DataGeneral

DIAGNOSTIC LISTING

LISTING

096-000348-04

PROGRAM

NOVA 3 MULTI-PROGRAMMING
RELIABILITY TEST - LONG

TAPE

095-000348-04

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 THREE LENGTHS; SHORT, LONG AND PERIPHERAL.

```
 0001 N3MRT MACRO REV 03.00              16:36:31 04/22/77
01
02
03
04
05              ;*************************************************************
06              ;
07              ; NAME: N3MORTL.TX                          PART NUMBER: 097-000348
08              ;
09              ;
10              ; DESCRIPTION: NOVA 3 MULTI-PROGRAMMING RELIABILITY TEST
11              ;               LONG 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              ;
```

```
!0002 N3MRT
01              ;CONDITIONAL ASSEMBLY FLAGS
02              ;FILE FOR NMORTV1 CPU AND PRIMARY PERIPHERAL DEVICES
03        000000 C8RDS=0
04        000000 DCUTS = 0
05        000000 SCMTS=0
06        000000 ARITH=0
07        000000 FPTST=0
08        000000 MUDVT=0
09        000000 PGDSK=0
10        000000 NVDSK=0
11        000000 MVDSK=0
12        000000 PZDSK=0
13        000000 MTTES=0
14        000000 CATES=0
15        000000 LPTTS=0
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        000001 IOTST=1
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 DCU IS TO BE EXERCISED THE
03                ;       DCU DEVICE CODE MUST BE INSERTED INTO
04                ;       THE LOCATION FOLLOWING:
05        015360          LOC.=DCUDV
06                ;       IF THE MOVING HEAD DISKS ARE TO BE
07                ;       EXERCISED THEY MUST HAVE A PACK INSTALLED
08                ;       AND BE IN THE READY STATE
09                ;       IF MAGNETIC TAPES ARE TO BE EXERCISED
10                ;       THEY MUST BE ON LINE WRITE ENABLED
11                ;       IF CASSETTES ARE TO BE EXERCISED
12                ;       THEY MUST BE ON LINE WRITE ENABLED
13                ;       IF THE LINE PRINTER IS TO BE EXERCISED
14                ;       IT MUST BE ON LINE AND IN THE READY STATE
15                ;
16                ;3.     KEY ENTERED OPTIONS
17                ;               ENTRIES TYPED ON TTY SET BITS IN SWREG
18                ;               FOR USE BY THE PROGRAM.
19                ;
20                ;       KEY 1 =1 DON'T RELEASE AND ALLOW REASSIGNMENT
21                ;                OF MEMORY AFTER ERROR
22                ;       KEY 2 =1 DELETE TYPEOUTS
23                ;       KEY " =1 DELETE MEM ALLOCATION TABLE
24                ;                FROM TYPEOUTS
25                ;       KEY # =1 CAUSES THE DELETION OF THE RANDOM
26                ;                WAIT STATES IN THE TTY AND LPT
27                ;                TESTS.
28                ;       KEY 4 =1 WILL CAUSE THE ELAPSED RUN
29                ;                TIME AND ACCUMULATED ERRORS
30                ;                TO BE TYPED ON THE TTY.
31                ;                (NOTE: A RTC MUST EXIST)
32                ;       KEY 5 =1 DIRECT ALL ERROR AND RUNTIME TYPEOUTS
33                ;                ALSO TO THE LINE PRINTER.
34                ;       KEY 6 =1 THE ERROR ROUTINE WILL PAUSE AFTER
35                ;                EACH PHASE OF AN ERROR TYPEOUT.
36                ;                TYPE A CR KEY ON DEVICE TTI TO PROCEED.
37                ;
38                ;       EACH KEY ENTRY COMPLEMENTS THE PREVIOUS STATE OF
39                ;       SWREG BIT EXCEPT CONTROL CHARACTERS
40                ;       FOLLOWING:
41                ;
42                ;       KEY (C)O        ENTER THE ODT EDITOR
43                ;                       (SEE DESCRIPTION AT PARAGRAPH 7.0)
44                ;       KEY (C)S        PRINT THE RUN STATISTICS OF EACH
45                ;                       TEST.
46                ;       KEY (C)D        DEFAULT MODE RESTART. SWREG
47                ;                       SET TO 0.
48                ;       KEY (C)R        RESTART WITHOUT RESETTING SWREG BITS.
49                ;       KEY M           TYPE THE CURRENT CONTENTS OF SWREG.
50                ;
51                ;       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.
```

```
!0007 N3MRT
01              ;5.     ERROR DESCRIPTION
02              ;       MOST ERRORS DETECTED BY EITHER
03              ;       THE INDIVIDUAL TEST PROGRAMS OR
04              ;       BY THE DIAGNOSTIC LINKER WILL
05              ;       RESULT IN AN EXTENSIVE ERROR
06              ;       TYPEOUT.  SOME SMALL NUMBER OF
07              ;       HIGHLY IMPROBABLE ERRORS MAY RESULT
08              ;       IN A PROGRAM HALT IF THEY ARE
09              ;       OF A NATURE THAT THE LINKER CANNOT
10              ;       RECOVER FROM AND LOGICALLY PROCEED,
11              ;       (I.E. INTERRUPT STACK OVERFLOWS)
12              ;
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.)
```

```
!0008 N3MRT
01              ;5.2    ERROR ANALYSIS
02              ;       DUE TO THE INTERACTIVE NATURE OF
03              ;       THE TESTS INVOLVED, A SERIES OF
04              ;       ERROR TYPEOUTS WILL PROBABLY BE
05              ;       REQUIRED FOR ANALYSIS BEFORE A
06              ;       PROBLEM WILL BE ISOLATED.
07              ;       A RESTART AT 202 AND DELETION OF ALL
08              ;       BUT THE TEST THAT ORIGINALLY
09              ;       FAILED MAY HELP TO ISOLATE
10              ;       INTERACTIVE PROBLEMS AS FOLLOWS:
11              ;
12              ;5.2.1          IF THE TEST RUNS BY ITSELF THE PROBLEM
13              ;       IS INTERACTIVE-RE-ENABLE ONE OTHER TEST AT
14              ;       A TIME TO DETERMINE WHICH ONE IS THE PROBLEM.
15              ;       IF THE TEST DOES NOT RUN BY ITSELF
16              ;       RESORT TO SIMILAR BUT LOWER LEVEL TESTS
17              ;       FOR ISOLATION
18              ;5.3    PERTINENT MEMORY LOC'S TYPED
19              ;
20              ;5.3.1  CHECKERBOARD RAN
21              ;               THE AC'S AT ERROR WILL INDICATE:
22              ;               GOOD DATA= BAD DATA=LOGICAL ADDRESS
23              ;
24              ;               IN ADDITION THE FOLLOWING LOCATIONS ARE TYPED:
25              ;               CB.TK   TEST COUNTER
26              ;               0       GENERATE CHECKERBOARD
27              ;               1       DISTURB PASS
28              ;               2       CHECK PATTERN
29              ;               3       CHECKSUM THE # OF -1'S IN PATTERN
30              ;               CB.LC   STARTING LOGICAL ADDRESS OF "BEGIN"
31              ;                       RELOCATED TO SCRATCH
32              ;               CB.SE   AC3 AT ERROR CALL
33              ;
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              ;5.3.7  6063/64  DISK
02              ;
03              ;       AC0     GOOD DATA
04              ;       AC1     BAD DATA
05              ;       AC2     ADDRESS OF GOOD DATA(BAD IS AT AC2+4)
06              ;       PD.SA   MAY=ANY OF THE FOLLOWING
07              ;               A) THE ADDRESS OF A RANDOM DATA CONTROL
08              ;               WORD(PDDW1 TO PDDW4). THE ERROR WAS IN
09              ;               ONE OF THE FIRST 4 WORDS IN THE BUFFER.
