIOUS TEST.

WRT CHK 1000 (OCTAL) WORDS, 2 SECTORS
FROM THIS BUS ADRES
WITH THIS DISK ADRES, SEC 0, CYL 0
WRT CHK, GO, SSE
GO CHECK IF CONTROL RDY IS SET
IF SO, SKIP THE EROR MESSAGE.
CNTRL RDY DID NOT SET AFTER WRT
CHK. A SOFT ERROR (WCE) IN
SECTOR 0 SHOULD HAVE STOPPED
ALL CONTROL ACTION.
CHECK ONLY 'WCE' BIT SHOULD
BE SET?
YES, BRANCH
GET EXPCTD RKER
GET RKER RECVD
ONLY BIT 'WCE' OF RKER
SHOULD BE SET (WCE WAS
SIMULATED ABOVE). ERROR
IF IT'S NOT
CHECK THAT RKDA INCREMENTED BY
1 SECTOR ONLY IMPLYING THAT
CNTRL ACTION DID STOP AFTER
SOFT ERROR IN SECTOR 0
YES, EXIT
GET EXPCTD RKDA
GET RKDA RECVD
RKDA SHOULD HAVE INCRMNTD
BY 1 SECTOR ONLY. IT DIDN'T.
WCE WAS SIMULATED IN THE
FIRST SECTOR & A WRT CHK
OF 2 SECTORS WAS ISSUED.
CONTROLLER SHOULD STOP AFTER
DETECTING WCE IN THE FIRST
SECTOR. HENCE RKDA SHOULD
INCREMENT BY 1 SECTOR ONLY

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

M08

MAINDEC-11-DZRKK-C
DZRKKC.P11 T51

MACY11 27(732) 16-SEP-76 16:00 PAGE 104
CHECK THAT RK11 INTERRUPTS ON SOFT ERROR WHEN SSE & IDE ARE SET

```

5483 ;*TEST. PREREQUISITES FOR THIS TEST ARE THE SAME AS THOSE
5484 ;*FOR THE PREVIOUS TEST.
5485 ;*****
5486 016376 000004 TST51: SCOPE
5487 016400 104412 CNT.RESET
5488
5489 ;GO, DO CONTROL RESET
5490 ;THIS IS A CALL FOR THE 'CNTRL-
5491 ;RESET' ROUTINE. A CONTROL RESET IS
5492 ;ISSUED AND AFTER A CERTAIN TIME
5493 ;IF THE 'CNTRL RDY' DOES NOT SET
5494 ;AN ERROR IS REPORTED. NOTE THAT
5495 ;THE PC IN ERROR MESSAGE IS THE
5496 ;PC WHERE 'CNT.RESET' IS LOCATED.
5497 ;THIS IS A VERY BASIC ERR& IF IT
5498 ;OCCURS GO BACK TO TEST 10
5499 016402 104420 TST.SIN
5500 ;CHECK IF SIN IS SET, IF
5501 ;SET DO DRIVE RESET TO CLR IT
5502 ;WCE IS SIMULATED BY DROPPING A BIT
5503 ;IN THE EIGHTH WORD (WHICH IS 174007)
5504 ;NOTE THAT THE 256 WORD MEMORY
5505 ;BUFFER (STARTING AT OUTBUF) IS
5506 ;CREATED IN A PREVIOUS TEST.
5507 016412 013701 002544 MOV DRIVAD,R1
5508 016416 012777 177000 164104 MOV #-1000,ARKWC
5509 016424 012777 032724 164100 MOV #OUTBUF,ARKBA
5510 016432 010177 164076 MOV R1,ARKDA
5511 016436 013700 002576 MOV RKVEC,R0
5512 016442 012720 016474 MOV #1$, (R0)+
5513 016446 012710 000340 MOV #340,ARO
5514 016452 012777 000507 164046 MOV #507,ARKCS
5515 016460 104417 177777 WAT.INT,177777
5516
5517 ;WRT CHK 1000 (OCTAL) WORDS, 2 SECTORS
5518 ;FROM THIS BUS ADRES
5519 ;WITH THIS DISK ADRES, SEC 0, CYL 0
5520 016464 004737 020422 JSR PC,GT2RG
5521 016470 104111 ERROR 111
5522
5523 ;SET UP INTERRUPT VECTOR FOR RK11
5524 ;SET PSW ON INTERRUPT
5525 ;WRT CHK, GO, SSE, IDE SET
5526 ;WAIT FOR INTERRUPT FROM RK11
5527 ;TIME=485 MS FOR 11/20,
5528 ;97 MS FOR 11/45
5529 ;11/05
5530 ;RK11 DID NOT INTERRUPT AFTER A SOFT
5531 ;ERROR (SIMULATED) IN SECTOR 0
5532
5533 016472 000417 BR 2$
5534
5535 1$: CMP (SP)+,(SP)+
5536 016474 022626 ;RESTORE STACK POINTER (FROM RK11 INTRUPT)
5537 016476 022626 ;POP STACK (FROM WAT.INT)
5538 016500 012777 004270 164070 MOV #BADINT,ARKVEC
5539 ;RESTORE RK11 INTERRUPT VECTOR
5540 ;ADRES FOR UNEXPECTED INTERRUPTS
5541
5542 016506 005201 INC R1
5543 016510 020177 164020 CMP R1,ARKDA
5544 ;CHECK THAT RKDA INCREMENTED
5545 ;BY ONLY 1 SECTOR BEFORE INTERRUPT
5546 ;OCCURRED
5547
5548 016514 001406 BEQ 2$
5549 016516 010137 001162 MOV R1,$REG0
5550 016522 017737 164006 001164 MOV ARKDA,$REG1
5551 016530 104003 ERROR 3
5552 ;GET EXPCTD RKDA
5553 ;GET RKDA RECVD
5554 ;RKDA SHOULD HAVE INCREMENTED BY
5555 ;1 SECTOR ONLY, IF ALL CNTRL ACTION
5556 ;HAD STOPPED AFTER SOFT ERROR
5557 ;(SIMULATED) IN SECTOR 0. IT DID NOT.
5558
5559 2$: MOV #340,-(SP)
5560 016532 012746 000340 MOV #64$,-(SP)
5561 016536 012746 016544 RTI
5562 016542 000002

```


MAINDEC-11-DZRKK-C
DZRKKC.P11 T51

MACY11 27(732) 16-SEP-76 16:00 PAGE 105
CHECK THAT RK11 INTERRUPTS ON SOFT ERROR WHEN SSE & IDE ARE SET

```

5539 016544          64$: CLR      RKCS          ;CLEAR THE IDE BIT
5540 016544 005077 163756
5541
5542
5543 ::*****
5544 ;*TEST 52      CHECK THE MEX BITS IN RKCS
5545 ;*THIS TEST CHECKS OUT THE EXTENDED MEMORY BITS OF THE RKCS.
5546 ;*THE RKBA IS SET TO 177776 AND A ONE WORD WRITE CHECK IS TRIED.
5547 ;*THIS COULD GIVE RISE TO NXM ERROR, BUT EVEN THEN THE RKBA
5548 ;*SHOULD OVERFLOW INTO THE MEX BITS. SIMILIARLY IT IS CHECKED
5549 ;*THAT THE OVERFLOWING BIT CAN MAKE THE MEX BITS COUNT
5550 ;*01,10,11,00.
5551 ::*****
5552 016550 000004      1$T52: SCOPE
5553 016552 013700      MOV      RKCS,R0
5554 016556 0.2701      MOV      #-4,R1
5555 016562 005002      CLR      R2
5556 016564 012737 016572 001110      MOV      #13,$LPERR
5557
5558 016572 104416 000142      1$: DELAY      142
5559 016576 004737 021050      JSR      PC,TSTRWS
5560 016602 104016      ERROR      16
5561 016604 104412      CNT.RESET
5562
5563
5564
5565
5566
5567
5568
5569
5570
5571 016606 010210      MOV      R2,R0
5572 016610 012777 177777 163712      MOV      #-1,RKWC
5573 016616 013777 002544 163710      MOV      DRIVAD,RKDA
5574 016624 012777 177776 163700      MOV      #177776,RKBA
5575
5576 016632 052710 000007      BIS      #7,R0
5577
5578
5579
5580 016636 104411      CHKCRDY
5581
5582 016640 104065      3$: ERROR      65
5583 016642 010205      MOV      R2,R5
5584 016644 062705 000020      ADD      #20,R5
5585 016650 042705 000100      BIC      #100,R5
5586 016654 011004      MOV      R0,R4
5587 016656 042704 177717      BIC      #177717,R4
5588 016662 020504      CMP      R5,R4
5589 016664 001405      BEQ      4$
5590 016666 010537 001162      MOV      R5,$REG0
5591 016672 010437 001164      MOV      R4,$REG1
5592 016676 104112      ERROR      112
5593
5594

```

;SET UP THE COUNT
 ;INITIALIZE MEX BITS TO B SET IN RKCS
 ;SET RETURN ADRES FOR
 ;LUPING ON EROR (SW9)
 ;TIME DELAY
 ;WAIT FOR R/W/S RDY
 ;R/W/S RDY IS NOT SET
 ;GO, DO CONTROL RESET
 ;THIS IS A CALL FOR THE 'CNTRL-
 ;RESET' ROUTINE. A CONTROL RESET IS
 ;ISSUED AND AFTER A CERTAIN TIME
 ;IF THE 'CNTRL RDY' DOES NOT SET
 ;AN ERROR IS REPORTED. NOTE THAT
 ;THE PC IN ERROR MESSAGE IS THE
 ;PC WHERE 'CNT.RESET' IS LOCATED.
 ;THIS IS A VERY BASIC ERR& IF IT
 ;OCCURS GO BACK TO TEST 10
 ;SET MEX BITS (AS IN R2) IN RKCS
 ;WRT CHK 1 WORD
 ;THIS DISK ADRES, SEC 0, CYL 0
 ;THIS BUS ADRES. NOTE THIS BA
 ;IN CONJUNCTION WITH MEX BITS OF RKCS
 ;WRT CHK, GO
 ;THERE MAY BE A NXM OR WCE BUT
 ;WHATEVER THE CASE RKBA SHOULD
 ;OVERFLOW MAKING THE MEX BITS COUNT
 ;GO CHECK IF CONTROL RDY IS SET
 ;IF SO, SKIP THE EROR MESSAGE.
 ;CNTRL RDY DID NOT SET AFTER WRT CHK
 ;MEX BITS SHOULD INCREMENT BY 1 TO THIS
 ;MASK OUT IDE BIT POSITION. IF SET
 ;GET RKCS
 ;MASK OUT ALL BITS EXCEPT MEX
 ;DID MEX BITS INCREMENT CORRECTLY?
 ;YES, BRANCH
 ;GET EXPCTD MEX BITS
 ;GET MEX BITS RECVD
 ;MEX BITS DID NOT INCREMENT AS
 ;'EXPCTD' WHEN RKBA OVERFLOWED.
 ;NOTE THAT BIT POSITION 4 & 5

MAINDEC-11-DZRAK-C
DZRAK.C.P11 T52

MACY11 27(732) 16-SEP-76 16:00 PAGE 106
CHECK THE MEX BITS IN RKCS

```
; REFLECT MEX BITS 0 & 1 IN THE
; ERROR MESSAGE.
; GET RKER
; MASK WCE DLT, NXM BIT, IF SET
; BRANCH IF REST OF RKER CLR.
; MASK NON-WCE BITS
; THIS IS THE EXPCTD RKER
; GET RKER RECVD
; ERROR IN RKER. IT SHOULD
; BE AS EXPECTED IN
; ERROR MESSAGE
; INCREMENT TO NXT MEX BIT
; HAVE U CHKD THE MEX BITS 4 TIMES?
; IF NOT, LUP BACK
```

```
::*****
; *TEST 53 TRANSFER FROM DISK TO ITY
; * THIS TEST CHECKS THE HIGH ORDER BITS OF THE ADDRESS
; * LINES. FIRST A ONE WORD (100) IS WRITTEN ON SECTOR
; * 2, CYL 0. THEN IT IS READ BACK, BUT THE NPR IS DONE
; * NOT TO THE MEMORY, BUT THE TELETYPE BUFFER (TKS 177560)
; * AND IT CHECKED THAT THE WORD WAS RECIEVED CORRECTLY.
; * IF IT IS NOT, AN ERROR IS REPORTED. THIS TEST IS
; * SKIPPED ON AN 11/05.
```

```
::*****
; *DO 1 ITERATION
; *THIS CODE FINDS OUT IF THE CPU
; *IS AN 11/05 OR ELSE.
; *ON AN 11/05, RO (177700) CAN BE
; *ADDRESSED AS A MEMORY LOCATION, BUT
; *ON ANY OTHER CPU IF 177700 IS REFERENCED
; *A TIME OUT WILL OCCUR.
; *SET UP TIME OUT VECTOR
; *REFERENCE RO
; *RO WAS REFERENCED W/O TIMEOUT
; *HENCE 11/05
; *SKIP THIS TEST
; *RESTORE STACK POINTER
; *RESTORE TIMEOUT VECTOR
```

```
; GO, DO CONTROL RESET
; THIS IS A CALL FOR THE 'CNTRL-
; RESET' ROUTINE. A CONTROL RESET IS
; ISSUED AND AFTER A CERTAIN TIME
; IF THE 'CNTRL ROY' DOES NOT SET
; AN ERROR IS REPORTED. NOTE THAT
; THE PC IN ERROR MESSAGE IS THE
; PC WHERE 'CNT.RESET' IS LOCATED.
; THIS IS A VERY BASIC ERR3 IF IT
; OCCURS GO BACK TO TEST 1C
```

```
5595
5596
5597 016700 017703 16362C 45: MOV JAKER,R3
5598 016704 010305 MOV R3,R5
5599 016706 042703 003001 BIC #3001,R3
5600 016712 001410 BEQ 55
5601 016714 042705 177776 BIC #177776,R5
5602 016720 010537 001162 MOV R5,$REG0
5603 016724 017737 163574 001164 MOV JAKER,$REG1
5604 016732 104107 ERROR 107
5605
5606
5607 016734 062702 000020 55: ADD #20,R2
5608 016740 005201 INC R1
5609 016742 001313 BNE 15
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619
5620
5621 016744 000004
5622 016746 012737 000001 001206
5623
5624
5625
5626
5627
5628
5629 016754 012737 016776 000004 MOV #55,2#4
5630 016762 005737 177700 TST 2#177700
5631 016766 012737 004224 000004 MOV #BADTMO,2#4
5632
5633 016774 000520
5634 016776 022626 55: BR TST54
5635 017000 012737 004224 000004 CMP (SP)+,(SP)+
5636 017006 012746 000340 MOV #BADTMO,2#4
5637 017012 012746 017020 MOV #340,-(SP)
5638 017016 000002 RTI
5639 017020
5640 017020 013700 002526 645: MOV RKCS,RO
5641 017024 104412 CNT.RESET
5642
5643
5644
5645
5646
5647
5648
5649
5650
```


MAINDEC-11-DZKCC-C
DZKCC.P11 T53MACY11 27(732) 16-SEP-76 16:00 PAGE 107
TRANSFER FROM DISK TO TTY

5651	017026	012701	032724		MOV	#OUTBUF,R1	
5652	017032	013704	002532		MOV	RKBA,R4	
5653	017036	012711	000100		MOV	#100,DR1	:WRITE THIS WORD
5654	017042	012777	177777	163460	MOV	#-1,DRKWC	:WRITE 1 WORD
5655	017050	013702	002544		MOV	DRIVAD,R2	
5656	017054	052702	000002		BIS	#2,R2	:ON CYL 0, SEC 2
5657	017060	010277	163450		MOV	R2,DRKDA	
5658	017064	010114			MOV	R1,DR4	:FROM THIS MEMORY LOC
5659	017066	012710	000003		MOV	#3,DR0	:WRITE, GO
5660	017072	005003			CLR	R3	
5661	017074	105710		15:	TSTB	DR0	
5662	017076	100410			BMI	25	
5663	017100	005203			INC	R3	
5664	017102	001374			BNE	15	
5665	017104	004737	020406		JSR	PC,GT4RG	:GET RKCS, ER, DS
5666	017110	010237	001202		MOV	R2,\$REG10	:GET THE STARTING ADRES
5667	017114	104415			BRKDAY		:BREAK IT INTO DRV #, CYL, SUR, SEC #
5668	017116	104031			ERROR	31	:CNTRL RDY DID NOT SET AFTER
5669							:WRITE OF 1 WORD ON CYL 0, SEC 2
5670	017120	012777	177777	163402	25:	MOV	#-1,DRKWC
5671	017126	010277	163402		MOV	R2,DRKDA	:READ 1 WORD
5672	017132	013714	001144		MOV	\$TKS,DR4	:FROM SEC 2, CYL 0
5673	017136	005077	162002		CLR	\$TKS	:INTO TTY STATUS REGISTER
5674							:CLEAR TTY KEY BRD STATUS REG
5675	017142	012710	000065		MOV	#65,DR0	:READ, MEX BITS SET
5676	017146	005003			CLR	R3	
5677	017150	105710		35:	TSTB	DR0	
5678	017152	100410			BMI	45	
5679	017154	005203			INC	R3	
5680	017156	001374			BNE	35	
5681	017160	004737	020406		JSR	PC,GT4RG	:GET THE STARTING ADRES
5682	017164	010237	001202		MOV	R2,\$REG10	:BREAK IT INTO DR#, CYL, SUR, SEC#
5683	017170	104415			BRKDAY		:CNTRL RDY DIDN'T SET AFTER
5684	017172	104045			ERROR	45	:READ OF 1 WORD FROM CYL 0, SEC 2.
5685							:IN EROR MSGE, <DSK-ADRES> GIVES
5686							:ADRES WHERE READ BEGAN. 'RKDA'
5687							:GIVES CONTENTS OF RKDA AT TIME OF EROR
5688							:WAS THE CORRECT WORD READ INTO
5689	017174	032737	000100	001144	45:	BIT	#100,\$TKS
5690							:THE TTY STATUS REGISTER?
5691	017202	001015			BNE	TST54	:YES, EXIT
5692	017204	017705	161734		MOV	\$TKS,R5	:GET THE WORD RECVD FROM DISK
5693	017210	010537	001164		MOV	R5,\$REG1	
5694	017214	052705	000100		BIS	#100,R5	:THIS WORD WAS EXPCTD
5695	017220	010537	001162		MOV	R5,\$REG0	:STORE EXPCTD WORD
5696	017224	011437	001166		MOV	DR4,\$REG2	:GET RKBA
5697	017230	011037	001170		MOV	DR0,\$REG3	:GET RKCS
5698	017234	104115			ERROR	115	:DATA ERROR. A ONE WORD (100)
5699							:NPR WAS TRIED FROM DISK TO
5700							:TTY KEYBOARD STATUS REGISTER
5701							: (17756) . BIT 6 SHOULD HAVE BEEN
5702							: SET AS RESULT OF THIS
5703							: BUT IT WAS NOT
5704							
5705							
5706							

;*****

```

5707          : *TEST 54      CHECK THAT RKBA CAN COUNT CORRECTLY
5708          : *THIS TEST CHECKS THAT RKBA CAN COUNT CORRECTLY. IT IS SET
5709          : *TO THE DESIRED INITIAL VALUE. THEN A ONE WORD WRITE CHECK
5710          : *IS TRIED, WITH MEX (MEMORY EXTENSION) BITS SET. IF THERE IS
5711          : *NO MEMORY PRESENT (FOR CERTAIN BUS ADDRESSES), THERE
5712          : *WILL BE AN NXM ERROR STOPPING CONTROLLER ACTION. BUT RKBA
5713          : *SHOULD HAVE INCREMENTED BY 1 FROM ITS INITIAL VALUE. IF IT
5714          : *HAS NOT, AN ERROR IS REPORTED.
5715          : *****
5716          017236 000004          1ST54: SCOPE
5717          017240 012737 000005 001206      MOV      #5,$TIMES          ;DO 5 ITERATIONS
5718          017246 104420          TST.SIN      ;CHECK IF SIN SET, IF SET DRV RESET
5719          017250 005001          CLR      R1      ;INITIALIZE (VALUE OF RKBA)
5720          017252 012702 000002          MOV      #2,R2      ;INITIALIZE (INCMNTD VALUE OF RKBA)
5721
5722          017256 012737 017270 001110      MOV      #1,$SLPERR      ;SET RETURN ADRES FOR LUPING
5723
5724
5725          017264 013705 002532          MOV      RKBA,R5
5726          017270 004737 021050          JSR      PC,TSTRWS      ;WAIT FOR R/W/S RDY
5727          017274 104016          ERROR      16      ;R/W/S RDY IS NOT SET
5728          017276 104412          CNT.RESET      ;DO CONTROL RESET
5729          017300 012777 177777 163222      MOV      #-1,$RKWC      ;WRITE CHK 1 WORD
5730          017306 010115          MOV      R1,$R5      ;THIS BUS ADRES
5731          017310 013777 002544 163216      MOV      $R5,$RKDA      ;SET DISK ADRES
5732          017316 012777 000067 163202      MOV      #67,$RKCS      ;WRITE CHECK, GO, MEX BITS SET
5733          017324 104411          CHKCRDY      ;GO CHECK IF CONTROL RDY IS SET
5734
5735          017326 104065          ERROR      65      ;IF SO, SKIP THE ERROR MESSAGE.
5736
5737
5738
5739          017330 005237 002552          INC      INDX1      ;CNTRL RDY DID NOT SET AFTER
5740          017334 001417          BEQ      5$      ;WRT CHK WAS TRIED TO NXM LOC
5741
5742          017336 020215          3$: CMP      R2,$R5      ;U MIGHT WANT TO USE TESTS
5743
5744          017340 001410          BEQ      4$      ;CHECKING MEX BITS & NXM.
5745          017342 010137 001162          MOV      R1,$REGO      ;ALLOW ONLY 5 ERRORS OF ABOVE KIND
5746          017346 011537 001164          MOV      $R5,$REG1
5747          017352 104017          ERROR      17
5748
5749
5750
5751
5752
5753
5754
5755          017354 005237 002554          INC      INDX2
5756          017360 001405          BEQ      5$
5757          017362 060201          4$: ADD      R2,R1
5758          017364 010102          MOV      R1,R2
5759          017366 062702 000002          ADD      #2,R2
5760          017372 001336          BNE      1$
5761
5762          017374          5$:

```

;DUMMY EXIT POINT

5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783
5784
5785
5786
5787
5788
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818

017374 000004
017376 012737 000001 001206
017404 005737 002604
017410 001403
017412 004537 024542
017416 104120

017420

017420 000004
017422 012737 000001 001206
017430 005237 002546

017434 004737 021116
017440 023737 002602 002546 BTEOP:

017446 001405
017450 062737 020000 002544
017456 000137 004514

017462 005037 001112

017466 000004

;TEST 55 CHECK FOR RK-05F

;*THIS TEST CHECKS RK-05F TYPE DRIVES
;*TO INSURE THAT IF SEEKS ARE ISSUED ON ONE
;*DRIVE, THE OTHER DRIVE BECOMES BUSY

†ST55: SCOPE
MOV #1, \$TIMES ;DO 1 ITERATION
TST FFLAG ;SEE IF RK-05F
BEQ 1\$;NOT F
JSR R5, FCHECK ;SEE IF OTHER GOES BUSY
ERROR 120

1\$:

;TEST 56 END OF PROGRAM

;*THIS IS NOT A TEST, BUT A LINKAGE PROVIDED TO PERFORM
;*THE ABOVE SUB-TESTS FOR ALL DRIVES THAT ARE PRESENT.
;*NOTE THAT THE NEXT TEST- HARDWARE POLLING LOGIC-
;*IS DONE USING ALL THE DRIVES THAT ARE INDICATED PRESENT.
;*DO NOT LOOP ON THIS 'TEST'.

†ST56: SCOPE
MOV #1, \$TIMES ;DO 1 ITERATION
INC DRVDON ;INCREMENT THE COUNT FOR THE NUMBER
;OF DRIVES THAT ARE CHECKED
JSR PC, DRESET ;RESET THE DRIVE
CMP DRIVS, DRVDON ;HAVE U TESTED ALL THE DRIVES
;THAT ARE PRESENT?
BEQ 1\$;IF YES, EXIT
ADD #20000, DRIVAD ;ADRES THE NXT POSSIBLE DRIVE
JMP NUORV ;GO BACK AND TEST THE NEXT
;DRIVE PRESENT

1\$: CLR \$ERTTL

;TEST 57 CHECK HARDWARE POLLING LOGIC

;*THIS TEST CHECKS THE HARDWARE POLL LOGIC, USING ALL THE DRIVES
;*PRESENT ON THE RK11. ATLEAST TWO DRIVES SHOULD BE PRESENT
;*TO DO A MEANINGFUL HARDWARE POLL. SEQUENCE OF OPERATIONS IS
;*AS FOLLOWING:
;*1) NUMBER OF DRIVES ON THE RK11 IS ASCERTAINED.
;*2) HAVING LOCKED OUT ALL INTERRUPTS (CPU PR 7), SEEK IS INITIATED
;*FOR ONE DRIVE AT A TIME, ONLY WHEN 'CNTRL RDY' IS SET.
;*3) CPU PRIORITY IS DROPPED TO 4 SO THAT RK11 CAN INTERRUPT, THE INCOMING
;*INTERRUPT IS PROCESSED TO CHECK IF IT WAS DUE TO 'SEEK DONE' BY
;*ONE OF THE DRIVES.
;*4) IF BY THE END OF THE SET TIME A DRIVE HAS NOT INTERRUPTED
;*AN ERROR MESSAGE IS GIVEN INDICATING WHICH DRIVE DID NOT
;*INTERRUPT AFTER SEEK WAS DONE.

