

LISTING

096-000347-04

PROGRAM

NOVA 3 MULTI-PROGRAMMING  
RELIABILITY TEST - SHORT

TAPE

095-000347-04

ABSTRACT

THE NOVA MULTIPROGRAMMING RELIABILITY TEST CONSISTS OF A SERIES OF INDIVIDUAL PROCESSOR AND PERIPHERAL TESTS AND A SUPERVISOR PROGRAM - THE DIAGNOSTIC LINKER. THE DIAGNOSTIC LINKER PROGRAM IS DESIGNED TO "LINK" THE VARIETY OF PROCESSOR AND PERIPHERAL TESTS IN SUCH A FASHION THAT THEY MAY BE RUN CONCURRENTLY. THIS TEST IS PROVIDED IN TWO LENGTHS; SHORT, LONG, AND PERIPHERAL.



0001 N3MRT MACRO REV 03.00

15:35:08 04/22/77

```
01
02
03
04
05 :*****
06 :
07 : NAME: N3MORTS.TX          PART NUMBER: 097-000347
08 :
09 :
10 : DESCRIPTION: NOVA 3 MULTI-PROGRAMMING RELIABILITY TEST
11 :                SHORT VERSION TEXT FILE
12 :
13 :
14 : REVISION HISTORY
15 :
16 :     REV.      DATE
17 :
18 :     00        02/28/76
19 :     01        10/08/76
20 :     02        12/31/76
21 :     03        XX/XX/XX
22 :     04        04/22/77
23 :
24 :
25 : COPYRIGHT (C) DATA GENERAL CORPORATION, 1973,74,75,76,77
26 : ALL RIGHTS RESERVED.
27 :*****
28 :
```

10002 N3MRT

```
01 :CONDITIONAL ASSEMBLY FLAGS
02 :FILE FOR NMORTV2 CPU TESTS ONLY
03 000000 CBRDS=0
04 000001 DCUTS = 1
05 000000 SCMTS=0
06 000000 ARITH=0
07 000000 FPTST=0
08 000000 MUDVT=0
09 000001 PGDSK=1
10 000001 NVDSK=1
11 000001 MVDSK=1
12 000001 PZDSK=1
13 000001 MTES=1
14 000001 CATES=1
15 000001 LPTTS=1
16 000001 PXDSK=1
17 000001 NXDSK=1
18 000001 MXDSK=1
19 000001 SZDSK=1
20 000001 CXTES=1
21 000001 MXTES=1
22 000000 IOTST=0
23 000000 .DUSR .MAPD=0
```

10003 N3MRT

```
01 : TYPE NOVA MULTIPROGRAMMING
02 : RELIABILITY TEST
03 :1. ABSTRACT
04 : THE NOVA MULTIPROGRAMMING RELIABILITY TEST
05 : CONSISTS OF A SERIES OF INDIVIDUAL PROCESSOR
06 : AND PERIPHERAL TESTS AND A
07 : SUPERVISOR PROGRAM. (THE DIAGNOSTIC LINKER)
08 :
09 : THE DIAGNOSTIC LINKER IS A PROGRAM
10 : DESIGNED TO "LINK" THE VARIETY OF
11 : PROCESSOR AND PERIPHERAL TESTS IN
12 : SUCH A FASHION THAT THEY MAY BE
13 : RUN CONCURRENTLY. THEREBY, TESTING
14 : THE INTERACTIVE CAPABILITIES OF
15 : THE PROCESSOR AND ITS PERIPHERAL
16 : EQUIPMENT.
17 : THIS TEST IS PROVIDED IN THREE LENGTHS
18 :
19 : THE SHORT VERSION ONLY INCLUDES THOSE TESTS
20 : THAT APPLY TO THE CPU, MEMORY, FLOATING POINT,
21 : MUL\DIV,TTY, REAL TIME CLOCK, AND THE IOTESTER.
22 :
23 : THE LONG VERSION INCLUDES THE ABOVE + PRIMARY DEVICE
24 : CODE TESTS FOR THE NOVA DISK, MOVING HEAD
25 : DISK, 6060/61 DISK,6063/64 DISK,MAGNETIC TAPE,CASSETTE,
26 : DCU=50 ,AND THE LINE PRINTER
27 :
28 : THE PERIPHERAL VERSION INCLUDES ONLY THE
29 : CHECKERBOARD TEST AS A BACKGROUND TEST BUT
30 : DOES INCLUDE THE PRIMARY AND SECONDARY (WITH
31 : LPT AS AN EXCEPTION) DEVICE CODE TESTS FOR
32 : THE DEVICES MENTIONED ABOVE.
33 :
34 :2. MACHINE REQUIREMENTS
35 :2.1 NOVA 3 PROCESSOR
36 :2.2 16K TO 128K OF READ WRITE MEMORY
37 : ( MEMORY MUST BE CONTIGUOUS)
38 :2.3 OPTIONAL EQUIPMENT
39 :2.3.0 MAP OPTION
40 :2.3.1 FIXED HEAD DISK
41 :2.3.2 MOVING HEAD DISK (ANY/ALL DRIVES)
42 :2.3.3 MAGNETIC TAPE (ANY/ALL DRIVES)
43 :2.3.4 REAL TIME CLOCK
44 :2.3.5 FLOATING POINT UNIT
45 :2.3.6 CASSETTE (ANY/ALL DRIVES)
46 :2.3.7 LINE PRINTER
47 :2.3.8 DCU FEATURE
48 :2.3.9 PARITY OPTION
49 :2.3.10 6060/61 DISK
50 :2.3.11 6063/64 DISK
51 :2.3.12 I/O TESTER
52 :
53 :2.4 PREREQUISITES
54 :2.4.1 SOFTWARE PREREQUISITES
55 : THE SYSTEM SHOULD BE CAPABLE
56 : OF RUNNING ALL INDIVIDUAL LOGIC AND
57 : RELIABILITY TESTS PERTAINING TO THE
58 : PROCESSOR AND ITS PERIPHERAL EQUIPMENT
59 : BEFORE ATTEMPTING TO RUN THIS TEST
60 :NOTE: ALTHOUGH THIS TEST MAY AT TIMES BE USEFUL
```

0004 N3MRT

```
01 :IN DETERMING THE GO/NO GO STATUS OF AN
02 :UNKNOWN SYSTEM, IT IS RECOMMENDED THAT:
03 :A. ALL OTHER DIAGNOSTICS BE RUN EVEN IN THE
04 : EVENT THAT THIS TEST FINDS NO PROBLEMS.
05 :B. AN ATTEMPT BE MADE TO ISOLATE ANY PROBLEMS
06 : FOUND BY FIRST UTILIZING THE LOWER
07 : LEVEL TESTS FOR MORE CONSISE ERROR REPORTS.
```

10005 N3MRT

```
01 :2.4.2 SYSTEM SETUP
02 : IF THE MOVING HEAD DISKS ARE TO BE
03 : EXERCISED THEY MUST HAVE A PACK INSTALLED
04 : AND BE IN THE READY STATE
05 : IF MAGNETIC TAPES ARE TO BE EXERCISED
06 : THEY MUST BE ON LINE WRITE ENABLED
07 : IF CASSETTES ARE TO BE EXERCISED
08 : THEY MUST BE ON LINE WRITE ENABLED
09 : IF THE LINE PRINTER IS TO BE EXERCISED
10 : IT MUST BE ON LINE AND IN THE READY STATE
11 :
12 :3. KEY ENTERED OPTIONS
13 : ENTRIES TYPED ON TTY SET BITS IN SWREG
14 : FOR USE BY THE PROGRAM.
15 :
16 : KEY 1 =1 DON'T RELEASE AND ALLOW REASSIGNMENT
17 : OF MEMORY AFTER ERROR
18 : KEY 2 =1 DELETE TYPEOUTS
19 : KEY " =1 DELETE MEM ALLOCATION TABLE
20 : FROM TYPEOUTS
21 : KEY # =1 CAUSES THE DELETION OF THE RANDOM
22 : WAIT STATES IN THE TTY AND LPT
23 : TESTS.
24 : KEY 4 =1 WILL CAUSE THE ELAPSED RUN
25 : TIME AND ACCUMULATED ERRORS
26 : TO BE TYPED ON THE TTY.
27 : (NOTE: A RTC MUST EXIST)
28 : KEY 5 =1 DIRECT ALL ERROR AND RUNTIME TYPEOUTS
29 : ALSO TO THE LINE PRINTER.
30 : KEY 6 =1 THE ERROR ROUTINE WILL PAUSE AFTER
31 : EACH PHASE OF AN ERROR TYPEOUT.
32 : TYPE A CR KEY ON DEVICE TTI TO PROCEED.
33 :
34 : EACH KEY ENTRY COMPLEMENTS THE PREVIOUS STATE OF
35 : SWREG BIT EXCEPT CONTROL CHARACTERS
36 : FOLLOWING:
37 :
38 : KEY (C)O ENTER THE ODT EDITOR
39 : (SEE DESCRIPTION AT PARAGRAPH 7.0)
40 : KEY (C)S PRINT THE RUN STATISTICS OF EACH
41 : TEST.
42 : KEY (C)D DEFAULT MODE RESTART. SWREG
43 : SET TO 0.
44 : KEY (C)R RESTART WITHOUT RESETTING SWREG BITS.
45 : KEY M TYPE THE CURRENT CONTENTS OF SWREG.
46 :
47 : WHERE (C) SIGNIFIES A CONTROL KEY.
```

10006 N3MRT

```
01 :4. OPERATING PROCEDURES
02 :4.1 LOAD THE PROGRAM VIA THE BINARY LOADER
03 :4.2 SET SWITCHES TO:
04 : 200 FOR AUTO SIZE AND GO
05 : 202 FOR MANUAL SELECT/DELETE
06 : 204 FOR RESTARTING LAST PROGRAM SELECTED
07 : 206 FOR IGNOREING THE MAP
08 : 210 FOR STARTING ODT BEFORE PROGRAM START
09 :
10 : ****NOTE: TEST MUST BE RUN BEFORE ADDRESS
11 : 204 CAN BE UTILIZED.
12 :
13 :4.3 PRESS START
14 :4.4 PROCESSOR WILL TYPE:
15 : NAME/REV/DATE OF REV
16 : TOTAL #1K'S=XXX(DECIMAL) MAP OR NO MAP
17 : PROGRAM RUN LIST
18 : PROG# DESCRIPTION
19 :4.5 IF START WAS 200 OR 204 THE LIST OF
20 : PROGRAMS TO BE RUN CONCURRENTLY WILL
21 : THEN BE LISTED AND THE TEST SYSTEM
22 : WILL AUTO START
23 :4.6 IF START WAS 202 LINKER WILL
24 : PAUSE AT THE END OF EACH TEST
25 : DESCRIPTION AND WAIT FOR KEYBOARD
26 : INPUT. TYPING IN A SPACE WILL
27 : ENABLE THAT TEST TO BE RUN.
28 : TYPING IN ANY OTHER CHARACTER WILL
29 : DELETE THAT TEST FROM BEING RUN
30 :
31 :4.7 IF START WAS 206 LINKER WILL SIZE MEMORY
32 : WITHOUT UTILIZING OR EVEN LOOKING FOR THE
33 : MAP AND THEN PROCEED AS IN STARTING AT
34 : ADDRESS 202 WITH THE MAP NON-EXISTENT.
```

10007 N3MRT

```
01 :5. ERROR DESCRIPTION
02 :
03 : MOST ERRORS DETECTED BY EITHER
04 : THE INDIVIDUAL TEST PROGRAMS OR
05 : BY THE DIAGNOSTIC LINKER WILL
06 : RESULT IN AN EXTENSIVE ERROR
07 : TYPEOUT. SOME SMALL NUMBER OF
08 : HIGHLY IMPROBABLE ERRORS MAY RESULT
09 : IN A PROGRAM HALT IF THEY ARE
10 : OF A NATURE THAT THE LINKER CANNOT
11 : RECOVER FROM AND LOGICALLY PROCEED,
12 : (I.E. INTERRUPT STACK OVERFLOWS)
13 :
14 :5.1 ERROR FORMAT
15 : ERROR TYPEOUTS INCLUDE
16 :5.1.1 PROGRAM # AT TIME OF ERROR
17 : (SEE PROGRAM RUN LIST TO CORRELATE)
18 :5.1.2 THE CURRENT CONTENTS OF AC0, AC1, AC2.
19 :5.1.3 LOGICAL SCRATCH AND DATA CHANNEL LIMITS
20 :5.1.4 MEMORY ALLOCATION TABLE
21 : PHYSICAL 1K PAGE# + LOGICAL ADDRESS +RELOCATED ADDRESS
22 :5.1.5 CONTINUATION INFORMATION IN GROUPS
23 : OF 3 MEMORY LOCATIONS PERTINENT TO
24 : THE INDIVIDUAL TEST THAT FAILED
25 :
26 :5.1.6 THE CPU TESTS THAT RELOCATE/REMAP WILL
27 : IN THEIR ERROR TYPEOUTS:
28 :ST.LA START/ERROR (RES.)
29 :XXXXXX YYYYYY ZZZZZZ
30 :
31 :ST.LA THE LOGICAL START OF THE RELOCATED TEST LOOP
32 :XXXXXX (I.E. THE LAST LCALL SETUL)
33 :
34 :START THIS NUMBER INDICATES WHERE THE RESIDENT COPY
35 : YYYYYY OF THE TEST LOOP MAY BE FOUND IN THE LISTING
36 :
37 :ERROR THIS NUMBER INDICATES WHERE IN THE RESIDENT
38 : ZZZZZZ COPY OF THE LISTING THE ERROR CALL MAY BE FOUND
39 : (FOR SOME VALIDITY TRAP ERRORS THIS NUMBER
40 : MAY NOT APPEAR TO BE VALID.)
```

10008 N3MRT

```
01 :5.2 ERROR ANALYSIS
02 :
03 : DUE TO THE INTERACTIVE NATURE OF
04 : THE TESTS INVOLVED, A SERIES OF
05 : ERROR TYPEOUTS WILL PROBABLY BE
06 : REQUIRED FOR ANALYSIS BEFORE A
07 : PROBLEM WILL BE ISOLATED.
08 : A RESTART AT 202 AND DELETION OF ALL
09 : BUT THE TEST THAT ORIGINALLY
10 : FAILED MAY HELP TO ISOLATE
11 : INTERACTIVE PROBLEMS AS FOLLOWS:
12 :
13 :5.2.1 IF THE TEST RUNS BY ITSELF THE PROBLEM
14 : IS INTERACTIVE-RE-ENABLE ONE OTHER TEST AT
15 : A TIME TO DETERMINE WHICH ONE IS THE PROBLEM.
16 : IF THE TEST DOES NOT RUN BY ITSELF
17 : RESORT TO SIMILAR BUT LOWER LEVEL TESTS
18 : FOR ISOLATION
19 :5.3 PERTINENT MEMORY LOC'S TYPED
20 :
21 :5.3.1 CHECKERBOARD RAN
22 : THE AC'S AT ERROR WILL INDICATE:
23 : GOOD DATA- BAD DATA-LOGICAL ADDRESS
24 :
25 : IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
26 : CB.TK TEST COUNTER
27 : 0 GENERATE CHECKERBOARD
28 : 1 DISTURB PASS
29 : 2 CHECK PATTERN
30 : 3 CHECKSUM THE # OF -1'S IN PATTERN
31 : CB.LC STARTING LOGICAL ADDRESS OF "BEGIN"
32 : RELOCATED TO SCRATCH
33 : CB.SE AC3 AT ERROR CALL
34 :
35 :5.3.2 SC MEMORY TEST
36 : THIS IS AN ISZ/DSZ TEST FOR SC-MEMORIES.
37 :
38 : THE AC'S AT ERROR WILL INDICATE:
39 : ACTUAL-EXPECTED-LOGICAL ADDRESS
40 :
41 : IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
42 : MM.TK ERROR NUMBER:
43 : 0 PATTERN STORING ERROR(SHD BE -1)
44 : 1 LOCATION NOT -1 BEFORE DOING ISZ
45 : 2 ISZ DIDN'T SKIP
46 : 3 LOCATION NOT EQUAL TO 0 AFTER ISZ
47 : 4 DSZ SKIP ERROR
48 : 5 DSZ TEST-LOCATION NOT -1 AFTER DSZ
49 : 6 SAME AS 1, EXCEPT TESTING IN REV DIRECTION
50 : 7 SAME AS 2, EXCEPT " " " " " "
51 : 10 SAME AS 3, EXCEPT " " " " " "
52 : MM.SE INSTRUCTION ADDRESS FOLLOWING ERROR CALL
53 : LOCATION ADDRESS OF FAILING LOCATION(LOGICAL)
54 :
55 :
```

10009 N3MRT

01  
02 :5.3.3 ARITHMETIC TEST  
03 : THE AC'S WILL BE TYPED AS THEY WERE AT THE  
04 : TIME OF ERROR DETECTION  
05 :  
06 : IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:  
07 : AT.LC STARTING ADDRESS OF ARITH IN SCRATCH  
08 : AT.LO LOW LIMIT OF SCRATCH AREA AFTER IT IS  
09 : REMAPPED FOR EXECUTION  
10 : AT.LA AT.LC IN RELATION TO AT.LO  
11 : (LOGICAL START OF ARITH AFTER REMAPPING).  
12 : THE LAST THREE RANDOM NUMBERS GENERATED  
13 : (SEE DISCUSSION OF ST.LA,ETC AT PARA.5.1.6)  
14 :  
15 :5.3.4 FLT PT TEST  
16 :  
17 : AC0 GOOD DATA  
18 : AC1 BAD DATA  
19 : AC2 ADDRESS OF GOOD DATA DURING TEST EXECUTION  
20 :  
21 : IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:  
22 :  
23 : FP.LC START OF LOCATIONS INCLUDING CURRENT  
24 : TEST THAT IS IN SCRATCH(SEE FP.EN)  
25 : FP.LO SCRLO AFTER REMAPPING FOR EXECUTION  
26 : FP.EN END OF TEST OR START OF RANDOM DATA  
27 : IN SCRATCH AREA  
28 : PFS03 AC3 AT TIME OF ERROR CALL (ADDR OF JSR )  
29 : FP.LP LOGICAL PAGE SCRATCH IS REMAPPED TO  
30 : FOR EXECUTION  
31 : FP.ES =-1 FIRST ERROR =0 FOR ALL SUCCEEDING ERRORS  
32 : FP.HI SCRHI AFTER TEST IS REMAPPED FOR EXEC.  
33 : FP.BG START ADDRESS OF TEST THAT CORRELATES TO LISTING  
34 : FP.GA START ADDRESS OF TEST AFTER REMAPPING FOR EXEC.  
35 : L ADRS ADDRESS OF DATA IN AC2 AFTER REMAPPING  
36 : DATA THE DATA IN THAT LOCATION  
37 : A ADRS L ADRS IN RELATION TO SCRLO BEFORE REMAPPING  
38 : OR IS AT AC2 + MH.SA IF IT IS =0 OR 1  
39 :  
40 :  
41 :5.3.5 MUL/DIV TEST  
42 :  
43 :  
44 : MULTIPLY DIVIDE FAILURES WILL INDICATE  
45 : EITHER MUL FOR MULTIPLY OR DIV FOR DIVIDE  
46 : IN ADDITION, THREE SETS OF AC'S ARE TYPED  
47 : ORIGINAL OPERANDS  
48 : HARDWARE RESULT (ASSUMED TO BE INCORRECT )  
49 : SOFTWARE RESULT (ASSUMED TO BE CORRECT )  
50 :  
51 :

10010 N3MRT

01 :5.3.13 IOTESTER TEST  
02 :  
03 : THE AC'S WILL BE TYPED AS THEY WERE AT THE TIME  
04 : OF ERROR DETECTION.  
05 :  
06 : IN ADDITION THE FOLLOWING LOCATIONS ARE  
07 : TYPED:  
08 : IO.OC THE COMMAND BEING PERFORMED BY THE I/OTESTER  
09 : BIT 1=0 FOR WRITE,=1 FOR READ  
10 : IO.BK THE NUMBER OF 64 WORD BLOCKS READ/Written  
11 : B#ERR THE BLOCK NUMBER IN ERROR  
12 : ERRDST THE LOGICAL ADDR. OF LOCATION CONTAINING  
13 : THE BAD DATA  
14 : IOBST THE BEGINNING DATA LOCATION IN SCRATCH  
15 : IOBST THE BEGINNING DATA CHANNEL ADDRESS  
16 :  
17 :  
18 :

10011 N3MRT

```
01 :5.4 SPECIAL CASE ERROR TYPEOUTS
02 :
03 :5.4.1 POWER FAIL INTERRUPT
04 : UPON DETECTION OF A POWER FAIL INTERRUPT
05 : THE LOGICAL ADRS. OF THE P.C. AT INTERRUPT
06 : WILL BE SAVED.
07 : IF AUTO-RESTART IS ENABLED OR THE POWER
08 : FAIL WAS ONLY MOMENTARY, THE TEST WILL RE-
09 : START AS IN A START AT 204 AFTER TYPING
10 : POWER FAIL @XXXXXX (WHERE XXXXXX IS THE PC AT INTR.)
11 :
12 :5.4.2 ILLEGAL SUPERVISOR CALL
13 :
14 : UPON DETECTION OF A SUPERVISOR CALL
15 : WHICH DIDN'T MATCH THE LIST OF SUBROUTINES
16 : CALLS THE FOLLOWING MESSAGE WILL BE TYPED:
17 :
18 : ILLEGAL SUPER CALL AT XXXXXX
19 :
20 : PROG# NNN
21 :
22 : AC'S QQQQQQ YYYYYY ZZZZZ
23 :
24 : TTTTTT WWWW SSSSS
25 :
26 : WHERE XXXXXX IS THE LOGICAL ADDRESS OF THE
27 : SUPER CALL, TTTTTT IS AC3 CONTENTS
28 : AND WWWW IS THE PHYSICAL PAGE #,SSSSS
29 : IS THE INSTRUCTION CAUSING THE SUPER-
30 : CALL.
31 :
32 : NOTE: IF THE ADDRESS TYPED IN THE ILLEGAL SUPERCALL
33 : WAS 000000 THEN THE PROGRAM WAS EXECUTING
34 : LOCATION 0.
```

10012 N3MRT

```
01 :5.4.3 I/O OR VALIDITY TRAP
02 : DEFER OR WRITE CHECK TRAP
03 : AN I/O,WRITE,DEFER OR VALIDITY TRAP
04 : OCCURED THAT WAS NOT FORCED BY ANY TEST
05 :
06 : THE AC'S TYPED AFTER THE PROGRAM #
07 : ARE ASSOCIATED WITH THE FOLLOWING:
08 : AC0: ADDRESS OF INSTR TRAPPED
09 : AC1: VIOLATION DATA REGISTER CONTENTS
10 : AC2: MAP STATUS BITS
11 :
12 : STATUS BITS:
13 : 0: PROGRAM MAP ENABLE
14 : 1: DCH MAP ENABLE
15 : 2: PROGRAM MAP INHIBIT
16 : 9: SINGLE CYCLE WRITE PROTECT
17 : 10: SINGLE CYCLE MAP SELECT A/B
18 : 11: AUTOINDEX PROTECT
19 : 12: DEFER PROTECT
20 : 13: I/O PROTECT
21 : 14: WRITE PROTECT
22 : 15: PROGRAM MAP SELECT A/B
23 :
24 :5.4.4 INTERRUPT WAIT ELAPSED
25 : THE PERIPHERAL DEVICE ASSOCIATED WITH THE
26 : PROG. NUMBER TYPED HAS NOT RESPONDED WITH
27 : A PROGRAM INTERRUPT FOR AN EXTENDED
28 : PERIOD OF TIME. THE 2ND NUMBER TYPED
29 : SHOULD POINT AT THE INTERRUPT HANDLER
30 : FOR THE DEVICE THAT FAILED
```



10013 N3MRT

```
01 :5.4.5 PARITY ERROR INTERRUPT
02 :
03 : IN CASE OF A PARITY ERROR THIS TEST WILL
04 : HALT AFTER LOADING THE FOLLOWING AC'S
05 : WITH THE FOLLOWING INFORMATION:
06 :
07 : AC0 ADDRESS(LOGICAL) WHERE INTERRUPT OCCURED
08 : AC1 BIT 0 IS THE PARITY BIT
09 : BITS 1-15 ARE THE MEM ADDR BITS 1-15
10 : AC2 EXTENDED MEM ADDRESS BITS 0-2
11 :
12 :
13 :5.4.6 DATA CHANNEL PROTECTION ERROR
14 :
15 : IF THE DATA CHANNEL PROTECTION FLAG IS
16 : EVER FOUND TO BE SET(BUSY FLAG-DEV. #2)
17 : THE FOLLOWING DATA WILL BE TYPED:
18 :
19 : DCH VIOLATION ERROR
20 : PROG# ERRORS
21 : XX YY
22 :
23 : WHERE,
24 : XX IS THE PROGRAM NUMBER EXECUTING WHEN
25 : THE ERROR WAS DETECTED.
26 : YY IS THE NUMBER OF DCH ERRS DETECTED SINCE
27 : LAST REPORTED.
28 :
```

10014 N3MRT

```
01 :#6.0 DIAGNOSTIC LINKER
02 :
03 :#6.1 PROGRAM INITIALIZE
04 :#THE DIAGNOSTIC LINKER INITIALIZES ITSELF
05 :#AND INDIVIDUAL TESTS IN THE FOLLOWING
06 :#SEQUENCE:
07 : 1. SYSTEM IS RESET, MAP OPTION IS
08 : DETERMINED TO EXIST OR NOT EXIST
09 : AND SWITCHES ARE SET UP
10 : ACCORDINGLY
11 : 2. ANY OTHER NECESSARY CONSTANTS
12 : ARE INITIALIZED
13 : (MEM ALLOCATION TABLES)
14 : 3. INTERRUPT VECTOR TABLES ARE SET UP TO
15 : PROCESS UNEXPECTED DEVICE INTERRUPTS
16 : 4. MEMORY IS SIZED
17 : FROM 0 TO 128K AND BUILDS AN 8 WORD
18 : BIT MAP OF EXISTING CONTIGUOUS
19 : MEMORY
20 :
21 :
22 : 5. THE EXIST MAP IS MOVED TO THE
23 : AVAILABLE MAP AND EACH BIT
24 : CORRESPONDING TO 1K OF UTILIZED
25 : MEMORY IS REMOVED FROM THE MAP
26 : SO THAT IT WILL NOT BE ASSIGNED
27 : AS A SCRATCH AREA TO ANY TEST.
28 : (INCLUDES PROGRAM STORAGE, MEMORY ALLOC.
29 : TABLES, INTERRUPT MASKS AND STACK AREA AND
30 : THE LAST 1K OF MEMORY TO PRESERVE THE
31 : BINARY LOADER)
32 : 6. EACH TEST IS ENTERED IN SEQUENCE AT ITS
33 : INIT. ENTRY POINT. OPTION TESTS DETERMINE
34 : IF THE DEVICE THEY ARE ASSOC. WITH EXISTS
35 : OR NOT AND PASS INTERRUPT SERVICE PARAM'S
36 : TO THE LINKER.
37 : (DEV#, MASK AND INTERRUPT SERVICE
38 : ADDRESS)
39 : 7. LINKER THEN TYPES THE SYSTEM SIZE
40 : INFORMATION ALONG WITH THE PROGRAM
41 : RUN LIST AND WILL ALLOW THE OPERATOR
42 : TO SELECT OR DELETE SPECIFIC TESTS
43 : IF START WAS 202 .
44 : 8. AFTER STARTING, THOSE TESTS THAT HAVE
45 : "SIZED" THEIR SUBSYSTEM FOR SPECIFIC
46 : PARAMETERS TYPE AN INDICATION OF THE PARAMETERS
47 : THEY DETERMINED TO EXIST.(SEE THE NOVA DISK.6061/62 DISK,
48 : AND MOVING HEAD DISK TEST DESCRIPTIONS.)
49 :
```

10015 N3MRT

```
01      : 16.2 PROGRAM RUN
02      : ONCE THE LINKER HAS COMPLETED ALL
03      : INITIALIZATION THE FOLLOWING SERIES
04      : OF OPERATIONS IS LOOPED THROUGH
05      :
06      :
07      : 1. LINKER RANDOMLY SELECTS ONE OF
08      : THE INDIVIDUAL TESTS UNTIL IT
09      : FINDS ONE THAT IS NOT WAITING
10      : FOR INTERRUPT (WAIT IS BIT 0 OF
11      : THE THIRD WORD IN TEST=1) AND THAT
12      : THE NEXT RANDOM NUMBER FALLS WITHIN
13      : ITS ENTER LIMITS
14      :
15      : 2. IF THE MAP OPTION EXISTS, ALL LOGICAL PAGES
16      : EXCEPT PAGE 0 ARE ACCESS PROTECTED WITH
17      : THE PHYSICAL AREA OF THE SELECTED TEST
18      : MAPPED TO ITSELF AND ANY ASSIGNED
19      : SCRATCH AREA MAPPED TO START AT 1K
20      : ABOVE THE TEST, MEMORY LOCATIONS SCRLO
21      : AND SCRHI (SCRATCH LOW AND HIGH) ARE
22      : SET TO INDICATE THE LIMITS OF
23      : THE SCRATCH AREA AVAILABLE TO THE TEST.
24      :
25      : 3. DATA CHANNEL LIMITS (DCHLO AND DCHHI)
26      : ARE CALCULATED AND ENTERED
27      :
28      : 4. THE SELECTED TEST IS ENTERED AT
29      : ITS SPECIFIED EXECUTE ENTRY POINT
30      :
```

10016 N3MRT

```
01      : 16.4 INDIVIDUAL TEST DESCRIPTIONS
02      :
03      : 16.4.1 CHECKERBOARD RAN
04      :
05      : THIS MEMORY CHECKER BOARD TEST IS A SUBSET OF OTHER MEMORY
06      : CHECKERBOARDS. A COMPLETE TEST OF AN AVAILABLE SCRATCH
07      : AREA IS COMPRISED OF THE FOLLOWING SEQUENCE:
08      :
09      : CB.TK=0      :REQUEST 1 TO 20K OF SCRATCH, RANDOMLY RE-
10      :             :LOCATE THE EXECUTE PORTION OF CHECKERBOARD
11      :             :INTO SCRATCH AND GENERATE THE CHECKERBOARD
12      :             :PATTERN
13      :
14      : CB.TK=1      :DISTURB PASS-COMPLIMENT A SINGLE BIT IN EACH
15      :             :OF THE FIRST 16 WORDS OF SCRATCH, SHUFFLE THESE
16      :             :WORDS 16 TIMES SUCH THAT THEY END UP IN THEIR
17      :             :ORIGINAL POSITION, RE-COMPLIMENT THE SINGLE
18      :             :BIT IN EACH WORD-PROCEED WITH EACH GROUP OF
19      :             :16 WORDS UNTIL ALL MEMORY HAS BEEN EXERCISED.
20      :
21      : CB.TK=2      :CHECK PASS-COMPARE EACH WORD IN SCRATCH WITH
22      :             :THE PATTERN EXPECTED
23      :
24      : CB.TK=3      :FAST CHECKSUM MEMORY TO ENSURE THAT ALL DATA
25      :             :IS INTACT (RETURNS TO CHECK PASS IF CHECK-
26      :             :SUM DOES NOT AGREE.)
27      :
28      :
29      :
```

10017 N3MRT

```
01 ;6.4.3 SC MEMORY TEST
02 ;
03 ;THIS MEMORY TEST DOES A READ/MODIFY/WRITE TO THE AVAILABLE
04 ;SCRATCH AREA USING AN "ISZ" INSTRUCTION. TEST IS BROKEN INTO THE
05 ;FOLLOWING CHECKS:
06 ;
07 ; MM.TK= 0 WRITE INTO EACH MEMORY LOCATION A MINUS
08 ; ONE STARTING AT SCRLO AND ENDING AT SCRHI
09 ; VERIFYING EACH GOT THERE.
10 ;
11 ; MM.TK= 1 READ A LOCATION BEFORE DOING THE ISZ
12 ; TO VERIFY IT HASN'T BEEN DISTURBED.
13 ;
14 ; MM.TK= 2 ISZ DIDN'T SKIP
15 ;
16 ; MM.TK= 3 LOCATION NOT 0 AFTER ISZ
17 ;
18 ; MM.TK= 4 DSZ SKIPPED-ERROR
19 ;
20 ; MM.TK= 5 DSZ TST- LOCATION NOT -1 AFTER DSZ
21 ;
22 ; MM.TK= 6 SAME AS 1, EXCEPT TESTING IN THE REVERSE
23 ; DIRECTION
24 ;
25 ; MM.TK= 7 SAME AS 2, EXCEPT TESTING IN THE REVERSE
26 ; DIRECTION.
27 ;
28 ; MM.TK= 10 SAME AS 3, EXCEPT TESTING IN THE REVERSE
29 ; DIRECTION.
```

10018 N3MRT

```
01 ;6.4.4 ARITHMETIC TEST
02 ;
03 ;THE MULTIPROGRAMMING RELIABILITY ARITHMETIC TEST WAS
04 ;DERIVED FROM THE STAND ALONE ARITHMETIC TEST. THIS TEST
05 ;REQUIRES 2K OF SCRATCH FOR EXECUTION. THE EXECUTE POR-
06 ;TION OF THE TEST IS RANDOMLY RELOCATED WITHIN AVAILABLE
07 ;SCRATCH. IF THE SYSTEM IS MAPPED, (HAS AN MMU) THE
08 ;SCRATCH AREA IS RANDOMLY REMAPPED TO SOME OTHER LOGICAL AD-
09 ;DRESS FOR EXECUTION. AT THE END OF EACH EXECUTION PASS SCRATCH
10 ;AREA IS RANDOMLY RELEASED OR HELD. IF HELD, THE NEXT TIME
11 ;THE TEST IS ENTERED, THE EXECUTABLE PORTION OF THE TEST WILL
12 ;AGAIN BE RANDOMLY RELOCATED WITHIN SCRATCH FOR EXECUTION.
13 ;
14 ;6.4.5 FLOATING POINT TEST
15 ;
16 ; THE NOVA MULTIPROGRAMMING RELIABILITY TEST
17 ;PERFORMS A LOAD AND STORE SINGLE PRECISION WITH
18 ;COMPARE TEST AND A LOAD AND STORE DOUBLE PRECISION
19 ;WITH COMPARE TEST.
20 ; THIS TEST IS ENTERED IF A DEVICE CODE OF 76 WAS
21 ;FOUND DURING THE INITIALIZATION PORTION.
22 ; A RANDOM SCRATCH AREA OF 1 - 32 K IS ASSIGNED FOR
23 ;FOR EXECUTION. THE EXECUTE PORTION OF THE TEST IS
24 ;RANDOMLY RELOCATED WITHIN THE AVAILABLE SCRATCH AREA.
25 ; IF THE SYSTEM IS MAPPED, (HAS A MPMU) THE SCRATCH
26 ;AREA IS RANDOMLY REMAPPED TO SOME OTHER LOGICAL
27 ;ADDRESS FOR EXECUTION. AT THE END OF EACH EXECUTION
28 ;PASS SCRATCH AREA IS RANDOMLY RELEASED FOR OR HELD.
29 ;IF HELD, THE NEXT TIME THE TEST IS ENTERED, THE
30 ;EXECUTABLE PORTION OF THE TEST WILL AGAIN BE RANDOMLY
31 ;RELOCATED WITHIN SCRATCH FOR EXECUTION.
32 ;
33 ;6.4.6 MUL/DIV TEST
34 ; THE NOVA MULTIPROGRAMMING MULTIPLY/DIVIDE TEST
35 ;PERFORMS A TRIAL INSTRUCTION TO DETERMINE
36 ;IF THE MULTIPLY/DIVIDE OPTION IS INSTALLED.
37 ; THIS TEST WAS DERIVED FROM THE STAND ALONE
38 ; MUL/DIV TEST.
39 ; NO MEMORY REALLOCATING IS DONE IN THIS TEST.
```

10019 N3MRT

```
01 ;6.4.12 REAL TIME CLOCK
02 ;
03 ;THE REAL TIME CLOCK IS RUN AT 1K HERTZ. RUNTIME ALONG
04 ;WITH ACCUMULATED ERROR COUNT ARE PRINTED AT 5 MINUTES
05 ;15 MINUTES, 30 MINUTES AND EVERY 30 MINUTES OF RUNTIME
06 ;THEREAFTER. THIS TYPEOUT ALSO OCCURS AFTER EVERY ERROR
07 ;TYPEOUT OR IF TTY KEY 4 IS TYPED.
08 ;
09 ;6.4.13 TELETYPE TEST
10 ;
11 ;THE TELETYPE TEST PRINTS A SINGLE LINE CONSISTING OF THE
12 ;CHARACTERS SPACE TO Z. THE TEST WILL ALSO ECHO CHARACTERS
13 ;AS TYPED.
14 ;
```

10020 N3MRT

```
01 ;6.4.14 I/O TESTER TEST DESCRIPTION
02 ;
03 ; THE I/O TESTER TEST ASSIGNS 2 -4K OF SCRATCH,
04 ; SETS UP A DATA AND CHANNEL ADDRESS IN THE FIRST 64
05 ; WORDS OF SCRATCH, BUILDS A 64 WORD RANDOM DATA
06 ; TABLE IN LOCAL MEMORY.
07 ; THEN THE NEXT OPERATION OF READ OR WRITE
08 ; IS RANDOMLY SELECTED.
09 ; IF A WRITE OPERATION:
10 ; THE DATA TABLE IS REPEATEDLY MOVED TO SCRATCH UNTIL
11 ; FILLED. I/O TESTER MODE AND DATA REGISTERS ARE INITIALIZED
12 ; AND A RANDOM SPACE BETWEEN DCH CYCLES SELECTED.
13 ; A RANDOM COUNT IS SELECTED AT WHICH TIME THE I/O TESTER
14 ; START PULSE WILL BE GIVEN.
15 ; THE I/O TESTER IS STARTED BY THE RANDOM NUMBER GENERATOR
16 ; WHEN LOCATION IO.WK GOES TO ZERO.
17 ; THE I/O TESTER INTERRUPTS AT THE END OF EACH 64 WORD
18 ; BLOCK. AT THIS TIME THE DATA REGISTER IS READ AND
19 ; SAVED FOR LATER CHECKING. UPON COMPLETION OF ALL BLOCKS THE
20 ; TEST VERIFIES THE WRITE BUFFER IS STILL INTACK AND
21 ; THE XOR RESULTS OF EACH 64 WORD TRANSFER ARE CORRECT.
22 ;
23 ; IF A READ OPERATION, THE FIRST RANDOM DATA WORD IS
24 ; PLACED IN THE I/O TESTER DATA REGISTER. THE ADDRESS AND MODE
25 ; REGISTERS ARE SET UP AS ABOVE, THE START PULSE IS GIVEN AS IN
26 ; THE WRITE CASE. AT EACH INTERRUPT A NEW DATA WORD IS
27 ; PLACED INN THE DATA REGISTER .
28 ; AFTER COMPLETION OF ALL BLOCKS THE TEST CHECKS THE
29 ; RECIEVED DATA IN SCRATCH AGAINST THE DATA TABLE.
```

10021 N3MRT

```
01 :7.0 ODT EDITOR
02 :7.1 REQUESTING THE ODT EDITOR
03 : TO ENTER THE ODT TYPE A CONTROL O ON
04 : THE TTI. THIS CAN BE DONE AT ANY POINT IN THE
05 : PROGRAM.
06 :7.2 RESPONSE
07 : ON ENTERING THE ODT A CARRIGE RETURN, LINE FEED
08 : AND AN @ IS TYPED ON THE TTO.
09 :
10 :7.3 CONVENTIONS AND SYMBOLS IN COMMAND LINES
11 :-----
12 :
13 : CR PRESSING THE RETURN KEY IS REPRESENTED BY CR .
14 :
15 : LF PRESSING THE LINE FEED KEY IS REPRESENTED BY LF .
16 :
17 :? PRESSING AN ILLEGAL KEY CAUSES THE ODT TO RESPONSE WITH
18 : A ?..
19 :
20 :@ ODT IS READY AND AT YOUR SERVICE.
21 :
22 :
23 :7.4 COMMAND STRUCTURE
24 :-----
25 :
26 : AN ODT COMMAND HAS THE GENERAL FORMAT:
27 :
28 : [ARGUMENT] [COMMAND]
29 :
30 : ARGUMENT MAY BE ONE OF THE FOLLOWING:
31 :
32 : ADR AN OCTAL ADDRESS OR AN EXPRESSION OF THE FORM:
33 : X+X+X....
34 : WHERE EACH X IS AN OCTAL INTEGER, SEPARATED
35 : FROM THE FOLLOWING X BY EITHER +(PLUS)
36 : OR -(MINUS). LEADING ZEROS NEED NOT BE TYPED.
37 :
38 : N AN OCTAL INTEGER.
39 :
40 : A COMMAND IS A SINGLE TELETYPE CHARACTER
41 :
42 : CHARACTERS USED TO OPEN/CLOSE LOCATIONS INCLUDE:
43 : "/" "CR" "LF" "@"
44 :
45 : CHARACTERS USED TO ENTER/EXIT ODT INCLUDE:
46 : "@"(CTRL O) "R" "P"
47 :
48 : CHARACTERS USED TO MODIFY CURRENT ARGUMENTS ARE:
49 : "RUBOUT" "+" "-" AND THE INTEGERS 0 TO 7
50 :
51 : THE CHARACTER "=" ALLOWS THE CURRENT ARGUMENT TO BE
52 : EXAMINED WITHOUT OPENING OR CLOSING THE CURRENT LOC.
53 : CHARACTERS USED TO MANIPULATE THE NOVA 3 MAP INCLUDE:
54 : "M" "A" "B" "U" "E" "T"
55 :
```

10022 N3MRT

```
01 :
02 :
03 :7.5 COMMANDS TO OPEN A LOCATION
04 :-----
05 :
06 :
07 : THE MEMORY LOCATION TO BE OPENED IS TYPEDOUT.
08 : ADR/ OPEN THE LOCATION AND PRINT ITS CONTENTS
09 : ./ OPEN THE LOCATION CURRENTLY POINTED BY THE POINTER
10 : AND PRINT ITS CONTENTS.
11 :.+ADR/ ADD ADR TO THE POINTER, OPEN THE LOCATION AND
12 : PRINT ITS CONTENTS.
13 :.-ADR/ SUBTRACT ADR FROM THE POINTER, OPEN THE LOCATION AND
14 : PRINT ADDR CONTENTS.
15 : CLOSE THE OPEN LOCATION WITH OR WITHOUT
16 : MODIFICATION OF ITS CONTENTS.
17 : CLOSE THE OPEN LOCATION WITH OR WITHOUT
18 : MODIFICATION OF ITS CONTENTS AND OPEN THE
19 : SUCCEEDING LOCATION.
20 :/ CLOSE THE OPEN LOCATION WITHOUT MODIFYING
21 : ITS CONTENTS AND OPEN THE CELL POINTED
22 : BY ITS CONTENTS
23 :.+ADR/ CLOSE THE OPEN LOCATION WITHOUT MODIFYING
24 : ITS CONTENTS AND OPEN THE LOCATION POINTED
25 : BY ITS CONTENTS+ADR
26 :.-ADR/ CLOSE THE OPEN LOCATION WITHOUT MODIFYING ITS
27 : CONTENTS AND OPEN THE LOCATION POINTED BY
28 : ITS CONTENTS-ADR.
29 : ^ CLOSE THE CURRENT LOCATION AND OPEN ".-1"
30 :
31 : LF CLOSE THE CURRENT LOCATION AND OPEN ".+1"
32 :
33 :
```

10023 N3MRT

```
01 ;7.6 OTHER COMMANDS
02 ; -----
03 ;
04 ;RUBOUT THE RUBOUT KEY IS USED TO DELETE ERRONEOUSLY TYPED
05 ; DIGITS EACH TIME THE RUBOUT KEY IS PRESSED, THE RIGHT
06 ; MOST DIGIT IS DELETED AND ECHOED ON THE TERMINAL.
07 ; IF THE RUBOUT KEY IS PRESSED RIGHT AFTER OPENING A CELL
08 ; THEN IT ALLOWS THE MODIFICATION OF THE CONTENTS AS IF
09 ; THEY WERE TYPED JUST BEFORE THE KEY WAS PRESSED.
10 ;
11 ; P RESTART THE EXECUTION OF THE PROGRAM AT THE LOCATION
12 ; POINTED BY SAVED AC3 .
13 ;
14 ;ADRR START EXECUTING THE USERS PROGRAM AT LOCATION ADR AFTER
15 ; AN IO RESET.
16 ;
17 ; K KILL THE STRING TYPED SO FAR. ODT RESPONDS WITH A ? AND
18 ; THE OPEN LOCATION IS CLOSED WITHOUT MODIFICATION.
19 ; = PRINT THE CURRENT ARGUMENT (I.E. TYPING "= " WILL
20 ; PRINT THE ADRS OF THE LAST OPENED LOCATION)
21 ;
22 ; M SETS SWITCH SO THAT ALL MEMORY ACCESSES ARE
23 ; MAPPED USING THE LAST USER'S SETUP UNTIL A "U"
24 ; "A","B","R",OR "P" IS INPUTTED.
25 ;
26 ; A SETS SWITCH SO THAT ALL FURTHER MEMORY ACCESSES
27 ; ARE MAPPED USER-A UNTIL A "B","R","U",OR "P" IS
28 ; ENTERED.
29 ;
30 ; B SETS SWITCH SO THAT ALL FURTHER MEMORY ACCESSES
31 ; ARE MAPPED USING USER-B UNTIL A "A","R","U",OR "P"
32 ; IS ENTERED.
33 ;
34 ; T PRINTS THE CURRENT CONTENTS OF ALL MAP ENTRY TABLES
35 ;
36 ;NE PRINTS THE MAP ENTRIES CORRESPONDING TO THE LOGICAL ADDR. "N" |
37 ; TO GET SUCCESSIVE ENTRIES TYPE AN E WITH OUT
38 ; PRECEDING IT WITH A NEW "N" VALUE.
39 ; U SETS SWITCH TO KILL MAPPING OF MEMORY ACCESSES.
```

10024 N3MRT

```
01 ;
02 ;7.7 MODIFICATION OF A LOCATION
03 ; -----
04 ;
05 ; ONCE A LOCATION HAS BEEN OPENED ITS CONTENTS CAN BE
06 ; MODIFIED IN ONE OF THE FOLLOWING WAYS:
07 ;
08 ; 1) TYPE THE OCTAL NUMBER OR A STRING OF NUMBERS SEPERATED
09 ; BY + OR -, FOLLOWED BY CR , OR LF . IN THIS CASE THE SUM
10 ; OF THE TOTAL NUMBERS TYPED-IN WILL BE DEPOSITED. LEADING
11 ; ZEROS NEED NOT BE TYPED.
12 ;
13 ; 2) TYPE + OR - FOLLOWED BY A NUMBER OR A STRING OF NUMBERS
14 ; SEPERATED BY + OR -, FOLLOWED BY CR , OR LF . IN THIS
15 ; CASE SUM OF THE TOTAL NUMBERS TYPED IN WILL BE ADDED TO
16 ; OR SUBTRACTED FROM THE PREVIOUS CONTENTS OF THE LOCATION.
17 ; LEADING ZEROS NEED NOT BE TYPED.
18 ;
19 ; 3) ADDRESS ITSELF OR AN OCTAL NUMBER RELATIVE TO THE
20 ; ADDRESS OF THE LOCATION CAN BE DEPOSITED IN A MEMORY
21 ; LOCATION BY TYPING A . OR .+ADR FOLLOWED BY A CR , OR LF .
22 ; 4) A RUBOUT COMMAND GIVEN RIGHT AFTER OPENING A
23 ; LOCATION ALLOWS THE MODIFICATION OF ITS CONTENTS
24 ; AS IF THEY WERE TYPEDIN JUST BEFORE THE COMMAND
25 ; WAS ISSUED.
26 ;
```

10025 N3MRT

```
01
02
03
04
05 *****
06 ;
07 ; NAME: N3MORTS.SR          PART NUMBER: 094-000751
08 ;
09 ;
10 ; DESCRIPTION: NOVA 3 MULTI-PROGRAMMING RELIABILITY TEST
11 ;                   SHORT VERSION
12 ;
13 ; REVISION HISTORY
14 ;
15 ;
16 ;     REV.      DATE
17 ;
18 ;     00      02/28/76
19 ;     01      10/08/76
20 ;     02      12/31/76
21 ;     03      XX/XX/XX
22 ;     04      04/22/77
23 ;
24 ;
25 ; COPYRIGHT (C) DATA GENERAL CORPORATION, 1973,74,75,76,77
26 ; ALL RIGHTS RESERVED.
27 ; *****
28 ;
```

10026 N3MRT

```
01
02 ; .TITL LINKR
03 ;DIAGNOSTIC PROGRAM LINKER
04         000000 .LOC 0
05 00000 015464 DIRT
06 ;CONCURRENTLY LINK A VARIETY OF PROC.
07 ;AND I/O TESTS VIA RANDOM SELECTION
08 ;INTERNAL DEFS TO LINK FOR DEBUG
09         000200 .LOC 200
10 00200 002201 STRT1: JMP @.+1 ;SIZE AND GO
11 00201 001010 LINKR
12 00202 002203 STRT2: JMP @.+1 ;SIZE AND WAIT FOR SELECTIONS
13 00203 001011 LINKR+1
14 00204 002205 STRT3: JMP @.+1 ;RESTART LAST TESTS SELECTED
15 00205 001317 GSTRT
16 00206 002207 STRT4: JMP @.+1 ;IGNORE MAP START
17 00207 001035 IGMAP
18 00210 006212 STRODT: JSR @.+2 ;START ODT DIRECTLY
19 00211 000202 JMP STRT2
20 00212 004206 ODT
21 ;STUFF FOR ODT
22 00213 000000 SAV0: 0
23 00214 000000 SAV1: 0
24 00215 000000 SAV2: 0
25 00216 000000 SAV3: 0
26 00217 000000 SAVCR: 0
27 00220 000000 DP.EN: 0
28 00221 000000 LOPNL: 0
29 00222 000000 STFLG: 0 ;-1 IF PRINT STATISTICS
30 ;DTOS EGGS BLOCK
31 00223 000000 .BDEGG: 0 ;DTOS AUTO SW
32 00224 000000 0 ;
33 00225 000000 0 ;CAT/KITTEN SWITCH
34 00226 000000 0 ;PASS CNT
35 00227 000000 0 ;DTOS START ADDR
36 00230 000000 SWREG: 0 ;SWITCH REGISTER
37
38 ;PROTECTION STUFF
39 00231 000000 PRICK: 0 ;NON ZERO IF PROTECTION OPTION EXISTS
40 00232 000000 DCHER: 0 ;# OF DCH ERRS
41 00233 000000 VLPCR: 0 ;VIOL. PC REGISTER
42 00234 000000 VLDTA: 0 ;SAVED VIOL. DATA REG.
43 00235 005710 IOVL: IOVAL ;I/O VALIDITY TRAP HANDLER
44 00236 006016 IDWCK: DWCHK ;DEFER/WRITE TRAP HANDLER
45 00237 000000 RELLO: 0
46 00240 000000 RELHI: 0
47
48         000045 .LOC 45
49 00045 000223 .BDEGG
50         000060 .LOC 60
51 00060 000000 RNSEL: 0 ;-1 IF DELETE RANDOM WAIT FOR
52 ;FOR TTY/LPT TESTS
53 00061 000000 DLTBL: 0 ;-1 IF DON'T PRINT ALLOC. TBL
54 00062 000000 USESW: 0
```

10027 N3MRT

01 :CALL HANDLER SUBROUTINE LINK  
02 00063 002740 ICDIS: CDISP :CALL DISPATCH  
03  
04 :PAGE 0 LINKS FOR CALLS  
05 :ASCRA MUST BE FIRST WITH RDMAP LAST  
06 :ANY CALL EXPANSION MUST BE MADE BETWEEN THE TWO  
07 00064 003231 ASCRA: ASSCR :ASSIGN SCRATCH  
08 00065 003250 ESCRA: EXSCR :EXPAND SCRATCH  
09 00066 003275 RSCRA: RLSCR :RELEASE SCRATCH  
10 00067 003322 GSCRA: GOSCR  
11 00070 005030 PCRLF: CLF? :CAR. RETURN LINE FEED  
12 00071 004746 PZOCT: ZOC? :ZERO SUPPRESS OCTAL  
13 00072 004753 PDECI: PDE? :DECIMAL PRINT  
14 00073 004047 ERRTX: ERTXT :TEXT TYPEOUT CALL  
15 00074 002140 RETRN: LRETP :RETURN FROM TEST  
16 00075 004121 ERPAD: EPADR  
17 00076 004146 ERPAC: EPACS  
18 00077 002167 ARANG: RANGN :RANDOM # GENERATOR  
19 00100 002321 ADIVI: DIVID :INTEGR UNSIGNED DIV  
20 00101 005564 EINTS: EINTP :ENTER INTR SERVICE  
21 00102 003614 ERROI: ERROH :INIT ERROR TYPEOUTS  
22 00103 004027 ERROC: ERROE :APPEND TO ERR TYPEOUT  
23 00104 003546 SETUL: SETLP :SET UP STRT OF LOOP  
24 00105 003566 LLOOP: LOOPL :LOOP BACK TO SETUP  
25 00106 002244 FRANG: RANG3 :RAN #'S TO AC'S 0,1,AND 2  
26 00107 003430 ERRET: ERRRT :2ND LEVEL ERR RETURN  
27 00110 003454 RETU2: RETN2 :2ND LEVEL NO ERR RETURN  
28 00111 006457 RNQMP: RANDCH :RANDOM SEL DCH MAP A/B

10028 N3MRT

01 00112 006476 ADMAP: AMSCR :ASSIGN SCR TO A DCH  
02 00113 006566 EDMAP: EMSCR :EXPAND DATA CHANNEL ASSIGN  
03 00114 006651 RDMAP: RDSCR :RELEASE DCH MAP  
04 :TEST PARAMETER LOCATIONS  
05 00115 000000 USRMP: 0 :1000 FOR USER B  
06 00116 000000 USRSE: 0 :1 FOR USER B  
07 00117 000000 PRFLG: 0 :=-1 IF PARITY EXISTS  
08 00120 000000 CURPR: 0  
09 00121 000000 ST.LC: 0  
10 00122 000000 ST.LP: 0  
11 00123 000000 ST.LA: 0  
12 :CONTENTS OF LAST 3 RANDOM AC'S FROM RANG3  
13 00124 000000 RNAC0: 0  
14 00125 000000 RNAC1: 0  
15 00126 000000 RNAC2: 0  
16 :LINKS TO TYPEOUT ROUTINES  
17 00127 005030 ICLF?: CLF?  
18 00130 004753 IPDE?: PDE?  
19 00131 004746 IZOC?: ZOC?  
20 00132 004720 LMESS: MES?  
21 000070 LCRLF= PCRLF  
22 00133 004747 LPOCT: POC?  
23 000071 LZOCT= PZOCT  
24 000072 LPDEC= PDECI  
25 : POWER FAIL AUTO RESTART LINKS  
26 00134 001271 PFAIL: PWRUP  
27 00135 000000 PFAIS: 0  
28 :I/O TESTER CONTROL WORDS  
29 00136 000000 IO.OC: 0 :IO TESTER COMMAND  
30 00137 000000 IO.WK: 0 :IO TESTER WAIT COUNT  
31 00140 000000 MPSWT: 0 :MAP EXIST SWITCH  
32 00141 000000 ALTBL: 0 :ADRS MEM ALLOCATION TABLES  
33 00142 000000 PSTRT: 0 :FIRST LOC OF TST  
34 00143 000000 PENOA: 0 :LAST LOC  
35 00144 000000 RTTIM: 0 :ELAPSED RUNTIME IN MINUTES  
36 00145 000000 STATS: 0 :MAP STATUS REG.  
37 00146 177777 UDEVI: -1  
38 00147 000000 TOTPK: 0  
39 00150 177777 TIMSW: -1 :=0 IT'S TIME TO PRINT  
40 00151 000000 ERTOT: 0 :ERROR ACCUMULATOR  
41 00152 000000 EACTV: 0 :1B0=TT0 1B1=LPT  
42 00153 000000 LASTI: 0 :CHAR FROM LAST INPUT FROM TTI  
43 :SCRATCH AREA SIZE PARAMETER LOCS FOR TEST USAGE  
44 00154 000000 SCRLO: 0 :LOWEST LOGICAL SCRATCH ADRS  
45 00155 000000 SCRHI: 0 :HIGHEST LOGICAL SCRATCH ADRS  
46 00156 000000 DCHLO: 0 :LOW LOG DCH ADRS  
47 00157 000000 DCHHI: 0 :HIGH LOG DCH ADRS  
48 000160 LZMAX=DCHHI+1  
49 000160 LPGO=  
50 061401 .DIAC PSH=061401  
51 061601 .DIAC POP=061601  
52 062401 .DUSR SAVE=062401  
53 062601 .DUSR RTRN=062601  
54 060001 .DIAC MTFP=60001  
55 061001 .DIAC MTSP=61001  
56 060201 .DIAC MFFP=60201  
57 061201 .DIAC MFFP=61201  
58 000004 .DUSR PRY = 4  
59 000047 TPADR=47  
60 000046 TPLOC=46