10              ;               B) = AC2, ERROR WAS A DISK STATUS ERR
11              ;               C) A NEGATIVE # ,DATA ERROR IS AT AC2+4
12              ;       PDDST   DATA START IN CORE
13              ;       PD.CA   LOGICAL ADDR OF CHANNEL  IN 1K'S OCTAL
14              ;       PDSTA   LAST DISK STATUS
15              ;       PDADR   DRIVE+TRACK+SECTOR(FIRST WORD OF QUEUE)
16              ;       PD.CO   0=READ,1=WRITE,2=DATA VERIFY
17              ;
18              ;5.3.8  NOVA DISK
19              ;       AC0     GOOD DATA
20              ;       AC1     BAD DATA
21              ;       AC2     ADRS. OF GOOD DATA (BAD IS AT AC2+4)
22              ;       ND.SA   MAY= ANY OF THE FOLLOWING
23              ;               A. THE ADRS. OF A RANDOM DATA CONTROL WORD
24              ;               (NDDW1 TO NDDW4) THE ERROR WAS IN ONE OF
25              ;               THE FIRST 4 WORDS IN THE BUFFER
26              ;               B. =AC2 ERROR WAS DISK STATUS
27              ;               C. A -# DATA ERR IS AT AC2+4
28              ;       NDDST   DATA START IN CORE
29              ;       NDCST   DATA START FOR DCH MAP (IF MMPU EXISTS)
30              ;       NDSTA   LAST DISK STATUS
31              ;       NDADR   START SECTOR # OF THESE EXERCISED
32              ;       ND.CO   =1 OPERATION WAS A WRITE
33              ;       ND.CO   =0 OPERATION WAS A READ
34              ;
```

```
10011 N3MRT
01              ;5.3.9  MOVING HEAD DISK
02              ;       AC0     GOOD DATA (SEE MH.SA)
03              ;       AC1     BAD DATA
04              ;       AC2     ADRS. OF GOOD DATA
05              ;               BAD IS AT AC2+4 IF MH.SA IS A -#
06              ;       IN ADDITION THE FOLLOWING LOC'S ARE TYPED
07              ;       MH.SA   =ADRESS OF A RANDOM DATA CONTROL WORD
08              ;               (MHDW1 TO MHDW4) THE  ERR WAS IN ONE OF
09              ;                THE FIRST 4 WORDS IN THE BUFFER
10              ;               =-# ERR IS AT AC2+4
11              ;               =AC2  ERROR WAS DISK STATUS
12              ;                     IN WHICH CASE:
13              ;               AC2=DISK STATUS
14              ;               AC1=DIC DISK
15              ;               AC0=DIB DISK
16              ;       MHDST   DATA START IN CORE
17              ;       MHCST   DATA START FOR DCH MAP
18              ;       MHSTA   LAST DISK STATUS
19              ;       MHDOA   LAST DOA TO DISK
20              ;       MHDOC   LAST DOC TO DISK
21              ;
22              ;5.3.10 MAGNETIC TAPE TEST
23              ;       AC0     GOOD DATA
24              ;       AC1     BAD DATA
25              ;       AC2     ADRS OF BAD DATA (GOOD IS AT AC2-4)
26              ;IN ADDITION THE FOLLOWING LOC'S ARE TYPED
27              ;       READ    INDICATES # BLOCKS IN BUFFER IF MODE=2
28              ;       W/DOB   STARTING ADRS. WHEN BLOCKS WERE WRITTEN
29              ;       LAST/DOB  STARTING ADRS. CURRENT OPERATION
30              ;       MODE    0=REWIND 1=WRITE 2=BACKSPACE OR READ
31              ;               DRIVE # WILL APPEAR IN BITS 4,5&6
32              ;       STATUS  LAST TAPE STATUS
33              ;       MT.EK   ERROR COUNTER STARTS AT 3 AND COUNTS
34              ;               DOWN FOR EACH REREAD
35              ;NOTE:  IF STATUS INDICATES TAPE ERR (BIT 0=1)
36              ;       THE CONTENTS OF AC0,1,AND 2 SHOULD BE IGNORED.
37              ;
```

```
10012 N3MRT
01              ;5.3.11 CASSETTE TAPE TEST
02              ;       AC0     GOOD DATA
03              ;       AC1     BAD DATA
04              ;       AC2     ADRS OF BAD DATA (GOOD IS AT AC2-4)
05              ;IN ADDITION THE FOLLOWING LOC'S ARE TYPED
06              ;       MODE    0=REWIND 1=WRITE 2=BACKSPACE OR READ
07              ;               DRIVE# WILL APPEAR IN BITS 4,5 & 6
08              ;       STATUS  LAST TAPE STATUS
09              ;       CA.EK   ERROR COUNTER STARTS AT 6 AND COUNTS
10              ;               DOWN FOR EACH REREAD
11              ;NOTE:  IF STATUS INDICATES TAPE ERR (BIT 0=1)
12              ;       THE CONTENTS OF AC0,1,AND 2 SHOULD BE IGNORED.
13              ;
14              ;
15              ;5.3.12 LINE PRINTER
16              ;       NO ERROR TYPEOUTS.
17              ;       THE PRINTER PATTERN IS THE SAME AS THE
18              ;       TELETYPE TEST WITH 10 TO 60 LINES
19              ;       PER PAGE.  PRINTED WITH RANDOM STALLS
20              ;       BETWEEN EVERY 1 TO 10 LINES
21              ;       PRINTER OUTPUT MUST BE EXAMINED VISUALLY.
22              ;
23              ;5.3.14 DCU-50 TEST
24              ;
25              ;       THE AC'S WILL BE TYPED AS THEY WERE AT THE
26              ;       TIME OF ERROR DETECTION.
27              ;
28              ;       IN ADDITION THE FOLLOWING DATA IS TYPED:
29              ;       RANDOM DATA AC0,1,2
30              ;       DCLOR   LOGICAL START OF LOOP IN DCU
31              ;       DCLPK   LOOP COUNT
32              ;       DCLER   LOGICAL ERROR ADDRESS
33              ;       DC.LA   LOGICAL START OF TEST
34              ;       DC.LC   LISTING START OF LOOP
35              ;       ERROR   LISTING ADDR OF ERROR
```

```
10013 N3MRT
01              ;5.3.15 6060/6061 DISK TEST
02              ;
03              ;       ERROR TYPEOUTS ARE SIMULAR TO THE
04              ;       MOVING HEAD DISK TEST.
05              ;
06              ;5.4    SPECIAL CASE ERROR TYPEOUTS
07              ;
08              ;5.4.1  POWER FAIL INTERRUPT
09              ;       UPON DETECTION OF A POWER FAIL INTERRUPT
10              ;       THE LOGICAL ADRS. OF THE P.C. AT INTERRUPT
11              ;       WILL BE SAVED.
12              ;       IF AUTO-RESTART  IS ENABLED OR THE POWER
13              ;       FAIL WAS ONLY MOMENTARY, THE TEST WILL RE-
14              ;       START AS IN A START AT 204 AFTER TYPING
15              ;POWER FAIL @XXXXXX (WHERE XXXXXX IS THE PC AT INTR.)