†ST57: SCOPE

MAINDEC-11-DZKCC-C
DZKCC.P11 T57

MACY11 27(732) 16-SEP-76 16:00 PAGE 110
CHECK HARDWARE POLLING LOGIC

5819	017470	012737	000005	001206	MOV	#5, \$TIMES	:: DO 5 ITERATIONS
5820	017476	005237	002634		INC	\$IZYET	:: FOUNR RK05F YET?
5821	017502	001002			BNE	25\$	:: YES
5822	017504	004737	024666		JSR	PC, SIZEF	:: FIND WHICH ARE RK-05F
5823	017510	005037	002632	25\$:	CLR	PHYDRV	:: NUMBER OF ACTUAL DRIVES
5824	017514	012700	002610		MOV	#DRIVO, R0	:: TABLE
5825	017520	005710		23\$:	TST	(R0)	:: DRIVE HERE+?
5826	017522	001405			BEQ	22\$	:: NO
5827	017524	005237	002632		INC	PHYDRV	:: COUNT DRIVE
5828	017530	005710			TST	(R0)	:: RK05F?
5829	017532	100001			BPL	22\$	:: NO
5830	017534	005720			TST	(R0)+	:: DONT COUNT F TWICE
5831	017536	005720		22\$:	TST	(R0)+	:: NEXT DRIVE
5832	017540	020027	002627		CMP	R0, #DRIV7+1	:: ALL YET
5833	017544	002765			BLT	23\$	:: NO
5834	017546	005037	002606		CLR	ODDEVN	:: EVEN DRIVES FIRST IF F
5835	017552	005737	002602	75\$:	TST	DRIVS	:: ANY DRIVES PRESENT?
5836	017556	001002			BNE	20\$	:: YES
5837	017560	000137	020264		JMP	\$EOP	:: NO
5838	017564	005237	002630	20\$:	INC	T56FLG	
5839	017570	013700	002526		MOV	RKCS, R0	
5840	017574	005037	002552		CLR	INDX1	:: FLAG TO INDICATE:
5841							:: (INDX1)=0 POLLING DONE AFTER ALL
5842							:: DRIVES SEEK TO CYL 0
5843							:: (INDX1)=1 POLLING DONE AFTER ALL
5844							:: DRIVES SEEK TO CYL 4
5845	017600	005037	002554	15\$:	CLR	INDX2	:: FLAG INDICATING TYPE OF INTERRUPT
5846							:: SET TO NON-ZERO TO INDICATE
5847							:: THAT THE INTERRUPT IS DUE TO
5848							:: SEEK DONE
5849	017604	104412					:: GO, DO CONTROL RESET
5850							:: THIS IS A CALL FOR THE 'CNTRL-
5851							:: RESET' ROUTINE. A CONTROL RESET IS
5852							:: ISSUED AND AFTER A CERTAIN TIME
5853							:: IF THE 'CNTRL RDY' DOES NOT SET
5854							:: AN ERROR IS REPORTED. NOTE THAT
5855							:: THE PC IN ERROR MESSAGE IS THE
5856							:: PC WHERE 'CNT.RESET' IS LOCATED.
5857							:: THIS IS A VERY BASIC ERR8 IF IT
5858							:: OCCURS GO BACK TO TEST 10
5859	017606	005737	002552		TST	INDX1	:: PERFORMING SEEKS TO CYL 4
5860	017612	001002			BNE	.+6	:: YES, BRANCH
5861	017614	005002			CLR	R2	:: NO
5862	017616	000402			BR	.+6	
5863	017620	012702	000200		MOV	#200, R2	:: SET ADRES FOR FOURTH CYLINDER
5864	017624	012701	002610		MOV	#DRIVO, R1	:: INITIALIZE POINTER
5865	017630	012703	177770		MOV	#-10, R3	:: SET COUNT FOR 8 DRIVES
5866	017634	012705	032724		MOV	#OUTBUF, R5	:: INITIALIZE POINTER TO INDICATOR AREA
5867	017640	005025			CLR	(R5)+	:: CLEAR OUT THE 8-WORD INDICATOR
5868	017642	005203			INC	R3	:: AREA WHICH IS USED FOR DOING
5869	017644	001375			BNE	.-4	:: SOFTWARE POLLING LATER ON
5870	017646	012703	177770		MOV	#-10, R3	:: SET COUNT FOR 8 POSSIBLE DRIVES
5871	017652	012705	032724		MOV	#OUTBUF, R5	:: INITIALIZE POINTER TO INDICATOR AREA
5872	017656			15\$:			
5873	017656	012746	000340		MOV	#340, -(SP)	
5874	017662	012746	017670		MOV	#64\$, -(SP)	

MAINDEC-11-DZKRC-C
DZKRC.P11 T57

MACY11 27(732) 16-SEP-76 16:00 PAGE 111
CHECK HARDWARE POLLING LOGIC

5875	017666	000002			RTI		
5876	017670			64\$:			
5877	017670	032711	000001		BIT	#BIT0,(R1)	: IS THIS DRIVE PRESENT?
5878	017674	001433			BEQ	4\$	: IF NOT, BRANCH
5879	017676	005711			TST	(R1)	: RK06F?
5880	017700	100012			BPL	17\$	: NO, CONTINUE
5881	017702	032702	020000		BIT	#BIT13,R2	: DRIVE EVEN?
5882	017706	001404			BEQ	16\$	: YES
5883	017710	005737	002606		TST	ODDEVN	: DO WE WANT ODD?
5884	017714	001423			BEQ	4\$	: NO, SO DO NOT TEST
5885	017716	000403			BR	17\$	: ADD THIS DRIVE TO LIST
5886	017720	005737	002606	16\$:	TST	ODDEVN	: DO WE WANT EVEN?
5887	017724	001017			SNE	4\$	: NO, SO SKIP
5888	017726	010215		17\$:	MOV	R2,(R5)	: SET UP THIS WORD IN THE
5889							: INDICATOR AREA SHOWING THAT THIS
5890							: DRIVE (AS IN BITS 13-15 OF R2)
5891							: IS PRESENT
5892	017730	042725	017777		BIC	#17777,(R5)+	: MASK OUT UNWANTED BITS (CYL,SUR,SEC BITS)
5893	017734	005004			CLR	R4	
5894	017736	105710		2\$:	TSTB	2R0	: IS CNTRL RDY SET?
5895	017740	100405			BMI	3\$	: YES, BRANCH
5896	017742	005204			INC	R4	: NO, WAIT FOR IT
5897	017744	001374			BNE	2\$	: IF WAITED LONG REPORT ERROR
5898	017746	004737	020406		JSR	PC,GT4RG	: GO, GET RKCS,ER,DS,DA
5899	017752	104021			ERROR	21	: CNTRL RDY DID NOT SET AFTER ACCEPTING
5900							: ADRES FROM PREVIOUS SEEK
5901	017754	010277	162554	3\$:	MOV	R2,2RKDA	: ADRES THIS DRIVE, CYL 0 OR CYL 4
5902							: (WHICHEVER THE CASE MAY BE)
5903	017760	012710	000111		MOV	#111,2R0	: SEEK,GO,IDE SET
5904	017764	005721		4\$:	TST	(R1)+	: NEXT DRIVE DATA
5905	017766	062702	020000		ADD	#20000,R2	: INCREMENT DRIVE ADRES (BITS 15,14,13)
5906	017772	005203			INC	R3	: TO NEXT ONE
5907	017774	001330			BNE	1\$	: BRANCH BACK IF ALL DRIVES ARE
5908							: NOT CHECKED TO SEE IF THE NEXT
5909							: DRIVE IS PRESENT (& IF SO ISSUE A
5910							: SEEK TO IT)
5911							: BY NOW SEEKS HAVE BEEN ISSUED
5912							: TO ALL DRIVES PRESENT & POLLING
5913							: HAS BEGUN
5914	017776	005004			CLR	R4	
5915	020000	013702	002576	5\$:	MOV	RKVEC,R2	: SET ADRES FOR RK11 TO INTERRUPT
5916	020004	012722	020036		MOV	#6\$,(R2)+	: SET PSW ON INTERRUPT
5917	020010	012712	000340		MOV	#340,(R2)	: DROP CPU PRIORITY TO 4 SO THAT
5918	020014	013746	002574		MOV	RKPRI,-(SP)	: RK11 CAN INTERRUPT
5919	020020	012746	020026		MOV	#18\$,-(SP)	
5920	020024	000002			RTI		
5921	020026	000240		18\$:	NOP		: THIS IS A TIME LOOP DURING
5922	020030	005204			INC	R4	: WHICH ALL DRIVES PRESENT SHOULD
5923	020032	001375			BNE	18\$	: INTERRUPT
5924	020034	000452			BR	11\$	: BRANCH AND CHECK IF ALL AVAILABLE
5925							: DRIVES INTERRUPTED CORRECTLY
5926	020036	022626		6\$:	CMP	(SP)+,(SP)+	: RESTORE STACK POINTER
5927	020040	005737	002554		TST	INDX2	: WAS THIS FIRST INTERRUPT
5928							: DUE TO 'ADRES ACK' AFTER INITIATION
5929							: OF SEEK?
5930	020044	001021			BNE	9\$	: IF YES, CHECK THE FOLLOWING

MAINDEC-11-DZKRC-C
DZKRC.P11 T57

MACY11 27(732) 16-SEP-76 16:00 PAGE 112
CHECK HARDWARE POLLING LOGIC

5931									
5932	020046	032710	020000		BIT	#20000,3R0			;CHECK THAT SCP IS NOT SET
5933	020052	001403			BEQ	7\$			;BRANCH IF SCP CLEAR
5934	020054	011037	001162		MOV	3R0,\$REGO			;GET RKCS
5935	020060	104076			ERROR	76			;AFTER THE FIRST INTERRUPT WHICH
5936									;IS DUE TO INITIATION OF SEEK, SCP
5937									;SHOULD NOT HAVE SET. IT DID
5938	020062	017701	162434	7\$:	MOV	3RKDS,R1			
5939	020066	032701	160000		BIT	#160000,R1			;RKDS BITS 15-13 SHLOULD BE CLR
5940	020072	001403			BEQ	8\$			
5941	020074	010137	001162		MOV	R1,\$REGO			;GET RKDS
5942	020100	104050			ERROR	50			;SEEK, WITH IDE SET WAS ISSUED TO
5943									;ALL AVAILABLE DRIVES. THE FIRST
5944									;INTERUPT IS DUE TO SEEK INITIATED
5945									;BY FRST DRV. DRV ID BITS 13-15
5946									;SHOULD BE CLR AFTR THIS FRST INRUPT.
5947									;THEY WERE NOT IF THIS ERROR OCCURS.
5948	020102	005237	002554	8\$:	INC	INDX2			;SET UP FLAG INDICATING
5949									;THAT THE FIRST INTERRUPT DUE
5950									;TO INITIATION OF SEEK WAS
5951									;PROCESSED
5952	020106	000734			BR	5\$			;GO BACK TO THE WAIT LOOP & WAIT
5953									;FOR NEXT INTERRUPT FROM RK11
5954	020110	013703	002632	9\$:	MOV	PHYDRV,R3			;SET COUNT OF # OF DRIVES PRESENT
5955	020114	012705	032724		MOV	#OUTBUF,R5			;INITIALIZE POINTER
5956	020120	017701	162376		MOV	3RKDS,R1			;GET RKDS
5957	020124	042701	017777		BIC	#17777,R1			;MASK BITS 0-12
5958									;THE FOLLOWING CODE IS A SOFTWARE
5959									;POLL WHICH FINDS OUT WHICH DRIVE
5960									;CAUSED THE PRESENT INTERRUPT
5961									;AND SETS UP A FLAG BIT FOR
5962									;THE DRIVE #, INDICATING THAT
5963									;THIS DRIVE # INTERRUPTED
5964	020130	020125			CMP	R1,(R5)+			
5965	020132	001411			BEQ	10\$			;BRANCH IF INTERRUPTING DRIVE WAS FOUND
5966	020134	005303			DEC	R3			;HAVE U CHKD ALL DRVS PRESENT?
5967	020136	001374			BNE	.-6			;IF NOT LUP BAK & CHK
5968									;REPORT ERROR IF THE INTERRUPTING
5969									;DRIVE # (AS IN RKDS 13-15) WAS NOT
5970									;ANY ONE OF THOSE THAT ARE PRESENT
5971	020140	010146			MOV	R1,-(R6)			;GET WORD TO B SHFTD RT
5972	020142	004737	020612		JSR	PC,SHFTRT			;GO SHIFT IT
5973	020146	012637	001162		MOV	(R6)+,\$REGO			;THIS DRIVE # WAS RECVD IN RKDS AS
5974									;THE INTERRUPTING DRIVE, BUT THIS
5975									;DRIVE IS NOT PHYSICALY PRESENT
5976	020152	104051			ERROR	51			;RKDS INDICATES AN INTERRUPTING
5977									;DRIVE # (DURING H'WARE POLL) BUT
5978									;THAT DRIVE IS ACTUALLY NOT PRESENT
5979	020154	000401			BR	10\$+2			
5980	020156	005245		10\$:	INC	-(R5)			;SET UP FLAG INDICATING THAT
5981									;THE INTERRUPT FOR THIS DRIVE
5982									; (AFTER IT HAD COMPLETED ITS SEEK)
5983									;WAS PROCESSED
5984	020160	000707			BR	5\$			;GO BAK & WAIT FOR FURTHER INTRUPTS
5985	020162	013703	002632	11\$:	MOV	PHYDRV,R3			;GET # OF DRIVES
5986	020166	012705	032724		MOV	#OUTBUF,R5			;INITIALIZE POINTER

MAINDEC-11-DZRKK-C
DZRKKC.P11 T57

MACY11 27(732) 16-SEP-76 16:00 PAGE 113
CHECK HARDWARE POLLING LOGIC

```

5987
5988 020172 105715          14$: TSTB (R5)      ;DID THIS DRIVE INTERRUPT?
5989 020174 001006          BNE 13$      ;YES, BRANCH
5990 020176 011546          MOV (R5),-(R6) ;GET THIS DRIVE #
5991 020200 004737 020612   JSR PC,SHFTRT ;SHIFT IT TO THE RIGHT
5992 020204 012637 001162   MOV (R6)+,$REGO ;THIS DRIVE # DID NOT INTERRUPT
5993                                     ;DURING H'WARE POLL
5994 020210 104052          ERROR 52      ;DRIVE # (AS IN $REGO) DID NOT
5995                                     ;INTERRUPT DURING HARDWARE POLL
5996 020212 062705 000002   13$: ADD #2,R5 ;INCREMENT POINTER TO THE NEXT FLAG
5997 020216 005303          DEC R3      ;CHKD FOR ALL DRIVES?
5998 020220 001364          BNE 14$      ;IF NOT LUP BACK
5999
6000 020222 005737 002552   TST INDX1    ;DONE POLLING FOR SEEKS TO CYL 312?
6001 020226 001004          BNE TSTEND   ;IF YES, EXIT
6002 020230 005237 002552   INC INDX1    ;IF NOT, INCREMENT FLAG
6003 020234 000137 017600   JMP 15$    ;GO DO IT
6004
6005                                     ;INDICATOR TABLE
6006                                     ;THE 8-WORD INDICATOR TABLE USED IN
6007                                     ;THE FORMER PART OF THIS SUB-TEST
6008                                     ;IS LOCATED STARTING AT 'OUTBUF'.
6009                                     ;WORDS ARE SET UP TO INDICATE
6010                                     ;PRESENCE OF A DRIVE EG: IF
6011                                     ;DRIVES 0,1,2 ARE PRESENT, IT WILL
6012                                     ;LOOK LIKE
6013                                     ;OUTBUF:      000000 BITS 13,14,15
6014                                     ;              020000 CONTAIN THE
6015                                     ;              040000 DRIVE NO.
6016                                     ;              000000 REST 0'S
6017                                     ;WHEN A DRIVE INTERRUPTS AFTER SEEK
6018                                     ;IS DONE BIT 0 OF THE CORRESPONDING
6019                                     ;INDICATOR WORD IS SET. THUS FOR THE
6020                                     ;ABOVE EXAMPLE IF ALL DRIVES INTERRUPTED
6021                                     ;CORRECTLY THEN IT WILL LOOK LIKE:
6022                                     ;12$:      000001 BIT 0 SET
6023                                     ;              020001 TO INDICATE
6024                                     ;              040001 DR INTERRUPTED
6025                                     ;              000000 REST 0'S
6026
6027
6028 020240 005237 002606   TSTEND: INC ODDEVN ;NOW ODD IF RK05F
6029 020244 022737 000002 002606   CMP #2,ODDEVN ;SEE IF DONE
6030 020252 001402          BEQ 21$      ;ALL DONE
6031 020254 000137 017552   JMP T56    ;TEST AGAIN
6032 020260 005037 002630   21$: CLR T56FLG
6033
6034
6035 .SBTTL END OF PASS ROUTINE
6036
6037 ;*****
6038 ;*INCREMENT THE PASS NUMBER ($PASS)
6039 ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
6040 ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
6041 ;*IF THERES A MONITOR GO TO IT
6042 ;*IF THERE ISN'T JUMP TO ST4

```

```

6043
6044 020264
6045 020264 000004
6046 020266 005037 001102
6047 020272 005037 001206
6048 020276 005237 001100
6049 020302 042737 100000 001100
6050 020310 005327
6051 020312 000001
6052 020314 003022
6053 020316 012737
6054 020320 000001
6055 020322 020312
6056 020324 104400 020371
6057 020330 013746 001100
6058 020334 104404
6059 020336 104400 020366
6060 020342 013700 000042
6061 020346 001405
6062 020350 000005
6063 020352 004710
6064 020354 000240
6065 020356 000240
6066 020360 000240
6067 020362
6068 020362 000137
6069 020364 004146
6070 020366 377 377 000
6071 020371 015 042412 042116
6072 020376 050040 051501 020123
6073 020404 003043
6074
6075
6076
6077
6078
6079
6080
6081 .SBTTL GT2RG: ROUTINE FOR GETTING RKCS, RKER
6082
6083 ;SUBROUTINE FOR TRANSFERRING THE CONTENTS OF RKCS, RKER
6084 ;TO $REG0, $REG1 RESPECTIVELY BEFORE TYPING OUT AN ERROR MESSAGE.
6085 ;CALL: JSR PC,GT2RG
6086
6087 .SBTTL GT3RG: ROUTINE FOR GETTING RKCS, RKER, RKDS
6088
6089 ;GT3RG
6090 ;SUBROUTINE FOR TRANSFERRING THE CONTENTS OF RKCS, RKER, RKDS
6091 ;TO $REG0, $REG1, $REG2 RESPECTIVELY BEFORE TYPING OUT AN
6092 ;ERROR MESSAGE.
6093 ;CALL: JSR PC,GT3RG
6094
6095 .SBTTL GT4RG: ROUTINE FOR GETTING RKCS, RKER, RKDS, RKDA
6096
6097 ;GT4RG
6098 ;SUBROUTINE FOR TRANSFERRING CONTENTS OF RKCS, RKER, RKDS

```

```

$EOP:
  SCOPE
  CLR $TSTNM
  CLR $TIMES
  INC $PASS
  BIC #100000,$PASS
  DEC (PC)+
  $EOPCT: .WORD 1
  BGT $DOAGN
  MOV (PC)+,2(PC)+
  $ENDCT: .WORD 1
  $EOPCT
  TYPE $ENDMG
  MOV $PASS,-(SP)
  TYPDS
  TYPE $ENULL
  $GET42: MOV 2#42,R0
  BEQ $DOAGN
  RESET
  $ENDAD: JSR PC,(R0)
  NOP
  NOP
  NOP
  $DOAGN:
  JMP 2(PC)+
  $RTNAD: .WORD $T4
  $ENULL: .BYTE -1,-1,0
  $ENDMG: .ASCIZ <15><12>/END PASS #/
  ;;ZERO THE TEST NUMBER
  ;;ZERO THE NUMBER OF ITERATIONS
  ;;INCREMENT THE PASS NUMBER
  ;;DON'T ALLOW A NEG. NUMBER
  ;;LOOP?
  ;;YES
  ;;RESTORE COUNTER
  ;;TYPE "END PASS #"
  ;;SAVE $PASS FOR TYPEOUT
  ;;GO TYPE--DECIMAL ASCII WITH SIGN
  ;;TYPE A NULL CHARACTER
  ;;GET MONITOR ADDRESS
  ;;BRANCH IF NO MONITOR
  ;;CLEAR THE WORLD
  ;;GO TO MONITOR
  ;;SAVE ROOM
  ;;FOR
  ;;ACT11
  ;;RETURN
  ;;NULL CHARACTER STRING

```


K09

MAINDEC-11-DZRKK-C
DZRKKC.P11MACY11 27(732) 16-SEP-76 16:00 PAGE 115
GT4RG: ROUTINE FOR GETTING RKCS, RKER, RKDS, RKDA

```

6099          :RKDA TO $REG0, $REG1, $REG2, $REG3 RESPECTIVELY BEFORE
6100          :TYPING OUT AN ERROR MESSAGE.
6101          :CALL: JSR      PC,GT4RG
6102
6103 020406 017737 162122 001170 GT4RG: MOV      BRKDA,$REG3          ;GET RKDA
6104 020414 017737 162102 001166 GT3RG: MOV      BRKDS,$REG2          ;GET RKDS
6105 020422 017737 162076 001164 GT2RG: MOV      BRKER,$REG1          ;GET RKER
6106 020430 017737 162072 001162      MOV      BRKCS,$REG0
6107 020436 000207      RTS      PC
6108
6109
6110
6111
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126
6127
6128
6129
6130
6131
6132
6133
6134
6135
6136
6137
6138
6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150
6151
6152
6153
6154

```

.SBTTL TYERM: SPECIAL ERROR MESSAGE ROUTINE

```

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

```

```

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

```

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

;BDAO, BDA4

```

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

```

```

BDAO:  MOV      RD,-(SP)          ;PUSH RD ONTO THE STACK
        MOV      #$REG3+2,RD      ;SET UP POINTER

```

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 116
 DZRKKC.P11 BDAO. BDA4: BREAK DISK ADDRESS INTO SEC, SUR, CYL, DRIVE

```

6155 020476 000403          BR      BDAR
6156
6157 020500 010046          BDA4:  MOV    RD, -(SP)      ; PUSH RD ONTO THE STACK
6158 020502 012700 001202      MOV    # $REG7+2, RD    ; SET UP POINTER
6159
6160 020506 032777 020000 160424 BDAR:  BIT    #20000, DSWR ; INHIBIT TYPEOUT?
6161 020514 001034          BNE     2$              ; YES, BRANCH TO EXIT POINT
6162
6163 020516 010146          MOV    R1, -(SP)      ; PUSH R1 ON STACK
6164 020520 010246          MOV    R2, -(SP)      ; PUSH R2 ON STACK
6165 020522 013701 001202      MOV    $REG10, R1    ; GET THE ADDRESS WHICH
6166                      ; HAS TO BE BROKEN
6167 020526 042701 177760      BIC     #177760, R1    ; EXTRACT SECTOR BITS 0-3
6168 020532 010140          MOV    R1, -(R0)      ; MOVE SECTOR BITS TO $REG3 OR $REG7
6169 020534 013701 001202      MOV    $REG10, R1    ; GET THE DSK-ADRES TO BE BROKEN
6170 020540 006201          ASR     R1              ; SHIFT RIGHT 4 TIMES
6171 020542 006201          ASR     R1
6172 020544 006201          ASR     R1
6173 020546 006201          ASR     R1
6174 020550 010102          MOV    R1, R2          ; STORE THIS
6175 020552 042702 177776      BIC     #177776, R2    ; EXTRACT THE SURFACE BIT
6176 020556 010240          MOV    R2, -(R0)      ; MOVE SURFACE BIT TO $REG3 OR $REG6
6177 020560 006201          ASR     R1
6178 020562 010102          MOV    R1, R2          ; STORE IT
6179 020564 042702 177400      BIC     #177400, R2    ; EXTRACT THE CYLINDER BITS
6180 020570 010240          MOV    R2, -(R0)      ; MOVE CYLINDER BITS TO $REG1 OR $REG5
6181 020572 000301          SWAB   R1              ; SWAB HI-LO BYTES
6182 020574 042701 177770      BIC     #177770, R1    ; EXTRACT THE DRIVE #
6183 020600 010140          MOV    R1, -(R0)      ; MOVE DRIVE # TO $REG0 OR $REG4
6184
6185 020602 012602          MOV    (SP)+, R2      ; RESTORE R2
6186 020604 012601          MOV    (SP)+, R1      ; RESTORE R1
6187 020606 012600          2$:    MOV    (SP)+, R0  ; RESTORE RD FROM THE STACK
6188 020610 000002          RTI      ; RETURN FROM INTERRUPT, EXIT THIS
6189                      ; ROUTINE
6190
6191
6192
6193
6194
6195
6196
6197
6198
6199
6200

```

.SBTTL SHFTRT: SHIFT RIGHT ROUTINE

```

6195          ; SHFTRT
6196          ; THIS ROUTINE SHIFTS A WORD TO THE RIGHT 13 TIMES. THE WORD TO BE SHIFTED
6197          ; IS PUT ON THE STACK BEFORE ENTERING THIS ROUTINE AND IT IS POPPED UP
6198          ; FROM THE STACK AFTER THE SHIFT HAS BEEN DONE.
6199          ; CALL: JSR      PC, SHFTRT
6200
6201 020612 012737 177763 020636 SHFTRT: MOV    #-15, 2$    ; SET UP A COUNT OF 13
6202 020620 000241          CLC                    ; CLEAR THE C BIT
6203 020622 006066 000002      1$:    ROR     2(R6)      ; ROTATE RIGHT THE WORD TO B SHFTD
6204 020626 005237 020636      INC     2$              ; SHIFTED 13 TIMES?
6205 020632 001373          BNE     1$              ; IF NOT LUP BAK & SHIFT
6206 020634 000207          RTS     PC              ; EXIT FROM THIS SUBROUTINE
6207 020636 000000      2$:    0
6208
6209
6210

```

```

6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263
6264
6265
6266

```

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

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

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

020640	010137	001202	CHKHE1: MOV	R1,\$REG10	;SAVE THE DISK ADRES
020644	000403		BR	CHKHE1	
020646	013737	002544	CHKHE: MOV	DRIVAD,\$REG10	;SAVE THE DISK ADRES
020654	032777	140000	CHKHE1: BIT	#140000,\$RKCS	;IS 'HE' OR 'ERR' BIT SET?
020662	001467		BEQ	CRETRN	;NO
020664	004737	020406	JSR	PC,GT4RG	;GET RKCS,ER,DS,DA
020670	104415		BRKDA4		;GO TO 'BDA4' & BREAK CONTENTS 0
020672	000207		RTS	PC	;SREG10 INTO DR#,CYL,SUR,SEC BITS
					;RETURN TO THE ERROR MESSAGE

.SBTTL CHKDA: CHECK IF RKDA INCREMENTED CORRECTLY

;CHKDA
 ;THIS ROUTINE CHECKS IF RKDA INCREMENTED CORRECTLY. IF RKDA INCREMENTED
 ;CORRECTLY RETURN IS MADE TO SKIP THE ERROR MESSAGE.
 ;IF RKDA DID NOT INCREMENT CORRECTLY, THE EXPECTED AND RECIEVED VALUES
 ;OF RKDA ARE SAVED AND A RETURN IS MADE TO THE ERROR MESSAGE FOLLOWING THE
 ;'JSR' CALL.