0029 N3MRT

```
01
02
03      001001  ;DEFINITIONS FOR USER STATUS TABLE ACCESS
04      001002      .DUSR USTZM=1001
05      001003      .DUSR USTSS=1002
06      001004      .DUSR USTES=1003
07
08      ;
09      ;LINKER MAIN LINE DISPATCH ROUTINE
10      001010      .LOC 1010
11 01010 102401 LINKR: SUB 0,0,SKP      ;AUTO START ENTRY
12 01011 102000      ADC 0,0      ;MANUAL SELECT ENTRY
13 01012 126400      SUB 1,1
14 01013 044062      STA 1,USES      ;SET USE MAP SW
15 01014 040420      STA 0,LAUTO      ;SET ENTRY TYPE SW
16 01015 102000      ADC 0,0
17 01016 040135      STA 0,PFAIS      ;SET NOT PWR/FA RESTART
18 01017 004405      JSR LOSPR      ;START DISPATCH
19 01020 001040      LTBL1 ;THROUGH INIT SEG
20 01021 060177      INTEN
21 01022 004402      LRUNS: JSR LOSPR
22 01023 001054      LTBL2 ;THROUGH RUN TABLE
23
24      ;DISPATCH ROUTINE
25      ;ENTER SUBROUTINES IN SEQ VIA TABLE SPEC BY (R3)
26      ;END OF EACH TABLE IS LRUNS WHICH WILL START US
27      ;BACK AT THE BEGINNING OF THE RUN TABLE
27 01024 021400      LOSPR: LDA 0,0,3      ;ADRS OF DISPATCH TABLE
28 01025 040406      STA 0,LIDIS
29 01026 036405      LDS.1: LDA 3,@LIDIS
30 01027 005400      JSR 0,3
31 01030 010403      ISZ LIDIS
32 01031 004434      JSR CKODT
33 01032 000774      JMP LDS.1
34 01033 000000      LIDIS: 0
35 01034 000000      LAUTO: 0
36      ;SET SWITCH SO MAP WILL NOT BE USED IF IT EXISTS
37 01035 102000      IGMAP: ADC 0,0
38 01036 040062      STA 0,USES      ;SET MAP SWITCH TO OFF
39 01037 000755      JMP LINKR+4
40      ;DEFINE SYSTEM MACROS FOR INDIVIDUAL TESTS
41      .MACRO LCALL
42      ^1-ASCRA*1B11+100010
43
44      .MACRO NEXTT
45      LMEML=.
46      .LOC LPGO
47      ^1
48      LPGO=.
49      .LOC LMEML
50      0      ;TEST PASS CTR
51      0      ;TEST ERROR CTR
52      0      ;INTERRUPT TIMEOUT SWITCH
53      X
```

10030 N3MRT

```
01
02      ;LTBL1-INIT SYSTEM DISPATCH TABLE
03      ;END OF TABLE IS LRUNS
04 01040 001331      LTBL1: LSYR      ;RESET SYS
05 01041 001470      GPRK      ;DET # TESTS LOADED
06 01042 001411      LWSET      ;SET UP RUNNING CONS.
07 01043 005575      LCINT      ;INIT INTR VECTORS
08 01044 001503      LSIZE      ;SIZE MEMORY
09 01045 001647      MVETA      ;MOVE EXISM TO AVALM
10 01046 001666      UBL32      ;REMOVE USED CORE FROM AVALM
11 01047 001714      TINIT      ;INIT EACH TEST LOADED
12 01050 007013      LPRSL      ;LISTS TESTS TO BE RUN
13 01051 001021      LRUNS-1
14 01052 000000      0
15 01053 000000      0
16
17      ;LTBL2-RUN SYS DISPATCH TABLE
18 01054 001752      LTBL2: LRANP      ;RANDOM PROG SELECT
19 01055 003106      LDMAP      ;LOAD MAP OPTION
20 01056 006747      LDCHL      ;CALC DCHLO/HI
21 01057 002107      LSTRP      ;START TEST RUNNING
22 01060 001150      LSTAT      ;ADJUST RUN STATISTICA
23 01061 002336      CHSTK
24 01062 001022      LRUNS
25 01063 000000      0
26 01064 000000      0
```

```

10031 N3MRT
01          :CKODT - CHECK IF TTI REQUESTS HAVE BEEN INPUTTED
02          : BY THE OPERATOR.
03 01065 040452 CKODT: STA 0,CK.S0
04 01066 044452          STA 1,CK.S1
05 01067 050452          STA 2,CK.S2
06 01070 054452          STA 3,CK.S3
07 01071 020153          LDA 0,LASTI          :GET LAST INPUT FROM TTI
08 01072 101103          MOVL 0,0,SNC          :SKP IS NEW INPUT
09 01073 000437          JMP CKRTN          :NON ENTERED,EXIT
10 01074 101220          MOVZR 0,0
11 01075 030447          LDA 2,J177
12 01076 143400          AND 2,0          :MASK FOR LOWER BITS ONLY
13 01077 040153          STA 0,LASTI
14 01100 006445 CKKEY: JSR @IINP?          :CHECK IF KEY ENTRY
15 01101 024445          LDA 1,J23          :CK FOR CONTROL S
16 01102 136414          SUB# 1,3,SZR
17 01103 000404          JMP .+4
18 01104 126000          ADC 1,1
19 01105 044222          STA 1,STFLG          :SET FLAG TO PRINT STATS.
20 01106 000424          JMP CKRTN          :EXIT
21 01107 024434          LDA 1,J42
22 01110 136414          SUB# 1,3,SZR          :CHECK FOR " KEY ENTRY
23 01111 000405          JMP .+5
24 01112 034061          LDA 3,DLTBL          :COMPLEMENT
25 01113 174000          COM 3,3          :STATE OF
26 01114 054061          STA 3,DLTBL          :ALLOCATION TABLE FLAG
27 01115 000415          JMP CKRTN          :EXIT
28 01116 125400          INC 1,1
29 01117 136414          SUB# 1,3,SZR          :CHECK FOR # KEY ENTRY
30 01120 000405          JMP .+5
31 01121 034060          LDA 3,RNSEL
32 01122 174000          COM 3,3
33 01123 054060          STA 3,RNSEL          :COM TTY/LPT RANDOM SEL FLAG
34 01124 000406          JMP CKRTN
35 01125 024422          LDA 1,CK.64
36 01126 136404          SUB 1,3,SZR          :SKP=KEY 4 TYPED
37 01127 000403          JMP CKRTN
38 01130 102400          SUB 0,0
39 01131 040150          STA 0,TIMSW          :SET SW SO TIME WILL FOLLOW
40 01132 020405 CKRTN: LDA 0,CK.S0
41 01133 024405          LDA 1,CK.S1
42 01134 030405          LDA 2,CK.S2
43 01135 002405          JMP @CK.S3
44 01136 004206 GODT:  ODT
45 01137 000000 CK.S0:  0
46 01140 000000 CK.S1:  0
47 01141 000000 CK.S2:  0
48 01142 000000 CK.S3:  0
49 01143 000042 J42:   42
50 01144 000177 J177:  177
51 01145 005255 IINP?:  INP?
52 01146 000023 J23:   23
53 01147 000064 CK.64:  64

```

```

10032 N3MRT
01          :END OF TEST PASS, SEE IF ANY EXTRANEIOUS ERRORS
02          :NEED TO BE REPORTED
03 01150 054516          LSTAT: STA 3,LST.3
04 01151 024146          LDA 1,UDEVI
05 01152 124015          COM# 1,1,SNR          :SKP =UNEXP. INTA
06 01153 000406          JMP .+6          :NONE REC'D
07 01154 006132          JSR @LMESS
08 01155 007325          UDEVT
09 01156 006071          JSR @LZOCT
10 01157 102000          ADC 0,0
11 01160 040146          STA 0,UDEVI
12          :CHECK FOR ANY DCH ERRS
13 01161 020232          LDA 0,DCHER          :ANY DCH ERRS
14 01162 101005          MOV 0,0,SNR          :SKP=YES
15 01163 000414          JMP PRSTAT
16 01164 006132          JSR @LMESS
17 01165 006230          DCHTX
18 01166 024120          LDA 1,CURPR          :PROG#
19 01167 006071          JSR @LZOCT
20 01170 024232          LDA 1,DCHER
21 01171 006072          JSR @PDECI          :PRINT # OF ERRS
22 01172 102400          SUB 0,0
23 01173 040150          STA 0,TIMSW          :SO TIME WILL FOLLOW
24 01174 040232          STA 0,DCHER
25 01175 010151          ISZ ERTOT
26 01176 000401          JMP .+1
27 01177 030142          PRSTAT: LDA 2,PSTRT          :PTR. TO XX.00
28 01200 011375          ISZ -3,2          :+1 PASS CTR THIS TEST
29 01201 000406          JMP .+6          :NOT OFLOW'D
30 01202 006132          JSR @LMESS          :TYPE 65K HEADER
31 01203 007257          TX65K
32 01204 024120          LDA 1,CURPR
33 01205 006071          JSR @LZOCT          :AND PROG#
34 01206 006070          JSR @LCRLF
35 01207 020222          LDA 0,STFLG          :CHECK IF WANT PRINTOUT
36 01210 101005          MOV 0,0,SNR          :OF STATISTICS
37 01211 000430          JMP LEXIT          :NOBODY'S INTERESTED
38 01212 102400          SUB 0,0
39 01213 040222          STA 0,STFLG          :CLR FLAG
40 01214 040120          STA 0,CURPR          :START WITH 0

```

```

10033 N3MRT
01          :PRINT PASSES AND ERROR COUNTS BY INDIVIDUAL TEST
02 01215 006132      JSR @LMESS
03 01216 007272      STHDR
04 01217 024120      PRSTL: LDA 1,CURPR
05 01220 010120      ISZ CURPR
06 01221 034444      LDA 3,ALZMAX
07 01222 137000      ADD 1,3
08 01223 031400      LDA 2,0,3
09 01224 151005      MOV 2,2,SNR
10 01225 000414      JMP LSXIT
11 01226 021375      LDA 0,-3,2
12 01227 101005      MOV 0,0,SNR
13 01230 000767      JMP PRSTL
14 01231 006071      JSR @LZOCT
15 01232 025375      LDA 1,-3,2
16 01233 006072      JSR @LPDEC
17 01234 025376      LDA 1,-2,2
18 01235 125004      MOV 1,1,SZR
19 01236 006072      JSR @LPDEC
20 01237 006070      JSR @LCRLF
21 01240 000757      JMP PRSTL
22 01241 020223      LSXIT: LDA 0,.BDEGG
23 01242 101005      MOV 0,0,SNR
24 01243 002423      JMP @LST.3
25 01244 010147      ISZ TOTPK
26 01245 034227      LDA 3,.BDEGG+4
27 01246 020151      LDA 0,ERTOT
28 01247 101004      MOV 0,0,SZR
29 01250 000412      JMP DTRET
30 01251 024147      LDA 1,TOTPK
31 01252 030415      LDA 2,LS.NN
32 01253 146404      SUB 2,1,SZR
33 01254 002412      JMP @LST.3
34 01255 044147      STA 1,TOTPK
35 01256 014226      DSZ .BDEGG+3
36 01257 002407      JMP @LST.3
37 01260 062677      IORST
38 01261 045776      STA 1,-2,3
39 01262 062677      DTRET: IORST
40 01263 060203      NIOC 3
41 01264 001400      JMP 0,3
42 01265 000160      ALZMAX: LZMAX
43 01266 000000      LST.3: 0
44 01267 004000      LS.NN: 4000

```

```

10034 N3MRT
01          :POWER FAIL AUTO RESTART/ WAIT FOR POWER UP
02 01270 002134      JMP @PFAIL
03 01271 020000      PWRUP: LDA 0,0
04 01272 024776      LDA 1,PWRUP-1
05 01273 122414      SUB# 1,0,SZR
06 01274 040430      STA 0,PWRS0
07 01275 044000      STA 1,0
08 01276 063777      SKPDZ CPU
09 01277 000777      JMP -1
10 01300 006132      JSR @LMESS
11 01301 006445      PFTX
12 01302 024422      LDA 1,PWRS0
13 01303 006133      JSR @LPOCT
14 01304 006132      JSR @LMESS
15 01305 014304      RTTEX+1
16 01306 024144      LDA 1,RTTIM
17 01307 006072      JSR @LPDEC
18 01310 024151      LDA 1,ERTOT
19 01311 125004      MOV 1,1,SZR
20 01312 006072      JSR @LPDEC
21 01313 102400      SUB 0,0
22 01314 034412      LDA 3,AGSTRT
23 01315 054751      STA 3,LST.3
24 01316 000675      JMP PRSTL-4
25 01317 102400      GSTRT: SUB 0,0
26 01320 040135      STA 0,PFAIS
27 01321 042406      STA 0,@ALAUTO
28 01322 006406      JSR @ALDSPR
29 01323 001040      LTBL1
30 01324 000000      PWRS0: 0
31 01325 000404      JMP LSYSR
32 01326 001317      AGSTRT: GSTRT
33 01327 001034      ALAUTO: LAUTO
34 01330 001024      ALDSPR: LDSPR

```

```

10035 N3MRT
01      ;LSYSR-RESET SYSTEM
02      ;SET LOGICAL PAGE 37=PHYS 37
03      ;SET MPSWT=0 NO MAP =1'S IF MAP OPTION
04 01331 062677 LSYSR: IORST
05 01332 102400     SUB 0,0
06 01333 040000     STA 0,0
07 01334 040140     STA 0,MPSWT
08 01335 040137     STA 0,IO.WK     ;CLEAR TSTR WAIT COUNT
09 01336 040151     STA 0,ERTOT   ;CLR ACCUM ERRS
10 01337 040147     STA 0,TOTPK
11 01340 040152     STA 0,EACTV
12 01341 040153     STA 0,LASTI
13 01342 040232     STA 0,DCHER
14 01343 126000     ADC 1,1
15 01344 044146     STA 1,UDEVI   ;UNEXPECTED INTERRUPT FLAG
16      ;CHECK FOR PARITY OPTION
17 01345 040117     STA 0,PRFLG   ;CLR PARITY EXISTS FLAG
18 01346 060304     NIOP PRTY   ;SETS EVEN PARITY MODE
19 01347 063404     SKPBN PRTY  ;BSY=1 IF PARITY EXISTS
20 01350 000405     JMP .+5      ;NO PARITY EXISTS
21 01351 060104     NIOS PRTY   ;SET ODD MODE
22 01352 060204     NIOC PRTY   ;CLR ANY ERRS
23 01353 060104     NIOS PRTY   ;ENABLE INTR
24 01354 044117     STA 1,PRFLG   ;SET PARITY FLAG
25 01355 024062     LDA 1,USES9W  ;CHECK IF WANT TO USE MAP
26 01356 125004     MOV 1,1,SZR   ;SKP IS OK TO USE MAP
27 01357 001400     JMP 0,3      ;EXIT NO MAPPING
28 01360 060203     NIOC 3       ;DIAG. RESET
29 01361 030427     LDA 2,LS.K2   ;PAGE 37
30 01362 072002     DOB 2,2      ;MAP ENTRY
31 01363 060302     NIOP 2       ;SINGLE CYCLE CMD
32 01364 025000     LDA 1,0,2     ;READ LOC. 0
33 01365 125004     MOV 1,1,SZR   ;SKP IS MAYBE MAP EXISTS
34 01366 001400     JMP 0,3      ;NO MAP-EXIT
35 01367 102000     ADC 0,0       ;=-1
36 01370 040000     STA 0,0       ;SET LOC 0 TO -1
37 01371 060302     NIOP 2       ;SING. CYCLE
38 01372 031000     LDA 2,0,2     ;GET LOC 0
39 01373 044000     STA 1,0       ;RESTORE LOC 0
40 01374 150014     COM# 2,2,SZR  ;SKP IS MAP EXISTS
41 01375 001400     JMP 0,3      ;NO MAP EXIT
42 01376 050140     STA 2,MPSWT   ;MPSWT = -1
43 01377 030402     LDA 2,.,+2    ;CHECK FOR PROTECTION OPTION
44 01400 000402     JMP .+2
45 01401 001402     .+1
46 01402 065403     DIB 1,3       ;READ VIOL. PC REG
47 01403 146404     SUB 2,1,SZR   ;SHD = AC2
48 01404 126000     ADC 1,1
49 01405 120000     COM 1,0       ;=-1 IF PROTECTION EXISTS
50 01406 040231     STA 0,PRTEK   ;SET FLAG
51 01407 001400     JMP 0,3      ;RETURN
52 01410 076000     LS.K2: 76000

```

```

10036 N3MRT
01
02      ;LWSET-SET UP SYSTEM FOR RUNNING
03      ;DCHMAP AND INIT MAP OPTION TRAP LOCATIONS
04 01411 022444 LWSET: LDA 0,@LW.K1 ;GET NMAX
05 01412 040454     STA 0,LSYTB
06 01413 040454     STA 0,LSETB
07 01414 024466     LDA 1,PROGK
08 01415 124400     NEG 1,1
09 01416 044440     STA 1,LW.C1
10 01417 030450     LWS.1: LDA 2,LSETB
11 01420 102400     SUB 0,0
12 01421 041000     STA 0,0,2     ;SET UP MEM ALLOC
13 01422 151400     INC 2,2       ;TABLES FOR 1 PROG
14 01423 041000     STA 0,0,2     ;1ST 2 WRDS =0
15 01424 151400     INC 2,2       ;NXT 32=-1
16 01425 024432     LDA 1,LW.K4
17 01426 100000     COM 0,0
18 01427 041000     LWS.2: STA 0,0,2
19 01430 151400     INC 2,2
20 01431 125404     INC 1,1,SZR   ;STORE -1 32 TIMES
21 01432 000775     JMP LWS.2
22 01433 050434     STA 2,LSETB   ;NEW END SYS TABLES
23 01434 010422     ISZ LW.C1
24 01435 000762     JMP LWS.1
25 01436 042422     STA 0,@LW.K5   ;DO ONE MORE TABLE
26 01437 042422     STA 0,@LW.K5+1 ;DCH ALLOC TABLE
27 01440 042422     STA 0,@LW.K5+2 ;AVAILABLE 32K
28 01441 042422     STA 0,@LW.K5+3 ;DCH B MAP ALLOC. TBL
29 01442 014425     DSZ LSETB     ;REAL END OF SYS TABLES
30 01443 024421     LDA 1,LW.K6
31 01444 030421     LDA 2,LW.K7
32 01445 100000     COM 0,0
33 01446 041000     LWS.3: STA 0,0,2 ;CLEAR CORE EXIST MAP
34 01447 151400     INC 2,2
35 01450 125404     INC 1,1,SZR
36 01451 000775     JMP LWS.3

```

10037 N3MRT

```
01
02 ;NOW SET UP JMP @'S IN MAP TRAPS
03 01452 020063 LDA 0,ICDIS
04 01453 040047 STA 0,TPADR
05 01454 001400 JMP 0,3
06 01455 001004 LW.K1: USTNM
07 01456 000000 LW.C1: 0
08 01457 177740 LW.K4: -32.
09 01460 015457 LW.K5: DCHM0
10 01461 015460 DCHM1
11 01462 015461 DCHM2
12 01463 015462 DCHM3
13 01464 177770 LW.K6: -8.
14 01465 015437 LW.K7: EXISM
15 01466 000000 LSYTB: 0
16 01467 000000 LSETB: 0
17
```

10038 N3MRT

```
01
02 ;GPRGK-GENERATE PROGRAM COUNT
03 ;THE FOLLOWING SUBROUTINE SIMPLY DETERMINES
04 ;HOW MANY TEST PROGRAMS ARE IN CORE
05 ;ALONG WITH THE DIAGNOSTIC LINKER
06 ;ZMAX-LAST LINKER ZLOC=#TESTS INTO PROGK
07 01470 024410 GPRGK: LDA 1,KLZMX ;LAST LINKER ZLOC
08 01471 022410 LDA 0,@ISTZM ;LAST ZPAGE FILLED
09 01472 111000 MOV 0,2
10 01473 122400 SUB 1,0 ;ACO=NUMBER TESTS
11 01474 040406 STA 0,PROGK
12 01475 102400 SUB 0,0
13 01476 041000 STA 0,0,2
14 01477 001400 JMP 0,3
15 01500 000160 KLZMX: LZMAX
16 01501 001001 ISTZM: USTZM
17 01502 000000 PROGK: 0
18
19 ;
20 ;LSIZE-LINK THE 32K AND ABOVE 32K MEM SIZERS
21 ;SET UP UP32L AND HIGHK
22 ;EXISTM=0'S
23 01503 054421 LSIZE: STA 3,LS.S3
24 01504 004431 JSR MSG32 ;SIZE 0 TO 32K
25 01505 040420 STA 0,UP32L ;ACO=LAST ADRS IN 32K
26 01506 024702 LDA 1,LS.K2 ;5 BITS PHYS PAGE 37
27 01507 107400 AND 0,1
28 01510 125300 MOVS 1,1 ;AC1=LAST PHYS PAGE(32K)
29 01511 125220 MOVZR 1,1
30 01512 125220 MOVZR 1,1
31 01513 044413 STA 1,HIGHK ;IN CASE NOT 32K OR NO MAP
32 01514 101533 INCZL# 0,0,SNR ;SKP IS 32K
33 01515 000412 JMP LSIZR ;EXIT MEM <32K
34 01516 024140 LDA 1,MPSWT
35 01517 125005 MOV 1,1,SNR ;=0 IS NO MAP OPTION
36 01520 000407 JMP LSIZR
37 01521 004461 JSR MSG32 ;SIZE ABOVE 32K
38 01522 040404 STA 0,HIGHK ;# OF PHYS PAGES(1K)
39 01523 000404 JMP LSIZR
40 01524 000000 LS.S3: 0
41 01525 000000 UP32L: 0
42 01526 000000 HIGHK: 0
43 01527 024117 LSIZR: LDA 1,PRFLG ;PARITY EXISTS?
44 01530 125005 MOV 1,1,SNR ;SKP IS PARITY EXISTS
45 01531 002773 JMP @LS.S3 ;RETURN
46 01532 060204 NI0C PRTY ;CLR ANY PARITY ERRS
47 01533 060104 NI0S PRTY ;ENABLE PARITY INTR
48 01534 002770 JMP @LS.S3 ;RETURN
```

```

10039 N3MRT
01
02
03      ;MSZ32-MEMORY SIZER 32K
04      ;DOES NOT USE MAP OPTION
05      ;SETS EXIST BIT FOR CONTIGUOUS MEM TO 32K
06      ;WILL NOT SIZE NNN CONTIGUOUS MEM
07      ;ALSO ASSUMES THAT EXISM=0'S
08      ;I.E.-CMAPB SKIP ON RETURN IS NONSENSE
09      ; EXIT ACO=HIGHEST AVAIL ADRS.
10 01535 054440 MSZ32: STA      3,XMS32 ;SAVE
11 01536 030731      LDA      2,LSETB
12 01537 050435      STA      2,M32TE
13 01540 151400      INC       2,2
14 01541 025000      LDA      1,0,2 ;GET CELL
15 01542 120000      COM      1,0 ;CHNG BITS
16 01543 041000      STA      0,0,2
17 01544 021000      LDA      0,0,2 ;=COM MEM EXISTS
18 01545 122405      SUB      1,0,SNR ;AND WE'LL SKIP
19 01546 000424      JMP      M32SZ ;LST WAS NONEXIST
20 01547 050425      STA      2,M32TE
21 01550 045000      STA      1,0,2 ;RESTORE CELL
22 01551 141000      MOV      2,0
23 01552 024427      LDA      1,K1K
24 01553 123400      AND      1,0
25 01554 122404      SUB      1,0,SZR ;SKP IS DONE ALL OF 1K
26 01555 000763      JMP      MSZ32+3
27      ;NOW SET BIT IN EXIST MAP
28 01556 141300      MOVS     2,0
29 01557 101200      MOVR    0,0
30 01560 101200      MOVR    0,0
31 01561 024415      LDA      1,K37
32 01562 123400      AND      1,0 ;ACO=PHYS PAGE #
33 01563 030414      LDA      2,KXIST ;ADRS EXIST TABLE
34 01564 006414      JSR     @MS.L1 ;SET EXIST BIT=1
35 01565 101001      MOV      0,0,SKP
36 01566 063077      HALT    ;*****CAN'T HAPPEN EXIST BIT HAD TO =0
37 01567 030405      LDA      2,M32TE ;AC1=LAST 1K TOP ADDRESS
38 01570 151533      INCZL# 2,2,SNR ;SKP IF LAST CELL =32K
39 01571 000747      JMP      MSZ32+3 ;NOT DONE SIZING
40 01572 020402 M32SZ: LDA      0,M32TE ;ACO=HIGHEST AVAIL.
41 01573 020402      JMP      @XMS32
42 01574 000000 M32TE: 0
43 01575 000000 XMS32: 0
44 01576 000037 K37: 37
45 01577 015437 KXIST: EXISM
46 01600 002627 MS.L1: CMAPB
47 01601 001777 K1K: 1777

```

```

10040 N3MRT
01
02      ;MSG32-MEMORY SIZING TO ABOVE 32K
03      ;MAP OPTION MUST EXIST
04      ;USES SINGLE CYCLE MAPPING TO SIZE
05      ;EXIT IS WITH ACO=LAST PHYSICAL PAGE #
06      ;EXISM(EXIST MAP)MUST BE 0'S ABOVE 32K
07      ;ASSUMES MEMORY TO BE CONTIGUOUS
08 01602 054773 MSG32: STA 3,XMS32
09 01603 020723      LDA 0,HIGHK
10 01604 101400      INC 0,0
11 01605 105300      MOVS 0,1
12 01606 125122      MOVZL 1,1,SZC ;SKP IS NOT SIZED TO 128K
13 01607 000431      JMP MG32S
14 01610 034436      LDA 3,MPF32 ;BITS FOR 32K MAP
15 01611 117000      ADD 0,3 ;+PHYSICAL PAGE #
16 01612 076002      DOB 3,2 ;WRITE MAP ENTRY
17 01613 034433      LDA 3,MPF32 ;ADDRESS TO USE
18 01614 060302 MS.LL: NIOP 2 ;SINGLE CYCLE COMMAND
19 01615 031400      LDA 2,0,3 ;GET CELL
20 01616 144000      COM 2,1 ;COMPLEMENT IT
21 01617 060302      NIOP 2
22 01620 045400      STA 1,0,3 ;STORE COMP.
23 01621 060302      NIOP 2
24 01622 025400      LDA 1,0,3 ;GET IT BACK
25 01623 060302      NIOP 2
26 01624 051400      STA 2,0,3 ;RESTORE ORIGINAL CELL
27 01625 146405      SUB 2,1,SNR ;SKP IS CELL EXISTS
28 01626 000412      JMP MG32S ;MEM IS SIZED
29 01627 175400      INC 3,3
30 01630 175133      MOVZL# 3,3,SNC
31 01631 000763      JMP MS.LL
32 01632 040674      STA 0,HIGHK ;SAVE NEW TOP MEM
33 01633 030744      LDA 2,KXIST ;EXIST MAP
34 01634 006744      JSR @MS.L1 ;SET EXIST BIT
35 01635 101001      MOV 0,0,SKP
36 01636 063077      HALT ;CMAP GOOF!!!
37 01637 000744      JMP MSG32+1
38 01640 020666 MG32S: LDA 0,HIGHK
39 01641 024405      LDA 1,MPF32
40 01642 030734      LDA 2,K37
41 01643 133000      ADD 1,2
42 01644 072002      DOB 2,2 ;LOG = PHYS
43 01645 002730      JMP @XMS32 ;RETURN
44 01646 076000 MPF32: 76000

```

```

10041 N3MRT
01 ;MVETA-MOVE THE EXIST MAP
02 ;INT THE AVAILABLE MAP POSITION
03 01647 054413 MVETA: STA 3,XMVET
04 01650 030413 LDA 2,KEXMP ;A2=STR EXIST
05 01651 034413 LDA 3,KAVMP ;A3=STR AVAILABLE
06 01652 024413 LDA 1,KMX ; FOR COUNTING
07 01653 021000 LDA 0,0,2 ;WORD
08 01654 041400 STA 0,0,3 ;TO WORD
09 01655 151400 INC 2,2
10 01656 175400 INC 3,3
11 01657 125404 INC 1,1,SZR ;SKP IS DONE ALL
12 01660 000773 JMP .-5
13 01661 002401 JMP @XMVET ;
14 01662 001647 XMVET: MVETA
15 01663 015437 KEXMP: EXISM
16 01664 015447 KAVMP: AVALM
17 01665 177770 KMX: -8.
18
19 ;UBL32-SET UP USUABLE SCRATCH LIMITS IN 32K
20 ;ALL CORE ABOVE 32K IS ASSUMED TO BE USEABLE SCRATCH
21 ;CLEAR AVAILABLE BITS FOR THOSE AREAS USED
22 ;SO THAT THEY WILL NOT BE ASSIGNED AS A SCRATCH AREA
23 ;TO ANY TEST
24 ;SUBR CBLIM IS USED TO CLEAR AVAILABLE BITS
25 01666 054416 UBL32: STA 3,XUBL3 ;SAVE RETURN
26 01667 102400 SUB 0,0
27 01670 026415 LDA 1,@KNMAX ;0 TO NMAX
28 01671 006422 JSR @UBLIM ;PROTECTS PROGRAMS
29 01672 022414 LDA 0,@KUP32 ;LAST ADRS IN 32K
30 01673 105000 MOV 0,1
31 01674 006417 JSR @UBLIM ;PROTECTS LOADER
32 01675 026412 LDA 1,@KSTSS ;STRT SYMBOLS
33 01676 022412 LDA 0,@KSTES ;END SYMBOLS
34 01677 006414 JSR @UBLIM ;PROTECTS SYMBOLS
35 01700 022411 LDA 0,@KLSTB ;STRT LINKER TABLES
36 01701 026411 LDA 1,@KLETB ;END LINKER TABLES
37 01702 006411 JSR @UBLIM ;PROTECTS LINKER TABLES
38 01703 002401 JMP @XUBL3 ;RETURN
39 01704 000000 XUBL3: 0
40 01705 001004 KNMAX: USTNM
41 01706 001525 KUP32: UP32L
42 01707 001002 KSTSS: USTSS
43 01710 001003 KSTES: USTES
44 01711 001466 KLSTB: LSYTB
45 01712 001467 KLETB: LSETB
46 01713 002562 UBLIM: CBLIM

```

```

10042 N3MRT
01 ;TINIT-TEST INITIALIZE
02 ;SEQUENCE THROUGH THE INITIALIZE ADDRESSES
03 ;FOR EACH TEST LOADED ALONG WITH LINKER
04 ;1 POINTER FOR EACH TESTS PARAMETERS
05 ;IS IN ALL USED LOCATIONS ABOVE ZLOC
06 01714 054434 TINIT: STA 3,XTINI
07 01715 102400 SUB 0,0
08 01716 040433 STA 0,NPROG ;PROG TO INIT
09 01717 026565 LDA 1,@LROGK
10 01720 034431 LDA 3,NPROG ;NEXT PROG TO INIT
11 01721 166415 SUB# 3,1,SNR ;SKP IS NOT DONE ALL
12 01722 002426 JMP @XTINI ;EXIT ALL PROGS INITED
13 01723 020557 LDA 0,LR.K1
14 01724 117000 ADD 0,3
15 01725 031400 LDA 2,0,3 ;GET INIT ADRS
16 01726 020135 LDA 0,PFAS
17 01727 101024 MOVZ 0,0,SZR ;SKP IF P/F RESTART
18 01730 000407 JMP TINIA ;NOT RESTART
19 01731 021002 LDA 0,2,2 ;GET WAITI SWITCH
20 01732 101045 MOVO 0,0,SNR ;SKP=MAYBE DELETED
21 01733 000404 JMP TINIA ;RESTART NOT DELETED
22 01734 021377 LDA 0,-1,2
23 01735 101045 MOVO 0,0,SNR ;SKP=PROG WAS SELECTED
24 01736 000410 JMP XTINI-2
25 01737 102460 TINIA: SUBC 0,0
26 01740 041375 STA 0,-3,2 ;0 PASS K
27 01741 041376 STA 0,-2,2 ;0 ERR CTR
28 01742 041002 STA 0,2,2 ;CLEAR WAIT INT SW
29 01743 101003 MOV 0,0,SNR ;SKP IF P/FAIL RESTART
30 01744 041377 STA 0,-1,2 ;CLRS INTA WAIT CTR
31 01745 007000 JSR @0,2 ;AND INIT THIS TEST
32 01746 010403 ISZ NPROG ;STEP TO NXT PROG
33 01747 000750 JMP TINIT+3 ;AND DO AGN
34 01750 000000 XTINI: 0
35 01751 000000 NPROG: 0
36

```

10043 N3MRT

```
01 :LRANP-RANDOMLY SELECT A PROGRAM
02 :SCAN WAITING FOR INTERRUPT SWITCHES FOR INTR DONE
03 :ENTER ANY TEST COMPLETED INTA
04 :IF NONE ENTER RANDOM SELECT MODE
05 :IF TEST SELECTED IS WAITING INTA SELECT ANOTHER
06 :GENERATE A 2ND NUMBER CHECK TO
07 :SEE IF IT IS BETWEEN THE ENTRY LIMITS SPEC
08 :IF SO ENTER PROGRAM IF NOT SELECT ANOTHER TEST
09 01752 054531 LRANP: STA 3,LR.S3
10 01753 102400 SUB 0,0
11 01754 030526 LDA 2,LR.K1 :START OF TEST LINKS
12 01755 035000 LPR1: LDA 3,0,2 :GET TEST LINK
13 01756 175005 MOV 3,3,SNR :SKP IS TEST EXISTS
14 01757 000437 JMP LPR2 :NO ONE INTR DONE USE RAN
15 01760 025402 LDA 1,2,3 :GET INTR SW
16 01761 044141 STA 1,ALTB :SAVE FOR LATER
17 01762 125005 MOV 1,1,SNR :WAITING INT OR DISABLED?
18 01763 000430 JMP LPS1E :NO TRY NEXT TEST
19 01764 125103 MOVL 1,1,SNR :SKP=WAIT BIT STILL ON
20 01765 000435 JMP LPRGO :ENTER THIS TEST
21 01766 025777 LDA 1,-1,3 :GET INTA ELAPSED TIMER
22 01767 125005 MOV 1,1,SNR :SKP IF ACTIVE
23 01770 000423 JMP LPS1E
24 01771 034144 LDA 3,RTTIM :GET CUR RUN TIME
25 01772 166414 SUB# 3,1,SZR :SKP=RUN TIME ERR
26 01773 000420 JMP LPS1E
27 01774 035000 LDA 3,0,2
28 01775 040120 STA 0,CURPR
29 01776 011777 ISZ -1,3
30 01777 011777 ISZ -1,3
31 02000 006132 JSR @LMESS :TIME ELAPSED MESS.
32 02001 002144 LP.TX
33 02002 024120 LDA 1,CURPR :PRINT PROG#
34 02003 006071 JSR @PZOCT
35 02004 024141 LDA 1,ALTB
36 02005 006133 JSR @LPOCT
37 02006 102400 SUB 0,0
38 02007 040150 STA 0,TIMSW :FORCE TIME TYPE
39 02010 010151 ISZ ERTOT
40 02011 000401 JMP .+1
41 02012 000741 JMP LRANP+1
42 02013 101400 LPS1E: INC 0,0
43 02014 151400 INC 2,2
44 02015 000740 JMP LPR1 :TRY NEXT TEST
45 :NO TEST WAITING TO PROCESS INTERRUPT RANDOM SELECT
46 02016 006077 LPR2: JSR @ARANG
47 02017 105000 MOV 0,1
48 02020 032464 LDA 2,@LROGK :DIV RAN/#PROGS
49 02021 006100 JSR @ADIVI
```

10044 N3MRT

```
01 :SEE IF NEXT RAN# FITS BETWEEN TEST ENTER LIMITS
02 :BUT DO NOT ENTER TEST IF IT IS WAITING INTA
03 02022 040120 LPRGO: STA 0,CURPR :CURRENT PROG #
04 02023 030457 LDA 2,LR.K1 :LAST LINKR ZLOC+1
05 02024 113000 ADD 0,2 :A2=PNTR TO PARAM ADRS
06 02025 035000 LDA 3,0,2
07 02026 054142 STA 3,PSTRT :START ADDRES OF PRG
08 02027 031001 LDA 2,1,2 :GET STRT NXT PROG
09 02030 151005 MOV 2,2,SNR :#0 IS LAST PRG SEL
10 02031 032454 LDA 2,@LW.K9 :AND WE USE NMAX
11 02032 050143 STA 2,PENDA :AS THE END OF PROG
12 02033 014143 DSZ PENDA :=-1 FOR REAL END OF PROG
13 02034 031402 LDA 2,2,3
14 02035 151132 MOVZL# 2,2,SZC :C=1 IS WAITING INTR
15 02036 000760 JMP LPR2 :SELECT DIFF PROG
16 02037 030144 LDA 2,RTTIM
17 02040 151600 INCR 2,2
18 02041 151500 INCL 2,2
19 02042 051777 STA 2,-1,3
20 :IF ACS3=1 DELETE ALL RANDOM SELECT DELAYS
21 02043 034060 LDA 3,RNSEL :SW SET?
22 02044 175004 MOV 3,3,SZR
23 02045 000411 JMP LPRGA :DELETE DELAYS=1
24 02046 006077 JSR @ARANG :GET NEW RANDOM
25 02047 034142 LDA 3,PSTRT
26 02050 025403 LDA 1,3,3 :GET PRG LWR LIMIT
27 02051 122433 SUBZ# 1,0,SNR
28 02052 000744 JMP LPR2
29 02053 025404 LDA 1,4,3 :GET HIGH LIMIT
30 02054 122032 ADCZ# 1,0,SZC
31 02055 000741 JMP LPR2
32 02056 020120 LPRGA: LDA 0,CURPR :PROGRAM #
33 02057 103020 ADDZ 0,0 :*2
34 02060 105120 MOVZL 0,1 :*4
35 02061 127120 ADDZL 1,1 :*16
36 02062 125120 MOVZL 1,1 :*32
37 02063 107000 ADD 0,1 :*34
38 02064 022422 LDA 0,@XSYTB :+START OF SYS TABLES
39 02065 123000 ADD 1,0
40 02066 040141 STA 0,ALTB
41 02067 020140 LDA 0,MPSWT
42 02070 101005 MOV 0,0,SNR :USE MAP?
43 02071 000407 JMP .+7 :NOPE!
44 02072 006077 JSR @ARANG
45 02073 126520 SUBZL 1,1 :BIT 6=1 IS USER MAP B
46 02074 123700 ANDS 1,0 :IF BIT 15 = 1
47 02075 105120 MOVZL 0,1 :THEN AC0=1000 =USER B
48 02076 044115 STA 1,USRMP
49 02077 101300 MOVZ 0,0
50 02100 040116 STA 0,USRSE :AC0 = 000001=USER B FOR STATUS
51 02101 002402 JMP @LR.S3
52 02102 000160 LR.K1: LZMAX
53 02103 000000 LR.S3: 0
54 02104 001502 LROGK: PROGK
55 02105 001004 LW.K9: USTNM
56 02106 001466 XSYTB: LSYTB
```



```

10045 N3MRT
01          :LSTRP-START PROGRAM
02          :ENTER TEST SELECTED AT ITS EXECUTION ENTRY POINT
03 02107 054432 LSTRP: STA 3,L,SS3
04 02110 060104      NIOS PRTY      ;ENABLE PARITY INTR
05 02111 030142      LDA 2,PSTRY
06 02112 021001      LDA 0,1,2      ;GET EXEC ADRS
07 02113 040427      STA 0,LS,I1
08 02114 024140      LDA 1,MPSWT
09 02115 125005      MOV 1,1,SNR      ;DON'T USE MAP IF MAPSWT=0
10 02116 002424      JMP @LS,I1
11 02117 044237      STA 1,RELLO
12 02120 044240      STA 1,RELHI
13 02121 060277      INTOS
14 02122 020145      LDA 0,STATS      ;GET STATUS
15 02123 101220      MOVZR 0,0
16 02124 101120      MOVZL 0,0      ;SHIFT OUT BIT 15
17 02125 030116      LDA 2,USRSE      ;GET USER INFO
18 02126 143000      ADD 2,0      ;ADD INTO STATUS WORD
19 02127 024414      LDA 1,LS,I2      ;GET ENABLE BITS
20 02130 100000      COM 0,0
21 02131 107400      AND 0,1
22 02132 106000      ADC 0,1      ;OR IN PGM+DCH ENABLE BITS
23 02133 065002      DOA 1,2      ;SET UP STATUS REG FOR MAP
24 02134 102400      SUB 0,0
25 02135 040232      STA 0,DCHER      ;CLR DCH ERR CTR
26 02136 060177      INTEN      ;STALL
27 02137 002403      JMP @LS,I1      ;ENTER TEST
28          :LRETP-RETURN FROM TEST PROG CALL
29 02140 002401 LRETP: JMP @L,SS3      ;
30 02141 000000 L,SS3: 0
31 02142 000000 LS,I1: 0
32 02143 140000 LS,I2: 140000      ;PGM AND DCH MAP ENABLE
33 02144 005215 LP,TX: ,TXTE (<15><12>INTERRUPT WAIT ELAPSED
34 02160 005215 <15><12>PROG. NO. (

```

```

10046 N3MRT
01          :RANGN-RANDOM # GENERATOR
02          :SPIN #'S OUT IN A HURRY FORGET THE MATH
03 02167 044436 RANGN: STA 1,RN,S1
04 02170 050436      STA 2,RN,S2
05 02171 030437      LDA 2,RN,K1      ;7 FOR MASKING AT 8
06 02172 020437      LDA 0,RN,C2      ;CYCLIC CONSTANT
07 02173 024434      LDA 1,RN,C1
08 02174 133404      AND 1,2,SZR      ;ROTAT C2 EVERY 8
09 02175 000404      JMP RAN,1
10 02176 101122      MOVZL 0,0,SZC
11 02177 101400      INC 0,0
12 02200 040431      STA 0,RN,C2
13 02201 024432 RAN,1: LDA 1,RTABL
14 02202 133000      ADD 1,2      ;TO GET NXT SUM VAR
15 02203 025000      LDA 1,0,2
16 02204 123000      ADD 1,0
17 02205 041000      STA 0,0,2
18 02206 024424      LDA 1,RANNM      ;NEW SUM IN VAR
19 02207 123300      ADDS 1,0      ;LAST RAN#
20 02210 040124      STA 0,RN,ACO
21 02211 040421      STA 0,RANNM
22 02212 024137      LDA 1,IO,WK
23 02213 125004      MOV 1,1,SZR
24 02214 014137      DSZ IO,WK      ;SKP=TIME TO START TESTER
25 02215 000403      JMP ,+3
26 02216 030136      LDA 2,IO,OC
27 02217 072000      DOB 2,0      ;START I/O TESTER
28 02220 024405      LDA 1,RN,S1
29 02221 030405      LDA 2,RN,S2
30 02222 010405      ISZ RN,C1
31 02223 001400      JMP 0,3
32 02224 001400      JMP 0,3
33 02225 000000 RN,S1: 0
34 02226 000000 RN,S2: 0
35 02227 000000 RN,C1: 0
36 02230 000007 RN,K1: 7
37 02231 123456 RN,C2: 123456
38 02232 000000 RANNM: 0
39 02233 002234 RTABL: RTABL+1
40 02234 027247      027247
41 02235 145651      145651
42 02236 162724      162724
43 02237 071352      071352
44 02240 034565      034565
45 02241 116272      116272
46 02242 047135      047135
47 02243 113523      113523
48 02244 054411 RANG3: STA 3,RN,S3      ;FILL ACO TO 2 WITH RAN #'S
49 02245 004722      JSR RANGN
50 02246 111000      MOV 0,2
51 02247 004720      JSR RANGN
52 02250 105000      MOV 0,1
53 02251 004716      JSR RANGN
54 02252 044125      STA 1,RN,AC1
55 02253 050126      STA 2,RN,AC2
56 02254 002401      JMP @RN,S3
57 02255 000000 RN,S3: 0

```

10047 N3MRT

```
01 ;RMSEL-RANDOM MAP SELECT
02 ;RANDOMLY SELECT A BIT IN A MAP
03 ;CALL IS MADE:
04 ; JSR RMSEL
05 ; TRYPTR ;POINTS AT MAP SIZE#
06 ; MAPADRS ;START OF MAP ADRS
07 ;IF A 1 BIT IS NOT FOUND IN THE MAP
08 ;AFTER A # OF RANDOM TRYS = TO TRYPTR
09 ;A SEQUENTIAL SEARCH OF THE MAP IS MADE
10 ;IF NO 1 BIT EXISTS EXIT IS MADE TO CALL+3
11 ;OTHERWISE EXIT IS TO CALL+4 WITH ACO=BIT#
12 ;THIS ROUTINE IS USED BY ASCRA ESCRA AND ADMAP
13 ;TO SELECT SCRATCH AND DATA CHANNEL ASSIGNMENTS
14 02256 054441 RMSEL: STA 3,XRMSE
15 02257 023400 LDA 0,00,3
16 02260 025401 LDA 1,1,3
17 02261 040433 STA 0,RM,P1
18 02262 044433 STA 1,RM,P2
19 02263 100000 COM 0,0
20 02264 040432 STA 0,RM,P3
21 02265 006077 RM.L1: JSR @ARANG ;GET RANDOM
22 02266 030426 LDA 2,RM,P1
23 02267 151400 INC 2,2
24 02270 105000 MOV 0,1
25 02271 006100 JSR @ADIVI ;REM IN ACO=BIT SEL
26 02272 030423 LDA 2,RM,P2
27 02273 006425 JSR @ICMPB ;SKIP IF BIT =1
28 02274 006424 JSR @ICMPB ;CHNG IT BK TO 0 SKP
29 02275 000415 JMP RM,FN ;FOUND 1 EXIT
30 02276 010420 ISZ RM,P3
31 02277 000766 JMP RM,L1
32 02300 020416 RM.L2: LDA 0,RM,P3 ;GET NXT SEQ BIT
33 02301 006417 JSR @ICMPB ;SKP IF =1
34 02302 006416 JSR @ICMPB ;0 BIT AND SKIP
35 02303 000407 JMP RM,FN
36 02304 024410 LDA 1,RM,P1
37 02305 010411 ISZ RM,P3
38 02306 106404 SUB 0,1,SZR ;SKP IF SRCHED WHOLE TBL
39 02307 000771 JMP RM,L2
40 02310 034407 LDA 3,XRMSE
41 02311 001402 JMP 2,3 ;NO FIND EXIT
42 02312 010405 RM.FN: ISZ XRMSE
43 02313 000775 JMP RM,FN-2
44 02314 000000 RM.P1: 0
45 02315 000000 RM.P2: 0
46 02316 000000 RM.P3: 0
47 02317 000000 XRMSE: 0
48 02320 002627 ICMPB: CMAPB
```

10048 N3MRT

```
01 ;DIVID-DIVIDE AC1 BY AC2
02 ;LEAVE WITH REM IN ACO
03 02321 102400 DIVID: SUB 0,0
04 02322 054412 STA 3,DI,S3
05 02323 034412 LDA 3,DI,K1
06 02324 125120 MOVZL 1,1
07 02325 101100 DI.L1: MOVL 0,0
08 02326 142412 SUB# 2,0,SZC
09 02327 142400 SUB 2,0
10 02330 125100 MOVL 1,1
11 02331 175404 INC 3,3,SZR
12 02332 000773 JMP DI,L1
13 02333 002401 JMP @DI,S3
14 02334 000000 DI.S3: 0
15 02335 177760 DI.K1: -16.
```

```

10049 N3MRT
01
02 ;CHANGE STACK POSITIONS IN MEMORY
03 02336 054452 CHSTK: STA 3,CHSAV
04 02337 006077 JSR @ARANG
05 02340 101203 MOVR 0,0,SNC ;BITS 14/15=00
06 02341 101202 MOVR 0,0,SZC ;THEN CHANGE STACK
07 02342 000454 JMP SPLAY ;PLAY WITH STACK
08 ;CHECK TO SEE IF STACK OK BEFORE CHANGE
09 02343 071201 MFSP 2
10 02344 024445 LDA 1,STADR
11 02345 050445 STA 2,NWSTK
12 02346 132414 SUB# 1,2,SZR ;SKP=STACK OK
13 02347 004537 JSR SKERR ;STACK IS FUNNY
14 02350 151300 MOVS 2,2
15 02351 151225 MOVZR 2,2,SNR ;SKP IF STK NOT AT 400
16 02352 000421 JMP CHPAG ;MOVE IT ELSEWHERE
17 ;STACK IS NOT AT 400 RELEASE 1K PAGE
18 ;AFTER RESTORING STACK TO ADRS 400
19 02353 034441 LDA 3,CHSK1 ;400
20 02354 054435 STA 3,STADR
21 02355 060277 INTDS
22 02356 075001 MTSP 3
23 02357 060177 INTEN
24 02360 074001 MTFP 3
25 02361 030414 LDA 2,CHPAG+2 ;AVALM
26 02362 020431 LDA 0,STKPG ;PHYS PAG#
27 02363 101005 MOV 0,0,SNR ;SHD NOT BE PAGO
28 02364 004522 JSR SKERR ;STACK IS IN ERROR
29 02365 006430 JSR @CHSK1+1 ;CMAPB MAKE 1K AVAL
30 02366 102401 SUB 0,0,SKP ;:BIT HAD TO GO 0 TO1
31 02367 004517 JSR SKERR ;THIS CAN'T HAPPEN
32 02370 040423 STA 0,STKPG
33 02371 002417 JMP @CHSAV
34
35
36 02372 000037 STKLM: 37 ;LIMIT STACK TO 32K
37 ;RANDOMLY SELECT 1K OF CORE TO MOV THE STACK INTO
38 02373 004663 CHPAG: JSR RMSEL ;SELECTS 1K CORE
39 02374 002372 STKLM ;WITHIN LIMIT
40 02375 015447 AVALM ;OF AVAL CORE
41 02376 002412 JMP @CHSAV ;NONE AVAL TO USE
42 02377 040414 STA 0,STKPG ;SAVE PHYS PAGE
43 02400 101300 MOVS 0,0
44 02401 103120 ADDZL 0,0
45 02402 040407 STA 0,STADR
46 02403 060277 INTDS
47 02404 060001 MTFP 0
48 02405 060177 INTEN
49 02406 061001 MTSP 0
50 02407 002401 JMP @CHSAV

```

```

10050 N3MRT
01
02 02410 000000 CHSAV: 0
03 02411 000000 STADR: 0
04 02412 000000 NWSTK: 0
05 02413 000000 STKPG: 0
06 02414 000400 CHSK1: 400
07 02415 002627 CMAPB
08
09 ;DO SOME FUNNY THINGS WITH STACK TO
10 ;TRY TO GOOF IT UP
11 02416 071201 SPLAY: MFSP 2
12 02417 126000 ADC 1,1 ;:-1
13 02420 133000 ADD 1,2
14 02421 044411 STA 1,STKIN-1 ;STK INTA FLG
15 02422 060277 INTDS
16 02423 071001 MTSP 2 ;RELOAD STACK POINTER
17 02424 061401 PSH 0
18 02425 025001 LDA 1,1,2
19 02426 060177 INTEN
20 02427 122414 SUB# 1,0,SZR ;(ACO) GET THERE OK
21 02430 004456 JSR SKERR ;NO! PSH ERROR
22 02431 002757 JMP @CHSAV
23
24 ;STACK OVERFLOW ROUTINE
25 02432 000000 0
26 02433 040427 STKIN: STA 0,STSV0
27 02434 044427 STA 1,STSV1
28 02435 050427 STA 2,STSV2
29 02436 054427 STA 3,STSV3
30 02437 101200 MOVR 0,0
31 02440 040426 STA 0,STSV0
32 02441 010771 ISZ STKIN-1
33 02442 000402 JMP .+2
34 02443 000406 JMP .+6
35 02444 006441 JSR @SKERR-1 ;PRINT ERR MESS.
36 02445 002467 STKTZ
37 02446 024000 LDA 1,0
38 02447 006435 JSR @SKERR-2
39 02450 006411 JSR @ST.K6 ;WAIT FOR OPERATOR INPUT
40 02451 034414 LDA 3,STSV3
41 02452 030412 LDA 2,STSV2
42 02453 024410 LDA 1,STSV1
43 02454 020412 LDA 0,STSV0
44 02455 101100 MOVL 0,0
45 02456 020404 LDA 0,STSV0
46 02457 060177 INTEN
47 02460 002000 JMP @0 ;RETURN
48
49 02461 004065 ST.K6: KEY6W
50 02462 000000 STSV0: 0
51 02463 000000 STSV1: 0
52 02464 000000 STSV2: 0
53 02465 000000 STSV3: 0
54 02466 000000 STSVC: 0
55 02467 005215 STKTZ: .TXTE !<15><12>STACK OVERFLOW ERROR @1
56 000000 .NOLOC 0

```

10051 N3MRT

```
01
02 ;STACK IS IN ERROR SOMEHOW
03 ;MAKE ERROR TYPEOUT
04 02504 004747 POC?
05 02505 004720 MES?
06 02506 054020 SKERR: STA 3,20
07 02507 006776 JSR @SKERR-1
08 02510 002536 STKTX
09 02511 024020 LDA 1,20 ;ADRS ERR DETECTED
10 02512 006772 JSR @SKERR-2
11 02513 024676 LDA 1,STADR ;WHERE STACK SHD BE
12 02514 006770 JSR @SKERR-2
13 02515 024675 LDA 1,NWSTK
14 02516 006766 JSR @SKERR-2 ;WHERE IT'S AT
15 ;RESTORE STACK TO 400
16 02517 034675 LDA 3,CHSK1
17 02520 060277 INTDS
18 02521 075001 MTSP 3
19 02522 074001 MTFP 3
20 02523 060177 INTEN
21 02524 054665 STA 3,STADR
22 02525 020666 LDA 0,STKPG ;MAKE 1K AVAL
23 02526 101005 MOV 0,0,SNR ;IF NOT PAGE 0
24 02527 002661 JMP @CHSAV
25 02530 030645 LDA 2,CHPAG+2
26 02531 006664 JSR @CHSK1+1
27 02532 102401 SUB 0,0,SKP ;IT'S RELEASED
28 02533 000776 JMP .-2
29 02534 040657 STA 0,STKPG
30 02535 002653 JMP @CHSAV
31 02536 005215 STKTX: .TXTE 1<15><12>STACK ERROR<15><12>
32 02545 142412 ERROR@ EXPECTED ACTUAL<15><12>!
```

10052 N3MRT

```
01
02 ;CBLIM-CLEAR AVAILABLE BITS BETWEEN LIMITS
03 ;ACO=LOWEST ADRS AC1=HIGHEST ADDRESS
04 ;SINCE SOME USED AREAS MAY OVERLAP IN PAGES
05 ;OCCASIONALLY 2 PASSES THROUGH CMAPB WILL BE REQ
06 02562 054426 CBLIM: STA 3,XCBLM
07 02563 101300 MOV# 0,0 ;POS 1K
08 02564 125300 MOV# 1,1 ;FIELD BITS
09 02565 101220 MOVZR 0,0 ;FOR ADRS LIMITS
10 02566 101220 MOVZR 0,0 ;IN AC 0 AND 1
11 02567 125220 MOVZR 1,1
12 02570 125220 MOVZR 1,1
13 02571 030424 LDA 2,K37C
14 02572 143400 AND 2,0
15 02573 147400 AND 2,1
16 02574 040417 STA 0,CBLWR ;LWST 1K FLD
17 02575 044417 STA 1,CBUPR ;HGHST 1K FLD
18 02576 020415 LDA 0,CBLWR ;LWR INCS TO=UPR
19 02577 030413 LDA 2,KAVLM ;AVAILABLE MAP
20 02600 006411 JSR @XCMPB ;COM BIT IN AVAIL MAP
21 02601 000777 JMP .-1 ;WENT 0-1 MAKE IT 1-0
22 02602 020411 LDA 0,CBLWR ;ACO=LST CLRED
23 02603 010410 ISZ CBLWR ;+1 LWR IN CASE NOT DONE
24 02604 024410 LDA 1,CBUPR
25 02605 106404 SUB 0,1,SZR ;SKP IS ALL REQ UNAVAILABLE
26 02606 000770 JMP .-10
27 02607 002401 JMP @XCBLM
28 02610 000000 XCBLM: 0
29 02611 002627 XCMPB: CMAPB
30 02612 015447 KAVLM: AVALM
31 02613 000000 CBLWR: 0
32 02614 000000 CBUPR: 0
33 02615 000037 K37C: 37
```

10053 N3MRT

```
01 ;GETPA-GET A PHYSICAL ASSIGNMENT
02 ;ACO=ALLOCATION TABLE POSITION
03 ;RETURN WITH AC1=PHYSICAL PAGE ASSIGNMENT
04 ;SKIP EXIT IF THE ALLOC ASSIGN ISN'T ACCESS PROTECTED
05 ;THE INTEGRITY OF ACO IS PRESERVED
06 02616 040410 GETPA: STA 0,GPA.0
07 02617 030141 LDA 2,ALTB1 ;ADRS OF ALLOCATION TABLE
08 02620 113000 ADD 0,2 ;+ WORD POSITION IN TABLE
09 02621 025002 LDA 1,2,2 ;2 PAST CTR AND DCH CONTR.
10 02622 124014 COM# 1,1,SZR ;ACCESS PROTECTED ?
11 02623 175400 INC 3,3 ;SKP EXIT PAGE ASSIGNED
12 02624 020402 LDA 0,GPA.0
13 02625 001400 JMP 0,3
14 02626 000000 GPA.0: 0
15
16 ;CMAPB-COMPLIMENT MAP BIT
17 ;COMPLIMENT THE STATE OFAMEMORY MAP BIT
18 ;THE START ADRS OF THE MAP IS INAC2
19 ;ACO CONTAINS UP TO 7 BITS OF ADDRESSWITH
20 ;BITS 12TO15=# BITS TO SHIFT LEFT
21 ;BITS 9TO11=WORD POSITION IN MAP
22 ;THOSE 7 BITS ARE THE PHYSICL PG # OF A 1K OF MEM
23 ;SKIP ON EXIT IF THE BIT IS GOING 1-0
24 02627 040436 CMAPB: STA 0,CM.S0
25 02630 044436 STA 1,CM.S1
26 02631 050436 STA 2,CM.S2
27 02632 105220 MOVZR 0,1 ;POSITION WORD # IN AC1
28 02633 125220 MOVZR 1,1
29 02634 125220 MOVZR 1,1
30 02635 125220 MOVZR 1,1
31 02636 133000 ADD 1,2 ;AC2=TABLE (MAP) ADDRESS
32 02637 024431 LDA 1,K17
33 02640 107400 AND 0,1 ;AC1=#PLACES TO SHIFT
34 02641 124000 COM 1,1
35 02642 044422 STA 1,CCTR ;FOR COUNTING SHIFTS
36 02643 126420 SUBZ 1,1 ;AC1=0 C=1
37 02644 125100 MOVL 1,1 ;POSIT BIT
38 02645 010417 ISZ CCTR
39 02646 000776 JMP .-2
40 02647 021000 LDA 0,0,2 ;GET MAP WORD
41 02650 050414 STA 2,CCTR ;SV ADRS
42 02651 131000 MOV 1,2 ;FOR BIT XOR
43 02652 113525 ANDZL 0,2,SNR ;NOT 0 IN RESULT
44 02653 151002 MOV 2,2,SZC ;OR IN CARRY BIT
45 02654 175400 INC 3,3 ;IS SKIP WHEN EXIT
46 02655 107000 ADD 0,1 ;FORM REST OF
47 02656 146400 SUB 2,1 ;BIT XOR
48 02657 046405 STA 1,CCTR ;PUT NEW WORD BACK
49 02660 020405 LDA 0,CM.S0
50 02661 024405 LDA 1,CM.S1
51 02662 030405 LDA 2,CM.S2
52 02663 001400 JMP 0,3
53 02664 000000 CCTR: 0
54 02665 000000 CM.S0: 0
55 02666 000000 CM.S1: 0
56 02667 000000 CM.S2: 0
57 02670 000017 K17: 17
```