16              ;
17              ;5.4.2  ILLEGAL SUPERVISOR CALL
18              ;
19              ;       UPON DETECTION OF A SUPERVISOR CALL
20              ;       WHICH DIDN'T MATCH THE LIST OF SUBROUTINES
21              ;       CALLS THE FOLLOWING MESSAGE WIIL BE TYPED:
22              ;
23              ;       ILLEGAL SUPER CALL AT XXXXXX
24              ;
25              ;       PROG#   NNN
26              ;
27              ;       AC'S    QQQQQQ  YYYYYY  ZZZZZZ
28              ;
29              ;       TTTTTT  WWWWWW  SSSSSS
30              ;
31              ;               WHERE XXXXXX IS THE LOGICAL ADDRESS OF THE
32              ;               SUPER CALL, TTTTTT IS AC3 CONTENTS
33              ;               AND WWWWWW IS THE PHYSICAL PAGE #,SSSSSS
34              ;               IS THE INSTRUCTION CAUSING THE SUPER-
35              ;               CALL.
36              ;
37              ;       NOTE:   IF THE ADDRESS TYPED IN THE ILLEGAL SUPERCALL
38              ;               WAS 000000 THEN THE PROGRAM WAS EXECUTING
39              ;               LOCATION 0.
```

```
10014 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
```

```
!0015 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              ;
```

```
!0016 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 UITILIZED
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              ;
```

```
10017 N3MRT
01                ;6.2    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                ;
```

```
10018 N3MRT
01                ;6.4    INDIVIDUAL TEST DESCRIPTIONS
02                ;
03                ;6.4.1  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
30                ;6.4.2  DCU - 50 TEST
31                ;
32                ;       THE MULTI-PROGRAMMING RELIABILITY DCU TEST RUNS
33                ;       AN ARITHMETIC TEST VIA THE DCH .
34                ;       THE DCU INTERRUPTS THE CPU
35                ;       WHEN EITHER IT HAS COMPLETED THE TEST OR
36                ;        FINDS AN ERROR. THIS TEST REQUIRES THE OPERATOR TO
37                ;       INPUT INTO THE  FOLLOWING LOCATION:
38          015360        LOC.=DCUDV
39                ;       THE DEVICE CODE FOR THE DCU. IF NO DEVICE CODE IS
40                ;       ENTERED THE TEST IS AUTOMATICALLY DELETED.
41
```

```
10019 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.
```

```
10020 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 MMPU) 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.
```

```
!0021 N3MRT
01              :6.4.7  6063/64  DISK
02              :
03              :       DURING INITIALIZATION THE TEST CHECKS FOR THE
04              :       EXISTANCE OF A DISK CONTROLLER AND THEN CHECKS
05              :       FOR THE EXISTANCE OF ANY/ALL DRIVES. A WRITE
06              :       BUFFERS COMMAND IS USED TO SIZE FOR AVAILABLE
07              :       DRIVES.
08              :
09              :       THE TESTING OF EACH AVAILABLE DRIVE IS CONTROLLED
10              :       BY SELECTION OF ONE OF THREE OPERATION TABLES
11              :       PER DRIVE. EACH CONTROL TABLE IS 7 WORDS IN
12              :       LENGTH. THE FIRST WORD CONTAINS THE TRACK NUMBER
13              :       (RANDOMLY SELECTED), THE SECOND WORD CONTAINS THE
14              :       STARTING SECTOR AND NUMBER OF SECTORS USED. THE
15              :       STARTING SECTOR IS RANDOMLY SELECTED AND THE NUMBER
16              :       OF SECTORS IS DETERMINED BY THE AMOUNT OF SCRATCH
17              :        AREA ASSIGNED TO THE DISK TEST WHEN THE DISK
18              :       IS WRITTEN.
19              :       THE THIRD WORD CONTAINS THE FIRST WORD OF THE
20              :       COMMAND QUEUE,I.E.,DRIVE+TRACK+SECTOR.
21              :       THE FOURTH THRU SEVENTH WORDS ARE THE RANDOM DATA
22              :       WORDS USED TO CREATE THE TEST PATTERN.(THEY REPEAT
23              :       EVERY FOUR WORDS)
24              :
25              :       UPON ENTERING FOR INITIAL EXECUTION, THE TEST ATTEMPS TO
26              :       ACQUIRE 1-4K OF SCRATCH AREA. THE TEST THEN RANDOMLY
27              :       SELECTS A DATA STARTING ADDRESS AFTER THE FIRST
28              :       256 WORDS IN SCRATCH. THE FIRST 256 WORDS ARE RESERVED
29              :       FOR THE COMMAND QUEUE.
30              :       THE TEST THEN SELECTS ONE OF THE AVAILABLE
31              :       DRIVES AND ONE OF THE THREE OP TABLES FOR THAT DRIVE.
32              :       IF THE FIRST WORD OF THE TABLE IS NON-ZERO(INDICATING
33              :       THE THE TRACK # IN WORD #1,STARTING AT THE SECTOR IN
34              :       BITS 11-15 OF THE SECOND WORD,FOR THE # OF SECTORS
35              :       SPECIFIED BY THE BITS 3-7 IN THE SECOND WORD, RANDOM DATA
36              :       HAS BEEN WRITTEN THAT IS EQUAL TO THE CONTENTS OF WORDS
37              :       4 THRU 7 F THE OP TABLE) THEN THE NEXT OPERATION OF
38              :       READ OR DATA VERIFY IS RANDOMLY SELECTED.
39              :       IF THE FIRST WORD OF THE TABLE IS ZERO, A TRACK IS
40              :       SELECTED WHICH IS NOT CURRENTLY IN AN OP TABLE,
41              :       AND A STARTING SECTOR # IS RANDOMLY SELECTED SUCH THAT
42              :       THE # OF SECTORS WRITTEN WILL NOT MAKE THE SECTOR #
43              :       FIELD OVERFLOW INTO THE TRACK FIELD.(I.E.,THE STARTING
44              :       SECTOR FALLS BETWEEN 0 AND 32-# OF SECTORS TO BE WRITTEN)
45              :       THE DATA PATTERN IS GENERATED IN SCRATCH AREA AND
46              :       A WRITE OPERATION IS SELECTED.
47              :
48              :       AFTER SELECTION OF THE OPERATION TO BE PERFORMED,
49              :       A COMMAND QUEUE IS GENERATED IN THE FIRST 256 WORDS
50              :       OF SCRATCH(FIVE WORDS PER SECTOR). THE FIRST WORD
51              :       CONTAINS THE DRIVE,TRACK AND SECTOR TO BE EXERCISED.
52              :       THE SECOND WORD CONTAINS THE COMMAND AND EXTENDED
53              :       ADDRESS BITS OF THE DATA ADDRESS. WORD # 3 CONTAINS
54              :       THE THE LOWER 16 BITS OF THE LOGICAL ADDRESS OF THE
55              :       DATA . WORD # 4 IS USED BY THE DISK TO STORE THE
56              :       STATUS WORD. WORD # 5 IS NOT CURRENTLY USED.
```

```
!0022 N3MRT
01              :6.4.7  6063/64  DISK (CONTINUED)
02              :
03              :       THE LAST QUEUE BLOCK CONTAINS A HALT BIT IN WORD
04              :       # 2 WHICH TERMINATES THE OPERATION AFTER
05              :       COMPLETION OF THE SECTOR AND CAUSES THE DISK
06              :       TO INTERRUPT THE CPU.
07              :       THE DISK STATUS IS CHECKED AT THE TIME OF THE INTERRUPT
08              :       AND IF OK THE SCRATCH AREA IS VERIFIED. AS THE
09              :       DATA COMPARES ARE PERFORMED A NEGITIVE COUNT IS
10              :       STORED IN THE CHECKED LOCATION TO CLEAR THE BUFFER.