020674	013705	002544	CHKDA: MOV	DRIVAD,R5	;RKDA SHOULD INCREMENT TO THIS
020700	005205		INC	R5	;AFTER DATA TRANSFER IS DONE
020702	020577	161626	CHKDA1: CMP	R5,\$RKDA	;DID RKDA INCREMENT CORRECTLY?
020706	001455		BEQ	CRETRN	;IF YES, BRANCH
020710	010537	001202	MOV	R5,\$REG10	;IF NOT, REPORT ERROR
020714	104414		BRKDA0		;GET EXPCD RKDA
020716	017737	161612	MOV	\$RKDA,\$REG10	;GO TO 'BDA0' & BREAK CONTENTS OF
020724	104415	001202	BRKDA4		;SREG10 INTO DR #,CYL,SUR,SEC BITS
020726	000207		RTS	PC	;GET ACTUAL RKDA
					;GO TO 'BDA4' & BREAK CONTENTS OF
					;SREG10 INTO DR #,CYL,SUR,SEC BITS
					;RETURN TO THE ERROR MESSAGE

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 118
 DZRKKC.P11 CHKDA: CHECK IF RKDA INCREMENTED CORRECTLY

```

6267
6268
6269          .SBTTL  CHKWC:  CHECK IF RKWC OVERFLOWED
6270
6271          ;CHKWC
6272          ;THIS ROUTINE CHECKS IF RKWC OVERFLOWED TO 0. IF IT DID A RETURN IS MADE
6273          ;TO SKIP THE ERROR MESSAGE. IF NOT, THE CONTENTS OF RKWC AND RKDA ARE SAVED
6274          ;AND A RETURN IS MADE TO THE ERROR MESSAGE FOLLOWING THE 'JSR' CALL.
6275 020730 005777 161574      CHKWC:  TST      @RKWC          ;DID WORD COUNT OVERFLOW TO 0?
6276 020734 001442          BEQ      CRETRN          ;IF YES, BRANCH
6277          ;IF NOT, ERROR
6278 020736 017737 161566 001162      MOV      @RKWC,$REG0      ;GET RKWC
6279 020744 017737 161564 001164      MOV      @RKDA,$REG1      ;GET RKDA
6280 020752 000207          RTS      PC          ;RETURN TO THE ERROR MESSAGE
6281
6282          .SBTTL  CHKER:  CHECK RKER CONTENTS
6283
6284          ;CHKER
6285          ;THIS ROUTINE CHECKS IF ANY BIT IN RKER SET. IF NOT RETURN IS MADE TO SKIP
6286          ;THE ERROR MESSAGE. IF ANY BIT IS SET THE CONTENTS OF RKCS, RKER, RKDS ARE
6287          ;SAVED AND A RETURN IS MADE TO THE ERROR MESSAGE.
6288
6289 020754 005777 161544      CHKER:  TST      @RKER          ;DID ANY BIT IN RKER SET?
6290 020760 001430          BEQ      CRETRN          ;NO, BRANCH
6291          ;YES, ERROR
6292 020762 004737 020414      JSR      PC,GT3RG      ;GO, GET RKCS, ER, DS
6293
6294 020766 000207          RTS      PC          ;RETURN TO THE ERROR MESSAGE
6295
6296          ;CHKECLR
6297          ;THIS ROUTINE CHECKS THAT RKER IS CLEAR. IF NOT, THE CONTENTS OF RKER
6298          ;ARE SAVED AND A RETURN IS MADE TO THE ERROR MESSAGE FOLLOWING THE "JSR"
6299          ;CALL. IF RKER IS CLEAR THE ERROR MESSAGE IS SKIPPED ON RETURN.
6300
6301 020770 005777 161530      CHKECLR: TST      @RKER          ;ANY BIT IN RKER SET?
6302 020774 001422          BEQ      CRETRN          ;NO
6303 020776 013737 002524 001162      MOV      RKER,$REG0      ;GET ADRES OF RKER
6304 021004 017737 161514 001164      MOV      @RKER,$REG1      ;GET CONTENTS OF RKER
6305 021012 000207          RTS      PC          ;RETURN TO THE ERROR MESSAGE
6306
6307          ;CHKCCLR
6308          ;THIS ROUTINE CHECKS THAT RKCS IS CLEAR. IF NOT, THE CONTENTS OF RKCS ARE
6309          ;SAVED AND A RETURN IS MADE TO THE ERROR MESSAGE. IF RKCS IS CLEAR THE
6310          ;ERROR MESSAGE IS SKIPPED ON RETURN.
6311
6312 021014 022777 000200 161504      CHKCCLR: CMP      #200,@RKCS      ;IS RKCS CLEAR?
6313 021022 001407          BEQ      CRETRN          ;YES
6314 021024 013737 002526 001162      MOV      RKCS,$REG0      ;SAVE ADRES OF RKCS
6315 021032 017737 161470 001164      MOV      @RKCS,$REG1      ;SAVE THE CONTENT OF RKCS
6316 021040 000207          RTS      PC          ;RETURN TO THE ERROR MESSAGE
6317
6318 021042 062716 000002      CRETRN: ADD      #2,(SP)          ;SKIP ERROR MESSAGE ON
6319 021046 000207          RTS      PC          ;RETURN
6320
6321
6322

```

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

:TSTRWS

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

6333	021050	013777	002544	161456	TSTRWS: MOV	DRIVAD, DRKDA	:ADRES THE DRIVE
6334	021056	005037	002562		CLR	TIMER	:INITIALIZE COUNT
6335	021062	032777	000100	161432	15: BIT	#100, DRKDS	:DID R/W/S RDY SET?
6336	021070	001007			BNE	25	:YES, BRANCH
6337	021072	005237	002562		INC	TIMER	:WAIT FOR R/W/S RDY
6338	021076	001371			BNE	15	:ERROR IF IT'S NOT SET BY NOW
6339	021100	017737	161416	001162	MOV	DRKDS, \$REGO	:GET RKDS
6340	021106	000207			RTS	PC	:EXIT (TO ERROR FOLLOWING 'JSR TSTRWS')
6341							
6342	021110	062716	000002		25: ADD	#2, (SP)	:ADJUST RETURN ADRES TO SKIP OVER
6343							:ERROR (FOLLOWING 'JSR TSTRWS')
6344	021114	000207			RTS	PC	:EXIT
6345							
6346							
6347							
6348							
6349							

.SBTTL DRESET: DRIVE RESET ROUTINE

:DRESET

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

6363	021116	005037	002560		DRESET: CLR	COUNT1	:INITIALIZE THE COUNT
6364	021122	013777	002544	161404	MOV	DRIVAD, DRKDA	:ADRES THE DRIVE
6365	021130	012777	000015	161370	MOV	#15, DRKCS	:DRIVE RESET, GO
6366	021136	104413			CNT, RDY		:THIS IS A CALL FOR 'CN, RDY'
6367							:ROUTINE WHICH WAITS FOR CNT
6368							:RDY TO SET. IF CNTRL RDY DOES
6369							:NOT SET WITHIN 883 MS/ 11-20
6370							: (176 MS FOR 11-45 WITH BIPOLAR)
6371							:AN ERROR IS REPORTED
6372	021140	032777	000100	161354	15: BIT	#100, DRKDS	:DID R/W/S RDY SET?
6373	021146	001013			BNE	25	
6374	021150	012746	177770		MOV	#-10, -(SP)	:PUSH COUNT ON SP
6375	021154	005216			INC	(SP)	:COUNT IT DOWN
6376	021156	001376			BNE	-2	
6377	021160	005726			TST	(SP)+	:POP UP SP
6378	021162	005237	002560		INC	COUNT1	:IF NOT WAIT

MAINDEC-11-DZRAK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 120
 DZRAK.C.P11 DRESET: DRIVE RESET ROUTINE

```

6379 021165 001364      BNE      1$              ;WAITED LONG?
6380 021170 004737 020406 JSR      PC,GT4RG
6381 021174 000402      BR       2$+4
6382 021176 062716 000002 2$: ADD     #2,AR6
6383 021202 000207      RTS      PC
6384
6385
6386
6387
6388      .SBTTL  TSTSIN: CHECK 'SIN' ROUTINE
6389
6390      :TSTSIN
6391      :THIS ROUTINE CHECKS IF 'SIN' IS SET. IF IT IS SET A
6392      :DRIVE RESET IS DONE TO CLEAR 'SIN' AND INITIALIZE POSITIONER.
6393      :CALL:      TST.SIN
6394      :IF ON DOING DRIVE RESET R/W/S RDY DOES NOT SET A MESSAGE
6395      :      ERROR PC=XXXXXX IS GIVEN.
6396      :XXXXXX=PC IN THE MAIN PROGRAM WHERE 'TST.SIN' CALL IS LOCATED.
6397
6398 021204 013777 002544 161322 TSTSIN: MOV     DRIVAD,ARKDA      ;ADRES THE DRIVE
6399 021212 032777 001000 161302      BIT      #1000,ARKDS      ;IS SIN SET?
6400 021220 001403      BEQ      1$
6401 021222 004737 021116      JSR      PC,DRESET      ;GO DO DRIVE RESET, SIN SET
6402 021226 000401      BR       2$      ;REPORT ERROR
6403 021230 000002      1$: RTI
6404 021232 032777 020000 157700 2$: BIT      #SW13,ASWR      ;INHIBIT TYPEOUT?
6405 021240 001373      BNE      1$      ;IF YES, SKIP TYPEOUT
6406 021242 104400 021250      TYPE     #55$      ;TYPE ASCII STRING
6407 021246 000406      BR       64$      ;GET OVER THE ASCII
6408      ::65$: .ASCIIZ /ERROR PC= /
6409      64$:
6410 021264      MOV     (SP),-(SP)
6411 021266 011646      ADD     #-2,(SP)      ;GET THE PC WHERE 'TST.SIN' IS LOCATED
6412 021272 062716 177776      TYP0C     ;GO TYPE OUT PC
6413 021274 104401      BR       1$
6414
6415
6416
6417      .SBTTL  DELAY: TIME DELAY ROUTINE
6418
6419      :DELAY
6420      :THIS ROUTINE PROVIDES A VARIABLE TIME DELAY. THE CALL FOR THIS
6421      :ROUTINE IS AN ENCODED 'TRAP' INSTRUCTION.
6422      :CALL:      DELAY,N      N IS ANY OCTAL NO. FROM 1 TO 177777
6423      :THE DELAY PROVIDED IS 7.5N US (CONVERT N TO DECIMAL) FOR 11/20
6424      :1.5N US FOR 11/45
6425      :IF THE USER WANTS TO CHANGE THE DELAY TIME (EXMP: SHORTER DELAY TO
6426      :GET A TIGHTER SCOPE LOOP) THE VARIABLE 'N' FOLLOWING 'DELAY' SHOULD
6427      :BE CHANGED TO SUIT THE INDIVIDUAL NEED.
6428
6429 021276 017637 000000 002562 DELAY: MOV     2(SP),TIMER      ;GET 'AMOUNT' (N) FOR WHICH
6430 021304 062716 000002      ADD     #2,(SP)      ;DELAY IS TO BE PROVIDED
6431      ;ADJUST STACK POINTER TO SKIP OVER 'N'
6432 021310 005337 002562      1$: DEC     TIMER      ;COUNT DOWN TO 0
6433 021314 001375      BNE      1$
6434

```


6435 021316 000002

RTI

;RETURN TO MAIN PROGRAM

6436

6437

6438

6439

6440

6441

6442

6443

6444

6445

6446

6447

6448

6449

6450

6451

6452

6453

6454

6455

6456

6457

021320 017637 000000

002562

WATINT: P

2(SP),TIMER

;GET 'AMOUNT' (N) FOR WHICH

6458

021326 062716 000002

AL

#2,(SP)

;WAITING IS TO BE DONE

6459

021332 013746 002574

MOV

RKPRI,-(SP)

;ADJUST STACK POINTER FOR CORRECT RETURN

6460

021336 012746 021344

MOV

#1\$,-(SP)

;DROP CPU PRIORITY SO THAT RK11 CAN

6461

021342 000002

RTI

; INTERRUPT

6462

021344 005337 002562

1\$:

DEC

TIMER

;WAIT FOR RK11 TO INTERRUPT

6463

021350 001375

BNE

1\$

;IF INTERRUPT HAS NOT OCCURED BY NOW

6464

021352 000002

RTI

;RETURN AND REPORT ERROR

6465

6466

6467

6468

6469

6470

6471

6472

;WATIME

6473

021354 005000

WATIME: CLR

R0

6474

021356 005001

CLR

R1

6475

021360 005200

1\$:

INC

R0

6476

021362 001376

BNE

1\$

6477

021364 105201

INCB

R1

6478

021366 001374

BNE

1\$

6479

021370 000207

RTS

PC

6480

6481

6482

6483

6484

6485

6486

6487

6488

6489

6490

.SBTTL CHKCRDY:

CHECK CONTROL READY

;CH.CRDY

 ;THIS ROUTINE WAITS FOR THE CONTROL READY TO SET. IF THE CONTROL READY BIT
 ;DOES NOT SET WITHIN A CERTAIN TIME, THEN THE CONTENTS OF RKCS, RKER, RKDS
 ;AND RKDA ARE SAVED AND AN EXIT MADE TO THE ERROR MESSAGE FOLLOWING THE
 ;'JSR' CALL FOR THIS ROUTINE.

 ;IF CONTROL READY SETS THEN THE RETURN ADDRESS IS ADJUSTED TO SKIP THE
 ;ERROR MESSAGE ON RETURN.

E10

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 122
 DZRKKC.P11 CHKCRDY: CHECK CONTROL READY

```

6491      :CALL:  CHKCRDY
6492      :      ERROR
6493      :      ---
6494      :      :RETURN HERE IF ERROR
6495      :      :RETURN HERE IF NO ERROR
6496      021372 005037 002562      CH.CRDY: CLR TIMER
6497      021376 105777 161124      15:  TSTB  3RKCS      :CNTRL RDY SET?
6498      021402 100406      :      :YES
6499      021404 005237 002562      :      9MI  25
6500      021410 001372      :      INC  TIMER
6501      021412 004737 020406      :      BNE  15      :NO, WAIT
6502      021416 000002      :      JSR  PC,GT4RG      :SAVE RKCS, ER, DS, DA
6503      :      RTI
6504      021420 062716 000002      25:  ADD  #2,(SP)      :ADJUST RETURN ADDRESS TO
6505      021424 000002      :      RTI      :SKIP ERROR MESSAGE ON RETURN
6506
6507      .SBTTL  CON.RESET:      CONTROL REST ROUTINE
6508
6509      :CON.RESET
6510      :THIS ROUTINE ISSUES A CONTROL RESET AND WAITS FOR
6511      :THE 'CNTRL RDY' FLAG TO SET. WHEN THE FLAG SETS
6512      :AN EXIT IS MADE OUT OF THE ROUTINE. IF 'CNTRL-RDY'
6513      :DOES NOT SET WITHIN A CERTAIN TIME AN ERROR MESSAGE
6514      :CNT RDY DIDN'T SET
6515      :PC=XXXXXX RKCS=YYYYYY
6516      :IS GIVEN. NOTE THAT XXXXXX IS THE PC WHERE 'CNT.RESET' OR 'CNT.RDY'
6517      :IS CALLED.
6518
6519      :CALL:  CNT.RESET
6520
6521
6522
6523
6524      .SBTTL  CNT.RDY:      WAIT FOR CONTROL READY ROUTINE
6525
6526      :CN.RDY
6527      :THIS ROUTINE WAITS FOR THE CONTROL READY BIT TO SET AND WHEN IT
6528      :SETS EXITS OUT. IF WITHIN A CERTAIN TIME CNTRL RDY DOES
6529      :NOT SET AN ERROR IS REPORTED. WAITING TIME IS 893 MS FOR 11/20
6530      :175 MS FOR 11/45 WITH BIPOLAR MEMORY.
6531      :CALL:  CNT.RDY
6532      021426 012777 000001 161072  CN.RST: MOV  #1,3RKCS      :ISSUE A CONTROL RESET
6533      021434 012737 177500 001170  :      MOV  #300,$REG3      :SET UP COUNT
6534      021442 000402      :      BR   CN.RDY+4      :SKIP OVER CN.RDY
6535      021444 005037 001170      :      CLR  $REG3
6536      021450 105777 161052      15:  TSTB  3RKCS      :DID CNTRL-RDY SET?
6537      021454 100435      :      BMI  35      :YES, EXIT
6538      021456 005237 001170      :      INC  $REG3      :WAITED LONG?
6539      021462 001372      :      BNE  15      :IF NOT, GO BAK & WAIT
6540      021464 032777 020000 157446  25:  BIT  #SW13,$SWR      :INHIBIT TIMEOUT?
6541      021472 001026      :      BNE  35      :IF YES, SKIP TIMEOUT
6542      021474 104400      :      TYPE MSG3
6543      021476 002441      :      TYPE 65$      :TYPE ASCIZ STRING
6544      021500 104400 021506      :      BR   64$      :GET OVER THE ASCIZ
6545      021504 000403      :      :
6546      :;65$: .ASCIZ  <15><12>/PC=

```


MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 123
 DZRKKC.P11 CNT.RDY: WAIT FOR CONTROL READY ROUTINE

```

6547 021514          645:      MOV      (SP), -(SP)
6548 021514 011646      SUB      #2, (SP)
6549 021516 162716 000002      TPOC
6550 021522 104401          ;GO TYPE PC IN THE MAIN PROGRAM.
6551          ;WHERE ERROR OCCURRED
6552 021524 10440C 021532      TYPE      675
6553 021530 000404          ;TYPE ASCIZ STRING
6554          ;GET OVER THE ASCIZ
6555 021542          ;:675: .ASCIZ / RKCS=/
6556 021542 017746 160760 665:      MOV      @RKCS, -(SP) ;GET RKCS
6557 021546 104401          ;GO TYPE IT
6558          TPOC
6559 021550 000002      35:      RTI ;RETURN FROM THIS
6560          ;ROUTINE TO THE MAIN
6561          ;PROGRAM
6562
6563
6564 ;THIS PART OF THE PROGRAM CONTAINS THE COMMON ROUTINES CALLED
6565 ;FROM THE SYSMAC.SML PACKAGE
6566 ;
6567
6568 .SBTTL SCOPE HANDLER ROUTINE
6569
6570 ;*****
6571 ;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
6572 ;AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
6573 ;AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
6574 ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
6575 ;*SW14=1 LOOP ON TEST
6576 ;*SW11=1 INHIBIT ITERATIONS
6577 ;*SW09=1 LOOP ON ERROR
6578 ;*SW08=1 LOOP ON TEST IN SWR<7:0>
6579 ;*CALL
6580 ;* SCOPE ;;SCOPE=IOT
6581
6582 021552          $SCOPE:
6583 021552 104406          CKSWR
6584 021554 032777 040000 157356 15:      BIT      #BIT14, @SWR ;;TEST FOR CHANGE IN SOFT-SWR
6585 021562 0C1111          BNE      $OVER ;;LOOP ON PRESENT TEST?
6586          ;*****START OF CODE FOR THE XOR TESTER***** ;;YES IF SW14=1
6587 021564 000416          $XTSTR: BR      65 ;;IF RUNNING ON THE "XOR" TESTER CHANGE
6588          ;THIS INSTRUCTION TO A "NOP" (NOP=240)
6589 021566 013746 000004          MOV      @ERRVEC, -(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
6590 021572 012737 021612 000004          MOV      #55, @ERRVEC ;;SET FOR TIMEOUT
6591 021600 005737 177060          TST      @177060 ;;TIME OUT ON XOR?
6592 021604 012637 000004          MOV      (SP)+, @ERRVEC ;;RESTORE THE ERROR VECTOR
6593 021610 000463          BR      $SVLAD ;;GO TO THE NEXT TEST
6594 021612 022626          55:      CMP      (SP)+, (SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
6595 021614 012637 000004          MOV      (SP)+, @ERRVEC ;;RESTORE THE ERROR VECTOR
6596 021620 000423          BR      75 ;;LOOP ON THE PRESENT TEST
6597 021622          65: ;*****END OF CODE FOR THE XOR TESTER*****
6598 021622 032777 000400 157310          BIT      #BIT08, @SWR ;;LOOP ON SPEC. TEST?
6599 021630 001404          BEQ      25 ;;BR IF NO
6600 021632 127737 157302 001102          CMPB   @SWR, $STNM ;;ON THE RIGHT TEST? SWR<7:0>
6601 021640 001462          BEQ      $OVER ;;BR IF YES
6602 021642 105737 001103          25:      TSTB   $ERFLG ;;HAS AN ERROR OCCURRED?

```

MAINDEC-11-DZK-K-C MACY11 27.(732) 16-SEP-76 16:00 PAGE 124
DZK-KC.P11 SCOPE HANDLER ROUTINE

```

6603 021646 001421      BEQ      3$      ;;BR IF NO
6604 021650 123737 001115 001103      CMPB     $ERMAX,$ERFLG  ;;MAX. ERRORS FOR THIS TEST OCCURRED?
6605 021656 101015      BHI      3$      ;;BR IF NO
6606 021660 032777 001000 157252      BIT      #BIT09,$SWR  ;;LOOP ON ERROR?
6607 021666 001404      BEQ      4$      ;;BR IF NO
6608 021670 013737 001110 001106 7$:  MOV      $LPERR,$LPADR  ;;SET LOOP ADDRESS TO LAST SCOPE
6609 021676 000443      BR       $OVER
6610 021700 105037 001103      4$:  CLRB     $ERFLG      ;;ZERO THE ERROR FLAG
6611 021704 005037 001206      CLR      $TIMES      ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
6612 021710 000415      BR       1$      ;;ESCAPE TO THE NEXT TEST
6613 021712 032777 004000 157220 3$:  BIT      #BIT11,$SWR  ;;INHIBIT ITERATIONS?
6614 021720 001011      BNE      1$      ;;BR IF YES
6615 021722 005737 001100      TST      $PASS      ;;IF FIRST PASS OF PROGRAM
6616 021726 001406      BEQ      1$      ;;INHIBIT ITERATIONS
6617 021730 005237 001104      INC      $ICNT      ;;INCREMENT ITERATION COUNT
6618 021734 023737 001206 001104      CMP      $TIMES,$ICNT  ;;CHECK THE NUMBER OF ITERATIONS MADE
6619 021742 002021      BGE      $OVER      ;;BR IF MORE ITERATION REQUIRED
6620 021744 012737 000001 001104 1$:  MOV      #1,$ICNT      ;;REINITIALIZE THE ITERATION COUNTER
6621 021752 013737 022022 001206      MOV      $MXCNT,$TIMES  ;;SET NUMBER OF ITERATIONS TO DO
6622 021760 105237 001102      SSVLAD: INCB     $TSTNM      ;;COUNT TEST NUMBERS
6623 021764 011637 001106      MOV      (SP),$LPADR  ;;SAVE SCOPE LOOP ADDRESS
6624 021770 011637 001110      MOV      (SP),$LPERR  ;;SAVE ERROR LOOP ADDRESS
6625 021774 005037 001210      CLR      $ESCAPE  ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
6626 022000 112737 000001 001115      MOVB     #1,$ERMAX  ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
6627 022006 013777 001102 157126 $OVER: MOV      $TSTNM,$DISPLAY  ;;DISPLAY TEST NUMBER
6628 022014 013716 001106      MOV      $LPADR,(SP)  ;;FUDGE RETURN ADDRESS
6629 022020 000002      RTI
6630 022022 000050      $MXCNT: 50      ;;FIXES PS
6631                                     ;;MAX. NUMBER OF ITERATIONS
6632
6633                                     ;;*****
6634
6635 .SBTTL ERROR HANDLER ROUTINE
6636
6637 ;;$SW15=1      HALT ON ERROR
6638 ;;$SW13=1      INHIBIT ERROR TYPEOUTS
6639 ;;$SW10=1      TESTING ON SIMULATOR
6640 ;;$SW09=1      LOOP ON ERROR
6641 ;;$SW12=1      CYCLE ON ERROR TO PREVIOUS 'SCOPE'
6642 ;;$SW06=1      DROP DRIVE AFTER MAXIMUM (ALLOWABLE) ERRORS ON THE DRIVE
6643 ;;*GO TO $ERRTYP ON ERROR
6644
6645 $ERROR:
6646
6647 022024 105237 001103      7$:  INCB     $ERFLG      ;;SET THE ERROR FLAG
6648 022030 001775      BEQ      7$      ;;DON'T LET THE FLAG GO TO ZERO
6649 022032 013777 001102 157102      MOV      $TSTNM,$DISPLAY  ;;DISPLAY TEST NUMBER AND ERROR FLAG
6650 022040 005237 001112      1$:  INC      $ERTTL      ;;COUNT THE NUMBER OF ERRORS
6651
6652 022044 032777 000100 157066      BIT      #BIT6,$SWR  ;;DESELECT DRIVE SW SET?
6653 022052 001404      BEQ      6$      ;;NO
6654 022054 023727 001112 000005      CMP      $ERTTL,#5      ;;MORE THAN 5 ERRORS ON THIS DRIVE?
6655 022062 101046      BHI      8$      ;;YES, DESELECT THE DRIVE
6656
6657 022064 011637 001116      6$:  MOV      (SP),$ERRPC  ;;GET ADDRESS OF ERROR INSTRUCTION
6658 022070 162737 000002 001116      SUB      #2,$ERRPC

```

```

6659 022076 117737 157014 001114      MOVB    @SERRPC,$ITEMB      ;STRIP AND SAVE THE ERROR ITEM CODE
6660 022104 032777 020000 157026      BIT     #SW13,@SWR        ;SKIP TYPEOUT IF SET
6661 022112 001004                BNE     2$                ;SKIP TYPEOUTS
6662 022114 004737 022332      JSR     PC,@$SERRTYP      ;GO TO USER ERROR ROUTINE
6663 022120 104400 001213      TYPE    $CRLF
6664 022124 005777 157010      2$:    TST     @SWR
6665 022130 100001                BPL     3$
6666 022132 000000      HALT
6667 022134 032777 010000 156776 3$:    BIT     #SW12,@SWR      ;SW 12 SET?
6668 022142 001402                BEQ     .+6                ;NO BRANCH
6669 022144 013716 001106      MOV     $LPADR,(SP)      ;ADJUST RETURN ADRES FOR SW12
6670 022150 032777 001000 156762      BIT     #SW09,@SWR      ;LOOP ON ERROR SWITCH SET?
6671 022156 001402                BEQ     4$
6672 022160 013716 001110      MOV     $LPERR,(SP)      ;FUDGE RETURN FOR LOOPING
6673 022164 005737 001210      4$:    TST     $ESCAPE      ;CHECK FOR AN ESCAPE ADDRESS
6674 022170 001402                BEQ     5$
6675 022172 013716 001210      MOV     $ESCAPE,(SP)      ;FUDGE RETURN ADDRESS FOR ESCAPE
6676 022176 000002      5$:    RTI
6677
6678 022200 005737 002630      8$:    TST     T56FLG      ;IF EROR WAS IN LAST TEST (POLL)
6679
6680 022204 001407                BEQ     10$
6681 022206 104400 002477      TYPE    MSG5
6682 022212 005037 002602      CLR     DRIVS
6683 022216 022626      CMP     (SP)+,(SP)+
6684 022220 000137 020264      JMP     $EOP
6685 022224 013746 002550      10$:   MOV     DRVPTR,-(SP)      ;DROP THE DRIVE FROM THE
6686 022230 162716 000002      SUB     #2,(SP)      ;SELECTION LIST
6687 022234 013746 002544      MOV     DRIVAD,-(SP)      ;DRIVE ADDR TO STACK
6688 022240 004737 020612      JSR     PC,$HFTRT      ;RIGHT JUSTIFY
6689 022244 042716 000001      BIC     #1,(R6)      ;MAKE EVEN
6690 022250 062716 002610      ADD     #DRIV0,(SP)      ;POINTS TO TABLE FOR EVEN DRIVE
6691 022254 042776 100000 000000      BIC     #BIT15,@(R6)      ;TEST REMAINING DRIVE AS RK05E
6692 022262 062716 000002      ADD     #2,(R6)      ;POINT TO ODD
6693 022266 042736 100000      BIC     #BIT15,@(SP)+      ;TEST AS RK-05E
6694 022272 012736 010000      MOV     #BIT12,@(SP)+      ;INDICATE THIS DRIVE DROPPED
6695 022276 104400 002466      TYPE    MSG4
6696 022302 013746 002544      MOV     DRIVAD,-(R6)      ;PUSH DRIVE # ON STACK
6697 022306 004737 020612      JSR     PC,$HFTRT      ;SHIFT IT BEFORE TYPING
6698 022312 104401      TYP0C
6699 022314 104400 002511      TYPE    MSG6
6700 022320 005337 002602      DEC     DRIVS      ;DECREMENT # OF DRIVES PRESNT
6701 022324 022626      9$:    CMP     (SP)+,(SP)+      ;RESTORE STACK
6702 022326 000137 017440      JMP     BTEOP      ;GO BACK TO THE END OF PROGRM
6703
6704
6705
6706
6707
6708
6709
6710
6711
6712 022332
6713 022332 104400 001213      $ERRTYP: TYPE    $CRLF      ;;"CARRIAGE RETURN" & "LINE FEED"
6714 022336 010046      MOV     R0,-(SP)      ;;SAVE R0

```

.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

```

;*****
;THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
;ERROR IS TO BE REPORTED. IT THEN OBTAINS FROM THE "ERROR TABLE" ($ERRTB),
;AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE EPOR.