10054 N3MRT

```
01 ;CDISP-LINKER CALL DISPATCH ROUTINE
02 ;DIRECTS MEM ALLOCATION AND OTHER CALLS
03 ;TO THE CORRECT HANDLER FOR PROCESSING
04 ;CALLS ARE MADE AS FOLLOWS:
05 ; LCALL CALLN ;ZPAGE LOCS =CALL ADRS
06 ; ERROR RETURN
07 ; NORMAL RETURN
08 ;CALLN WILL BE = TO ONE OF THE FOLLOWING
09 ;ASCRA ASSIGN A SCRATCH AREA
10 ;ESCRA EXPAND SCRATCH AREA
11 ;RSCRA RELEASE SCRATCH AREA
12 ;MSCRA MOVE TEST TO SCRATCH
13 ;GSCRA GO TO SCRATCH FOR EXECUTION
14 ;SSCRA SHUFFLE SCRATCH AREA ASSIGNED
15 ;RETRN RETURN FROM TEST EXECUTION
16 ;ARANG RANDOM # GENERATION
17 ;RNDMP RANDOM SEL DCH MAP A/B
18 ;ADMAP ASSIGN DCH MAP
19 ;EDMAP EXPAND DCH MAP
20 ;RDMAP RELEASE DCH FROM MAP
```

```

10055 N3MRT
01      ;ILLEGAL SUPERVISOR CALL TYPEOUT
02 02671 006132 ICALL: JSR @LMESS
03 02672 002717 ICALT      ;TEXT
04 02673 024576      LDA 1,CD.LA
05 02674 006133      JSR @LPDCT
06 02675 020566      LDA 0,CD.S0
07 02676 024566      LDA 1,CD.S1
08 02677 030566      LDA 2,CD.S2
09 02700 006102      JSR @ERROI      ;TYPE PR# AC'S ETC
10 02701 000401      JMP .+1
11 02702 020564      LDA 0,CD.S3
12 02703 024570      LDA 1,CD.PP
13 02704 030564      LDA 2,CD.IN      ;INSTRUCTION CAUSING TRAP
14 02705 006103      JSR @ERROC
15 02706 000401      JMP .+1
16 02707 030142      LDA 2,PSTRY
17 02710 035006      LDA 3,6,2
18 02711 175120      MOVZL 3,3
19 02712 024062      LDA 1,USESW
20 02713 125004      MOV 1,1,SZR      ;USE ICALX RETURN IF USESW NOT = 0
21 02714 000420      JMP ICALX
22 02715 002401      JMP @.+1
23 02716 006067      IOV,E
24 02717 005215 ICALT: .TXTE (<15><12>
25 02720 144411      ILLEGAL SUPER CALL AT (
26      ;
27      ;
28      ;
29      ;ICALX- TRAP RETURN IF ILLEGAL CALL BECAUSE OF USESW
30      ;
31 02734 175220 ICALX: MOVZR 3,3      ;AC0-1-2 = MAP REGISTERS
32 02735 054402      STA 3,ICLX.3      ;
33 02736 002401      JMP @ICLX.3
34 02737 000000 ICLX.3: 0

```

```

10056 N3MRT
01      ;SYSTEM CALL DISPATCH ROUTINE
02 02740 040523 COISP: STA 0,CD.S0
03 02741 044523      STA 1,CD.S1
04 02742 050523      STA 2,CD.S2
05 02743 054523      STA 3,CD.S3
06 02744 020140      LDA 0,MPSWT
07 02745 101004      MOV 0,0,SZR      ;SKIP IS DON'T USE MAP
08 02746 000403      JMP .+3
09 02747 034046      LDA 3,TPLOC      ;PC OF TRAPPED INSTRUCTION
10 02750 000440      JMP CDGLC      ;MAP WASN'T USED
11 02751 060277      INTDS
12 02752 030231      LDA 2,PRTCK      ;PROECTION OPTION EXIST?
13 02753 151005      MOV 2,2,SNR      ;SKP=EXIST
14 02754 000414      JMP COIST      ;NO PROTECT OPTION
15 02755 063402      SKPBN 2      ;ANY DCH ERRS?
16 02756 000403      JMP .+3      ;NOT YET
17 02757 010232      ISZ DCHER      ;AT LEAST ONE HAS OCCURED
18 02760 000401      JMP .+1
19 02761 071402      DIB 2,2      ;CHECK FOR A VIOL..
20 02762 151133      MOVZL# 2,2,SNC      ;SKIP IS A VIOLATION
21 02763 000404      JMP .+4
22 02764 075403      DIB 3,3      ;GET VIOL. PC REG.
23 02765 054233      STA 3,VLPCR      ;SAVE IT
24 02766 050234      STA 2,VLDTA      ;SAVE VIOL. DATA
25 02767 060202      NIOC 2      ;CLR ANY VIOLATIONS
26 02770 060402 COIST: DIA 0,2      ;GET STATUS
27 02771 040145      STA 0,STATS      ;SAVE IT
28 02772 126620      SUBZR 1,1
29 02773 125220      MOVZR 1,1      ;FORM DCH ENABLE BIT USER A
30 02774 101232      MOVZR# 0,0,SZC      ;SKP IF USER A
31 02775 024501      LDA 1,CD.40
32 02776 065002      DOA 1,2      ;WRITE STATUS WORD
33 02777 151133      MOVZL# 2,2,SNC      ;SKIP IS VIOL
34 03000 000405      JMP .+5
35 03001 024476      LDA 1,CD.44      ;MASK TO USE
36 03002 133415      AND# 1,2,SNR      ;SKIP IS A I/O OR VALIDITY VIOL.
37 03003 002236      JMP @IDWCK      ;MUST BE A WRITE/DEFER VIOL.
38 03004 002235      JMP @IIOVL      ;WAS AN I/O OR VALIDITY VIOL.
39 03005 034046      LDA 3,TPLOC      ;GET TRAP PC
40 03006 060177      INTEN
41 03007 060302      NIOP 2      ;SINGLE CYCLE CMD
42 03010 021400      COGLC: LDA 0,0,3      ;GET CALL
43 03011 054460      STA 3,CD.LA      ;SV LGICL ADRS
44 03012 040456      STA 0,CD.IN      ;SAVE INSTRUCTION CAUSING TRAP
45 03013 030454      LDA 2,K1777
46 03014 150000      COM 2,2
47 03015 157400      AND 2,3
48 03016 054454      STA 3,CD.LP      ;SAVE LOG PAGE
49 03017 024140      LDA 1,MPSWT
50 03020 125005      MOV 1,1,SNR      ;OK TO USE MAP?
51 03021 000407      JMP .+7      ;NOPE!
52 03022 024115      LDA 1,USRMP
53 03023 137000      ADD 1,3
54 03024 075003      DOA 3,3      ;SELECT PAGE
55 03025 074403      DIA 3,3      ;READ MAP ENTRY TO GET PHYS PG #
56 03026 150220      COMZR 2,2
57 03027 157400      AND 2,3
58 03030 054443      STA 3,CD.PP      ;SAVE PHYS. PG#
59      ;NOW DETERMINE IF IT IS A VALID CALL
60 03031 030443      LDA 2,CALLS      ;START OF CALLS

```

```

0057 N3MRT
01 03032 142433 SUBZ# 2,0,SNC ;MUST BE=>
02 03033 000636 JMP ICALL ;ILLEGAL CALL?
03 03034 024441 LDA 1,CALLE ;LAST VALID CALL
04 03035 122032 ADCZ# 1,0,SZC ;MUST BE =<
05 03036 000633 JMP ICALL ;INVALID CALL?
06 03037 142640 SUBOR 2,0 ;CREATE CALL ADRS
07 03040 115220 MOVZR 0,3 ;MOV 4 R
08 03041 175220 MOVZR 3,3
09 03042 175220 MOVZR 3,3 ;(AC3)=CALL#
10 03043 020420 LDA 0,CD.S0
11 03044 024420 LDA 1,CD.S1
12 03045 030420 LDA 2,CD.S2
13 03046 007464 CD.EX: JSR @ASCRA,3 ;CALL JSR
14 03047 101011 MOV# 0,0,SKP ;ERROR RET
15 03050 010421 ISZ CD.LA ;NORMAL +1 RETURN
16 03051 010420 ISZ CD.LA ;TO GET PAST JSR
17 03052 034140 LDA 3,MPSWT
18 03053 175005 MOV 3,3,SNR
19 03054 000403 JMP .+3
20 03055 034145 LDA 3,STATS
21 03056 060277 INTDS
22 03057 075002 DOA 3,2 ;RESTORE MAP STATUS
23 03060 034406 LDA 3,CD.S3
24 03061 060177 INTEN
25 03062 002407 JMP @CD.LA
26 ;ABOVE JMP RETURNS TO USER
27 03063 000000 CD.S0: 0
28 03064 000000 CD.S1: 0
29 03065 000000 CD.S2: 0
30 03066 000000 CD.S3: 0
31 03067 001777 K1777: 1777
32 03070 000000 CD.IN: 0
33 03071 000000 CD.LA: 0
34 03072 000000 CD.LP: 0
35 03073 000000 CD.PP: 0
36 CALLS: LCALL ASCRA
37 03074 100010 ASCRA-ASCRA*1811+100010
38 CALLE: LCALL RDMAP
39 03075 100610 RDMAP-ASCRA*1811+100010
40 03076 040040 CD.40: 40040
41 03077 000044 CD.44: 44

```

```

10058 N3MRT
01 ;ENTPA=ENTER PHYSICAL ASSIGNMENT
02 ;THE PHYSICAL PAGE # IN ACO IS ENTERED INTO
03 ;THE MEMORY ALLOCATION TABLE ASSOCIATED WITH
04 ;THE TEST THAT IS CURRENTLY ACTIVE
05 03100 026141 ENTPA: LDA 1,@ALTBL ;GET # ENTRIES
06 03101 030141 LDA 2,ALTBL ;ADDRS OF TBL
07 03102 133000 ADD 1,2 ;+WORD #
08 03103 041002 STA 0,2,2 ;PLACE ENTRY IN TABLE
09 03104 012141 ISZ @ALTBL ;+1# OF ENTRIES
10 03105 001400 JMP 0.3

```

```

10059 N3MRT
01      ;LDMAP=LOAD MAP OPTION FOR FIRST LEVEL TEST
02      ;ADJUST THE CONTENTS OF SCRLO AND SCRHI
03      ;PROTECT ALL PAGES NOT REQ BY TEST
04      LDMAP:
05 03106 062401      SAVE
06 03107 102400      SUB      0,0
07 03110 040154      STA      0,SCRLO
08 03111 040155      STA      0,SCRHI
09 03112 024140      LDA      1,MPSWT ;=0 IS MAP OPT. NOT EXIST
10 03113 125005      MOV      1,1,SNR
11 03114 000477      JMP      LM,NM ;MAP OPT NONEXIS
12 03115 020115      LDA      0,USRMP ;ADD USER
13 03116 062002      DOB      0,2 ;MAP PAGE 0 TO ITSELF
14 03117 102400      SUB      0,0
15 03120 101400      LM.L1: INC      0,0
16 03121 024465      LDA      1,LM,K1 ;LOAD PAGES 1-37
17 03122 111300      MOVS     0,2
18 03123 153120      ADDZL   2,2
19 03124 034115      LDA      3,USRMP
20 03125 173000      ADD      3,2
21 03126 133000      ADD      1,2 ;ADD IN 777
22 03127 072002      DOB      2,2 ;VALIDITY PROTECT THIS PAGE
23 03130 030457      LDA      2,LM,37
24 03131 112414      SUB#    0,2,SZR ;DONE?
25 03132 000766      JMP      LM,L1
26      ;ALL PAGES BUT LOG 0(IT =90) ACCESS PROTECTED
27      ;MAP TEST PROGRAM TO ITSELF-MUST BE LESS THAN 1K
28 03133 024142      LDA      1,PSTRT ;START ADRS OF PROG
29 03134 121300      LM.L2: MOVS     1,0
30 03135 101220      MOVZR   0,0 ;1K FIELD TO
31 03136 101220      MOVZR   0,0 ;LOWER 5 BITS
32 03137 024450      LDA      1,LM,37
33 03140 123400      AND      1,0
34 03141 105300      MOVS     0,1
35 03142 127120      ADDZL   1,1
36 03143 107000      ADD      0,1
37 03144 030115      LDA      2,USRMP ;GET USER INFO
38 03145 147000      ADD      2,1 ;PUT IN MAP ENTRY WORD
39 03146 066002      DOB      1,2 ;LOG=PHYS
40      .ENDC
41 03147 105700      INCS     0,1
42 03150 127120      ADDZL   1,1
43 03151 030143      LDA      2,PENDA ;LAST USED BY TEST
44 03152 132432      SUBZ#   1,2,SZC ;C=0 IS MAPPED LAST ALRDY
45 03153 000761      JMP      LM,L2
46      ;ENTER NEXT SEQUENCE WITH AC0=HIGHEST PAGE USED FOR TEST

```

```

10060 N3MRT
01      ;IF TEST HAS SCRATCH AREA ASSIGNED MAP AS MUCH AS POSSIBLE
02      ;STARTING AT THE FIRST 1K ABOVE PROGRAM STORAGE
03 03154 026141      LDA      1,@ALTLB ;# 1K FIELDS ASSIGNED
04 03155 125005      MOV      1,1,SNR
05 03156 000427      JMP      LM,DN ;NO SCRATCH EXIT
06 03157 101400      INC      0,0
07 03160 105300      MOVS     0,1
08 03161 127120      ADDZL   1,1
09 03162 044154      STA      1,SCRLO ;START OF ACCESSABLE SCRAT
10 03163 040425      STA      0,LM,TM
11 03164 102400      SUB      0,0
12 03165 006424      LM.L3: JSR      @LGTPA ;NEXT PHYS PAGE ASSIGNED
13 03166 000417      JMP      LM,DN ;DONE ALL EXIT
14 03167 030421      LDA      2,LM,TM ;AC2=LOGICAL PG. #
15 03170 151300      MOVS     2,2
16 03171 153120      ADDZL   2,2
17 03172 147000      ADD      2,1
18 03173 034115      LDA      3,USRMP
19 03174 167000      ADD      3,1
20 03175 066002      DOB      1,2 ;LOG# PHYS
21 03176 024414      LDA      1,LM,K2 ;K2=1777
22 03177 133000      ADD      1,2
23 03200 050155      STA      2,SCRHI ;NEW HI SCRATCH LIMIT
24 03201 101400      INC      0,0 ;+1 ALLOC TBL POS
25 03202 010406      ISZ     LM,TM ;+1 LOGICAL PAGE
26 03203 151523      INCZL   2,2,SNC ;C=1 IF HI IS 77777(32K)
27 03204 000761      JMP      LM,L3
28      ;MAP OPTION IS SET UP FOR A FIRST LEVEL TEST
29      LM.DN:
30 03205 062601      RTRN
31 03206 000777      LM.K1: 777
32 03207 000037      LM.37: 37
33 03210 000000      LM.TM: 0
34 03211 002616      LGTPA: GETPA
35 03212 001777      LM.K2: 1777

```



10061 N3MRT

```
01 ;MAP OPTION DOES NOT EXIST
02 ;SIMPLY SET LIMITS TO SCRATCH AREA ASSIGNED
03 03213 006776 LM.NM: JSR @LGTPA ;SKP=AC1 PHYS PAGE#
04 03214 000771 JMP LM.DN ;EXIT NO SCRATCH
05 03215 125300 MOVS 1,1
06 03216 127120 ADDZL 1,1
07 03217 044154 STA 1,SCRLO ;LOW=FIRST PHYS 1K
08 03220 006771 LM.L4: JSR @LGTPA ;SKP=AC1=PHYS PG#
09 03221 000764 JMP LM.DN ;EXIT SCRHI ADJUSTED
10 03222 125300 MOVS 1,1
11 03223 127120 ADDZL 1,1 ;PG# POSITIONED TO PHYS
12 03224 030766 LDA 2,LM.K2
13 03225 133000 ADD 1,2
14 03226 050155 STA 2,SCRHI ;NO TEST CAN HAVE 32K IF
15 03227 101400 INC 0,0
16 03230 000770 JMP LM.L4 ;MAP OPTION NONEXIST
```

10062 N3MRT

```
01 ;ASSCR=ASSIGN A SCRATCH AREA
02 ;RANDOMLY OR SEQUENTIALLY IF NECESSARY
03 ;ASSIGN 1K SCRATCH TO TEST SKIP ON EXIT
04 ;NO SKIP IF MEMORY ALREADY ASSIGNED
05 ;OR NO SCRATCH AREA AVAILABLE TO ASSIGN
06 ASSCR:
07 03231 062401 SAVE
08 03232 054414 STA 3,AS.S3
09 03233 022141 LDA 0,@ALTB ;GET #1K'S ASSIGNED
10 03234 101004 MOV 0,0,SZR ;NOT=0 INVALID
11 03235 000407 JMP AS.XT
12 03236 006411 AS.G1: JSR @MSEL ;SELECT A PAGE
13 03237 001526 HIGHK ;MAX # 1K PAGES
14 03240 015447 AVALM ;AVAILABLE MAP
15 03241 000403 JMP AS.XT ;NO CORE AVAILABLE
16 03242 004636 JSR ENTPA ;ACO=PHYS PAGE# ENTER IT
17 03243 012403 ISZ @AS.S3
18 03244 004642 AS.XT: JSR LDMAP ;LOAD MAP OPT. SET SCRLO+HI
19 03245 062601 RTRN
20 03246 000000 AS.S3: 0
21 03247 002256 MSEL: RMSEL
```

10063 N3MRT

```
01 ;EXSCR-EXPAND SCRATCH AREA ASSIGNED
02 ;IF MAPPING OPTION EXISTS RANDOM SELECT
03 ;NO MAPPING OPT. TRY NEXT SEQUENTIAL
04 ;RETURN IS TO CALL +1 NO SCRATCH ASSIGNED
05 ;RETURN CALL +2 IF SCRATCH WAS EXPANDED
06 EXSCR:
07 03250 062401 SAVE
08 03251 054775 STA 3,AS.S3
09 03252 022141 LDA 0,@ALTB1 ;# 1K'S ASSIGNED
10 03253 101005 MOV 0,0,SNR
11 03254 000770 JMP AS.XT ;CANT EXPAND 0 ASSIGNED
12 03255 024140 LDA 1,MPSWT ;=0 IS NO MAP OPT.
13 03256 125004 MOV 1,1,SZR ;SKP ON NO MAPPING
14 03257 000757 JMP AS.G1 ;USE RANDOM SELECT
15 ;MAPPING OPTION DOES NOT EXIST ASSIGN NEXT SEQ 1K
16 ;UNLESS IT IS ALREADY BEING USED
17 03260 100400 NEG 0,0
18 03261 100000 COM 0,0 ;CALC ALLOC TBLE POS
19 03262 006436 JSR @IGTPA ;EXTRACT PHYS PGE #
20 03263 063077 HALT ;#=-1 CAN'T HAPPEN
21 03264 121400 INC 1,0 ;AC0=NXT PHYS PAGE
22 03265 030753 LDA 2,AS.G1+2 ;ADRS OF AVAILABLE TBL
23 03266 006406 JSR @EX.I1 ;CMAPB SKPS IF AVAILABLE
24 03267 101001 MOV 0,0,SKP
25 03270 000752 JMP AS.XT-2 ;1K AVAILABLE ENTER AND REMAP
26 03271 006403 JSR @EX.I1 ;CMAPB HAS TO SKP
27 03272 063077 HALT
28 03273 000751 JMP AS.XT ;EXIT NO EXPANSION
29 03274 002627 EX.I1: CMAPB
```

10064 N3MRT

```
01 ;RLSCR-RELEASE SCRATCH AREA
02 ;REMOVE 1 1K SCRATCH FROM MEM ALLOCATION
03 ;EXIT IS TO CALL +1 ALL SCRATCH RELEASED
04 ;EXIT TO CALL +2 IF STILL SCRATCH LEFT
05 RLSCR:
06 03275 062401 SAVE
07 03276 054750 STA 3,AS.S3
08 03277 022141 LDA 0,@ALTB1 ;#1K'S ASSIGNED
09 03300 100405 NEG 0,0,SNR ;SKP IF ANY ASSIGNED
10 03301 000743 JMP AS.XT ;EXIT NONE TO RELEASE
11 03302 100000 COM 0,0 ;AC0=#1K'S -1
12 03303 042141 STA 0,@ALTB1 ;TO ENTER 377 LATER
13 03304 006414 JSR @IGTPA ;GET PHYS PAGE #
14 03305 063077 HALT ;ASSIGNED CAN'T=-1
15 03306 121000 MOV 1,0
16 03307 030731 LDA 2,AS.G1+2 ;A2=AVAILABLE MAP
17 03310 006764 JSR @EX.I1 ;CMAPB MAKES 1K AVAIL
18 03311 101001 MOV 0,0,SKP
19 03312 063077 HALT ;BIT FOR THAT 1K HAD TO =0
20 03313 102000 ADC 0,0
21 03314 006405 JSR @NTPA ;PUT -1 IN ALLOCATION TBL
22 03315 016141 DSZ @ALTB1 ;-1 # PAGES ASSIGNED
23 03316 000725 JMP AS.XT-1 ;STILL PAGES LEFT+1 EXIT
24 03317 000725 JMP AS.XT ;0 MEM ALLOCATED DON'T SKIP
25 03320 002616 IGTPA: GETPA
26 03321 003100 NTPA: ENTPA
```

```

10065 N3MRT
01          ;GOSCR = GO TO SCRATCH
02          ;ENTERED WITH ACO=LOGICAL PAGE TO
03          ;REMAP SCRATCH TO AC1=ERROR RET
04          ;AC2=ADDRESS TO START EXECUTION IN
05          ;THE REMAPPED SCRATCH
06
07 03322 040556 GOSCR: STA 0,GO.00      ;SAVE CALL PARAMS
08 03323 044556      STA 1,GO.01
09 03324 050556      STA 2,GO.02
10 03325 054556      STA 3,GO.S3
11 03326 022561      LDA 0,@GO.K2
12 03327 040561      STA 0,GO.LP
13 03330 022561      LDA 0,@GO.K3
14 03331 040562      STA 0,GO.LA
15 03332 034140      LDA 3,MPSWT
16 03333 175005      MOV 3,3,SNR      ;SKP MAP OPT EXISTS
17 03334 002547      JMP @GO.S3      ;GO BACK TO TEST
18 03335 036141      LDA 3,@ALTB1
19 03336 175005      MOV 3,3,SNR      ;SKP SCRATCH ASSIGN
20 03337 002544      JMP @GO.S3      ;NO SCRATCH RETRN
21 03340 004560      JSR GO.CU      ;CHANGE USERS
22 03341 020154      LDA 0,SCRLO
23 03342 040543      STA 0,GO.SRLO      ;SAVE SCRLO
24 03343 020155      LDA 0,SCRHI
25 03344 040542      STA 0,GO.SHI      ;SAVE SCRHI
26 03345 004440      JSR LDMP2      ;LOAD USER MAP
27 03346 102400      SUB 0,0
28 03347 024531      LDA 1,GO.00
29 03350 125300      MOVS 1,1
30 03351 127120      ADDZL 1,1      ;NEW SCRLO FOR REMAP
31 03352 044154      STA 1,SCRLO
32 03353 044237      STA 1,RELLO
33 03354 006530      GO.L1: JSR @GO.K1      ;GET PHYS ASSIGN
34 03355 000422      JMP GO.GO      ;NO MORE AVAIL
35 03356 030522      LDA 2,GO.00
36 03357 151300      MOVS 2,2
37 03360 153120      ADDZL 2,2
38 03361 147000      ADD 2,1
39 03362 034115      LDA 3,USRMP
40 03363 167000      ADD 3,1
41 03364 066002      ODB 1,2      ;WRITE MAP ENTRY
42 03365 024417      LDA 1,GO.1K
43 03366 147000      ADD 2,1
44 03367 044155      STA 1,SCRHI      ;NEW SCR HI FOR REMAP
45 03370 044240      STA 1,RELHI
46 03371 010507      ISZ GO.00      ;+1 LOG PAGE
47 03372 101400      INC 0,0      ;+1 ALLOCATION POS
48 03373 115300      MOVS 0,3
49 03374 177120      ADDZL 3,3
50 03375 175123      MOVZL 3,3,SNC      ;LAST LOG=37 DONE
51 03376 000756      JMP GO.L1
52 03377 060277      GO.GO: INTOS
53 03400 020145      LDA 0,STATS      ;GET SAVED STATUS
54 03401 061002      DOA 0,2      ;WRITE TO STATUS REGISTER
55 03402 060177      INTEN
56 03403 002477      JMP @GO.02
57 03404 001777      GO.1K: 1777

```

```

10066 N3MRT
01          ;LDMP2=LOAD MAP FOR LEVEL 2
02          ;
03 03405 054420      LDMP2: STA 3,LD.S3      ;SAVE RETURN
04 03406 020115      LDA 0,USRMP
05 03407 062002      ODB 0,2      ;MAP PAGE 0 TO ITSELF
06 03410 102400      SUB 0,0
07 03411 101400      LD.L1: INC 0,0
08 03412 024415      LDA 1,LD.K1      ;LOAD PAGES 1 - 37
09 03413 111300      MOVS 0,2
10 03414 153120      ADDZL 2,2
11 03415 034115      LDA 3,USRMP
12 03416 173000      ADD 3,2
13 03417 133000      ADD 1,2      ;ADD IN 777
14 03420 072002      ODB 2,2      ;VALIDITY PROTECT THIS PAGE
15 03421 030405      LDA 2,LD.37
16 03422 112414      SUB# 0,2,SZR      ;DONE?
17 03423 000766      JMP LD.L1      ;NOT YET
18 03424 002401      JMP @LD.S3      ;RETURN TO CALLER
19 03425 000000      LD.S3: 0
20 03426 000037      LD.37: 37
21 03427 000777      LD.K1: 777

```

```

10067 N3MRT
01          ;ERRRT - 2ND LEVEL ERROR RETURN
02
03 03430 054453 ERRRT: STA 3,GO.S3
04 03431 034140 LDA 3,MPSWT
05 03432 177020 ADDZ 3,3          ;C=1 IF MAP
06 03433 034446 LDA 3,GO.01
07 03434 056455 STA 3,@GO.K3
08 03435 016454 DSZ @GO.K3      ;PREPARE CD.LA FOR ISZ IN CDISP
09 03436 175002 MOV 3,3,SZC
10 03437 000403 JMP .+3
11 03440 036452 LDA 3,@ERRK2   ;GET AC3 AT CALL
12 03441 002440 JMP @GO.01     ;NO MAP OPT GO DRECT
13 03442 040452 STA 0,ERSV0
14 03443 044452 STA 1,,ERSV1
15 03444 050452 STA 2,ERSV2
16 03445 004453 JSR GO.CU      ;CHANGE USERS
17 03446 006451 JSR @ERRK1     ;RELOAD ORIG MAP
18 03447 020445 LDA 0,ERSV0
19 03450 024445 LDA 1,,ERSV1
20 03451 030445 LDA 2,ERSV2
21 03452 036440 LDA 3,@ERRK2   ;GET AC3 AT CALL
22 03453 002430 JMP @GO.S3     ;RETURN TO TEST ERR VIA CDISP
23
24          ;RETN2 - NORMAL 2ND LEVEL END OF TEST RET
25
26 03454 010427 RETN2: ISZ GO.S3      ;+1 RETURN ADDRESS
27 03455 034433 LDA 3,GO.LP
28 03456 056431 STA 3,@GO.K2   ;RESTORE LOGICAL
29 03457 034434 LDA 3,GO.LA     ;PAGE AND ADDR
30 03460 056431 STA 3,@GO.K3   ;OF ORIGINAL GSCRA
31 03461 040433 STA 0,ERSV0
32 03462 044433 STA 1,,ERSV1
33 03463 050433 STA 2,ERSV2
34 03464 034140 LDA 3,MPSWT
35 03465 175005 MOV 3,3,SNR
36 03466 002415 JMP @GO.S3     ;NO MAP GO DIRECT
37 03467 004431 JSR GO.CU      ;CHANGE USERS
38 03470 020415 LDA 0,GO.SLO
39 03471 040154 STA 0,SCRLO
40 03472 020414 LDA 0,GO.SHI
41 03473 040155 STA 0,SCRHI   ;RESTORE SCRLO/HI
42 03474 020420 LDA 0,ERSV0
43 03475 024420 LDA 1,,ERSV1
44 03476 030420 LDA 2,ERSV2
45 03477 002404 JMP @GO.S3     ;RETURN TO 1ST LEVEL TST

```

```

10068 N3MRT
01
02 03500 000000 GO.00: 0
03 03501 000000 GO.01: 0
04 03502 000000 GO.02: 0
05 03503 000000 GO.03: 0
06 03504 002616 GO.K1: GETPA
07 03505 000000 GO.SLO: 0
08 03506 000000 GO.SHI: 0
09 03507 003072 GO.K2: CD.LP
10 03510 000000 GO.LP: 0
11 03511 003071 GO.K3: CD.LA
12 03512 003066 ERRK2: CD.S3
13 03513 000000 GO.LA: 0
14
15 03514 000000 ERSV0: 0
16 03515 000000 ERSV1: 0
17 03516 000000 ERSV2: 0
18 03517 003106 ERRK1: LDMAP
19
20          ;GO.CU- COMPLEMENT TO OTHER USER MAP
21
22 03520 102520 GO.CU: SUBZL 0,0
23 03521 101300 MOVZ 0,0
24 03522 101120 MOVZL 0,0      ;MOVE TO BIT 6
25 03523 024115 LDA 1,USRMP
26 03524 131000 MOV 1,2        ;XOR ACO WITH AC1
27 03525 113520 ANDZL 0,2
28 03526 107000 ADD 0,1
29 03527 146400 SUB 2,1
30 03530 044115 STA 1,USRMP   ;CHANGES USER
31 03531 102520 SUBZL 0,0   ;BIT 15 = USER B
32 03532 024116 LDA 1,USRSE
33 03533 131000 MOV 1,2
34 03534 113520 ANDZL 0,2
35 03535 107000 ADD 0,1
36 03536 146400 SUB 2,1
37 03537 044116 STA 1,USRSE   ;CHANGE USER STATUS BIT
38 03540 020145 LDA 0,STATS  ;GET STATUS WORD
39 03541 101220 MOVZ 0,0
40 03542 101120 MOVZL 0,0   ;CHANGE USER BIT
41 03543 123000 ADD 1,0
42 03544 040145 STA 0,STATS
43 03545 001400 JMP 0,3      ;RETURN TO CALLER

```

10069 N3MRT

```
01 ;SETLP - SET UP LOOP CALL HANDLER
02 ;PERFORMS SAME FUNCTION AS SETUP IN NORMAL TSTS
03 ;ENTERED VIA JSR @SETUL
04
05 03546 040437 SETLP: STA 0,ST.S0
06 03547 044437 STA 1,ST.S1
07 03550 024437 LDA 1,ST.K1
08 03551 136414 SUB# 1,3,SZR
09 03552 000412 JMP STNMP ;NOT AN LCALL
10 03553 022435 LDA 0,@ST.K2 ;GET LOG AORS
11 03554 040123 STA 0,ST.LA ;FOR LOOP
12 03555 026434 LDA 1,@ST.K3 ;AND LOG PAGE
13 03556 044122 STA 1,ST.LP
14 03557 020433 SETXI: LDA 0,ST.LK ;-4
15 03560 040121 STA 0,ST.LC ;FOR LOOP RPT COUNT
16 03561 020424 LDA 0,ST.S0
17 03562 024424 LDA 1,ST.S1
18 03563 001400 JMP 0,3
19 ;LOOP SETUP WAS NOT VIA LCALL
20
21 03564 054123 STNMP: STA 3,ST.LA
22 03565 000772 JMP SETXI
23
24 ;LOOPL - PERFORMS SAME FUNCTION AS LOOP
25 ;ENTERED VIA JSR @LLOOP
26
27 03566 010121 LOOPL: ISZ ST.LC ;SKIP IS FINI LOOP
28 03567 101001 MOV 0,0,SKP ;LOOP BACK
29 03570 001400 JMP 0,3 ;CONTINUE ON
30 03571 040414 STA 0,ST.S0
31 03572 020415 LDA 0,ST.K1 ;CHK FOR
32 03573 116415 SUB# 0,3,SNR ;SUPER CALL
33 03574 000403 JMP .+3 ;SUPER CALL
34 03575 020410 LDA 0,ST.S0 ;NOT LCALL
35 03576 002123 JMP @ST.LA ;JUST CONTINUE
36 03577 020123 LDA 0,ST.LA ;LOGICAL START LOOP
37 03600 042410 STA 0,@ST.K2
38 03601 020122 LDA 0,ST.LP ;IN LOGICAL PAGE
39 03602 042407 STA 0,@ST.K3
40 03603 020402 LDA 0,ST.S0
41 03604 001400 JMP 0,3
42
43 03605 000000 ST.S0: 0
44 03606 000000 ST.S1: 0
45 03607 003047 ST.K1: CD.EX+1
46 03610 003071 ST.K2: CD.LA
47 03611 003072 ST.K3: CD.LP
48 03612 177774 ST.LK: -4.
```

10070 N3MRT

```
01 ;ERROH - ERROR HANDLER - PRINT ALL ERR INFO
02 03613 000010 8.
03 ;FIRST PRINT PRG# AND (AC'S)
04 03614 040572 ERROH: STA 0,ER.S0
05 03615 020776 LDA 0,ERROH-1
06 03616 044571 STA 1,ER.S1
07 03617 050571 STA 2,ER.S2
08 03620 054571 STA 3,ER.S3
09 03621 030142 LDA 2,PSTRT
10 03622 011376 ISZ -2,2
11 03623 113000 ADD 0,2
12 03624 050426 STA 2,ERTIT
13 03625 101000 MOV 0,0
14 03626 010151 ISZ ERTOT ;+1#ERROR CALLS
15 03627 101000 MOV 0,0
16 03630 020151 LDA 0,ERTOT
17 03631 024552 LDA 1,ER50.
18 03632 122414 SUB# 1,0,SZR ;HALT AFTER FIRST 50 ERRORS
19 03633 000404 JMP .+4
20 03634 006132 JSR @LMESS
21 03635 007235 FTYTX
22 03636 006554 JSR @IWAIT
23 03637 102400 SUB 0,0
24 03640 040150 STA 0,TIMSW ;SO TIME TYPE WILL FOLLOW
25 03641 034230 LDA 3,SWREG
26 03642 101100 MOVL 0,0
27 03643 103102 ADDL 0,0,SZC ;SW2=1 NO TYPE
28 03644 000550 JMP EREXI
29 03645 006132 JSR @LMESS
30 03646 006254 TXT.0
31 03647 024120 LDA 1,CURPR ;GET PROG #
32 03650 006071 JSR @LZOCT ;PRINT IT
33 03651 006132 JSR @LMESS
34 03652 006263 ERTIT: TXT.1 ;PRINTS TEST TITL
35 03653 006132 JSR @LMESS
36 03654 006263 TXT.1
37 03655 024531 LDA 1,ER.S0 ;PRINT AC'S
38 03656 006133 JSR @LPOCT ;AT ERROR CALL
39 03657 024530 LDA 1,ER.S1
40 03660 006133 JSR @LPOCT
41 03661 024527 LDA 1,ER.S2
42 03662 006133 JSR @LPOCT
```

```

10071 N3MRT
01          :PRINT MEM ALLOCATION ASSIGNMENTS
02 03663 006132  ERMPP: JSR @LMESS
03 03664 006267          TXT.2
04 03665 024154          LDA 1,SCRLO
05 03666 006133          JSR @LPOCT          :PRINT SCRATCH LIMITS
06 03667 024155          LDA 1,SCRHI
07 03670 006133          JSR @LPOCT
08 03671 024140          LDA 1,MPSWT          :GET MAP EXIST
09 03672 125005          MOV 1,1,SNR          :TYPE MEM ALLOC IF MAP
10 03673 000521          JMP EREXI          :NO MAP FORGET REST OF TYPE
11 03674 020237          LDA 0,RELLO
12 03675 100015          COM# 0,0,SNR          :SKP=RELOCATED TEST
13 03676 000407          JMP NOTRL
14 03677 006132          JSR @LMESS          :PRINT RELLO/HI
15 03700 006437          TXT.B
16 03701 024237          LDA 1,RELLO
17 03702 006133          JSR @LPOCT
18 03703 024240          LDA 1,RELHI
19 03704 006133          JSR @LPOCT
20          NOTRL:
21 03705 020116          LDA 0,USRSE
22 03706 101004          MOV 0,0,SZR
23 03707 000404          JMP .+4          :USER B
24 03710 006132          JSR @LMESS
25 03711 006341          TXT.5
26 03712 000403          JMP .+3
27 03713 006132          JSR @LMESS
28 03714 006345          TXT.6
29 03715 020157          LDA 0,DCHHI
30 03716 101005          MOV 0,0,SNR
31 03717 000407          JMP ERNDC          :DON'T PRINT DCH LOC
32 03720 006132          JSR @LMESS
33 03721 006275          TXT.3
34 03722 024156          LDA 1,DCHLO          :PRINT DCH LIMITS
35 03723 006133          JSR @LPOCT
36 03724 024157          LDA 1,DCHHI
37 03725 006133          JSR @LPOCT
38 03726 020061  ERNDC: LDA 0,DLTBL
39 03727 101004          MOV 0,0,SZR          :SKP IS PRINT ALLOCATION TABLE
40 03730 000464          JMP EREXI
41 03731 020155          LDA 0,SCRHI          :SKP=MEM ALLOCATED TO THIS TEST
42 03732 101005          MOV 0,0,SNR
43 03733 000461          JMP EREXI
44 03734 032141          LDA 2,@ALTBL
45 03735 050447          STA 2,ER.C2          :SAVE # 1K'S ALLOCATED
46 03736 102400          SUB 0,0
47 03737 040442          STA 0,ER.C1          :CLR COUNTER
48 03740 006132          JSR @LMESS
49 03741 006302          TXT.4
50 03742 024237          LDA 1,RELLO
51 03743 124014          COM# 1,1,SZR          :SKP=NOT RELOCATED TEST
52 03744 000404          JMP .+4
53 03745 006132          JSR @LMESS
54 03746 006375          TXT.9
55 03747 000403          JMP ERMPL
56 03750 006132          JSR @LMESS
57 03751 006413          TXT.A
58 03752 020427  ERMPL: LDA 0,ER.C1          :PRINT PHYS PAGES
59 03753 006427          JSR @ER.K4          :ALSO IN MODULO 1K
60 03754 000440          JMP EREXI

```

```

0072 N3MRT
01 03755 101213          MOVR# 0,0,SNC
02 03756 006070          JSR @PCRLF
03 03757 006071          JSR @LZOCT
04 03760 020421          LDA 0,ER.C1
05 03761 103120          ADDZL 0,0          :FORM LOGICAL ADDRESS
06 03762 101300          MOVS 0,0
07 03763 024154          LDA 1,SCRLO
08 03764 107000          ADD 0,1
09 03765 040420          STA 0,ERSSO          :SAVE DISPLACEMENT FROM SCRLO
10 03766 006133          JSR @LPOCT
11 03767 024237          LDA 1,RELLO
12 03770 124015          COM# 1,1,SNR          :SKP=WANT TO PRINT RELOCATED ADDRESS
13 03771 000404          JMP .+4
14 03772 020413          LDA 0,ERSSO
15 03773 107000          ADD 0,1
16 03774 006133          JSR @LPOCT          :RELLO+ENTRY # EXPRESSED AS LOGICAL PAGE
17 03775 010404          ISZ ER.C1
18 03776 014406          DSZ ER.C2
19 03777 000753          JMP ERMPL
20 04000 000414          JMP EREXI
21 04001 000000  ER.C1: 0
22 04002 002616  ER.K4: GETPA
23 04003 000062  ER.S0.: 50.
24 04004 000000  ER.C2: 0
25 04005 000000  ER.SS0: 0
26 04006 000000  ER.S0: 0
27 04007 000000  ER.S1: 0
28 04010 000000  ER.S2: 0
29 04011 000000  ER.S3: 0
30 04012 004065  IWAIT: KEY6W

```

```

10073 N3MRT
01 04013 001000      186
02 04014 020230  EREXI: LDA 0,SWREG
03 04015 024776      LDA 1,EREXI-1      :GET 186
04 04016 123414      AND# 1,0,SZR        :SKP=NOT ERROR WAIT
05 04017 004446      JSR KEY6W          :WAIT FOR TTI INPUT
06 04020 103102      ADDL 0,0,SZC       :C=0 IS ERROR RELEASE
07 04021 010770      ISZ ER,S3
08 04022 020764  ERXXT: LDA 0,ER,S0
09 04023 024764      LDA 1,ER,S1
10 04024 030764      LDA 2,ER,S2
11 04025 034764      LDA 3,ER,S3
12 04026 001400      JMP 0,3
13
14
15          :2ND OR FOLLOWING CALLS PRINT
16          :ACO 1 AND 2 - USED FOR TYPEDUT
17          :EXPANSION BY INDIVIDUAL TESTS
18
19 04027 040757  ERROE: STA 0,ER,S0
20 04030 044757      STA 1,ER,S1
21 04031 050757      STA 2,ER,S2
22 04032 054757      STA 3,ER,S3
23 04033 020230      LDA 0,SWREG
24 04034 101100      MOVL 0,0
25 04035 103102      ADDL 0,0,SZC       :CHECK BIT 2 TO SEE IF WANT PRINT
26 04036 000756      JMP EREXI
27 04037 006070      JSR @LCRLF
28 04040 024746      LDA 1,ER,S0
29 04041 006133      JSR @LPOCT
30 04042 024745      LDA 1,ER,S1
31 04043 006133      JSR @LPOCT
32 04044 024744      LDA 1,ER,S2
33 04045 006133      JSR @LPOCT
34 04046 000746      JMP EREXI
35
36          :TEXT CALL ADRS OF TEXT IS IN ACO
37          :CALL MUST ONLY BE MADE WHILE IN FIRST LEVEL TEST
38 04047 040737  ERTXT: STA 0,ER,S0
39 04050 040411      STA 0,ER,TP
40 04051 044736      STA 1,ER,S1
41 04052 050736      STA 2,ER,S2
42 04053 054736      STA 3,ER,S3
43 04054 020230      LDA 0,SWREG
44 04055 101100      MOVL 0,0
45 04056 103102      ADDL 0,0,SZC       :SKP IS OK TO TYPE
46 04057 000743      JMP ERXXT          :EXIT TYPE DELETED
47 04060 006132      JSR @LMESS
48 04061 000000  ER.TP: 0          :TEXT ADRS STORED HERE
49 04062 000740      JMP ERXXT
50 04063 000000      0          :TO SAVE ACO
51 04064 000003      3          :FOR MASKO

```

```

10074 N3MRT
01 04065 040427  KEY6W: STA 0,KEY.0
02 04066 054427      STA 3,KEY.3
03 04067 006132      JSR @LMESS
04 04070 007244      KEY6T          :TYPE WAIT MESS
05 04071 020153  KEYLP: LDA 0,LASTI      :GET TTI INPUTTED CHARACTER
06 04072 101112      MOVL# 0,0,SZC    :WAIT FOR TYPE IN
07 04073 000406      JMP .+6
08 04074 063610      SKPDN TTI
09 04075 000774      JMP .-4
10 04076 060610      DIAC 0,TTI
11 04077 103240      ADDOR 0,0
12 04100 040153      STA 0,LASTI      :CLR BIT 0
13 04101 034412      LDA 3,TTOXS+1    :=177 FOR MASK
14 04102 163400      AND 3,0          :MASK OFF PARITY BIT
15 04103 034407      LDA 3,TTOXS      :CK FOR SWREG REQUEST
16 04104 116414      SUB# 0,3,SZR
17 04105 000411      JMP KEY.C
18 04106 102400      SUB 0,0
19 04107 040153      STA 0,LASTI
20 04110 020404      LDA 0,KEY.0
21 04111 002404      JMP @KEY.3
22 04112 000015  TTOXS: 15
23 04113 000177      177
24 04114 000000  KEY.0: 0
25 04115 000000  KEY.3: 0
26 04116 006402  KEY.C: JSR @KY.CK
27 04117 000752      JMP KEYLP
28 04120 001065  KY.CK: CKOOT

```

10075 N3MRT

```
01 ;EPADR = PRINT THE FOLLOWING ADDRESSES
02 ;PERTINENT TO THE CURRENT CALL
03 ;(AC0)=AC3 AT THE LAST ERROR CALL OR TRAP PC
04 ;(AC1)=LOGICAL START OF PROGRAM (IN SCRATCH)
05 ;(AC2)=PHYSICAL START RESIDENT TEST
06 ;(ST.LA)=LOGICAL START OF LAST LOOP
07 ;(ST.LA)-(AC1)+(AC2)=START LOOP IN RESIDENT
08 ;(AC0)-(AC1)+(AC2)"MAYBE"= ADRS OF ERROR
09
10 04121 040665 EPADR: STA 0,ER.S0
11 04122 044665 STA 1,ER.S1
12 04123 050665 STA 2,ER.S2
13 04124 054665 STA 3,ER.S3
14 04125 006132 JSR @LMESS ;PRINT HEADER
15 04126 006351 TXT.7
16 04127 024123 LDA 1,ST.LA
17 04130 006133 JSR @LPOCT ;PRINT LOG. START OF LOOP
18 04131 030656 LDA 2,ER.S1
19 04132 146500 SUBL 2,1
20 04133 125220 MOVZR 1,1 ;GET RID OF CRY
21 04134 020654 LDA 0,ER.S2 ;PHYS STRT OF RESIDENT
22 04135 107000 ADD 0,1 ;CREATE PHYS STRT LOOP
23 04136 006133 JSR @LPOCT
24 04137 024647 LDA 1,ER.S0
25 04140 146500 SUBL 2,1 ;CREATE PHYSICAL ERR ADRS
26 04141 125220 MOVZR 1,1
27 04142 020646 LDA 0,ER.S2
28 04143 107000 ADD 0,1
29 04144 006133 JSR @LPOCT
30 04145 002426 JMP @EP.RT
31
32
33 ;PRINT THE LAST THREE RANDOM #'S GENERATED BY FRANG
34 ;THESE WERE IN AC0 AC1 AND AC2 RESPECTIVELY
35 04146 040421 EPACS: STA 0,EP.0
36 04147 044421 STA 1,EP.1
37 04150 050421 STA 2,EP.2
38 04151 054421 STA 3,EP.3
39 04152 006132 JSR @LMESS
40 04153 006367 TXT.8
41 04154 024124 LDA 1,RNAC0
42 04155 006133 JSR @LPOCT
43 04156 024125 LDA 1,RNAC1
44 04157 006133 JSR @LPOCT
45 04160 024126 LDA 1,RNAC2
46 04161 006133 JSR @LPOCT
47 04162 020405 LDA 0,EP.0
48 04163 024405 LDA 1,EP.1
49 04164 030405 LDA 2,EP.2
50 04165 034405 LDA 3,EP.3
51 04166 001400 JMP 0,3
52 04167 000000 EP.0: 0
53 04170 000000 EP.1: 0
54 04171 000000 EP.2: 0
55 04172 000000 EP.3: 0
56 04173 004014 EP.RT: EREXI
```

10076 N3MRT

```
01 ;*****
02 ;ODT=OCTAL EDITOR
03 ;*****
04 04174 000136 N136: 136
05 04175 000075 N75: 75
06 04176 004420 IMAPS: MAP.S
07 04177 000007 N7: 7
08 04200 020777 RUBOUT: LDA 0,N7
09 04201 123420 ANDZ 1,0
10 04202 034570 LDA 3,N60
11 04203 163000 ADD 3,0 ;AC0=LAST DIGIT TYPED
12 04204 004575 JSR SHIFT ;ECHO AND ERASE THE CHARACTER
13 ;BEING RUBBED OUT
14 04205 000433 JMP OPRIN-1
15
16 04206 044214 ODT: STA 1,SAV1 ;SAVE THE ACCUMULATORS
17 04207 040213 STA 0,SAV0
18 04210 050215 STA 2,SAV2
19 04211 054216 STA 3,SAV3
20 04212 101200 MOVR 0,0 ;SAVE THE CARRY
21 04213 040217 STA 0,SAVCR
22 04214 102620 SUBZR 0,0
23 04215 101400 INC 0,0
24 04216 040152 STA 0,EACTV ;PREPARE TTI/TTO FLAG
25 04217 040220 STA 0,OP.EN
26 04220 030221 LDA 2,L0PNL
27 04221 102400 CLRWT: SUB 0,0
28 04222 040220 STA 0,OP.EN
29 04223 042753 STA 0,@IMAPS ;CLR MAP SWITCH
30 04224 006476 WAIT: JSR @CR.LF ;TYPE CR,LF
31 04225 020220 LDA 0,OP.EN
32 04226 100014 COM# 0,0,SZR
33 04227 000405 JMP WAITX
34 04230 034570 LDA 3,MAP.S
35 04231 175004 MOV 3,3,SZR ;SKP IS DON'T USE MAP
36 04232 060302 NIOP MAP ;SINGLE REFERENCE
37 04233 025000 LDA 1,0,2 ;GET CONTENTS OF LOC
38 04234 176440 WAITX: SUBO 3,3 ;AC3=0
39 04235 054561 STA 3,TEMP
40 04236 176000 ADC 3,3 ;AC3 = 177777
41 04237 054555 STA 3,SIGN
42 04240 054555 STA 3,STRAC2
```



```

10077 N3MRT
01 04241 020153 OPRIN: LDA 0, LASTI
02 04242 101112 MOVL# 0,0,SZC ;SKP IS NO INPUT YET
03 04243 000404 JMP .+4
04 04244 063610 SKPDN TTI ;ALSO CHECK DONE
05 04245 000774 JMP .-4 ;INCASE ION IS OFF
06 04246 060610 DIAC 0,TTI ;GET CHAR IF DONE = 1
07 04247 034455 LDA 3,N177
08 04250 163400 AND 3,0 ;REMOVE PARITY BIT
09 04251 040153 STA 0, LASTI ;SAVE CHAR FOR LATER
10 04252 116415 SUB# 0,3,SNR ;SKIP IF ACO IS NOT 177
11 04253 000725 JMP RUBOUT ;OTHERWISE GO TO RUBOUT
12 04254 034440 LDA 3,N67 ;AC3=67
13 04255 116452 SUBO# 0,3,SZC ;IF THE ASCII VALUE IS HIGHER THAN
14 04256 000404 JMP WHERE ;67 THEN GO TO "WHERE"
15 04257 034573 LDA 3,N57 ;AC3 = 57
16 04260 116442 SUBO 0,3,SZC ;SKIP IF ACO IS LESS THAN 57
17 04261 000423 JMP DIGIT ;OTHERWISE GO TO "DIGIT"
18 04262 010532 WHERE: ISZ SIGN ;SKIP IF THE PREVIOUS SIGN WAS
19 ;A "+"
20 04263 124400 NEG 1,1 ;NEGATE IF IT WAS "-"
21 04264 034532 LDA 3,TEMP
22 04265 167000 ADD 3,1
23 04266 044530 STA 1,TEMP ;STORE THE NEW VALUE
24 04267 034564 LDA 3,N12
25 04270 116415 SUB# 0,3,SNR ;DON'T ECHO A LINE FEED
26 04271 000521 JMP JPNXT
27 04272 034424 LDA 3,X17
28 04273 116415 SUB# 0,3,SNR ;CONTROL "D" ?
29 04274 000730 JMP WAIT
30 04275 034426 LDA 3,N15
31 04276 116415 SUB# 0,3,SNR ;CHECK FOR A "CR"
32 04277 000513 JMP JPNXT ;CLOSE THE OPEN LOC.
33 04300 034415 LDA 3,N40
34 04301 116432 SUBZ# 0,3,SZC ;SKP IF > 40
35 04302 000474 JMP WHAT ;"?
36 04303 000422 JMP CH.R
37 04304 010511 DIGIT: ISZ STRAC2
38 04305 125121 MOVZL 1,1,SKP
39 04306 126440 SUBO 1,1
40 04307 125120 MOVZL 1,1
41 04310 125120 MOVZL 1,1
42 04311 166000 ADC 3,1 ;SHIFT AC1 TO LEFT BY 3 PLACES
43 ;AC1=OCTAL WORD BEING INPUTED BY
44 04312 004473 JSR TPCHR ;THE OPERATOR
45 04313 000726 JMP OPRIN ;ECHO EVERY CHARACTER

```

```

10078 N3MRT
01 04314 000067 N67: 67
02 04315 000040 N40: 40
03 04316 000017 X17: 17
04 04317 000705 JXWT: JMP WAIT
05 04320 000124 N124: 124
06 04321 000101 N101: 101
07 04322 004537 CR.LF: CRLFX
08 04323 000015 N15: 15
09 04324 000177 N177: 177
10 04325 034647 CH.R: LDA 3,N136
11 04326 116405 SUB 0,3,SNR ;SKP IF NOT "-"
12 04327 000535 JMP NXTLOC ;OPEN PREVIOUS LOC
13 04330 004455 JSR TPCHR
14 04331 034644 LDA 3,N75
15 04332 116405 SUB 0,3,SNR
16 04333 002525 JMP @EQUALS ;PRINT CURRENT ARGUMENT(AC1)
17 04334 034520 LDA 3,N121 ;AC3=121
18 04335 162015 ADC# 3,0,SNR
19 04336 002520 JMP @IRUN ;IF IT WAS A "R" THEN START THE
20 ;USERS PROGRAM
21 04337 116005 ADC 0,3,SNR
22 04340 002517 JMP @IPKCD ;IF IT IS A "P" THEN PROCEED
23 04341 034140 LDA 3,MPSWT
24 04342 175005 MOV 3,3,SNR ;SKP IS CAN USE MAP
25 04343 000421 JMP NOMAP
26 04344 034511 LDA 3,N115
27 04345 116415 SUB# 0,3,SNR ;SKP IS NOT AN "M"
28 04346 000461 JMP MAPIT
29 04347 034752 LDA 3,N101
30 04350 116405 SUB 0,3,SNR ;SKP IS NOT AN "A"
31 04351 000460 JMP MAP.A
32 04352 175405 INC 3,3,SNR ;SKP IS NOT A "B"
33 04353 000457 JMP MAP.B
34 04354 034744 LDA 3,N124
35 04355 116415 SUB# 0,3,SNR ;SKP IS NOT A "T"
36 04356 000443 JMP MAP.T
37 04357 175415 INC# 3,3,SNR ;SKP IS NOT "U"
38 04360 000444 JMP UNMAP
39 04361 034573 LDA 3,N105
40 04362 116415 SUB# 0,3,SNR ;IS IT AN "E"
41 04363 000476 JMP MAP.E ;SKP IS NOPE
42 04364 034466 NOMAP: LDA 3,N57 ;IT WAS AN "E"
43 04365 162405 SUB 3,0,SNR ;AC3 = 57
44 04366 000514 JMP OPNLOC ;SKIP IF ACO IS NOT "/"
45 04367 145000 MOV 2,1
46 04370 101405 INC 0,0,SNR ;SKIP IF IT IS NOT A "."
47 04371 000643 JMP WAITX
48 04372 126460 N60: SUBC 1,1
49
50 04373 115644 INCOR 0,3,SZR
51 04374 161665 INCCR 3,0,SNR
52 04375 000642 JMP OPRIN-2
53 04376 020554 WHAT: LDA 0,N77 ;BITS 8-15 OF ACO=77
54 04377 004406 JSR TPCHR ;TYPE IT"?"
55 04400 000624 JMP WAIT

```

## 10079 N3MRT

```

01 04401 125220 SHIFT: MOVZR 1,1
02 04402 125220 MOVZR 1,1
03 04403 125221 MOVZR 1,1,SKP
04 04404 020711 LDA 0,N40 ;PREPARE TO TYPE A SPACE
05 04405 061111 TPCHR: DOAS 0,TTO ;TYPE AC0
06 04406 063511 SKPBZ TTO ;WAIT FOR THE PRINTER
07 04407 000777 JMP -1
08 04410 060211 NIOC TTO ;CLR DNE FOR TTO
09 04411 001400 JMP 0,3 ;RETURN TO CALLER
10 04412 000452 JPNXT: JMP NXTLOC
11 04413 000000 TPRET: 0
12 04414 000000 SIGN: 0
13 04415 000000 STRAC2: 0
14 04416 000000 TEMP: 0
15 04417 000000 SUBRET: 0
16 04420 000000 MAP.S: 0
17 04421 004562 MAP.T: JSR MPPRT ;PRINT MAP ENTRY TABLES
18 04422 002401 JMP @,+1
19 04423 004221 CLRWT
20 04424 176400 UNMAP: SUB 3,3
21 04425 054773 STA 3,MAP.S ;SET SWITCH
22 04426 000671 JMP JXWT ;WAIT FOR OPERATOR INPUT
23 04427 074402 MAPIT: DIA 3,MAP
24 04430 000411 JMP MAPBB
25 04431 126401 MAP.A: SUB 1,1,SKP
26 04432 126000 MAP.B: ADC 1,1
27 04433 074402 DIA 3,2 ;GET CURRENT STATUS
28 04434 020514 LDA 0,K40 ;USER B BIT
29 04435 100000 COM 0,0
30 04436 117400 AND 0,3
31 04437 125004 MOV 1,1,SZR ;SKP=USER "A"
32 04440 116000 ADC 0,3 ;OR IN BIT 10
33 04441 063477 MAPBB: SKPBN CPU ;SKP=ION
34 04442 175041 MOVD 3,3,SKP
35 04443 060277 INTDS
36 04444 075002 DOA 3,MAP
37 04445 036001 LDA 3,@1
38 04446 175003 MOV 3,3,SNC ;SKP IF INTRUPTS WEREN'T ON
39 04447 060177 INTEN
40 04450 176000 ADC 3,3
41 04451 000754 JMP UNMAP+1
42 04452 000057 N57: 57
43 04453 000012 N12: 12
44 04454 000121 N121: 121
45 04455 000115 N115: 115
46 04456 004555 IRUN: RUN
47 04457 004557 IPRCD: PRCD
48 04460 004575 IQUALS: EQUALS
49 04461 176000 MAP.E: ADC 3,3
50 04462 054465 STA 3,ENT.F
51 04463 000736 JMP MAP.T