11              :6.4.8  NOVA DISK TEST
12              :
13              :WHEN ENTERED FOR ITS INITIALIZATION PASS THE
14              :NOVA DISK TEST SIZES THE DISK SYSTEM. IT DOES THIS BY
15              :INITIATING A READ AT THE HIGHEST ADDRESS OF
16              :EACH AVAILABLE DISK UNTIL IT NO LONGER RECEIVES A
17              :NONEXISTENT DISK BIT IN THE STATUS.
18              :THIS "HIGH DISK ADRS." IS TYPED THE FIRST
19              :TIME THE NOVA DISK TEST IS ENTERED.
20              :DURING RUNNING, THE DISK TEST KEEPS 3 RANDOM
21              :CONTROL TABLES THAT CONTAIN THE FOLLOWING INFO:
22              :       A. THE START SECTOR# OF 16 RANDOM SELECTED
23              :          SECTORS. (THIS # IS RIGHT JUSTIFIED 4 BITS)
24              :       B. A START SECTOR RANDOMLY SELECTED WITHIN 16
25              :       C. THE NUMBER OF SECTORS TO BE WRITTEN/READ
26              :       D. 4 RANDOM DATA WORDS THAT KEY THE PATTERN
27              :          (THESE 4 WORDS REPEAT EVERY 4 WORDS)
28              :THE TEST OPERATES OFF THESE RANDOM CONTROL TABLES
29              :IN A MANNER SIMILAR TO THAT DESCRIBED FOR THE INDIVIDUAL
30              :DISKS IN THE MOVING HEAD DISK TEST
31
```

```
!0023 N3MRT
01              ;6.4.9  MOVING HEAD DISK TEST
02              ;WHEN ENTERED FOR INITIALIZATION, THE MOVING HEAD
03              ;DISK TEST SIZES EACH DISK THAT IS "READY".
04              ;A. THE HIGHEST AVAILABLE SECTOR IS FOUND BY A SERIES
05              ;   OF SEEKS AND RECALIBRATE COMMANDS.
06              ;B. THE NUMBER OF AVAILABLE SECTORS ON A SURFACE ARE
07              ;   FOUND BY INITIATING A SERIES OF 2 SECTORS READS.
08              ;C. THE NUMBER OF SURFACES ARE DETERMINED BY INITIATING
09              ;   ANOTHER SERIES OF 2 SECTOR READS AT THE LAST
10              ;   SECTOR ON EACH SURFACE UNTIL END OF CYLINDER.
11              ;THIS INFORMATION IS TYPED (BY DISK) THE
12              ;FIRST TIME THE TEST IS ENTERED DURING "RUN"
13              ;
14              ;  NOTE:   DUE TO A MORE COMPLEX IMPLEMENTATION
15              ;       OF MULTIPROCESSOR APPLICATION, THIS TEST CAN
16              ;       NO LONGER DETERMINE WHICH ADAPTER PORT THE
17              ;       MOVING HEAD DISK CONTROL IS ON.
18              ;THE TESTING OF EACH AVAILABLE DRIVE IS CONTROLLED BY
19              ;AN INDIVIDUAL CONTROL TABLE FOR EACH DRIVE.
20              ;AFTER RANDOMLY SELECTING AN AVAILABLE DRIVE, THE
21              ;SEQUENCE PROCEEDS AS FOLLOWS:
22              ;THE OPERATION OF THE EACH MOVING HEAD DISK IS CONTROLLED BY
23              ;THE CONTENTS OF 3 TABLES.  EACH CONTROL TABLE IS 8 WORDS
24              ;IN LENGTH.  THE FIRST WORD CONTAINS THE CYLINDER NUMBER,
25              ;(RANDOMLY SELECTED), THE SECOND WORD CONTAINS THE STARTING
26              ;SECTOR AND NUMBER OF SECTORS UTILIZED.  THE START SECTOR IS
27              ;RANDOMLY SELECTED AND THE NUMBER OF SECTORS IS CONTROLLED BY THE
28              ;AMOUNT OF SCRATCH AREA AVAILABLE TO THE DISK TEST WHEN THE
29              ;DISK IS WRITTEN.
30              ;THE 3RD WORD IS AN ERROR COUNTER. FOR EACH ERROR DETECTED, THE
31              ;DISK IS RECALIBRATED AND THE OPERATION IS REPEATED.
32              ;THIS "RECAL/REPEAT" IS EXECUTED UP TO 4 TRYS.
33              ;THE FOURTH THROUGH 7TH WORDS ARE THE RANDOM DATA
34              ;USED TO GENERATE THE TEST PATTERN.(THEY REPEAT EVERY
35              ;4 WORDS. THE 8TH WORD IS A RANDOM SEC.# THAT IS TREATED
36              ;SINGLY AS THE CONTIGUOS SECTORS IN WORD 2. IF THE LAST
37              ;SECTOR ON THE CYLINDER, READS AND WRITES FORCE THE EOC
38              ;STATUS AS EVERY OPERATION IS DONE WITH A SEC. COUNT=2
39              ;WHEN INITIALLY ENTERED, THE DISK TEST ATTEMPTS TO ACQUIRE 1
40              ;TO 4K OF SCRATCH AREA.  THE TEST THEN RANDOMLY SELECTS A DATA
41              ;START ADDRESS WITHIN THE FIRST 256 WORDS OF SCRATCH.
42              ;THE TEST THEN RANDOMLY SELECTS ONE OF THE AVAIL DISKS
43              ;THE TEST THEN RANDOMLY SELECTS ONE OF THE 3 OP TABLES. IF
44              ;THE FIRST WORD OF THE OP TABLE IS NOT=0 IT INDICATES THAT -
45              ;AT THE CYLINDER NUMBER IN THE FIRST WORD - STARTING WITH THE
46              ;SECTOR # IN BITS 6 TO 15 OF THE 2ND WORD - FOR THE # OF SECTORS
47              ;SPECIFIED BY IN BITS 2 TO 5 OF THE 2ND WORD - RANDOM DATA HAS
48              ;BEEN WRITTEN THAT IS EQUAL TO THE DATA IN WORDS 4 TO 7
49              ;OF THE OP TABLE.  READ FROM DISK IS SELECTED.
50              ;IF THE FIRST WORD OF THE TABLE IS=0 THE TEST - RANDOMLY
51              ;SELECTS A CYLINDER NOT CURRENTLY IN AN OP TABLE - RANDOMLY
52              ;SELECTS A START SECTOR (THE # OF SECTORS IS = TO THE AMOUNT
53              ;OF SCRATCH AVAILABLE) - AND GETS FOUR RANDOM DATA WORDS WRITE
54              ;TO DISK IS SELECTED
```

```
!0024 N3MRT
01              ;6.4.9 (CONTINUED)
02              ;THE TEST THEN INITIATES A SEEK TO THE CYLINDER SELECTED
03              ;AND AT SUCCESSFUL COMPLETION OF THE SEEK EITHER READS OR
04              ;WRITES THE # OF SECTORS AVAILABLE.
05              ;AT SUCCESSFUL COMPLETION OF EITHER THE READ OR WRITE, THE
06              ;DATA BUFFER IS CHECKED TO VERIFY THAT IT CONTAINS THE
07              ;CORRECT DATA.  AS DATA COMPARES CORRECTLY, THE CORRECT
08              ;WORDS ARE FILLED WITH THE NEGATIVE COUNT=TO THE NUMBER
09              ;OF WORDS LEFT IN THE BUFFER.