```

```

$ERRTYP:
        TYPE    $CRLF      ;;"CARRIAGE RETURN" & "LINE FEED"
        MOV     R0,-(SP)    ;;SAVE R0

```

```

6715 022340 005000          CLR      RO          ;;PICKUP THE ITEM INDEX
6716 022342 153700 001114   BISB     2*ITEMB,RO
6717 022346 001004          BNE      1$          ;; IF ITEM NUMBER IS ZERO, JUST
6718                                ;; TYPE THE PC OF THE ERROR
6719 022350 013746 001116   MOV      $ERRPC, -(SP) ;;SAVE $ERRPC FOR TYPEOUT
6720                                ;;ERROR ADDRESS
6721 022354 104401          TYP0C          ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
6722 022356 000426          BR        6$          ;;GET OUT
6723 022360 005300          1$: DEC      RO          ;;ADJUST THE INDEX SO THAT IT WILL
6724 022362 006300          ASL      RO          ;;WORK FOR THE ERROR TABLE
6725 022364 006300          ASL      RO
6726 022366 006300          ASL      RO
6727 022370 062700 001216   ADD      *$ERRTB, RO ;;FORM TABLE POINTER
6728 022374 012037 022404   MOV      (RO)+, 2$    ;;PICKUP "ERROR MESSAGE" POINTER
6729 022400 001404          BEQ      3$          ;;SKIP TYPEOUT IF NO POINTER
6730 022402 104400          TYPE          ;;TYPE THE "ERROR MESSAGE"
6731 022404 000000          2$: .WORD    0          ;;"ERROR MESSAGE" POINTER GOES HERE
6732 022406 104400 001213   TYPE      $CRLF      ;; "CARRIAGE RETURN" & "LINE FEED"
6733 022412 012037 022422   3$: MOV      (RO)+, 4$    ;;PICKUP "DATA HEADER" POINTER
6734 022416 001404          BEQ      5$          ;;SKIP TYPEOUT IF 0
6735 022420 104400          TYPE          ;;TYPE THE "DATA HEADER"
6736 022422 000000          4$: .WORD    0          ;; "DATA HEADER" POINTER GOES HERE
6737 022424 104400 001213   TYPE      $CRLF      ;; "CARRIAGE RETURN" & "LINE FEED"
6738 022430 011000          5$: MOV      (RO), RO    ;;PICKUP "DATA TABLE" POINTER
6739 022432 001004          BNE      7$          ;;GO TYPE THE DATA
6740 022434 012600          6$: MOV      (SP)+, RO    ;;RESTORE RO
6741 022436 104400 001213   TYPE      $CRLF      ;; "CARRIAGE RETURN" & "LINE FEED"
6742 022442 000207          RTS      PC          ;;RETURN
6743 022444          7$:
6744 022444 013046          MOV      2(RO)+, -(SP) ;;SAVE 2(RO)+ FOR TYPEOUT
6745 022446 104401          TYP0C          ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
6746 022450 005710          TST      (RO)          ;;IS THERE ANOTHER NUMBER?
6747 022452 001770          BEQ      6$          ;;BR IF NO
6748 022454 104400 022462   TYPE      8$          ;;TYPE TWO(2) SPACES
6749 022460 000771          BR        7$          ;;LOOP
6750 022462 020040 000      8$: .ASCIZ  / /          ;;TWO(2) SPACES
6751 022466
6752
6753 .SBTTL TYPE ROUTINE
6754
6755 ;;*****
6756 ;;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
6757 ;;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
6758 ;;NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
6759 ;;NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
6760 ;;NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
6761 ;;
6762 ;;CALL:
6763 ;;1) USING A TPCP INSTRUCTION
6764 ;; TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
6765 ;;OR
6766 ;; TYPE
6767 ;; MESADR
6768 ;;
6769
6770 022466 105737 001157   $TYPE: TSTB     $TPFLG          ;;IS THERE A TERMINAL?

```

```

6771 022472 100002      BPL      1$      ;;BR IF YES
6772 022474 000000      HALT      ;;HALT HERE IF NO TERMINAL
6773 022476 000407      BR        3$      ;;LEAVE
6774 022500 010046      1$: MOV     RO, -(SP)  ;;SAVE RO
6775 022502 017600 000002  MOV     32(SP), RO  ;;GET ADDRESS OF ASCIZ STRING
6776 022506 112046      2$: MOVB    (RO)+, -(SP)  ;;PUSH CHARACTER TO BE TYPED ONTO STACK
6777 022510 001005      BNE      4$      ;;BR IF IT ISN'T THE TERMINATOR
6778 022512 005726      TST      (SP)+  ;;IF TERMINATOR POP IT OFF THE STACK
6779 022514 012600      60$: MOV     (SP)+, RO  ;;RESTORE RO
6780 022516 062716 000002  3$: ADD     #2, (SP)  ;;ADJUST RETURN PC
6781 022522 000002      RTI          ;;RETURN
6782 022524 122716 000011  4$: CMPB    #HT, (SP)  ;;BRANCH IF <HT>
6783 022530 001430      BEQ      8$      ;;BRANCH IF NOT <CRLF>
6784 022532 122716 000200  CMPB    #CRLF, (SP)
6785 022536 001006      BNE      5$      ;;POP <CR><LF> EQUIV
6786 022540 005726      TST      (SP)+  ;;TYPE A CR AND LF
6787 022542 104400      TYPE      $CRLF
6788 022544 001213      $CRLF
6789 022546 105037 022702  CLRB     $CHARCNT  ;;CLEAR CHARACTER COUNT
6790 022552 000755      BR        2$      ;;GET NEXT CHARACTER
6791 022554 004737 022636  5$: JSR     PC, $TYPEPC  ;;GO TYPE THIS CHARACTER
6792 022560 123726 001156  6$: CMPB    $FILLC, (SP)+  ;;IS IT TIME FOR FILLER CHARS.?
6793 022564 001350      BNE      2$      ;;IF NO GO GET NEXT CHAR.
6794 022566 013746 001154  MOV     $NULL, -(SP)  ;;GET # OF FILLER CHARS. NEEDED
6795                                AND THE NULL CHAR.
6796 022572 105366 000001  7$: DECB    1(SP)  ;;DOES A NULL NEED TO BE TYPED?
6797 022576 002770      BLT      6$      ;;BR IF NO--GO POP THE NULL OFF OF STACK
6798 022600 004737 022636  JSR     PC, $TYPEPC  ;;GO TYPE A NULL
6799 022604 105337 022702  DECB    $CHARCNT  ;;DO NOT COUNT AS A COUNT
6800 022610 000770      BR        7$      ;;LOOP
6801
6802      :HORIZONTAL TAB PROCESSOR
6803
6804 022612 112716 000040  8$: MOVB    #' , (SP)  ;;REPLACE TAB WITH SPACE
6805 022616 004737 022636  9$: JSR     PC, $TYPEPC  ;;TYPE A SPACE
6806 022622 132737 000007 022702  BITB    #7, $CHARCNT  ;;BRANCH IF NOT AT
6807 022630 001372      BNE      9$      ;;TAB STOP
6808 022632 005726      TST      (SP)+  ;;POP SPACE OFF STACK
6809 022634 000724      BR        2$      ;;GET NEXT CHARACTER
6810 022636 105777 156306  $TYPEPC: TSTB    32$TPS  ;;WAIT UNTIL PRINTER IS READY
6811 022642 100375      BPL      $TYPEPC
6812 022644 116677 000002 156300  MOVB    2(SP), 32$TPB  ;;LOAD CHAR TO BE TYPED INTO DATA REG.
6813 022652 122766 000015 000002  CMPB    #CR, 2(SP)  ;;IS CHARACTER A CARRIAGE RETURN?
6814 022660 001003      BNE      1$      ;;BRANCH IF NO
6815 022662 105037 022702      CLRB    $CHARCNT  ;;YES--CLEAR CHARACTER COUNT
6816 022666 000406      BR        $TYPEPC  ;;EXIT
6817 022670 122766 000012 000002  1$: CMPB    #LF, 2(SP)  ;;IS CHARACTER A LINE FEED?
6818 022676 001402      BEQ      $TYPEPC  ;;BRANCH IF YES
6819 022700 105227      INCB    (PC)+  ;;COUNT THE CHARACTER
6820 022702 000000      $CHARCNT: .WORD 0  ;;CHARACTER COUNT STORAGE
6821 022704 000207      $TYPEPC: RTS     PC
6822
6823
6824      .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
6825
6826      ;;*****

```

```

6827      ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
6828      ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
6829      ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
6830      ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
6831      ;*REPLACED WITH SPACES.
6832      ;*CALL:
6833      ;*      MOV      NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
6834      ;*      TYPDS      ;;GO TO THE ROUTINE
6835
6836      $TYPDS:
6837      022706      010046      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
6838      022706      010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
6839      022712      010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
6840      022714      010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
6841      022716      010546      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
6842      022720      012746      020200      MOV      #20200,-(SP)      ;;SET BLANK SWITCH AND SIGN
6843      022724      016605      000020      MOV      20(SP),R5      ;;GET THE INPUT NUMBER
6844      022730      100004      BPL      1$      ;;BR IF INPUT IS POS.
6845      022732      005405      NEG      R5      ;;MAKE THE BINARY NUMBER POS.
6846      022734      112766      000055      000001      MOVB      #'-,1(SP)      ;;MAKE THE ASCII NUMBER NEG.
6847      022742      005000      1$:      CLR      R0      ;;ZERO THE CONSTANTS INDEX
6848      022744      012703      023122      MOV      #SDBLK,R3      ;;SETUP THE OUTPUT POINTER
6849      022750      112723      000040      MOVB      #'',(R3)+      ;;SET THE FIRST CHARACTER TO A BLANK
6850      022754      005002      2$:      CLR      R2      ;;CLEAR THE BCD NUMBER
6851      022756      016001      023112      MOV      $DTBL(R0),R1      ;;GET THE CONSTANT
6852      022762      160105      3$:      SUB      R1,R5      ;;FORM THIS BCD DIGIT
6853      022764      002402      BLT      4$      ;;BR IF DONE
6854      022766      005202      INC      R2      ;;INCREASE THE BCD DIGIT BY 1
6855      022770      000774      BR      3$
6856      022772      060105      4$:      ADD      R1,R5      ;;ADD BACK THE CONSTANT
6857      022774      005702      TST      R2      ;;CHECK IF BCD DIGIT=0
6858      022776      001002      BNE      5$      ;;FALL THROUGH IF 0
6859      023000      105716      TSTB      (SP)      ;;STILL DOING LEADING 0'S?
6860      023002      100407      BMI      7$      ;;BR IF YES
6861      023004      106316      5$:      ASLB      (SP)      ;;MSD?
6862      023006      103003      BCC      6$      ;;BR IF NO
6863      023010      116663      000001      177777      MOVB      1(SP),-1(R3)      ;;YES--SET THE SIGN
6864      023016      052702      000060      6$:      BIS      #'0,R2      ;;MAKE THE BCD DIGIT ASCII
6865      023022      052702      000040      7$:      BIS      #' ,R2      ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
6866      023026      110223      MOVB      R2,(R3)+      ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
6867      023030      005720      TST      (R0)+      ;;JUST INCREMENTING
6868      023032      020027      000010      CMP      R0,#10      ;;CHECK THE TABLE INDEX
6869      023036      002746      BLT      2$      ;;GO DO THE NEXT DIGIT
6870      023040      003002      BGT      8$      ;;GO TO EXIT
6871      023042      010502      MOV      R5,R2      ;;GET THE LSD
6872      023044      000764      BR      6$      ;;GO CHANGE TO ASCII
6873      023046      105726      8$:      TSTB      (SP)+      ;;WAS THE LSD THE FIRST NON-ZERO?
6874      023050      100003      BPL      9$      ;;BR IF NO
6875      023052      116663      177777      177776      MOVB      -1(SP),-2(R3)      ;;YES--SET THE SIGN FOR TYPING
6876      023060      105013      9$:      CLRB      (R3)      ;;SET THE TERMINATOR
6877      023062      012605      MOV      (SP)+,R5      ;;POP STACK INTO R5
6878      023064      012603      MOV      (SP)+,R3      ;;POP STACK INTO R3
6879      023066      012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
6880      023070      012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
6881      023072      012600      MOV      (SP)+,R0      ;;POP STACK INTO R0
6882      023074      104400      023122      TYPE      ,SDBLK      ;;NOW TYPE THE NUMBER

```


MAINDEC-11-DZRKC-C MACY11 27(732) 16-SEP-76 16:00 PAGE 129
 DZRKC.P11 CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```

6883 023100 016666 000002 000004      MOV      2(SP),4(SP)      ;;ADJUST THE STACK
6884 023106 012616      MOV      (SP)+,(SP)
6885 023110 000002      RTI                      ;;RETURN TO USER
6886 023112 023420      $DTBL: 10000.
6887 023114 001750      1000.
6888 023116 000144      100.
6889 023120 000012      10.
6890 023122 000004      $DBLK: .BLKW 4
6891
6892      .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
6893
6894      ;*****
6895      ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
6896      ;OCTAL (ASCII) NUMBER AND TYPE IT.
6897      ;$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
6898      ;CALL:
6899      ;      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
6900      ;      TYPOS      ;;CALL FOR TYPEOUT
6901      ;      .BYTE      N      ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
6902      ;      .BYTE      M      ;;M=1 OR 0
6903      ;      ;;1=TYPE LEADING ZEROS
6904      ;      ;;0=SUPPRESS LEADING ZEROS
6905      ;
6906      ;$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
6907      ;$TYPOS OR $TYPOC
6908      ;CALL:
6909      ;      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
6910      ;      TYPON      ;;CALL FOR TYPEOUT
6911      ;
6912      ;$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
6913      ;CALL:
6914      ;      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
6915      ;      TYPOC      ;;CALL FOR TYPEOUT
6916      ;
6917 023132 017646 000000      $TYPOS: MOV      2(SP),-(SP)      ;;PICKUP THE MODE
6918 023136 116637 000001 023355      MOV      1(SP), $OFILL      ;;LOAD ZERO FILL SWITCH
6919 023144 112637 023357      MOV      (SP)+, $OMODE+1      ;;NUMBER OF DIGITS TO TYPE
6920 023150 062716 000002      ADD      #2, (SP)      ;;ADJUST RETURN ADDRESS
6921 023154 000406      BR      $TYPON
6922 023156 112737 000001 023355      $TYPOC: MOV      #1, $OFILL      ;;SET THE ZERO FILL SWITCH
6923 023164 112737 000006 023357      MOV      #6, $OMODE+1      ;;SET FOR SIX(6) DIGITS
6924 023172 112737 000005 023354      $TYPON: MOV      #5, $OCNT      ;;SET THE ITERATION COUNT
6925 023200 010346      MOV      R3, -(SP)      ;;SAVE R3
6926 023202 010446      MOV      R4, -(SP)      ;;SAVE R4
6927 023204 010546      MOV      R5, -(SP)      ;;SAVE R5
6928 023206 113704 023357      MOV      $OMODE+1, R4      ;;GET THE NUMBER OF DIGITS TO TYPE
6929 023212 005404      NEG      R4
6930 023214 062704 000006      ADD      #6, R4      ;;SUBTRACT IT FOR MAX. ALLOWED
6931 023220 110437 023356      MOV      R4, $OMODE      ;;SAVE IT FOR USE
6932 023224 113704 023355      MOV      $OFILL, R4      ;;GET THE ZERO FILL SWITCH
6933 023230 016605 000012      MOV      12(SP), R5      ;;PICKUP THE INPUT NUMBER
6934 023234 005003      CLR      R3      ;;CLEAR THE OUTPUT WORD
6935 023236 006105      1$: ROL      R5      ;;ROTATE MSB INTO "C"
6936 023240 000404      BR      3$      ;;GO DO MSB
6937 023242 006105      2$: ROL      R5      ;;FORM THIS DIGIT
6938 023244 006105      ROL      R5

```

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 130
 DZRKKC.P11 BINARY TO OCTAL (ASCII) AND TYPE

6939	023246	006105			ROL	R5		
6940	023250	010503			MOV	R5,R3		
6941	023252	006103			ROL	R3	;; GET LSB OF THIS DIGIT	
6942	023254	105337	023356		DECB	\$OMODE	;; TYPE THIS DIGIT?	
6943	023260	100016			BPL	7\$	;; BR IF NO	
6944	023262	042703	177770		BIC	#177770,R3	;; GET RID OF JUNK	
6945	023266	001002			BNE	4\$	;; TEST FOR 0	
6946	023270	005704			TST	R4	;; SUPPRESS THIS 0?	
6947	023272	001403			BEQ	5\$	;; BR IF YES	
6948	023274	005204			INC	R4	;; DON'T SUPPRESS ANYMORE 0'S	
6949	023276	052703	000060		BIS	#'0,R3	;; MAKE THIS DIGIT ASCII	
6950	023302	052703	000040		BIS	#' ,R3	;; MAKE ASCII IF NOT ALREADY	
6951	023306	110337	023352		MOVB	R3,8\$	;; SAVE FOR TYPING	
6952	023312	104400	023352		TYPE	8\$	;; GO TYPE THIS DIGIT	
6953	023316	105337	023354		DECB	\$OCNT	;; COUNT BY 1	
6954	023322	003347			BGT	2\$	;; BR IF MORE TO DO	
6955	023324	002402			BLT	6\$	;; BR IF DONE	
6956	023326	005204			INC	R4	;; INSURE LAST DIGIT ISN'T A BLANK	
6957	023330	000744			BR	2\$	;; GO DO THE LAST DIGIT	
6958	023332	012605			MOV	(SP)+,R5	;; RESTORE R5	
6959	023334	012604			MOV	(SP)+,R4	;; RESTORE R4	
6960	023336	012603			MOV	(SP)+,R3	;; RESTORE R3	
6961	023340	016666	000002	000004	MOV	2(SP),4(SP)	;; SET THE STACK FOR RETURNING	
6962	023346	012616			MOV	(SP)+,(SP)		
6963	023350	000002			RTI		;; RETURN	
6964	023352	000			8\$: .BYTE	0	;; STORAGE FOR ASCII DIGIT	
6965	023353	000			.BYTE	0	;; TERMINATOR FOR TYPE ROUTINE	
6966	023354	000			\$OCNT: .BYTE	0	;; OCTAL DIGIT COUNTER	
6967	023355	000			\$OFILL: .BYTE	0	;; ZERO FILL SWITCH	
6968	023356	000000			\$OMODE: .WORD	0	;; NUMBER OF DIGITS TO TYPE	
6969								
6970					.SBTTL	TTY INPUT ROUTINE		
6971								
6972					;;*****			
6973					.ENABL	LSB		
6974								
6975					;;*****			
6976					;;SOFTWARE SWITCH REGISTER CHANGE ROUTINE.			
6977					;;ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL			
6978					;;SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL			
6979					;;WHEN OPERATING IN TTY FLAG MODE.			
6980	023360	022737	000176	001140	\$CKSWR: CMP	#SWREG,SWR	;; IS THE SOFT-SWR SELECTED?	
6981	023366	001074			BNE	15\$	;; BRANCH IF NO	
6982	023370	105777	155550		TSTB	\$TKS	;; CHAR THERE?	
6983	023374	100071			BPL	15\$	;; IF NO, DON'T WAIT AROUND	
6984	023376	117746	155544		MOVB	\$TKB,-(SP)	;; SAVE THE CHAR	
6985	023402	042716	177600		BIC	#1C177,(SP)	;; STRIP-OFF THE ASCII	
6986	023406	022726	000007		CMP	#7,(SP)+	;; IS IT A CONTROL G?	
6987	023412	001062			BNE	15\$	;; NO, RETURN TO USER	
6988	023414	123727	001134	000001	CMPB	\$AUTOB,#1	;; ARE WE RUNNING IN AUTO-MODE?	
6989	023422	001456			BEQ	15\$	;; BRANCH IF YES	
6990								
6991	023424	104400	024245		\$GTSWR: TYPE	,\$CNTLG	;; ECHO THE CONTROL-G (↑G)	
6992	023430	104400	024252		TYPE	\$MSWR	;; TYPE CURRENT CONTENTS	
6993	023434	013746	000176		MOV	\$WREG,-(SP)	;; SAVE SWREG FOR TYPEOUT	
6994	023440	104401			TYPOC		;; GO TYPE--OCTAL ASCII(ALL DIGITS)	

N10

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 131
DZRKKC.P11 TTY INPUT ROUTINE

```

6995 023442 104400 024263          TYPE    , $MNEW      ;; PROMPT FOR NEW SWR
6996 023446 005046          19$: CLR      -(SP)      ;; CLEAR COUNTER
6997 023450 005046          CLR      -(SP)      ;; THE NEW SWR
6998 023452 105777 155466      7$: TSTB    @STKS      ;; CHAR THERE?
6999 023456 100375          BPL      7$          ;; IF NOT TRY AGAIN
7000
7001 023460 117746 155462          MOVB    @STKB, -(SP)  ;; PICK UP CHAR
7002 023464 042716 177600          BIC      #1C177, (SP) ;; MAKE IT 7-BIT ASCII
7003
7004
7005
7006 023470 021627 000025      9$: CMP      (SP), #25    ;; IS IT A CONTROL-U?
7007 023474 001005          BNE      10$          ;; BRANCH IF NOT
7008 023476 104400 024240          TYPE    , $CNTLU      ;; YES, ECHO CONTROL-U (↑U)
7009 023502 062706 000006      20$: ADD      #6, SP      ;; IGNORE PREVIOUS INPUT
7010 023506 000757          BR        19$          ;; LET'S TRY IT AGAIN
7011
7012
7013 023510 021627 000015      10$: CMP      (SP), #15    ;; IS IT A <CR>?
7014 023514 001022          BNE      16$          ;; BRANCH IF NO
7015 023516 005766 000004          TST      4(SP)        ;; YES, IS IT THE FIRST CHAR?
7016 023522 001403          BEQ      11$          ;; BRANCH IF YES
7017 023524 016677 000002 155406      MOV      2(SP), @SWR  ;; SAVE NEW SWR
7018 023532 062706 000006      11$: ADD      #6, SP      ;; CLEAR UP STACK
7019 023536 104400 001213      14$: TYPE    , $CRLF      ;; ECHO <CR> AND <LF>
7020 023542 123727 001135 000001      CMPB    $INTAG, #1  ;; RE-ENABLE TTY KBD INTERRUPTS?
7021 023550 001003          BNE      15$          ;; BRANCH IF NOT
7022 023552 012777 000100 155364      MOV      #100, @STKS  ;; RE-ENABLE TTY KBD INTERRUPTS
7023 023560 000002          RTI                    ;; RETURN
7024 023562 004737 022636      16$: JSR      PC, $TYPEC  ;; ECHO CHAR
7025 023566 021627 000060          CMP      (SP), #60    ;; CHAR < 0?
7026 023572 002420          BLT      18$          ;; BRANCH IF YES
7027 023574 021627 000067          CMP      (SP), #67    ;; CHAR > 7?
7028 023600 003015          BGT      18$          ;; BRANCH IF YES
7029 023602 042726 000060          BIC      #60, (SP)+    ;; STRIP-OFF ASCII
7030 023606 005766 000002          TST      2(SP)        ;; IS THIS THE FIRST CHAR
7031 023612 001403          BEQ      17$          ;; BRANCH IF YES
7032 023614 006316          ASL      (SP)          ;; NO, SHIFT PRESENT
7033 023616 006316          ASL      (SP)          ;; CHAR OVER TO MAKE
7034 023620 006316          ASL      (SP)          ;; ROOM FOR NEW ONE.
7035 023622 005266 000002      17$: INC      2(SP)        ;; KEEP COUNT OF CHAR
7036 023626 056616 177776          BIS      -2(SP), (SP)  ;; SET IN NEW CHAR
7037 023632 000707          BR        7$          ;; GET THE NEXT ONE
7038 023634 104400 001212      18$: TYPE    , $QUES      ;; TYPE ?<CR><LF>
7039 023640 000720          BR        20$          ;; SIMULATE CONTROL-U
7040 .DSABL  LSB
7041
7042
7043 *****
7044 *THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
7045 *CALL:
7046 *   RDCHR                      ;; INPUT A SINGLE CHARACTER FROM THE TTY
7047 *   RETURN HERE                ;; CHARACTER IS ON THE STACK
7048 *                               ;; WITH PARITY BIT STRIPPED OFF
7049
7050

```

```

7051 023642 011646 $RDCHR: MOV (SP),-(SP) :: PUSH DOWN THE PC
7052 023644 016656 000004 000002 MOV 4(SP),2(SP) :: SAVE THE PS
7053 023652 105777 155266 15: TSTB 3$TKS :: WAIT FOR
7054 023656 100375 BPL 15 :: A CHARACTER
7055 023660 117766 155262 000004 MOV 3$TKB,4(SP) :: READ THE TTY
7056 023666 042766 177600 000004 BIC #10<177>,4(SP) :: GET RID OF JUNK IF ANY
7057 023674 026627 000004 000023 CMP 4(SP),#23 :: IS IT A CONTROL-S?
7058 023702 001013 BNE 3$ :: BRANCH IF NO
7059 023704 105777 155234 2$: TSTB 3$TKS :: WAIT FOR A CHARACTER
7060 023710 100375 BPL 2$ :: LOOP UNTIL ITS THERE
7061 023712 117746 155230 MOV 3$TKB,-(SP) :: GET CHARACTER
7062 023716 042716 177600 BIC #10<177>,(SP) :: MAKE IT 7-BIT ASCII
7063 023722 022627 000021 CMP (SP)+,#21 :: IS IT A CONTROL-Q?
7064 023726 001366 BNE 2$ :: IF NOT DISCARD IT
7065 023730 000750 BR 15 :: YES, RESUME
7066 023732 026627 000004 000140 3$: CMP 4(SP),#140 :: IS IT UPPER CASE?
7067 023740 002407 BLT 4$ :: BRANCH IF YES
7068 023742 026627 000004 000175 CMP 4(SP),#175 :: IS IT A SPECIAL CHAR?
7069 023750 003003 BGT 4$ :: BRANCH IF YES
7070 023752 042766 000040 000004 BIC #40,4(SP) :: MAKE IT UPPER CASE
7071 023760 000002 4$: RTI :: GO BACK TO USER
7072 *****
7073 :: THIS ROUTINE WILL INPUT A STRING FROM THE TTY
7074 :: CALL:
7075 :: * RDLIN :: INPUT A STRING FROM THE TTY
7076 :: * RETURN HERE :: ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
7077 :: * :: TERMINATOR WILL BE A BYTE OF ALL 0'S
7078
7079 023762 010346 $RDLIN: MOV R3,-(SP) :: SAVE R3
7080 023764 005046 CLR -(SP) :: CLEAR THE RUBOUT KEY
7081 023766 012703 024216 15: MOV #STTYIN,R3 :: GET ADDRESS
7082 023772 022703 024240 2$: CMP #STTYIN+22,R3 :: BUFFER FULL?
7083 023776 101456 BLOS 4$ :: BR IF YES
7084 024000 104407 RDCHR :: GO READ ONE CHARACTER FROM THE TTY
7085 024002 112613 MOV (SP)+,(R3) :: GET CHARACTER
7086 024004 122713 000177 10$: CMPB #177,(R3) :: IS IT A RUBOUT
7087 024010 001022 BNE 5$ :: BR IF NO
7088 024012 005716 TST (SP) :: IS THIS THE FIRST RUBOUT?
7089 024014 001007 BNE 6$ :: BR IF NO
7090 024016 112737 000134 024214 MOV #'\,9$ :: TYPE A BACK SLASH
7091 024024 104400 024214 TYPE 9$
7092 024030 012716 177777 MOV #-1,(SP) :: SET THE RUBOUT KEY
7093 024034 005303 6$: DEC R3 :: BACKUP BY ONE
7094 024036 020327 024216 CMP R3,#STTYIN :: STACK EMPTY?
7095 024042 103434 BLO 4$ :: BR IF YES
7096 024044 111337 024214 MOV (R3),9$ :: SETUP TO TYPEOUT THE DELETED CHAR.
7097 024050 104400 024214 TYPE 9$ :: GO TYPE
7098 024054 000746 BR 2$ :: GO READ ANOTHER CHAR.
7099 024056 005716 5$: TST (SP) :: RUBOUT KEY SET?
7100 024060 001406 BEQ 7$ :: BR IF NO
7101 024062 112737 000134 024214 MOV #'\,9$ :: TYPE A BACK SLASH
7102 024070 104400 024214 TYPE 9$
7103 024074 005016 CLR (SP) :: CLEAR THE RUBOUT KEY
7104 024076 122713 000025 7$: CMPB #25,(R3) :: IS CHARACTER A CTRL U?
7105 024102 001003 BNE 8$ :: BR IF NO
7106 024104 104400 024240 TYPE ,SCNTLU :: TYPE A CONTROL "U"

```