```

## 10080 N3MRT

```

01 04464 010220 NXTLOC: ISZ OP.EN
02 04465 000405 JMP NTOPN ;LOC WASN'T OPENED
03 04466 020732 LDA 0,MAP.S
04 04467 101004 MOV 0,0,SZR ;SKP IF NOT TO USE MAP
05 04470 060302 NIOP MAP ;SINGLE REFERENCE
06 04471 045000 STA 1,0,2 ;RESTORE THE OPEN LOCATION
07 04472 175232 NTOPN: MOVZR# 3,3,SZC ;SKIP IF IT WAS NOT A "CR"
08 04473 000624 JMP JXWT
09 04474 145400 INC 2,1 ;IF IT IS A LINE FEED THEN ADD
10 ;1 TO AC2
11 04475 175004 MOV 3,3,SZR ;SKP WAS A "
12 04476 000403 JMP ,+3
13 04477 166000 ADC 3,1
14 04500 166000 ADC 3,1
15 04501 004436 JSR CRLFX ;TYPE A CR, LF
16 04502 152000 OPNLOC: ADC 2,2 ;AC2 = 177777
17 04503 050220 STA 2,OP.EN ;STORE THE FLAG FOR OPEN LOC.
18 04504 151220 MOVZR 2,2
19 04505 147400 AND 2,1 ;NEGLECT ADDRESS BIT 0
20 04506 044221 STA 1,LOPNL ;SAVE ADRS LAST OPEN'D
21 04507 004410 JSR POCTX-1 ;TYPE AC1
22 04510 034710 LDA 3,MAP.S
23 04511 175004 MOV 3,3,SZR ;SKP IS NOT TO USE MAP
24 04512 060302 NIOP MAP
25 04513 025000 LDA 1,0,2
26 04514 004404 JSR POCTX ;TYPE THE OPEN LOCATION
27 04515 002401 JMP @,+1
28 04516 004225 WAIT+1 ;WAIT FOR THE OPERATOR INPUT
29
30 04517 131000 MOV 1,2
31 04520 054677 POCTX: STA 3,SUBRET ;SAVE THE RETURN ADDRESS
32 04521 050674 STA 2,STRAC2 ;SAVE AC2
33 04522 131000 MOV 1,2
34 04523 126621 SUBZR 1,1,SKP ;AC1=100000
35 04524 132401 SUB 1,2,SKP
36 04525 020725 LDA 0,N57 ;AC0=57
37 04526 101400 INC 0,0
38 04527 132453 FRMDGT: SUBO# 1,2,SNC ;IF AC2 IS LESS THAN AC1 THEN
39 ;GO TO PRINT THE DIGIT
40 04530 000774 JMP -4
41
42 04531 004650 PRIDGT: JSR SHIFT
43 04532 125004 MOV 1,1,SZR
44 04533 000772 JMP FRMDGT-2
45 04534 004650 JSR TPCHR-1 ;TYPE SPACES
46 04535 030660 LDA 2,STRAC2 ;RESTORE AC2
47 04536 002661 JMP @SUBRET

```

```

10081 N3MRT
01 04537 054660 CRLFX: STA 3,SUBRET ;SAVE THE RETURN ADDRESS
02 04540 020413 LDA 0,X15 ;ACO=15
03 04541 004644 JSR TPCHR ;TYPE A "CR"
04 04542 020711 LDA 0,N12 ;ACO=12
05 04543 004642 JSR TPCHR ;TYPE A LINE FEED
06 04544 020405 LDA 0,N100 ;BITS 8-15 OF ACO=100
07 04545 004640 JSR TPCHR ;TYPE A "a"
08 04546 002651 JMP @SUBRET
09 04547 000000 ENT.F: 0
10 04550 000040 K40: 40
11 04551 000100 N100: 100
12 04552 000077 N77: 77
13 04553 000015 X15: 15
14 04554 000105 N105: 105
15
16 04555 062677 RUN: IORST
17 04556 044216 STA 1,SAV3
18 04557 102400 PRCD: SUB 0,0
19 04560 040152 STA 0,EACTV ;RESET TTI/TTO FLAG
20 04561 040637 STA 0,MAP.S
21 04562 020213 LDA 0,SAV0
22 04563 024214 LDA 1,SAV1
23 04564 030215 LDA 2,SAV2
24 04565 036407 LDA 3,@EQUALS-1
25 04566 177005 ADD 3,3,SNR ;SKP=TTY DELETED
26 04567 056405 STA 3,@EQUALS-1 ;CLEAR INTR WAIT
27 04570 034217 LDA 3,SAVCR
28 04571 175100 MOVL 3,3
29 04572 034216 LDA 3,SAV3
30 04573 002216 JMP @SAV3 ;RETURN TO PGM
31
32
33
34 04574 014325 TT.00+2 ;TTY TEST WAIT SWITCH
35 04575 044405 EQUALS: STA 1,..+5
36 04576 004722 JSR POCTX
37 04577 024403 LDA 1,..+3
38 04600 002401 JMP @.+1
39 04601 004234 WAITX
40 04602 000000 0

```

```

10082 N3MRT
01 ;THIS SUBROUTINE PRINTS CONTENTS OF MAP TABLES
02 ;TWO ODT COMMANDS ACCESS THIS SUBROUTINE
03 ; E PRINT ENTRY
04 ; T PRINT ALL ENTRIES
05 04603 054445 MPPRT: STA 3,MP.S3
06 04604 006132 JSR @LMESS
07 04605 004664 MPTXT
08 04606 010741 ISZ ENT.F ;SKP="E" COMMAND
09 04607 000407 JMP MP.LP-1
10 04610 014737 DSZ ENT.F ;RESTORE FLG
11 04611 012436 ISZ @GP.K1 ;NO SKP=ENTERED ADDR.
12 04612 000402 JMP .+2
13 04613 024447 LDA 1,GP.S2 ;SAVED AC2
14 04614 030445 LDA 2,GP.76
15 04615 133401 AND 1,2,SKP ;AC2=ADDRESS OF PAGE TO PRINT
16 04616 152400 SUB 2,2
17 04617 145000 MP.LP: MOV 2,1
18 04620 006133 JSR @LPOCT ;PRINT LOGICAL PAGE #
19 04621 141000 MOV 2,0
20 04622 004427 JSR GPRMP ;PRINT USER=A ENTRY
21 04623 020433 LDA 0,GP.B6
22 04624 143000 ADD 2,0
23 04625 004424 JSR GPRMP ;PRINT USER=B ENTRY
24 04626 102620 SUBZR 0,0
25 04627 143000 ADD 2,0
26 04630 004421 JSR GPRMP ;PRINT DCH=A ENTRY
27 04631 020426 LDA 0,GP.B06
28 04632 143000 ADD 2,0
29 04633 004416 JSR GPRMP ;PRINT DCH=B ENTRY
30 04634 020424 LDA 0,GP.B5
31 04635 113000 ADD 0,2
32 04636 151132 MOVZL# 2,2,SZC ; NO SKP IS DONE ALL ENTRIES
33 04637 000405 JMP MPRTN
34 04640 050422 STA 2,GP.S2
35 04641 006070 JSR @PCRLF
36 04642 010705 ISZ ENT.F
37 04643 000754 JMP MP.LP ;CONTINUE WITH NEXT ENTRY
38 04644 102400 MPRTN: SUB 0,0
39 04645 040702 STA 0,ENT.F
40 04646 002402 JMP @MP.S3
41 04647 004415 GP.K1: STRAC2
42 04650 000000 MP.S3: 0
43 04651 054412 GPRMP: STA 3,GP.S3
44 04652 061003 DOA 0,3
45 04653 064403 DIA 1,3 ;GET ENTRY
46 04654 006133 JSR @LPOCT ;PRINT IT
47 04655 002406 JMP @GP.S3 ;RETURN
48 04656 001000 GP.B6: 186
49 04657 101000 GP.B06: 180+186
50 04660 002000 GP.B5: 185
51 04661 076000 GP.76: 76000
52 04662 000000 GP.S2: 0
53 04663 000000 GP.S3: 0
54 04664 005215 MPTXT: .TXTE 1<15><12>MAP ENTRY TABLE<15><12>LOGICAL USER-A<11>
55 04705 051525 USER-B<11>DCH-A<11>DCH-B<15><12>I

```

10083 N3MRT

```
01
02 ;TELETYPE INTERRUPT PACKAGE
03 ;AC1,AC2 SAVED
04 ;"MES?" PRINTS ASCII MESSAGES AS SPECIFIED BY ASSEMBLER
05 ;"CLF?" PRINTS A CARRIAGE RETURN
06 ;"POC?" PRINTS C(1) IN OCTAL
07 ;"ZOC?" PRINTS C(1) IN OCTAL, LEADING ZEROS SUPPRESSED
08 ;"PDE?" PRINTS C(1) IN DECIMAL, LEADING ZEROS SUPPRESSED,
09 ;THE ABOVE THREE ARE FOLLOWED BY A TAB UNLESS LOCATION PTB? IS
10 ;ALTERED IN WHICH CASE CONTENTS OF PTB? WILL BE PRINTED AFTER
11 ;THE NUMBER.
12 ;"TIO?" ACCEPTS OCTAL, AND
13 ;"TID?" ACCEPTS DECIMAL SINGLE PRECISION SIGNED INTEGERS
14 ;INTO AC1 FROM THE TTI. LEADING NULLS, TABS,
15 ;AND SPACES ARE IGNORED. A 16 BIT UNSIGNED INTEGER IS
16 ;FORMED, THEN NEGATED IF A MINUS SIGN IS TYPED.
17 ;EXIT AT CALL+1 IF INPUT ERROR WITH ACO=BAD CHARACTER.
18 ; (NOT A LEGAL DIGIT OR TERMINATING CHARACTER)
19 ;EXIT AT CALL+2 UPON TERMINATING CHARACTER
20 ; WITH ACO=0, 0, 40, 12, 55
21 ; FOR NULL, TAB, SPACE, CARRIAGE RETURN, COMMA
22 ;THE ABOVE WAIT FOR TIO DONE, THEN CLEAR TIO.
23 ;"CHC?T" PRINTS ASCII CHARACTER IN C(0)R; C(0)L MUST BE 0.
24 ;EXITS CALL +2 IF C(0)R=0; SIMULATES TAB
25 ;"TYP?" PRINTS C(0)R. EXITS AT CALL+1. REPLACE "TYP?" WITH
26 ;INTERRUPT TYP? IF DESIRED.
27 ;"TPS?" PRINTS A SPACE AND EXITS AT CALL+1 WITH ACO = 40
28
29 04720 054524 MES?: STA 3,MES?R ;PRINT A TEXT MESSAGE
30 04721 044465 STA 1,PAC?1
31 04722 050465 STA 2,PAC?2
32 04723 010521 ISZ MES?R
33 04724 031400 LDA 2,0,3 ;C(2) POINTS TO MESSAGE
34 04725 024463 LDA 1,P37?7 ;A 8 BIT MASK
35 04726 021000 LDA 0,0,2 ;C(2)=DATA WORD
36 04727 125112 MOVL# 1,1,SZC
37 04730 123701 ANDS 1,0,SKP
38 04731 123401 AND 1,0,SKP ;C(0)=DATA CHARACTER RIGHT
39 04732 151400 INC 2,2 ;INC TO NEXT WORD
40 04733 124000 COM 1,1 ;FLIP MASK
41 04734 004455 JSR CHC?T ;PRINT
42 04735 000771 JMP MES?+6 ;ANOTHER
43 04736 000402 JMP .+2
44 04737 004452 PL3?T: JSR CHC?T
45 04740 024446 PEX?T: LDA 1,PAC?1
46 04741 030446 LDA 2,PAC?2
47 04742 063511 SKPBZ TIO
48 04743 000777 JMP .-1
49 04744 060211 NIOC TIO
50 04745 002477 JMP @MES?R ;LAST
```

10084 N3MRT

```
01 04746 102401 ZOC?: SUB 0,0,SKP
02 04747 020563 POC?: LDA 0,PC6?0
03 04750 050437 STA 2,PAC?2
04 04751 030565 LDA 2,OCT?AB ;PRINT C(1) IN OCTAL
05 04752 000404 JMP .+4
06 04753 050434 PDE?: STA 2,PAC?2
07 04754 030572 LDA 2,DEC?TB ;PRINT C(1) IN DECIMAL
08 04755 102400 SUB 0,0
09 04756 054466 STA 3,MES?R ;BOTH ENTRIES PRINT NUMBER
10 04757 044427 STA 1,PAC?1
11 04760 040425 STA 0,ZSU?P ;THEN TAB TO NEXT POSITION
12 04761 050401 STA 2,..+1
13 04762 000000 DEC?OC: 0 ;"LDA 2, TABLE" INSTRUCTION
14 04763 010777 ISZ .-1
15 04764 020571 LDA 0,PTB?
16 04765 151005 MOV 2,2,SNR ;IF TABLE ENTRY=0
17 04766 000751 JMP PLS?T ;EXIT WITH A SPACE/TAB
18 04767 034416 LDA 3,ZSU?P ;ZEROS SUPPRESS STUF
19 04770 102400 SUB 0,0
20 04771 146452 DEO?T: SUBO# 2,1,SZC
21 04772 000405 JMP DEC?P
22 04773 146400 SUB 2,1 ;FORM THE DIGIT
23 04774 034536 LDA 3,PC6?0
24 04775 101400 INC 0,0
25 04776 000773 JMP DEO?T
26 04777 151235 DEC?P: MOVZR# 2,2,SNR
27 05000 034532 LDA 3,PC6?0
28 05001 054404 STA 3,ZSU?P ;C(0)=DIGIT
29 05002 163004 ADD 3,0,SZR ;MAKE ASCII
30 05003 004406 JSR CHC?T ;PRINT
31 05004 000756 JMP DEC?OC ;GET NEXT DIGIT
32
33 05005 000000 ZSU?P: 0
34 05006 000000 PAC?1: 0
35 05007 000000 PAC?2: 0
36 05010 000377 P37?7: 377
37 05011 054432 CHC?T: STA 3,CHR?T ;PRINT C(0) RIGHT
38 05012 101315 MOV3# 0,0,SNR ;RETURN +2 IF NULL
39 05013 001401 JMP 1,3
40 05014 034426 LDA 3,PC1?1 ;AC3 = 11
41 05015 116415 SUB# 0,3,SNR ;SKIP IF A TAB IS NOT TO
;BE SIMULATED
42
43 05016 000403 JMP CHA?3
44 05017 004540 JSR TYP? ;PRINT IT
45 05020 002423 JMP @CHR?T ;EXIT
46 05021 004535 CHA?3: JSR TPS? ;PRINT A SPACE
47 05022 020417 LDA 0,CHR?2
48 05023 034415 LDA 3,PC7? ;AC3 = 7
49 05024 163404 AND 3,0,SZR
50 05025 000774 JMP CHA?3 ;SIMULATE A TABE WITH 1
;TO 7 SPACES
51
52 05026 040413 STA 0,CHR?2
53 05027 002414 JMP @CHR?T
54
```

```

10085 N3MRT
01 05030 054414 CLF?: STA 3,MES?R
02 05031 044755 STA 1,PAC?1
03 05032 050755 STA 2,PAC?2
04 05033 020404 LDA 0,XJ15
05 05034 004755 JSR CHC?T
06 05035 020515 LDA 0,PC1?2
07 05036 000701 JMP PL9?T
08
09 05037 000015 XJ15: 15
10 05040 000007 PC7?: 7
11 05041 000000 CHR?Z: 0
12 05042 000011 PC1?1: 11
13 05043 000000 CHR?T: 0
14 05044 000000 MES?R: 0
15 05045 020505 TIC?: LDA 0,PC1?2
16 05046 004511 JSR TYP?
17 05047 010775 TIX?: ISZ MES?R
18 05050 024736 TIR?: LDA 1,PAC?1
19 05051 034734 LDA 3,ZSU?P
20 05052 175102 MOVL 3,3,SZC
21 05053 124400 NEG 1,1
22 05054 000665 JMP PEX?T+1
23
24 05055 102121 TIO?: ADCZL 0,0,SKP
25 05056 102440 TID?: SUBO 0,0
26 05057 054765 STA 3,MES?R
27 05060 050727 STA 2,PAC?2
28 05061 030471 LDA 2,PC1?2
29 05062 113000 ADD 0,2
30 05063 102440 SUBO 0,0
31 05064 040721 STA 0,ZSU?P
32
33 05065 034720 TIS?: LDA 3,ZSU?P
34 05066 175004 MOV 3,3,SZR
35 05067 000760 JMP TIX?
36 05070 054716 TIW?: STA 3,PAC?1
37 05071 063610 SKPDN TTI
38 05072 000777 JMP .-1
39 05073 060610 DIAC 0,TTI
40 05074 004715 JSR CHC?T
41 05075 034436 LDA 3,PC4?0
42 05076 116414 SUB# 0,3,SZR
43 05077 101015 MOV# 0,0,SNR
44 05100 000765 JMP TIS?
45 05101 024434 LDA 1,TIN?2
46 05102 106015 ADC# 0,1,SNR
47 05103 000744 JMP TIX?
48 05104 106424 SUBZ 0,1,SZR
49 05105 000405 JMP TIM?
50 05106 034677 LDA 3,ZSU?P
51 05107 177200 ADDR 3,3
52 05110 054675 STA 3,ZSU?P
53 05111 000760 JMP TIW?+1
54 05112 136415 TIM?: SUB# 1,3,SNR
55 05113 000732 JMP TIC?

```

```

:SAVE RETURN
:PRINT CARRIAGE AND LF
:OCTAL ENTRY
:DECIMAL ENTRY
:AC2 IS SAVED
:MINUS SIGN AND LEADING SPACES
:FLAG
:SPACE, TAB, OR NULL
:COMMA
:OR
:MINUS
:IF NEITHER THEN GO TO TIM?
:COMPLEMENT SIGN
:IS IT A CARRIAGE RETURN?
:IF CR THEN GO TO TIC?

```

```

10086 N3MRT
01 05114 024420 TIN?: LDA 1,TIN?1
02 05115 107022 ADDZ 0,1,SZC
03 05116 146513 SUBL# 2,1,SNC
04 05117 000731 JMP TIR?
05 05120 010665 ISZ ZSU?P
06 05121 020665 LDA 0,PAC?1
07 05122 101120 MOVZL 0,0
08 05123 115120 MOVZL 0,3
09 05124 175120 MOVZL 3,3
10 05125 137000 ADD 1,3
11 05126 145220 MOVZR 2,1
12 05127 125232 MOVZR# 1,1,SZC
13 05130 117000 ADD 0,3
14 05131 000737 JMP TIW?
15 05132 000060 PC6?0: 60
16 05133 000040 PC4?0: 40
17 05134 177720 TIN?1: -60
18 05135 000055 TIN?2: 55
19 05136 030555 OCT?AB: LDA 2,..+1,-DEC?OC
20 05137 100000 100000
21 05140 010000 10000
22 05141 001000 1000
23 05142 000100 C10?0: 100
24 05143 000010 10
25 05144 000001 1
26 05145 000000 0
27 05146 030565 DEC?TB: LDA 2,..+1,-DEC?OC
28 000012 .RDX 10
29 05147 023420 10000
30 05150 001750 1000
31 05151 000144 100
32 05152 000012 PC1?2: 10
33 05153 000001 1
34 05154 000000 0
35 000010 .RDX 8
36
37 05155 000011 PTB?: 11
38

```

```

:SKIP IF NOT A DIGIT
:SKIP IF DIGIT
:OUT OF LEADING SPACES
:8 OLD PAC?1'S + NEW DIGIT
:SKIP IF OCTAL MODE
:ADD 2 OLD PAC?1'S

```

## 10087 N3MRT

```

01 05156 020755 TPRS?: LDA 0,PC4?0 ;PREPARE TO PRINT A SPACE
02 05157 054461 TYP?: STA 3,TYP?R ;TYPE THE RIGHT BYTE OF ACO
03 05160 034456 LDA 3,INT? ;IF IT IS HERE DUE TO SWPAK
04 05161 175004 MOV 3,3,SZR ;TTO OUTPUTS WILL BE ENABLED.
05 05162 034230 LDA 3,SWREG ;READ THE SWITCHES
06 05163 040456 STA 0,P.MS0 ;SAVE ACO
07 05164 020457 LDA 0,P.MB5
08 05165 163404 AND 3,0,SZR ;BIT 5 OF SWREG SET?
09 05166 020454 LDA 0,P.MB1
10 05167 040152 STA 0,EACTV ;SET UP EACTV FOR LPT
11 05170 177100 ADDL 3,3
12 05171 175112 MOVL# 3,3,SZC ;SKIP IF TYPEOUTS ARE NOT
13 05172 000416 JMP PLP?T ;SUPPRESSED
14 05173 176620 SUBZR 3,3 ;BIT 0=1
15 05174 117000 ADD 0,3
16 05175 020444 LDA 0,P.MS0 ;GET SAVED CHARACTER
17 05176 063511 TPT?Y: SKPBZ TTO
18 05177 000777 JMP -,-1
19 05200 000401 JMP -,+1
20 05201 063511 SKPBZ TTO
21 05202 000774 JMP -,-4
22 05203 054152 STA 3,EACTV ;SET UP EACTV
23 05204 061111 DOAS 0,TTO ;OUTPUT TO TTO
24 05205 063511 SKPBZ TTO
25 05206 000777 JMP -,-1
26 05207 060211 NIOC TTO
27 05210 020431 PLP?T: LDA 0,P.MS0 ;GET CHARACTER
28 05211 034424 LDA 3,P17?7
29 05212 163400 AND 3,0 ;REMOVE PARITY BIT FOR LPT
30 05213 034152 LDA 3,EACTV
31 05214 177123 ADDZL 3,3,SNC ;SKP IS WANT LPT OUTPUT
32 05215 000405 JMP TPR?T ;FORGET LPT
33 05216 061117 DOAS 0,LPT ;OUTPUT THE CHARACTER TO LPT
34 05217 063517 SKPBZ LPT ;WAIT FOR LPT
35 05220 000777 JMP -,-1
36 05221 060217 NIOC LPT
37 05222 176400 TPR?T: SUB 3,3
38 05223 054152 STA 3,EACTV ;CLEAR EACTV SWITCH
39 05224 034411 LDA 3,P17?7
40 05225 116043 ADCO 0,3,SNC ;SKIP IF IT WAS A RUBOUT
41 05226 034705 LDA 3,PC4?0 ;AC3 = 40
42 05227 162432 SUBZ# 3,0,SZC ;SKIP FOR NON PRINTING
43 05230 010611 ISZ CHR?Z ;CHARACTERS
44 05231 034406 LDA 3,PC1?5 ;AC3 = 15
45 05232 116445 SUBO 0,3,SNR ;SKIP IF IT WAS NOT A "CR"
46 05233 054606 STA 3,CHR?Z ;CLEAR HORZ POS
47 05234 002404 JMP @TYP?R
48 05235 000177 P17?7: 177
49 05236 177777 INT?: -1
50 05237 000015 PC1?5: 15
51 05240 000000 TYP?R: 0
52 05241 000000 P.MS0: 0
53 05242 040000 P.MB1: 1B1
54 05243 002000 P.MB5: 1B5

```

## 10088 N3MRT

```

01
02
03 ;THIS PACKAGE IS USED TO CHANGE THE SETTINGS OF LOCATION
04 ;"SWREG" OF PAGE 0. THE PROGRAM CONTROL SHOULD ENTER "INP?"
05 ;WITH AC3 HAVING THE RETURN ADDRESS. THE CHARACTER INPUTED
06 ;BY THE OPERATOR IS ECHOED AFTER A "CR". IF THE COMMAND IS
07 ;NOT A LEGAL ONE THEN THE CONTROL IS RETURNED WITHOUT DOING
08 ;ANY THING, OTHERWISE ONE OF THE FOLLOWING COMMANDS IS
09 ;EXECUTED:
10 ;KEYS 1-9 AND A-F ARE USED TO COMPLEMENT THE CURRENT VALUE
11 ;OF BITS 1-15 OF "SWREG". IF ONE OF THESE KEYS IS HIT THE
12 ;CORRESPONDING BIT OF "SWREG" IS COMPLEMENTED AND THE CONTROL
13 ;IS RETURNED TO THE STATE PROGRAM HAD BEFORE HITTING THE KEY
14 ;TYPING OF A "0" WILL LOCK THE PROGRAM IN A SWITCH MODIFICATION
15 ;MODE IN WHICH CASE MORE THAN ONE BITS CAN BE CHANGED BEFORE
16 ;THE CONTROL IS ALLOWED TO RETURN TO THE MAIN PROGRAM. HITTING
17 ;THE "CR" KEY WILL UNLOCK THE PROGRAM FROM THIS MODE.
18 ;"D" THIS COMMAND GIVEN AT ANY TIME WILL RESET THE "SWREG"
19 ;TO DEFAULT MODE (ALL ZEROS) AND RESTART THE PROGRAM AT ADD.
20 ;STORED IN LOCATION "INS?"
21 ;"R" THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE PROG.
22 ;AT ADDRESS STORED IN LOCATION "INS?"
23 ;"M" THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE CURRENT
24 ;OPERATING MODES.
25 ;BEFORE THE CONTROL IS RETURNED TO THE MAIN PROGRAM BIT 0 WILL
26 ;BE SET IF ANY OF THE OTHER BITS OF "SWREG" IS SET, OTHERWISE
27 ;IT WILL BE CLEARED
28
29 ;THIS PACKAGE EXITS WITH C(AC3) = CHARACTER TYPED IN (PARITY
30 ;STRIPPED).
31 05244 000136 IN1?36: 136
32 05245 000104 IN1?04: 104
33 05246 000122 IN1?22: 122
34 05247 000033 INC?33: 33
35 05250 000000 INL?K: 0
36 05251 000202 INS?: STRT2
37 05252 000000 SVP?TB: 0
38 05253 005155 IPTB?: PTB?
39 05254 000115 IN115: 115

```

```

10089 N3MRT
01 05255 054454 INP?: STA 3,INR?T ;SAVE THE RETURN ADDRESS
02 05256 040454 STA 0,INS?0 ;SAVE ACO
03 05257 044454 STA 1,INS?1 ;AC1
04 05260 050454 STA 2,INS?2 ;AND AC2
05 05261 175200 MOVR 3,3 ;
06 05262 054454 STA 3,INS?C ;SAVE CARRY
07 05263 176400 SUB 3,3 ;
08 05264 054752 STA 3,INT? ;
09 05265 040763 STA 0,INL?K ;"INL?K" IS NOT -1
10 05266 020153 IN0?: LDA 0,LASTI ;READ THE INPUT
11 05267 034746 LDA 3,P1?7 ;AC3 = 177
12 05270 163400 AND 3,0 ;GET RID OF THE PARITY BIT
13 05271 024746 LDA 1,PC1?5 ;AC1 = 15
14 05272 106415 SUB# 0,1,SNR ;SKIP IF THE CHARACTER TYPED
15 ;WAS NOT "CR"
16 05273 000552 JMP INR? ;
17 05274 040441 STA 0,INS?3 ;SAVE CHARACTER
18 05275 024645 LDA 1,C10?0 ;AC1 = 100
19 05276 034654 LDA 3,PC1?2 ;AC3 = 12
20 05277 116414 SUB# 0,3,SZR ;SKIP IF IT IS A LINE FEED
21 05300 034747 LDA 3,INC?33 ;AC3 = 33
22 05301 162452 SUB0# 3,0,SZC ;SKIP IF ACO IS EQUAL OR MORE
23 ;THAN AC3
24 05302 000522 JMP INS? ;
25 05303 004654 IN1?: JSR TYP? ;ECHO THE CHARACTER
26 05304 034750 LDA 3,IN1?5 ;
27 05305 116405 SUB 0,3,SNR ;CK FOR "M"
28 05306 000431 JMP INM? ;
29 05307 034623 LDA 3,PC6?0 ;AC3 = 60
30 05310 152620 SUBZR 2,2 ;AC2 = 100000
31 05311 116405 SUB 0,3,SNR ;SKIP IF THE DIGIT TYPED WAS
32 ;NOT 0
33 05312 000457 JMP IN3? ;
34 05313 010735 ISZ INL?K ;IN SWREG LOOP?
35 05314 000465 JMP IN4?+1 ;NOPE-DON'T SET SWREG BITS
36 05315 014733 DSZ INL?K ;RESTORE LOOP FLG
37
38 05316 151221 IN2?: MOVR 2,2,SKP ;SHIFT AC2 TO RIGHT
39 05317 126520 SUBZL 1,1 ;AC1 = 1
40 05320 175405 INC 3,3,SNR ;
41 05321 000452 JMP IN3?+2 ;
42 05322 147415 AND# 2,1,SNR ;STAY IN LOOP UNTIL ALL BITS
43 ;OF SWREG ARE CHECKED
44 05323 000773 JMP IN2? ;
45 05324 106400 SUB 0,1 ;WHEN THE CONTROL COMES HERE
46 ;FOR THE FIRST TIME AC1 = 100
47 05325 135000 MOV 1,3 ;
48 05326 151225 MOVR 2,2,SNR ;
49 05327 000451 JMP IN4? ;
50 05330 000767 JMP IN2?+1 ;
51 05331 000000 INR?T: 0 ;
52 05332 000000 INS?0: 0 ;
53 05333 000000 INS?1: 0 ;
54 05334 000000 INS?2: 0 ;
55 05335 000000 INS?3: 0 ;
56 05336 000000 INS?C: 0 ;

```

```

10090 N3MRT
01 05337 126400 INM?: SUB 1,1 ;
02 05340 036713 LDA 3,@IPTB? ;SAVE PTB?
03 05341 054711 STA 3,SVP?TB ;PREPARE TO PRINT A SPACE
04 05342 034461 LDA 3,PC.40 ;AFTER EACH NUMBER
05
06 05343 056710 STA 3,@IPTB? ;
07 05344 006127 JSR @ICLF? ;TYPE A "CR" AND "LF"
08 05345 006130 JSR @IPDE? ;PRINT THE CONTENTS OF AC1
09 05346 034604 LDA 3,PC1?2 ;AC3 = 12
10 05347 125400 INC 1,1 ;
11 05350 166452 SUB0# 3,1,SZC ;SKIP IF AC1 IS GREATER OR EQUAL
12 ;TO AC3
13 05351 006450 JSR @ITPS? ;TYPE A SPACE
14 05352 101220 MOVR 0,0 ;ACO = 20
15 05353 122414 SUB# 1,0,SZR ;SKIP AFTER TYPING # 15
16 05354 000771 JMP INM?+6 ;
17 05355 006127 JSR @ICLF? ;
18 05356 030230 LDA 2,SWREG ;AC2 HAD SWITCH SETTINGS
19 05357 151140 MOVOL 2,2 ;BRING THE CARRY BIT IN AC1
20 05360 126560 SUBCL 1,1 ;TYPE THE CONTENTS OF AC1
21 05361 006131 JSR @IZOC? ;TYPE A SPACE
22 05362 006437 JSR @ITPS? ;SKIP AFTER TYPING ALL THE 16
23 05363 151124 MOVZL 2,2,SZR ;BITS
24
25 05364 000774 JMP -4 ;
26 05365 006127 JSR @ICLF? ;
27 05366 034664 LDA 3,SVP?TB ;RESTORE PTB?
28 05367 056664 STA 3,@IPTB? ;
29 05370 000410 JMP IN4? ;
30
31 05371 176000 IN3?: ADC 3,3 ;AC3 = -1
32 05372 054656 STA 3,INL?K ;LOCK IN SWITCH INPUT MODE
33 05373 024230 LDA 1,SWREG ;READ THE CURRENT VALUE OF
34 ;"SWREG"
35 05374 133414 AND# 1,2,SZR ;TAKE XOR OF AC1 AND AC2
36 05375 146401 SUB 2,1,SKP ;
37 05376 147000 ADD 2,1 ;
38 05377 044230 STA 1,SWREG ;SAVE THE NEW VALUE OF "SWREG"
39 05400 010650 IN4?: ISZ INL?K ;SKIP IF THE PROGRAM IS LOCKED
40 ;IN SWITCH INPUT MODE
41 05401 000444 JMP INR? ;
42 05402 014646 DSZ INL?K ;NEVER SKIP
43 05403 102520 SUBZL 0,0 ;SAVE EACTV
44 05404 024152 LDA 1,EACTV ;SET TTI ACTIVE BIT
45 05405 040152 STA 0,EACTV ;WHILE TTI ACTIVE IS SET
46 05406 020153 LDA 0,LASTI ;TTI INTR HANDLR WON'T SET SWREG BITS
47 05407 101102 MOVL 0,0,SZC ;
48 05410 000405 JMP +5 ;
49 05411 063610 SKPDN TTI ;WAIT FOR OPERATOR INPUT
50 05412 000774 JMP -4 ;
51 05413 060610 DIAC 0,TTI ;
52 05414 000402 JMP +2 ;
53 05415 101220 MOVR 0,0 ;
54 05416 040153 STA 0,LASTI ;RESTORE EACTV
55 05417 044152 STA 1,EACTV ;
56 05420 000647 JMP IN0?+1 ;
57 05421 005156 ITPS?: TPS? ;
58 05422 005157 ITYP?: TYP? ;
59 05423 000040 PC.40: 40 ;

```

```

10091 N3MRT
01 05424 107000 IN5?: ADD 0,1 ;AC1 = 100+ ASCII VALUE OF
02 ;CONTROL CHARACTER
03 05425 020617 LDA 0,IN1236 ;ACO = 136
04 05426 006774 JSR @ITYP? ;TYPE ^
05 05427 121000 MOV 1,0 ;ECHO CHARACTER
06 05430 006772 JSR @ITYP? ;CHECK FOR CNTL "0"
07 05431 034433 LDA 3,IN?17 ;AC3 = 122
08 05432 116405 SUB 0,3,SNR ;SKIP IF IT IS NOT ^R
09 05433 006427 JSR @INODT ;AC3 = 104
10 05434 034612 LDA 3,IN1722 ;SKIP IF IT WAS A ^D
11 05435 116405 SUB 0,3,SNR ;SET SWITCHES TO DEFAULT
12 05436 000405 JMP IN6? ;MODE
13 05437 034606 LDA 3,IN1704 ;AC3 = ADDRESS OF THE LOCATION
14 05440 116404 SUB 0,3,SZR ;WHERE THE PROGRAM WILL START
15 05441 000737 JMP IN4?
16 05442 054230 STA 3,SWREG
17
18 05443 034606 IN6?: LDA 3,INS?
19
20 05444 054665 STA 3,INR?T
21 05445 012416 INR?: ISZ @IINT?
22 05446 030230 LDA 2,SWREG
23 05447 176220 ADCZR 3,3
24 05450 173404 AND 3,2,SZR
25
26 05451 172000 ADC 3,2
27 05452 050230 STA 2,SWREG
28 05453 020663 LDA 0,INS?C
29 05454 101100 MOVL 0,0
30 05455 020655 LDA 0,INS?0
31 05456 024655 LDA 1,INS?1
32 05457 030655 LDA 2,INS?2
33 05460 034655 LDA 3,INS?3
34 05461 002650 JMP @INR?T
35 05462 004206 INODT: ODT
36 05463 005236 IINT?: INT?
37 05464 000117 IN?17: 117

```

```

10092 N3MRT
01 ;LINTR - LINKER INTERRUPT HANDLER
02 ;SAVES AC'S CARRY MSKO (0) AND USERMODE
03 ;STATUS ON A HARDWARE STACK.
04 ;THEREBY MAKING INTERRUPT PROCESSING
05 ;RE-ENTRANT
06 ;DEVICE INTERRUPT ADDRESS AND MSKO HAVE
07 ;BEEN ENTERED BY EACH DEVICE TEST
08 ;PREVIOUSLY PERFORMING THE APPROPRIATE
09 ;NUMBER OF "EINTS" ENTER INTERRUPT SERVICE
10 ;CALLS TO FILL THE APPROPRIATE DEV TABLES
11
12 05465 063777 LINTR: SKPDZ CPU ;POWER FAIL?
13 05466 002134 JMP @PFAIL
14 05467 075401 PSH 3 ;SAVE AC3
15 05470 071401 PSH 2 ;AC2
16 05471 065401 PSH 1 ;AC1
17 05472 061401 PSH 0 ;ACO
18 05473 020000 LDA 0,0
19 05474 040467 STA 0,LOC.0
20 05475 101102 MOVL 0,0,SZC ;CARRY
21 05476 063077 HALT ;BIT 0 OF 0=1?
22 05477 061401 PSH 0 ;PC*2+CRY IN BIT 15
23 05500 020462 LDA 0,ILLSC ;SET UP ILLEGAL SUPER.
24 05501 040000 STA 0,0 ;CALL IN LOC 0.
25 05502 020456 LDA 0,MSKRG
26 05503 061401 PSH 0 ;SAVE OLD MSKO
27 05504 020140 LDA 0,MPSWT
28 05505 101005 MOV 0,0,SNR ;SKIP IS USE MAP
29 05506 000406 JMP .+6 ;IGNORE MAP
30 05507 060402 DIA 0.2 ;GET STATUS
31 05510 061401 PSH 0 ;PUT ON STACK
32 05511 101120 MOVZL 0,0 ;TURN BIT 0 OFF
33 05512 101220 MOVZR 0,0
34 05513 061002 DOA 0.2
35 05514 022001 LDA 0,@1
36 05515 030537 LDA 2,LK300
37 05516 034441 LDA 3,LMSKS
38 05517 061477 INTA 0
39 05520 113000 ADD 0.2
40 05521 117000 ADD 0.3
41 05522 025400 LDA 1.0,3
42 05523 035000 LDA 3.0,2
43 05524 044434 STA 1,MSKRG
44 05525 030434 LDA 2,K4
45 05526 112415 SUB# 0,2,SNR ;WAS IT A DEV #4 INTR?
46 05527 001400 JMP 0.3 ;YEP, SERVICE IT
47 05530 101005 MOV 0,0,SNR ;SKP DOB IF TESTER
48 05531 005400 JSR 0.3 ;GO DIRECT
49 05532 066077 MSKO 1
50 05533 060177 INTEN
51 05534 005400 JSR 0.3 ;GO TO INTR SERV

```



10093 N3MRT

```
01 ;LINTD - LINKER INTERRUPT DISMISS
02 ;RETURN ADDRESS WAS GIVEN TO DEVICE
03 ;SERVICE ROUTINES VIA THE JSR 03
04
05 05535 060277 LINTD: INTDS
06 05536 020140 LDA 0,MP9WT
07 05537 101005 MOV 0,0,SNR ;SKIP IS USE MAP
08 05540 000403 JMP .+3
09 05541 061601 POP 0 ;RETRIEVE STATUS OFF STACK
10 05542 061002 DDA 0,2 ;WRITE IT
11 05543 065601 POP 1 ;RETRIEVE MASK
12 05544 044414 STA 1,MSKRG
13 05545 066077 MSKO 1
14 05546 061601 POP 0 ;GET CRY+2*PC
15 05547 101220 MOVZR 0,0 ;RESTORE CRY
16 05550 040413 STA 0,LOC.0
17 05551 061601 POP 0
18 05552 065601 POP 1
19 05553 071601 POP 2
20 05554 075601 POP 3
21 05555 060177 INTEN
22 05556 002405 JMP @LOC.0 ;GETS TO LOGICAL 0
23 05557 000000 LMSKS: 0
24 05560 000000 MSKRG: 0
25 05561 000004 K4: 4
26 05562 177770 ILLSC: 177770
27 05563 000000 LOC.0: 0
28 ;EINTP. - ENTER INTERRUPT SERVICE PARAMETERS
29 ;(AC0)=DEV# JSR @EINTS
30 ;(AC1)=MSKO
31 ;(AC2)=ADDRESS OF DEV INTR SERV
32
33 05564 054410 EINTP: STA 3,EI.S3
34 05565 034467 LDA 3,LK300
35 05566 117000 ADD 0,3
36 05567 051400 STA 2,0,3
37 05570 034767 LDA 3,LMSKS
38 05571 117000 ADD 0,3
39 05572 045400 STA 1,0,3
40 05573 002401 JMP @EI.S3
41
42 05574 000000 EI.S3: 0
```

10094 N3MRT

```
01 ;LCINT - INITIALIZE INTERRUPT SERVICE TABLES
02 ;VECTOR ADDRESSES ARE SET TO ILLEGAL INT
03 ;AND MSKO'S ARE SET TO -1
04 ;LSTKP IS SET TO START 1 AFTER MSKO'S
05 05575 020470 LCINT: LDA 0,LILLI
06 05576 030456 LDA 2,LK300
07 05577 041000 STA 0,0,2 ;FILL SERVICE
08 05600 151400 INC 2,2 ;VECTORS WITH
09 05601 145300 MOV5 2,1 ;AORS ILLEGAL INTR
10 05602 125224 MOVZR 1,1,SZR
11 05603 000774 JMP LCINT+2
12 05604 071001 MTSP 2 ;STACK POINTER TO 400
13 05605 070001 MTFP 2 ;FRAME POINTER TO 400
14 05606 052450 STA 2,@LC.K1+1
15 05607 145220 MOVZR 2,1
16 05610 131220 MOVZR 1,2 ;AC2=100,AC1=200
17 05611 022444 LDA 0,@LC.K1 ;RESERVE 100 WORDS
18 05612 113000 ADD 0,2 ;ABOVE MEM ALLOC. TBL5
19 05613 052442 STA 2,@LC.K1 ;100 FOR MSKO
20 05614 125220 MOVZR 1,1 ;AC1=100
21 05615 102000 ADC 0,0
22 05616 041000 STA 0,0,2
23 05617 113000 ADD 0,2
24 05620 107004 ADD 0,1,SZR
25 05621 000775 JMP .-3
26 05622 151400 INC 2,2
27 05623 050734 STA 2,LMSKS
28 05624 020427 LDA 0,LC.K2
29 05625 040001 STA 0,1
30 05626 102400 SUB 0,0
31 05627 040731 STA 0,MSKRG ;STRT INT MSK =0
32 05630 042427 STA 0,@LC.K1+2
33 05631 020427 LDA 0,LC.K1+3
34 05632 040003 STA 0,3 ;INIT FOR STACK INTA'S
35 05633 024117 LDA 1,PRFLG ;PARITY EXIST?
36 05634 125005 MOV 1,1,SNR ;SKP IS HAVE PARITY
37 05635 000405 JMP .+5 ;NO PARITY
38 05636 020423 LDA 0,LC.K7 ;ADDR OF PARITY HANDLER
39 05637 040304 STA 0,300+PRTY ;PLACE IN TABLE
40 05640 060204 NIOC PRTY ;CLR ANY PREVIOUS ERRS
41 05641 060104 NIOS PRTY ;INIT PARITY INTR
```

```

10095 N3MRT
01 05642 020420 LDA 0,LC.K8 ;ADDR OF TTI INTR HANDLER
02 05643 040310 STA 0,300+TTI ;PUT IN INTR TBL
03 05644 020417 LDA 0,LC.K9 ;ADDR OF TTO INTR HANDLER
04 05645 040311 STA 0,300+TTO ;PUT IN INTR TBL
05 05646 020134 LDA 0,PFFAIL ;PWR FAIL RTN ADDR
06 05647 040300 STA 0,300 ;PUT IN INTR TBL
07 05650 020414 LDA 0,LC.K6
08 05651 040317 STA 0,300+LPT ;SET UP FOR LPT INTA
09 05652 001400 JMP 0,3
10 05653 005465 LC.K2: LINTR
11 05654 000300 LK300: 300
12 05655 001467 LC.K1: LSETB
13 05656 002411 STADR
14 05657 002413 STKPG
15 05660 002433 STKIN
16 05661 005675 LC.K7: PRTHL ;PARITY INTR HANDLER ADDRESS
17 05662 014371 LC.K8: TT.TI
18 05663 014467 LC.K9: TT.TO
19 05664 005706 LC.K6: LPT.I

```

```

10096 N3MRT
01 05665 005666 LILLI: LILLI+1
02 05666 040146 STA 0,UDEVI
03 05667 024402 LDA 1,..+2
04 05670 123001 ADD 1,0,SKP
05 05671 060200 NIOC 0
06 05672 040401 STA 0,..+1
07 05673 060200 NIOC 0 ;CHANGED TO DEV#
08 05674 001400 JMP 0,3
09
10 ;PARITY INTERRUPT HANDLER SUBROUTINE
11 05675 022410 PRTHL: LDA 0,@PR.R0 ;GET SAVED LOC 0
12 05676 064404 DIA 1,PRTY ;GET MADR BITS
13 05677 071404 DIB 2,PRTY ;GET XMAOR BITS
14 05700 063077 HALT ;PARITY ERROR OCCURRED
15 05701 060204 NIOC PRTY ;CLEAR ERROR
16 05702 060104 NIOS PRTY ;REENABLE INTR
17 05703 002401 JMP @PR.RT ;RETURN TO INTERRUPT DISMISS
18 05704 005535 PR.RT: LINTD
19 05705 005563 PR.R0: LOC.0
20
21 ;LPT.I- LPT INTA HANDLER IF NO LPT TEST
22 05706 060217 LPT.I: NIOC LPT
23 05707 001400 JMP 0,3 ;RETURN

```

```

10097 N3MRT
01      ;IOVAL
02      ;I/O VALIDITY TRAP HANDLER
03      ;DETERMINE LEGALITY OF TRAP ERROR OR IF NOT EXPECTED
04 05710 022501 IOVAL: LDA 0,@ICD.0
05 05711 040567      STA 0,IOV.0
06 05712 026500      LDA 1,@ICD.1
07 05713 044566      STA 1,IOV.1
08 05714 032477      LDA 2,@ICD.2
09 05715 050565      STA 2,IOV.2
10 05716 036476      LDA 3,@ICD.3
11 05717 054560      STA 3,IOV.3
12 05720 102560      SUBCL 0,0
13 05721 040562      STA 0,IOV.4
14 05722 020233      LDA 0,VLPCR
15 05723 024234      LDA 1,VLDTA
16 05724 030145      LDA 2,STATS
17 05725 060177      INTEN
18 05726 040556      STA 0,IOV.A
19 05727 044556      STA 1,IOV.B
20 05730 050556      STA 2,IOV.C
21 05731 036560      LDA 3,@IOV.7
22 05732 175005      MOV 3,3,SNR
23 05733 000410      JMP NTODT
24 05734 101400      ETODT: INC 0,0
25 05735 040553      STA 0,IVODT
26 05736 020542      LDA 0,IOV.0
27 05737 024542      LDA 1,IOV.1
28 05740 030542      LDA 2,IOV.2
29 05741 034536      LDA 3,IOV.3
30 05742 002546      JMP @IVODT
31 05743 034142      NTODT: LDA 3,PSTRT
32 05744 035406      LDA 3,6,3
33 05745 175122      MOVZL 3,3,SZC
34 05746 000521      JMP IOV.E
35 05747 175220      MOVZR 3,3
36 05750 054537      STA 3,IOV.R
37 05751 034234      LDA 3,VLDTA
38 05752 020443      LDA 0,IOV40
39 05753 117404      AND 0,3,SZR
40 05754 000404      JMP .+4
41 05755 006132      JSR @LMESS
42 05756 006132      IO.TX
43 05757 000403      JMP IOVPR
44 05760 006132      JSR @LMESS
45 05761 006145      VL.TX
;VIOL. DATA
;STATUS WORD
;WAS THIS A ODT VIOLATION?
;SKIP=ODT VIOL.
;NOT A ODT VIOL.
;PC + 1
;RESTORE AC'S
;RETURN TO ODT
;BIT 0=1 IS EXPECTED TRP
;EXPECTED TRAP RETURN
;VIOL. DATA REG.
;=40
;SKP IS NOT VALIDITY VIOL.
;MUST BE I/O PROTECT ERR
;VALIDITY PROTECT ERR

```

```

10098 N3MRT
01 05762 020522 IOVPR: LDA 0,IOV.A
02 05763 006102      JSR @ERROI
03 05764 000401      JMP .+1
04 05765 006132      JSR @LMESS
05 05766 006113      IOTX5
06 05767 026046      LDA 1,@TPLOC
07 05770 006133      JSR @LPOCT
08 05771 006132      JSR @LMESS
09 05772 006317      IOTX3
10 05773 020505      LDA 0,IOV.0
11 05774 024505      LDA 1,IOV.1
12 05775 030505      LDA 2,IOV.2
13 05776 006103      JSR @ERROC
14 05777 000401      JMP .+1
15 06000 006132      JSR @LMESS
16 06001 006326      IOTX4
17 06002 020475      LDA 0,IOV.3
18 06003 024500      LDA 1,IOV.4
19 06004 030503      LDA 2,IOV.R
20 06005 006103      JSR @ERROC
21 06006 000401      JMP .+1
22 06007 155100      MOVL 2,3
23 06010 000457      JMP IOV.E
24 06011 003063 ICD.0: CD.S0
25 06012 003064 ICD.1: CD.S1
26 06013 003065 ICD.2: CD.S2
27 06014 003066 ICD.3: CD.S3
28 06015 000040 IOV40: 40
;FINISH PRGM MAP ETC
;INSTRUCTION MESSAGE
;GET SAVED INSTRUCTION
;PRINT IT
;TYPE AC'S HEADER
;PRINT AC 0,1 AND 2 AT TRP
;PRINT AC3 CARRY AND RETRN

```

10099 N3MRT

```
01          ;DWCHK - DEFER OR WRITE CHECK TRAP
02          ;DETERMINE LEGALITY OF TRAP
03          ;OR ERROR TYPE IF NOT EXPECTED
04
05 06016 022773 DWCHK: LDA 0,@ICD.0
06 06017 040461 STA 0,IOV.0
07 06020 026772 LDA 1,@ICD.1
08 06021 044460 STA 1,IOV.1
09 06022 032771 LDA 2,@ICD.2
10 06023 050457 STA 2,IOV.2
11 06024 036770 LDA 3,@ICD.3
12 06025 054452 STA 3,IOV.3
13 06026 102560 SUBCL 0.0
14 06027 040454 STA 0,IOV.4
15 06030 020233 LDA 0,VLPCR          ;PC REG.
16 06031 024234 LDA 1,VLDTA         ;VIOL. DATA
17 06032 030145 LDA 2,STATS       ;STATUS REG
18 06033 060177 INTEN
19 06034 040450 STA 0,IOV.A
20 06035 044450 STA 1,IOV.B
21 06036 050450 STA 2,IOV.C
22 06037 036452 LDA 3,@IOV.7
23 06040 175004 MOV 3,3,SZR
24 06041 000673 JMP ETODT
25 06042 034142 LDA 3,PSTRT
26 06043 035407 LDA 3,7,3
27 06044 175122 MOVZL 3,3,SZC      ;GET DEFER WRITE CHK RETRN
28 06045 000422 JMP IOV.E
29 06046 175220 MOVZR 3,3
30 06047 054440 STA 3,IOV.R
31 06050 125220 MOVZR 1,1
32 06051 125223 MOVZR 1,1,SNC     ;SKP IS WRITE PROTECT VIOL.
33 06052 000404 JMP .+4
34 06053 006132 JSR @LMESS
35 06054 006176 WP.TX
36 06055 000705 JMP IOVPR
37 06056 125220 MOVZR 1,1
38 06057 125223 MOVZR 1,1,SNC     ;SKP IS DEFER PROTECT ERR
39 06060 000404 JMP .+4
40 06061 006132 JSR @LMESS
41 06062 006162 DF.TX
42 06063 000677 JMP IOVPR
43 06064 006132 JSR @LMESS
44 06065 006212 AI.TX
45 06066 000674 JMP IOVPR          ;MUST BE AUTO-INDEX PROTECT ERR
                    ;FINISH TYPEOUT
```

10100 N3MRT

```
01          ;I/O OR VALIDITY TRAP RETURN
02          ;RETURN TO TEST FORCING VIOLATION
03          ;LOGICAL ADDRESS IS IN AC3 - LEFT 1
04 06067 175220 IOV.E: MOVZR 3,3      ;AC0-1-2=MAP REGISTERS
05 06070 054407 STA 3,IOV.3
06 06071 006421 JSR @IORLM      ;RELOAD MAP IN CASE 2ND LVL
07 06072 060277 INTDS
08 06073 034145 LDA 3,STATS   ;STATUS WORD
09 06074 075002 DDA 3,2        ;WRITE IT
10 06075 060177 INTEN
11 06076 002401 JMP @IOV.3
12 06077 000000 IOV.3: 0
13 06100 000000 IOV.0: 0
14 06101 000000 IOV.1: 0
15 06102 000000 IOV.2: 0
16 06103 000000 IOV.4: 0
17 06104 000000 IOV.A: 0
18 06105 000000 IOV.8: 0
19 06106 000000 IOV.C: 0
20 06107 000000 IOV.R: 0
21 06110 000000 IVOOT: 0
22 06111 004420 IOV.7: MAP.3
23 06112 003106 IORLM: LDMAP
24 06113 005215 IOTXS: .TXTE 1<15><12>INSTRUCTION CAUSING TRAP = 1
25 06132 005215 IO.TX: .TXTE 1<15><12>
26 06133 144411 I/O PROTECT ERROR!
27 06145 005215 VL.TX: .TXTE 1<15><12>
28 06146 053011 VALIDITY PROTECT ERROR!
29 06162 005215 DF.TX: .TXTE 1<15><12>
30 06163 042011 DEFER PROTECT ERROR!
31 06176 005215 WP.TX: .TXTE 1<15><12>
32 06177 153411 WRITE PROTECT ERROR!
33 06212 005215 AI.TX: .TXTE 1<15><12>
34 06213 040411 AUTO-INDEX PROTECT ERROR!
35 06230 005215 DCHTX: .TXTE 1<15><12>
36 06231 042011 DCH VIOLATION ERROR<15><12>
37 06244 151120 PROG#<11>ERRORS<15><12>!
```

```

10101 N3MRT
01 06254 005215 TXT.0: .TXTE (<15><12>PROGRAM # (
02 06263 005215 TXT.1: .TXTE (<15><12>AC'S (
03 06267 005215 TXT.2: .TXTE (<15><12>SCRLO/HI (
04 06275 141504 TXT.3: .TXTE (DCHLO/HI (
05 06302 005215 TXT.4: .TXTE (<15><12>MEM ALLOCATION TABLE<15><12>(
06 06317 005215 IOTX3: .TXTE (<15><12>AC0<11>AC1<11>AC2(
07 06326 005215 IOTX4: .TXTE (<15><12>AC3<11>CARRY<11>RET ADRS(
08 06341 051525 TXT.5: .TXTE IUSER A<11>I
09 06345 051525 TXT.6: .TXTE IUSER B<11>I
10 06351 005215 TXT.7: .TXTE I<15><12>ST.LA<11>START<11>ERROR(RES.)<15><12>I
11 06367 005215 TXT.8: .TXTE I<15><12>RAN AC= I
12 06375 050240 TXT.9: .TXTE I PHYS<11>LOGICAL PHYS<11>LOGICALI
13 06413 050240 TXT.A: .TXTE I PHYS<11>LOGICAL RELOC<11>PHYS<11>LOGICAL RELOC I
14 06437 005215 TXT.B: .TXTE I<15><12>RELLO/HII
15 06445 177777 PFTX: .TXTE I<177><177><177><15><12>POWER FAIL @ I
16
17 ;RANDCH= RANDOM SELECT DCH MAP A OR B
18 ;
19 06457 054416 RANDCH: STA 3,RND.3
20 06460 006077 JSR @ARANG ;GET RANDOM #
21 06461 126520 SUBZL 1,1
22 06462 107404 AND 0,1,SZR ;SKP IS USE DCH MAP A
23 06463 126621 SUBZR 1,1,SKP ;USE DCH MAP B
24 06464 126400 SUB 1,1
25 06465 030141 LDA 2,ALTBL ;GET ADDR OF ALLOC TBL
26 06466 021001 LDA 0,1,2 ;GET DCH ALLOC. WORD
27 06467 101120 MOVZL 0,0
28 06470 101220 MOVZR 0,0 ;DELETE BIT 0
29 06471 123000 ADD 1,0 ;ADD IN BIT 0 SELECTED
30 06472 041001 STA 0,1,2 ;PLACE IN DCH ALLOC. WORD
31 06473 010402 ISZ RND.3 ;SKIP ON EXIT
32 06474 002401 JMP @RND.3 ;RETURN
33 06475 000000 RND.3: 0
34