10
11      000000          .NOLOC MTTES
12              ;6.4.10 MAGNETIC TAPE OR CASSETTE TEST
13              ;
14              ;THE MULTIPROGRAMMING MAGNETIC TAPE AND CASSETTE TESTS ARE
15              ;IDENTICAL IN OPERATION.  THREE TO SIXTY THREE RECORDS
16              ;OF RANDOM DATA ARE WRITTEN, THE DATA BUFFER IS CHECKED,
17              ;THE MAG TAPE OR CASSETTE IS BACKSPACED TO THE BEGIN-
18              ;NING OF THE JUST WRITTEN RECORDS.  THEN, AS MANY RECORDS
19              ;AS THE SCRATCH AREA WILL CONTAIN ARE READ BACK, THE
20              ;DATA IS VERIFIED AND THE SEQUENCE (READ/VERIFY) IS RE-
21              ;PEATED UNTIL ALL RECORDS IN THE SEQUENCE HAVE BEEN READ.
22              ;
23              ;FOR TAPE WRITE STATUS ERRORS, THE TAPE IS BACKSPACED/RE-
24              ;WRITTEN UNTIL THE ERROR NO LONGER OCCURS.  FOR TAPE READ
25              ;STATUS ERRORS THE TEST BACKSPACES AND REREADS A TOTAL OF
26              ;3 TRYS.  STATUS ERRORS DURING BACKSPACE ARE CONSIDERED NON
27              ;RECOVERABLE.  FOR ALL ERRORS TYPED, THE TAPE IS REWOUND
28              ;AND THE TEST RESTARTED AT LOAD POINT.
29              ;
30              ;WHEN THE TAPE REACHES EOT DURING THE WRITE OPERATION, TAPE
31              ;IS REWOUND AND THE TEST RESTARTS AT LOAD POINT.
32              ;
33              ;THE RANDOM DATA IS A SEQUENCE OF 4 WORDS REPEATED EVERY 4TH
34              ;WORD.  ALL RECORDS ARE 256 WORDS IN LENGTH.  AS DATA IS
35              ;VERIFIED IN THE BUFFER IT IS REPLACED WITH A WORD EQUAL
36              ;TO THE NEGATIVE COUNT OF THE NUMBER OF WORDS LEFT
37              ;TO BE COMPARED.
38              ;
39              ;THESE TAPE TESTS UTILIZE 1 TO 6K OF SCRATCH AND THE
40              ;DATA BUFFER START IS RANDOMLY SELECTED TO BE IN THE FIRST 256
41              ;WORDS.
42              ;
43              ;ANY COMBINATION OF 1 TO 8 DRIVES MAY BE TESTED SIMPLY BY
44              ;HAVING THEM ON LINE WRITE ENABLED.
```

```
10025 N3MRT
01              ;6.4.11 LINE PRINTER TEST
02              ;
03              ;THE LINE PRINTER TEST RANDOMLY PRINTS 10 TO 60 LINES OF
04              ;PRINT PER PAGE WITH RANDOM STALLS EVERY 1 TO 9 LINES.
05              ;EACH LINE OF PRINT CONSISTS OF THE CHARACTERS SPACE
06              ;(40) TO Z (132).  THE TEST FILLS THE PRINT BUFFER UNTIL THE
07              ;FIRST PRINT CYCLE STARTS.  CONTINUATION OF PRINTING UNTIL
08              ;RANDOM STALL IS THEN RUN OFF INTERRUPTS  FROM THE PRINTER
09              ;
10              ;6.4.12 REAL TIME CLOCK
11              ;
12              ;THE REAL TIME CLOCK IS RUN AT 1K HERTZ.  RUNTIME ALONG
13              ;WITH ACCUMULATED ERROR COUNT ARE PRINTED AT 5 MINUTES
14              ;15 MINUTES, 30 MINUTES AND EVERY 30  MINUTES OF RUNTIME
15              ;THEREAFTER.  THIS TYPEOUT ALSO OCCURS AFTER EVERY ERROR
16              ;TYPEOUT OR IF TTY KEY 4 IS TYPED.
17              ;
18              ;6.4.13 TELETYPE TEST
19              ;
20              ;THE TELETYPE TEST PRINTS A SINGLE LINE CONSISTING OF THE
21              ;CHARACTERS SPACE TO Z.  THE TEST WILL ALSO ECHO CHARACTERS
22              ;AS TYPED.
23              ;
```

```
10026 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              ;       "^O"(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              ;
```

```
10027 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 PRINIT 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              ;       SUCCEDING 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              ;
```

```
10028 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" ENT RE
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.
```

```
10029 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                ;
```

```
10030 N3MRT
01
02
03
04
05                ;*************************************************************
06                ;
07                ; NAME: N3MORTL.SR                        PART NUMBER: 094-000752
08                ;
09                ;
10                ; DESCRIPTION: NOVA 3 MULTI-PROGRAMMING RELIABILITY TEST
11                ;                LONG VERSION
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                ;
```

```
10031 N3MRT
01
02                ;       .TITL   LINKR
03                ;DIAGNOSTIC PROGRAM LINKER
04        000000          .LOC 0
05 00000 027206           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 001274           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 004154           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 OP.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 PRTCK:    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 005654 IIOVL:    IOVAL     ;I/O VALIDITY TRAP HANDLER
44 00236 005762 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
```

```
10032 N3MRT
01                ;CALL HANDLER SUBROUTINE LINK
02 00063 002706 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 003177 ASCRA:    ASSCR     ;ASSIGN SCRATCH
08 00065 003216 ESCRA:    EXSCR     ;EXPAND SCRATCH
09 00066 003243 RSCRA:    RLSCR     ;RELEASE SCRATCH
10 00067 003270 GSCRA:    GOSCR
11 00070 004776 PCRLF:     CLF?     ;CAR. RETURN LINE FEED
12 00071 004714 PZOCT:     ZOC?     ;ZERO SUPRESS OCTAL
13 00072 004721 PDECI:    PDE?      ;DECIMAL PRINT
14 00073 004015 ERRTX:    ERTXT     ;TEXT TYPEOUT CALL
15 00074 002114 RETRN:    LRETP     ;RETURN FROM TEST
16 00075 004067 ERPAD:    EPADR
17 00076 004114 ERPAC:    EPACS
18 00077 002143 ARANG:    RANGN     ;RANDOM # GENERATOR
19 00100 002267 ADIVI:    DIVID     ;INTEGR UNSIGNED DIV
20 00101 005530 EINTS:     EINTP    ;ENTER INTR SERVICE
21 00102 003562 ERROI:     ERROH    ;INIT ERROR TYPEOUTS
22 00103 003775 ERROC:     ERROE    ;APPEND TO ERR TYPEOUT
23 00104 003514 SETUL:     SETLP    ;SET UP STRT OF LOOP
24 00105 003534 LLOOP:     LOOPL    ;LOOP BACK TO SETUP
25 00106 002212 FRANG:     RANG3    ;RAN #'S TO AC'S 0,1,AND 2
26 00107 003376 ERRET:     ERRRT    ;2ND LEVEL ERR RETURN
27 00110 003422 RETU2:     RETN2    ;2ND LEVEL NO ERR RETURN
28 00111 006423 RNDMP:     RANDCH   ;RANDOM SEL DCH MAP A/B
```

```
10033 N3MRT
01 00112 006442 ADMAP: AMSCR     ;ASSIGN SCR TO A DCH
02 00113 006532 EDMAP: EMSCR     ;EXPAND DATA CHANNEL ASSIGN
03 00114 006615 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 004776 ICLF?:  CLF?