```

7107 024110 000726      BH      1$      ;;GO START OVER
7108 024112 122713 000022 8$: CMPB  #22,(R3)      ;;IS CHARACTER A "R"?
7109 024116 001011      BNE      3$      ;;BRANCH IF NO
7110 024120 105013      CLRB      (R3)      ;;CLEAR THE CHARACTER
7111 024122 104400 001213 TYPE .SCLF      ;;TYPE A "CR" & "LF"
7112 024126 104400 024216 TYPE $TTYIN      ;;TYPE THE INPUT STRING
7113 024132 000717      SR      2$      ;;GO PICKUP ANOTHER CHARACTER
7114 024134 104400 001212 4$: TYPE $QUES      ;;TYPE A '?'
7115 024140 000712      BR      1$      ;;CLEAR THE BUFFER AND LOOP
7116 024142 111337 024214 3$: MOVB (R3),9$      ;;ECHO THE CHARACTER
7117 024146 104400 024214 TYPE 9$
7118 024152 122723 000015 CMPB #15,(R3)+      ;;CHECK FOR RETURN
7119 024156 001305      SNE      2$      ;;LOOP IF NOT RETURN
7120 024160 105063 177777 CLRB -1(R3)      ;;CLEAR RETURN (THE 15)
7121 024164 104400 001214 TYPE .SLF      ;;TYPE A LINE FEED
7122 024170 005726      TST      (SP)+      ;;CLEAN RUBOUT KEY FROM THE STACK
7123 024172 012603      MOV      (SP)+,R3      ;;RESTORE R3
7124 024174 011646      MOV      (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
7125 024176 016666 000004 000002 MOV 4(SP),2(SP)      ;;FIRST ASCII CHARACTER ON IT
7126 024204 012766 024216 000004 MOV $TTYIN,4(SP)
7127 024212 000002      RTI      ;;RETURN
7128 024214      000      9$: .BYTE 0      ;;STORAGE FOR ASCII CHAR. TO TYPE
7129 024215      000      .BYTE 0      ;;TERMINATOR
7130 024216 000022      STTYIN: .BLKB 22      ;;RESERVE 22 BYTES FOR TTY INPUT
7131 024240 052536 005015 000 SCNTLU: .ASCIZ /U<15><12>      ;;CONTROL "U"
7132 024245      136 006507 000012 SCNTLG: .ASCIZ /G<15><12>      ;;CONTROL "G"
7133 024252 005015 053523 020122 SMSWR: .ASCIZ <15><12>/SWR = ,
7134 024260 020075      000
7135 024263      040 047040 053505 $MNEW: .ASCIZ / NEW = ,
7136 024270 036440 000040
7137
7138 ;CONTROL U, RUBOUT CAPABILITY
7139 .SBTTL TRAP DECODER
7140
7141 ;*****
7142 ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
7143 ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
7144 ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
7145 ;*GO TO THAT ROUTINE.
7146
7146 024274 010046      STRAP: MOV RO, -(SP)      ;;SAVE RO
7147 024276 016600 000002 MOV 2(SP),RO      ;;GET TRAP ADDRESS
7148 024302 005740      TST      -(RO)      ;;BACKUP BY 2
7149 024304 111000      MOVB      (RO),RO      ;;GET RIGHT BYTE OF TRAP
7150 024306 006300      ASL      RO      ;;POSITION FOR INDEXING
7151 024310 016000 024316 MOV $TRPAD(RO),RO      ;;INDEX TO TABLE
7152 024314 000200      RTS      RO      ;;GO TO ROUTINE
7153
7154 .SBTTL TRAP TABLE
7155
7156 ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
7157 ;*BY THE "TRAP" INSTRUCTION.
7158
7159 : ROUTINE
7160 : -----
7161 024316      $TRPAD: $TYPE ;;CALL=TYPE TRAP+0(104400) TTY TYPEOUT ROUTINE
7162 024316 022466

```

7163	024320	023156	\$TYPOC	::CALL=TYPOC	TRAP+1(104401)	TYPE OCTAL NUMBER (WITH LEADING ZEROS)
7164	024322	023132	\$TYPOS	::CALL=TYPOS	TRAP+2(104402)	TYPE OCTAL NUMBER (NO LEADING ZEROS)
7165	024324	023172	\$TYPON	::CALL=TYPON	TRAP+3(104403)	TYPE OCTAL NUMBER (AS PER LAST CALL)
7166	024326	022706	\$TYPDS	::CALL=TYPDS	TRAP+4(104404)	TYPE DECIMAL NUMBER (WITH SIGN)
7167						
7168	024330	023430	\$GTSWR	::CALL=GTSWR	TRAP+5(104405)	GET SOFT-SWR SETTING
7169						
7170	024332	023360	\$CKSWR	::CALL=CKSWR	TRAP+6(104406)	TEST FOR CHANGE IN SOFT-SWR
7171	024334	023642	\$RDCHR	::CALL=RDCHR	TRAP+7(104407)	TTY TYPEIN CHARACTER ROUTINE
7172	024336	023762	\$RDLIN	::CALL=RDLIN	TRAP+10(104410)	TTY TYPEIN STRING ROUTINE
7173						
7174	024340	021372	CH.CRDY	::CALL=CHKCRDY	TRAP+11(104411)	CHECK CONTROL READY
7175						
7176	024342	021426	CN.RST	::CALL=CNTR.RESET	TRAP+12(104412)	CONTROL RESET ROUTINE
7177						
7178	024344	021444	CN.RDY	::CALL=CNTR.RDY	TRAP+13(104413)	WAIT FOR CNTRL RDY TO SET
7179						
7180	024346	020470	BDAO	::CALL=BRKDAO	TRAP+14(104414)	BREAK RKDA INTO DR #,CYL,SUR,SEC BITS
7181						
7182	024350	020500	BDA4	::CALL=BRKDA4	TRAP+15(104415)	BREAK RKDA INTO DR #,CYL,SUR,SEC BITS
7183						
7184	024352	021276	DELA.Y	::CALL=DELAY	TRAP+16(104416)	TIME DELAY ROUTINE
7185						
7186	024354	021320	WATINT	::CALL=WAT.INT	TRAP+17(104417)	WAIT FOR RK11 INTERRUPT ROUTINE
7187						
7188	024356	021204	TSTSIN	::CALL=TST.SIN	TRAP+20(104420)	TEST SIN ROUTINE
7189						
7190						
7191						
7192						
7193						
7194						

.SBTTL POWER DOWN AND UP ROUTINES

:*****

:POWER DOWN ROUTINE

```

$PWRDN: MOV    $SILLUP, $PWRVEC    ;; SET FOR FAST UP
        MOV    $340, $PWRVEC+2    ;; PRI0:7
        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
        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

```

E11

MAINDEC-11-DZRRK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 135
 DZRRKC.P11 POWER DOWN AND UP ROUTINES

```

7219 024466 012603      MOV      (SP)+,R3      ;; POP STACK INTO R3
7220 024470 012602      MOV      (SP)+,R2      ;; POP STACK INTO R2
7221 024472 012601      MOV      (SP)+,R1      ;; POP STACK INTO R1
7222 024474 012600      MOV      (SP)+,R0      ;; POP STACK INTO R0
7223 024476 012737 024360 000024      MOV      #SPWRDN,2#PWRVEC ;; SET UP THE POWER DOWN VECTOR
7224 024504 012737 000340 000026      MOV      #340,2#PWRVEC+2 ;; PRIO:7
7225 024512 104400      TYPE      $PWRMG: .WORD $POWER      ;; REPORT THE POWER FAILURE
7226 024514 024532      MOV      (PC)+,(SP)      ;; POWER FAIL MESSAGE POINTER
7227 024516 012716      $PWRAD: .WORD PFSTRT      ;; RESTART AT PFSTRT
7228 024520 004372      RTI      PFSTRT      ;; RESTART ADDRESS
7229 024522 000002      HALT      .-2      ;; THE POWER UP SEQUENCE WAS STARTED
7230 024524 000000      BR      0      ;; BEFORE THE POWER DOWN WAS COMPLETE
7231 024526 000776      $SAVR6: 0      ;; PUT THE SP HERE
7232 024530 000000      $POWER: .ASCIZ <15><12>"POWER"
7233 024532 005015 047520 042527      .EVEN
7234 024540 000122
7235
7236
7237 024542 004737 021116      FCHECK: JSR      PC,DRESET      ; RESETB DRIVE
7238 024546 104026      ERROR      26
7239 024550 104412      CNT.RESET
7240 024552 013737 002544 024664      MOV      DRIVAD,DRHOLD      ; SAVE DRIVE ADDR
7241 024560 032737 020000 002544      BIT      #20000,DRIVAD      ; SEE IF ODD
7242 024566 001404      BEQ      1$
7243 024570 042737 020000 002544      BIC      #20000,DRIVAD      ; MAKE EVEN
7244 024576 000403      BR      2$
7245 024600 052737 020000 002544 1$: BIS      #20000,DRIVAD      ; MAKE ODD
7246 024606 013777 002544 155720 2$: MOV      DRIVAD,2RKDA      ; DRIVE ADDR
7247 024614 012777 000011 155704      MOV      #11,2RKCS      ; DRIVE SEEK
7248 024622 104413      CNT.RDY
7249 024624 013777 024664 155702      MOV      DRHOLD,2RKDA      ; OTHER DRIVE
7250 024632 104413      CNT.RDY
7251 024634 032777 000100 155660      BIT      #100,2RKDS      ; HEADS IN MOTION?
7252 024642 001001      BNE      3$      ; NO SO RK-05J
7253 024644 005725      TST      (R5)+      ; YES RK-05F
7254 024646 013737 024664 002544 3$: MOV      DRHOLD,DRIVAD      ; RESTORE ADDR
7255 024654 004737 021116      JSR      PC,DRESET      ; WAIT FOR RESET
7256 024660 104026      ERROR      26
7257 024662 000205      RTS      R5
7258 024664 000000      DRHOLD: 0
7259 024666 005037 002544      SIZEF: CLR      DRIVAD      ; START AT DRO
7260 024672 012700 002610      MOV      #DRIVO,R0      ; TABLE OF AVAIL DRIVES
7261 024676 005710      4$: TST      (R0)      ; THIS DRIVE HERE?
7262 024700 001413      BEQ      2$      ; NO
7263 024702 005760 000002      TST      2(R0)      ; COMPLEMENT HERE?
7264 024706 001410      BEQ      2$      ; NO
7265 024710 004537 024542      JSR      R5,FCHECK      ; SEE IF F MODEL
7266 024714 000405      BR      2$      ; J MODEL
7267 024716 052710 100000      BIS      #100000,(R0)      ; SET SIGN FOR F
7268 024722 052760 100000 000002      BIS      #100000,2(R0)      ; BOTH DRIVES
7269 024730 005720      2$: TST      (R0)+      ; NEXT PAIR OF DRIVES
7270 024732 005720      TST      (R0)+      ; NEXT ACTUL ADD
7271 024734 062737 040000 002544      ADD      #40000,DRIVAD      ; CHECKED ALL?
7272 024742 022700 002627      CMP      #DRIV7+1,R0      ; NOT YET
7273 024746 003353      BGT      4$
7274 024750 000207      RTS      PC

```

F11

MAINDEC-11-DZKCC-MACY11 27(732) 16-SEP-76 16:00 PAGE 136
 DZKCC.P11 POWER DOWN AND UP ROUTINES

7275							
7276						:ERROR MESSAGES	
7277							
7278						.SBTTL ERROR MESSAGES	
7279							
7280	024752	045522	041527	042440	EM11:	.ASCIZ /RKWC EROR/	
7281	024760	047522	000122				
7282							
7283							
7284	024764	044523	020116	051511	EM12:	.ASCIZ /SIN IS SET/	
7285	024772	051440	052105	000			
7286							
7287	024777	122	041113	020101	EM13:	.ASCIZ /RKBA EROR/	
7288	025004	051105	051117	000			
7289							
7290	025011	122	042113	020101	EM16:	.ASCIZ /RKDA WRONG AFTER 'SSE'/	
7291	025016	051127	047117	020107			
7292	025024	043101	042524	020122			
7293	025032	051447	042523	000047			
7294							
7295	025040	045522	051504	042440	EM21:	.ASCIZ /RKDS EROR/	
7296	025046	047522	000122				
7297							
7298	025052	050104	020114	042523	EM30:	.ASCIZ /DPL SET/	
7299	025060	000124					
7300							
7301	025062	051104	020125	042523	EM31:	.ASCIZ /DRU SET/	
7302	025070	000124					
7303							
7304	025072	045522	032460	041040	EM32:	.ASCIZ /RKOS BIT NOT SET/	
7305	025100	052111	047040	052117			
7306	025106	051440	052105	000			
7307							
7308	025113	104	054522	041040	EM33:	.ASCIZ /DRY BIT NOT SET/	
7309	025120	052111	047040	052117			
7310	025126	051440	052105	000			
7311							
7312	025133	123	045517	042040	EM34:	.ASCIZ /SOK DIDN'T SET/	
7313	025140	042111	023516	020124			
7314	025146	042523	000124				
7315							
7316	025152	042523	026503	047103	EM35:	.ASCIZ /SEC-CNTR DIDN'T COUNT TO 0/	
7317	025160	051124	042040	042111			
7318	025166	023516	020124	047503			
7319	025174	047125	020124	047524			
7320	025202	030040	000				
7321							
7322	025205	123	041505	041455	EM36:	.ASCIZ /SEC-CNTR DIDN'T INCRMNT/	
7323	025212	052116	020122	044504			
7324	025220	047104	052047	044440			
7325	025226	041516	046522	052116			
7326	025234	000					
7327							
7328	025235	123	041505	041455	EM37:	.ASCIZ /SEC-COUNTR INCRMENTED WRONG/	
7329	025242	052517	052116	020122			
7330	025250	047111	051103	042515			

G11

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 137
 DZRKKC.P11 ERROR MESSAGES

7331	025256	052116	042105	053440	
7332	025264	047522	043516	000	
7333					
7334	025271	104	042111	023516	EM40: .ASCIZ /DIDN'T GET SC=SA FOR THIS SECTR/
7335	025276	020124	042507	020124	
7336	025304	041523	051475	020101	
7337	025312	047506	020122	044124	
7338	025320	051511	051440	041505	
7339	025326	051124	000		
7340					
7341	025331	105	047522	026522	EM41: .ASCIZ "EROR-R/W/S RDY SHOULD BE SET"
7342	025336	027522	027527	020123	
7343	025344	042122	020131	044123	
7344	025352	052517	042114	041040	
7345	025360	020105	042523	000124	
7346					
7347	025366	047125	054105	042520	EM43: .ASCIZ /UNEXPECTED RK11 INTERRUPT/
7348	025374	052103	042105	051040	
7349	025402	030513	020061	047111	
7350	025410	042524	051122	050125	
7351	025416	000124			
7352					
7353	025420	047103	051124	020114	EM44: .ASCIZ /CNTRL RDY DIDN'T SET AFTER SEEK OR DR RESET/
7354	025426	042122	020131	044504	
7355	025434	047104	052047	051440	
7356	025442	052105	040440	052106	
7357	025450	051105	051440	042505	
7358	025456	020113	051117	042040	
7359	025464	020122	042522	042523	
7360	025472	000124			
7361					
7362	025474	051105	020122	051117	EM45: .ASCIZ /ERR OR HE BIT SET ON SEEK OR DR RESET/
7363	025502	044040	020105	044502	
7364	025510	020124	042523	020124	
7365	025516	047117	051440	042505	
7366	025524	020113	051117	042040	
7367	025532	020122	042522	042523	
7368	025540	000124			
7369					
7370	025542	045522	051105	041040	EM46: .ASCIZ /RKER BIT, ON SEEK OR DR RESET/
7371	025550	052111	020054	047117	
7372	025556	051440	042505	020113	
7373	025564	051117	042040	020122	
7374	025572	042522	042523	000124	
7375					
7376	025600	045522	051503	041440	EM47: .ASCIZ /RKCS CHNGD AFTR FUNCTION WAS DONE/
7377	025606	047110	042107	040440	
7378	025614	052106	020122	052506	
7379	025622	041516	044524	047117	
7380	025630	053440	051501	042040	
7381	025636	047117	000105		
7382					
7383	025642	027522	027527	020123	EM50: .ASCIZ "R/W/S RDY DIDN'T CLEAR"
7384	025650	042122	020131	044504	
7385	025656	047104	052047	041440	
7386	025664	042514	051101	000	

H11

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 138
 DZRKKC.P11 ERROR MESSAGES

7387					
7388	025671	122	053457	051457	EMS1: .ASCIZ "R/W/S RDY DIDN'T SET AFTR SEEK OR DR RESET"
7389	025676	051040	054504	042040	
7390	025704	042111	023516	020124	
7391	025712	042523	020124	043101	
7392	025720	051124	051440	042505	
7393	025726	020113	051117	042040	
7394	025734	020122	042522	042523	
7395	025742	000124			
7396					
7397	025744	045522	040504	041440	EMS2: .ASCIZ /RKDA CHNGD AFTR SEEK/
7398	025752	047110	042107	040440	
7399	025760	052106	020122	042523	
7400	025766	045505	000		
7401					
7402	025771	103	052116	046122	EMS3: .ASCIZ /CNTRL RDY DIDN'T CLR AS GO WAS SET/
7403	025776	051040	054504	042040	
7404	026004	042111	023516	020124	
7405	026012	046103	020122	051501	
7406	026020	043440	020117	040527	
7407	026026	020123	042523	000124	
7408					
7409	026034	047103	051124	020114	EMS4: .ASCIZ "CNTRL RDY DIDN'T SET ON WRT/FMT STARTING FROM <DSK-ADRES>"
7410	026042	042122	020131	044504	
7411	026050	047104	052047	051440	
7412	026056	052105	047440	020116	
7413	026064	051127	027524	046506	
7414	026072	020124	052123	051101	
7415	026100	044524	043516	043040	
7416	026106	047522	020115	042074	
7417	026114	045523	040455	051104	
7418	026122	051505	000076		
7419					
7420	026126	042510	047440	020122	EMS5: .ASCIZ "HE OR ERR ON WRT/FMT STARTING FROM <DSK-ADRES>"
7421	026134	051105	020122	047117	
7422	026142	053440	052122	043057	
7423	026150	052115	051440	040524	
7424	026156	052122	047111	020107	
7425	026164	051106	046517	036040	
7426	026172	051504	026513	042101	
7427	026200	042522	037123	000	
7428					
7429	026205	122	042113	020101	EMS6: .ASCIZ /RKDA INCRMNTD WRONG ON WRT-FMT/
7430	026212	047111	051103	047115	
7431	026220	042124	053440	047522	
7432	026226	043516	047440	020116	
7433	026234	051127	026524	046506	
7434	026242	000124			
7435					
7436	026244	045522	041527	042040	EMS7: .ASCIZ /RKWC DIDN'T OVRFLO ON WRT FMT/
7437	026252	042111	023516	020124	
7438	026260	053117	043122	047514	
7439	026266	047440	020116	051127	
7440	026274	020124	046506	000124	
7441					
7442	026302	045522	040502	044440	EMS8: .ASCIZ /RKBA INCRMNTD WRONG ON WRT FMT/

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 139
DZRKKC.P11 ERROR MESSAGES

7443	026310	041516	046522	052116	
7444	026316	020104	051127	047117	
7445	026324	020107	047117	053440	
7446	026332	052122	043040	052115	
7447	026340	000			
7448					
7449	026341	122	042513	020122	EM61: .ASCIZ /RKER SET,ON WRT OR RD OR FMT/
7450	026346	042523	026124	047117	
7451	026354	053440	052122	047440	
7452	026362	020122	042122	047440	
7453	026370	020122	046506	000124	
7454					
7455	026376	045522	041104	042440	EM62: .ASCIZ /RKDB EROR/
7456	026404	047522	000122		
7457					

J11

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 140
 DZRKKC.P11 ERROR MESSAGES

7458	026410	045522	040504	044440	EM63:	.ASCIZ /RKDA INCRMNTD WRONG ON RD OR RD FMT/
7459	026416	041516	046522	052116		
7460	026424	020104	051127	047117		
7461	026432	020107	047117	051040		
7462	026440	020104	051117	051040		
7463	026446	020104	046506	000124		
7464						
7465	026454	045522	041527	042040	EM64:	.ASCIZ /RKWC DIDN'T OVRFLO ON RD OR RD FMT/
7466	026462	042111	023516	020124		
7467	026470	053117	043122	047514		
7468	026476	047440	020116	042122		
7469	026504	047440	020122	042122		
7470	026512	043040	052115	000		
7471						
7472	026517	122	041113	020101	EM65:	.ASCIZ /RKBA INCRMNTD WRONG ON RD OR RD FMT/
7473	026524	047111	051103	047115		
7474	026532	042124	053440	047522		
7475	026540	043516	047440	020116		
7476	026546	042122	047440	020122		
7477	026554	042122	043040	052115		
7478	026562	000				
7479						
7480	026563	111	041516	051117	EM66:	.ASCIZ /INCORRECT HEADER FROM 'SECTOR'/
7481	026570	042522	052103	044040		
7482	026576	040505	042504	020122		
7483	026604	051106	046517	023440		
7484	026612	042523	052103	051117		
7485	026620	000047				
7486						
7487	026622	040504	040524	042440	EM67:	.ASCIZ /DATA ERROR/
7488	026630	051122	051117	000		
7489						
7490	026635	103	052116	046122	EM70:	.ASCIZ "CNTRL RDY DIDN'T SET ON RD/FMT STARTING FROM <DSK-ADRES>"
7491	026642	051040	054504	042040		
7492	026650	042111	023516	020124		
7493	026656	042523	020124	047117		
7494	026664	051040	027504	046506		
7495	026672	020124	052123	051101		
7496	026700	044524	043516	043040		
7497	026706	047522	020115	042074		
7498	026714	045523	040455	051104		
7499	026722	051505	000076			
7500						
7501	026726	042510	047440	020122	EM71:	.ASCIZ "HE OR ERR ON RD/FMT STARTING FROM <DSK-ADRES>"
7502	026734	051105	020122	047117		
7503	026742	051040	027504	046506		
7504	026750	020124	052123	051101		
7505	026756	044524	043516	043040		
7506	026764	047522	020115	042074		
7507	026772	045523	040455	051104		
7508	027000	051505	000076			
7509						
7510	027004	051127	047117	020107	EM72:	.ASCIZ /WRONG DRIVE ID IN RKDS AFTER SEEK/
7511	027012	051104	053111	020105		
7512	027020	042111	044440	020116		
7513	027026	045522	051504	040440		

K11

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 141
 DZRKKC.P11 ERROR MESSAGES

7514	027034	052106	051105	051440	
7515	027042	042505	000113		
7516					
7517	027046	051110	053504	042522	EM73: .ASCIZ /HRDWRE POLL-DRV ID BITS(13-15) SHLDBE CLR/
7518	027054	050040	046117	026514	
7519	027062	051104	020126	042111	
7520	027070	041040	052111	024123	
7521	027076	031461	030455	024465	
7522	027104	051440	046110	041104	
7523	027112	020105	046103	000122	
7524					
7525	027120	051110	053504	042522	EM74: .ASCIZ /HRDWRE POLL-INTRUPTING DRV * NOT PRSNT/
7526	027126	050040	046117	026514	
7527	027134	047111	051124	050125	
7528	027142	044524	043516	042040	
7529	027150	044522	020126	020043	
7530	027156	047516	020124	051120	
7531	027164	047123	000124		
7532					
7533	027170	051104	053111	021440	EM75: .ASCIZ /DRV * DIDN'T INTRUPT AFTER HRDWRE POLL/
7534	027176	042040	042111	023516	
7535	027204	020124	047111	051124	
7536	027212	050125	020124	043101	
7537	027220	042524	020122	051110	
7538	027226	053504	042522	050040	
7539	027234	046117	000114		
7540					
7541	027240	041523	020120	044504	EM76: .ASCIZ /SCP DIDN'T SET AFTER SEEK WAS DONE/
7542	027246	047104	052047	051440	
7543	027254	052105	040440	052106	
7544	027262	051105	051440	047505	
7545	027270	020113	040527	020123	
7546	027276	047504	042516	000	
7547					
7548	027303	122	042113	020101	EM77: .ASCIZ /RKDA CHANGD AFTER DRV RESET/
7549	027310	044103	047101	042107	
7550	027316	040440	052106	051105	
7551	027324	042040	044522	020126	
7552	027332	042522	042523	000124	
7553					
7554	027340	040504	040524	042440	EM100: .ASCIZ /DATA EROR AT WORD#/
7555	027346	047522	020122	052101	
7556	027354	053440	051117	021504	
7557	027362	000			
7558					
7559	027363	103	052116	046122	EM101: .ASCIZ /CNTRL RDY DIDN'T SET AFTER RD CHK/
7560	027370	051040	054504	042040	
7561	027376	042111	023516	020124	
7562	027404	042523	020124	043101	
7563	027412	042524	020122	042122	
7564	027420	041440	045510	000	
7565					
7566	027425	105	051122	047440	EM102: .ASCIZ /ERR OR HE ON RD CHK/
7567	027432	020122	042510	047440	
7568	027440	020116	042122	041440	
7569	027446	045510	000		

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 142
 DZRKKC.P11 ERROR MESSAGES

7570					
7571	027451	103	042523	047440	EM103: .ASCIZ /CSE ON RD CHK/
7572	027456	020116	042122	041440	
7573	027464	045510	000		
7574					
7575	027467	122	053513	020103	EM104: .ASCIZ /RKWC DIDN'T OVERFLO ON RD CHK OR WRT CHK/
7576	027474	044504	047104	052047	
7577	027502	047440	042526	043122	
7578	027510	047514	047440	020116	
7579	027516	042122	041440	045510	
7580	027524	047440	020122	051127	
7581	027532	020124	044103	000113	
7582					
7583	027540	045522	040504	044440	EM105: .ASCIZ /RKDA INCRMNTD WRONG ON RD CHK/
7584	027546	041516	046522	052116	
7585	027554	020104	051127	047117	
7586	027562	020107	047117	051040	
7587	027570	020104	044103	000113	
7588					
7589	027576	045522	040502	041440	EM106: .ASCIZ /RKBA CHANGD AFTER RD CHK/
7590	027604	040510	043516	020104	
7591	027612	043101	042524	020122	
7592	027620	042122	041440	045510	
7593	027626	000			
7594					
7595	027627	115	046505	051117	EM107: .ASCIZ /MEMORY WORD CHANGED AFTER RD CHK/
7596	027634	020131	047527	042122	
7597	027642	041440	040510	043516	
7598	027650	042105	040440	052106	
7599	027656	051105	051040	020104	
7600	027664	044103	000113		
7601					
7602	027670	047103	051124	020114	EM110: .ASCIZ /CNTRL RDY DIDN'T SET AFTER WRT CHK/
7603	027676	042122	020131	044504	
7604	027704	047104	052047	051440	
7605	027712	052105	040440	052106	
7606	027720	051105	053440	052122	
7607	027726	041440	045510	000	
7608					
7609	027733	110	020105	051117	EM111: .ASCIZ /HE OR ERR ON WRT CHK/
7610	027740	042440	051122	047440	
7611	027746	020116	051127	020124	
7612	027754	044103	000113		
7613					
7614	027760	051127	052111	020105	EM112: .ASCIZ /WRITE CHECK EROR/
7615	027766	044103	041505	020113	
7616	027774	051105	051117	000	
7617					
7618	030001	122	042113	020101	EM113: .ASCIZ /RKDA INCRMNTD WRONG ON WRT CHK/
7619	030006	047111	051103	047115	
7620	030014	042124	053440	047522	
7621	030022	043516	047440	020116	
7622	030030	051127	020124	044103	
7623	030036	000113			
7624					
7625	030040	045522	040502	044440	EM114: .ASCIZ /RKBA INCRMNTD WRONG ON WRT CHK/

M11

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 143
 DZRKKC.P11 ERROR MESSAGES

7626	030046	041516	046522	052116	
7627	030054	020104	051127	047117	
7628	030062	020107	047117	053440	
7629	030070	052122	041440	045510	
7630	030076	000			
7631					
7632	030077	122	041113	020101	EM115: .ASCIZ /RKBA INCRMNTD, WITH IBA SET/
7633	030104	047111	051103	047115	
7634	030112	042124	020054	044527	
7635	030120	044124	044440	040502	
7636	030126	051440	052105	000	
7637					
7638	030133	127	047522	043516	EM116: .ASCIZ /WRONG MEMORY LOCATION CHANGED WITH IBA SET/
7639	030140	046440	046505	051117	
7640	030146	020131	047514	040503	
7641	030154	044524	047117	041440	
7642	030162	040510	043516	042105	
7643	030170	053440	052111	020110	
7644	030176	041111	020101	042523	
7645	030204	000124			
7646					
7647	030206	045522	030461	042040	EM117: .ASCIZ /RK11 DIDN'T INTRUPT WHEN IDE WAS SET/
7648	030214	042111	023516	020124	
7649	030222	047111	051124	050125	
7650	030230	020124	044127	047105	
7651	030236	044440	042504	053440	
7652	030244	051501	051440	052105	
7653	030252	000			
7654					
7655	030253	122	030513	020061	EM120: .ASCIZ /RK11 DIDN'T INTRUPT AFTER SK WAS INITIATED/
7656	030260	044504	047104	052047	
7657	030266	044440	052116	052522	
7658	030274	052120	040440	052106	
7659	030302	051105	051440	020113	
7660	030310	040527	020123	047111	
7661	030316	052111	040511	042524	
7662	030324	000104			
7663					
7664	030326	041523	020120	042523	EM121: .ASCIZ /SCP SET BEFORE SEEK COMPLETED/
7665	030334	020124	042502	047506	
7666	030342	042522	051440	042505	
7667	030350	020113	047503	050115	
7668	030356	042514	042524	000104	
7669					
7670	030364	045522	030461	042040	EM122: .ASCIZ /RK11 DIDN'T INTRUPT AFTER SK COMPLETED/
7671	030372	042111	023516	020124	
7672	030400	047111	051124	050125	
7673	030406	020124	043101	042524	
7674	030414	020122	045523	041440	
7675	030422	046517	046120	052105	
7676	030430	042105	000		
7677					
7678	030433	103	052116	046122	EM123: .ASCIZ /CNTRL RESET DIDN'T CLEAR 'SCP'/
7679	030440	051040	051505	052105	
7680	030446	042040	042111	023516	
7681	030454	020124	046103	040505	

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 144
 DZRKKC.P11 ERROR MESSAGES

7682	030462	020122	051447	050103	
7683	030470	000047			
7684					
7685	030472	045522	030461	042040	EM124: .ASCIZ /RK11 DIDN'T INTRUPT AFTER RD DONE/
7686	030500	042111	023516	020124	
7687	030506	047111	051124	050125	
7688	030514	020124	043101	042524	
7689	030522	020122	042122	042040	
7690	030530	047117	000105		
7691					
7692	030534	047103	051124	020114	EM125: .ASCIZ /CNTRL RESET DIDN'T CLR REGISTR/
7693	030542	042522	042523	020124	
7694	030550	044504	047104	052047	
7695	030556	041440	051114	051040	
7696	030564	043505	051511	051124	
7697	030572	000			
7698					
7699	030573	122	030513	020061	EM126: .ASCIZ /RK11 DIDN'T INTRUPT AT CPU LEVEL/
7700	030600	044504	047104	052047	
7701	030606	044440	052116	052522	
7702	030614	052120	040440	020124	
7703	030622	050103	020125	042514	
7704	030630	042526	000114		
7705					
7706	030634	045522	030461	044440	EM127: .ASCIZ /RK11 INTRUPTED AT WRONG CPU LEVEL/
7707	030642	052116	052522	052120	
7708	030650	042105	040440	020124	
7709	030656	051127	047117	020107	
7710	030664	050103	020125	042514	
7711	030672	042526	000114		
7712					
7713	030676	042447	051122	041040	EM130: .ASCIZ /'ERR BIT' DIDN'T SET IN RKER/
7714	030704	052111	020047	044504	
7715	030712	047104	052047	051440	
7716	030720	052105	044440	020116	
7717	030726	045522	051105	000	
7718					
7719	030733	110	020105	051117	EM131: .ASCIZ /HE OR ERR DIDN'T SET/
7720	030740	042440	051122	042040	
7721	030746	042111	023516	020124	
7722	030754	042523	000124		
7723					
7724	030760	045522	051105	042440	EM132: .ASCIZ /RKER ERROR/
7725	030766	047522	000122		
7726					
7727	030772	054116	020103	044502	EM133: .ASCIZ /NXC BIT DIDN'T SET/
7728	031000	020124	044504	047104	
7729	031006	052047	051440	052105	
7730	031014	000			
7731					
7732	031015	122	030513	020061	EM134: .ASCIZ /RK11 DIDN'T INTRUPT ON SOFT ERROR/
7733	031022	044504	047104	052047	
7734	031030	044440	052116	052522	
7735	031036	052120	047440	020116	
7736	031044	047523	052106	042440	
7737	031052	047522	000122		

7739	031056	042515	020130	044502	EM135:	.ASCIZ	/MEX BITS INCRMNTD WRONG-RKCS/
7740	031064	051524	044440	041516			
7741	031072	046522	052116	020104			
7742	031100	051127	047117	026507			
7743	031106	045522	051503	000			
7744							
7745	031113	127	051520	047040	EM137:	.ASCIZ	/WPS NOT CLEAR/
7746	031120	052117	041440	042514			
7747	031126	051101	000				
7748							
7749	031131	104	052101	020101	EM140:	.ASCIZ	/DATA EROR ON TRANSFER FROM DISK TO TTY/
7750	031136	051105	051117	047440			
7751	031144	020116	051124	047101			
7752	031152	043123	051105	043040			
7753	031160	047522	020115	044504			
7754	031166	045523	052040	020117			
7755	031174	052124	000131				
7756							
7757	031200	042047	044522	020126	EM141:	.ASCIZ	/DRIV # PRESENT, BUT NOT INDICATED/
7758	031206	023443	050040	042522			
7759	031214	042523	052116	020054			
7760	031222	052502	020124	047516			
7761	031230	020124	047111	044504			
7762	031236	040503	042524	000104			
7763	031244	047040	020117	052502	EM142:	.ASCIZ	/NO BUSY ON OTHER HALF OF RK-05F/
7764	031252	054523	047440	020116			
7765	031260	052117	042510	020122			
7766	031266	040510	043114	047440			
7767	031274	020106	045522	030055			
7768	031302	043065	000				
7769							
7770							
7771							
7772							
7773							
7774		031306				.EVEN	
7775						.SBTTL	ERROR DATA POINTERS
7776							
7777							
7778	031306	001116	001162	000000	DT1:	.WORD	\$ERRPC,\$REG0,0
7779							
7780	031314	001116	001162	001164	DT2:	.WORD	\$ERRPC,\$REG0,\$REG1,0
7781	031322	000000					
7782							
7783	031324	001116	001162	001164	DT20:	.WORD	\$ERRPC,\$REG0,\$REG1,\$REG2,\$REG3,0
7784	031332	001166	001170	000000			
7785							
7786	031340	001116	000000		DT21:	.WORD	\$ERRPC,0
7787							
7788	031344	001116	001162	001164	DT26:	.WORD	\$ERRPC,\$REG0,\$REG1,\$REG2,0
7789	031352	001166	000000				
7790							
7791	031356	001116	001162	001164	DT54:	.WORD	\$ERRPC,\$REG0,\$REG1,\$REG2,\$REG3,\$REG4,\$REG5,\$REG6,\$REG7,0
7792	031364	001166	001170	001172			
7793	031372	001174	001176	001200			

Line	Address	Value	Label	Header	Field	Value	Header	Field	Value	Header	Field	Value	Header	Field	Value
7794	031400	000000													
7795															
7796															
7797															
7798															
7799															
7800															
7801	031402	020040	041520	020040	DH2:	.ASCIZ	/	PC	REGADD	RECVD/					
7802	031410	051040	043505	042101											
7803	031416	020104	020040	051040											
7804	031424	041505	042126	000											
7805															
7806	031431	040	050040	020103	DH4:	.ASCIZ	/	PC	EXPCT	RECVD/					
7807	031436	020040	042440	050130											
7808	031444	052103	020040	051040											
7809	031452	041505	042126	000											
7810															
7811	031457	040	050040	020103	DH5:	.ASCIZ	/	PC	RECVD/						
7812	031464	020040	051040	041505											
7813	031472	042126	000												
7814															
7815	031475	040	050040	020103	DH14:	.ASCIZ	/	PC	RKCS	RKER	RKWC/				
7816	031502	020040	051040	041513											
7817	031510	020123	020040	051040											
7818	031516	042513	020122	020040											
7819	031524	051040	053513	000103											
7820															
7821	031532	020040	041520	000	DH21:	.ASCIZ	/	PC/							
7822															
7823	031537	040	050040	020103	DH30:	.ASCIZ	/	PC	RKCS	RKER	RKDS/				
7824	031544	020040	020040	045522											
7825	031552	051503	020040	020040											
7826	031560	045522	051105	020040											
7827	031566	020040	045522	051504											
7828	031574	000													
7829															
7830	031575	040	050040	020103	DH34:	.ASCIZ	/	PC	RKDS/						
7831	031602	020040	020040	045522											
7832	031610	051504	000												
7833															
7834	031613	040	050040	020103	DH35:	.ASCIZ	/	PC	SEC-CNTR/						
7835	031620	020040	042523	026503											
7836	031626	047103	051124	000											
7837															
7838	031633	040	050040	020103	DH36:	.ASCIZ	/	PC	PRSNT	NXT-CNT/					
7839	031640	020040	020040	051120											
7840	031646	047123	020124	047040											
7841	031654	052130	041455	052116											
7842	031662	000													
7843															
7844	031663	040	050040	020103	DH40:	.ASCIZ	/	PC	SECTOR	RKDS/					
7845	031670	020040	051440	041505											
7846	031676	047524	020122	020040											
7847	031704	045522	051504	000											
7848															
7849	031711	040	050040	020103	DH44:	.ASCIZ	/	PC	RKCS	RKER	RKDS	RKDA/			

7950	031716	020040	051040	041513
7851	031724	020123	020040	051040
7952	031732	042513	020122	020040
7853	031740	051040	042113	020123
7854	031746	020040	051040	042113
7855	031754	000101		

7857	031756	020040	041520	020040	DH54:	.ASCIIZ	/	PC	RKCS	RKER	RKDS	RKDA	DRV#....CYL.<DSK-ADRS>.SUP..SEC/
7958	031764	020040	045522	051503									
7959	031772	020040	020040	045522									
7960	032000	051105	020040	020040									
7861	032006	045522	051504	020040									
7862	032014	020040	045522	040504									
7863	032022	020040	042040	053122									
7864	032030	027043	027056	041456									
7865	032036	046131	036056	051504									
7866	032044	026513	042101	051522									
7867	032052	027076	052523	027122									
7868	032060	051456	041505	000									

[illegible]

7882	032172	020040	041520	020040	DH64:	.ASCII	/	PC	RKWC	RKDA/
7883	032200	020040	045522	041527						
7895	032206	020040	051040	042113						
7896	032214	000101								

7887	032216	020040	041520	020040	DH66:	.ASCIIZ	/	PC	SECTR	EXPCT	RECVD/
7889	032224	020040	042523	052103							
7890	032232	020122	020040	054105							
7891	032240	041520	020124	020040							
7892	032246	042522	053103	000104							

7894	032254	020040	041520	020040	DH67:	.ASCIZ / PC	EXPCT	RECVD	DSK-ADRS/
7895	032262	020040	054105	041520					
7896	032270	020124	020040	042522					
7897	032276	053103	020104	042040,					
7898	032304	045523	040455	051104					
7899	032312	000123							