```

```

10102 N3MRT
01 ;AMSCR=ASIGN A SCRATCH AREA TO DCH
02 ;NO SKIP IF MEMORY ASSIGNED
03 ;OR NO SCRATCH AREA AVAILABLE TO ASSIGN
04 AMSCR:
05 06476 062401 SAVE
06 06477 054456 STA 3,AM.S3
07 06500 022141 LDA 0,@ALTBL ;GET #1K'S ASSIGNED
08 06501 101005 MOV 0,0,SNR ;NOT=0 INVALID NONSCR TO ASSIGN
09 06502 000441 JMP AM.XT
10 06503 030141 LDA 2,ALTBL
11 06504 025001 LDA 1,1,2
12 06505 135000 MOV 1,3 ;SAVE AC1
13 06506 125120 MOVZL 1,1
14 06507 125224 MOVZR 1,1,SZR ;ERROR EXIT IF
15 06510 000433 JMP AM.XT ;DCH ALREADY ASSIGNED
16 06511 020140 LDA 0,MPSWT
17 06512 101005 MOV 0,0,SNR ;SKP IF MAPPED
18 06513 000432 JMP AM,NM ;NO MAP
19 06514 175113 MOVL# 3,3,SNC ;SKP IF B DCH MAP (BIT 0=1)
20 06515 000403 JMP .+3
21 06516 034447 LDA 3,AM.BB ;B MAP DCH
22 06517 000402 JMP .+2
23 06520 034444 LDA 3,AM.AA ;A DCH MAP
24 06521 054403 STA 3,AM.GI+2
25 06522 006440 AM.GI: JSR @AM,RM ;SELECT A PAGE
26 06523 006556 AM.37 ;PAGE 0 TO 37
27 06524 000000 0 ;A OR B DCH MAP
28 06525 000416 JMP AM.XT ;NO CORE AVAILABLE
29 06526 105300 MOV 0,1 ;0=LOG
30 06527 125700 INCS 1,1 ;1 OPR=+1
31 06530 030141 LDA 2,ALTBL
32 06531 035001 LDA 3,1,2
33 06532 167000 ADD 3,1 ;ADD IN BIT 0 IF ON
34 06533 045001 STA 1,1,2 ;SAVE DCH
35 06534 040424 STA 0,AM.TM
36 06535 102400 SUB 0,0
37 06536 006423 JSR @AM,GA
38 06537 063077 HALT
39 06540 020420 LDA 0,AM.TM
40 06541 004561 JSR LDCHM
41 06542 012413 ISZ @AM,S3
42 06543 006414 AM.XT: JSR @AM,K1 ;LOAD MAP OPT. SET SCRLO+HI
43 06544 062601 RTRN

```

10103 N3MRT

```
01          ;MAP OPT DOES NOT EXIST
02          ;USE FIRST 1K SCR ASSIGNED
03 06545 102400 AM.NM:  SUB 0,0
04 06546 006413          JSR @AM.GA
05 06547 063077          HALT
06 06550 125300          MOV# 1,1
07 06551 125700          INCS 1,1
08 06552 030141          LDA 2,ALTBL      ;ENTER DCH ASSIGND
09 06553 045001          STA 1,1,2
10 06554 000766          JMP AM.XT-1
11 06555 000000 AM.S3:  0
12 06556 000037 AM.37: 37
13 06557 006747 AM.K1:  LDCHL
14 06560 000000 AM.TM:  0
15 06561 002616 AM.GA:  GETPA
16 06562 002256 AM.RM:  RMSEL
17 06563 000777 AM.77:  777
18 06564 015457 AM.AA:  DCHM0
19 06565 015461 AM.BB:  DCHM2      ;A DCH MAP
                                      ;B DCH MAP
```

10104 N3MRT

```
01          ;EMSCR=
02          ;EXPAND DCH SCRATCH AREA
03
04          EMSCR:
05 06566 062401          SAVE
06 06567 054766          STA 3,AM.S3
07 06570 030141          LDA 2,ALTBL
08 06571 021000          LDA 0,0,2      ;AC0=#1K SCR
09 06572 025001          LDA 1,1,2      ;AC1=DCH LIMIT
10 06573 044417          STA 1,EMS.0      ;SAVE IT
11 06574 125120          MOVZL 1,1
12 06575 125220          MOVZR 1,1      ;TURN BIT 0 OFF
13 06576 034760          LDA 3,AM.37
14 06577 137400          AND 1,3      ;3=START DCH LOG
15 06600 166700          SUBS 3,1      ;1=#1K'S DCH
16 06601 122415          SUB# 1,0,SNR      ;SKIP=NOT ALL AS-DCH
17 06602 000741          JMP AM.XT      ;EXIT ALL SR-DCH
18 06603 020140          LDA 0,MPSWT
19 06604 101004          MOV 0,0,SZR      ;SKIP IF NOT MAPPED
20 06605 000410          JMP EM.UM
21 06606 125700          INCS 1,1      ;+1 #DCH 1K'S
22 06607 137000          ADD 1,3
23 06610 055001          STA 3,1,2
24 06611 000731          JMP AM.XT-1      ;SKIP EXIT 1 MORE 1K
25
26 06612 000000 EMS.0:  0
```

```

10105 N3MRT
01
02      ;MAP OPTION EXISTS SEE IF THE NEXT
03      ;LOGICAL 1K DCH IS AVAILABLE
04 06613 000040 EM40: 40
05 06614 002627 EM,CM: CMAPB
06 06615 137000 EM,UM: ADD 1,3
07 06616 161000      MOV 3,0      ;(0)=NEXT HIGH DCH
08 06617 034774      LDA 3,EM40
09 06620 162415      SUB# 3,0,SNR      ;SKIP IS STILL IN 32K
10 06621 000722      JMP AM,XT
11 06622 030702      LDA 2,AM,GI+2      ;DCH BIT MAP
12 06623 006771      JSR @EM,CM        ;SKIP IF AVAIL
13 06624 006770      JSR @EM,CM        ;WASN'T SET IT=0 AND SKP
14 06625 135701      INCS 1,3,SKP      ;(3)UPPR=NEW DCH+1K
15 06626 000715      JMP AM,XT        ;EXIT CANT EXPAND UP
16 06627 040731      STA 0,AM,TM
17 06630 122400      SUB 1,0
18 06631 117000      ADD 0,3
19 06632 030141      LDA 2,ALTBL
20 06633 020757      LDA 0,EMS,0
21 06634 101103      MOVL 0,0,SNC      ;SKP IS B DCH MAP
22 06635 000403      JMP .+3
23 06636 102620      SUBZR 0,0        ;TURN ON BIT 0
24 06637 000402      JMP .+2
25 06640 102400      SUB 0,0
26 06641 117000      ADD 0,3          ;ADD IN BIT 0(ON OR OFF)
27 06642 055001      STA 3,1,2       ;RESTORE DCH ASSIGNMENTS
28 06643 121000      MOV 1,0         ;GET PHYSICAL PAGE #
29 06644 006715      JSR @AM,GA
30 06645 063077      HALT
31 06646 020712      LDA 0,AM,TM
32 06647 004453      JSR LDCHM
33 06650 000672      JMP AM,XT-1

```

```

10106 N3MRT
01      ;RDSCR - RELEASE SCRATCH FROM DCH
02      RDSCR:
03 06651 062401      SAVE
04 06652 054703      STA 3,AM,S3
05 06653 030141      LDA 2,ALTBL
06 06654 021001      LDA 0,1,2
07 06655 126620      SUBZR 1,1
08 06656 107400      AND 0,1         ;AC1 CONTAINS B DCH FLG
09 06657 044442      STA 1,RDSVO     ;SAVE IT
10 06660 101120      MOVZL 0,0
11 06661 101225      MOVZR 0,0,SNR
12 06662 000661      JMP AM,XT       ;EXIT NO DCH TO RELEASE
13 06663 126000      ADC 1,1        ;=-1
14 06664 101300      MOVS 0,0
15 06665 123300      ADDS 1,0       ;#1K'S IN DCH-1
16 06666 034433      LDA 3,RDSVO    ;GET SAVED DCH FLG
17 06667 163000      ADD 3,0        ;ADD IN BIT 0(ON/OFF)
18 06670 041001      STA 0,1,2
19 06671 162400      SUB 3,0
20 06672 024664      LDA 1,AM,37    ;(AC1) AFTER AND
21 06673 107400      AND 0,1        ;=LOGICAL PAGE #
22 06674 122704      SUBS 1,0,SZR   ;AC0=#1K'S LEFT
23 06675 000405      JMP .+5
24 06676 163000      ADD 3,0        ;ADD IN BIT 0 IF ON
25 06677 041001      STA 0,1,2     ;0 DCH ASSIGNED
26 06700 162400      SUB 3,0       ;REMOVE BIT 0 IF ON
27 06701 014654      DSZ AM,S3     ;NO SKIP ON EXIT
28 06702 034140      LDA 3,MPSWT
29 06703 175005      MOV 3,3,SNR   ;SKIP IF REALLY MAPPED
30 06704 000636      JMP AM,XT-1
31 06705 123000      ADD 1,0       ;AC0=LOG PAGE #
32 06706 024655      LDA 1,AM777   ;
33 06707 004413      JSR LDCHM
34 06710 034411      LDA 3,RDSVO   ;A OR B DCH
35 06711 175103      MOVL 3,3,SNC  ;SKP IS B DCH MAP
36 06712 000403      JMP .+3
37 06713 030652      LDA 2,AM,BB   ;B DCH MAP
38 06714 000402      JMP .+2
39 06715 030647      LDA 2,AM,AA   ;A DCH MAP
40 06716 006676      JSR @EM,CM     ;MAKE 1K DCH AVAIL
41 06717 000623      JMP AM,XT-1   ;EXIT NEW DCH LIM'S
42 06720 063077      HALT         ;DCH MAP BIT WAS ALRDY 1(CAN'T HAPPEN)
43 06721 000000      RDSV0: 0

```

```

10107 N3MRT
01      ;LDCHM - LOAD DATA CHANNEL MAP
02      ;LOGICAL DCH IS IN ACO
03      ;PHYSICAL DCH IS IN AC1
04      ;USED BY ASSIGN DCH AND EXPAND DCH
05 06722 040421 LDCHM: STA 0,LDC.0
06 06723 101300      MOV5 0,0
07 06724 103120      ADDZL 0,0
08 06725 044417      STA 1,LDC.1
09 06726 123000      ADD 1,0
10 06727 030141      LDA 2,ALTB1      ;ALLOCATION TBL
11 06730 031001      LDA 2,1,2      ;GET DCH FLG
12 06731 151103      MOVL 2,2,SNC   ;SKP IS B DCH WAS USED
13 06732 000403      JMP .+3
14 06733 024412      LDA 1,LDC.B    ;B DCH MAP
15 06734 000402      JMP .+2
16 06735 024411      LDA 1,LDC.A    ;A DCH MAP
17 06736 123000      ADD 1,0
18 06737 062002      DOB 0,2      ;WRITE MAP ENTRY
19 06740 024404      LDA 1,LDC.1
20 06741 020402      LDA 0,LDC.0
21 06742 001400      JMP 0,3
22 06743 000000      LDC.0: 0
23 06744 000000      LDC.1: 0
24 06745 101000      LDC.B: 101000 ;B DCH MAP
25 06746 100000      LDC.A: 100000 ;A DCH MAP

```

```

10108 N3MRT
01      ;LDCHL - LOAD DATA CHANNEL LIMITS
02      ;CALCULATE DCHHI/LO FOR THIS TST
03      ;INTEGRITY OF AC'S NOT PRESERVED
04 06747 030141      LDCHL: LDA 2,ALTB1
05 06750 025001      LDA 1,1,2      ;AC1 UPPER=#1K'S
06 06751 131000      MOV 1,2
07 06752 020604      LDA 0,AM.37   ;AC1 LOWER=1ST LOG 1K
08 06753 123400      AND 1,0
09 06754 106400      SUB 0,1
10 06755 101300      MOV5 0,0
11 06756 103120      ADDZL 0,0      ;ACO=LO LIMIT DCH
12 06757 151103      MOVL 2,2,SNC  ;SKP IS USE B DCH MAP
13 06760 000403      JMP .+3
14 06761 152620      SUBZR 2,2     ;BIT 0 = 1
15 06762 143000      ADD 2,0      ;ADD INTO LOGICAL ADDRESS
16 06763 040156      STA 0,DCHLO
17 06764 152620      SUBZR 2,2
18 06765 150000      COM 2,2
19 06766 143400      AND 2,0
20 06767 147400      AND 2,1      ;REMOVE BIT 0
21 06770 127125      ADDZL 1,1,SNR ;AC1=MOD 1K # 1'K
22 06771 125040      MOVO 1,1     ;SET C=1 NO 1K'S
23 06772 123000      ADD 1,0     ;ACO=HIGH SCR+1
24 06773 126500      SUBL 1,1    ;AC1=1 IF DCH ASSIGNED
25 06774 122400      SUB 1,0     ;AC1=0 IF NO DCH
26 06775 040157      STA 0,DCHHI ;HI DCH LOG LIMIT
27 06776 001400      JMP 0,3
28
29      ;THIS SUBROUTINE ALLOWS OPERATOR TO INPUT OPTIONS BEFORE
30      ;STARTING TESTS IF THE PROGRAM WASN'T AUTO STARTED.
31 06777 026413      LPRTN: LDA 1,@LPRSL-1
32 07000 125005      MOV 1,1,SNR
33 07001 002504      JMP @LPSV3    ;AUTO-RUN, DON'T WAIT
34 07002 024407      LDA 1,LPR60
35 07003 044153      STA 1,LASTI  ;FORCE 0 SWREG CMD
36 07004 006132      JSR @LMESS
37 07005 007215      LPR15
38 07006 006402      JSR @LPRCK  ;ENTER OPTIONS MESSAGE
39 07007 002476      JMP @LPSV3   ;ENTER KEY PACKAGE
40 07010 001065      LPRCK: CKODT
41 07011 100060      LPR60: 100060 ;EXIT

```



```

10109 N3MRT
01      ;LPRSL-LINKER PROGRAM SELECT DR,
02      ;ALLOW USER TO SELECT PROGRAMS TO RUN
03      ;TYPES CORE SIZE MAP EXIST AND ACTIVE PROGRAMS
04      ;IF (LAUTO)=-1 GIVES OPERATOR A CHANCE TO
05      ;DELETE SPECIFIC PROGRAMS
06 07012 001034      LAUTO
07 07013 054472 LPRSL: STA 3,LPSV3
08 07014 102400      SUB 0,0
09 07015 040471      STA 0,LPRGN
10 07016 020476      LDA 0,LPR1
11 07017 006073      JSR @ERRTX
12 07020 026467      LDA 1,@LPHIK      ;GET HIGHK PHYS.
13 07021 125400      INC 1,1          ;=MOD 1K
14 07022 006072      JSR @LDEC        ;PRINT SUPR 0'S
15 07023 024140      LDA 1,MPSWT
16 07024 020516      LDA 0,MPXTX      ;MAP EXIST TEXT
17 07025 125005      MOV 1,1,SNR      ;SKP IF MAP EXIS
18 07026 020523      LDA 0,NMPTX      ;NO MAP TEXT
19 07027 006073      JSR @ERRTX
20 07030 026762      LDA 1,@LPRSL-1
21 07031 020525      LDA 0,LPR2
22 07032 125004      MOV 1,1,SZR      ;SKP IF AUTO STRT
23 07033 020534      LDA 0,LPR3      ;NOT AUTO USE OTHR HDR
24 07034 006073      JSR @ERRTX
25 07035 020447      LDA 0,LPR4
26 07036 006073      JSR @ERRTX      ;PRT PRG HDR
27 07037 006070      JSR @PCRLF      ;CARRET LFEED TST 0
28      ;PRINT INDIVIDUAL TEST DESCRIPTIONS
29      ;GIVE OPR CHANCE TO DELETE IF LAUTO=-1
30 07040 020446 LPRLP: LDA 0,LPRGN      ;CUR PROG #
31 07041 034442      LDA 3,LPLZM
32 07042 117000      ADD 0,3
33 07043 031400      LDA 2,0,3
34 07044 151005      MOV 2,2,SNR      ;SKP IF NOT LAST
35 07045 000732      JMP LPRTN
36 07046 105000      MOV 0,1
37 07047 021002      LDA 0,2,2      ;GET PRG WAIT SW
38 07050 101004      MOV 0,0,SZR      ;SKP IF PRG NOT WAIT
39 07051 000430      JMP LPR1E      ;DEV MUST NOT EXIST
40 07052 050441      STA 2,LPIDX
41 07053 006071      JSR @LZOCT      ;TYPE PRG #
42 07054 020437      LDA 0,LPIDX
43 07055 024435      LDA 1,LPR10
44 07056 123000      ADD 1,0          ;CALC ADRS DESC TXT
45 07057 006073      JSR @ERRTX
46 07060 026732      LDA 1,@LPRSL-1
47 07061 125005      MOV 1,1,SNR      ;SKP IS LET OPR SELECT
48 07062 000416      JMP LPR1E-1      ;CR/LF AND DO NXT PRG

```

```

10110 N3MRT
01      ;WAIT FOR OPERATOR INPUT TO SELECT TEST
02      ;SPACE IS SELECT ANY OTHER IS DELETE
03 07063 063610      SKPDN TTI
04 07064 000777      JMP -,-1
05 07065 064610      DIAC 1,TTI      ;GET CHAR
06 07066 030422      LDA 2,LPR77
07 07067 147400      AND 2,1
08 07070 030421      LDA 2,LPR40
09 07071 146415      SUB# 2,1,SNR      ;SKP IF DELETED
10 07072 000406      JMP LPR1E-1      ;SELECTED CR/LF DO NEXT
11 07073 030420      LDA 2,LPIDX
12 07074 102000      ADC 0,0
13 07075 041002      STA 0,2,2      ;SET WAIT SW IN PRG
14 07076 006132      JSR @LMESS
15 07077 007320      LPDXT+1
16 07100 006070      JSR @PCRLF      ;CR/LF
17 07101 010405 LPR1E: ISZ LPRGN      ;+1 PROG #
18 07102 000736      JMP LPRLP      ;DO NEXT PROG
19 07103 000160 LPLZM: LZMAX
20 07104 007305 LPR4T: LPR4T
21 07105 000000 LPSV3: 0
22 07106 000000 LPRGN: 0
23 07107 001526 LPHIK: HIGHK
24 07110 000077 LPR77: 77
25 07111 000040 LPR40: 40
26 07112 000010 LPR10: 10
27 07113 000000 LPIDX: 0
28 07114 007115 LPR1: .+1
29 07115 005215      .TXTE "<15><12>N3MRT SHORT - REV 04
30 07131 005215 <15><12>TOTAL # 1K'S= "
31 07142 007143 MPXTX: .+1
32 07143 040515      .TXTE !MAP EXISTS!
33 07151 007152 NMPTX: .+1
34 07152 147516      .TXTE !NO MAP!
35 07156 007157 LPR2: .+1
36 07157 005215      .TXTE (<15><12>TEST RUN LIST(
37 07167 007170 LPR3: .+1
38 07170 005215      .TXTE (<15><12>SPACE SELECTS
39 07177 120123      -ANY OTHER CHAR. DELETES(
40 07215 005215 LPR15: .TXTE !<15><12>ENTER OPTIONS, CR TO CONTINUE!
41 07235 005215 FTYTX: .TXTE !<15><12>50 ERRORS<15><12>!
42 07244 005215 KEY6T: .TXTE !<15><12>TYPE CR TO CONTINUE!
43 07257 005215 TX65K: .TXTE !<15><12>65K PASSES TEST # 1
44 07272 005215 STHDR: .TXTE !<15><12>TST#<11>PASSES<11>ERRS<15><12>1
45 07305 005215 LPR4T: .TXTE (<15><12>TST#<11>DESCRIPTION(
46 07317 007320 LPDXT: .+1
47 07320 042240      .TXTE ( DELETED(
48 07325 005215 UDEVT: .TXTE !<15><12>UNEX. INTR. DEV# 1

```

```

10111 N3MRT
01          .MACRO SETUP
02          LCALL SETUL
03          X
04
05          .MACRO LOOP
06          LCALL LLOOP
07          X
08          .MACRO ERROR
09          JSR .+2
10          JMP .+2
11          LCALL ERRET
12          X
13          .MACRO EHALT
14          JSR .+2
15          JMP .+2
16          LCALL ERRET
17          X

```

```

10112 N3MRT
01          ;          .TITL CBRDS
02          ;MEMORY CHECKERBOARD RANDOM TO RUN WITH LINKER
03          ;PATTERN GENERATION AND CHECKING IS
04          ;MOVED INTO THE SELECTED SCRATCH AREAS
05          ;FOR EXECUTION
06          ;
07
08          ;DEF'S TO LINKER PARAMETER FILE FOLLOWS
09          NEXTT CB.00
10          007337 LMEML=.
11          000160          .LOC LPG0
12 00160 007342          CB.00
13          000161 LPG0=.
14          007337          .LOC LMEML
15 07337 000000          0          ;TEST PASS CTR
16 07340 000000          0          ;TEST ERROR CTR
17 07341 000000          0          ;INTERRUPT TIMEOUT SWITCH
18 07342 007361 CB.00: CB.01          ;INIT ENTRY ADRS.
19 07343 007364          CB.02          ;EXECUTE ENTRY ADRS
20 07344 000000          0          ;NO INTR WAITS
21 07345 000000          0          ;RAND SEL LIMITS
22 07346 177777          -1          ;=ALWAYS ENTER
23 07347 176000          176000          ;EVERY PROTECT BIT ON
24 07350 007512          CB.EC          ;NO I/O VALIDITY TRAPS
25 07351 007512          CB.EC          ;NO WRITE OR DEFER TRAPS
26          007352          .TXTE (
27 07352 044303 CHKRBRD RAN.(
28          151113
29          151102
30          120104
31          040722
32          027116
33          000000

```

10113 N3MRT

```
01 ;INITIALIZE CHECKERBOARD TEST SEQUENCE
02 07361 102000 CB.01: ADC 0,0
03 07362 040555 STA 0,CB.TK ;-1 TO TEST COUNTER
04 07363 001400 JMP 0,3 ;RETURN TO LINKER TEST INIT
05 ;
06 ;EXECUTE ENTRY POINT
07 07364 010553 CB.02: ISZ CB.TK ;SKP IS NO SCRATCH
08 07365 000473 JMP CB.03 ;DO NXT IN SEQ
09 LCALL ASCRA ;TRY TO GET 1K
10 07366 100010 ASCRA-ASCRA*1B11+100010
11 07367 000507 JMP CB.X1 ;NONE AVAILABLE
12 07370 102000 ADC 0,0 ;-1 TO
13 07371 040557 STA 0,CB.ES ;NO ERRSW
14 LCALL ARANG ;GET RAN#
15 07372 100270 ARANG-ASCRA*1B11+100010
16 07373 030545 LDA 2,CB.37
17 07374 105000 MOV 0,1
18 LCALL ADIVI ;REM=#1K'S TO EXPAND
19 07375 100310 ADIVI-ASCRA*1B11+100010
20 07376 100405 NEG 0,0,SNR
21 07377 000405 JMP CB.2A
22 CB.2L: LCALL ESCRA ;EXPAND SCRATCH 1K
23 07400 100030 ESCRA-ASCRA*1B11+100010
24 07401 000403 JMP CB.2A ;NO MORE AVAILABLE
25 07402 101404 INC 0,0,SZR
26 07403 000775 JMP CB.2L ;KEEP EXPANDING
```

10114 N3MRT

```
01
02 ;SCRATCH AREA HAS BEEN ASSIGNED RANDOMLY SEL
03 ;WHERE TO MOVE TEST WITHIN SCRATCH AREA
04 CB.2A: LCALL ARANG ;GET RAN #
05 07404 100270 ARANG-ASCRA*1B11+100010
06 07405 024154 LDA 1,SCRLO
07 07406 030155 LDA 2,SCRHI
08 07407 132400 SUB 1,2 ;# WORDS IN SCRATCH
09 07410 105000 MOV 0,1
10 LCALL ADIVI ;CREATE AN ADRS
11 07411 100310 ADIVI-ASCRA*1B11+100010
12 07412 024154 LDA 1,SCRLO ;WITHIN SCRATCH
13 07413 123000 ADD 1,0 ;TO RELOCATE TO
14 07414 030525 CB.2C: LDA 2,CB.PL ;PROG LENGTH
15 07415 142400 SUB 2,0
16 07416 024534 LDA 1,CBPL2
17 07417 122400 SUB 1,0 ;CALC LOW REL ADRS
18 07420 024154 LDA 1,SCRLO
19 07421 122433 SUBZ# 1,0,SNC ;SKP >LOLIMIT
20 07422 143000 ADD 2,0 ;MAKE GRTR THAN LO
21 07423 122433 SUBZ# 1,0,SNC
22 07424 000776 JMP -2
23 07425 040515 STA 0,CB.LC ;ADRS TO STORE TSTS
24 07426 144000 CB.RL: COM 2,1 ;-#WORDS TO MOV
25 07427 111000 MOV 0,2 ;TO ADRS
26 07430 034513 LDA 3,CB.BG ;FROM ADRS
27 07431 021400 CB.L2: LDA 0,0,3
28 07432 041000 STA 0,0,2
29 07433 175400 INC 3,3
30 07434 151400 INC 2,2
31 07435 125404 INC 1,1,SZR
32 07436 000773 JMP CB.L2 ;MOV ALL TO SCRATCH
33 07437 020513 LDA 0,CBPL2
34 07440 104000 COM 0,1
35 07441 143000 ADD 2,0
36 07442 101400 INC 0,0
37 07443 041377 STA 0,-1,2 ;SET EPROG
38 07444 020476 LDA 0,CB.LC
39 07445 041376 STA 0,-2,2 ;SET BPROG
40 07446 102400 SUB 0,0
41 07447 041374 STA 0,-4,2 ;CLR PONES
42 07450 050474 STA 2,CB.EN
43 ;MOV REST OF TEST INTO SCRATCH AREA
44 07451 034500 LDA 3,CB.BG2 ;STRT OF 2ND SECTION
45 07452 021400 CB.L3: LDA 0,0,3
46 07453 041000 STA 0,0,2
47 07454 151400 INC 2,2
48 07455 175400 INC 3,3
49 07456 125404 INC 1,1,SZR
50 07457 000773 JMP CB.L3
51 ;2ND PART OF TEST IS MOVED
```

10115 N3MRT

```
01
02          :TEST PROGRAMS HAVE BEEN MOVED TO SCRATCH AREA
03          :EXECUTE NEXT TES IN SEQUENCE
04 07460 030457 CB.03: LDA 2,CB.TK      ;GET TEST COUNT
05 07461 024472          LDA 1,CB.TS      ;ADRS SEQ TABLE
06 07462 133000          ADD 1,2
07 07463 035000          LDA 3,0,2      ;GET RELATIVE POSITION
08 07464 024456          LDA 1,CB.LC      ;STRT POS IN SCRATCH
09 07465 137000          ADD 1,3
10 07466 050457          STA 2,CB.TI
11 07467 054460          STA 3,CB.SE
12 07470 005400          JSR 0,3 ;***GO TO TEST ***
13 07471 000412          JMP CB.04      ;NO ERROR RETURN
14          ;SKIP ON RETURN IS GROSS ERROR
15          ;FAST CHECKSUM OF MEMORY WAS NOT CORRECT
16          ;BUT THE SECOND PASS THROUGH THE DATA CHECK
17          ;DID NOT FIND ANY ERRORS IN PATTERN GENERATED
18 07472 020454          LDA 0,OTOTL     ;FAST SUM RESULT
19 07473 030451          LDA 2,CB.EN
20 07474 025374          LDA 1,-4,2      ;#-1'S GENERATED
21 07475 004413          JSR CB.ER
22          CB.X1: LCALL RSCRA      ;RELEASE 1K SCRATCH
23 07476 100050          RSCRA=ASCRA*1B11+100010
24 07477 102001          ADC 0,0,SKP     ;RET ALL RELEASED
25 07500 000776          JMP CB.X1      ;RELEASE ALL
26 07501 040436          STA 0,CB.TK     ;SET NO SCRATCH SW
27          LCALL RETRN
28 07502 100210          RETRN=ASCRA*1B11+100010
29          ;NORMAL RETURN FROM TEST SEE IF PASS COMPLETE
30 07503 010442          CB.04: ISZ CB.TI
31 07504 022441          LDA 0,0,CB.TI   ;NEXT IN SEQ
32 07505 100005          COM 0,0,SNR     ;=-1 WAS END SEQ
33 07506 000417          JMP CB.X2      ;AND RELEASE SCRA.
34          LCALL RETRN
35 07507 100210          RETRN=ASCRA*1B11+100010
```

10116 N3MRT

```
01          ;PATTERN CHECK FOUND AN ERROR
02 07510 054437          CB.ER: STA 3,CB.SE
03          LCALL ERROI
04 07511 100350          ERROI=ASCRA*1B11+100010
05 07512 000401          CB.EC: JMP .+1 ;CONTINUE ERROR TYPEOUT
06 07513 020446          LDA 0,CB.TXT
07          LCALL ERRTX
08 07514 100170          ERRTX=ASCRA*1B11+100010
09 07515 020422          LDA 0,CB.TK     ;TEST #
10 07516 024424          LDA 1,CB.LC     ;LOGICAL ADDRESS
11 07517 030430          LDA 2,CB.SE     ;START TEST OR E CALL
12          LCALL ERROC
13 07520 100370          ERROC=ASCRA*1B11+100010
14 07521 000755          JMP CB.X1      ;SW0=1 RELEASE SCRATCH
15 07522 102400          SUB 0,0      ;OTHERWISE HOLD IT
16 07523 040425          STA 0,CB.ES     ;AS IS UNTIL SW0=1
17 07524 000757          JMP CB.04
18
19 07525 020230          CB.X2: LDA 0,SWREG ;CHECK IF RELEASEING
20 07526 103123          ADDZL 0,0,SNC ;SCRATCH-SKP IS NOT REL.
21 07527 000445          JMP CB.05      ;RELEASE IT
22 07530 020420          LDA 0,CB.ES     ;GET ERR SWITCH
23 07531 040406          STA 0,CB.TK     ;-1 IS NO ERRS
24 07532 101004          MOV 0,0,SZR
25 07533 000441          JMP CB.05      ;AND SCRATCH IS RELEASED
26 07534 020406          LDA 0,CB.LC     ;OTHERWISE MOVE
27 07535 030404          LDA 2,CB.PL     ;PROGRAM UP AND
28 07536 000670          JMP CB.RL     ;RESTART AT TEST 0
29
30 07537 000000          CB.TK: 0
31 07540 000037          CB.37: 37
32 07541 000151          CB.PL: EPROG-BEGIN
33 07542 000000          CB.LC: 0
34 07543 007615          CB.BG: BEGIN
35 07544 000000          CB.EN: 0
36 07545 000000          CB.TI: 0
37 07546 000000          OTOTL: 0
38 07547 000000          CB.SE: 0
39 07550 000000          CB.ES: 0
40 07551 007767          CBGG2: DISTUR
41 07552 000104          CBPL2: DIRET-DISTUR
42 07553 007554          CB.TS: CB.TS+1 ;TEST SEQUENCE TABLE
43 07554 000000          BEGIN-BEGIN
44 07555 000152          DISTUR-BEGIN
45 07556 000045          ICHECK-BEGIN
46 07557 000101          CB.FA-BEGIN
47 07560 177777          -1
48 07561 007562          CB.TXT: .+1
49 07562 005215          .TXTE (<15><12>CB.TK CB.LC CB.SE(
```

```

10117 N3MRT
01          :DETERMINE IF IT IS TIME TO SWAP MEMORY
02          :CROSSOVER CONSTANTS
03          CB.05: LCALL ARANG
04 07574 100270 ARANG=ASCRA*1811+100010
05 07575 024557 LDA 1,C17
06 07576 123404 AND 1,0,SZR
07 07577 000677 JMP CB.X1
08 07600 020411 LDA 0,CB17
09 07601 030411 LDA 2,CB400
10 07602 106414 SUB# 0,1,SZR
11 07603 000403 JMP .+3
12 07604 020407 LDA 0,CB37
13 07605 030407 LDA 2,CB10K
14 07606 040546 STA 0,C17
15 07607 050544 STA 2,C400
16 07610 000666 JMP CB.X1
17 07611 000017 CB17: 17
18 07612 000400 CB400: 400
19 07613 000037 CB37: 37
20 07614 010000 CB10K: 10000

```

```

10118 N3MRT
01
02          :THE FOLLOWING SERIES OF TESTS ARE RELOCATED TO
03          :SCRATCH AREA ASSIGNED FOR EXECUTION
04 07615 054532 BEGIN: STA 3,RETURN
05 07616 102400 SUB 0,0
06 07617 040544 STA 0,PONES
07 07620 034154 LDA 3,SCRLO
08 07621 030527 LDA 2,C076000
09 07622 020155 LDA 0,SCRHI ;3/0=LIMITS
10 07623 143400 AND 2,0 ;END DISTURB
11 07624 040526 STA 0,EDIST
12 07625 173400 AND 3,2
13 07626 050523 STA 2,MODUAL
14 07627 030154 IPAT: LDA 2,SCRLO
15 07630 024523 LDA 1,C400
16 07631 020533 LDA 0,PATT ;START PATTERN
17 07632 147404 AND 2,1,SZR
18 07633 100000 IPAT1: COM 0,0
19 07634 024520 LDA 1,C17
20 07635 034530 FILL: LDA 3,BPROG ;AVOID OVERWRITE TST
21 07636 156436 SUBZ# 2,3,SEZ
22 07637 000404 JMP .+4
23 07640 034526 LDA 3,EPROG
24 07641 172433 SUBZ# 3,2,SNC ;SKP>PROG END
25 07642 000405 JMP .+5
26 07643 041000 STA 0,0,2 ;STR PAT WORD
27 07644 034517 LDA 3,PONES
28 07645 117000 ADD 0,3
29 07646 054515 STA 3,PONES ;ACCUMULATE CHKSUM
30 07647 151400 INC 2,2 ;FOR NXT STOR
31 07650 034506 LDA 3,C77
32 07651 133414 AND# 1,2,SZR ;FILL FOR 16
33 07652 000763 JMP FILL
34 07653 157414 AND# 2,3,SZR ;THEN COM PAT
35 07654 000757 JMP IPAT1 ;EVERY 64
36 07655 024155 LDA 1,SCRHI ;TST FOR END
37 07656 125400 INC 1,1
38 07657 146434 SUBZ# 2,1,SZR
39 07660 000750 JMP IPAT+1
40 07661 002466 JMP @RETURN

```

10119 N3MRT

```
01 ;CHECK PATTERN IN SCRATCH AREA AGAINST GENERATED
02 07662 054465 ICHECK: STA 3,RETURN
03 07663 030154 LDA 2,SCRLO ;INIT PAT
04 07664 024467 LDA 1,C400
05 07665 020477 LDA 0,PATT
06 07666 133414 AND# 1,2,SZR
07 07667 100000 ICK: COM 0,0
08 07670 024475 LDA 1,BPROG ;DON'T CMP TST STOR
09 07671 146436 SUBZ# 2,1,SEZ
10 07672 000405 JMP CHECK
11 07673 024473 LDA 1,EPROG
12 07674 132436 SUBZ# 1,2,SEZ
13 07675 000402 JMP CHECK
14 07676 000404 JMP ECHECK-4
15 07677 025000 CHECK: LDA 1,0,2 ;GET WORD FROM MEM
16 07700 106414 SUB# 0,1,SZR ;SKP=
17 07701 006461 JSR @ERR1
18 07702 151400 INC 2,2
19 07703 024451 LOA 1,C17
20 07704 133414 AND# 1,2,SZR
21 07705 000763 JMP ICK+1
22 07706 034450 ECHECK: LDA 3,C77
23 07707 157414 AND# 2,3,SZR ;CHK END OF LINE
24 07710 000757 JMP ICK
25 07711 024155 LDA 1,SCRHI
26 07712 125400 INC 1,1
27 07713 146434 SUBZ# 2,1,SZR ;CHK END OF CORE
28 07714 000750 JMP ICHECK+2
29 07715 002432 JMP @RETURN
```

10120 N3MRT

```
01
02 ;FAST CHECKSUM PATTERN SHOULD = THAT STORED
03 07716 054431 CB.FA: STA 3,RETURN
04 07717 102400 SUB 0,0
05 07720 030154 LDA 2,SCRLO ;BEGIN OF SCRA.
06 07721 034444 LDA 3,BPROG ;START OF TST STORE
07 07722 156405 CB.F1: SUB 2,3,SNR
08 07723 000407 JMP CB.F3
09 07724 174400 NEG 3,3 ;(3)=#WORDS TO ADD
10 07725 025000 CB.F2: LDA 1,0,2 ;GET WRD
11 07726 123000 ADD 1,0 ;ACCU SUM-1'S
12 07727 151400 INC 2,2
13 07730 175404 INC 3,3,SZR
14 07731 000774 JMP CB.F2
15 07732 034155 CB.F3: LDA 3,SCRHI
16 07733 175400 INC 3,3
17 07734 156415 SUB# 2,3,SNR ;DONE ALL CORE
18 07735 000403 JMP CB.F4 ;YES EXIT
19 07736 030430 LDA 2,EPROG
20 07737 000763 JMP CB.F1 ;ADD ABOVE TSTS
21 07740 024423 CB.F4: LDA 1,PONES ;(1)=#-1'S GEN
22 07741 122414 SUB#1,0,SZR ;SHD BE=
23 07742 000402 JMP CB.F5 ;BUT AREN'T
24 07743 002404 JMP @RETURN
25 07744 010403 CB.F5: ISZ RETURN ;STEP EXIT
26 07745 042414 STA 0,@CB.WK ;SAVE IN CASE
27 07746 000715 JMP ICHECK+1 ;CHECK CAN'T FIND ERR
28 07747 000000 RETURN: 0
29 07750 076000 C07600: 76000
30 07751 000000 MODJAL: 0
31 07752 000000 EDIST: 0
32 07753 000400 C400: 400
33 07754 000017 C17: 17
34 07755 000020 C20: 20
35 07756 000077 C77: 77
36 07757 001777 C1777: 1777
37 07760 000101 C101: 101
38 07761 007546 CB.WK: 0TOTL
39 07762 007510 ERR1: CB.ER
40 07763 000000 PONES: 0
41 07764 000000 PATT: 0
42 07765 000000 BPROG: 0
43 07766 000000 EPROG: 0
```

```

10121 N3MRT
01
02
03
04 07767 054760 DISTUR: STA 3,RETURN
05 07770 030154 LDA 2,SCRLO
06 07771 145000 DISTL: MOV 2,1 ;2=START OF 16
07 07772 020763 LDA 0,C20 ;MAKE 1=END+1
08 07773 107000 ADD 0,1
09 07774 034771 LDA 3,BPROG ;START OR "BEGIN"
10 07775 136436 SUBZ# 1,3,SEZ ;END 16>BEG
11 07776 000411 JMP DISDO ;NO<DO THIS 16
12 07777 034767 LDA 3,EPROG ;GET END PRG
13 10000 172436 SUBZ# 3,2,SEZ ;STRT 16>END PRG
14 10001 000403 JMP .+3 ;YES
15 10002 131000 MOV 1,2 ;TRY STRT NXT16
16 10003 000766 JMP DISTL
17 10004 034155 LDA 3,SCRHI
18 10005 166436 SUBZ# 3,1,SEZ ;END 16>SCRHI
19 10006 002741 JMP @RETURN ;YES EXIT JOB DONE
20 10007 004450 DISDO: JSR DIXOR ;FLT A 1 BIT 16 WORDS
21 10010 020745 LDA 0,C20
22 10011 112400 SUB 0,2 ;BACK TO START OF 16
23 10012 114400 NEG 0,3 ;SHUFFLE 16 TIMES
24 10013 021000 DISDL: LDA 0,0,2 ;WORD 0
25 10014 025001 LDA 1,1,2
26 10015 041001 STA 0,1,2 ;GOES TO WORD 1
27 10016 021002 LDA 0,2,2
28 10017 045002 STA 1,2,2 ;1 GOES TO 2
29 10020 025003 LDA 1,3,2
30 10021 041003 STA 0,3,2 ;2 GOES TO 3
31 10022 021004 LDA 0,4,2
32 10023 045004 STA 1,4,2 ;3 GOES TO 4
33 10024 025005 LDA 1,5,2
34 10025 041005 STA 0,5,2 ;4 GOES TO 5
35 10026 021006 LDA 0,6,2
36 10027 045006 STA 1,6,2 ;5 GOES TO 6
37 10030 025007 LDA 1,7,2
38 10031 041007 STA 0,7,2 ;6 TO 7
39 10032 021010 LDA 0,10,2
40 10033 045010 STA 1,10,2 ;7 TO 10
41 10034 025011 LDA 1,11,2
42 10035 041011 STA 0,11,2 ;10 TO 11
43 10036 021012 LDA 0,12,2
44 10037 045012 STA 1,12,2 ;11 TO 12
45 10040 025013 LDA 1,13,2
46 10041 041013 STA 0,13,2 ;12 TO 13
47 10042 021014 LDA 0,14,2
48 10043 045014 STA 1,14,2 ;13 TO 14
49 10044 025015 LDA 1,15,2
50 10045 041015 STA 0,15,2 ;14 TO 15
51 10046 021016 LDA 0,16,2
52 10047 045016 STA 1,16,2 ;15 TO 16
53 10050 025017 LDA 1,17,2
54 10051 041017 STA 0,17,2 ;16 TO 17
55 10052 045000 STA 1,0,2 ;AND 17 BACK TO 0
56 10053 175404 INC 3,3,SZR ;DONE 16 TIMES
57 10054 000737 JMP DISDL ;WORDS NOT BACK TO ORIG YET
58 10055 004402 JSR DIXOR ;XOR BITS BACK TO ORIG
59 10056 000713 JMP DISTL ;DO REST OF SCRATCH

```

```

10122 N3MRT
01 ;COMS A SINGLE BIT IN EACH OF NEXT 16 WORDS
02 10057 054414 DIXOR: STA 3,DIRET ;THEN RESTORES TO ORIG
03 10060 102520 SUBZL 0,0 ;XOR 1 BIT
04 10061 025000 LDA 1,0,2 ;GET NEXT WRD
05 10062 135000 MOV 1,3 ;START BIT XOR
06 10063 117520 ANDZL 0,3 ;SO FLT BIT PAT
07 10064 107000 ADD 0,1 ;APPEARS IN THESE
08 10065 166400 SUB 3,1 ;16 WORDS
09 10066 045000 STA 1,0,2
10 10067 151400 INC 2,2
11 10070 101124 MOVZL 0,0,SZR ;DONE ALL - POS BIT L1
12 10071 000770 JMP DIXOR+2
13 10072 002401 JMP @DIRET
14 10073 000000 DIRET: 0
15 .ENDC

```

10123 N3MRT

```
01
02          :SC-MEMORY TEST - DEFINE PARAMETERS TO LINKR
03
04          NEXTT MM.00
05          010074 LMEML=.
06          000161          .LOC LPGO
07 00161 010077          MM.00
08          000162 LPGO=.
09          010074          .LOC LMEML
10 10074 000000          0          :TEST PASS CTR
11 10075 000000          0          :TEST ERROR CTR
12 10076 000000          0          :INTERRUPT TIMEOUT SWITCH
13
14 10077 010117 MM.00: MM.01          :INIT ENTRY
15 10100 010124          MM.02          :EXEC ENTRY
16 10101 000000          0          :WAIT SW
17 10102 000000          0          :
18 10103 177777          -1          :ALWAYS ENTER
19 10104 176000          176000          :EVERY PROTECT ON
20 10105 010265          MM.EC          :NO I/O VALIDITY TRAPS
21 10106 010265          MM.EC          :NO WRITE/DEFER TRAPS
22 10107 141523          .TXTE !SC MEMORY TEST!
23          046640
24          046705
25          151317
26          120131
27          142724
28          152123
29          000000
30
31          :INIT ENTRY
32
33 10117 102000 MM.01: ADC 0,0
34 10120 040531          STA 0,MM.ES          :CLR ERR SW
35 10121 001400          JMP 0,3          :RETURN
36
37 10122 000067 MM.PL: MM.EN-MM.3A
38 10123 010160 MM.LC: MM.3A
```

10124 N3MRT

```
01
02          :EXECUTE ENTRY POINT
03          :GET SCRATCH AREA
04 10124 020525 MM.02: LDA 0,MM.ES          :ERR SWITCH
05 10125 101005          MOV 0,0,SNR          :SKP IS NOT HAD ERROR
06 10126 000407          JMP MM.2A          :ERR HAS OCCURRED
07          :DON'T NEED TO GET SCRATCH
08 10127 102000          ADC 0,0
09 10130 040521          STA 0,MM.ES          :CLR ERR SWITCH
10          LCALL ASCRA          :GET 1K SCRATCH
11 10131 100010          ASCRA-ASCRA*1B11+100010
12 10132 000520          JMP MM.X1          :EXIT IF NONE AVAILABLE
13          MM.2L: LCALL ESCRA          :EXPAND SCRATCH BY 1K
14 10133 100030          ESCRA-ASCRA*1B11+100010
15 10134 000401          JMP MM.2A          :NO MORE AVAILABLE
16
17          :MOVE TESTS TO SCRATCH AREA
18 10135 030154 MM.2A: LDA 2,SCRLO
19 10136 034765          LDA 3,MM.LC          :FROM HERE
20 10137 141000          MOV 2,0
21 10140 024762          LDA 1,MM.PL
22 10141 123000          ADD 1,0
23 10142 040504          STA 0,MM.SA          :SAVE STARTING ADDR
24 10143 124000          COM 1,1          :=# OF WORDS TO MOVE
25 10144 021400 MM.L2: LDA 0,0,3
26 10145 041000          STA 0,0,2
27 10146 151400          INC 2,2
28 10147 175400          INC 3,3
29 10150 125404          INC 1,1,SZR
30 10151 000773          JMP MM.L2          :CONTINUE
31
32          :NOW EXECUTE TESTS IN SCRATCH
33 10152 034154 MM.03: LDA 3,SCRLO
34 10153 005400          JSR 0,3          :GO TO SCRATCH
35 10154 020475          LDA 0,MM.ES          :ERR SW
36 10155 101004          MOV 0,0,SZR          :SKP IS ERR HAS OCCURRED
37 10156 000474          JMP MM.X1          :RELEASE SCRATCH
38 10157 000500          JMP MM.X2          :DON'T REL. SCRATCH
```



10125 N3MRT

```
01
02      :SCRATCH AREA HAS BEEN ASSIGNED, FILL AREA WITH BACKGND
03
04 10160 054454 MM.3A: STA 3,MM.S3
05 10161 030465      LDA 2,MM.SA      :FILL SCRATCH WITH MINUS 1
06 10162 034155      LDA 3,SCRHI
07 10163 126000      ADC 1,1
08 10164 045000      STA 1,0,2
09 10165 021000      LDA 0,0,2      :CHECK IT GOT THERE
10 10166 106434      SUBZ# 0,1,SZR
11 10167 006446      JSR @MM3ER      :NOT -1,ERROR
12 10170 151400      INC 2,2
13 10171 156432      SUBZ# 2,3,SZC      :DONE ALL?
14 10172 000771      JMP MM.3A+3      :NOT YET
15
16      :ALL SCRATCH IS FILLED, NOW TEST WITH ISZ
17 10173 030453 MM.3B: LDA 2,MM.SA
18 10174 021000      LDA 0,0,2
19 10175 100014      COM# 0,0,SZR      :CHECK LOC BEFORE ISZ'ING
20 10176 006440      JSR @MM3E1      :NOT -1 BEFORE ISZ'NG
21 10177 011000      ISZ 0,2          :ISZ THIS LOCATION
22 10200 006437      JSR @MM3E2      :ERROR,ISZ DIDN'T SKIP
23 10201 021000      LDA 0,0,2          :GET CONTENTS
24 10202 101004      MOV 0,0,SZR      :LOC CONTENTS = 0?
25 10203 006435      JSR @MM3E3      :LOC NOT 0 AFTER ISZ
26 10204 015000      DSZ 0,2          :-1 TO LOC AGAIN
27 10205 000402      JMP .+2
28 10206 006433      JSR @MM3E4      :DSZ SKIPPED-ERROR
29 10207 021000      LDA 0,0,2          :CHECK CONTENTS
30 10210 100014      COM# 0,0,SZR      :=-1?
31 10211 006431      JSR @MM3E5      :LOC NOT -1 AFTER DSZ
32 10212 151400      INC 2,2
33 10213 156432      SUBZ# 2,3,SZC      :DONE ALL?
34 10214 000760      JMP MM.3B+1      :NOT YET
35
```

10126 N3MRT

```
01
02      :NOW DO ISZ TEST IN REVERSE DIRECTION
03 10215 030155 MM.3C: LDA 2,SCRHI      :STARTING ADDRESS
04 10216 034430      LDA 3,MM.SA      :FINISHING ADDRESS
05 10217 126000      ADC 1,1          :AC1=EXPECTED VALUE BEFORE ISZ
06 10220 021000      LDA 0,0,2          :CHECK LOC BEFORE DOING ISZ
07 10221 100014      COM# 0,0,SZR      :AC0=-1?
08 10222 006421      JSR @MM3E6      :LOC NOT -1 BEFORE ISZ
09 10223 011000      ISZ 0,2          :ISZ THIS LOCATION
10 10224 006420      JSR @MM3E7      :ISZ DIDN'T SKIP
11 10225 021000      LDA 0,0,2          :NOW CHECK LOC.
12 10226 101004      MOV 0,0,SZR      :LOC = 0?
13 10227 006416      JSR @MM3E8      :LOC NOT 0 AFTER ISZ
14 10230 133000      ADD 1,2          :DECREMENT AC2 BY 1
15 10231 172432      SUBZ# 3,2,SZC      :IS SCRHI<SCRLO?
16 10232 000766      JMP MM.3C+3      :NOT YET
17 10233 002401      JMP @MM.S3      :RETURN TO MAIN TEST SECTION
18
19      :TEST CONSTANTS
20 10234 000000 MM.S3: 0
21 10235 010260 MM3ER: MM.ER
22 10236 010300 MM3E1: MM.E1
23 10237 010304 MM3E2: MM.E2
24 10240 010311 MM3E3: MM.E3
25 10241 010315 MM3E4: MM.E4
26 10242 010322 MM3E5: MM.E5
27 10243 010326 MM3E6: MM.E6
28 10244 010332 MM3E7: MM.E7
29 10245 010337 MM3E8: MM.E8
30 10246 000000 MM.SA: 0
31 10247 000000 MM.EN: 0
32 10250 000000 MM.SE: 0
33 10251 000000 MM.E5: 0      :ERROR SWITCH
34
35      :DONE ALL TESTING,RELEASE SCRATCH
36
37 10252 102000 MM.X1: ADC 0,0
38 10253 040776      STA 0,MM.E5      :CLR ERR REL. SWITCH
39      LCALL RSCRA      :RELEASE 1K SCRATCH
40 10254 100050      RSCRA=ASCRA*1B11+100010
41 10255 102001      ADC 0,0,SKP
42 10256 000774      JMP MM.X1
43      MM.X2: LCALL RETRN
44 10257 100210      RETRN=ASCRA*1B11+100010
```

10127 N3MRT

```
01 ;PATTERN CHECK FOUND AN ERROR
02 10260 054770 MM.ER: STA 3,MM.SE
03 10261 176400 SUB 3,3
04 10262 050471 STA 2,MMLOC ;SAVE LOC ADDR.
05 10263 054467 STA 3,MM.TK ;SET ERROR #
06 LCALL ERROI
07 10264 100350 ERROI-ASCRA*1B11+100010
08 10265 000401 MM.EC: JMP .+1
09 10266 020466 LDA 0,MMTXT
10 LCALL ERRTX
11 10267 100170 ERRTX-ASCRA*1B11+100010
12 10270 020462 LDA 0,MM.TK ;ERR#
13 10271 024757 LDA 1,MM.SE ;ERROR CALL ADDR
14 10272 030461 LDA 2,MMLOC ;GET SAVED LOC
15 LCALL ERROC
16 10273 100370 ERROC-ASCRA*1B11+100010
17 10274 000756 JMP MM.X1
18 10275 102400 SUB 0,0
19 10276 040753 STA 0,MM.E5 ;SET ERR OCCURRED SW
20 10277 000636 JMP MM.2A
21
22 ;FORWARD ISZ TST - LOC NOT -1 BEFORE ISZ
23 10300 054750 MM.E1: STA 3,MM.SE
24 10301 126000 ADC 1,1
25 10302 176520 SUBZL 3,3
26 10303 000757 JMP MM.ER+2
27 ;FORWARD ISZ TST - ISZ DIDN'T SKIP
28 10304 054744 MM.E2: STA 3,MM.SE
29 10305 021000 LDA 0,0,2 ;ACTUAL
30 10306 126400 SUB 1,1 ;EXPECTED = 0
31 10307 034434 LDA 3,MM.K2
32 10310 000752 JMP MM.ER+2
33
34 ;FORWARD ISZ-LOC NOT 0 AFTER ISZ
35 10311 054737 MM.E3: STA 3,MM.SE
36 10312 126400 SUB 1,1
37 10313 034431 LDA 3,MM.K3
38 10314 000746 JMP MM.ER+2
39
40 ;D9Z TEST-DSZ SKIPPED
41 10315 054733 MM.E4: STA 3,MM.SE
42 10316 021000 LDA 0,0,2 ;ACTUAL
43 10317 126000 ADC 1,1 ;EXPECTED = -1
44 10320 034425 LDA 3,MM.K4
45 10321 000741 JMP MM.ER+2
46
47 ;D9Z TEST-LOC NOT -1 AFTER D9Z
48 10322 054726 MM.E5: STA 3,MM.SE
49 10323 126000 ADC 1,1
50 10324 034422 LDA 3,MM.K5
51 10325 000735 JMP MM.ER+2
52
53 ;REVERSE ISZ TEST-LOC NOT -1 BEFORE ISZ
54 10326 054722 MM.E6: STA 3,MM.SE
55 10327 126000 ADC 1,1
56 10330 034417 LDA 3,MM.K6
57 10331 000731 JMP MM.ER+2
58
```

10128 N3MRT

```
01 ;REVERSE ISZ TEST-ISZ DIDN'T SKIP
02 10332 054716 MM.E7: STA 3,MM.SE
03 10333 021000 LDA 0,0,2
04 10334 126400 SUB 1,1
05 10335 034413 LDA 3,MM.K7
06 10336 000724 JMP MM.ER+2
07
08 ;REVERSE ISZ TST-LOC NOT 0 AFTER ISZ
09 10337 054711 MM.E8: STA 3,MM.SE
10 10340 126400 SUB 1,1
11 10341 034410 LDA 3,MMK10
12 10342 000720 JMP MM.EC-3
13
14 10343 000002 MM.K2: 2
15 10344 000003 MM.K3: 3
16 10345 000004 MM.K4: 4
17 10346 000005 MM.K5: 5
18 10347 000006 MM.K6: 6
19 10350 000007 MM.K7: 7
20 10351 000010 MMK10: 10
21 10352 000000 MM.TK: 0
22 10353 000000 MMLOC: 0
23 10354 010355 MMTXT: .+1
24 10355 005215 .TXTE 1<15><12>MM.TK<11>MM.SE<11>LOCATIONI
```

10129 N3MRT

```
01 ; .TITL ARITH
02 ;ARITHMETIC TEST MODIFIED TO RUN WITH LINKR
03 ;MACRO DEF'S TO INTER COMMUNICATE FOLLOW
04
05
06 .MACRO CALL
07 X
08 .MACRO XORA
09 JSR XOR.0
10 X
11 .MACRO XOR1
12 JSR XOR.1
13 X
14 .MACRO XOR2
15 JSR XOR.2
16 X
17 .MACRO .DIVU
18 JSR DIVU
19 X
20 .MACRO .MPYA
21 JSR MPYA
22 X
23 .MACRO RANDOM
24 LCALL FRANG
25 X
26 .MACRO SQRT
27 JSR SQRT.
28 X
29 .MACRO SQ
30 JSR SQ.
31 X
32 .MACRO .MPYU
33 JSR MPYU
34 X
```

10130 N3MRT

```
01 ;ARITH - DEFINE PARAMETERS TO LINKR
02 NEXTT AT.00
03 010371 LMEML=.
04 000162 .LOC LPG0
05 00162 010374 AT.00
06 000163 LPG0=.
07 010371 .LOC LMEML
08 10371 000000 0 ;TEST PASS CTR
09 10372 000000 0 ;TEST ERROR CTR
10 10373 000000 0 ;INTERRUPT TIMEOUT SWITCH
11 10374 010414 AT.00: AT.01
12 10375 010417 AT.02
13 10376 000000 0
14 10377 000000 0
15 10400 177777 -1
16 10401 176000 176000
17 10402 010557 AT.EC
18 10403 010557 AT.EC
19 010404 .TXTE (
20 10404 151101 ARITHMETIC TEST(
21 152311
22 046510
23 152305
24 141711
25 152240
26 051705
27 000324
28 ;SET NO SCRATCH ASSIGNED SWITCH
29 10414 102000 AT.01: ADC 0,0
30 10415 040471 STA 0,AT.TK
31 10416 001400 JMP 0,3
32
```

```

10131 N3MRT
01          ;EXECUTE ENTRY POINT GET SCRATCH IF NONE AVAIL
02
03 10417 010467 AT.02: ISZ AT.TK      ;SKIP IS NO SCRATCH
04 10420 000467      JMP AT.03
05          LCALL ASCRA      ;GET 1K SCRATCH
06 10421 100010      ASCRA-ASCRA*1B11+100010
07 10422 000453      JMP AT.05      ;EXIT NONE AVAILABLE
08 10423 102000      ADC 0,0
09 10424 040555      STA 0,AT.ES      ;SET NO ERR SWTCH
10          LCALL ESCRA      ;TRY TO GET 1K MORE
11 10425 100030      ESCRA-ASCRA*1B11+100010
12 10426 000401      JMP .+1      ;BUT USE 1K IF NO MORE AVAIL
13 10427 030557      LDA 2,AT.37
14 10430 150400      NEG 2,2      ;TRY RANDOM BETWEEN LIM 37 TRYS
15          AT.2L: LCALL ARANG
16 10431 100270      ARANG-ASCRA*1B11+100010
17 10432 024155      LDA 1,SCRHI
18 10433 103000      ADD 0,0
19 10434 101220      MOVZR 0,0      ;CLR BIT 0
20 10435 122032      ADCZ# 1,0,SZC
21 10436 000411      JMP AT.2A      ;GRTR THAN HILIM
22 10437 034543      LDA 3,AT.PL
23 10440 166400      SUB 3,1      ;ENOUGH ROOM
24 10441 122032      ADCZ# 1,0,SZC      ;TO RELOCATE UP
25 10442 162400      SUB 3,0      ;NO LOWER NUMBER
26 10443 024154      LDA 1,SCRLO
27 10444 125400      INC 1,1
28 10445 122432      SUBZ# 1,0,SZC
29 10446 000404      JMP AT.2B
30 10447 151404 AT.2A: INC 2,2,SZR
31 10450 000761      JMP AT.2L
32 10451 121000      MOV 1,0