18 00130 004721 IPDE?:  PDE?
19 00131 004714 IZOC?:  ZOC?
20 00132 004666 LMESS:  MES?
21       000070 LCRLF= PCRLF
22 00133 004715 LPOCT:  POC?
23       000071 LZOCT= PZOCT
24       000072 LPDEC= PDECI
25              ; POWER FAIL AUTO RESTART LINKS
26 00134 001246 PFAIL:  PWRUP
27 00135 000000 PFAIS:  0
28 00136 000000 MPSWT: 0         ;MAP EXIST SWITCH
29 00137 000000 ALTBL: 0         ;ADRS MEM ALLOCATION TABLES
30 00140 000000 PSTRT: 0         ;FIRST LOC OF TST
31 00141 000000 PENDA: 0         ;LAST LOC
32 00142 000000 RTTIM: 0         ;ELAPSED RUNTIME IN MINUTES
33 00143 000000 STATS:  0        ;MAP STATUS REG.
34 00144 177777 UDEVI: -1
35 00145 000000 TOTPK: 0
36 00146 177777 TIMSW: -1        ;=0 IT'S TIME TO PRINT
37 00147 000000 ERTOT: 0         ;ERROR ACCUMULATOR
38 00150 000000 EACTV:  0        ;180=TTO  181=LPT
39 00151 000000 LASTI: 0         ;CHAR FROM LAST INPUT FROM TTI
40              ;SCRATCH AREA SIZE PARAMETER LOCS FOR TEST USAGE
41 00152 000000 SCRLO: 0         ;LOWEST LOGICAL SCRATCH ADRS
42 00153 000000 SCRHI: 0         ;HIGHEST LOGICAL SCRATCH ADRS
43 00154 000000 DCHLO: 0         ;LOW LOG DCH ADRS
44 00155 000000 DCHHI: 0         ;HIGH LOG DCH ADRS
45       000156 LZMAX=DCHHI+1
46       000156 LPG0=.
47       061401 .DIAC PSH=061401
48       061601 .DIAC POP=061601
49       062401 .DUSR SAVE=062401
50       062601 .DUSR RTRN=062601
51       060001 .DIAC MTFP=60001
52       061001 .DIAC MTSP=61001
53       060201 .DIAC MFFP=60201
54       061201 .DIAC MFSP=61201
55       000004 .DUSR PRTY = 4
56       000047 TPADR=47
57       000046 TPLOC=46
```

```
10034 N3MRT
01
02              ;DEFINITIONS FOR USER STATUS TABLE ACCESS
03       001001         .DUSR USTZM=1001
04       001002         .DUSR USTSS=1002
05       001003         .DUSR USTES=1003
06       001004         .DUSR USTNM=1004
07              ;
08              ;LINKER MAIN LINE DISPATCH ROUTINE
09       001010         .LOC 1010
10 01010 102401 LINKR:  SUB 0,0,SKP      ;AUTO START ENTRY
11 01011 102000         ADC 0,0          ;MANUAL SELECT ENTRY
12 01012 126400         SUB 1,1
13 01013 044062         STA 1,USESW      ;SET USE MAP SW
14 01014 040420         STA 0,LAUTO      ;SET ENTRY TYPE SW
15 01015 102000         ADC 0,0
16 01016 040135         STA 0,PFAIS      ;SET NOT PWR/FA RESTART
17 01017 004405         JSR LDSPR        ;START DISPATCH
18 01020 001040         LTBL1    ;THROUGH INIT SEQ
19 01021 060177         INTEN
20 01022 004402 LRUNS:  JSR LDSPR
21 01023 001054         LTBL2    ;THROUGH RUN TABLE
22
23              ;DISPATCH ROUTINE
24              ;ENTER SUBROUTINES IN SEQ VIA TABLE SPEC BY (R3)
25              ;END OF EACH TABLE IS LRUNS WHICH WILL START US
26              ;BACK AT THE BEGINNING OF THE RUN TABLE
27 01024 021400 LDSPR:  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,USESW      ;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 LPG0
47                      ^1
48              LPG0=.
49                      .LOC LMEML
50                      0        ;TEST PASS CTR
51                      0        ;TEST ERROR CTR
52                      0        ;INTERRUPT TIMEOUT SWITCH
53              %
```

```
!0035 N3MRT
01
02                ;LTBL1-INIT SYSTEM DISPATCH TABLE
03                ;END OF TABLE IS LRUNS
04 01040 001306 LTBL1:  LSYSR    ;RESET SYS
05 01041 001444         GPRGK    ;DET # TESTS LOADED
06 01042 001365         LWSET    ;SET UP RUNNING CONS.
07 01043 005541         LCINT    ;INIT INTR VECTORS
08 01044 001457         LSIZE    ;SIZE MEMORY
09 01045 001623         MVETA    ;MOVE EXISM TO AVALM
10 01046 001642         UBL32    ;REMOVE USED CORE FROM AVALM
11 01047 001670         TINIT    ;INIT EACH TEST LOADED
12 01050 006757         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 001726 LTBL2:  LRANP    ;RANDOM PROG SELECT
19 01055 003054         LDMAP    ;LOAD MAP OPTION
20 01056 006713         LDCHL    ;CALC DCHLO/HI
21 01057 002063         LSTRP    ;START TEST RUNNING
22 01060 001150         LSTAT    ;ADJUST RUN STATISTICA
23 01061 002304         CHSTK
24 01062 001022         LRUNS
25 01063 000000         0
26 01064 000000         0
```

```
!0036 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 020151         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 040151         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 040146         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 004154 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 005223 IINP?:  INP?