```

7901 032314 020040 041520 020040 CH74: .ASCIZ / PC      DRV #/
7902 032322 020040 020040 051104
7903 032330 053111 021440      000

```

7905 032335 040 050040 020103 DH100: .ASCIZ / PC WORD # EXPCT RECVD/

7905	032342	053440	051117	020104															
7907	032350	020043	020040	054105															
7908	032356	041520	020124	020040															
7909	032364	042522	053103	000104															
7910																			
7911	032372	020040	041520	020040	DH103:	.ASCIZ	/	PC	RKER/										
7912	032400	051040	042513	000122															
7913																			
7914	032406	020040	041520	020040	DH104:	.ASCIZ	/	PC	RECVD	RKCS/									
7915	032414	051040	041505	042126															
7916	032422	020040	051040	041513															
7917	032430	000123																	
7918																			
7919	032432	020040	041520	020040	DH107:	.ASCIZ	/	PC	LOC	EXPCT	RECVD/								
7920	032440	020040	046040	041517															
7921	032446	020040	020040	054105															
7922	032454	041520	020124	020040															
7923	032462	042522	053103	000104															
7924																			
7925	032470	020040	041520	020040	DH117:	.ASCIZ	/	PC	RKCS/										
7926	032476	051040	041513	000123															
7927																			
7928	032504	020040	041520	020040	DH126:	.ASCIZ	/	PC	LEVEL	RKCS/									
7929	032512	020040	042514	042526															
7930	032520	020114	020040	051040															
7931	032526	041513	000123																
7932																			
7933	032532	020040	041520	020040	DH130:	.ASCIZ	/	PC	RKCS	RKER	ERR BIT/								
7934	032540	020040	051040	041513															
7935	032546	020123	020040	051040															
7936	032554	042513	020122	042440															
7937	032562	051122	041040	052111															
7938	032570	000																	
7939																			
7940	032571	040	050040	020103	DH131:	.ASCIZ	/	PC	RKCS	RKER/									
7941	032576	020040	020040	045522															

F12

MAINDEC-11-DZKRC-C MACY11 27(732) 16-SEP-76 16:00 PAGE 149
DZKRC.P11 ERROR HEADERS