```

```

10132 N3MRT
01          ;MOVE TESTS TO SELECTED AREA
02
03 10452 111000 AT.2B: MOV 0,2      ;ADJUST SCRHI
04 10453 020527      LDA 0,AT.PL      ;SO THAT TESTS
05 10454 024155      LDA 1,SCRHI      ;WILL FIT
06 10455 106400      SUB 0,1      ;INTO SCRATCH
07 10456 132032      ADCZ# 1,2,SZC
08 10457 131000      MOV 1,2
09 10460 050523      STA 2,AT.LC      ;START ADRS IN SCRATCH
10 10461 024521 AT.RL: LDA 1,AT.PL
11 10462 124000      COM 1,1      ;=# WORDS TO MOVE
12 10463 030520      LDA 2,AT.LC      ;TO
13 10464 034520      LDA 3,AT.BG      ;FROM
14 10465 021400 AT.L2: LDA 0,0,3      ;MOVE LOOP
15 10466 041000      STA 0,0,2
16 10467 151400      INC 2,2
17 10470 175400      INC 3,3
18 10471 125404      INC 1,1,SZR
19 10472 000773      JMP AT.L2
20 10473 050512      STA 2,AT.EN
21 10474 000413      JMP AT.03
22 10475 102000 AT.05: ADC 0,0
23 10476 040410      STA 0,AT.TK
24          AT.XI: LCALL RETRN
25 10477 100210      RETRN-ASCRA*1B11+100010
26          ;MAP OPTION DOES NOT EXIST GO DIRECT
27 10500 020503 AT.GD: LDA 0,AT.LC
28 10501 040507      STA 0,AT.LA
29 10502 020154      LDA 0,SCRLO
30 10503 040504      STA 0,AT.LO
31 10504 040505      STA 0,AT.LP
32 10505 002503      JMP @AT.LA
33 10506 000000 AT.TK: 0

```

```

10133 N3MRT
01          ;TESTS HAVE BEEN MOVED TO SCRATCH
02          ;AT.03 SELECTS LOGICAL PAGE ASSIGNMENT
03          ;AND INITIATES TEST VIA GSCRA
04
05 10507 020472 AT.03: LDA 0,AT.ES      ;GET ERR SWITCH
06 10510 101005      MOV 0,0,SNR      ;SKIP IS NO ERR
07 10511 000416      JMP AT.04        ;USE PREV ASSIGN
08          LCALL ARANG
09 10512 100270      ARANG=ASCRA*1B11+100010
10 10513 105000      MOV 0,1
11 10514 030472      LDA 2,AT.37
12          LCALL ADIVI
13 10515 100310      ADIVI=ASCRA*1B11+100010
14 10516 040473      STA 0,AT.LP
15 10517 024464      LDA 1,AT.LC
16 10520 030154      LDA 2,SCRLO
17 10521 146400      SUB 2,1
18 10522 111300      MOVS 0,2
19 10523 153120      ADDZL 2,2
20 10524 050463      STA 2,AT.LO
21 10525 133000      ADD 1,2
22 10526 050462      STA 2,AT.LA
23 10527 020462 AT.04: LDA 0,AT.LP      ;REMAP SCR TO HERE
24 10530 030460      LDA 2,AT.LA      ;STARTING LOGICAL ADRS
25 10531 024422      LDA 1,ATERR     ;1=ERROR RET ADPRS
26          LCALL GSCRA      ;GO TO SCRATCH
27 10532 100070      GSCRA=ASCRA*1B11+100010
28 10533 000745      JMP AT.GD        ;COULDNT REMAP GO DIRECT
29          ;RETURN TO NEXT LOC PASS COMPLETE NO ERRS
30 10534 020230      LDA 0,SWREG     ;CHECK IF REL. SCRATCH
31 10535 103123      ADDZL 0,0,SNC    ;SKP IS NOT REL
32 10536 000406      JMP .+6         ;RELEASE SCRATCH
33 10537 020442      LDA 0,AT.ES     ;CHK FOR PREV ERR
34 10540 101004      MOV 0,0,SZR     ;SKP ON PREV ERR
35 10541 000403      JMP .+3         ;RELEASE ASSIGN
36 10542 040744      STA 0,AT.TK
37 10543 000734      JMP AT.XI
38          LCALL ARANG      ;IF BITS 0 AND 1=1
39 10544 100270      ARANG=ASCRA*1B11+100010
40 10545 103043      ADDO 0,0,SNC    ;RELEASE AND REMAP
41 10546 103003      ADD 0,0,SNC     ;IF EITHER =0 MAKE
42 10547 000773      JMP .-5         ;NEXT PASS SELECT NEW LP
43          AT.4A: LCALL RSCRA
44 10550 100050      RSCRA=ASCRA*1B11+100010
45 10551 000724      JMP AT.05        ;SET NO SCRATCH
46 10552 000776      JMP AT.4A

```

```

10134 N3MRT
01          ;ERROR IN TEST DURING EXECUTION
02
03 10553 010554 ATERR: ATERR+1
04 10554 054424      STA 3,ATS03      ;PRINT ERR HEADERS
05          LCALL ERROI
06 10555 100350      ERROI=ASCRA*1B11+100010
07 10556 000401      JMP .+1
08 10557 020433 AT.EC: LDA 0,ATTX1
09          LCALL ERRTX
10 10560 100170      ERRTX=ASCRA*1B11+100010
11 10561 020422      LDA 0,AT.LC      ;FOLLOW UP WITH
12 10562 024425      LDA 1,AT.LO      ;TEST RELOC INFO
13 10563 030425      LDA 2,AT.LA
14          LCALL ERROC
15 10564 100370      ERROC=ASCRA*1B11+100010
16 10565 000401      JMP .+1
17          LCALL ERPAC
18 10566 100250      ERPAC=ASCRA*1B11+100010
19 10567 020411      LDA 0,ATS03      ;CONTINUE TYPE
20 10570 024420      LDA 1,AT.LA
21 10571 030413      LDA 2,AT.BG
22          LCALL ERPAD
23 10572 100230      ERPAD=ASCRA*1B11+100010
24 10573 000755      JMP AT.4A      ;SW0=1 RELEASE SCR
25 10574 102400      SUB 0,0
26 10575 040404      STA 0,AT.ES     ;SET ERR SW
27 10576 040710      STA 0,AT.TK     ;CLR TEST K
28 10577 000700      JMP AT.XI      ;RETURN TO LINKR
29

```

```

10135 N3MRT
01 10600 000000 AT.S03: 0
02 10601 000000 AT.ES: 0
03 10602 001361 AT.PL: ATEND-MS1
04 10603 000000 AT.LC: 0
05 10604 010641 AT.BG: MS1
06 10605 000000 AT.EN: 0
07 10606 000037 AT.37: 37
08 10607 000000 AT.LO: 0
09 10610 000000 AT.LA: 0
10 10611 000000 AT.LP: 0
11 10612 010613 ATTX1: .+1
12 10613 005215 .TXTE (<15><12>AT.LC AT.LO AT.LA(
13 10625 010626 ATTX2: .+1
14 10626 005215 .TXTE (<15><12>ATS03 AT.LP E ADRS(

```

```

10136 N3MRT
01 MS1: SETUP ;MISC TEST OF NEG/COM
02 LCALL SETUL
03 10641 100410 SETUL-ASCRA*1B11+100010
04 RANDOM
05 LCALL FRANG
06 10642 100450 FRANG-ASCRA*1B11+100010
07 10643 104700 NEGS 0,1
08 10644 130342 COMOS 1,2,SZC
09 10645 142014 ADC# 2,0,SZR
10 ERROR
11 10646 004402 JSR .+2
12 10647 000402 JMP .+2
13 LCALL ERRET
14 10650 100470 ERRET-ASCRA*1B11+100010
15 LOOP
16 LCALL LLOOP
17 10651 100430 LLOOP-ASCRA*1B11+100010
18
19 ;MS2:
20 SETUP ;MISC TEST OF INC SWAPPED.
21 LCALL SETUL
22 10652 100410 SETUL-ASCRA*1B11+100010
23 RANDOM
24 LCALL FRANG
25 10653 100450 FRANG-ASCRA*1B11+100010
26 10654 111700 INCS 0,2
27 10655 145323 MOVZS 2,1,SNC
28 10656 106314 ADCS# 0,1,SZR
29 ERROR
30 10657 004402 JSR .+2
31 10660 000402 JMP .+2
32 LCALL ERRET
33 10661 100470 ERRET-ASCRA*1B11+100010
34 LOOP
35 LCALL LLOOP
36 10662 100430 LLOOP-ASCRA*1B11+100010
37

```

```

10137 N3MRT
01
02          ;MS3:          SETUP          ;MISC NEG TEST
03          LCALL SETUL
04 10663 100410 SETUL=ASCRA*1B11+100010
05 10664 102400 SUB 0,0
06 10665 100644 NEGOR 0,0,SZR
07          ERROR
08 10666 004402 JSR .+2
09 10667 000402 JMP .+2
10          LCALL ERRET
11 10670 100470 ERRET=ASCRA*1B11+100010
12 10671 100664 NEGCR 0,0,SZR
13          ERROR
14 10672 004402 JSR .+2
15 10673 000402 JMP .+2
16          LCALL ERRET
17 10674 100470 ERRET=ASCRA*1B11+100010
18 10675 100664 NEGCR 0,0,SZR
19          ERROR
20 10676 004402 JSR .+2
21 10677 000402 JMP .+2
22          LCALL ERRET
23 10700 100470 ERRET=ASCRA*1B11+100010
24          LOOP
25          LCALL LLOOP
26 10701 100430 LLOOP=ASCRA*1B11+100010
27

```

```

10138 N3MRT
01
02          ;ROT1:        SETUP          ;TEST ROTATE LEFT
03          LCALL SETUL
04 10702 100410 SETUL=ASCRA*1B11+100010
05          RANDOM
06          LCALL FRANG
07 10703 100450 FRANG=ASCRA*1B11+100010
08 10704 105000 MOV 0,1
09 10705 101100 MOVL 0,0
10 10706 111100 MOVL 0,2
11 10707 151100 MOVL 2,2
12 10710 151100 MOVL 2,2
13 10711 155100 MOVL 2,3
14 10712 175100 MOVL 3,3
15 10713 175100 MOVL 3,3
16 10714 161100 MOVL 3,0
17 10715 101100 MOVL 0,0
18 10716 101100 MOVL 0,0
19 10717 101100 MOVL 0,0
20 10720 101100 MOVL 0,0
21 10721 101100 MOVL 0,0
22 10722 101100 MOVL 0,0
23 10723 101100 MOVL 0,0
24 10724 101100 MOVL 0,0
25 10725 101100 MOVL 0,0
26 10726 106414 SUB# 0,1,SZR
27          ERROR
28 10727 004402 JSR .+2
29 10730 000402 JMP .+2
30          LCALL ERRET
31 10731 100470 ERRET=ASCRA*1B11+100010
32          LOOP
33          LCALL LLOOP
34 10732 100430 LLOOP=ASCRA*1B11+100010

```

```

10139 N3MRT
01          ;ROT2:
02          SETUP          ;TEST ROTATE RIGHT
03          LCALL SETUL
04 10733 100410 SETUL-ASCRA*1B11+100010
05          RANDOM
06          LCALL FRANG
07 10734 100450 FRANG-ASCRA*1B11+100010
08 10735 131000 MOV 1,2
09 10736 121200 MOVR 1,0
10 10737 101200 MOVR 0,0
11 10740 101200 MOVR 0,0
12 10741 101200 MOVR 0,0
13 10742 101200 MOVR 0,0
14 10743 101200 MOVR 0,0
15 10744 101200 MOVR 0,0
16 10745 101200 MOVR 0,0
17 10746 101200 MOVR 0,0
18 10747 101200 MOVR 0,0
19 10750 105200 MOVR 0,1
20 10751 125200 MOVR 1,1
21 10752 135200 MOVR 1,3
22 10753 175200 MOVR 3,3
23 10754 175200 MOVR 3,3
24 10755 161200 MOVR 3,0
25 10756 101200 MOVR 0,0
26 10757 112414 SUB# 0,2,SZR
27          ERROR
28 10760 004402 JSR .+2
29 10761 000402 JMP .+2
30          LCALL ERRET
31 10762 100470 ERRET-ASCRA*1B11+100010
32          LOOP
33          LCALL LLOOP
34 10763 100430 LLOOP-ASCRA*1B11+100010

```

```

10140 N3MRT
01          ;AND0:
02          SETUP          ;ANY NUMBER ANDED WITH
03          LCALL SETUL
04 10764 100410 SETUL-ASCRA*1B11+100010
05          RANDOM          ;ITSELF SHOULD NOT
06          LCALL FRANG
07 10765 100450 FRANG-ASCRA*1B11+100010
08 10766 131000 MOV 1,2          ;BE CHANGED.
09 10767 127700 ANDS 1,1
10 10770 127700 ANDS 1,1
11 10771 127700 ANDS 1,1
12 10772 127700 ANDS 1,1
13 10773 127700 ANDS 1,1
14 10774 135300 MOVS 1,3
15 10775 177700 ANDS 3,3
16 10776 177700 ANDS 3,3
17 10777 161300 MOVS 3,0
18 11000 103700 ANDS 0,0
19 11001 112414 SUB# 0,2,SZR
20          ERROR
21 11002 004402 JSR .+2
22 11003 000402 JMP .+2
23          LCALL ERRET
24 11004 100470 ERRET-ASCRA*1B11+100010
25          LOOP
26          LCALL LLOOP
27 11005 100430 LLOOP-ASCRA*1B11+100010
28
29          ;AND1:
30          SETUP          ;A NUMBER ANDED WITH ITS
31          LCALL SETUL
32 11006 100410 SETUL-ASCRA*1B11+100010
33          RANDOM          ;COMPLIMENT SHOULD
34          LCALL FRANG
35 11007 100450 FRANG-ASCRA*1B11+100010
36 11010 104042 COMO 0,1,SZC
37 11011 123704 ANDS 1,0,SZR          ;PRODUCE ZERO RESULT.
38          ERROR
39 11012 004402 JSR .+2
40 11013 000402 JMP .+2
41          LCALL ERRET
42 11014 100470 ERRET-ASCRA*1B11+100010
43          LOOP
44          LCALL LLOOP
45 11015 100430 LLOOP-ASCRA*1B11+100010
46

```



```

10141 N3MRT
01          ;AND3:
02          SETUP          ;PERFORM A AND INSTRUCTION
03          LCALL SETUL
04 11016 100410 SETUL-ASCRA*1B11+100010
05          RANDOM         ;WITH THE RESULT IN AC2.
06          LCALL FRANG
07 11017 100450 FRANG-ASCRA*1B11+100010
08 11020 176620 SUBZR 3,3      ;SIMULATE THE AND VIA
09 11021 131000 MOV 1,2      ;LOOKING FOR ADDER CARRY.
10 11022 113400 AND 0,2
11 11023 101113 AND3L: MOVL# 0,0,SNC
12 11024 000403 JMP .+3
13 11025 125112 MOVL# 1,1,SZC
14 11026 101141 MOVOL 0,0,SKP
15 11027 101121 MOVZL 0,0,SKP
16 11030 125141 MOVOL 1,1,SKP
17 11031 125120 MOVZL 1,1
18 11032 175224 MOVZR 3,3,SZR
19 11033 000770 JMP AND3L
20 11034 106415 SUB# 0,1,SNR ;CHECK IF AC0-1 ARE
21 11035 132414 SUB# 1,2,SZR ;THE SAME AND IF THEY
22          ERROR         ;ARGE WITH INST.
23 11036 004402 JSR .+2
24 11037 000402 JMP .+2
25          LCALL ERRET
26 11040 100470 ERRET-ASCRA*1B11+100010
27          LOOP
28          LCALL LLOOP
29 11041 100430 LLOOP-ASCRA*1B11+100010

```

```

10142 N3MRT
01          ;TEST PROCESSOR VIA EXCLUSIVE OR ROUTINES.
02
03          ;X1:
04          SETUP          ;C(AC1) IS SAVED IN C(AC2).
05          LCALL SETUL
06 11042 100410 SETUL-ASCRA*1B11+100010
07          RANDOM         ;AC0 IS EXCLUSIVE ORED
08          LCALL FRANG
09 11043 100450 FRANG-ASCRA*1B11+100010
10 11044 131000 MOV 1,2      ;WITH AC1 TWICE. THE SECOND
11          CALL          ;EXCLUSIVE OR SHOULD
12          XORA          ;RESTORE AC1 TO ITS
13 11045 004467 JSR XOR.0
14          CALL          ;ORIGINAL CONTENTS.
15          XORA
16 11046 004466 JSR XOR.0
17 11047 132414 SUB# 1,2,SZR
18          ERROR
19 11050 004402 JSR .+2
20 11051 000402 JMP .+2
21          LCALL ERRET
22 11052 100470 ERRET-ASCRA*1B11+100010
23          LOOP
24          LCALL LLOOP
25 11053 100430 LLOOP-ASCRA*1B11+100010
26
27          ;X2:
28          SETUP          ;THE FIRST EXCLUSIVE OR
29          LCALL SETUL
30 11054 100410 SETUL-ASCRA*1B11+100010
31          RANDOM         ;ROUTINE EXCHANGES THE
32          LCALL FRANG
33 11055 100450 FRANG-ASCRA*1B11+100010
34          CALL          ;CONTENTS
35          XOR2          ;OF AC0 AND AC1, IT ALSO
36 11056 004474 JSR XOR.2
37          CALL          ;FORMS THE EXCLUSIVE OR
38          XOR1          ;IN AC2. THE SECOND EXCLUSIVE
39 11057 004463 JSR XOR.1
40 11060 112414 SUB# 0,2,SZR ;OR FORMS THE RESULT
41          ERROR         ;OF AC0-AC1 IN AC0.
42 11061 004402 JSR .+2
43 11062 000402 JMP .+2
44          LCALL ERRET
45 11063 100470 ERRET-ASCRA*1B11+100010
46          LOOP
47          LCALL LLOOP
48 11064 100430 LLOOP-ASCRA*1B11+100010

```

```

10143 N3MRT
01
02
03
04 11065 100410
05
06
07 11066 100450
08 11067 130400
09
10
11 11070 004444
12
13
14 11071 004451
15 11072 113014
16
17 11073 004402
18 11074 000402
19
20 11075 100470
21
22
23 11076 100430
24
25
26
27
28 11077 100410
29
30
31 11100 100450
32 11101 126400
33
34
35 11102 004432
36 11103 106414
37
38 11104 004402
39 11105 000402
40
41 11106 100470
42
43
44 11107 100430

```

;X3:
   
 SETUP ;SAVE C(AC1) NEGATED IN
   
 LCALL SETUL
   
 SETUL=ASCRA\*1B11+100010
   
 RANDOM ;C(AC2). EXCLUSIVE OR C(AC0) TO
   
 LCALL FRANG
   
 FRANG=ASCRA\*1B11+100010
   
 NEG 1,2 ;C(AC1). EXCLUSIVE OR THE
   
 CALL ;RESULT BACK TO AC0.
   
 XORA ;CHECK VIA ADDITION TO
   
 JSR XOR.0
   
 CALL ;COMPLIMENT OF ORIGINAL
   
 XOR1 ;NUMBER.
   
 JSR XOR.1
   
 ADD# 0,2,SZR
   
 ERROR
   
 JSR .+2
   
 JMP .+2
   
 LCALL ERRET
   
 ERRET=ASCRA\*1B11+100010
   
 LOOP
   
 LCALL LLOOP
   
 LLOOP=ASCRA\*1B11+100010

;X4:
   
 SETUP ;EXCLUSIVE OR C(AC0) TO
   
 LCALL SETUL
   
 SETUL=ASCRA\*1B11+100010
   
 RANDOM ;ALL ZEROS IN C(AC1).
   
 LCALL FRANG
   
 FRANG=ASCRA\*1B11+100010
   
 SUB 1,1
   
 CALL
   
 XORA
   
 JSR XOR.0
   
 SUB# 0,1,SZR
   
 ERROR
   
 JSR .+2
   
 JMP .+2
   
 LCALL ERRET
   
 ERRET=ASCRA\*1B11+100010
   
 LOOP
   
 LCALL LLOOP
   
 LLOOP=ASCRA\*1B11+100010

```

10144 N3MRT
01
02
03
04 11110 100410
05
06
07 11111 100450
08 11112 102400
09
10
11 11113 004427
12 11114 106414
13
14 11115 004402
15 11116 000402
16
17 11117 100470
18
19
20 11120 100430
21
22
23
24
25 11121 100410
26
27
28 11122 100450
29 11123 102000
30
31
32 11124 004416
33 11125 110000
34 11126 146414
35
36 11127 004402
37 11130 000402
38
39 11131 100470
40
41
42 11132 100430
43 11133 000436

```

;X5:
   
 SETUP ;EXCLUSIVE OR C(AC1) TO
   
 LCALL SETUL
   
 SETUL=ASCRA\*1B11+100010
   
 RANDOM ;ALL ZEROS IN C(AC0).
   
 LCALL FRANG
   
 FRANG=ASCRA\*1B11+100010
   
 SUB 0,0
   
 CALL
   
 XOR1
   
 JSR XOR.1
   
 SUB# 0,1,SZR
   
 ERROR
   
 JSR .+2
   
 JMP .+2
   
 LCALL ERRET
   
 ERRET=ASCRA\*1B11+100010
   
 LOOP
   
 LCALL LLOOP
   
 LLOOP=ASCRA\*1B11+100010

;X6:
   
 SETUP ;EXCLUSIVE OR C(AC1) TO
   
 LCALL SETUL
   
 SETUL=ASCRA\*1B11+100010
   
 RANDOM ;(-1) IN C(AC0).
   
 LCALL FRANG
   
 FRANG=ASCRA\*1B11+100010
   
 ADC 0,0 ;THE COMPLIMENT OF
   
 CALL ;THIS RESULT SHOULD
   
 XOR1 ;EQUAL C(AC1).
   
 JSR XOR.1
   
 COM 0,2
   
 SUB# 2,1,SZR
   
 ERROR
   
 JSR .+2
   
 JMP .+2
   
 LCALL ERRET
   
 ERRET=ASCRA\*1B11+100010
   
 LOOP
   
 LCALL LLOOP
   
 LLOOP=ASCRA\*1B11+100010
   
 JMP X7

```

10145 N3MRT
01 11134 054433 XOR.0: STA 3,XOR.4
02 11135 135000      MOV 1,3      ;EXCLUSIVE OR C(AC0),C(AC1).
03 11136 117520      ANDZL 0,3    ;RESULT IS IN C(AC1).
04 11137 107000      ADD 0,1
05 11140 166400      SUB 3,1
06 11141 002426      JMP @XOR.4
07
08 11142 054425 XOR.1: STA 3,XOR.4
09 11143 135000      MOV 1,3      ;EXCLUSIVE OR C(AC0),C(AC1).
10 11144 117400      AND 0,3     ;RESULT IS IN C(AC0).
11 11145 174000      COM 3,3
12 11146 163400      AND 3,0
13 11147 123000      ADD 1,0
14 11150 163400      AND 3,0
15 11151 002416      JMP @XOR.4
16
17 11152 054415 XOR.2: STA 3,XOR.4
18 11153 034415      LDA 3,XM20  ;EXCLUSIVE OR C(AC0),C(AC1).
19 11154 054412      STA 3,XORTEM;RESULT IN C(AC2).
20 11155 115000      MOV 0,3     ;THE CONTENTS OF AC0 AND
21 11156 137200      ADDR 1,3   ;AC1 ARE EXCHANGED.
22 11157 151200      MOVR 2,2
23 11160 101220      MOVZR 0,0
24 11161 125200      MOVR 1,1
25 11162 103200      ADDR 0,0
26 11163 010403      ISZ XORTEM
27 11164 000771      JMP XOR.2+3
28 11165 002402      JMP @XOR.4
29 11166 000000 XORTEM: 0
30 11167 000000 XOR.4: 0
31 11170 177760 XM20: -20

```

```

10146 N3MRT
01
02
03 11171 100410      X7:
04
05
06 11172 100450      SETUP      ;C(AC1) IS SET EQUAL TO
07 11173 105000      LCALL SETUL
08
09
10 11174 004756      SETUL-ASCRA*1B11+100010
11 11175 151004      RANDOM    ;C(AC0). C(AC0) AND C(AC1)
12
13 11176 004402      LCALL FRANG
14 11177 000402      FRANG-ASCRA*1B11+100010
15
16 11200 100470      MOV 0,1    ;ARE EXCLUSIVE ORED WITH
17
18
19 11201 100430      CALL      ;THE RESULT GOING TO AC2.
20
21
22
23
24 11202 100410      XOR2
25
26
27 11203 100450      JSR XOR.2
28 11204 120000      MOV 2,2,SZR
29
30
31 11205 004745      ERROR
32 11206 150014      JSR .+2
33
34 11207 004402      JMP .+2
35 11210 000402      LCALL ERRET
36
37 11211 100470      ERRET-ASCRA*1B11+100010
38
39
40 11212 100430      LOOP
      LCALL LLOOP
      LLOOP-ASCRA*1B11+100010

```

```

;X8:
SETUP      ;C(AC0) IS SET TO THE
LCALL SETUL
SETUL-ASCRA*1B11+100010
RANDOM    ;COMPLIMENT OF C(AC1). THE
LCALL FRANG
FRANG-ASCRA*1B11+100010
COM 1,0   ;RESULT OF A EXCLUSIVE OR
CALL      ;SHOULD BE ALL BITS
XOR2     ;SET (-1).
JSR XOR.2
COM# 2,2,SZR
ERROR
JSR .+2
JMP .+2
LCALL ERRET
ERRET-ASCRA*1B11+100010
LOOP
LCALL LLOOP
LLOOP-ASCRA*1B11+100010

```

```

10147 N3MRT
01
02          ;X9:
03          SETUP          ;EXCLUSIVE OR ALL ONES
04          LCALL SETUL
05          SETUL-ASCRA*1B11+100010
06          ADCS 0,0        ;TO ALL ONES. THE
07          ADC 1,1         ;RESULT SHOULD BE
08          CALL           ;ALL ZEROS.
09          XORA
10          JSR XOR.0
11          ANDS 1,1,SZR
12          ERROR
13          JSR .+2
14          JMP .+2
15          LCALL ERRET
16          ERRET-ASCRA*1B11+100010
17          LOOP
18          LCALL LLOOP
19          LLOOP-ASCRA*1B11+100010
20
21          ;X10:
22          SETUP          ;EXCLUSIVE OR ALL ZEROS
23          LCALL SETUL
24          SETUL-ASCRA*1B11+100010
25          SUBS 0,0        ;TO ALL ZEROS. THE
26          SUB 1,1         ;RESULT SHOULD BE
27          CALL           ;ALL ZEROS IN C(AC1).
28          XORA
29          JSR XOR.0
30          ADDS# 0,1,SZR
31          ERROR
32          JSR .+2
33          JMP .+2
34          LCALL ERRET
35          ERRET-ASCRA*1B11+100010
36          LOOP
37          LCALL LLOOP
38          LLOOP-ASCRA*1B11+100010

```

```

10148 N3MRT
01
02          ;X11:
03          SETUP          ;FORM EXCLUSIVE OF FUNCTION
04          LCALL SETUL
05          SETUL-ASCRA*1B11+100010
06          RANDOM         ;IN C(AC2).
07          LCALL FRANG
08          FRANG-ASCRA*1B11+100010
09          CALL           ;CALL ANOTHER EXCLUSIVE
10          XOR2           ;OR FUNCTION SEVEN TIMES.
11          JSR XOR.2
12          CALL           ;THE RESULT SHOULD BE THE
13          XORA          ;SAME AS THE FIRST XORA
14          JSR XOR.0
15          CALL
16          XORA
17          JSR XOR.0
18          CALL
19          XORA
20          JSR XOR.0
21          CALL
22          XORA
23          JSR XOR.0
24          CALL
25          XORA
26          JSR XOR.0
27          CALL
28          XORA
29          JSR XOR.0
30          CALL
31          XORA
32          JSR XOR.0
33          SUB# 1,2,SZR
34          ERROR
35          JSR .+2
36          JMP .+2
37          LCALL ERRET
38          ERRET-ASCRA*1B11+100010
39          LOOP
40          LCALL LLOOP
41          LLOOP-ASCRA*1B11+100010
42          JMP A1
43          11254 000402    JMP A1
44          11255 000657    XOR.L: JMP XOR.0

```

10149 N3MRT

```
01          A1:  SETUP          ;SAVE C(AC2) IN C(AC1).
02          LCALL SETUL
03 11256 100410 SETUL=ASCRA*1811+100010
04          RANDOM          ;ADD AND ADD C(AC0) TO
05          LCALL FRANG
06 11257 100450 FRANG=ASCRA*1811+100010
07 11260 145000 MOV 2,1          ;C(AC2). THE VALUE IN
08 11261 112400 SUB 0,2          ;AC2 SHOULD NOT BE
09 11262 113000 ADD 0,2          ;CHANGED.
10 11263 146414 SUB# 2,1,SZR
11          ERROR
12 11264 004402 JSR .+2
13 11265 000402 JMP .+2
14          LCALL ERRET
15 11266 100470 ERRET=ASCRA*1811+100010
16          LOOP
17          LCALL LLOOP
18 11267 100430 LLOOP=ASCRA*1811+100010
19
20          A2:
21          SETUP
22          LCALL SETUL
23 11270 100410 SETUL=ASCRA*1811+100010
24          RANDOM
25          LCALL FRANG
26 11271 100450 FRANG=ASCRA*1811+100010
27 11272 102000 ADC 0,0
28 11273 123000 ADD 1,0
29 11274 111400 INC 0,2
30 11275 146414 SUB# 2,1,SZR
31          ERROR
32 11276 004402 JSR .+2
33 11277 000402 JMP .+2
34          LCALL ERRET
35 11300 100470 ERRET=ASCRA*1811+100010
36          LOOP
37          LCALL LLOOP
38 11301 100430 LLOOP=ASCRA*1811+100010
```

10150 N3MRT

```
01          A3:  SETUP          ;INCREMENT THE VALUE IN
02          LCALL SETUL
03          SETUL=ASCRA*1811+100010
04 11302 100410 RANDOM          ;AC1 AND ADD THAT VALUE
05          LCALL FRANG
06          FRANG=ASCRA*1811+100010
07 11303 100450 INCC 1,2          ;TO (-1). THE RESULT
08 11304 131460 ADCCS 0,0          ;SHOULD BE THE ORIGINAL
09 11305 102360 ADDC 2,0          ;NUMBER.
10 11306 143060 SUB# 0,1,SZR
11 11307 106414 ERROR
12          JSR .+2
13 11310 004402 JMP .+2
14 11311 000402 LCALL ERRET
15          ERRET=ASCRA*1811+100010
16 11312 100470 LOOP
17          LCALL LLOOP
18          LLOOP=ASCRA*1811+100010
19 11313 100430
20          A4:
21          SETUP          ;SAVE THE C(AC1) IN C(AC0).
22          LCALL SETUL
23 11314 100410 SETUL=ASCRA*1811+100010
24          RANDOM          ;A "ADDR" INSTRUCTION SHOULD
25          LCALL FRANG
26 11315 100450 FRANG=ASCRA*1811+100010
27 11316 121020 MOVZ 1,0          ;NOT CHANGE THE VALUE OF
28 11317 127200 ADDR 1,1          ;THE AC.
29 11320 127200 ADDR 1,1
30 11321 127200 ADDR 1,1
31 11322 127200 ADDR 1,1
32 11323 127200 ADDR 1,1
33 11324 127200 ADDR 1,1
34 11325 127200 ADDR 1,1
35 11326 127200 ADDR 1,1
36 11327 127200 ADDR 1,1
37 11330 127200 ADDR 1,1
38 11331 127200 ADDR 1,1
39 11332 127200 ADDR 1,1
40 11333 127200 ADDR 1,1
41 11334 127200 ADDR 1,1
42 11335 106454 SUBO# 0,1,SZR
43          ERROR
44 11336 004402 JSR .+2
45 11337 000402 JMP .+2
46          LCALL ERRET
47 11340 100470 ERRET=ASCRA*1811+100010
48          LOOP
49          LCALL LLOOP
50 11341 100430 LLOOP=ASCRA*1811+100010
```

10151 N3MRT

```
01                                     :A5:
02      SETUP                          ;THE RANDOM NUMBER IN
03      LCALL SETUL
04 11342 100410      SETUL-ASCRA*1B11+100010
05      RANDOM                          ;C(AC0) IS INCREMENTED VIA "INC"
06      LCALL FRANG
07 11343 100450      FRANG-ASCRA*1B11+100010
08 11344 115400      INC 0,3           ;AND DECREMENTED VIA "ADD".
09 11345 126000      ADC 1,1           ;THE FINAL RESULT IN C(AC2)
10 11346 137000      ADD 1,3           ;SHOULD BE EQUAL TO THE
11 11347 175400      INC 3,3           ;ORIGINAL NUMBER IN C(AC0).
12 11350 137000      ADD 1,3
13 11351 171400      INC 3,2
14 11352 133000      ADD 1,2
15 11353 151400      INC 2,2
16 11354 133000      ADD 1,2
17 11355 142414      SUB# 2,0,SZR
18      ERROR
19 11356 004402      JSR .+2
20 11357 000402      JMP .+2
21      LCALL ERRET
22 11360 100470      ERRET-ASCRA*1B11+100010
23      LOOP
24      LCALL LLOOP
25 11361 100430      LLOOP-ASCRA*1B11+100010
```

10152 N3MRT

```
01                                     :A6:
02      SETUP                          ;THE SUM OF AC0-1 IS
03      LCALL SETUL
04 11362 100410      SETUL-ASCRA*1B11+100010
05      RANDOM                          ;CHECKED WITH THE SIMULATED
06      LCALL FRANG
07 11363 100450      FRANG-ASCRA*1B11+100010
08 11364 135000      MOV 1,3           ;SUM.
09 11365 117000      ADD 0,3
10 11366 054424      STA 3,ADDTEM      ;SUM VIA ADD INSTRUCTION.
11 11367 131000      MOV 1,2
12      CALL
13 11370 004665      JSR XOR.L           ;SIMULATE THE ADD VIA
14 11371 143524      ANDZL 2,0,SZR      ;EXCLUSIVE OR. C(AC2)=
15 11372 000775      JMP A6L           ;RIPPLE CARRY,C(AC1)=RESULT.
16 11373 020417      LOA 0,ADDTEM
17 11374 122414      SUB# 1,0,SZR
18      ERROR
19 11375 004402      JSR .+2
20 11376 000402      JMP .+2
21      LCALL ERRET
22 11377 100470      ERRET-ASCRA*1B11+100010
23      LOOP
24      LCALL LLOOP
25 11400 100430      LLOOP-ASCRA*1B11+100010
26
27                                     :A7:
28      SETUP                          ;ADDITION OF NEGATED
29      LCALL SETUL
30 11401 100410      SETUL-ASCRA*1B11+100010
31      RANDOM                          ;NUMBERS SHOULD PRODUCE
32      LCALL FRANG
33 11402 100450      FRANG-ASCRA*1B11+100010
34 11403 110440      NEGO 0,2         ;ZERO AND A CARRY.
35 11404 143204      ADDR 2,0,SZR
36      ERROR
37 11405 004402      JSR .+2
38 11406 000402      JMP .+2
39      LCALL ERRET
40 11407 100470      ERRET-ASCRA*1B11+100010
41      LOOP
42      LCALL LLOOP
43 11410 100430      LLOOP-ASCRA*1B11+100010
44
45 11411 101001      MOV 0,0,SKP
46 11412 177777      ADDTEM: -1
```

10153 N3MRT

```
01
02      ;A8:
03      SETUP      ;ADD TEST.
04      LCALL SETUL
05      SETUL-ASCRA*1B11+100010
06      RANDOM
07      LCALL FRANG
08      FRANG-ASCRA*1B11+100010
09      SUBZL 2,2      ;=(+1)
10      MOVZ 2,2      ;=(400)
11      NEG 2,2      ;=(177400)
12      ANDS 2,0      ;SAVE HIGH ORDER 8 BITS.
13      MOVZS 0,1      ;SAME 8 BITS TO C(AC1)L.
14      ADDL 0,0      ;MOVE C(AC0) LEFT VIA
15      ADDL 0,0      ;ADD SHIFT.
16      ADDL 0,0
17      ADDL 0,0
18      SUB# 0,1,SZR
19      ERROR
20      JSR .+2
21      JMP .+2
22      LCALL ERRET
23      ERRET-ASCRA*1B11+100010
24      LOOP
25      LCALL LLOOP
26      LLOOP-ASCRA*1B11+100010
```

10154 N3MRT

```
01 11433 101001
02 11434 177400 M400:
03      ;ARI:
04
05      MOV 0,0,SKP
06      -400
07      SETUP      ;THE ORIGINAL CONTENTS OF
08      LCALL SETUL
09      SETUL-ASCRA*1B11+100010
10      RANDOM      ;AC1. BITS 0-7 ARE SQUARED
11      LCALL FRANG
12      FRANG-ASCRA*1B11+100010
13      LDA 2,M400      ;VIA MULTIPLY. THE SQUARE
14      ANDS 1,2      ;ROOT OF THE PRODUCT SHOULD
15      MOV 2,1      ;EQUAL THE ORIGINAL.
16      CALL
17      .MPYU      ;SEE SYSTEM REFFERENCE
18      JSR MPYU
19      MOV 1,0      ;MANUAL FOR FURTHER INFORMATION
20      CALL      ;ON MULTIPLY/SQ ROOT
21      SQRT      ;PROGRAMS.
22      JSR SQRT.
23      SUB# 0,2,SZR
24      ERROR
25      JSR .+2
26      JMP .+2
27      LCALL ERRET
28      ERRET-ASCRA*1B11+100010
29      LOOP
30      LCALL LLOOP
31      LLOOP-ASCRA*1B11+100010
```

```

10155 N3MRT
01                                     ;AR2:
02      SETUP          ;TAKE THE SQUARE ROOT
03      LCALL SETUL
04 11452 100410      SETUL-ASCRA*1B11+100010
05      RANDOM        ;OF A NUMBER. THE SQUARE
06      LCALL FRANG
07 11453 100450      FRANG-ASCRA*1B11+100010
08      CALL          ;ROOT OF THE RESULT SQUARED
09      SQRT          ;SHOULD BE THE SAME AS THE
10 11454 004457      JSR SQRT.
11 11455 105060      MOVC 0,1          ;ORIGIONAL ROOT.
12 11456 131060      MOVC 1,2
13      CALL
14      .MPYU
15 11457 004425      JSR MPYU
16 11460 121000      MOV 1,0
17      CALL          ;SQUARED NOW TAKE ROOT.
18      SQ
19 11461 004462      JSR SQ.
20 11462 112714      SUBS# 0,2,SZR
21      ERROR
22 11463 004402      JSR .+2
23 11464 000402      JMP .+2
24      LCALL ERRET
25 11465 100470      ERRET-ASCRA*1B11+100010
26      LOOP
27      LCALL LLOOP
28 11466 100430      LLOOP-ASCRA*1B11+100010

```

```

10156 N3MRT
01                                     ;AR3:
02      SETUP          ;FIND SQUARE ROOT VIA
03      LCALL SETUL
04 11467 100410      SETUL-ASCRA*1B11+100010
05      RANDOM        ;DIFFIRENT SUBROUTINES.
06      LCALL FRANG
07 11470 100450      FRANG-ASCRA*1B11+100010
08 11471 141000      MOV 2,0
09      CALL
10      SQRT
11 11472 004441      JSR SQRT.
12 11473 105000      MOV 0,1          ;SAVE FIRST RESULT IN AC1
13 11474 141000      MOV 2,0
14      CALL
15      SQ           ;RESULT IN ACO.
16 11475 004446      JSR SQ.
17 11476 106714      SUBS# 0,1,SZR
18      ERROR
19 11477 004402      JSR .+2
20 11500 000402      JMP .+2
21      LCALL ERRET
22 11501 100470      ERRET-ASCRA*1B11+100010
23      LOOP
24      LCALL LLOOP
25 11502 100430      LLOOP-ASCRA*1B11+100010
26 11503 000455      JMP AR4

```



10157 N3MRT

01 11504 102460 MPYU: SUBC 0,0  
02 11505 054411 MPYA: STA 3,.CB03  
03 11506 034411 LDA 3,.CB20  
04 11507 125203 .CB99: MOVR 1,1,SNC  
05 11510 101201 MOVR 0,0,SKP  
06 11511 143220 ADDZR 2,0  
07 11512 175404 INC 3,3,SZR  
08 11513 000774 JMP .CB99  
09 11514 125260 MOVCR 1,1  
10 11515 002401 JMP @.CB03  
11 11516 000000 .CB03: 0  
12 11517 177760 .CB20: -20  
13  
14 11520 102400 DIVI: SUB 0,0  
15 11521 054775 DIVU: STA 3,.CB03  
16 11522 034775 LDA 3,.CB20  
17 11523 125120 MOVZL 1,1  
18 11524 101100 .CC98: MOVL 0,0  
19 11525 142412 SUB# 2,0,SZC  
20 11526 142400 SUB 2,0  
21 11527 125100 MOVL 1,1  
22 11530 175404 INC 3,3,SZR  
23 11531 000773 JMP .CC98  
24 11532 002764 JMP @.CB03  
25

;C(AC1)\*C(AC2)  
;RESULT IN AC0,AC1.  
;SEE SYSTEM REFFERANCE  
;MANUAL FOR FURTHER  
;INFORMATION.

;C(AC0),C(AC1)/C(AC2).  
;AC0=REMAINDER  
;AC1=QUOTIENT  
;SEE SYSTEM REFFERANCE  
;MANUAL.

10158 N3MRT

01 11533 054423 SQRT.: STA 3,SQ.S3  
02 11534 126520 SUBZL 1,1  
03 11535 135120 MOVZL 1,3  
04 11536 122422 SUBZ 1,0,SZC  
05 11537 167001 ADD 3,1,SKP  
06 11540 121221 MOVZR 1,0,SKP  
07 11541 000775 JMP .-3  
08 11542 002414 JMP @SQ.S3  
09  
10 11543 054413 SQ.: STA 3,SQ.S3  
11 11544 176400 SUB 3,3  
12 11545 054412 STA 3,SQTEM  
13 11546 162023 ADCZ 3,0,SNC  
14 11547 000405 JMP SQ1  
15 11550 010407 ISZ SQTEM  
16 11551 010406 ISZ SQTEM  
17 11552 034405 LDA 3,SQTEM  
18 11553 000773 JMP .-5  
19 11554 161200 SQ1: MOVR 3,0  
20 11555 002401 JMP @SQ.S3  
21  
22 11556 000000 SQ.S3: 0  
23 11557 000000 SQTEM: 0

;FIND SQ ROOT OF C(AC0).  
;SEE THE SYSTEM REFFERANCE  
;MANUAL.  
;RESULT IN AC0.  
;AC1 DESTROYED.

;FIND SQ ROOT OF C(AC0).  
;SAME RESULT AS PREVIOUS  
;TEST BUT CODE IS  
;DIFFIRENT.

;RESULT WILL BE IN AC0.

```

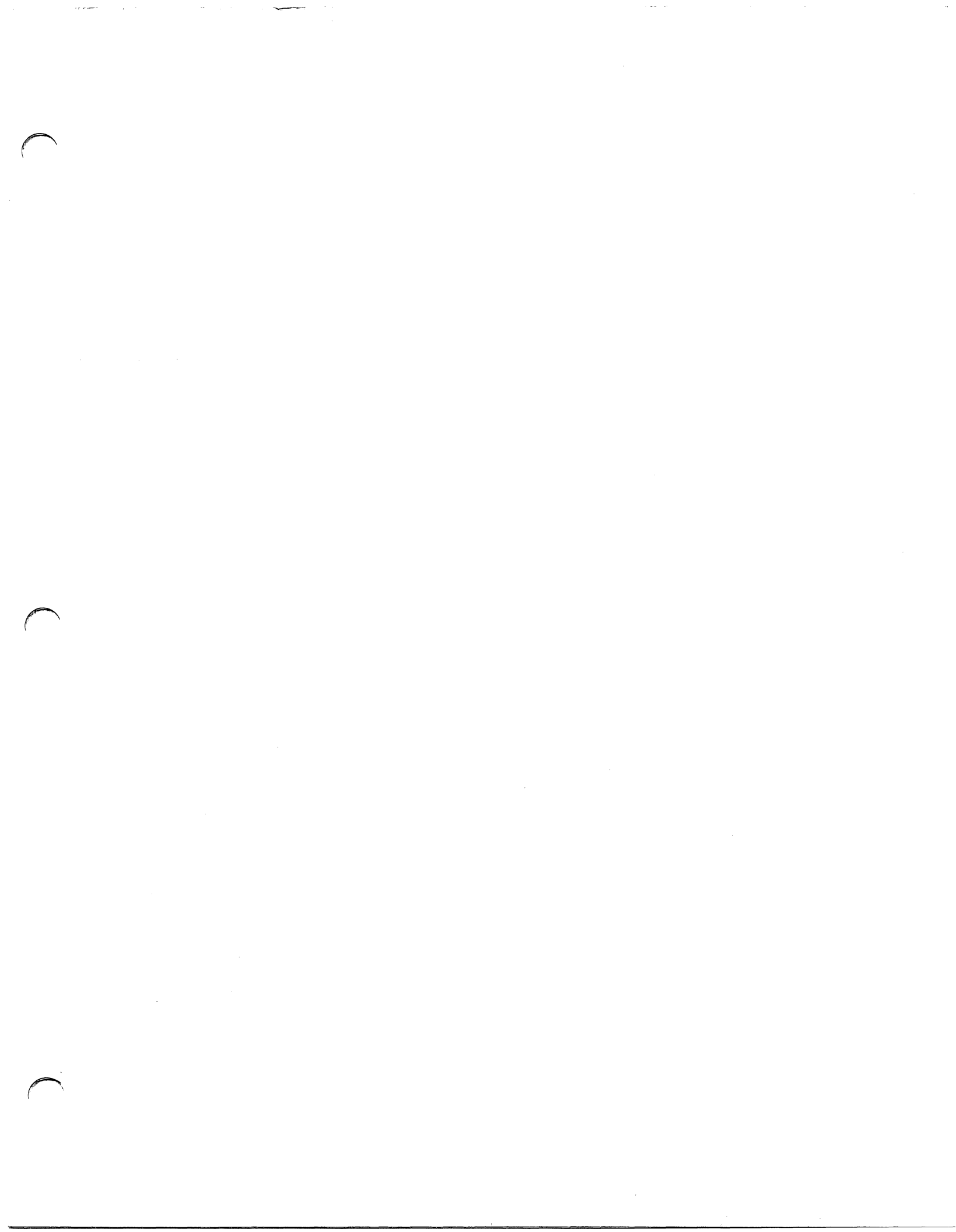
10159 N3MRT
01          AR4:  SETUP          ;THE CONTENTS OF AC2 IS
02          LCALL SETUL
03 11560 100410 SETUL-ASCRA*1B11+100010
04          AR4L: RANDOM        ;DIVIDED INTO AC0-1.
05          LCALL FRANG
06 11561 100450 FRANG-ASCRA*1B11+100010
07 11562 040471 STA 0,0AC0
08 11563 044471 STA 1,0AC1
09 11564 050471 STA 2,0AC2
10 11565 142432 SUBZ# 2,0,SZC ;THIS RESULT MULTIPLIED
11 11566 000773 JMP AR4L          ;BY AC2 SHOULD PRODUCE
12          CALL              ;THE ORIGINAL NUMBERS.
13          .DIVU
14 11567 004732 JSR DIVU
15          CALL
16          .MPYA
17 11570 004715 JSR MPYA
18 11571 034464 LDA 3,0AC2
19 11572 156414 SUB# 2,3,SZR
20          ERROR            ;AC2 CHANGED?
21 11573 004402 JSR .+2
22 11574 000402 JMP .+2
23          LCALL ERRET
24 11575 100470 ERRET-ASCRA*1B11+100010
25 11576 034456 LDA 3,0AC1
26 11577 136714 SUBS# 1,3,SZR
27          ERROR            ;AC1 CHANGED.
28 11600 004402 JSR .+2
29 11601 000402 JMP .+2
30          LCALL ERRET
31 11602 100470 ERRET-ASCRA*1B11+100010
32 11603 034450 LDA 3,0AC0
33 11604 116714 SUBS# 0,3,SZR
34          ERROR            ;AC0 CHANGED.
35 11605 004402 JSR .+2
36 11606 000402 JMP .+2
37          LCALL ERRET
38 11607 100470 ERRET-ASCRA*1B11+100010
39          LOOP
40          LCALL LLOOP
41 11610 100430 LLOOP-ASCRA*1B11+100010

```

```

10160 N3MRT
01          :ARS:
02          SETUP          ;USE INC TO FORM
03          LCALL SETUL
04 11611 100410 SETUL-ASCRA*1B11+100010
05 11612 152440 SUBO 2,2          ;THE NUMBER 177400
06 11613 151504 INCL 2,2,SZR ;IN AC1 AND 400 IN AC2.
07 11614 151504 INCL 2,2,SZR
08 11615 151504 INCL 2,2,SZR
09 11616 151504 INCL 2,2,SZR
10 11617 151507 INCL 2,2,SBN ;THE "SBN/SZR" SHOULD
11 11620 151507 INCL 2,2,SBN ;NOT CAUSE A SKIP.
12 11621 151507 INCL 2,2,SBN
13 11622 145707 INCS 2,1,SBN
14 11623 151407 INC 2,2,SBN
15 11624 151400 INC 2,2
16 11625 133014 ADD# 1,2,SZR
17          ERROR
18 11626 004402 JSR .+2
19 11627 000402 JMP .+2
20          LCALL ERRET
21 11630 100470 ERRET-ASCRA*1B11+100010
22          LOOP
23          LCALL LLOOP
24 11631 100430 LLOOP-ASCRA*1B11+100010

```



0163 N3MRT

01 11725 004402  
02 11726 000402  
03  
04 11727 100470  
05  
06  
07 11730 100430  
08

JSR .+2  
JMP .+2  
LCALL ERRET  
ERRET-ASCRA\*1B11+100010  
LOOP  
LCALL LLOOP  
LLOOP-ASCRA\*1B11+100010

10164 N3MRT

01  
02  
03  
04 11731 100410  
05 11732 102440  
06 11733 100762  
07  
08 11734 004402  
09 11735 000402  
10  
11 11736 100470  
12 11737 100762  
13  
14 11740 004402  
15 11741 000402  
16  
17 11742 100470  
18 11743 100706  
19  
20 11744 004402  
21 11745 000402  
22  
23 11746 100470  
24 11747 100706  
25  
26 11750 004402  
27 11751 000402  
28  
29 11752 100470  
30 11753 100544  
31  
32 11754 004402  
33 11755 000402  
34  
35 11756 100470  
36 11757 100403  
37  
38 11760 004402  
39 11761 000402  
40  
41 11762 100470  
42 11763 100644  
43  
44 11764 004402  
45 11765 000402  
46  
47 11766 100470  
48 11767 104704  
49  
50 11770 004402  
51 11771 000402  
52  
53 11772 100470  
54 11773 130704  
55  
56 11774 004402  
57 11775 000402  
58  
59 11776 100470  
60

!AR8:

SETUP ;NEGATION OF ZERO  
LCALL SETUL  
SETUL-ASCRA\*1B11+100010  
SUBO 0,0 ;SHOULD PRODUCE ZERO  
NEGCS 0,0,SZC ;AND A CARRY.  
ERROR  
JSR .+2  
JMP .+2  
LCALL ERRET  
ERRET-ASCRA\*1B11+100010  
NEGCS 0,0,SZC  
ERROR  
JSR .+2  
JMP .+2  
LCALL ERRET  
ERRET-ASCRA\*1B11+100010  
NEGS 0,0,SEZ  
ERROR  
JSR .+2  
JMP .+2  
LCALL ERRET  
ERRET-ASCRA\*1B11+100010  
NEGS 0,0,SEZ  
ERROR  
JSR .+2  
JMP .+2  
LCALL ERRET  
ERRET-ASCRA\*1B11+100010  
NEGOL 0,0,SZR  
ERROR  
JSR .+2  
JMP .+2  
LCALL ERRET  
ERRET-ASCRA\*1B11+100010  
NEG 0,0,SNC  
ERROR  
JSR .+2  
JMP .+2  
LCALL ERRET  
ERRET-ASCRA\*1B11+100010  
NEGOR 0,0,SZR  
ERROR  
JSR .+2  
JMP .+2  
LCALL ERRET  
ERRET-ASCRA\*1B11+100010  
NEGS 0,1,SZR  
ERROR  
JSR .+2  
JMP .+2  
LCALL ERRET  
ERRET-ASCRA\*1B11+100010  
NEGS 1,2,SZR  
ERROR  
JSR .+2  
JMP .+2  
LCALL ERRET  
ERRET-ASCRA\*1B11+100010  
LOOP

0165 N3MRT  
01  
02 11777 100430

LCALL LLOOP  
LLOOP-ASCRA\*1B11+100010

10166 N3MRT  
01  
02  
03  
04 12000 100410  
05  
06  
07 12001 100450  
08 12002 040651  
09 12003 152400  
10  
11 12004 004652  
12 12005 034646  
13 12006 166414  
14  
15 12007 004402  
16 12010 000402  
17  
18 12011 100470  
19  
20  
21 12012 100430  
22  
23  
24  
25  
26 12013 100410  
27 12014 102300  
28 12015 105705  
29 12016 130304  
30 12017 145705  
31 12020 124346  
32 12021 131707  
33 12022 150304  
34 12023 151704  
35  
36 12024 004402  
37 12025 000402  
38  
39 12026 100470  
40  
41  
42 12027 100430

;AR9:  
SETUP ;C(AC1)\*0+C(AC0) SHOULD  
LCALL SETUL  
SETUL-ASCRA\*1B11+100010  
RANDOM ;PLACE ACO IN AC1. SEE  
LCALL FRANG  
FRANG-ASCRA\*1B11+100010  
STA 0,OAC0  
SUB 2,2 ;SYSTEM REFFERENCE MANUAL  
CALL ;FOR FURTHER INFORMATION.  
JSR MPYAL  
LDA 3,OAC0  
SUB# 3,1,SZR  
ERROR  
JSR .+2  
JMP .+2  
LCALL ERRET  
ERRET-ASCRA\*1B11+100010  
LOOP  
LCALL LLOOP  
LLOOP-ASCRA\*1B11+100010  
;AR10:  
SETUP ;TEST "COM/INC" SWAPPED.  
LCALL SETUL  
SETUL-ASCRA\*1B11+100010  
ADCS 0,0  
INCS 0,1,SNR  
COMS 1,2,SZR  
INCS 2,1,SNR  
COMOS 1,1,SEZ  
INCS 1,2,SBN  
COMS 2,2,SZR  
INCS 2,2,SZR  
ERROR  
JSR .+2  
JMP .+2  
LCALL ERRET  
ERRET-ASCRA\*1B11+100010  
LOOP  
LCALL LLOOP  
LLOOP-ASCRA\*1B11+100010

```

10167 N3MRT
01
02          ;AR11:
03          SETUP          ;COMPLIMENT AND INCREMENT
04          LCALL SETUL
05          SETUL=ASCRA*1811+100010
06          RANDOM          ;SHOULD BE THE SAME AS
07          LCALL FRANG
08          FRANG=ASCRA*1811+100010
09          NEG 0,2          ;NEGATE
10          COM 0,1
11          INC 1,1
12          SUB# 1,2,SZR
13          ERROR
14          JSR .+2
15          JMP .+2
16          LCALL ERRET
17          ERRET=ASCRA*1811+100010
18          LOOP
19          LCALL LLOOP
20          LLOOP=ASCRA*1811+100010
21
22          ;AR12:
23          SETUP          ;TEST COM LEFT AND RIGHT.
24          LCALL SETUL
25          SETUL=ASCRA*1811+100010
26          RANDOM
27          LCALL FRANG
28          FRANG=ASCRA*1811+100010
29          COML 0,2
30          COMCR 2,1
31          SUBS# 0,1,SZR
32          ERROR
33          JSR .+2
34          JMP .+2
35          LCALL ERRET
36          ERRET=ASCRA*1811+100010
37          LOOP
38          LCALL LLOOP
39          LLOOP=ASCRA*1811+100010

```

```

10168 N3MRT
01
02          ;AR13:
03          SETUP          ;FORM THE PARITY OF
04          LCALL SETUL
05          SETUL=ASCRA*1811+100010
06          MOVZ 0,1          ;C(AC0) IN DIFFIRENT
07          ADC 3,3          ;ROUTINES. CHECK THAT
08          ADD 0,3          ;THE RESULTS ARE EQUAL.
09          ANDS 3,0,SZR
10          JMP .-3
11          SUBCR 0,0          ;SAVE PARITY IN BIT 0.
12          SUBZR 3,3
13          MOVL 1,1,SZC
14          INC 0,0          ;BIT 15 WILL CONTAIN
15          MOVZR 3,3,SZR    ;THE PARITY.
16          JMP .-3
17          MOVR 0,3          ;CHECK TO SEE IF BITS
18          ADD# 0,0,SZC     ;ARE LIKE.
19          ERROR
20          JSR .+2
21          JMP .+2
22          LCALL ERRET
23          ERRET=ASCRA*1811+100010
24          LOOP
25          LCALL LLOOP
26          LLOOP=ASCRA*1811+100010
27
28          ;AR15:
29          SETUP          ;MISC TEST OF SUB LEFT
30          LCALL SETUL
31          SETUL=ASCRA*1811+100010
32          RANDOM          ;AND RIGHT.
33          LCALL FRANG
34          FRANG=ASCRA*1811+100010
35          MOV 1,2
36          SUBL 0,1
37          ANDR 1,1
38          ADD 0,1
39          SUBL 0,1
40          ANDR 1,1
41          ADD 0,1
42          SUB# 1,2,SZR
43          ERROR
44          JSR .+2
45          JMP .+2
46          LCALL ERRET
47          ERRET=ASCRA*1811+100010
48          LOOP
49          LCALL LLOOP
50          LLOOP=ASCRA*1811+100010

```

10169 N3MRT

```
01          ;AR16:  SETUP                ;MISC TEST OF AND RIGHT.
02          LCALL SETUL
03          SETUL=ASCRA*1B11+100010
04 12113 100410  RANDOM
05          LCALL FRANG
06          FRANG=ASCRA*1B11+100010
07 12114 100450  MOV 1,2
08 12115 131040  ANDR 0,2
09 12116 113600  AND 0,1
10 12117 107400  MOVL 2,2
11 12120 151100  SUB# 2,1,SZR
12 12121 146414  ERROR
13          JSR .+2
14 12122 004402  JMP .+2
15 12123 000402  LCALL ERRET
16          ERRET=ASCRA*1B11+100010
17 12124 100470  LOOP
18          LCALL LLOOP
19          LLOOP=ASCRA*1B11+100010
20 12125 100430
```