52 01146 000023 J23:    23
53 01147 000064 CK.64:  64
```

```
!0037 N3MRT
01                      ;END OF TEST PASS, SEE IF ANY EXTRANEOUS ERRORS
02                      ;NEED TO BE REPORTED
03 01150 054473 LSTAT:  STA 3,LST.3
04 01151 024144         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 007270         UDEVT
09 01156 006071         JSR @LZOCT
10 01157 102000         ADC 0,0
11 01160 040144         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 006174         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 040146         STA 0,TIMSW     ;SO TIME WILL FOLLOW
24 01174 040232         STA 0,DCHER
25 01175 010147         ISZ ERTOT
26 01176 000401         JMP .+1
27 01177 030140 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 007222         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 LSXIT       ;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
```

```
!0038 N3MRT
01                      ;PRINT PASSES AND ERROR COUNTS BY INDIVIDUAL TEST
02 01215 006132         JSR @LMESS
03 01216 007235         STHDR
04 01217 024120 PRSTL:  LDA 1,CURPR
05 01220 010120         ISZ CURPR       ;STEP IT TO NEXT
06 01221 034421         LDA 3,ALZMAX    ;POINTS TO TEST 0
07 01222 137000         ADD 1,3         ;NOW TO  CURRENT TEST
08 01223 031400         LDA 2,0,3       ;GET "XX.00" ADRS
09 01224 151005         MOV 2,2,SNR     ;=0 IS TYPED ALL
10 01225 000414         JMP LSXIT               ;RETURN FROM WHENCE
11 01226 021375         LDA 0,-3,2      ;GET PASS CTR
12 01227 101005         MOV 0,0,SNR     ;SKP IF TEST EXECUTED
13 01230 000767         JMP PRSTL       ;TEST NEVER ENTERED
14 01231 006071         JSR @LZOCT      ;TYPE TEST#
15 01232 025375         LDA 1,-3,2      ;GET # PASSES
16 01233 006072         JSR @LPDEC      ;PRINT IT
17 01234 025376         LDA 1,-2,2      ;GET ACCUMULATED ERRS
18 01235 125004         MOV 1,1,SZR     ;=0 DON'T PRINT
19 01236 006072         JSR @LPDEC      ;PRINT # ERRS THIS TEST
20 01237 006070         JSR @LCRLF
21 01240 000757         JMP PRSTL
22 01241 002402 LSXIT:  JMP @LST.3
23 01242 000156 ALZMAX: LZMAX
24 01243 000000 LST.3:  0               ;SAVE AC3
25 01244 004000 LS.NN:  4000
```

```
!0039 N3MRT
01                ;POWER FAIL AUTO RESTART/ WAIT FOR POWER UP
02 01245 002134           JMP @PFAIL
03 01246 020000 PWRUP:    LDA 0,0
04 01247 024776           LDA 1,PWRUP-1
05 01250 122414           SUB# 1,0,SZR
06 01251 040430           STA 0,PWRS0       ;SAVE INTA ADRS
07 01252 044000           STA 1,0
08 01253 063777           SKPDZ CPU
09 01254 000777           JMP .-1
10 01255 006132           JSR @LMESS
11 01256 006411           PFTEX
12 01257 024422           LDA 1,PWRS0
13 01260 006133           JSR @LPOCT
14 01261 006132           JSR @LMESS
15 01262 026734           RTTEX+1
16 01263 024142           LDA 1,RTTIM
17 01264 006072           JSR @LPDEC
18 01265 024147           LDA 1,ERTOT
19 01266 125004           MOV 1,1,SZR
20 01267 006072           JSR @LPDEC
21 01270 102400           SUB 0,0
22 01271 034412           LDA 3,AGSTRT
23 01272 054751           STA 3,LST.3
24 01273 000720           JMP PRSTL-4
25 01274 102400 GSTRT:    SUB 0,0
26 01275 040135           STA 0,PFAIS
27 01276 042406           STA 0,@ALAUTO
28 01277 006406           JSR @ALDSPR
29 01300 001040           LTBL1
30 01301 000000 PWRS0:    0
31 01302 000404           JMP LSYSR
32 01303 001274 AGSTRT: GSTRT
33 01304 001034 ALAUTO: LAUTO
34 01305 001024 ALDSPR: LDSPR
```

```
!0040 N3MRT
01                ;LSYSR=RESET SYSTEM
02                ;SET LOGICAL PAGE 37=PHYS 37
03                ;SET MPSWT=0 NO MAP  =1'S IF MAP OPTION
04 01306 062677 LSYSR:    IORST
05 01307 102400           SUB 0,0
06 01310 040000           STA 0,0
07 01311 040136           STA 0,MPSWT
08 01312 040147           STA 0,ERTOT       ;CLR ACCUM ERRS
09 01313 040145           STA 0,TOTPK       ;
10 01314 040150           STA 0,EACTV
11 01315 040151           STA 0,LASTI
12 01316 040232           STA 0,DCHER
13 01317 126000           ADC 1,1
14 01320 044144           STA 1,UDEVI       ;UNEXPECTED INTERRUPT FLAG
15                ;CHECK FOR PARITY OPTION
16 01321 040117           STA 0,PRFLG       ;CLR PARITY EXISTS FLAG
17 01322 060304           NIOP PRTY         ;SETS EVEN PARITY MODE
18 01323 063404           SKPBN PRTY        ;BSY=1 IF PARITY EXISTS
19 01324 000405           JMP .+5           ;NO PARITY EXISTS
20 01325 060104           NIOS PRTY         ;SET ODD MODE
21 01326 060204           NIOC PRTY         ;CLR ANY ERRS
22 01327 060104           NIOS PRTY         ;ENABLE INTR
23 01330 044117           STA 1,PRFLG       ;SET PARITY FLAG
24 01331 024062           LDA 1,USESW       ;CHECK IF WANT TO USE MAP
25 01332 125004           MOV 1,1,SZR       ;SKP IS OK TO USE MAP
26 01333 001400           JMP 0,3           ;EXIT NO MAPPING
27 01334 060203           NIOC 3            ;DIAG. RESET
28 01335 030427           LDA 2,LS.K2       ;PAGE 37
29 01336 072002           DOB 2,2           ;MAP ENTRY
30 01337 060302           NIOP 2            ;SINGLE CYCLE CMD
31 01340 025000           LDA 1,0,2         ;READ LOC. 0
32 01341 125004           MOV 1,1,SZR       ;SKP IS MAYBE MAP EXISTS
33 01342 001400           JMP 0,3           ;NO MAP-EXIT
34 01343 102000           ADC 0,0           ;=-1
35 01344 040000           STA 0,0           ;SET LOC 0 TO -1
36 01345 060302           NIOP 2            ;SING. CYCLE
37 01346 031000           LDA 2,0,2         ;GET LOC 0
38 01347 044000           STA 1,0           ;RESTORE LOC 0
39 01350 150014           COM# 2,2,SZR      ;SKP IS MAP EXISTS
40 01351 001400           JMP 0,3           ;NO MAP EXIT
41 01352 050136           STA 2,MPSWT       ;MPSWT = -1
42 01353 030402           LDA 2,.+2         ;CHECK FOR PROTECTION OPTION
43 01354 000402           JMP .+2
44 01355 001356           .+1
45 01356 065403           DIB 1,3           ;READ VIOL. PC REG
46 01357 146404           SUB 2,1,SZR       ;SHD = AC2
47 01360 126000           ADC 1,1
48 01361 120000           COM 1,0           ;=-1 IF PROTECTION EXISTS
49 01362 040231           STA 0,PRTCK       ;SET FLAG
50 01363 001400           JMP 0,3           ;RETURN
51 01364 076000 LS.K2:    76000
```

```
!0041 N3MRT
01
02                ;LWSET=SET UP SYSTEM FOR RUNNING
03                ;DCHMAP AND INIT MAP OPTION TRAP LOCATIONS
04 01365 022444 LWSET:  LDA 0,@LW.K1     ;GET NMAX
05 01366 040454         STA 0,LSYTB
06 01367 040454         STA 0,LSETB
07 01370 024466         LDA 1,PROGK
08 01371 124400         NEG 1,1
09 01372 044440         STA 1,LW.C1
10 01373 030450 LWS.1:  LDA 2,LSETB
11 01374 102400         SUB 0,0
12 01375 041000         STA 0,0,2        ;SET UP MEM ALLOC
13 01376 151400         INC 2,2          ;TABLES FOR 1 PROG
14 01377 041000         STA 0,0,2        ;1ST 2 WRDS =0
15 01400 151400         INC 2,2          ;NXT 32=-1
16 01401 024432         LDA 1,LW.K4
17 01402 100000         COM 0,0
18 01403 041000 LWS.2:  STA 0,0,2
19 01404 151400         INC 2,2
20 01405 125404         INC 1,1,SZR      ;STORE -1 32 TIMES
21 01406 000775         JMP LWS.2
22 01407 050434         STA 2,LSETB      ;NEW END SYS TABLES
23 01410 010422         ISZ LW.C1
24 01411 000762         JMP LWS.1        ;DO ONE MORE TABLE
25 01412 042422         STA 0,@LW.K5     ;DCH ALLOC TABLE
26 01413 042422         STA 0,@LW.K5+1   ;AVAILABLE 32K
27 01414 042422         STA 0,@LW.K5+2
28 01415 042422         STA 0,@LW.K5+3   ;DCH 8 MAP ALLOC. TBL
29 01416 014425         DSZ LSETB        ;REAL END OF SYS TABLES
30 01417 024421         LDA 1,LW.K6
31 01420 030421         LDA 2,LW.K7
32 01421 100000         COM 0,0
33 01422 041000 LWS.3:  STA 0,0,2        ;CLEAR CORE EXIST MAP
34 01423 151400         INC 2,2
35 01424 125404         INC 1,1,SZR
36 01425 000775         JMP LWS.3
```

```
!0042 N3MRT
01
02                ;NOW SET UP JMP @'S IN MAP TRAPS
03 01426 020063         LDA 0,ICDIS
04 01427 040047         STA 0,TPADR
05 01430 001400         JMP 0,3
06 01431 001004 LW.K1: USTNM
07 01432 000000 LW.C1: 0
08 01433 177740 LW.K4: -32.