```
7962          :DATA BUFFER
7963
7964 032724 000400      OUTBUF: .BLKW 256.      :THIS 256 WORD BUFFER IS FOR
7965                                     :DATA TRANSFERS FROM AND
7966                                     :TO THE DISK.
7967
7968          000001
7969          .END
```

BADINT	004270	2096	2124#	4617	4662	4681	4692	4772	4780	4869	4878	5522			
BADTMO	004224	2031	2094	2105#	5631	5635									
BOAR	020506	6155	6160#												
BDA0	020470	6153#	7180												
BCA4	020500	6157#	7182												
BIT0 =	000001	1032#	1990	1994	1998	2257	5877								
BIT00 =	000001	1022#	1032												
BIT01 =	000002	1021#	1031												
BIT02 =	000004	1020#	1030												
BIT03 =	000010	1019#	1029												
BIT04 =	000020	1018#	1028												
BIT05 =	000040	1017#	1027												
BIT06 =	000100	1016#	1026												
BIT07 =	000200	1015#	1025												
BIT08 =	000400	1014#	1024	6598											
BIT09 =	001000	1013#	1023	6606											
BIT1 =	000002	1031#													
BIT10 =	002000	1012#													
BIT11 =	004000	1011#	6613												
BIT12 =	010000	1010#	6694												
BIT13 =	020000	1009#	5881												
BIT14 =	040000	1008#	6584												
BIT15 =	100000	1007#	1989	1994	1998	6691	6693								
BIT2 =	000004	1030#													
BIT3 =	000010	1029#													
BIT4 =	000020	1028#													
BIT5 =	000040	1027#													
BIT6 =	000100	1026#	6652												
BIT7 =	000200	1025#													
BIT8 =	000400	1024#													
BIT9 =	001000	1023#													
BPTVEC=	000014	1039#													
BRKDAO=	104414	3671	6261	7180#											
BRKDA4=	104415	2808	2904	3006	3134	3223	3344	3512	3644	3674	3793	4153	4239	4497	
		5667	5683	6241	6264	7182#									
BTEOP	017440	5793#	6702												
CHE1	020654	6235	6238#												
CHKCCL	021014	4961	5025	5036	5177	5336	5401	6313#							
CHKCRD=	104411	2590	2646	3899	4308	4402	5217	5384	5444	5580	5733	7174#			
CHKDA	020674	2821	2916	3019	4180	6255#									
CHKDA1	020702	3149	3239	3369	3533	6257#									
CHKECL	020770	4957	5025	5094	5173	5260	5334	5400	6302#						
CHKER	020754	2835	2931	3033	3269	4184	4782	6289#							
CHKHE	020646	2814	2910	3013	3230	4161	4501	6237#							
CHKHE1	020640	3140	3351	3526	3661	3807	6234#								
CHKWC	020730	2825	2921	3023	3252	3700	4176	4793	6275#						
CH. CRD	021372	6495#	7174												
CKSWR =	104406	6583	7170#												
CNT. RD=	104413	3941	3999	4017	4066	4995	5295	6366	7178#	7248	7250				
CNT. RE=	104412	2290	2337	2374	2426	2511	2546	2576	2630	2696	2750	2869	2970	3072	
		3102	3195	3319	3426	3482	3611	3766	3880	3977	4094	4208	4282	4371	
		4463	4570	4636	4722	4743	4821	4901	4947	4978	5015	5044	5084	5113	
		5163	5200	5233	5250	5279	5324	5354	5399	5422	5487	5561	5641	5729	
		5849	7176#	7239											
CN. RDY	021444	6534	6535#	7178											
CN. RST	021426	6532#	7176												

[illegible]

[illegible]

EM31	025062	1206	7301#
EM32	025072	1213	7304#
EM33	025113	1220	7308#
EM34	025133	1227	7312#
EM35	025152	1234	7316#
EM36	025205	1241	7322#
EM37	025235	1248	7328#
EM40	025271	1255	7334#
EM41	025331	1262	7341#
EM43	025366	1276	2131 7347#
EM44	025420	1283	7353#
EM45	025474	1290	7362#
EM46	025542	1297	7370#
EM47	025600	1304	7376#
EM50	025642	1311	7383#
EM51	025671	1318	7388#
EM52	025744	1325	7397#
EM53	025771	1332	7402#
EM54	026034	1339	7409#
EM55	026126	1349	7420#
EM56	026205	1359	7429#
EM57	026244	1368	7436#
EM60	026302	1375	7442#
EM61	026341	1382	7449#
EM62	026376	1389	7455#
EM63	026410	1396	7458#
EM64	026454	1405	7465#
EM65	026517	1412	7472#
EM66	026563	1419	7480#
EM67	026622	1426	7487#
EM70	026635	1178	1433 7490#
EM71	026726	1443	7501#
EM72	027004	1453	7510#
EM73	027046	1460	7517#
EM74	027120	1467	7525#
EM75	027170	1474	7533#
EM76	027240	1481	7541#
EM77	027303	1488	7548#
ERRVEC=	000004	1035#	1867 1868* 1879* 6589 6590* 6592* 6595*
FCHECK	024542	5775	7265 7237#
FFLAG	002604	1814#	2271* 2283* 5773
FTITLE	002542	1786#	
GNS = ***** U		1053	1902 1926 1935 1954 1978 2110 2135 2281 6124 6408 6546 6554
		7162	7163 7164 7165 7166 7168 7170 7171 7172 7174 7176 7178 7180
		7182	7184 7186 7188
GTSWR =	104405	1897	7168#
GT2RG	020422	4930	4937 4943 5004 5011 5066 5074 5080 5144 5160 5225 5246 5305
		5320	5390 5396 5515 6105#
GT3RG	020414	2315	2351 2675 2720 2797 2892 2994 3521 3655 3803 4141 4306 4315
		4399	4409 4414 4683 6104# 6292
GT4RG	020406	2208	2362 2387 2558 2600 2606 2611 2657 2661 2670 2806 2902 3004
		3131	3221 3342 3510 3641 3791 3910 3915 3919 4042 4049 4151 4237
		4495	5665 5681 5898 6103# 6240 6380 6500
HT =	000011	945#	6782 6823
INDX1	002552	1793#	2438* 2447* 3592* 3709 3711* 3717* 3778* 3788* 3872* 3951 3956* 5739*
		5840*	5859 6000 6002*

INX2	002554	1794*	2445*	2449*	3626*	3638*	5755*	5845*	5927	5948*				
IOTVEC=	000020	1040*	1852*	1853*										
LF	= 000012	946*	6817	6823										
MSG1	002412	1744*	2033											
MSG2	002432	1748*	2068											
MSG3	002441	1751*	6543											
MSG4	002466	1756*	2268	6695										
MSG5	002477	1759*	6681											
MSG6	002511	1762*	6699											
NUDRV	004514	2237*	5797											
ODDEVN	002606	1815*	5834*	5883	5886	6028*	6029							
OUTBUF	032724	2762	2787	2829	2831	2883	2925	2927	2985	3027	3029	3115*	3120	3211
		3246	3248	3308	3383	3495*	3499	3625	3755	3938	3942	3946	3988	4121
		4134	4221	4227	4251	4298*	4301	4339	4341	4346	4352	4354	4385	4392
		4436	4438	4477	4763	4803	4805	4917	5293	5366	5376*	5380	5435*	5441
		5499*	5506	5651	5866	5871	5955	5986	7964*					
PC	=:000007	966*	2043*	2074*	2087*	2162*	2208*	2270*	2315*	2351*	2362*	2387*	2558*	2600*
		2606*	2611*	2657*	2661*	2670*	2675*	2708*	2720*	2724*	2797*	2806*	2814*	2821*
		2825*	2835*	2892*	2902*	2910*	2916*	2921*	2931*	2994*	3004*	3013*	3019*	3023*
		3033*	3131*	3140*	3149*	3221*	3230*	3239*	3252*	3269*	3342*	3351*	3369*	3510*
		3521*	3526*	3533*	3641*	3655*	3661*	3700*	3791*	3803*	3807*	3910*	3915*	3919*
		4042*	4048*	4141*	4151*	4161*	4176*	4180*	4184*	4237*	4306*	4315*	4399*	4409*
		441*	4495*	4501*	4648*	4683*	4705*	4708*	4782*	4793*	4930*	4937*	4943*	4957*
		4961*	4965*	5004*	5011*	5025*	5029*	5066*	5074*	5080*	5094*	5098*	5102*	5144*
		5160*	5173*	5177*	5181*	5210*	5225*	5230*	5234*	5246*	5260*	5305*	5320*	5334*
		5338*	5390*	5396*	5400*	5404*	5515*	5559*	5665*	5681*	5726*	5792*	5822*	5898*
		5972*	5991*	6050*	6053*	6063*	6068	6107*	6128*	6206*	6240*	6243*	6266*	6280*
		6292*	6294*	6306*	6317*	6320*	6340*	6344*	6380*	6383*	6401*	6479*	6500*	6662*
		6688*	6697*	6742*	6791*	6798*	6805*	6819*	6821*	7024*	7227	7237*	7255*	7274*
		2162*	7228											
PFSTRT	004372	1839*	5823*	5827*	5954	5985								
PHYDRV	002632													
PIRQ	= 177772	952*												
PIRQVE=	000240	1046*												
PR0	= 000000	969*												
PR1	= 000040	970*												
PR2	= 000100	971*												
PR3	= 000140	972*												
PR4	= 000200	973*												
PR5	= 000240	974*												
PR6	= 000300	975*												
PR7	= 000340	976*												
PS	= 177776	949*	950											
PSW	= 177776	950*												
PWRVEC=	000024	1041*	1858*	1859*	7195*	7196*	7205*	7211*	7223*	7224*				
RDCHR =	104407	1928	7084	7171*										
RDLIN =	104410	1956	7172*											
RESVEC=	000010	1036*												
RKBA	002532	1776*	2789	2882	2984	3120*	3212*	3246	3249	3332*	3499*	3627*	3779*	3938*
		3994*	4060*	4134*	4169	4172	4227*	4297	4392*	4436	4439	4484*	4508	4511
		4758	4917*	5059*	5293*	5380*	5441*	5506*	5574*	5652	5725			
RKCS	002526	1774*	2307*	2586	2642	2715*	2788	2881	2982	3094	3126	3207	3214*	3216
		3305	3503*	3505	3634*	3636	3784*	3786	3874	3976	4106	4146	4230*	4232
		4294	4383	4475	4586	4646	4756	4835	4920*	4923	4941	4993*	5009	5057
		5062	5125	5194	5289	5353	5434	5511*	5540*	5553	5640	5732*	5839	6106
		6238	6313	6315	6316	6365*	6496	6532*	6536	6556	7247*			
RKDA	002534	1777*	2030	2039*	2040	2186*	2301*	2347*	2384*	2396*	2436*	2524*	2547*	2599*

		2643*	2712*	2713*	2792*	2887*	2989*	3096	3209	3331*	3501*	3632*	3668	3673
		3782*	3896*	3931	3934	3937*	3992*	4052	4055	4061*	4135*	4228*	4296	4393*
		4430	4433	4485*	4647*	4654*	4723*	4757	4836*	4918*	4991*	5060*	5128*	5215*
		5226	5290*	5291*	5381*	5442*	5460	5465	5507*	5525	5530	5573*	5657*	5671*
		5731*	5901*	6103	6257	6263	6279	6333*	6364*	6398*	7246*	7249*		
RKDB	002536	1778*	3058	3061	3073	3075	3076	3155	3160	3436	3438	3439	3828	3831
RKDS	002522	1772*	2029	2051	2187	2300	2325	2348	2357	2360	2385	2398	2402	2437
		2485	2522	2549	2596	2604	2653	2659	2714	3519	3653	3801	3875	4025
		4247	4514	4699	4724	4726	5131	5938	5956	6104	6335	6339	6372	6399
		7251												
RKER	002524	1773*	2673	3876	4318	4320	4412	4935	4990	5072	5142	5223	5302	5313
		5388	5450	5454	5597	5603	6105	6289	6302	6304	6305			
RKPRI	002574	1804*	4839	5918	6460									
RKVEC	002576	1809*	2096*	4587	4605*	4617*	4651	4662*	4671*	4681*	4692*	4764	4772*	4780*
		4843	4869*	4878*	5508	5522*	5915							
RKWC	002530	1775*	2791*	2886*	2988*	3122*	3213*	3333*	3500*	3629*	3781*	3939*	3993*	4059*
		4133*	4226*	4295	4391*	4422	4424	4483*	4761*	4916*	4929	5058*	5067	5292*
		5379*	5440*	5505*	5572*	5654*	5670*	5729*	6275	6278				
RO	=%000000	957*	1882*	1884*	1907*	1909*	1947*	1949*	1957*	1959	1985	2000	2008	2034*
		2063*	2105*	2106	2112	2124*	2125	2137	2182*	2190	2205	2215*	2217	2226
		2272*	2276*	2277	2300*	2302	2304	2310	2320	2348*	2349	2355	2437*	2452
		2455	2464	2467	2522*	2528	2549*	2550	2554	2556	2586*	2589*	2609	2614
		2618	2642*	2645*	2667	2678	2681	2776*	2778	2779	2781*	2782	2800*	2803*
		2868*	2895*	2899*	2983*	2997*	3001*	3114*	3128*	3208*	3259	3266*	3336*	3339*
		3383*	3385	3387	3392	3680*	3688	3696*	3763*	3843*	3846*	3874*	3897*	3917
		3924	3926	3940*	3987*	3998*	4024*	4039*	4065*	4121*	4126*	4129*	4169	4171
		4220*	4223*	4225*	4234*	4384*	4389*	4396*	4476*	4481*	4516*	4532*	4538	4587*
		4588*	4589*	4646*	4655*	4663	4672	4674	4694	4696	4731	4733	4756*	4767*
		4773	4835*	4850*	4859	4863*	4885	5057*	5061*	5078	5125*	5139*	5158	5194*
		5216*	5244	5264	5266	5267	5289*	5294*	5318	5353*	5382*	5394	5434*	5443*
		5508*	5509*	5510*	5553*	5571*	5576*	5586	5640*	5659*	5661	5675*	5677	5697
		5824*	5825	5828	5830	5831	5832	5839*	5894	5903*	5932	5934	6060*	6063
		6153	6154*	6157	6158*	6168*	6176*	6180*	6183*	6187*	6473*	6475*	6714	6715*
		6716*	6723*	6724*	6725*	6726*	6727*	6728	6733	6738*	6740*	6744	6746	6774
		6775*	6776	6779*	6837	6847*	6851	6867	6868	6881*	7146	7147*	7148	7149*
		7150*	7151*	7152*	7197	7222*	7260*	7261	7263	7267*	7268*	7269	7270	7272
R1	=%000001	958*	1883*	1885*	1908*	1910*	1948*	1950*	1958*	2010*	2036*	2039	2040	2061*
		2183*	2186	2227*	2255*	2257	2266	2309*	2312*	2319*	2323*	2397*	2400*	2452*
		2453	2455	2457*	2470	2476*	2480	2528*	2529	2537	2548*	2552*	2714*	2716
		2788*	2793*	2795	2801	2839	2842	2881*	2888*	2890	2896	2943	2946	2982*
		2990*	2992	2998	3065	3068	3097*	3098	3115	3123	3132	3164*	3207*	3273
		3276	3306*	3307	3313*	3331	3343	3358	3363	3365	3393	3411*	3479*	3480
		3495	3501	3511	3522	3539*	3551*	3555*	3556	3559*	3560	3594*	3595	3598*
		3632	3642	3656	3682	3712*	3713	3718*	3719*	3720	3758*	3761*	3782	3792
		3804	3814	3816	3818	3828	3830	3842*	3847*	3851*	3854*	3875*	3906	3913
		3976*	3996*	4016*	4045	4063*	4122*	4128	4221*	4222*	4249*	4258	4271*	4294*
		4302*	4304	4313	4327	4383*	4394*	4397	4407	4425	4443	4446	4475*	4488*
		4491	4586*	4590	4592*	4595	4651*	4652*	4653*	4699*	4700*	4701	4707	4759*
		4760*	4762	4795*	4787	4789	4837*	4858	4866*	4884	4913*	4914*	4918	4990*
		5001	5366*	5372*	5439*	5442	5459*	5460	5464	5504*	5507	5524*	5525	5529
		5554*	5608*	5651*	5653*	5658	5719*	5730	5745	5757*	5758	5864*	5877	5879
		5904	5938*	5939	5941	5956*	5957*	5964	5971	6163	6165*	6167*	6168	6169*
		6170*	6171*	6172*	6173*	6174	6177*	6178	6181*	6182*	6183	6186*	6234	6474*
		6477*	6838	6851*	6852	6856	6880*	7198	7221*					
R2	=%000002	959*	1959*	1960*	1964	1973	1985*	1986*	1987	2002	2004	2038*	2057*	2060
		2184*	2196	2225*	2464*	2465	2467	2469*	2470	2472	2490	2498	2789*	2790*

		2829	2832	2882*	2884*	2925	2928	2984*	2986*	3027	3030	3095*	3170*	3209*
		3210*	3316*	3417*	3478*	3541	3545*	3550*	3595*	3601*	3609*	3668	3670	3713*
		3714*	3720*	3721*	3876*	4027*	4028*	4123*	4127*	4130*	4145*	4148*	4250*	4253*
		4254	4260	4295*	4299*	4324	4326	4330*	4331*	4332	4334	4385*	4388*	4477*
		4478	4484	4508	4510	4517	4522	4530	4534	4537	4757*	4762*	4787	4790
		4838*	4846	4854	4864*	4880	4922*	4925*	5056*	5064*	5126*	5128	5133*	5196*
		5197*	5215	5240*	5241	5302*	5303	5309*	5367*	5373*	5555*	5571	5583	5607*
		5655*	5656*	5657	5666	5671	5682	5720*	5742	5757	5758*	5759*	5861*	5863*
		5881	5888	5901	5905*	5915*	5916*	5917*	6164	6174*	6175*	6176	6178*	6179*
		6180	6185*	6839	6850*	6854*	6857	6864*	6865*	6866	6871*	6879*	7199	7220*
R3	=%000003	960*	1961*	1968	1981	1983*	1989*	1992	1994*	1996	1998*	2042*	2071*	2073*
		2217*	2218*	2219*	2220*	2221*	2222*	2442*	2472	2475*	2491	2497	2523*	2532*
		2762*	2778*	2779*	2780*	2784*	2785*	2787*	2790	2883*	2884	2935	2940	2985*
		2986	3037	3041	3094*	3124*	3305*	3334*	3337	3420	3423	3475*	3552	3554*
		3562*	3625*	3627	3684	3687	3694	3755*	3779	3814	3817	3870*	3895	3942
		3945	3953	3954	3957	3958	3959	3988*	3994	4006	4009	4060	4073	4076
		4124*	4126	4128*	4129	4251*	4254	4262	4269	4296*	4300*	4332	4335	4386*
		4387*	4388	4478*	4480*	4839*	4854	4880	5368*	5371*	5372	5597*	5598	5599*
		5660*	5663*	5676*	5679*	5865*	5868*	5870*	5906*	5954*	5966*	5985*	5997*	6126
		6840	6848*	6849*	6863*	6866*	6875*	6876*	6878*	6925	6934*	6940*	6941*	6944*
		6949*	6950*	6951	6960*	7079	7081*	7082	7085*	7086	7093*	7094	7096	7104
		7108	7110*	7116	7118	7120*	7123*	7200	7219*					
R4	=%000004	961*	1962*	1969*	2037*	2058	2062*	2441*	2460*	2521*	2524	2531*	2536	2587*
		2598*	3096*	3123*	3211*	3212	3257	3262	3265	3308*	3332	3358	3364	3477*
		3563*	3682*	3683*	3684	3686	3756*	3848	3850*	3853*	3894*	3895*	3896	3931
		3933	3937	3989*	3990	4061	4073	4075	4077	4106*	4137*	4139	4188	4191
		4297*	4301*	4339	4342	4428*	4429*	4430	4432	4538*	4539*	4540	4758*	4763*
		4803	4806	4843*	4844*	4845*	5586*	5587*	5588	5591	5652*	5658*	5672*	5696
		5893*	5896*	5914*	5922*	6926	6928*	6929*	6930*	6931	6932*	6946	6948*	6956*
		6959*	7201	7218*										
R5	=%000005	962*	1963*	1964	1967*	1990	2439*	2462*	2477*	2525*	2526*	2652*	2655*	2711*
		2718*	3098*	3099*	3165*	3166	3168*	3194*	3218*	3236*	3237*	3307*	3314*	3317*
		3412*	3413	3415*	3480*	3481*	3540*	3544*	3556*	3557*	3560*	3561*	3606*	3722*
		3764*	3855*	3905*	3908*	3990*	3991*	3992	4006	4008	4010	4051*	4052	4054
		4252*	4267*	4531*	4557*	4764*	4765*	4766*	4951*	4852*	5583*	5584*	5585*	5588
		5590	5598*	5601*	5602	5692*	5693	5694*	5695	5725*	5730*	5742	5746	5775*
		5866*	5867*	5871*	5888*	5892*	5955*	5964	5980*	5986*	5988	5990	5996*	6255*
		6256*	6257	6260	6841	6843*	6845*	6852*	6856*	6871	6877*	6927	6933*	6935*
		6937*	6938*	6939*	6940	6958*	7202	7217*	7253	7257*	7265*			
R6	=%000006	963*	965	1846*	1847*	1848	2269*	2272	5971*	5973	5990*	5992	6203*	6382*
		6689*	6691*	6692*	6696*									
		964*	966											
R7	=%000007	1801*	3870	3959										
SEEK0	002566	1802*												
SEEK1	002570	1803*												
SEEK2	002572	1803*	3954											
SHFTRT	020612	2270	4705		5972	5991	6201*	6688	6697					
SIMUL	002540	1785*	2018*	2027*	3309	3407	3596	3706	3759	3840	3974			
SIZEF	024666	5822	7259*											
SIZYET	002634	1840*	2015*	2027*	5820*									
SP	=%000006	965*	1850*	1867*	1875*	1879	1957	2058*	2071	2072	2105	2107	2112*	2124
		2137*	2142	4582*	4583*	4600	4603	4612	4613	4619*	4620*	4667	4669	4687
		4690	4704*	4706	4707*	4709	4718*	4719*	4777	4779	4799*	4800*	4846*	4847*
		4871*	4872*	4877	5520	5521	5536*	5537*	5634	5636*	5637*	5873*	5874*	5919*
		5919*	5926	6057*	6126*	6153*	6157*	6163*	6164*	6195	6186	6187	6319*	6342*
		6374*	6375*	6377	6410*	6411*	6429	6430*	6457	6458*	6460*	6461*	6503*	6548*
		6549*	6556*	6589*	6592	6594	6595	6623	6624	6628*	6657	6669*	6672*	6675*

[illegible]

TST.SI= 104420	2640	2706	2760	2873	2980	3112	3205	3329	3492	3622	3776	3890	4104
	4218	4292	4391	4473	4580	4753	4831	4911	4999	5054	5123	5213	5364
	5432	5497	5718	7198									
TST1 004376	2098	2180											
TST10 005276	2461	2487	2498	2495	2510								
TST11 005464	2534	2545											
TST12 005540	2557	2575											
TST13 005666	2616	2629											
TST14 006032	2679	2695											
TST15 006124	2749												
TST16 006362	2840	2857											
TST17 006610	2944	2969											
TST2 004514	2246												
TST20 007120	3074	3093											
TST21 007316	3193												
TST22 007570	3274	3302											
TST23 010140	3437	3471											
TST24 010420	3598												
TST25 011030	3707	3723	3751										
TST26 011336	3841	3868											
TST27 011656	3972												
TST3 004672	2289												
TST30 012144	3975	4074	4093										
TST31 012400	4189	4207											
TST32 012604	4248	4268	4281										
TST33 013052	4350	4370											
TST34 013334	4444	4462											
TST35 013612	4515	4533	4558	4569									
TST36 013744	4634												
TST37 014270	4732	4742											
TST4 005004	2321	2336											
TST40 014504	4804	4820											
TST41 014710	4875	4900											
TST42 015072	4977												
TST43 015200	5043												
TST44 015354	5112												
TST45 015512	5137	5193											
TST46 015720	5265	5278											
TST47 016072	5352												
TST5 005072	2361	2373											
TST50 016250	5421												
TST51 016376	5463	5486											
TST52 016550	5552												
TST53 016744	5621												
TST54 017236	5633	5691	5716										
TST55 017374	5771												
TST56 017420	5788												
TST57 017466	5818												
TST6 005120	2386	2395											
TST7 005156	2399	2425											
TYERM 020440	2043	2074	6121										
TYPOS = 104404	6058	7166											
TYPE = 104400	1891	1924	1933	1952	1976	2032	2054	2067	2108	2128	2130	2133	2268
	2279	2284	6056	6059	6122	6406	6542	6544	6552	6653	6681	6695	6699
	6713	6730	6732	6735	6737	6741	6748	6787	6882	6952	6991	6992	6995
	7008	7019	7038	7091	7097	7102	7106	7111	7112	7114	7117	7121	7162

[illegible]

```

$MSWR      024252
$MXCNT     022022
$NULL      001154
$NWTST=    000001

```

6992	7133#											
6621	6630#											
1091#	6794	6823										
2166#	2168	2240#	2242	2286#	2333#	2370#	2390#	2392	2408#	2410	2505#	2507
2542#	2566#	2568	2622#	2624	2685#	2687	2734#	2736	2846#	2848	2953#	2955
3079#	3081	3174#	3176	3283#	3285	3444#	3446	3569#	3571	3728#	3730	3850#
3862	3964#	3966	4085#	4087	4196#	4198	4275#	4277	4361#	4363	4449#	4451
4561#	4563	4625#	4627	4736#	4738	4810#	4812	4892#	4894	4969#	4971	5034#
5036	5106#	5108	5186#	5188	5271#	5273	5343#	5345	5409#	5411	5477#	5479
5543#	5545	5611#	5613	5706#	5708	5765#	5767	5780#	5782	5802#	5804	

\$OCNT	023354
\$OMODE	023356
\$OVER	022006
\$PASS	001100
\$POWER	024532
\$PWRAD	024520
\$PWDN	024360
\$PWMG	024514
\$PWUP	024432
\$GLES	001212
\$RDCHR	023642
\$RDEC=	*****
\$RDLIN	023762
\$RDOCT=	*****
\$RDSZ =	000022
\$REGAD	001160
\$REGO	001162

6924*	6953*	6966*										
6919*	6923*	6928	6931*	6942*	6968*							
6585	6501	6609	6619	6627*								
1066*	6048*	6049*	6057	6070	6615	6631						
7226	7233*											
7228*												
1858	7195*	7223										
7226*												
7205	7211*											
1109*	6823	7038	7114	7131								
7051*	7171											
7173												
7079*	7172											
7173												
7072*												
1095*												
1097*	2196*	2304*	2325*	2357*	2402*	2480*	2485*	2490*	2497*	2536*	2554*	2617*
2680*	2831*	2841*	2927*	2939*	2945*	3029*	3040*	3060*	3067*	3159*	3248*	3259*
3260*	3275*	3363*	3391*	3422*	3438*	3688*	3689*	3816*	3830*	3927*	3933*	3944*
4008*	4054*	4075*	4171*	4190*	4258*	4259*	4320*	4326*	4334*	4341*	4352*	4424*
4432*	4438*	4445*	4510*	4521*	4540*	4595*	4663*	4674*	4696*	4706*	4726*	4733*
4773*	4789*	4805*	4858*	4884*	5266*	5312*	5453*	5464*	5529*	5590*	5602*	5695*
5745*	5934*	5941*	5973*	5992*	6106*	6278*	6304*	6315*	6339*	7778	7780	7793

SREG1 001164

1098*	2491*	2498*	2537*	2618*	2691*	2832*	2842*	2928*	2939*	2946*	3030*	3041*
3061*	3068*	3075*	3076*	3160*	3249*	3261*	3276*	3364*	3392*	3423*	3439*	3696*
3817*	3831*	3926*	3934*	3945*	4009*	4055*	4076*	4172*	4191*	4260*	4327*	4335*
4342*	4353*	4425*	4433*	4439*	4446*	4511*	4522*	4536*	4709*	4790*	4806*	4859*
4885*	5267*	5313*	5454*	5465*	5530*	5591*	5603*	5693*	5746*	6105*	6279*	6305*
6316*	7780	7783	7788	7791								
1105*	2807*	2903*	3005*	3132*	3222*	3343*	3511*	3642*	3670*	3673*	3792*	4152*
4238*	4496*	5666*	5662*	6165	6169	6234*	6237*	6260*	6263*			

REG 10 001202

SREG11 001204
SREG2 001166

4238*	4438*	3888*	3882*	8183	8183	8234*	8237*	8289*	8283*			
1106#												
1099#	2940*	3042*	3262*	3365*	3393*	3687*	3818*	3946*	4010*	4077*	4262*	4354*
4523*	4537*	4929*	4938*	5003*	5067*	5075*	5145*	5226*	5306*	5391*	5696*	6104*