10170 N3MRT

```
01          ;AR17:  SETUP                ;MISC ADD SWAPPED TEST.
02          LCALL SETUL
03          SETUL=ASCRA*1B11+100010
04 12126 100410  RANDOM
05          LCALL FRANG
06          FRANG=ASCRA*1B11+100010
07 12127 100450  MOV 1,2
08 12130 131000  ADD 0,2
09 12131 113000  ADDS 0,1
10 12132 107300  MOVS 1,1
11 12133 125300  SUB# 1,2,SZR
12 12134 132414  ERROR
13          JSR .+2
14 12135 004402  JMP .+2
15 12136 000402  LCALL ERRET
16          ERRET=ASCRA*1B11+100010
17 12137 100470  LOOP
18          LCALL LLOOP
19          LLOOP=ASCRA*1B11+100010
20 12140 100430
21
22          ;AR18:  SETUP                ;CHECK ADC LEFT.
23          LCALL SETUL
24          SETUL=ASCRA*1B11+100010
25 12141 100410  RANDOM
26          LCALL FRANG
27          FRANG=ASCRA*1B11+100010
28 12142 100450  MOV 1,2
29 12143 131000  ADCZL 0,2
30 12144 112120  COM 0,0
31 12145 100000  ADDZL 0,1
32 12146 107120  SUB# 1,2,SZR
33 12147 132414  ERROR
34          JSR .+2
35 12150 004402  JMP .+2
36 12151 000402  LCALL ERRET
37          ERRET=ASCRA*1B11+100010
38 12152 100470  LOOP
39          LCALL LLOOP
40          LLOOP=ASCRA*1B11+100010
41 12153 100430
```

```

10171 N3MRT
01
02      ;AR19:
03      SETUP          ;TEST ADC RIGHT.
04      LCALL SETUL
05      SETUL-ASCRA*1B11+100010
06      RANDOM
07      LCALL FRANG
08      FRANG-ASCRA*1B11+100010
09      MOV 1,2
10      ADCZR 0,2
11      COM 0,0
12      ADDZR 0,1
13      SUB# 1,2,SZR
14      ERROR
15      JSR .+2
16      JMP .+2
17      LCALL ERRET
18      ERRET-ASCRA*1B11+100010
19      LOOP
20      LCALL LLOOP
21      LLOOP-ASCRA*1B11+100010
22
23      ;AR20:
24      SETUP          ;TEST SUB RIGHT.
25      LCALL SETUL
26      SETUL-ASCRA*1B11+100010
27      RANDOM
28      LCALL FRANG
29      FRANG-ASCRA*1B11+100010
30      MOV 1,2
31      SUBZR 0,1
32      NEGZ 0,0
33      ADDR 0,2
34      SUB# 1,2,SZR
35      ERROR
36      JSR .+2
37      JMP .+2
38      LCALL ERRET
39      ERRET-ASCRA*1B11+100010
40      LOOP
41      LCALL LLOOP
42      LLOOP-ASCRA*1B11+100010

```

```

10172 N3MRT
01
02      ARJSR: SETUP          ;TEST THAT INDEX WITH
03      LCALL SETUL
04      SETUL-ASCRA*1B11+100010
05      JSR .+1          ;SIGN BIT SET DOES
06      MOVZL 3,2,SZC    ;NOT LOAD INTO PC ON JSR.
07      ERROR
08      JSR .+2
09      JMP .+2
10      LCALL ERRET
11      ERRET-ASCRA*1B11+100010
12      MOVOR 2,2
13      JSR 7,2          ;GO TO NEXT LOCATION
14      JSR .+1
15      MOV 3,1          ;JSR SHOULD NEVER
16      MOVL# 1,1,SZC    ;STORE THE SIGN BIT.
17      ERROR
18      JSR .+2
19      JMP .+2
20      LCALL ERRET
21      ERRET-ASCRA*1B11+100010
22      LOOP
23      LCALL LLOOP
24      LLOOP-ASCRA*1B11+100010
25
26      AREND: LCALL RETU2
27      RETU2-ASCRA*1B11+100010
28
29      ATEND: 0

```



10173 N3MRT

```
01      .TITL FPTST
02      ;FLT POINT UNIT TEST TO RUN WITH
03      ;THE DIAGNOSTIC LINKER
04      ;DEFINE FLT PT INSTRUCTIONS
05      000076 .DUSR FPU=76
06      000075 .DUSR FPU2=FPU-1
07      000074 .DUSR FPU1=FPU-2
08      060375 .DUSR FMFT=NIOP FPU2
09      060275 .DUSR FMFT=NIOP FPU2
10      062374 .DIAC FLOODS=DOBP 0,FPU1
11      062375 .DIAC FLOOD=DOBP 0,FPU2
12      062174 .DIAC FSTRS=DOBS 0,FPU1
13      062175 .DIAC FSTRD=DOBS 0,FPU2
14      060676 .DIAC FPRST=DIAC 0,FPU
15      061076 .DIAC FPLST=DDA 0,FPU
```

10174 N3MRT

```
01      ;TEST INTERFACE TO LINKER
02      NEXTT F.000
03      012223 LMEML=.
04      000163 .LOC LPG0
05      00163 012226 F.000
06      000164 LPG0=.
07      012223 .LOC LMEML
08      12223 000000 0 ;TEST PASS CTR
09      12224 000000 0 ;TEST ERROR CTR
10      12225 000000 0 ;INTERRUPT TIMEOUT SWITCH
11      12226 012245 F.000: FP.01
12      12227 012261 FP.02
13      12230 000000 0
14      12231 000000 0
15      12232 177777 -1
16      12233 176000 176000
17      12234 012500 FP.EC
18      12235 012500 FP.EC
19      012236 .TXTE (
20      12236 146306 FLT POINT TST(
21      120324
22      147520
23      047311
24      120324
25      051724
26      000324
27      ;TEST INITIALIZE ROUTINE
28      ;DETERMINE IF FPU EXISTS
29      ;ENABLE OR DISABLE TEST ACCORDINGLY
30      12245 102000 FP.01: ADC 0,0
31      12246 040762 STA 0,F.000+2 ;DISABLE TEST ENTER
32      12247 104400 NEG 0,1
33      12250 065076 FPLST 1 ;SETS DIAG MODE
34      12251 064676 FPRST 1 ;READ STATUS BACK
35      12252 124405 NEG 1,1,SNR ;SKP IF FPU
36      12253 001400 JMP 0,3 ;EXIT TEST DISABLED
37      12254 040464 STA 0,FP.TK ;SET NO SCRA SW
38      12255 102400 SUB 0,0
39      12256 040752 STA 0,F.000+2
40      12257 061076 FPLST 0
41      12260 001400 JMP 0,3
```

10175 N3MRT

```
01 ;
02 ;EXECUTE ENTRY POINT
03 12261 010457 FP.02: ISZ FP.TK ;SKP IS NO SCRATCH
04 12262 000464 JMP FP.03 ;DO NXT IN SEQ
05 12263 020155 LDA 0,SCRHI
06 12264 101004 MOV 0,0,SZR
07 12265 000422 JMP FP.2A
08 12266 102400 SUB 0,0
09 12267 040456 STA 0,FP.KK ;0 # EXTRA 1K'S
10 LCALL ASCRA ;TRY TO GET 1K
11 12270 100010 ASCRA-ASCRA*1811+100010
12 12271 000565 JMP FP.05 ;NONE AVAILABLE
13 12272 102000 ADC 0,0 ;-1 TO
14 12273 040447 STA 0,FP.ES ;NO ERRSW
15 LCALL ARANG ;GET RAN#
16 12274 100270 ARANG-ASCRA*1811+100010
17 12275 030444 LDA 2,FP.37
18 12276 105000 MOV 0,1
19 LCALL ADIVI ;REM=#1K'S TO EXPAND
20 12277 100310 ADIVI-ASCRA*1811+100010
21 12300 100405 NEG 0,0,SNR
22 12301 000406 JMP FP.2A
23 FP.2L: LCALL ESCRA ;EXPAND SCRATCH 1K
24 12302 100030 ESCRA-ASCRA*1811+100010
25 12303 000404 JMP FP.2A ;NO MORE AVAILABLE
26 12304 010441 ISZ FP.KK ;+1 # EXTRA 1K'S
27 12305 101404 INC 0,0,SZR
28 12306 000774 JMP FP.2L ;KEEP EXPANDING
```

10176 N3MRT

```
01 ;SCRATCH AREA HAS BEEN ASSIGNED RANDOMLY SEL
02 ;WHER TO MOVE TEST WITHIN SCRATCH AREA
03 ;SCRLO+1 TO SCRLO+400 DEPENDS ON RELATIONSHIP OF LAST
04 ;RANDOM # TO SCRATCH AREA
05 12307 030432 FP.2A: LDA 2,FP.37
06 12310 150400 NEG 2,2
07 LCALL ARANG ;GET RAN #
08 12311 100270 ARANG-ASCRA*1811+100010
09 12312 030432 LDA 2,FP255
10 12313 105000 MOV 0,1
11 LCALL ADIVI ;RET WITH ACO=DISPLACEMENT
12 12314 100310 ADIVI-ASCRA*1811+100010
13 12315 024154 LDA 1,SCRLO
14 12316 125400 INC 1,1
15 12317 123000 ADD 1,0
16 12320 040545 FP.2C: STA 0,FP.LC ;ADRESS TO RELOC TESTS
17 12321 000425 JMP FP.03 ;INIT FIRST TEST
18 ;RELOCATE TEST TO SELECTED SCRATCH AREA
19 12322 144000 FP.RL: COM 2,1 ;-#WORDS TO MOV
20 12323 111000 MOV 0,2 ;TO ADRS
21 12324 034543 LDA 3,FP.BG ;FROM ADRS
22 12325 020535 LDA 0,FP.HI ;GET HI REL SCR
23 12326 041000 STA 0,0,2
24 12327 151400 INC 2,2 ;HIGH LIMIT FOR TEST
25 12330 021400 FP.L2: LDA 0,0,3
26 12331 041000 STA 0,0,2
27 12332 175400 INC 3,3
28 12333 151400 INC 2,2
29 12334 125404 INC 1,1,SZR
30 12335 000773 JMP FP.L2 ;MOV ALL TO SCRATCH
31 12336 050532 STA 2,FP.EN ;SAVE LAST LOC+1
32 12337 000452 JMP FP.G0
33 12340 000000 FP.TK: 0
34 12341 000037 FP.37: 37
35 12342 000000 FP.ES: 0
36 12343 001777 FK1K: 1777
37 12344 000377 FP255: 255.
38 12345 000000 FP.KK: 0
```

10177 N3MRT

```
01 ;TESTS MOVED TO SCRATCH
02 ;FP.03 SELECTS LOGICAL PAGE ASSIGNMENT
03 ;MOVES NEXT TEST TO SCRATCH VIA FP.RL
04 ;RETURNS TO FP.GO
05 ;AND INITIATES TEST VIA GSCRA
06
07 12346 020774 FP.03: LDA 0,FP.ES ;GET ERR SWITCH
08 12347 101005 MOV 0,0,SNR ;SKIP IS NO ERR
09 12350 000441 JMP FP.GO ;USE PREV ASSIGN
10 LCALL ARANG
11 12351 100270 ARANG-ASCRA*1B11+100010
12 12352 105000 MOV 0,1
13 12353 030510 LDA 2,FP.40
14 12354 034771 LDA 3,FP.KK
15 12355 172400 SUB 3,2 ;DONT EXPAND ABOVE 32K
16 LCALL ADIVI
17 12356 100310 ADIVI-ASCRA*1B11+100010
18 12357 040514 STA 0,FP.LP
19 12360 024505 LDA 1,FP.LC
20 12361 030154 LDA 2,SCRLO
21 12362 146400 SUB 2,1
22 12363 111300 MOVS 0,2
23 12364 153120 ADDZL 2,2
24 12365 050504 STA 2,FP.LO
25 12366 034757 LDA 3,FP.KK ;RECALC SCRHI
26 12367 175300 MOVS 3,3
27 12370 177120 ADDZL 3,3
28 12371 157000 ADD 2,3
29 12372 020751 LDA 0,FK1K
30 12373 117000 ADD 0,3
31 12374 054466 STA 3,FP.HI
32 12375 133220 ADDZR 1,2
33 12376 151500 INCL 2,2
34 12377 050462 STA 2,FP.GA
35 12400 020740 LDA 0,FP.TK
36 12401 034545 LDA 3,FP.TT
37 12402 117000 ADD 0,3
38 12403 031400 LDA 2,0,3
39 12404 050463 STA 2,FP.BG
40 12405 020460 LDA 0,FP.LC
41 12406 031000 LDA 2,0,2
42 12407 054457 STA 3,FP.TP
43 12410 000712 JMP FP.RL
44 12411 020462 FP.GO: LDA 0,FP.LP ;REMAP SCR TO HERE
45 12412 030447 LDA 2,FP.GA ;STARTING LOGICAL ADRS
46 12413 024461 LDA 1,FPERR ;1=ERROR RET ADRS
47 LCALL GSCRA ;GO TO SCRFPC
48 12414 100070 GSCRA-ASCRA*1B11+100010
49 12415 000427 JMP FP.GO ;COULDNT REMAP GO DIRECT
```

10178 N3MRT

```
01 ;RETURN TO NEXT LOC PASS COMPLETE NO ERRS
02 12416 010450 ISZ FP.TP
03 12417 022447 LDA 0,FP.TP ;GET ADRS NXT TST
04 12420 100004 COM 0,0,SZR ;=-1 IS END OF SEQ
05 12421 000437 JMP FP.XI
06 12422 020230 LDA 0,SWREG ;CHK IF REL. SCRATCH
07 12423 103123 ADDZL 0,0,SNC ;SKP IS NOT REL.
08 12424 000406 JMP .+6
09 12425 020715 LDA 0,FP.ES ;CHK FOR PREV ERR
10 12426 101004 MOV 0,0,SZR ;SKP ON PREV ERR
11 12427 000403 JMP .+3 ;RELEASE ASSIGN
12 12430 040710 STA 0,FP.TK
13 12431 000427 JMP FP.XI
14 LCALL ARANG ;IF BITS 0 AND 1=1
15 12432 100270 ARANG-ASCRA*1B11+100010
16 12433 103043 ADDO 0,0,SNC ;RELEASE AND REMAP
17 12434 103003 ADD 0,0,SNC ;IF EITHER =0 MAKE
18 12435 000404 JMP FP.04 ;NEXT PASS SELECT NEW LP
19 FP.4A: LCALL RSCRA
20 12436 100050 RSCRA-ASCRA*1B11+100010
21 12437 000417 JMP FP.05 ;SET NO SCRFPC
22 12440 000776 JMP FP.4A
23 12441 102400 FP.04: SUB 0,0
24 12442 040676 STA 0,FP.TK
25 12443 000415 JMP FP.XI
26 ;MAP OPTION DOES NOT EXIST GO DIRECT
27 ;AFTER READJUSTING SCRHI PARAMETER
28 12444 020155 FP.GD: LDA 0,SCRHI
29 12445 040415 STA 0,FP.HI
30 12446 042417 STA 0,FP.LC
31 12447 034416 LDA 3,FP.LC
32 12450 020154 LDA 0,SCRLO
33 12451 040420 STA 0,FP.LO
34 12452 175400 INC 3,3
35 12453 175400 INC 3,3
36 12454 054405 STA 3,FP.GA
37 12455 001400 JMP 0,3
38 ;SET SCRATCH RELEASED SW AND EXIT
39 12456 102000 FP.05: ADC 0,0
40 12457 040661 STA 0,FP.TK
41 FP.XI: LCALL RETRN
42 12460 100210 RETRN-ASCRA*1B11+100010
43 12461 000000 FP.GA: 0
44 12462 000000 FP.HI: 0
45 12463 000040 FP.40: 040
46 12464 000000 FPS03: 0
47 12465 000000 FP.LC: 0
48 12466 000000 FP.TP: 0
49 12467 000000 FP.BG: 0
50 12470 000000 FP.EN: 0
51 12471 000000 FP.LO: 0
52 12472 000000 FP.LA: 0
53 12473 000000 FP.LP: 0
```

```

10179 N3MRT
01          ;ERROR IN TEST DURING EXECUTION
02
03 12474 012475 FPERR:  FPERR+1
04 12475 054767          STA 3,FP$03          ;PRINT ERR HEADERS
05                      LCALL ERROI
06 12476 100350          ERROI-ASCRA*1811+100010
07 12477 000401          JMP .+1
08 12500 020454 FP.EC:  LDA 0,F.TX1
09                      LCALL ERRTX
10 12501 100170          ERRTX-ASCRA*1811+100010
11 12502 020763          LDA 0,FP.LC          ;FOLLOW UP WITH
12 12503 024766          LDA 1,FP.LO          ;TEST RELOC INFO
13 12504 030764          LDA 2,FP.EN
14                      LCALL ERROC
15 12505 100370          ERROC-ASCRA*1811+100010
16 12506 000401          JMP .+1
17 12507 020460          LDA 0,F.TX2
18                      LCALL ERRTX
19 12510 100170          ERRTX-ASCRA*1811+100010
20 12511 020753          LDA 0,FP$03          ;CONTINUE TYPE
21 12512 024761          LDA 1,FP.LP          ;RELOC
22 12513 030627          LDA 2,FP.ES          ;=0 PREV ERR
23                      LCALL ERROC
24 12514 100370          ERROC-ASCRA*1811+100010
25 12515 000401          JMP .+1
26 12516 020464          LDA 0,F.TX3
27                      LCALL ERRTX
28 12517 100170          ERRTX-ASCRA*1811+100010
29 12520 020742          LDA 0,FP.HI          ;HI SCRATCH REL.
30 12521 024746          LDA 1,FP.BG          ;TST BEFORE MOVE
31 12522 030737          LDA 2,FP.GA          ;WHERE STARTS REL.
32                      LCALL ERROC
33 12523 100370          ERROC-ASCRA*1811+100010
34 12524 000401          JMP .+1
35 12525 020470          LDA 0,F.TX4
36                      LCALL ERRTX
37 12526 100170          ERRTX-ASCRA*1811+100010
38 12527 034741          LDA 3,FP.EN
39 12530 021776          LDA 0,-2,3
40 12531 024740          LDA 1,FP.LO
41 12532 111000          MOV 0,2          ;RECALCULATE ADRS
42 12533 132400          SUB 1,2          ;OF FAILED DATA
43 12534 024154          LDA 1,SCRLO          ;IN RELATION TO
44 12535 133000          ADD 1,2          ;SCRATCH AT 1ST LEVEL
45 12536 101004          MOV 0,0,SZR          ;NOT DATA ERR SKP
46 12537 025000          LDA 1,0,2
47                      LCALL ERROC
48 12540 100370          ERROC-ASCRA*1811+100010
49 12541 000675          JMP FP.4A          ; RELEASE SCRATCH
50 12542 102400          SUB 0,0
51 12543 042407          STA 0,@FP$SX          ;SET ERR SW
52 12544 042407          STA 0,@FPTKX          ;CLR TEST K
53 12545 000713          JMP FP.XI          ;RETURN TO LINKR
54 12546 012547 FP.TT:  FP.TT+1
55 12547 012631          FLT01
56 12550 012776          FLT02
57 12551 177777          -1
58 12552 012342 FP$SX:  FP.ES
59 12553 012340 FPTKX:  FP.TK

```

```

10180 N3MRT
01 12554 012555 F.TX1:  .+1
02 12555 005215          .TXTE (<15><12>FP.LC  FP.LO  FP.EN(
03 12567 012570 F.TX2:  .+1          .TXTE (<15><12>FP$03  FP.LP  FP.ES(
04 12570 005215          .TXTE (<15><12>FP.HI  FP.BG  FP.GA(
05 12602 012603 F.TX3:  .+1          .TXTE (<15><12>L ADRS  DATA  S ADRS(
06 12603 005215          .TXTE (<15><12>FP.HI  FP.BG  FP.GA(
07 12615 012616 F.TX4:  .+1
08 12616 005215          .TXTE (<15><12>L ADRS  DATA  S ADRS(

```

```

10181 N3MRT
01          ;FLOATING POINT LOAD AND STORE TEST
02          ;SINGLE PRECISION OPERANDS
03
04 12631 000144 FLT01: FLT1E-FLT01
05 12632 004401      JSR .+1
06 12633 020776      LDA 0,FLT01
07 12634 117000      ADD 0,3
08 12635 054534      STA 3,FLT1A      ;LOAD DATA START
09 12636 024772      LDA 1,FLT01-1    ;GET UPPER LOG ADDRESS
10 12637 166640      SUBOR 3,1        ;HALF SCRATCH 2 WORDS EACH
11 12640 121220      MOVZR 1,0       ;1 FOURTH SCRATCH
12 12641 117000      ADD 0,3
13 12642 054530      STA 3,FLT1B      ;START OF STORE BUFFER
14 12643 040530      STA 0,FLT1C
15 12644 105000      MOV 0,1
16 12645 030524      LDA 2,FLT1A
17 12646 124400      NEG 1,1
18          FLT1L:  LCALL ARANG      ;FILL 1 FOURTH
19 12647 100270      ARANG=ASCRA*1B11+100010
20 12650 041000      STA 0,0,2       ;OF BUFFER
21 12651 151400      INC 2,2        ;WITH RANDOM #'S
22 12652 125404      INC 1,1,SZR
23 12653 000774      JMP FLT1L
24 12654 020515      LDA 0,FLT1A      ;FILL BUFFER
25 12655 024515      LDA 1,FLT1B      ;1ST PAIR
26 12656 030515      LDA 2,FLT1C      ;MOVE 1ST TO TEMP
27 12657 150640      NEGOR 2,2       ;STORE 2ND PAIR
28 12660 151400      INC 2,2
29 12661 176520      SUBZL 3,3       ;MOVE 1ST PAIR BACK
30 12662 175120      MOVZL 3,3       ;AND STORE THEM
31 12663 062374      FLT1M:  FLOODS 0      ;LOAD FIRST 2 RAN #'S
32 12664 163000      ADD 3,0
33 12665 066174      FSTRS 1        ;STORE THEM BACK
34 12666 167000      ADD 3,1
35 12667 060375      FMFT          ;SAVE IN TEMP
36 12670 000401      JMP .+1
37 12671 062374      FLOODS 0       ;GET NEXT 2 RAN #'S
38 12672 000401      JMP .+1
39 12673 066174      FSTRS 1
40 12674 167000      ADD 3,1
41 12675 060275      FMTF          ;RESTORE 1ST 2 RAN #'S
42 12676 000401      JMP .+1
43 12677 066174      FSTRS 1        ;STORE THEM AGAIN
44 12700 167000      ADD 3,1
45 12701 151404      INC 2,2,SZR
46 12702 000761      JMP FLT1M

```

```

10182 N3MRT
01          ;COMPARE SINGLE PREC STORE COMPARE
02
03 12703 030466      LDA 2,FLT1A
04 12704 034466      LDA 3,FLT1B
05 12705 024466      LDA 1,FLT1C
06 12706 124640      NEGOR 1,1
07 12707 125400      INC 1,1
08 12710 044465      STA 1,FLT1E
09 12711 021000      LDA 0,0,2      ;GET FIRST WORD OF PAIR
10 12712 025400      LDA 1,0,3      ;GET FIRST STORE
11 12713 122414      SUB# 1,0,SZR   ;NOT=IS ERR
12 12714 000445      JMP FLT1D
13 12715 010453      ISZ F1WDS
14 12716 025404      LDA 1,4,3      ;GET 2ND TIME STORED
15 12717 122414      SUB# 1,0,SZR
16 12720 000441      JMP FLT1D
17 12721 014447      DSZ F1WDS
18 12722 063077      HALT
19 12723 151400      INC 2,2
20 12724 175400      INC 3,3
21 12725 021000      LDA 0,0,2      ;GET 2ND WORD OF PAIR
22 12726 025400      LDA 1,0,3      ;FIRST TIME IT STORED
23 12727 122414      SUB# 1,0,SZR
24 12730 000431      JMP FLT1D
25 12731 010437      ISZ F1WDS
26 12732 025404      LDA 1,4,3      ;2ND TIME 2ND WORD
27 12733 122414      SUB# 1,0,SZR
28 12734 000425      JMP FLT1D
29 12735 014433      DSZ F1WDS
30 12736 063077      HALT
31 12737 151400      INC 2,2
32 12740 175400      INC 3,3
33 12741 021000      LDA 0,0,2      ;FIRST WORD 2ND PAIR
34 12742 025400      LDA 1,0,3
35 12743 122414      SUB# 1,0,SZR
36 12744 000415      JMP FLT1D
37 12745 021001      LDA 0,1,2      ;2ND WORD 2ND PAIR
38 12746 025401      LDA 1,1,3
39 12747 106414      SUB# 0,1,SZR
40 12750 000407      JMP FLT1D-2
41 12751 175620      INCZR 3,3      ;+3 TO STORE
42 12752 175500      INCL 3,3      ;BUFFER ADDRESS
43 12753 175400      INC 3,3
44 12754 010421      ISZ FLT1E      ;DONE ALL SKIP
45 12755 000734      JMP FLT1T
46          LCALL RETU2      ;TEST PASSED RETURN
47 12756 100510      RETU2=ASCRA*1B11+100010

```

```

10183 N3MRT
01          :ERROR RETURN FROM THIS TEST
02
03 12757 151400      INC 2,2
04 12760 175400      INC 3,3
05 12761 040413 FLT1D: STA 0,FLT1F
06 12762 020406      LDA 0,F1WDS
07 12763 103120      ADDZL 0,0
08 12764 117000      ADD 0,3
09 12765 020407      LDA 0,FLT1F
10 12766 054406      STA 3,FLT1F      ;SAVE REAL ADRS OF ERR
11          LCALL ERRET
12 12767 100470      ERRET-ASCRA*1811+100010
13
14 12770 000000 F1WDS: 0
15 12771 000000 FLT1A: 0
16 12772 000000 FLT1B: 0
17 12773 000000 FLT1C: 0
18 12774 000000 FLT1F: 0
19 12775 000000 FLT1E: 0

```

```

10184 N3MRT
01          :FLOATING POINT LOAD AND STORE TEST
02          :DOUBLE PRECISION OPERANDS
03
04 12776 000166 FLT02: FLT2E-FLT02
05 12777 004401      JSR .+1
06 13000 020776      LDA 0,FLT02
07 13001 117000      ADD 0,3
08 13002 054556      STA 3,FLT2A      ;LOAD DATA START
09 13003 024772      LDA 1,FLT02-1    ;GET UPPER LOG ADDRESS
10 13004 166640      SUBOR 3,1        ;HALF SCRATCH 2 WORDS EACH
11 13005 121220      MOVZR 1,0        ;1 FOURTH SCRATCH
12 13006 117000      ADD 0,3
13 13007 054552      STA 3,FLT2B      ;START OF STORE BUFFER
14 13010 040552      STA 0,FLT2C
15 13011 105000      MOV 0,1
16 13012 030546      LDA 2,FLT2A
17 13013 124400      NEG 1,1
18          FLT2L: LCALL ARANG      ;FILL 1 FOURTH
19 13014 100270      ARANG-ASCRA*1811+100010
20 13015 041000      STA 0,0,2      ;OF BUFFER
21 13016 151400      INC 2,2        ;WITH RANDOM #'S
22 13017 125404      INC 1,1,SZR
23 13020 000774      JMP FLT2L
24 13021 020537      LDA 0,FLT2A      ;FILL BUFFER
25 13022 024537      LDA 1,FLT2B      ;IST 4
26 13023 030537      LDA 2,FLT2C      ;MOVE 1ST TO TEMP
27 13024 150640      NEGOR 2,2      ;STORE 2ND 4
28 13025 151240      MOVOR 2,2
29 13026 151400      INC 2,2
30 13027 176520      SUBZL 3,3      ;MOVE 1ST 4 BACK
31 13030 177120      ADDZL 3,3      ;AND STORE THEM
32 13031 062375 FLT2M: FLOOD 0      ;LOAD FIRST 4 RAN #'S
33 13032 163000      ADD 3,0
34 13033 066175      FSTRD 1      ;STORE THEM BACK
35 13034 167000      ADD 3,1
36 13035 060375      FMFT      ;SAVE IN TEMP
37 13036 000401      JMP .+1
38 13037 062375      FLOOD 0      ;GET NEXT 4 RAN #'S
39 13040 000401      JMP .+1
40 13041 066175      FSTRD 1
41 13042 167000      ADD 3,1
42 13043 060275      FMFT      ;RESTORE 1ST 4 RAN #'S
43 13044 000401      JMP .+1
44 13045 066175      FSTRD 1      ;STORE THEM AGAIN
45 13046 167000      ADD 3,1
46 13047 151404      INC 2,2,SZR
47 13050 000761      JMP FLT2M

```

```

10185 N3MRT
01 ;COMPARE DOUBLE PREC STORE COMPARE
02
03 13051 030507 LDA 2,FLT2A
04 13052 034507 LDA 3,FLT2B
05 13053 024507 LDA 1,FLT2C
06 13054 124640 NEGOR 1,1
07 13055 125240 MOVOR 1,1
08 13056 125400 INC 1,1
09 13057 044505 STA 1,FLT2E
10 13060 102120 FLT2T: ADCZL 0,0
11 13061 040475 STA 0,FLT2S
12 13062 021000 LDA 0,0,2 ;GET FIRST WORD OF 4
13 13063 025400 LDA 1,0,3 ;GET FIRST STORE
14 13064 122414 SUB# 1,0,SZR ;NOT=IS ERR
15 13065 000461 JMP FLT2D
16 13066 010471 ISZ F2WDS
17 13067 025410 LDA 1,10,3 ;GET 2ND TIME STORED
18 13070 122414 SUB# 1,0,SZR
19 13071 000455 JMP FLT2D
20 13072 014465 DSZ F2WDS
21 13073 063077 HALT
22 13074 151400 INC 2,2
23 13075 175400 INC 3,3
24 13076 021000 LDA 0,0,2 ;GET 2ND WORD OF 4
25 13077 025400 LDA 1,0,3 ;FIRST TIME IT STORED
26 13100 122414 SUB# 1,0,SZR
27 13101 000445 JMP FLT2D
28 13102 010455 ISZ F2WDS
29 13103 025410 LDA 1,10,3 ;2ND TIME 2ND WORD
30 13104 122414 SUB# 1,0,SZR
31 13105 000441 JMP FLT2D
32 13106 014451 DSZ F2WDS
33 13107 063077 HALT
34 13110 151400 INC 2,2
35 13111 175400 INC 3,3
36 13112 010444 ISZ FLT2S
37 13113 000747 JMP FLT2T+2
38 13114 102120 ADCZL 0,0
39 13115 040441 STA 0,FLT2S

```

```

10186 N3MRT
01 13116 021000 FLT2U: LDA 0,0,2 ;FIRST WORD 2ND 4
02 13117 025400 LDA 1,0,3
03 13120 122414 SUB# 1,0,SZR
04 13121 000425 JMP FLT2D
05 13122 021001 LDA 0,1,2 ;2ND WORD 2ND 4
06 13123 025401 LDA 1,1,3
07 13124 106414 SUB# 0,1,SZR
08 13125 000417 JMP FLT2D-2
09 13126 102520 SUBZL 0,0
10 13127 101120 MOVZL 0,0
11 13130 113000 ADD 0,2
12 13131 117000 ADD 0,3
13 13132 010424 ISZ FLT2S
14 13133 000763 JMP FLT2U ;DO NXT 2 2ND 4
15 13134 112400 SUB 0,2
16 13135 112400 SUB 0,2
17 13136 175400 INC 3,3
18 13137 175620 INCZR 3,3 ;+5 TO STORE
19 13140 175500 INCL 3,3 ;BUFFER ADDRESS
20 13141 010423 ISZ FLT2E ;DONE ALL SKIP
21 13142 000716 JMP FLT2T
22 LCALL RETU2 ;TEST PASSED RETURN
23 13143 100510 RETU2=ASCRA*1B11+100010
24
25 ;ERROR RETURN FROM THIS TEST
26
27 13144 151400 INC 2,2
28 13145 175400 INC 3,3
29 13146 040415 FLT2D: STA 0,FLT2F
30 13147 020410 LDA 0,F2WDS ;CALC REAL FAILURE
31 13150 103120 ADDZL 0,0 ;ADDRESS
32 13151 101120 MOVZL 0,0 ;COULD BE OFF BY 8
33 13152 117000 ADD 0,3
34 13153 020410 LDA 0,FLT2F
35 13154 054407 STA 3,FLT2F
36 LCALL ERRET
37 13155 100470 ERRET=ASCRA*1B11+100010
38 13156 000000 FLT2S: 0
39
40 13157 000000 F2WDS: 0
41 13160 000000 FLT2A: 0
42 13161 000000 FLT2B: 0
43 13162 000000 FLT2C: 0
44 13163 000000 FLT2F: 0
45 13164 000000 FLT2E: 0

```

```

10187 N3MRT
01      ;      .TITL MUDVT
02      ;MULTIPLY DIVIDE TEST COMPATABLE WIITH
03      ;THE DIAGNOSTIC LINKER
04      .MACRO STORE
05      JSR @IXTOR
06      X
07      .MACRO MULCK
08      JSR @IMCK
09      X
10      .MACRO DIVCK
11      JSR @IDCK
12      X
13      .MACRO DIVER.
14      JSR @EDIV
15      X
16      .MACRO MULER
17      JSR @EMUL
18      X
19      NEXTT MUDIO
20      013165 LMEML=.
21      000164      .LOC LPGO
22 00164 013170 MUDIO
23      000165 LPGO=.
24      013165      .LOC LMEML
25 13165 000000 0      ;TEST PASS CTR
26 13166 000000 0      ;TEST ERROR CTR
27 13167 000000 0      ;INTERRUPT TIMEOUT SWITCH
28

```

```

10188 N3MRT
01 13170 013207 MUDIO: MUDI1
02 13171 013220 MDA00
03 13172 000000 0
04 13173 000000 0
05 13174 177777 -1
06 13175 176000 176000
07 13176 014065 MDTX
08 13177 014065 MDTX
09 13200 052515 .TXTE (MUL/DIV TEST(
10      127714
11      144504
12      120126
13      142724
14      152123
15      000000
16
17 13207 102000 MUDI1: ADC 0,0
18 13210 040762 STA 0,MUDI0+2
19 13211 126400 SUB 1,1
20 13212 152400 SUB 2,2
21 13213 073301 DOCP 2,01
22 13214 125005 MOV 1,1,SNR      ;SKP=MUL/DIV EXISTS
23 13215 001400 JMP 0,3          ;NO MUL DIV EXIT
24 13216 040754 STA 0,MUDI0+2
25 13217 001400 JMP 0,3

```



```

10189 N3MRT
01          MDA00:  SETUP  ;0*0=0
02          LCALL  SETUL
03 13220 100410  SETUL-ASCRA*1B11+100010
04          STORE
05 13221 006530  JSR  @IXTOR
06 13222 000000  0          ;AC0
07 13223 000000  0          ;AC1
08 13224 000000  0          ;AC2
09 13225 006525  JSR  @HMUL
10          MULCK
11 13226 006525  JSR  @IMCK
12 13227 000000  0
13 13230 000000  0
14 13231 000000  0
15          LOOP
16          LCALL  LLOOP
17 13232 100430  LLOOP-ASCRA*1B11+100010
18
19          ;A01:
20          SETUP
21          LCALL  SETUL
22 13233 100410  SETUL-ASCRA*1B11+100010
23          STORE
24 13234 006515  JSR  @IXTOR
25 13235 000000  0
26 13236 000000  0
27 13237 177777  -1
28 13240 006512  JSR  @HMUL
29          MULCK
30 13241 006512  JSR  @IMCK
31 13242 000000  0
32 13243 000000  0
33 13244 177777  -1
34          LOOP
35          LCALL  LLOOP
36 13245 100430  LLOOP-ASCRA*1B11+100010
37

```

```

10190 N3MRT
01          ;A02:
02          SETUP
03          LCALL  SETUL
04 13246 100410  SETUL-ASCRA*1B11+100010
05          STORE
06 13247 006502  JSR  @IXTOR
07 13250 000000  0
08 13251 177777  -1
09 13252 000000  0
10 13253 006477  JSR  @HMUL
11          MULCK
12 13254 006477  JSR  @IMCK
13 13255 000000  0
14 13256 000000  0
15 13257 000000  0
16          LOOP
17          LCALL  LLOOP
18 13260 100430  LLOOP-ASCRA*1B11+100010
19
20          ;A03:
21          SETUP
22          LCALL  SETUL
23 13261 100410  SETUL-ASCRA*1B11+100010
24          STORE
25 13262 006467  JSR  @IXTOR
26 13263 177777  -1
27 13264 000000  0
28 13265 000000  0
29 13266 006464  JSR  @HMUL
30          MULCK
31 13267 006464  JSR  @IMCK
32 13270 000000  0
33 13271 177777  -1
34 13272 000000  0
35          LOOP
36          LCALL  LLOOP
37 13273 100430  LLOOP-ASCRA*1B11+100010

```

10191 N3MRT

```
01
02
03
04 13274 100410
05
06 13275 006454
07 13276 000000
08 13277 000001
09 13300 177777
10 13301 006451
11
12 13302 006451
13 13303 000000
14 13304 177777
15 13305 177777
16
17
18 13306 100430
19
20
21
22
23 13307 100410
24
25 13310 006441
26 13311 000000
27 13312 177777
28 13313 000001
29 13314 006436
30
31 13315 006436
32 13316 000000
33 13317 177777
34 13320 000001
35
36
37 13321 100430
38
```

1A04:

```

SETUP
LCALL SETUL
SETUL-ASCRA*1B11+100010
STORE
JSR @IXTOR
0
1
-1
JSR @HMUL
MULCK
JSR @IMCK
0
-1
-1
LOOP
LCALL LLOOP
LLOOP-ASCRA*1B11+100010

1A05:
SETUP
LCALL SETUL
SETUL-ASCRA*1B11+100010
STORE
JSR @IXTOR
0
-1
1
JSR @HMUL
MULCK
JSR @IMCK
0
-1
1
LOOP
LCALL LLOOP
LLOOP-ASCRA*1B11+100010
```

10192 N3MRT

```
01
02
03
04 13322 100410
05
06 13323 006426
07 13324 177777
08 13325 177777
09 13326 000000
10 13327 006423
11
12 13330 006423
13 13331 000000
14 13332 177777
15 13333 000000
16
17
18 13334 100430
19
20
21
22
23 13335 100410
24
25 13336 006413
26 13337 000001
27 13340 000002
28 13341 000002
29 13342 006410
30
31 13343 006410
32 13344 000000
33 13345 000005
34 13346 000002
35
36
37 13347 100430
38 13350 000406
39 13351 013557
40 13352 014002
41 13353 013523
42 13354 013524
43 13355 014014
```

1A06:

```

SETUP
LCALL SETUL
SETUL-ASCRA*1B11+100010
STORE
JSR @IXTOR
-1
-1
0
JSR @HMUL
MULCK
JSR @IMCK
0
-1
0
LOOP
LCALL LLOOP
LLOOP-ASCRA*1B11+100010

1A07:
SETUP
LCALL SETUL
SETUL-ASCRA*1B11+100010
STORE
JSR @IXTOR
1
2
2
JSR @HMUL
MULCK
JSR @IMCK
0
5
2
LOOP
LCALL LLOOP
LLOOP-ASCRA*1B11+100010
JMP MDA08
IXTOR: SXTOR
HMUL: XHMUL
IMCK: MCK
IDCK: DCK
HDIV: XHDIV
```

10193 N3MRT

```
01          MDA08:  SETUP   ;EXPECT A DIVIDE ERROR
02          LCALL  SETUL
03 13356 100410  SETUL-ASCRA*1B11+100010
04          STORE   ;AND NO CHANGE TO ACS.
05 13357 006772  JSR @IXTOR
06 13360 000000  0
07 13361 000000  0
08 13362 000000  0
09 13363 006772  JSR @HDIV
10          DIVCK
11 13364 006770  JSR @IDCK
12 13365 000000  0
13 13366 000000  0
14 13367 000000  0
15          LOOP
16          LCALL  LLOOP
17 13370 100430  LLOOP-ASCRA*1B11+100010
18
19          ;A09:
20          SETUP   ;EXPECT DIVIDE ERROR.
21          LCALL  SETUL
22 13371 100410  SETUL-ASCRA*1B11+100010
23          STORE   ;NO CHANGE TO ACS.
24 13372 006757  JSR @IXTOR
25 13373 177777  -1
26 13374 177777  -1
27 13375 177777  -1
28 13376 006757  JSR @HDIV
29          DIVCK
30 13377 006755  JSR @IDCK
31 13400 177777  -1
32 13401 177777  -1
33 13402 177777  -1
34          LOOP
35          LCALL  LLOOP
36 13403 100430  LLOOP-ASCRA*1B11+100010
37
```

10194 N3MRT

```
01          ;A10:
02          SETUP   ;DIVIDE ERROR SHOULD
03          LCALL  SETUL
04 13404 100410  SETUL-ASCRA*1B11+100010
05          STORE   ;SET THE CARRY
06 13405 006744  JSR @IXTOR
07 13406 177777  -1
08 13407 177777  -1
09 13410 177777  -1
10 13411 101020  MOVZ 0,0
11 13412 006743  JSR @HDIV
12 13413 101003  MOV 0,0,SNC
13          DIVER
14 13414 006505  JSR @EDIV
15          LOOP
16          LCALL  LLOOP
17 13415 100430  LLOOP-ASCRA*1B11+100010
18
19          ;A11:
20          SETUP
21          LCALL  SETUL
22 13416 100410  SETUL-ASCRA*1B11+100010
23          STORE
24 13417 006732  JSR @IXTOR
25 13420 000000  0
26 13421 000000  0
27 13422 000001  1
28 13423 006732  JSR @HDIV
29          DIVCK
30 13424 006730  JSR @IDCK
31 13425 000000  0
32 13426 000000  0
33 13427 000001  1
34          LOOP
35          LCALL  LLOOP
36 13430 100430  LLOOP-ASCRA*1B11+100010
```

```

10195 N3MRT
01
02          ;A12:
03          SETUP      ;NO DIVIDE ERROR SHOULD
04 13431 100410      LCALL SETUL
05          SETUL=ASCRA*1B11+100010
06 13432 006717      STORE      ;OCCURE AND CARRY SHOULD
07 13433 000000      JSR @IXTOR
08 13434 000000      0          ;BE A ZERO.
09 13435 000001      0
10 13436 101040      MOVO 0,0
11 13437 006716      JSR @HDIV
12 13440 101002      MOV 0,0,SZC
13          DIVER
14 13441 006460      JSR @EDIV
15          LOOP
16          LCALL LLOOP
17 13442 100430      LLOOP=ASCRA*1B11+100010
18
19          ;A13:
20          SETUP      ;4/2=2
21          LCALL SETUL
22 13443 100410      SETUL=ASCRA*1B11+100010
23          STORE
24 13444 006705      JSR @IXTOR
25 13445 000000      0
26 13446 000004      4
27 13447 000002      2
28 13450 006705      JSR @HDIV
29          DIVCK
30 13451 006703      JSR @IDCK
31 13452 000000      0
32 13453 000002      2
33 13454 000002      2
34          LOOP
35          LCALL LLOOP
36 13455 100430      LLOOP=ASCRA*1B11+100010
37

```

```

10196 N3MRT
01
02          ;A14:
03          SETUP      ;CHECK REMAINDER
04 13456 100410      LCALL SETUL
05          SETUL=ASCRA*1B11+100010
06 13457 006672      STORE
07 13460 000000      JSR @IXTOR
08 13461 077777      0
09 13462 100000      77777
10 13463 006672      100000
11          JSR @HDIV
12 13464 006670      DIVCK
13 13465 077777      JSR @IDCK
14 13466 000000      77777
15 13467 100000      0
16          100000
17          LOOP
18 13470 100430      LCALL LLOOP
19          LLOOP=ASCRA*1B11+100010
20
21          ;A15:
22          SETUP
23 13471 100410      LCALL SETUL
24          SETUL=ASCRA*1B11+100010
25 13472 006657      STORE
26 13473 000000      JSR @IXTOR
27 13474 177777      0
28 13475 177777      -1
29 13476 006657      -1
30          JSR @HDIV
31 13477 006655      DIVCK
32 13500 000000      JSR @IDCK
33 13501 000001      0
34 13502 177777      1
35          -1
36          LOOP
37 13503 100430      LCALL LLOOP
                    LLOOP=ASCRA*1B11+100010

```

10197 N3MRT

```
01
02      ;A16:
03      SETUP
04 13504 100410  LCALL SETUL
05      SETUL=ASCRA+1B11+100010
06 13505 006644  STORE
07 13506 000001  JSR @IXTOR
08 13507 000000  1
09 13510 000002  0
10 13511 006644  JSR @HOIV
11      DIVCK
12 13512 006642  JSR @IDCK
13 13513 000000  2
14 13514 100000  100000
15 13515 000002  2
16      LOOP
17      LCALL LLOOP
18 13516 100430  LLOOP=ASCRA+1B11+100010
19
20      ;A17:
21 13517 002401  JMP @.+1
22 13520 013575  MTST
23 13521 014033  EDIV: DERR
24 13522 014026  EMUL: MERR
```

10198 N3MRT

```
01
02 13523 101021 MCK: MOVZ 0,0,SKP ;CHECK MUL RESULT
03 13524 101040 DCK: MOV0 0,0 ;CHECK DIV RESULT
04 13525 054444 STA 3,XCKRET
05 13526 035402 LDA 3,2,3
06 13527 156414 SUB# 2,3,SZR
07 13530 000411 JMP CK1 ;AC2 WRONG
08 13531 034440 LDA 3,XCKRET
09 13532 031400 LDA 2,0,3
10 13533 035401 LDA 3,1,3
11 13534 112415 SUB# 0,2,SNR ;CHECK AC0
12 13535 136414 SUB# 1,3,SZR ;CHECK AC1
13 13536 000403 JMP CK1 ;ERROR
14 13537 034432 LDA 3,XCKRET
15 13540 001403 JMP 3,3
16 13541 034430 CK1: LDA 3,XCKRET
17 13542 021400 LDA 0,0,3
18 13543 025401 LDA 1,1,3
19 13544 031402 LDA 2,2,3
20 13545 040550 STA 0,OKAC
21 13546 044550 STA 1,OKMG
22 13547 050550 STA 2,OKMD
23
24 13550 101002 CK2: MOV 0,0,SZC ;IF C(CARRY)=0 ITS MUL ERR
25 13551 000403 JMP .+3 ;OTHERWISE ITS A DIVIDE ER
26      MULER
27 13552 006750 JSR @EMUL
28 13553 101001 MOV 0,0,SKP
29      DIVER
30 13554 006745 JSR @EDIV
31 13555 034414 LDA 3,XCKRET
32 13556 001403 JMP 3,3
33
```

10199 N3MRT

```
01 13557 021400 SXTOR: LDA 0,0,3 ;PICK UP ARGUEMENTS
02 13560 025401 LDA 1,1,3 ;AND STORE IN ORIG
03 13561 031402 LDA 2,2,3 ;NUMBERS TABLE.
04 13562 042404 STA 0,@XOAC
05 13563 046404 STA 1,@XOMQ
06 13564 052404 STA 2,@XOMD
07 13565 001403 JMP 3,3
08 13566 013772 XOAC: _OAC
09 13567 013773 XOMQ: OMQ
10 13570 013774 XOMD: OMD
11 13571 000000 XCKRET: 0
12 13572 014002 IHMUL: XHMUL
13 13573 013777 IHMD: HMD
14 13574 014014 IHDIV: XHDIV
15
```

10200 N3MRT

```
01 MTST: SETUP ;CHECK MULTIPLY WITH
02 LCALL SETUL
03 13575 100410 SETUL=ASCRA*1B11+100010
04 13576 004522 JSR RAN
05 13577 006773 JSR @IHMUL
06 13600 022766 LDA 0,@XOAC
07 13601 026766 LDA 1,@XOMQ
08 13602 032766 LDA 2,@XOMD
09 13603 004537 JSR XMUL ;PROGRAMED MULTIPLY
10 13604 036767 LDA 3,@IHMD
11 13605 156414 SUB# 2,3,SZR
12 13606 000405 JMP .+5
13 13607 030566 LDA 2,HAC
14 13610 034566 LDA 3,HMQ
15 13611 112415 SUB# 0,2,SNR
16 13612 136414 SUB# 1,3,SZR
17 MULER ;MULTIPLY FAILED
18 13613 006707 JSR @EMUL
19 LOOP
20 LCALL LLOOP
21 13614 100430 LLOOP=ASCRA*1B11+100010
22
23 DTST: SETUP ;CHECK DIVIDE WITH
24 LCALL SETUL
25 13615 100410 SETUL=ASCRA*1B11+100010
26 13616 004502 JSR RAN ;RANDOM NUMBERS.
27 13617 006755 JSR @IHDIV ;HARDWARE DIVIDE
28 13620 020552 LDA 0,OAC
29 13621 024552 LDA 1,OMQ
30 13622 030552 LDA 2,OMD
31 13623 004533 JSR XDIV ;PROGRAMED DIVIDE
32 13624 034553 LDA 3,HMD
33 13625 156414 SUB# 2,3,SZR
34 13626 000405 JMP .+5
35 13627 030546 LDA 2,HAC
36 13630 034546 LDA 3,HMQ
37 13631 112415 SUB# 0,2,SNR
38 13632 136414 SUB# 1,3,SZR
39 DIVER ;DIVIDE FAILED
40 13633 006666 JSR @EDIV
41 LOOP
42 LCALL LLOOP
43 13634 100430 LLOOP=ASCRA*1B11+100010
```

```

10201 N3MRT
01 13635 020454 MDTST: LDA 0,M100 ;MULTIPLY DIVIDE TEST
02 13636 040425 STA 0,FOB
03 SETUP
04 LCALL SETUL
05 13637 100410 SETUL=ASCRA*1B11+100010
06 13640 004460 JSR RAN
07 13641 004553 JSR XHDIV ;HARD DIVIDE
08 13642 040450 STA 0,D0
09 13643 044450 STA 1,D1
10 13644 050450 STA 2,D2
11 13645 004535 JSR XHMUL ;HARD MULTIPLY
12 13646 034526 LDA 3,OMD
13 13647 156414 SUB# 2,3,SZR
14 13650 000414 JMP MDT2
15 13651 030521 LDA 2,OAC
16 13652 034521 LDA 3,OMQ
17 13653 112415 SUB# 0,2,SNR
18 13654 136414 SUB# 1,3,SZR
19 13655 000407 JMP MDT2
20 MDT1: LOOP
21 LCALL LLOOP
22 13656 100430 LLOOP=ASCRA*1B11+100010
23 13657 010404 ISZ FOB
24 13660 000757 JMP MDTST+2
25 13661 002401 JMP @,+1
26 13662 014065 MDTEX
27 13663 000000 FOB: 0
28 13664 020506 MDT2: LDA 0,OAC ;EITHER MUL OR DIV
29 13665 024506 LDA 1,OMQ ;FAILED, TYR TO FIND
30 13666 030506 LDA 2,OMD ;WHICH ONE.
31 13667 004467 JSR XDIV
32 13670 034422 LDA 3,D0
33 13671 116414 SUB# 0,3,SZR
34 13672 000407 JMP MDT3
35 13673 034420 LDA 3,D1
36 13674 136414 SUB# 1,3,SZR
37 13675 000404 JMP MDT3
38 13676 034416 LDA 3,D2
39 13677 156415 SUB# 2,3,SNR
40 13700 000403 JMP .+3
41 MDT3: DIVER ;ITS A DIVIDE ERROR
42 13701 006620 JSR @EDIV
43 13702 000754 JMP MDT1
44 13703 040467 STA 0,OAC
45 13704 044467 STA 1,OMQ
46 13705 050467 STA 2,OMD
47 13706 004434 JSR XMUL
48 MULER ;ITS A MULTIPLY ERR
49 13707 006613 JSR @EMUL
50 13710 000746 JMP MDT1
51 13711 177700 M100: -100
52 13712 000000 D0: 0
53 13713 000000 D1: 0
54 13714 000000 D2: 0

```

```

10202 N3MRT
01 13715 000000 OKAC: 0
02 13716 000000 OKMQ: 0
03 13717 000000 OKMD: 0
04
05 13720 054421 RAN: STA 3,RANRET ;GET RANDOM OPERATORS
06 RAN1: LCALL ARANG
07 13721 100270 ARANG=ASCRA*1B11+100010
08 13722 110700 NEGS 0,2 ;FORM MQ+MD
09 13723 105120 MOVZL 0,1
10 13724 127100 ADDL 1,1
11 13725 107300 ADDS 0,1
12 13726 112415 SUB# 0,2,SNR
13 13727 000772 JMP RAN1 ;REJECT IF AC=MD
14 13730 142432 SUBZ# 2,0,SZC
15 13731 115001 MOV 0,3,SKP
16 13732 000403 JMP .+3
17 13733 141000 MOV 2,0 ;MAKE AC LESS THAN
18 13734 171000 MOV 3,2 ;MD IN ALL CASES.
19 13735 040435 RAN2: STA 0,OAC ;STORE IN ORIGINAL
20 13736 044435 STA 1,OMQ ;NUMBER BLOCK..
21 13737 050435 STA -2,OMD
22 13740 002401 JMP @RANRET
23 13741 000000 RANRET: 0
24 13742 054436 XMUL: STA 3,MSAV ;PROGRAMED MULTIPLY
25 13743 034436 LDA 3,M20
26 13744 125203 MOVR 1,1,SNC
27 13745 101201 MOVR 0,0,SKP
28 13746 143220 ADDZR 2,0
29 13747 175404 INC 3,3,SZR
30 13750 000774 JMP XMUL+2
31 13751 125260 MOVCR 1,1
32 13752 040743 XMUL1: STA 0,OKAC ;STORE RESULTS
33 13753 044743 STA 1,OKMQ
34 13754 050743 STA 2,OKMD
35 13755 002423 JMP @MSAV
36

```

## 10203 N3MRT

```

01 13756 054422 XDIV: STA 3,MSAV ;PROGRAMED DIVIDE
02 13757 142432 SUBZ# 2,0,SZC
03 13760 001400 JMP 0,3 ;OV EXIT
04 13761 034420 LDA 3,M20
05 13762 125120 MOVZL 1,1
06 13763 101100 XDIV1: MOVL 0,0
07 13764 142412 SUB# 2,0,SZC
08 13765 142400 SUB 2,0
09 13766 125100 MOVL 1,1
10 13767 175404 INC 3,3,SZR
11 13770 000773 JMP XDIV1
12 13771 000761 JMP XMUL1 ;STORE RESULTS.
13 13772 000000 DAC: 0
14 13773 000000 OHQ: 0
15 13774 000000 OMD: 0
16 13775 000000 HAC: 0
17 13776 000000 HMQ: 0
18 13777 000000 HMD: 0
19 14000 000000 MSAV: 0
20 14001 177760 M20: -20

```

## 10204 N3MRT

```

01 14002 054423 XHMUL: STA 3,XHRET ;HARDWARE MULTIPLY
02 14003 073301 DOCP 2,01
03 14004 004402 JSR .+2
04 14005 014005 .
05 14006 040767 STA 0,HAC
06 14007 020776 LDA 0,.-2
07 14010 116434 SUBZ# 0,3,SZR
08 14011 000415 JMP MERR
09 14012 020763 LDA 0,HAC
10 14013 000406 JMP XHCOM
11
12 14014 054411 XHDIV: STA 3,XHRET ;HARDWARE DIVIDE
13 14015 176440 SUBO 3,3
14 14016 073101 DOCS 2,01
15 14017 175004 MOV 3,3,SZR
16 14020 000413 JMP DERR
17 14021 040754 XHCOM: STA 0,HAC ;STORE HARDWARE RESULTS.
18 14022 044754 STA 1,HMQ
19 14023 050754 STA 2,HMD
20 14024 002401 JMP @XHRET
21 14025 000000 XHRET: 0
22
23 14026 054412 MERR: STA 3,MERET ;MULTIPLY ERROR
24 14027 101001 MOV 0,0,SKP
25 14030 014067 MULHED ;HEADING FOR MUL.
26 14031 020777 LDA 0,.-1
27 14032 000407 JMP GBR
28
29 14033 054405 DERR: STA 3,MERET ;DIVIDE ERROR
30 14034 101001 MOV 0,0,SKP
31 14035 014101 DIVHED
32 14036 020777 LDA 0,.-1
33 14037 000402 JMP GBR
34 14040 000000 MERET: 0

```



## 10205 N3MRT

```

01 14041 040425 GBR: STA 0,XHDR ;PRINT GOOD-BAD RESULTS
02 LCALL ERROI
03 14042 100350 ERROI-ASCRA*1B11+100010
04 14043 000403 JMP .+3
05 14044 101001 MOV 0,0,SKP
06 14045 020421 LDA 0,XHDR
07 LCALL ERRTX
08 14046 100170 ERRTX-ASCRA*1B11+100010
09 14047 020723 LDA 0,OAC
10 14050 024723 LDA 1,OMQ
11 14051 030723 LDA 2,OMD
12 LCALL ERROC
13 14052 100370 ERROC-ASCRA*1B11+100010
14 14053 000401 JMP .+1
15 14054 020721 LDA 0,HAC
16 14055 024721 LDA 1,HMQ
17 14056 030721 LDA 2,HMD
18 LCALL ERROC
19 14057 100370 ERROC-ASCRA*1B11+100010
20 14060 020635 LDA 0,OKAC
21 14061 024635 LDA 1,OKMQ
22 14062 030635 LDA 2,OKMD
23 LCALL ERROC
24 14063 100370 ERROC-ASCRA*1B11+100010
25 14064 000401 JMP .+1
26 MDTX: LCALL RETRN
27 14065 100210 RETRN-ASCRA*1B11+100010
28 14066 000000 XHDR: 0
29 14067 005215 MULHED: .TXTE 1<15><12>
30 14070 052515 MUL<15><12>ACO AC1 AC2 I
31 14101 005215 DIVHED: .TXTE 1<15><12>
32 14102 144504 DIV<15><12>ACO AC1 AC2 I