```

$REG3      001170
$REG4      001172
$REG5      001174
$REG6      001176
$REG7      001200
$RTNAD     020364
$R2A      = *****
$SAVE     = *****
$SAVE6    024530
$SCOPE     021552

```

[illegible]

E13

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 162
 DZRKKC.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

\$SETUP= 000117	1843#	1851	1852	1854	1856	1858	1860	1861	1863	1891	1892	6046	6583
	6975	7137											
\$STUP = 177777	1843#												
\$SVLAD 021760	6593	6622#											
\$SVPC = 000000	931#	936											
\$SWR = 165400	879#	898	906	907	908	909	910	911	912	913	1107	1108	1109
	1860	1861	1863	1864	2181	2247	2290	2337	2374	2396	2426	2511	2546
	2576	2630	2696	2750	2868	2970	3094	3194	3303	3472	3589	3752	3869
	3973	4094	4208	4282	4371	4463	4570	4635	4743	4821	4901	4978	5044
	5113	5194	5279	5353	5422	5487	5553	5622	5717	5772	5789	5819	6041
	6047	6062	6068	6070	6574	6575	6576	6577	6578	6584	6596	6598	6599
	6602	6603	6604	6611	6612	6613	6624	6627	6630	7229			
\$SWRMK= 000000	913	914	6578	6579	6600								
\$TIMES 001206	1107#	1860#	2247#	3472#	3589#	3752#	3869#	3973#	4635#	5622#	5717#	5772#	5789#
	5919#	6047#	6611#	6618	6621#	6630							
\$TKB 001146	1088#	6973	6984	7001	7055	7061							
\$TKS 001144	1087#	5672	5673#	5689	5692	6973	6982	6998	7022#	7053	7059		
\$TN = 000060	879#	898	2166	2181#	2240	2247#	2286	2290#	2321	2333	2337#	2361	2370
	2374#	2386	2390	2396#	2399	2408	2426#	2461	2483	2498	2495	2505	2511#
	2534	2542	2546#	2557	2566	2576#	2616	2622	2630#	2679	2685	2696#	2734
	2750#	2840	2846	2868#	2944	2953	2970#	3074	3079	3094#	3174	3194#	3274
	3283	3303#	3437	3444	3472#	3569	3589#	3707	3723	3728	3752#	3841	3860
	3869#	3964	3973#	3975	4074	4085	4094#	4189	4196	4208#	4248	4269	4275
	4282#	4350	4361	4371#	4444	4449	4463#	4515	4533	4558	4561	4570#	4625
	4635#	4732	4736	4743#	4804	4810	4821#	4875	4892	4901#	4969	4978#	5034
	5044#	5106	5113#	5137	5186	5194#	5265	5271	5279#	5343	5353#	5409	5422#
	5463	5477	5487#	5543	5553#	5611	5622#	5633	5691	5706	5717#	5765	5772#
	5780	5789#	5802	5819#									
\$TPB 001152	1090#	6812#	6823										
\$TPFLG 001157	1094#	6770	6823										
\$TPS 001150	1089#	6810	6823										
\$TRAP 024274	1856	7146#											
\$TRP = 000021	7154#	7163#	7164#	7165#	7166#	7167#	7168	7169#	7170	7171#	7172#	7173#	7174
	7175#	7176	7177#	7178	7179#	7180	7181#	7182	7183#	7184	7185#	7186	7187#
	7188	7189#											
	7151	7161#											
\$TRPAD 024316	1067#	2248#	6046#	6573	6600	6622#	6627	6631	6649				
\$STNM 001102	7081	7082	7094	7112	7126	7130#							
\$TTYIN 024216	7167												
\$TYPBN= ***** U	6836#	7166											
\$TYPDS 022706	6770#	7154	7162										
\$TYPE 022466	6791	6798	6805	6810#	6811	7024							
\$TYPEC 022636	6816	6818	6821#										
\$TYPEX 022704	6922#	7163											
\$TYPOC 023156	6921	6924#	7165										
\$TYPON 023172	6917#	7164											
\$TYPOS 023132	6587#												
\$XTSTR 021564	6062#												
\$SET4= 000000	6918#	6922#	6932	6967#									
\$OFILL 023355	6584												
\$DOCAT= ***** U	931	932#	934#	936#	1049#	1053#	1064#	1112	1747#	1849	1863	1864	1903#
	1936#	1955#	1979#	2111#	2136#	2253	3167	3414	3543	4029	4591	4953	5960
	5862	5869	5967	6070	6074	6125#	6376	6409#	6555#	6630	6631	6668	6751#
	6823	6890#	6973	7130#	7131	7137	7207	7231	7774#	7961#	7964#		

COMMENT ENDCOM ERROR	1047#														
	1047#														
	941#	2197	2209	2305	2316	2326	2352	2358	2363	2388	2403	2481	2486	2492	2499
	2538	2555	2559	2592	2601	2607	2612	2619	2648	2658	2662	2671	2676	2682	2709
	2721	2725	2798	2810	2817	2823	2827	2833	2837	2843	2893	2906	2912	2918	2923
	2929	2933	2941	2947	2995	3008	3015	3021	3025	3031	3035	3043	3062	3069	3077
	3136	3142	3151	3161	3225	3232	3241	3250	3254	3263	3271	3277	3346	3353	3366
	3371	3395	3424	3440	3514	3523	3528	3535	3646	3658	3663	3676	3691	3702	3795
	3805	3809	3819	3832	3901	3911	3916	3920	3928	3935	3947	4011	4043	4049	4056
	4078	4142	4155	4163	4173	4178	4182	4186	4192	4241	4263	4307	4311	4316	4321
	4328	4336	4343	4355	4400	4404	4410	4415	4426	4434	4440	4447	4499	4503	4512
	4526	4542	4596	4614	4649	4664	4675	4684	4697	4710	4727	4734	4774	4784	4791
	4795	4807	4860	4886	4931	4939	4944	4959	4963	4966	5005	5012	5027	5031	5068
	5076	5081	5096	5100	5104	5146	5161	5175	5179	5183	5212	5219	5227	5232	5235
	5247	5262	5268	5307	5314	5321	5336	5340	5386	5392	5397	5402	5406	5446	5455
5466	5516	5531	5560	5582	5592	5604	5668	5684	5698	5727	5735	5747	5776	5899	
5935	5942	5976	5994	7238	7256										
ESCAPE	1047#														
GETPRI	1047#														
GETSWR	879#	1047#	1892#												
MESSAGE	2165#	2168	2240#	2242	2390#	2392	2407#	2410	2504#	2507	2566#	2568	2622#	2624	2685#
	2687	2734#	2736	2846#	2848	2952#	2955	3079#	3081	3174#	3176	3282#	3285	3444#	3446
	3568#	3571	3727#	3730	3859#	3862	3963#	3966	4084#	4087	4195#	4198	4274#	4277	4361#
	4363	4449#	4451	4561#	4563	4625#	4627	4736#	4738	4810#	4812	4892#	4894	4969#	4971
	5034#	5036	5106#	5108	5186#	5188	5271#	5273	5343#	5345	5409#	5411	5476#	5479	5542#
	5545	5611#	5613	5705#	5708	5765#	5767	5780#	5782	5801#	5804				
MULT	1047#														
NEWST	1047#	2166	2240	2286	2333	2370	2390	2408	2505	2542	2566	2622	2685	2734	2846
	2953	3079	3174	3283	3444	3569	3728	3860	3964	4085	4196	4275	4361	4449	4561
	4625	4736	4810	4892	4969	5034	5106	5186	5271	5343	5409	5477	5543	5611	5706
	5765	5780	5802												
POP	1047#	6877	7216	7217											
PRIOR7	1129#	4582	4619	4717	4799	4871	5535	5636	587						

[illegible]

H13

 MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 167
 DZRKKC.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ADD	2061	2073	2220	2227	2259	3168	3237	3260	3415	3544	3555	3559	3609	3689	3712
	3714	3719	3721	3851	3854	4128	4253	4259	4387	4539	4786	4864	5133	5240	5371
	5584	5607	5757	5759	5796	5905	5996	6319	6342	6382	6411	6430	6458	6503	6690
	6692	6727	6780	6856	6920	6930	7009	7018	7271						
ASL	2276	6724	6725	6726	7032	7033	7034	7150							
ASLB	6861														
ASR	6170	6171	6172	6173	6177										
BCC	6862														
BEQ	1916	1919	1966	1974	1991	2003	2041	2053	2141	2146	2206	2263	2265	2350	2361
	2448	2450	2454	2461	2463	2466	2471	2527	2530	2557	2605	2610	2616	2660	2668
	2674	2679	2717	2830	2840	2926	2937	2944	3028	3039	3059	3066	3074	3158	3247
	3258	3274	3310	3362	3421	3437	3520	3597	3654	3669	3685	3723	3760	3802	3815
	3829	3914	3918	3925	3932	3943	4007	4046	4053	4074	4170	4189	4256	4268	4314
	4319	4325	4333	4340	4350	4408	4413	4423	4431	4437	4444	4509	4520	4533	4535
	4558	4673	4702	4725	4732	4788	4804	4942	5010	5079	5159	5239	5265	5310	5319
	5395	5452	5463	5528	5589	5600	5740	5744	5756	5774	5795	5826	5878	5882	5884
	5933	5940	5965	6030	6061	6239	6258	6276	6290	6303	6314	6400	6599	6601	6603
	6607	6616	6648	6653	6668	6671	6674	6680	6729	6734	6747	6783	6818	6947	6989
	7016	7031	7100	7242	7262	7264									
BGE	6619														
BGT	4855	6052	6870	6954	7028	7069	7273								
BHI	6605	6655													
BIC	1960	1986	2219	2221	2222	2457	2469	3551	3683	3718	3847	4700	5309	5585	5587
	5599	5601	5892	5957	6049	6167	6175	6179	6182	6689	6691	6693	6944	6985	7002
	7029	7036	7062	7070	7243										
BIS	1989	2713	3313	3314	3598	3601	3761	3895	3991	4654	4760	4914	5197	5291	5576
	5656	5694	6864	6865	6949	6950	7036	7245	7267	7268					
BISB	6716														
BIT	1990	2016	2051	2126	2140	2145	2257	2310	2320	2349	2355	2360	2398	2453	2465
	2529	2550	2556	2596	2604	2609	2653	2659	2667	2716	3519	3653	3801	3906	3913
	3917	4025	4045	4247	4313	4318	4407	4412	4514	4672	4694	4724	4935	5001	5072
	5223	5241	5244	5264	5303	5388	5689	5877	5881	5932	5939	6160	6238	6335	6372
	6399	6404	6540	6584	6598	6606	6613	6652	6660	6667	6670	7241	7251		
BITB	6806														
BLE	4881														
BLO	7095														
BLOS	7083														
BLT	5833	6797	6853	6869	6955	7026	7067								
BMI	2386	2802	2898	3000	3127	3217	3338	3506	3637	3787	4147	4233	4492	4924	5063
	5662	5678	5895	6497	6537	6860									
BNE	1849	1872	1886	1890	1894	1896	1911	1914	1921	1951	1970	1982	1988	1993	1997
	2005	2011	2017	2064	2066	2086	2127	2192	2228	2253	2258	2260	2303	2311	2313
	2321	2324	2356	2399	2401	2456	2458	2468	2473	2533	2551	2553	2597	2599	2654
	2656	2783	2804	2900	3002	3129	3167	3171	3219	3267	3340	3386	3389	3404	3408
	3414	3418	3508	3543	3547	3553	3564	3639	3697	3707	3710	3789	3841	3844	3849
	3856	3907	3909	3952	3955	3960	3975	4026	4029	4040	4131	4149	4224	4235	4248
	4272	4390	4482	4494	4515	4695	4853	4867	4926	4936	5002	5065	5073	5135	5143
	5224	5242	5245	5304	5374	5389	5609	5664	5680	5691	5760	5821	5936	5860	5869
	5887	5897	5907	5923	5930	5967	5989	5998	6001	6161	6205	6336	6338	6373	6376
	6379	6405	6433	6464	6476	6478	6499	6539	6541	6585	6614	6661	6717	6739	6777
	6785	6793	6807	6814	6858	6945	6981	6987	7007	7014	7021	7058	7064	7087	7089
	7105	7109	7119	7215	7252										
BPL	2188	2278	2719	2796	2891	2993	4140	4305	4398	4591	5132	5829	5890	6665	6771
	6811	6844	6874	6943	6983	6999	7054	7060							
BP	1874	1898	1901	1917	1925	1931	1934	1938	1953	1977	1980	1995	2001	2007	2012
	2019	2022	2050	2069	2098	2109	2134	2261	2280	2314	2478	2483	2489	2495	2502

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 168
 DZRKKC.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

	2534	2540	3315	3390	3558	3605	3762	3852	4559	4599	4610	4666	4686	4776	4875
	4890	5137	5518	5633	5862	5885	5924	5952	5979	5984	6123	6155	6235	6381	6402
	6407	6413	6534	6545	6553	6587	6593	6596	6609	6612	6722	6749	6773	6790	6800
	6809	6816	6855	6872	6921	6936	6957	7010	7037	7039	7065	7098	7107	7113	7115
	7207	7231	7244	7266											
CLC	6202														
CLR	1847	1860	1861	1884	1909	1949	2015	2018	2035	2036	2037	2084	2090	2092	2183
	2184	2215	2251	2271	2309	2319	2397	2438	2439	2445	2477	2525	2548	2587	2652
	2711	2784	2800	2868	2895	2938	2939	2983	2997	3114	3159	3194	3261	3336	3391
	3475	3494	3562	3592	3626	3717	3756	3778	3853	3872	3905	3944	3987	3998	4024
	4065	4145	4222	4225	4396	4480	4490	4536	4863	4922	5056	5540	5555	5660	5673
	5676	5719	5794	5823	5834	5840	5845	5861	5867	5893	5914	6032	6046	6047	6334
	6363	6473	6474	6495	6535	6611	6625	6682	6715	6847	6850	6934	6996	6997	7080
	7103	7213	7259												
CLRB	6610	6789	6815	6976	7110	7120									
CMP	1848	1871	1895	1964	1987	2002	2040	2072	2107	2142	2455	2467	2470	2472	2614
	2678	2782	2829	2839	2925	2943	3027	3037	3058	3065	3246	3273	3358	3420	3541
	3668	3684	3814	3828	3924	3931	3942	3954	3959	4006	4052	4073	4169	4188	4254
	4332	4339	4346	4430	4436	4443	4508	4517	4600	4603	4612	4613	4667	4669	4687
	4690	4701	4731	4777	4779	4787	4803	4854	4877	4880	4941	5009	5078	5158	5318
	5394	5450	5460	5520	5521	5525	5588	5634	5742	5793	5832	5926	5964	6029	6257
	6313	6594	6618	6654	6683	6701	6868	6980	6986	7006	7013	7025	7027	7057	7063
	7066	7068	7082	7094	7272										
CMPB	1915	1918	3166	3413	6600	6604	6782	6784	6792	6813	6817	6988	7020	7086	7104
	7108	7118													
DEC	2223	4866	5966	5997	6050	6432	6463	6700	6723	7093					
DECB	6796	6799	6942	6953											
EMT	941														
HALT	1053	2114	2148	6666	6772	7206	7230								
INC	1885	1889	1910	1950	1967	1969	1983	1984	1999	2010	2056	2057	2062	2063	2225
	2283	2312	2323	2400	2447	2449	2460	2462	2475	2476	2526	2531	2532	2552	2598
	2655	2718	2781	2803	2899	3001	3099	3128	3164	3165	3170	3218	3266	3317	3339
	3388	3403	3411	3412	3417	3481	3507	3539	3540	3545	3554	3557	3561	3563	3638
	3696	3711	3722	3788	3842	3843	3850	3855	3908	3956	4028	4039	4051	4127	4130
	4148	4223	4234	4267	4271	4331	4389	4429	4481	4493	4532	4557	4852	4925	5064
	5238	5373	5459	5524	5608	5663	5679	5739	5755	5790	5820	5827	5838	5868	5896
	5906	5922	5948	5980	6002	6028	6048	6204	6256	6337	6375	6378	6475	6498	6539
	6617	6650	6854	6948	6956	7035	7214								
	6477	6622	6647	6819											
INCB	942														
IOT	1057	1922	2088	2115	2143	2152	2254	3716	3724	5797	5837	6003	6031	6068	6684
JMP	6702														
JSR	2043	2074	2087	2162	2208	2270	2315	2351	2362	2387	2558	2600	2606	2611	2657
	2661	2670	2675	2708	2720	2724	2797	2806	2814	2821	2825	2835	2892	2902	2910
	2916	2921	2931	2994	3004	3013	3019	3023	3033	3131	3140	3149	3221	3230	3239
	3252	3269	3342	3351	3369	3510	3521	3526	3533	3641	3655	3661	3700	3791	3803
	3807	3910	3915	3919	4042	4048	4141	4151	4161	4176	4180	4194	4237	4306	4315
	4399	4409	4414	4495	4501	4648	4683	4705	4708	4782	4793	4930	4937	4943	4957
	4961	4965	5004	5011	5025	5029	5066	5074	5080	5094	5098	5102	5144	5160	5173
	5177	5181	5210	5225	5230	5234	5246	5260	5305	5320	5334	5338	5390	5396	5400
	5404	5515	5559	5665	5681	5726	5775	5792	5822	5898	5972	5991	6063	6240	6292
	6380	6401	6500	6662	6688	6697	6791	6798	6805	7024	7237	7255	7265		
MCV	1846	1850	1852	1853	1854	1855	1856	1857	1858	1859	1863	1864	1867	1868	1869
	1870	1875	1877	1878	1879	1882	1883	1907	1908	1937	1947	1948	1957	1958	1961
	1962	1963	1994	1998	2021	2027	2028	2031	2034	2039	2039	2042	2058	2071	2099
	2094	2096	2105	2112	2124	2137	2182	2186	2196	2217	2247	2248	2255	2266	2269

NEG
NCP
RESET
ROL

2272	2300	2301	2304	2307	2325	2347	2348	2357	2384	2396	2402	2436	2437	2441
2442	2452	2464	2480	2485	2490	2491	2497	2498	2521	2522	2523	2524	2528	2537
2547	2549	2554	2586	2588	2589	2617	2618	2642	2643	2645	2680	2681	2712	2714
2715	2762	2776	2778	2779	2785	2787	2788	2789	2790	2791	2792	2793	2807	2831
2922	2841	2842	2881	2882	2883	2884	2886	2887	2888	2903	2927	2928	2940	2945
2946	2982	2984	2985	2986	2988	2989	2990	3005	3029	3030	3040	3041	3042	3050
3061	3067	3068	3075	3076	3094	3095	3096	3097	3098	3100	3115	3120	3122	3123
3124	3132	3160	3207	3208	3209	3210	3211	3212	3213	3214	3222	3236	3248	3249
3259	3262	3275	3276	3303	3305	3306	3307	3308	3316	3331	3332	3333	3334	3343
3363	3364	3365	3382	3383	3384	3392	3393	3422	3423	3438	3439	3472	3473	3477
3478	3479	3480	3495	3499	3500	3501	3503	3511	3522	3550	3556	3560	3589	3590
3594	3595	3603	3606	3607	3625	3627	3629	3632	3634	3642	3656	3670	3673	3680
3682	3686	3687	3688	3713	3720	3752	3753	3755	3758	3763	3764	3779	3781	3782
3784	3792	3804	3816	3817	3818	3830	3831	3846	3869	3870	3874	3875	3876	3877
3894	3896	3897	3926	3927	3933	3934	3937	3938	3939	3940	3945	3946	3973	3976
3988	3989	3990	3992	3993	3994	3996	4008	4009	4010	4016	4027	4054	4055	4059
4060	4061	4063	4075	4076	4077	4106	4121	4122	4123	4124	4126	4129	4133	4134
4135	4137	4152	4171	4172	4190	4191	4220	4221	4226	4227	4228	4230	4238	4249
4250	4251	4252	4258	4260	4262	4294	4295	4296	4297	4298	4299	4300	4301	4302
4320	4326	4327	4330	4334	4335	4341	4342	4352	4353	4354	4383	4384	4395	4386
4388	4391	4392	4393	4394	4424	4425	4428	4432	4433	4438	4439	4445	4446	4475
4476	4477	4478	4483	4484	4485	4488	4496	4510	4511	4516	4521	4522	4523	4531
4537	4538	4540	4582	4583	4586	4587	4588	4589	4592	4595	4605	4617	4619	4620
4635	4646	4647	4651	4652	4653	4655	4662	4663	4671	4674	4681	4692	4696	4699
4704	4706	4707	4709	4718	4719	4723	4726	4733	4756	4757	4758	4759	4761	4762
4763	4764	4765	4766	4767	4772	4773	4780	4789	4790	4799	4800	4805	4806	4833
4835	4836	4837	4838	4839	4843	4844	4845	4846	4847	4850	4851	4858	4859	4869
4871	4872	4878	4884	4885	4913	4916	4917	4918	4920	4929	4938	4990	4991	4993
5003	5057	5058	5059	5060	5061	5067	5075	5125	5126	5128	5139	5145	5194	5195
5196	5198	5215	5216	5226	5266	5267</								

ROR	6203														
RTI	1876	2147	4584	4621	4720	4801	4848	4873	5538	5638	5875	5920	6188	6403	6435
	6466	6467	6501	6504	6559	6629	6676	6781	6885	6963	7023	7071	7127	7229	
RTS	6107	6128	6206	6243	6266	6280	6294	6306	6317	6320	6340	6344	6383	6479	6742
	6821	7152	7257	7274											
SUB	2218	6549	6658	6686	6852										
SWAB	6181														
TRAP	7154	7163	7164	7165	7166	7168	7170	7171	7172	7174	7176	7178	7180	7182	7184
	7186	7188													
TST	1893	1913	1929	1968	1973	1981	1992	1996	2004	2029	2030	2060	2065	2085	2106
	2125	2190	2205	2226	2252	2262	2264	2277	2302	2673	2935	3073	3155	3257	3265
	3309	3385	3387	3407	3436	3552	3596	3694	3706	3709	3759	3840	3848	3951	3953
	3957	3958	3974	4269	4324	4422	4530	4534	5630	5773	5825	5828	5830	5831	5835
	5859	5879	5883	5886	5904	5927	6000	6275	6289	6302	6377	6591	6615	6664	6673
	6678	6746	6778	6786	6808	6857	6867	6946	7015	7030	7088	7099	7122	7148	7253
	7261	7263	7269	7270											
TSTB	1920	2000	2008	2187	2385	2795	2801	2890	2896	2992	2998	3126	3216	3337	3505
	3636	3786	4139	4146	4232	4304	4397	4491	4590	4923	5062	5131	5142	5661	5677
	5894	5988	6496	6535	6602	6770	6810	6859	6873	6982	6998	7053	7059		
.ASCII	1109	1110	1759												
.ASCIIZ	1111	1744	1748	1751	1756	1762	1903	1927	1936	1955	1979	2111	2136	2282	6071
	6125	6409	6547	6555	6750	7131	7132	7133	7135	7233	7280	7284	7287	7290	7295
	7298	7301	7304	7308	7312	7316	7322	7328	7334	7341	7347	7353	7362	7370	7376
	7383	7388	7397	7402	7409	7420	7429	7436	7442	7449	7455	7458	7465	7472	7480
	7487	7490	7501	7510	7517	7525	7533	7541	7548	7554	7559	7566	7571	7575	7583
	7589	7595	7602	7609	7614	7618	7625	7632	7638	7647	7655	7664	7670	7678	7685
	7692	7699	7706	7713	7719	7724	7727	7732	7739	7745	7749	7757	7763	7801	7806
	7811	7815	7821	7823	7830	7834	7838	7844	7849	7857	7870	7883	7888	7894	7901
	7905	7911	7914	7919	7925	7928	7933	7940	7945	7952					
.BLKB	7130														
.BLKW	6890	7964													
.BYTE	1067	1068	1073	1074	1082	1083	1091	1092	1093	1094	2275	6070	6964	6965	6966
	6967	7128	7129												
.DSABL	7040														
.ENABL	879	6973													
.END	7968														
.ENDC	893	910	912	913	914	930	934	936	941	1033	1047	1058	1061	1065	1067
	1095	1107	1108	1109	1113	1843	1850	1851	1854	1856	1858	1860	1861	1863	1865
	1881	1891	1895	1901	1903	1927	1936	1955	1979	2111	2136	2167	2168	2179	2190
	2181	2241	2242	2245	2246	2247	2248	2282	2287	2288	2289	2290	2322	2334	2335
	2336	2337	2362	2371	2372	2373	2374	2387	2391	2392	2394	2395	2396	2400	2409
	2410	2424	2425	2426	2462	2484	2489	2496	2506	2507	2509	2510	2511	2535	2543
	2544	2545	2546	2558	2567	2568	2574	2575	2576	2617	2623	2624	2628	2629	2630
	2680	2686	2687	2694	2695	2696	2735	2736	2748	2749	2750	2841	2847	2848	2866
	2867	2868	2945	2954	2955	2968	2969	2970	3075	3080	3081	3092	3093	3094	3175
	3176	3192	3193	3194	3275	3284	3285	3301	3302	3303	3438	3445	3446	3470	3471
	3472	3473	3570	3571	3587	3588	3589	3590	3708	3724	3729	3730	3750	3751	3752
	3753	3842	3861	3862	3867	3868	3869	3870	3965	3966	3971	3972	3973	3974	3976
	4075	4086	4087	4092	4093	4094	4190	4197	4198	4206	4207	4208	4249	4269	4276
	4277	4280	4281	4282	4351	4362	4363	4369	4370	4371	4445	4450	4451	4461	4462
	4463	4516	4534	4559	4562	4563	4568	4569	4570	4626	4627	4633	4634	4635	4636
	4733	4737	4738	4741	4742	4743	4805	4811	4812	4819	4820	4821	4876	4893	4894
	4899	4900	4901	4970	4971	4976	4977	4978	5035	5036	5042	5043	5044	5107	5108
	5111	5112	5113	5138	5187	5188	5192	5193	5194	5266	5272	5273	5277	5278	5279
	5344	5345	5351	5352	5353	5410	5411	5420	5421	5422	5464	5478	5479	5485	5486
	5487	5544	5545	5551	5552	5553	5612	5613	5620	5621	5622	5623	5634	5692	5707

	5708	5715	5716	5717	5718	5766	5767	5770	5771	5772	5773	5781	5782	5787	5788
	5789	5790	5803	5804	5817	5818	5819	5820	6038	6040	6041	6043	6046	6052	6055
	6056	6060	6062	6068	6070	6071	6074	6125	6409	6547	6555	6571	6574	6579	6584
	6586	6597	6600	6601	6602	6604	6606	6613	6617	6622	6623	6627	6630	6631	6708
	6723	6752	6756	6776	6827	6895	6973	6974	6976	7004	7040	7044	7072	7073	7081
	7083	7086	7114	7131	7137	7141	7147	7150	7162	7163	7164	7165	7166	7167	7169
	7169	7170	7171	7172	7173	7174	7176	7178	7180	7182	7184	7186	7188	7194	7203
	7204	7210	7216	7217	7227	7229	7236								
.EQUIV	941	942	950	965	966	995	996	997	998	999	1000	1001	1002	1003	1004
	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032					
.EVEN	1747	1764	1771	1903	1927	1936	1955	1979	2111	2136	2282	6125	6409	6547	6555
	6751	7235	7774	7961											
.IF	899	909	911	912	913	914	929	932	934	939	1005	1033	1056	1060	1064
	1066	1095	1107	1108	1109	1112	1113	1843	1845	1850	1852	1854	1856	1858	1860
	1861	1863	1881	1890	1891	1892	1895	1902	1926	1935	1954	1978	2110	2135	2166
	2168	2179	2181	2240	2242	2245	2247	2248	2281	2286	2288	2290	2321	2333	2335
	2337	2361	2370	2372	2374	2386	2390	2392	2394	2396	2399	2408	2410	2424	2426
	2461	2483	2488	2495	2505	2507	2509	2511	2534	2542	2544	2546	2557	2566	2568
	2574	2576	2616	2622	2624	2628	2630	2679	2685	2687	2694	2696	2734	2736	2748
	2750	2840	2846	2848	2866	2868	2944	2953	2955	2968	2970	3074	3079	3091	3092
	3094	3174	3176	3192	3194	3274	3283	3285	3301	3303	3437	3444	3446	3470	3472
	3473	3569	3571	3587	3589	3590	3707	3723	3728	3730	3750	3752	3753	3841	3860
	3862	3867	3869	3870	3964	3966	3971	3973	3974	3975	4074	4085	4087	4092	4094
	4189	4196	4198	4206	4208	4248	4268	4275	4277	4280	4282	4350	4361	4363	4369
	4371	4444	4449	4451	4461	4463	4515	4533	4558	4561	4563	4568	4570	4625	4627
	4633	4635	4636	4732	4736	4738	4741	4743	4804	4810	4812	4819	4821	4875	4992
	4894	4899	4901	4969	4971	4976	4978	5034	5036	5042	5044	5106	5108	5111	5113
	5137	5186	5188	5192	5194	5265	5271	5273	5277	5279	5343	5345	5351	5353	5409
	5411	5420	5422	5463	5477	5479	5485	5487	5543	5545	5551	5553	5611	5613	5620
	5622	5623	5633	5691	5706	5708	5715	5717	5718	5765	5767	5770	5772	5773	5780
	5782	5787	5789	5790	5802	5804	5817	5819	5820	6037	6038	6039	6040	6041	6042
	6043	6045	6051	6054	6056	6060	6062	6068	6070	6071	6124	6408	6546	6554	6570
	6573	6578	6584	6596	6598	6599	6600	6602	6603	6604	6613	6615	6623	6624	6629
	6630	6631	6707	6722	6738	6755	6776	6826	6894	6972	6974	6975	6976	7004	7043
	7044	7072	7080	7082	7086	7087	7130	7131	7137	7140	7146	7150	7154	7163	7164
	7165	7166	7167	7168	7170	7171	7172	7173	7174	7176	7178	7180	7182	7184	7186
	7188	7193	7203	7204	7209	7216	7217	7225	7227	7229	7233				
.IFF	909	911	913	914	930	934	936	939	1061	1064	1066	1095	1113	1850	1890
	1891	2167	2168	2180	2181	2241	2242	2246	2247	2248	2287	2288	2289	2290	2322
	2334	2335	2336	2337	2362	2371	2372	2373	2374	2387	2391	2392	2395	2396	2400
	2409	2410	2425	2426	2462	2484	2489	2496	2506	2507	2510	2511	2535	2543	2544
	2545	2546	2558	2567	2568	2575	2576	2617	2623	2624	2629	2630	2680	2686	2687
	2695	2696	2735	2736	2749	2750	2841	2847	2848	2867	2868	2945	2954	2955	2969
	2970	3075	3080	3081	3093	3094	3175	3176	3193	3194	3275	3284	3285	3302	3303
	3438	3445	3446	3471	3472	3473	3570	3571	3588	3589	3590	3708	3724	3729	3730
	3751	3752	3753	3842	3861	3862	3868	3869	3965	3966	3972	3973	3976	4075	4096
	4087	4093	4094	4190	4197	4198	4207	4208	4249	4269	4276	4277	4281	4282	4351
	4362	4363	4370	4371	4445	4450	4451	4462	4463	4516	4534	4559	4562	4563	4569
	4570	4626	4627	4634	4635	4733	4737	4738	4742	4743	4805	4811	4812	4820	4821
	4876	4893	4894	4900	4901	4970	4971	4977	4978	5035	5036	5043	5044	5107	5108
	5112	5113	5138	5187	5188	5193	5194	5266	5272	5273	5278	5279	5344	5345	5352
	5353	5410	5411	5421	5422	5464	5478	5479	5486	5487	5544	5545	5552	5553	5612
	5613	5621	5622	5623	5634	5692	5707	5708	5716	5717	5766	5767	5771	5772	5773
	5781	5782	5788	5789	5790	5803	5804	5818	5819	6038	6042	6046	6051	6054	6070
	6571	6597	6600	6601	6604	6630	6708	6723	6752	6756	6827	6895	6973	6976	7044
	7046	7051	7072	7073	7082	7114	7130	7141	7147	7194	7210	7227			

.IFT	1903	1927	1936	1955	1979	2111	2136	2282	6125	6409	6547	6555	6612	7046	7051
.IFTF	1903	1927	1936	1955	1979	2111	2136	2282	6125	6409	6547	6555	6610	6991	7044
	7047														
.IIF	888	893	898	906	907	908	910	913	914	915	1053	1112	1851	1854	1860
	1861	1863	1864	1891	6040	6046	6047	6058	6070	6074	6574	6575	6576	6577	6578
	6579	6583	6611	6612	6627	6630	6631	6720	6745	6823	6973	6994	7122	7131	7137
	7162	7163	7164	7165	7166	7168	7170	7171	7172	7174	7176	7178	7180	7182	7184
	7186	7188													
.IRP	1843	2166	2240	2286	2333	2370	2390	2408	2505	2542	2566	2622	2685	2734	2846
	2953	3079	3174	3283	3444	3569	3728	3960	3964	4085	4196	4275	4361	4449	4561
	4625	4736	4810	4892	4969	5034	5106	5186	5271	5343	5409	5477	5543	5611	5706
	5765	5780	5802	6837	6877	7197	7203	7216	7217						
.LIST	879	913	1047	1053	1095	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106
	1107	1843	1865	1891	1892	1903	1927	1936	1955	1979	2111	2136	2166	2181	2240
	2247	2282	2286	2290	2333	2337	2370	2374	2390	2396	2408	2426	2505	2511	2542
	2546	2566	2576	2622	2630	2685	2696	2734	2750	2846	2868	2953	2970	3079	3094
	3174	3194	3283	3303	3444	3472	3569	3589	3728	3752	3860	3869	3964	3973	4085
	4094	4196	4208	4275	4282	4361	4371	4449	4463	4561	4570	4625	4635	4736	4743
	4810	4821	4892	4901	4969	4978	5034	5044	5106	5113	5186	5194	5271	5279	5343
	5353	5409	5422	5477	5487	5543	5553	5611	5622	5706	5717	5765	5772	5790	5789
	5802	5819	6046	6062	6125	6409	6547	6555	6578	7072	7154	7162	7163	7164	7165
	7166	7167	7168	7169	7170	7171	7172	7173	7174	7175	7176	7177	7178	7179	7180
	7181	7182	7183	7184	7185	7186	7187	7188	7189						
.MACRO	914	1058	1129	2165	2240	2390	2407	2504	2566	2622	2685	2734	2846	2952	3079
	3174	3282	3444	3568	3727	3859	3963	4084	4195	4274	4361	4449	4561	4625	4736
	4810	4892	4969	5034	5106	5186	5271	5343	5409	5476	5542	5611	5705	5765	5780
	5801	7154													
.MCALL	879	1047	1865	1892											
.NLIST	879	913	1047	1053	1095	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106
	1107	1843	1865	1891	1892	1903	1927	1936	1955	1979	2111	2136	2166	2181	2240
	2247	2282	2286	2290	2333	2337	2370	2374	2390	2396	2408	2426	2505	2511	2542
	2546	2566	2576	2622	2630	2685	2696	2734	2750	2846	2868	2953	2970	3079	3094
	3174	3194	3283	3303	3444	3472	3569	3589	3728	3752	3860	3869	3964	3973	4085
	4094	4196	4208	4275	4282	4361	4371	4449	4463	4561	4570	4625	4635	4736	4743
	4810	4821	4892	4901	4969	4978	5034	5044	5106	5113	5186	5194	5271	5279	5343
	5353	5409	5422	5477	5487	5543	5553	5611	5622	5706	5717	5765	5772	5780	5789
	5802	5819	6046	6062	6125	6409	6547	6555	6578	7072	7154	7162	7163	7164	7165
	7166	7167	7168	7169	7170	7171	7172	7173	7174	7175	7176	7177	7178	7179	7180
	7181	7182	7183	7184	7185	7186	7187	7188	7189						
.PAGE	1058	1113	1843												
.REM	1														
.REPT	1053	1097													
.SBTTL	902	927	937	1047	1056	1058	1113	1844	1887	1892	2166	2240	2286	2333	2370
	2390	2408	2505	2542	2566	2622	2685	2734	2846	2953	3079	3174	3283	3444	3569
	3728	3860	3964	4085	4196	4275	4361	4449	4561	4625	4736	4810	4892	4969	5034
	5106	5186	5271	5343	5409	5477	5543	5611	5706	5765	5780	5802	6035	6081	6087
	6095	6113	6135	6193	6213	6214	6247	6269	6283	6323	6350	6388	6417	6440	6482
	6507	6524	6568	6635	6705	6753	6824	6892	6970	7138	7154	7191	7278	7776	7799
.TITLE	888														
.WORD	935	1053	1054	1055	1066	1069	1070	1071	1072	1075	1076	1077	1078	1079	1080
	1081	1084	1085	1086	1095	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106
	6051	6054	6069	6731	6736	6820	6968	7226	7228	7778	7780	7783	7786	7788	7791

N13

MAINDEC-11-DZRKK-C MACY11 27(732) 16-SEP-76 16:00 PAGE 173
DZRKKC.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*DZRKKC.DZRKKC/SOL/CRF/DS:ERFZ/EN:ABS=DZRKKC.P11
RUN-TIME: 68 65 9 SECONDS
RUN-TIME RATIO: 742/143=5.1
CORE USED: 25K (49 PAGES)

Spooler runtime 20 Seconds, 836 KCS, 756 disk reads, 3 disk writes, 169 pages.

Printed on 10/1/01 10:10:01 AM. Page 1 of 1. Date: 10-01-01 10:10:01 AM. For: JPC-0 0070 (103) 000000

[illegible]