```

## 10206 N3MRT

```

01 : .TITL RTCTS
02 :REAL TIME CLOCK TEST TO RUN WITH LINKER
03 :IF A REAL TIME CLOCK EXISTS
04 :RUNTIME WILL BE TYPED AT 5 MINUTES
05 :15 MINUTES AND ON EACH HALF HOUR.
06 :ALSO, FOLLOWING ANY ERROR TYPEOUT
07 :OR ANY TYPE IN WITH ACS4=1
08 :(SEE TTY TEST TO CLR TIMSW)
09 NEXTT RT.00
10 014113 LMEML=.
11 000165 .LOC LPGO
12 00165 014116 RT.00
13 000166 LPGO=.
14 014113 .LOC LMEML
15 14113 000000 0 ;TEST PASS CTR
16 14114 000000 0 ;TEST ERROR CTR
17 14115 000000 0 ;INTERRUPT TIMEOUT SWITCH
18 14116 014136 RT.00: RT.01
19 14117 014165 RT.02
20 14120 000000 0
21 14121 000000 0
22 14122 177777 -1
23 14123 176000 176000
24 14124 014210 RT.03
25 14125 014210 RT.03
26 14126 142722 .TXTE (REAL TIME CLOCK)
27 146101
28 152240
29 046711
30 120305
31 146303
32 141717
33 000113

```

## 10207 N3MRT

```

01          ;DETERMINE WHETHER OR NOT A REAL TIME CLOCK EXISTS
02          ;ENABLE OR DISABLE TEST ACCORDINGLY
03 14136 060277 RT.01: INTDS
04 14137 102000          ADC 0,0
05 14140 040760          STA 0,RT.00+2 ;DISABLES TEST
06 14141 060114          NIOS RTC ;TURN CLOCK ON
07 14142 063514          SKPBZ RTC ;SKIP MAYBE NO CLOCK
08 14143 000402          JMP .+2 ;CLOCK EXISTS
09 14144 063714          SKPDZ RTC ;TRY FOR DONE =1
10 14145 000402          JMP .+2 ;CLOCK EXISTS
11          ;NO RTC ON SYSTEM EXIT WITH CLOCK DISABLED
12 14146 001400          JMP 0,3
13 14147 102400          SUB 0,0
14 14150 040750          STA 0,RT.00+2 ;ENABLE CLOCK TEST
15 14151 020413          LDA 0,RT.02-1
16 14152 040745          STA 0,RT.00+1
17 14153 020405          LDA 0,RT.K1
18 14154 024405          LDA 1,RT.K2
19 14155 030405          LDA 2,RT.K3
20 14156 060214          NI0C RTC ;TURN CLOCK OFF
21 14157 002101          JMP @EINTS
22 14160 000014 RT.K1: RTC
23 14161 000007 RT.K2: 7
24 14162 014250 RT.K3: RT.ID
25 14163 000005 RTFIV: 5

```

## 10208 N3MRT

```

01          ;START CLOCK TEST IS NOT DELETED
02 14164 014165          RT.02
03 14165 020424 RT.02: LDA 0,RT.K4
04 14166 040731          STA 0,RT.00+1
05 14167 102420          SUBZ 0,0
06 14170 101500          INCL 0,0
07 14171 024450          LDA 1,RT.K5
08 14172 044450          STA 1,RTSEC ;TO COUNT 1 SECOND
09 14173 024445          LDA 1,RT.K6
10 14174 044447          STA 1,RTMIN ;60 SEC'S =1MIN.
11 14175 024766          LDA 1,RTFIV
12 14176 131120          MOVZL 1,2
13 14177 044445          STA 1,RTCTR ;TO COUNT DOWN 1ST
14 14200 050445          STA 2,RTCTR+1 ;2ND AFTER 10 MORE
15 14201 133000          ADD 1,2
16 14202 126400          SUB 1,1
17 14203 044144          STA 1,RTTIM
18 14204 126000          ADC 1,1
19 14205 044150          STA 1,TIMSW ;SET RT=0 INHIBIT TIME
20 14206 050440          STA 2,RTCTR+2 ;3RD AT HALF HOUR
21 14207 061114          DOAS 0,RTC ;TURN CLOCK ON 1K HZ
22          RT.03: LCALL RETRN
23 14210 100210          RETRN=ASCRA*1B11+100010
24 14211 014212 RT.K4: RT.04
25 14212 102400 RT.04: SUB 0,0
26 14213 040700          STA 0,RT.00-3 ;SO NO 65K TYPEOUT
27 14214 020150          LDA 0,TIMSW
28 14215 101004          MOV 0,0,SZR ;PRINT TIME
29 14216 000772          JMP RT.03 ;NOT YET
30 14217 102000          ADC 0,0
31 14220 040150          STA 0,TIMSW ;RESET INH. SW
32 14221 020462          LDA 0,RTTEX
33          LCALL ERRTX
34 14222 100170          ERRTX=ASCRA*1B11+100010
35 14223 102400          SUB 0,0
36 14224 024144          LDA 1,RTTIM
37 14225 030413          LDA 2,RT.K6
38          LCALL ADIVI
39 14226 100310          ADIVI=ASCRA*1B11+100010
40 14227 111000          MOV 0,2 ;SAVE MINS
41          LCALL PDECI ;PRINT HRS
42 14230 100150          PDECI=ASCRA*1B11+100010
43 14231 145000          MOV 2,1
44          LCALL PDECI ;ELAPSED TIME IN MINUTES
45 14232 100150          PDECI=ASCRA*1B11+100010
46 14233 024151          LDA 1,ERTOT ;# ERR TYPEOUTS
47 14234 125005          MOV 1,1,SNR
48 14235 000753          JMP RT.03
49          LCALL PDECI ;PRINT ERROR TOTAL
50 14236 100150          PDECI=ASCRA*1B11+100010
51 14237 000751          JMP RT.03

```

10209 N3MRT

```
01 14240 000074 RT.K6: 60.
02 14241 001750 RT.K5: 1000.
03 14242 000000 RTSEC: 0
04 14243 000000 RTMIN: 0
05 14244 000000 RTCTR: 0
06 14245 000000 0
07 14246 000000 0
08 14247 000036 30.
09 14250 060114 RT.ID: NIOS RTC
10 14251 014771 DSZ RTSEC ;1 SECOND
11 14252 000422 JMP RTSTR ;NO
12 14253 020766 LDA 0,RT.K5
13 14254 040766 STA 0,RTSEC
14 14255 014766 DSZ RTMIN ;1 MINUTE
15 14256 000416 JMP RTSTR ;NO
16 14257 010144 ISZ RTTIM ;BUMP ELAPSED MINUTES
17 14260 024760 LDA 1,RT.K6
18 14261 044762 STA 1,RTMIN ;RES. MIN. CTR
19 14262 014762 DSZ RTCTR ;TIME TO PRINT
20 14263 000411 JMP RTSTR ;NOT YET
21 14264 126400 SUB 1,1
22 14265 044150 STA 1,TIMSW ;CLR PR. INH. SW
23 14266 020757 LDA 0,RTCTR+1
24 14267 024757 LDA 1,RTCTR+2
25 14270 030757 LDA 2,RTCTR+3
26 14271 040753 STA 0,RTCTR
27 14272 044753 STA 1,RTCTR+1
28 14273 050753 STA 2,RTCTR+2
29 RTSTR:
30 14274 024137 LDA 1,IO.WK
31 14275 125004 MOV 1,1,SZR
32 14276 014137 DSZ IO.WK ;-1 TESTER WAIT COUNT
33 14277 001400 JMP 0,3 ;NOT TIME TO START YET
34 14300 024136 LDA 1,IO.OC
35 14301 066000 DOB 1,0 ;START IO TESTER
36 14302 001400 JMP 0,3 ;RESTRT CLK DISMISS INTR
37 14303 014304 RTTEX: +1
38 14304 005215 .TXTE !<15><12>R/T(HRS,MIN,ERTOT)= 1
```

10210 N3MRT

```
01 ; .TITL TTYES
02 NEXTT TT.00
03 014320 LMEML=.
04 000166 .LOC LPGO
05 00166 014323 TT.00
06 000167 LPGO=.
07 014320 .LOC LMEML
08 14320 000000 0 ;TEST PASS CTR
09 14321 000000 0 ;TEST ERROR CTR
10 14322 000000 0 ;INTERRUPT TIMEOUT SWITCH
11 14323 014340 TT.00: TT.01
12 14324 014363 TT.02
13 14325 000000 0 ;WAIT FOR INTERRUPT SW
14 14326 177700 -64.
15 14327 177716 -50.
16 14330 176000 176000
17 14331 014367 TT.XI
18 14332 014367 TT.XI
19 014333 .TXTE (
20 14333 152324 TTY TEST(
21 120131
22 142724
23 152123
24 000000
25 ;INITIALIZE TTY TEST
26 14340 102400 TT.01: SUB 0,0
27 14341 040566 STA 0,TT.CK ;CLR CHAR COUNT
28 14342 040763 STA 0,TT.00+2 ;CLR WAIT INTR
29 14343 020425 LDA 0,TT.03 ;INTR ADRS
30 14344 054460 STA 3,TT.S3
31 14345 020411 LDA 0,TT.K1
32 14346 024411 LDA 1,TT.K2
33 14347 030411 LDA 2,TT.K3
34 14350 006101 JSR @EINTS ;ENTER KYBORD INT SERV
35 14351 020410 LDA 0,TT.K4
36 14352 030410 LDA 2,TT.K5
37 14353 006101 JSR @EINTS ;ENTER TTO INT SERV
38 14354 034450 LDA 3,TT.S3
39 14355 001400 JMP 0,3
40 14356 000010 TT.K1: TTI
41 14357 000003 TT.K2: 3
42 14360 014371 TT.K3: TT.TI
43 14361 000011 TT.K4: TTO
44 14362 014467 TT.K5: TT.TO
45 ;START TTY PRINTING
46 14363 102620 TT.02: SUBZR 0,0
47 14364 040741 STA 0,TT.00+2 ;SET WAITING INT
48 14365 020536 LDA 0,TT.CR ;OUT CARG RET
49 14366 061111 DOAS 0,TTO
50 TT.XI: LCALL RETRN
51 14367 100210 RETRN-ASCRA*1B11+100010
```

```

10211 N3MRT
01          ;TTY INTR SERVICE
02 14370 014371 TT.03:  .+1
03 14371 054433 TT.TI:  STA 3,TT.S3
04 14372 060610 TT.RD:  DIAC 0,TTI
05 14373 030434          LDA 2,TT177
06 14374 143400          AND 2,0
07 14375 024152          LDA 1,EACTV
08 14376 125202          MOVR 1,1,SZC
09 14377 000421          JMP TT.EX
10 14400 024153          LDA 1,LASTI      ;CK IF LAST TTI INTA SERVICED
11 14401 125113          MOVL# 1,1,SNC    ;SKP IS NOT SERVICED
12 14402 000437          JMP TT.RE
13 14403 147400          AND 2,1        ;MASK OFF PARITY BIT
14 14404 106404          SUB 0,1,SZR    ;SAME CHAR TWICE ?
15 14405 000434          JMP TT.RE      ;NOPE
16 14406 024420          LDA 1,TT.17
17 14407 106415          SUB#0,1,SNR    ;SKP IF NOT CONTROL 0
18 14410 000423          JMP EODT      ;GO DIRECT TO ODT
19 14411 024412          LDA 1,TTI04   ;CONTROL "D"
20 14412 122415          SUB# 1,0,SNR   ;SKIP IS NOT CONTROL D
21 14413 000422          JMP TT.P?     ;KEY INPUT SERVICE
22 14414 024411          LDA 1,TT.22   ;CK FOR CONTROL R
23 14415 122415          SUB# 1,0,SNR   ;SKP IS NOT CONTROL R
24 14416 000417          JMP TT.P?     ;KEY INPUT SERVICE
25 14417 000422          JMP TT.RE
26 14420 103240 TT.EX:  ADDOR 0,0
27 14421 040153          STA 0,LASTI   ;SAVE CHAR FOR LATER
28 14422 002402 TT.DI:  JMP @TT.S3
29 14423 000004 TT.I04:  4
30 14424 000000 TT.S3:  0
31 14425 000022 TT.22:  22
32 14426 000017 TT.17:  17
33 14427 000177 TT177:  177
34 14430 000100 TT100:  100
35 14431 000060 TT.60:  60
36 14432 004206 PODT:  ODT
37 14433 006777 EODT:  JSR @PODT
38 14434 002770          JMP @TT.S3     ;RETURN TO DISMIS INTR

```

```

10212 N3MRT
01
02 14435 040153 TT.P?:  STA 0,LASTI      ;PLACE CHARACTER IN LASTI
03 14436 006402          JSR @IN.P?
04 14437 002765          JMP @TT.S3
05 14440 005255 IN.P?:  INP?
06
07          ;NOW CHECK FOR SWREG COMMANDS
08 14441 034770 TT.RE:  LDA 3,TT.60
09 14442 116405          SUB 0,3,SNR    ;SKIP IF NOT 60
10 14443 000755          JMP TT.EX
11 14444 152620          SUBZR 2,2     ;NOW CK FOR KEYS 1-9,A-F
12 14445 024763          LDA 1,TT100
13 14446 151221 TT.SC:  MOVZR 2,2,SKP
14 14447 126520          SUBZL 1,1
15 14450 175405          INC 3,3,SNR   ;SKIP IF NOT FOUND IT YET
16 14451 000410          JMP TT.SS     ;FOUND IT, SET SWREG BIT
17 14452 133415          AND# 1,2,SNR  ;DONE THIS GROUP YET
18 14453 000773          JMP TT.SC     ;NOT YET
19 14454 106400          SUB 0,1 ;SET UP FOR KEY A-F NOW
20 14455 135000          MOV 1,3
21 14456 151225          MOVZR 2,2,SNR ;DONE ALL ?
22 14457 000741          JMP TT.EX     ;YEP
23 14460 000767          JMP TT.SC+1   ;NOT YET
24 14461 024230 TT.SS:  LDA 1,SWREG
25 14462 133414          AND# 1,2,SZR  ;XOR SWREG BIT
26 14463 146401          SUB 2,1,SKP
27 14464 147000          ADD 2,1
28 14465 044230          STA 1,SWREG   ;PLACE BACK IN SWREG
29 14466 000732          JMP TT.EX

```

## 10213 N3MRT

```

01          ;TTY OUTPUT INTR HANDLER
02 14467 020440 TT.T0: LDA 0,TT.CK
03 14470 060211          NIOC TTO
04 14471 054733          STA 3,TT.S3
05 14472 034152          LDA 3,EACTV
06 14473 175102          MOVL 3,3,SZC      ;SKP IF NOT ETYPE
07 14474 002730          JMP @TT.S3      ;THROW DONE AWAY
08 14475 034630          LDA 3,TT.00+2    ;WAITING TTO INTR?
09 14476 024432          LDA 1,TT.B0      ;=180
10 14477 136404          SUB 1,3,SZR      ;SKP IS TTO TEST OUTPUT
11 14500 002724          JMP @TT.S3      ;THROW DNE AWAY
12 14501 101004          MOV 0,0,SZR      ;SKP FOR LINE FEED
13 14502 000406          JMP TT.04      ;INTO PRINTING CHAR
14 14503 020421          LDA 0,TT.LF
15 14504 061111          DOAS 0,TTO
16 14505 020421          LDA 0,TT.SP      ;GET SPACE
17 14506 040421          STA 0,TT.CK      ;TO NXT OUT
18 14507 000713          JMP TT.DI      ;DISMISS INTR
19          ;OUTPUT SPACE TO Z
20 14510 024415 TT.04: LDA 1,TT.EN
21 14511 122415          SUB# 1,0,SNR      ;SKP IS NOT Z YET
22 14512 000404          JMP TT.05
23 14513 061111          DOAS 0,TTO
24 14514 010413          ISZ TT.CK
25 14515 000705          JMP TT.DI
26          ;ALL CHARACTERS PRINTED STOP INTERRUPTS
27 14516 102400 TT.05: SUB 0,0
28 14517 040606          STA 0,TT.00+2
29 14520 060211          NIOC TTO
30 14521 040406          STA 0,TT.CK
31 14522 000700          JMP TT.DI
32 14523 000215 TT.CR: 215
33 14524 000212 TT.LF: 212
34 14525 000333 TT.EN: 333
35 14526 000240 TT.SP: 240
36 14527 000000 TT.CK: 0
37 14530 100000 TT.B0: 180

```

## 10214 N3MRT

```

01          ;          .TITL IOTST
02          ;IO TESTER PROGRAM COMPATIBLE WITH LINKR
03
04          NEXTT IO.00
05          014531 LMEML=.
06          000167          .LOC LPGO
07 00167 014534          IO.00
08          000170 LPG0=.
09          014531          .LOC LMEML
10 14531 000000          0          ;TEST PASS CTR
11 14532 000000          0          ;TEST ERROR CTR
12 14533 000000          0          ;INTERRUPT TIMEOUT SWITCH
13
14 14534 014552 IO.00: IO.01
15 14535 014606          IO.02
16 14536 000000          0
17 14537 000000          0
18 14540 177777          -1
19 14541 176000          176000
20 14542 014745          IO.XI
21 14543 014745          IO.XI
22 14544 127711          .TXTE II/O TESTER I
23          120317
24          142724
25          152123
26          151305
27          000240
28

```

```

10215 N3MRT
01          ;DETERMINE IF THERE IS AN IO TESTER
02
03 14552 102000 IO.01: ADC      0,0
04 14553 024427 LDA      1,IO.K1
05 14554 066000 DOB      1,0 ;PUT TESTER IN NEW MODE
06 14555 063000 DDC      0,0 ;LD DATA BUFFER
07 14556 065400 DIB      1,0 ;READ IT BACK
08 14557 040757 STA      0,IO.00+2
09 14560 120004 COM      1,0,SZR
10 14561 001400 JMP      0,3 ;NO IO TESTER IN SYSTEM
11 14562 040754 STA      0,IO.00+2
12 14563 030140 LDA      2,MPSWT
13 14564 151005 MOV      2,2,SNR
14 14565 000403 JMP      .+3
15 14566 070400 DIA      2,0 ;GET TESTER ADDRESS WORD
16 14567 150014 COM#2,2,SZR ;SKP IS USE B DCH MAP
17 14570 152400 SUB      2,2 ;A DCH MAP
18 14571 050414 STA      2,IOT.B ;SET FLAG A/B DCH MAP
19 14572 024410 LDA      1,IO.K1 ;GET NEW MODE BIT
20 14573 066200 DOBC     1,0 ;CLEARS TESTER
21 14574 066000 DOB      1,0 ;KEEP TESTER IN NEW MODE
22 14575 063400 SKPBN    0 ;SKP IF NEW MODE TESTER
23 14576 063077 HALT     ;***CN'T RUN WITH OLD TYPE TESTER
24 14577 030404 LDA      2,IO.K1+1 ;AC0 = DEV#, AC1 = MASK
25 14600 126000 ADC      1,1 ;AC2 = INTA SERV ADRS
26 14601 002101 JMP      @EINTS
27
28 14602 001000 IO.K1: 186
29 14603 014773 IO.ID
30 14604 177775 =3
31 14605 000000 IOT.B: 0 ;-1 IF OK TO USE B DCH MAP

```

```

10216 N3MRT
01          ;I/O TESTER ENTER TO PERFORM CURRENT TEST
02
03 14606 030730 IO.02: LDA      2,IO.00+2 ;GET INTA WAIT/LINK,
04 14607 151004 MOV      2,2,SZR ;SKP = NO ACTIVE TEST
05 14610 001000 JMP      0,2 ;FINISH COMPARES
06          LCALL     ASCRA ;GET 1K CORE
07 14611 100010 ASCRA-ASCRA*1811+100010
08 14612 000533 JMP      IO.XI ;NONE AVAILABLE - EXIT
09 14613 020772 LDA      0,IOT.B ;IOTESTER B DCH FLG
10 14614 101005 MOV      0,0,SNR ;SKP IS USE EITHER A/B DCH MAP
11 14615 000402 JMP      .+2 ;USE ONLY A DCH MAP
12          LCALL     RNDMP ;SELECT A/B DCH MAP
13 14616 100530 RNDMP-ASCRA*1811+100010
14 14617 000401 JMP      .+1 ;NO ERROR RETURN
15          LCALL     ADMAP ;ASSIGN TO DCH
16 14620 100550 ADMAP-ASCRA*1811+100010
17 14621 000524 JMP      IO.XI ;NO DCH AVAIL RELEASE
18 14622 020762 LDA      0,IO.K1+2 ;-3
19          LCALL     ESCRA ;GET 1 TO 3K MORE
20 14623 100030 ESCRA-ASCRA*1811+100010
21 14624 000410 JMP      IO.03
22          LCALL     EDMAP ;OF SRA ASSIGNED
23 14625 100570 EDMAP-ASCRA*1811+100010
24 14626 000404 JMP      IO.28 ;TO DATA CHAN
25 14627 101404 INC      0,0,SZR ;IF POSSIBLE
26 14630 000773 JMP      IO.2A
27 14631 000403 JMP      IO.03
28          LCALL     RSCRA ;RELEASE 1K NOT DATA CHAN
29 14632 100050 RSCRA-ASCRA*1811+100010
30 14633 063077 HALT
31          ;SCRATCH RELEASED?

```

10217 N3MRT

```
01 ;IO TESTER HAS 1 TO 4K OF SCRATCH AND DATA CHANNEL
02 ;ASSIGNED. GET RAN #'S.
03
04 IO.03: LCALL ARANG
05 14634 100270 ARANG-ASCRA*1B11+100010
06 14635 030155 LDA 2,SCRHI
07 14636 024154 LDA 1,SCRLO
08 14637 132500 SUBL 1,2
09 14640 153300 ADDS 2,2
10 14641 034516 LDA 3,IO.K2 ;77 MASK 0 -> 63
11 14642 173000 ADD 3,2
12 14643 173400 AND 3,2
13 14644 163400 AND 3,0
14 14645 107000 ADD 0,1
15 14646 044517 STA 1,IODST
16 14647 024156 LDA 1,DCHLO
17 14650 107000 ADD 0,1
18 14651 044515 STA 1,IOCST
19 14652 050506 STA 2,IO.BK ;# 64 WORD BUFFERS
20 14653 050506 STA 2,IO.XX ;CTR
21 14654 030507 LDA 2,IO.RB ;ADRS 64 WRDRAN BUF
22 14655 024507 LDA 1,IOM64
23 IO.3A: LCALL ARANG ;GET ENUFF
24 14656 100270 ARANG-ASCRA*1B11+100010
25 14657 041000 STA 0,0,2 ;RANDOM WORDS
26 14660 151400 INC 2,2 ;64 FOR EACH
27 14661 125404 INC 1,1,SZR ;BUFFER
28 14662 000774 JMP IO.3A
29 14663 050477 STA 2,IO.RX
```

10218 N3MRT

```
01 ;TESTER HAS 1 TO 4K SCRATCH ASSIGNED TO DATA CHANNEL MAP.
02 ;ALSO, 64 RANDOM WORDS FOR DATA CHANNEL RANDOMLY SELECT
03 ;"READ" (FROM TESTER) OR "WRITE" (TO TESTER).
04
05 LCALL ARANG
06 14664 100270 ARANG-ASCRA*1B11+100010
07 14665 105300 MOVS 0,1 ;COMPARE LWR BYTE
08 14666 122033 ADCZ# 1,0,SNC ;AGAINST UPPER BYTE
09 14667 000417 JMP IO.05 ;READ TO MEM
10
11
12
13 ;SELECTION WAS TO WRITE TO TESTER
14 ;FILL THE BUFFER WITH RANDOM WORDS 64 WRD @ BUFFER
15
16 14670 030475 LDA 2,IODST
17 14671 024473 IO.04: LDA 1,IOM64
18 14672 034471 LDA 3,IO.RB
19 14673 021400 LDA 0,0,3
20 14674 041000 STA 0,0,2
21 14675 175400 INC 3,3
22 14676 151400 INC 2,2
23 14677 125404 INC 1,1,SZR ;FILLED 1 @ 64 WORDS
24 14700 000773 JMP IO.04+2 ;NOT YET
25 14701 014460 DSZ IO.XX ;DONE ALL BUFFERS?
26 14702 000767 JMP IO.04 ;NO, MOVE 64 MORE WORDS
27 14703 102400 SUB 0,0 ;AC0 = WRITE
28 14704 024501 LDA 1,IO.WI ;WRITE INTA HANDLER
29 14705 000406 JMP IO.06
30
31 ;RANDOM SELECTION WAS TO READ (FROM TESTER)
32 ;SET UP NECESSARY READ LINKS TO DATA
33
34 14706 030455 IO.05: LDA 2,IO.RB
35 14707 050453 STA 2,IO.RX
36 14710 024517 LDA 1,IO.RI ;READ INTA HANDLER
37 14711 102620 SUBZR 0,0
38 14712 101220 MOVZR 0,0 ;AC0 = READ
39
```

10219 N3MRT

```
01          ;COMMAND TO TESTER IS IN ACO 0=WRITE 40000=RD
02          ;AC1 = ADRS OF APPROPRIATE INTA HANDLER
03
04 14713 030454 IO.06: LDA    2,IO.K3
05 14714 143000      ADD    2,0
06 14715 040136      STA    0,IO.OC      ;COMMAND OUT
07 14716 044620      STA    1,IO.00+2      ;VECTOR LINK TO INTA
08 14717 024447      LDA    1,IOCST      ;CHANNEL STARTS ADRS
09 14720 067000      DOC    1,0      ;LOAD ADDRESS (DCH)
10 14721 065400      DIB    1,0      ;RD ADRES
11 14722 065000      DOA    1,0      ;CLR DB FOR WRITE
12 14723 026437      LDA    1,@IO.RX
13 14724 142404      SUB    2,0.SZR      ;SKP IF WRITE
14 14725 065000      DOA    1,0      ;READ = LOAD DB
15 14726 030432      LDA    2,IO.BK
16 14727 050432      STA    2,IO.XX      ;SET BLK CTR
17
18 14730 100270      IO.6A: LCALL  ARANG
19 14731 024437      ARANG-ASCRA*1B11+100010
20 14732 123405      LDA    1,IO.K4
21 14733 000775      AND    1,0.SNR      ;SELECT 1 TO 8
22 14734 024136      JMP    IO.6A      ;RGENB'S TWXT BRKS
23 14735 123000      LDA    1,IO.OC
24 14736 040136      ADD    1,0
25 14737 100270      STA    0,IO.OC
26 14740 030432      IO.6B: LCALL  ARANG      ;GET RAN#
27 14741 143505      ARANG-ASCRA*1B11+100010
28 14742 000775      LDA    2,IO.KM
29 14743 040137      ANDL   2,0.SNR      ;RANGEN WILL
30 14744 100210      JMP    IO.6B      ;START TESTER
31 14745 000775      STA    0,IO.WK      ;AFTER THIS # CALLS
32 14746 101001      LCALL  RETRN
33 14747 000776      RETRN-ASCRA*1B11+100010
34
```

10220 N3MRT

```
01          ;RELEASE ALL DATA CHANNEL AND SCRATCH
02
03          IO.XI:
04          LCALL  RDMAP
05 14745 100610      RDMAP-ASCRA*1B11+100010
06 14746 101001      MOV    0,0,SKP
07 14747 000776      JMP    IO.XI
08          IO.X1: LCALL  RSCRA
09 14750 100050      RSCRA-ASCRA*1B11+100010
10 14751 101001      MOV    0,0,SKP
11 14752 000776      JMP    IO.X1
12 14753 102400      SUB    0,0
13 14754 040137      STA    0,IO.WK
14 14755 042427      STA    0,@IO.WI-1
15          LCALL  RETRN
16 14756 100210      RETRN-ASCRA*1B11+100010
17
18 14757 000077      IO.K2:  77
19 14760 000000      IO.BK:  0
20 14761 000000      IO.XX:  0
21 14762 000000      IO.RX:  0
22 14763 015235      IO.RB:  IO.RS
23 14764 177700      IOM64: -64.
24 14765 000000      IODST:  0
25 14766 000000      IOCST:  0
26 14767 101077      IO.K3: 101077
27 14770 000700      IO.K4:  700
28 14771 000000      IO.VX:  0
29
30 14772 000377      IO.KM: 377
31 14773 050446      IO.ID:  STA 2,IOSAV
32 14774 032410      LDA 2,@IO.WI-1
33 14775 141113      MOVL# 2,0.SNC      ;SKP=TSSTR WAITING
34 14776 000403      JMP    .+3
35 14777 063700      SKPDZ 0      ;TESTER DONE
36 15000 001000      JMP    0,2 ;YES PROCESS INTA
37          ;DEVICE 0 INTERRUPT? TESTER NOT DONE?
38          ;MUST BE POWER FAIL
39 15001 152460      SUBC 2,2
40 15002 050146      STA 2,UDEVI      ;PLACE 0 IN UNEXPECTED INTR. FLG
41 15003 001400      JMP    0,3      ;RETURN TO LINKR
```



```

10221 N3MRT
01          ;WRITE INTERRUPT HANDLER I/O TESTER
02
03 15004 014536      IO.00+2
04 15005 115006 IO.WI: @.+1
05 15006 000401      JMP .+1
06 15007 071400      DIB 2,0
07 15010 052752      STA 2,@IO.RX      ;READ ACCUM DATA
08 15011 010751      ISZ IO.RX      ;SAVE CURRENT DATA BUFF
09 15012 014747      DSZ IO.XX      ;DONE ALL BLKS
10 15013 000407      JMP IO.GG      ;NOT YET
11 15014 030450      LDA 2,IO.DW
12 15015 052767      STA 2,@IO.WI-1
13 15016 030410      LDA 2,IO.RI-1
14 15017 060200      NIOC 0
15 15020 072000 IO.WX: DOB 2,0
16 15021 001400      JMP 0,3      ;RETURN
17 15022 071400 IO.GG: DIB 2,0
18 15023 071000      DOA 2,0
19 15024 030136      LDA 2,IO.OC
20 15025 000773      JMP IO.WX
21
22          ;READ INTERRUPT HANDLER I/O TESTER
23
24 15026 001000      186
25 15027 115030 IO.RI: @.+1
26 15030 014731      DSZ IO.XX      ;DONE ALL RDS
27 15031 000403      JMP .+3      ;NOT YET
28 15032 030410      LDA 2,IO.DR      ;DONE READING
29 15033 000762      JMP IO.WX-3
30 15034 071400      DIB 2,0      ;CLEAR DATA
31 15035 071000      DOA 2,0
32 15036 010724      ISZ IO.RX
33 15037 032723      LDA 2,@IO.RX      ;GET NEXT WORD
34 15040 000763      JMP IO.RI-4      ;STRT TSTR AGAIN
35
36 15041 000002 IOSAV: 2
37

```

```

10222 N3MRT
01          ;IO TESTER READ (TO MEMORY) IS COMPLETE
02          ;VERIFY THAT RANDOM DATA IS CORRECT IN
03          ;STRINGS OF 64 WORDS.
04
05 15042 015043 IO.DR: .+1
06 15043 034720      LDA 3,IO.RB
07 15044 024714      LDA 1,IO.BK
08 15045 044714      STA 1,IO.XX
09 15046 030717      LDA 2,IODST
10 15047 024715      LDA 1,IO.M64
11 15050 044721      STA 1,IO.VX
12 15051 021400      LDA 0,0,3      ;GET RAN WORD
13 15052 025000 IO.RR: LDA 1,0,2
14 15053 122414      SUB# 1,0,SZR
15 15054 004476      JSR IO.DE      ;DATA ERR READ
16 15055 151400      INC 2,2
17 15056 010713      ISZ IO.VX      ;SKP = DONE 64 WORDS
18 15057 000773      JMP IO.RR
19 15060 175400      INC 3,3      ;STEP TO NEXT RANDOM
20 15061 014700      DSZ IO.XX
21 15062 000765      JMP IO.RR-3      ;DO NEXT 64 WORDS
22 15063 000446      JMP IO.X3      ;EXIT ALL DATA OK
23

```

10223 N3MRT

```
01          ;TESTER WRITE OPERATION HAS BEEN COMPLETED
02          ;VERIFY THAT:
03          ; A. THE TESTER DATA BUFFER WAS OK @ 64 WORDS
04          ; B. THE OUTPUT BUFFERS THEMSELVES DID NOT CHANGE.
05
06 15064 015065 IO.DW:  .+1
07 15065 030700      LDA    2,IODST
08 15066 034672      LDA    3,IO.BK
09 15067 054672      STA    3,IO.XX          ;FIRST VERIFY
10 15070 034673      LDA    3,IO.RB          ;THAT THE CONTENTS
11 15071 024673      LDA    1,IOM64          ;OF THE OUTPUT
12 15072 044677      STA    1,IO.VX          ;BUFFER HAVE NOT
13 15073 021400 IO.WC:  LDA    0,0,3          ;CHANGED.
14 15074 025000      LDA    1,0,2
15 15075 122414      SUB#   1,0,SZR
16 15076 004454      JSR    IO.DE          ;IO DATA ERR
17 15077 151400      INC    2,2
18 15100 175400      INC    3,3
19 15101 010670      ISZ   IO.VX          ;DONE 64 WORDS
20 15102 000771      JMP    IO.WC          ;NOT YET
21 15103 014656      DSZ   IO.XX          ;DONE ALL BUFFERS
22 15104 000764      JMP    IO.WC-3          ;NOT YET
23 15105 030656      LDA    2,IO.RB
24 15106 024656      LDA    1,IOM64          ;CREATE THE XOR
25 15107 044662      STA    1,IO.VX          ;OF THE 64 RAN
26 15110 102400      SUB    0,0          ;WORDS
27 15111 025000 IO.WW:  LDA    1,0,2
28 15112 135000      MOV    1,3
29 15113 117520      ANDZL 0,3
30 15114 123000      ADD    1,0
31 15115 162400      SUB    3,0
32 15116 151400      INC    2,2
33 15117 010652      ISZ   IO.VX
34 15120 000771      JMP    IO.WW          ;NOW VERIFY THAT
35 15121 024637      LDA    1,IO.BK          ;THE RESULT OF EACH
36 15122 044637      STA    1,IO.XX          ;64 WORD XOR WAS OK
37 15123 025000      LDA    1,0,2
38 15124 122414      SUB#   1,0,SZR
39 15125 004425      JSR    IO.DE          ;I/O DATA ERROR
40 15126 151400      INC    2,2
41 15127 014632      DSZ   IO.XX
42 15130 000773      JMP    -5
43 15131 022420 IO.X3:  LDA    0,@IOHIK
44 15132 024140      LDA    1,MPSWT
45 15133 125005      MOV    1,1,SNR
46 15134 002446      JMP    @IO.KL
47 15135 024413      LDA    1,IO.37
48 15136 106032      ADCZ# 0,1,SZC
49 15137 002443      JMP    @IO.KL
50          IO.X4:  LCALL  RDMAP
51 15140 100610      RDMAP-ASCRA+1B11+100010
52 15141 101001      MOV    0,0,SKP
53 15142 000776      JMP    IO.X4
54          IO.X5:  LCALL  RSCRA
55 15143 100050      RSCRA-ASCRA+1B11+100010
56 15144 101001      MOV    0,0,SKP
57 15145 000776      JMP    IO.X5
58 15146 002401      JMP    @.+1
59 15147 014611      IO.02+3
60 15150 000037 IO.37: 37
```

0224 N3MRT

01 15151 001526 IOHIK: HIGHK  
02

10225 N3MRT

```
01 ;THE IO TESTER MADE A DATA ERROR
02 ;EITHER MEMORY OR IN ITS DATA BUFFER
03
04 IO.OE: LCALL ERROI
05 15152 100350 ERROI-ASCRA*1B11+100010
06 15153 000401 JMP .+1
07 15154 020432 LDA 0,IO.T1
08 LCALL ERRTX
09 15155 100170 ERRTX-ASCRA*1B11+100010
10 15156 020136 LDA 0,IO.OC
11 15157 024606 LDA 1,IODST
12 15160 030606 LDA 2,IOCST
13 LCALL ERROC
14 15161 100370 ERROC-ASCRA*1B11+100010
15 15162 000401 JMP .+1
16 15163 020436 LDA 0,IO.T2
17 LCALL ERRTX
18 15164 100170 ERRTX-ASCRA*1B11+100010
19 15165 022416 LDA 0,@IO.KL+1
20 15166 032416 LDA 2,@IO.KL+2
21 15167 105000 MOV 0,1
22 15170 146400 SUB 2,1
23 15171 131120 MOVZL 1,2 ;L1 (*2)
24 15172 153100 ADDL 2,2 ;L3 (OR *8)
25 15173 153100 ADDL 2,2 ;L5 (OR *32)
26 15174 151100 MOVL 2,2 ;*64
27 15175 036410 LDA 3,@IO.KL+3 ;GET CHANNEL STR.
28 15176 173000 ADD 3,2 ;WHERE 64 WORDS STARTED
29 LCALL ERROC
30 15177 100370 ERROC-ASCRA*1B11+100010
31 15200 000401 JMP .+1 ;RELEASE
32 15201 000730 JMP IO.X3 ;THE WORLD
33 15202 014745 IO.KL: IO.XI
34 15203 014760 IO.BK
35 15204 014761 IO.XX
36 15205 014765 IODST
37
38 15206 015207 IO.T1: .+1
39 15207 005215 .TXTE 1<15><12>IO.OC IODST IOCST1
40 15221 015222 IO.T2: .+1
41 15222 005215 .TXTE 1<15><12>IO.BK B#ERR ERROST1
42
43 15235 000202 IO.RS: .BLK 130.
```

10226 N3MRT

```
01 15437 000000 EXISM: 0
02 15440 000007 .BLK 7
03 15447 000000 AVALM: 0
04 15450 000007 .BLK 7
05 15457 000000 DCHM0: 0 ;DCH A MAP
06 15460 000000 DCHM1: 0
07 15461 000000 DCHM2: 0 ;DCH B MAP
08 15462 000000 DCHM3: 0
09 15463 000000 LSYSE: 0
11 15464 031516 DIRT: .TITL N3MRT
12 15472 000000 .TXTE IN3MRT S 041
13 15473 000200 0
14 15474 167773 STRT1
15 15475 000000 167773
16 15476 000000 0
17 15477 000000 0
18 15500 000000 0
19 15501 000000 0
20 15502 047503 .TXT (COPYRIGHT (C) DGC,1973,1974,1975,1976,1977
21 15527 040440 ALL RIGHTS RESERVED!
22 01000 000000 0
23 01001 000170 LPGO
24 01002 000000 0
25 01003 000000 0
26 01004 015463 LSYSE
27 .END
```

## 0227 N3MRT

A1	011256	148/42	149/01						
A6L	011367	152/11	152/15						
ADOTE	011412	152/10	152/16	152/46					
ADIVI	000100	27/19	43/49	47/25	113/19	114/11	133/13	175/20	
		176/12	177/17	208/39					
ADMAP	000112	28/01	216/16						
AGSTR	001326	34/22	34/32						
AI.TX	006212	99/44	100/33						
ALAUT	001327	34/27	34/33						
ALDSP	001330	34/28	34/34						
ALTB	000141	28/32	43/16	43/35	44/40	53/07	58/05	58/06	
		58/09	60/03	62/09	63/09	64/08	64/12	64/22	
		65/18	71/44	101/25	102/07	102/10	102/31	103/08	
		104/07	105/19	106/05	107/10	108/04			
ALZMA	001265	33/06	33/42						
AM777	006563	103/17	106/32						
AMSCR	006476	28/01	102/04						
AM.37	006556	102/26	103/12	104/13	106/20	108/07			
AM.AA	006564	102/23	103/18	106/39					
AM.BB	006565	102/21	103/19	106/37					
AM.GA	006561	102/37	103/04	103/15	105/29				
AM.GI	006522	102/24	102/25	105/11	106/34				
AM.K1	006557	102/42	103/13						
AM.NM	006545	102/18	103/03						
AM.RM	006562	102/25	103/16						
AM.S3	006555	102/06	102/41	102/42	102/44	103/11	104/06	106/04	
		106/27							
AM.TM	006560	102/35	102/39	103/14	105/16	105/31			
AM.XT	006543	102/09	102/15	102/28	102/42	103/10	104/17	104/24	
		105/10	105/15	105/33	106/12	106/30	106/41		
AND3L	011023	141/11	141/19						
AR4	011560	156/26	159/01						
AR4L	011561	159/04	159/11						
ARANG	000077	27/18	43/46	44/24	44/44	47/21	49/04	101/20	
		113/15	114/05	117/04	131/16	133/09	133/39	175/16	
		176/08	177/11	178/15	181/19	184/19	202/07	217/05	
		217/24	218/06	219/18	219/26				
AREND	012221	172/25							
ARITH	000000	2/02	2/06	2/24	9/02	18/01	129/01		
ARJSR	012202	172/01							
ASCR4	000064	27/07	55/01	57/13	57/37	57/39	113/10	113/15	
		113/19	113/23	114/05	114/11	115/23	115/28	115/35	
		116/04	116/08	116/13	117/04	124/11	124/14	126/40	
		126/44	127/07	127/11	127/16	131/06	131/11	131/16	
		132/25	133/09	133/13	133/27	133/39	133/44	134/06	
		134/10	134/15	134/18	134/23	136/03	136/06	136/14	
		136/17	136/22	136/25	136/33	136/36	137/04	137/11	
		137/17	137/23	137/26	138/04	138/07	138/31	138/34	
		139/04	139/07	139/31	139/34	140/04	140/07	140/24	
		140/27	140/32	140/35	140/42	140/45	141/04	141/07	
		141/26	141/29	142/06	142/09	142/22	142/25	142/30	
		142/33	142/45	142/48	143/04	143/07	143/20	143/23	
		143/28	143/31	143/41	143/44	144/04	144/07	144/17	
		144/20	144/25	144/28	144/39	144/42	146/03	146/06	
		146/16	146/19	146/24	146/27	146/37	146/40	147/04	
		147/15	147/18	147/23	147/34	147/37	148/04	148/07	
		148/37	148/40	149/03	149/06	149/15	149/18	149/23	
		149/26	149/35	149/38	150/04	150/07	150/16	150/19	
		150/23	150/26	150/47	150/50	151/04	151/07	151/22	

## 0228 N3MRT

		151/25	152/04	152/07	152/22	152/25	152/30	152/33	
		152/40	152/43	153/04	153/07	153/22	153/25	154/06	
		154/09	154/25	154/28	155/04	155/07	155/25	155/28	
		156/04	156/07	156/22	156/25	159/03	159/06	159/24	
		159/31	159/38	159/41	160/04	160/21	160/24	161/04	
		161/20	161/23	162/04	162/10	162/16	162/22	162/28	
		162/34	162/40	162/46	162/52	162/58	163/04	163/07	
		164/04	164/11	164/17	164/23	164/29	164/35	164/41	
		164/47	164/53	164/59	165/02	166/04	166/07	166/18	
		166/21	166/26	166/39	166/42	167/04	167/07	167/16	
		167/19	167/24	167/27	167/35	167/38	168/04	168/22	
		168/25	168/30	168/33	168/46	168/49	169/04	169/07	
		169/17	169/20	170/04	170/07	170/17	170/20	170/25	
		170/28	170/38	170/41	171/04	171/07	171/17	171/20	
		171/25	171/28	171/38	171/41	172/03	172/10	172/20	
		172/23	172/26	175/11	175/16	175/20	175/24	176/08	
		176/12	177/11	177/17	177/48	178/15	178/20	178/42	
		179/06	179/10	179/15	179/19	179/24	179/28	179/33	
		179/37	179/48	181/19	182/47	183/12	184/19	186/23	
		186/37	189/03	189/17	189/22	189/36	190/04	190/18	
		190/23	190/37	191/04	191/18	191/23	191/37	192/04	
		192/18	192/23	192/37	193/03	193/17	193/22	193/36	
		194/04	194/17	194/22	194/36	195/04	195/17	195/22	
		195/36	196/04	196/18	196/23	196/37	197/04	197/18	
		200/03	200/21	200/25	200/43	201/05	201/22	202/07	
		205/03	205/08	205/13	205/19	205/24	205/27	208/23	
		208/34	208/39	208/42	208/45	208/50	210/51	216/07	
		216/13	216/16	216/20	216/23	216/29	217/05	217/24	
		218/06	219/18	219/26	219/32	220/05	220/09	220/16	
		223/51	223/55	225/05	225/09	225/14	225/18	225/30	
ASSCR	003231	27/07	62/06						
AS.G1	003236	62/12	63/14	63/22	64/16				
AS.S3	003246	62/08	62/17	62/18	62/20	63/08	64/07		
AS.XT	003244	62/11	62/15	62/18	63/11	63/25	63/28	64/10	
		64/23	64/24						
ATEND	012222	135/03	172/27						
ATERR	010553	133/25	134/03						
ATS03	010600	134/04	134/19	135/01					
ATTX1	010612	134/08	135/11						
ATTX2	010625	135/13							
AT.00	010374	130/05	130/11						
AT.01	010414	130/11	130/29						
AT.02	010417	130/12	131/03						
AT.03	010507	131/04	132/21	133/05					
AT.04	010527	133/07	133/23						
AT.05	010475	131/07	132/22	133/45					
AT.2A	010447	131/21	131/30						
AT.2B	010452	131/29	132/03						
AT.2L	010431	131/15	131/31						
AT.37	010606	131/13	133/11	135/07					
AT.4A	010550	133/43	133/46	134/24					
AT.BG	010604	132/13	134/21	135/05					
AT.EC	010557	130/17	130/18	134/08					
AT.EN	010605	132/20	135/06						
AT.ES	010601	131/09	133/05	133/33	134/26	135/02			
AT.GD	010500	132/27	133/28						
AT.L2	010465	132/14	132/19						
AT.LA	010610	132/28	132/32	133/22	133/24	134/13	134/20	135/09	

















## 0243 N3MRT

MM.EC	010265	123/20	123/21	127/08	128/12				
MM.EN	010247	123/37	126/31						
MM.ER	010260	126/21	127/02	127/26	127/32	127/38	127/45	127/51	
		127/57	128/06						
MM.ES	010251	123/34	124/04	124/09	124/35	126/33	126/38	127/19	
MM.K2	010343	127/31	128/14						
MM.K3	010344	127/37	128/15						
MM.K4	010345	127/44	128/16						
MM.K5	010346	127/50	128/17						
MM.K6	010347	127/56	128/18						
MM.K7	010350	128/05	128/19						
MM.L2	010144	124/25	124/30						
MM.LC	010123	123/38	124/19						
MM.PL	010122	123/37	124/21						
MM.S3	010234	125/04	126/17	126/20					
MM.SA	010246	124/23	125/05	125/17	126/04	126/30			
MM.SE	010250	126/32	127/02	127/13	127/23	127/28	127/35	127/41	
		127/48	127/54	128/02	128/09				
		127/05	127/12	128/21					
MM.X1	010252	124/12	124/37	126/37	126/42	127/17			
MM.X2	010257	124/38	126/43						
MODUA	007751	118/13	120/30						
MPF32	001646	40/01	40/14	40/17	40/39	40/44			
MPPRT	004603	79/17	82/05						
MPRTN	004644	82/33	82/38						
MPSWT	000140	28/31	35/07	35/16	35/42	38/34	44/41	45/08	
		56/06	56/49	57/17	59/09	63/12	65/15	67/04	
		67/34	71/08	78/23	92/27	93/06	102/16	104/18	
		106/28	109/15	215/12	223/44				
MPTXT	004664	82/07	82/54						
MPXTX	007142	109/16	110/31						
MPYA	011505	157/02	159/17	161/28					
MPYAL	011656	161/28	166/11						
MPYU	011504	154/15	155/15	157/01					
MP.LP	004617	82/09	82/17	82/37					
MP.S3	004650	82/05	82/40	82/42					
MS1	010641	135/03	135/05	136/01					
MSAV	014000	202/24	202/35	203/01	203/19				
MSEL	003247	62/12	62/21						
MSG32	001602	38/37	40/01	40/08	40/37				
MSKRG	005560	92/01	92/25	92/43	93/12	93/24	94/31		
MSZ32	001535	38/24	39/10	39/26	39/39				
MS.L1	001600	39/34	39/46	40/01	40/34				
MS.LL	001614	40/18	40/31						
MTST	013575	197/22	200/01						
MTTES	000001	2/02	2/13	2/24	10/01	19/01			
MUDI0	013170	187/22	188/01	188/18	188/24				
MUDI1	013207	188/01	188/17						
MUDVT	000000	2/02	2/08	2/24	9/41	18/33	187/01		
MULCK	000062	MC	187/07	189/10	189/29	190/11	190/30	191/11	191/30
			192/11	192/30					
MULER	000070	MC	187/16	198/26	200/17	201/48			
MULHE	014067		204/25	205/29					
MVDSK	000001		2/02	2/11	2/24	10/01	19/01		
MVETA	001647		30/09	41/03	41/14				
MXDSK	000001		2/02	2/18	2/24				
MXTES	000001		2/02	2/21	2/24				
N100	004551		81/06	81/11					

## 0244 N3MRT

N101	004321	78/06	78/29						
N105	004554	78/39	81/14						
N115	004455	78/26	79/45						
N12	004453	77/24	79/43	81/04					
N121	004454	78/17	79/44						
N124	004320	78/05	78/34						
N136	004174	76/04	78/10						
N15	004323	77/30	78/08						
N177	004324	77/07	78/09						
N40	004315	77/33	78/02	79/04					
N57	004452	77/15	78/42	79/42	80/36				
N60	004372	76/10	78/48						
N67	004314	77/12	78/01						
N7	004177	76/07	76/08						
N75	004175	76/05	78/14						
N77	004552	78/53	81/12						
NEXTT	000003	MC	29/44	112/09	123/04	130/02	174/02	187/19	206/09
			210/02	214/04					
NMPTX	007151	109/18	110/31	110/33					
NOMAP	004364	78/25	78/42						
NOTRL	003705	71/13	71/20						
NOVA	000001	2/02	2/06	12/24	40/01	48/16	55/01	92/01	
		94/17	94/20	99/31	100/24	110/28	110/29	110/31	
		111/18	226/20						
NPROG	001751	42/08	42/10	42/32	42/35				
NTODT	005743	97/23	97/31						
NTOPN	004472	80/02	80/07						
NTPA	003321	64/21	64/26						
NVDSK	000001	2/02	2/10	2/24	10/01	19/01			
NWSTK	002412	49/11	50/04	51/13					
NXDSK	000001	2/02	2/17	2/24					
NXTLO	004464	78/12	79/10	80/01					
OAC	013772	199/08	200/28	201/15	201/28	201/44	202/19	203/13	
		205/09							
OAC0	011653	159/07	159/32	161/25	166/08	166/12			
OAC1	011654	159/08	159/25	161/26					
OAC2	011655	159/09	159/18	161/27					
OCT?A	005136	84/04	86/19						
ODT	004206	26/20	31/44	76/16	91/35	211/36			
OKAC	013715	198/20	202/01	202/32	205/20				
OKND	013717	198/22	202/03	202/34	205/22				
OKMQ	013716	198/21	202/02	202/33	205/21				
OMD	013774	199/10	200/30	201/12	201/30	201/46	202/21	203/15	
		205/11							
OMG	013773	199/09	200/29	201/16	201/29	201/45	202/20	203/14	
		205/10							
OPNLO	004502	78/44	80/16						
OPRN	004241	76/14	77/01	77/45	78/52				
OP.EN	000220	26/27	76/25	76/28	76/31	80/01	80/17		
OTOTL	007546	115/18	116/37	120/38					
P17?7	005235	87/28	87/39	87/48	89/11				
P37?7	005010	83/34	84/36						
PAC?1	005006	83/30	83/45	84/10	84/34	85/02	85/18	85/36	
		86/06							
PAC?2	005007	83/31	83/46	84/03	84/06	84/35	85/03	85/27	
PATT	007764	118/16	119/05	120/41					
PC1?1	005042	84/40	85/12						
PC1?2	005152	85/06	85/15	85/28	86/32	89/19	90/09		





## 0249 N3MRT

TT.01	014340	210/11	210/26						
TT.02	014363	210/12	210/46						
TT.03	014370	210/29	211/02						
TT.04	014510	213/13	213/20						
TT.05	014516	213/22	213/27						
TT.17	014426	211/16	211/32						
TT.22	014425	211/22	211/31						
TT.60	014431	211/35	212/08						
TT.80	014530	213/09	213/37						
TT.CK	014527	210/27	213/02	213/17	213/24	213/30	213/36		
TT.CR	014523	210/48	213/32						
TT.DI	014422	211/28	213/18	213/25	213/31				
TT.EN	014525	213/20	213/34						
TT.EX	014420	211/09	211/26	212/10	212/22	212/29			
TT.K1	014356	210/31	210/40						
TT.K2	014357	210/32	210/41						
TT.K3	014360	210/33	210/42						
TT.K4	014361	210/35	210/43						
TT.K5	014362	210/36	210/44						
TT.LF	014524	213/14	213/33						
TT.P2	014435	211/21	211/24	212/02					
TT.RD	014372	211/04							
TT.RE	014441	211/12	211/15	211/25	212/08				
TT.S3	014424	210/30	210/38	211/03	211/28	211/30	211/38	212/04	
		213/04	213/07	213/11					
TT.SC	014446	212/13	212/18	212/23					
TT.SP	014526	213/16	213/35						
TT.S8	014461	212/16	212/24						
TT.TI	014371	95/17	210/42	211/03					
TT.TO	014467	95/18	210/44	213/02					
TT.XI	014367	210/17	210/18	210/50					
TX65K	007257	32/31	110/43						
TXT.0	006254	70/30	101/01						
TXT.1	006263	70/34	70/36	101/02					
TXT.2	006267	71/03	101/03						
TXT.3	006275	71/33	101/04						
TXT.4	006302	71/49	101/05						
TXT.5	006341	71/25	101/08						
TXT.6	006345	71/28	101/09						
TXT.7	006351	75/15	101/10						
TXT.8	006367	75/40	101/11						
TXT.9	006375	71/54	101/12						
TXT.A	006413	71/57	101/13						
TXT.B	006437	71/15	101/14						
TYP?	005157	84/44	85/16	87/02	89/25	90/58			
TYP2R	005240	87/02	87/47	87/51					
UBL32	001666	30/10	41/25						
UBLIM	001713	41/28	41/31	41/34	41/37	41/46			
UDEVI	000146	28/37	32/04	32/11	35/15	96/02	220/40		
UDEVT	007325	32/08	110/48						
UNMAP	004424	78/38	79/20	79/41					
UP32L	001525	38/25	38/41	41/41					
USES8	000062	26/54	29/13	29/38	35/16	35/25	37/03	48/16	
		55/01	55/19						
USRMP	000115	28/05	44/48	56/52	59/12	59/19	59/37	60/18	
		65/39	66/04	66/11	68/25	68/30			
USRSE	000116	28/06	44/50	45/17	68/32	68/37	71/21		
VLDTA	000234	26/42	56/24	97/15	97/37	99/16			

## 0250 N3MRT

VLPCR	000233	26/41	56/23	97/14	99/15				
VL.TX	006145	97/45	100/27						
WAIT	004224	76/30	77/29	78/04	78/55	80/28			
WAITX	004234	76/33	76/38	78/47	81/39				
WHAT	004376	77/35	78/53						
WHERE	004262	77/14	77/18						
WP.TX	006176	99/35	100/31						
X15	004553	81/02	81/13						
X17	004316	77/27	78/03						
X7	011171	144/43	146/01						
XCBLM	002610	52/06	52/27	52/28					
XCKRE	013571	198/04	198/08	198/14	198/16	198/31	199/11		
XCMPB	002611	52/20	52/29						
XDIV	013756	200/31	201/31	203/01					
XDIV1	013763	203/06	203/11						
XHCOM	014021	204/10	204/17						
XHDIV	014014	192/43	199/14	201/07	204/12				
XHDR	014066	205/01	205/06	205/28					
XHMUL	014002	192/40	199/12	201/11	204/01				
XHRET	014025	204/01	204/12	204/20	204/21				
XJ15	005037	85/04	85/09						
XM20	011170	145/18	145/31						
XMS32	001575	39/10	39/41	39/43	40/01	40/08	40/43		
XMUL	013742	200/09	201/47	202/24	202/30				
XMUL1	013752	202/32	203/12						
XMVET	001662	41/03	41/13	41/14					
XOAC	013566	199/04	199/08	200/06					
XOMD	013570	199/06	199/10	200/08					
XOMQ	013567	199/05	199/09	200/07					
XOR1	000040	MC	129/11	142/38	143/13	144/10	144/31		
XOR2	000042	MC	129/14	142/35	146/09	146/30	148/09		
XORA	000036	MC	129/08	142/12	142/15	143/10	143/34	147/08	147/27
			148/12	148/15	148/18	148/21	148/24	148/27	148/30
XORTE	011166		145/19	145/26	145/29				
XOR.0	011134		142/13	142/16	143/11	143/35	145/01	147/09	147/28
			148/13	148/16	148/19	148/22	148/25	148/28	148/31
			148/43						
XOR.1	011142		142/39	143/14	144/11	144/32	145/08		
XOR.2	011152		142/36	145/17	145/27	146/10	146/31	148/10	
XOR.4	011167		145/01	145/06	145/08	145/15	145/17	145/28	145/30
XOR.L	011255		148/43	152/13					
XRMSE	002317		47/14	47/40	47/42	47/47			
XSYTR	002106		44/38	44/56					
XTINI	001750		42/06	42/12	42/24	42/34			
XUBL3	001704		41/25	41/38	41/39				
ZOC?	004746		27/12	28/19	84/01				
ZSU?P	005005		84/11	84/18	84/28	84/33	85/19	85/31	85/33
			85/50	85/52	86/05				
.BDEG	000223		26/31	26/49	33/22	33/26	33/35		
.CB03	011516		157/02	157/10	157/11	157/15	157/24		
.CB20	011517		157/03	157/12	157/16				
.CB99	011507		157/04	157/08					
.CC98	011524		157/18	157/23					
.DIVU	000044	MC	129/17	159/13					
.ERSV	003515		67/14	67/19	67/32	67/43	68/16		
.MPYA	000046	MC	129/20	159/16					
.MPYU	000056	MC	129/32	154/14	155/14				