09 01434 027201 LW.K5: DCHM0
10 01435 027202         DCHM1
11 01436 027203          DCHM2
12 01437 027204          DCHM3
13 01440 177770 LW.K6: -8.
14 01441 027161 LW.K7: EXISM
15 01442 000000 LSYTB: 0
16 01443 000000 LSETB: 0
17
```

```
10043 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 01444 024410 GPRGK:  LDA     1,KLZMX ;LAST LINKER ZLOC
08 01445 022410         LDA     0,@ISTZM        ;LAST ZPAGE FILLED
09 01446 111000         MOV     0,2
10 01447 122400         SUB     1,0     ;AC0=NUMBER TESTS
11 01450 040406         STA     0,PROGK
12 01451 102400         SUB     0,0
13 01452 041000         STA     0,0,2
14 01453 001400         JMP     0,3
15 01454 000156 KLZMX: LZMAX
16 01455 001001 ISTZM: USTZM
17 01456 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 01457 054421 LSIZE:  STA     3,LS.S3
24 01460 004431         JSR     MSZ32   ;SIZE 0 TO 32K
25 01461 040420         STA     0,UP32L ;AC0=LAST ADRS IN 32K
26 01462 024702         LDA     1,LS.K2 ;5 BITS PHYS PAGE 37
27 01463 107400         AND     0,1
28 01464 125300         MOVS    1,1     ;AC1=LAST PHYS PAGE(32K)
29 01465 125220         MOVZR   1,1
30 01466 125220         MOVZR   1,1
31 01467 044413         STA     1,HIGHK ;IN CASE NOT 32K OR NO MAP
32 01470 101533         INCZL#  0,0,SNC ;SKP IS 32K
33 01471 000412         JMP     LSIZR   ;EXIT MEM <32K
34 01472 024136         LDA     1,MPSWT
35 01473 125005         MOV     1,1,SNR ;=0 IS NO MAP OPTION
36 01474 000407         JMP     LSIZR
37 01475 004461         JSR     MSG32   ;SIZE ABOVE 32K
38 01476 040404         STA     0,HIGHK ;# OF PHYS PAGES(1K)
39 01477 000404         JMP     LSIZR
40 01500 000000 LS.S3: 0
41 01501 000000 UP32L: 0
42 01502 000000 HIGHK: 0
43 01503 024117 LSIZR:  LDA 1,PRFLG     ;PARITY EXISTS?
44 01504 125005         MOV 1,1,SNR     ;SKP IS PARITY EXISTS
45 01505 002773         JMP @LS.S3      ;RETURN
46 01506 060204         NIOC PRTY       ;CLR ANY PARITY ERRS
47 01507 060104         NIOS PRTY       ;ENABLE PARITY INTR
48 01510 002770         JMP @LS.S3      ;RETURN
```

```
10044 N3MRT
01
02                ;MSZ32-MEMORY SIZER 32K
03                ;DOES NOT USE MAP OPTION
04                ;SETS EXIST BIT FOR CONTIGUOS MEM TO 32K
05                ;WILL NOT SIZE NNN CONTIGUOS MEM
06                ;ALSO ASSUMES THAT EXISM=0'S
07                ;I.E.-CMAPB SKIP ON RETURN IS NONSENSE
08                ; EXIT AC0=HIGHEST AVAIL ADRS.
09                ;
10 01511 054440 MSZ32:  STA     3,XMS32 ;SAVE
11 01512 030731         LDA     2,LSET8
12 01513 050435         STA     2,M32TE
13 01514 151400         INC     2,2
14 01515 025000         LDA     1,0,2   ;GET CELL
15 01516 120000         COM     1,0     ;CHNG BITS
16 01517 041000         STA     0,0,2
17 01520 021000         LDA     0,0,2   ;=COM MEM EXISTS
18 01521 122405         SUB     1,0,SNR ;AND WE'LL SKIP
19 01522 000424         JMP     M32SZ   ;LST WAS NONEXIST
20 01523 050425         STA     2,M32TE
21 01524 045000         STA     1,0,2   ;RESTORE CELL
22 01525 141000         MOV     2,0
23 01526 024427         LDA     1,K1K
24 01527 123400         AND     1,0
25 01530 122404         SUB     1,0,SZR ;SKP IS DONE ALL OF 1K
26 01531 000763         JMP     MSZ32+3
27                ;NOW SET BIT IN EXIST MAP
28 01532 141300         MOVS    2,0
29 01533 101200         MOVR    0,0
30 01534 101200         MOVR    0,0
31 01535 024415         LDA     1,K37
32 01536 123400         AND     1,0     ;AC0=PHYS PAGE #
33 01537 030414         LDA     2,KXIST ;ADRS EXIST TABLE
34 01540 006414         JSR     @MS.L1  ;SET EXIST BIT=1
35 01541 101001         MOV     0,0,SKP
36 01542 063077         HALT    ;*****CAN'T HAPPEN EXIST BIT HAD TO =0
37 01543 030405         LDA     2,M32TE ;AC1=LAST 1K TOP ADDRESS
38 01544 151533         INCZL#  2,2,SNC ;SKP IF LAST CELL =32K
39 01545 000747         JMP     MSZ32+3 ;NOT DONE SIZING
40 01546 020402 M32SZ:  LDA     0,M32TE ;AC0=HIGHEST AVAIL.
41 01547 002402         JMP     @XMS32
42 01550 000000 M32TE: 0
43 01551 000000 XMS32: 0
44 01552 000037 K37:   37
45 01553 027161 KXIST: EXISM
46 01554 002575 MS.L1: CMAPB
47 01555 001777 K1K:     1777
```

```
10045 N3MRT
01                ;MSG32-MEMORY SIZING TO ABOVE 32K
02                ;MAP OPTION MUST EXIST
03                ;USES SINGLE CYCLE MAPPING TO SIZE
04                ;EXIT IS WITH AC0=LAST PHYSICAL PAGE #
05                ;EXISM(EXIST MAP)MUST BE 0'S ABOVE 32K
06                ;ASSUMES MEMORY TO BE CONTIGUOS
07                ;
08 01556 054773 MSG32:  STA 3,XMS32
09 01557 020723         LDA 0,HIGHK
10 01560 101400         INC 0,0
11 01561 105300         MOVS 0,1
12 01562 125122         MOVZL 1,1,SZC   ;SKP IS NOT SIZED TO 128K
13 01563 000431         JMP MG32S
14 01564 034436         LDA 3,MPF32     ;BITS FOR 32K MAP
15 01565 117000         ADD 0,3         ;+PHYSICAL PAGE #
16 01566 076002         DOB 3,2         ;WRITE MAP ENTRY
17 01567 034433         LDA 3,MPF32     ;ADDRESS TO USE
18 01570 060302 MS.LL:  NIOP 2          ;SINGLE CYCLE COMMAND
19 01571 031400         LDA 2,0,3       ;GET CELL
20 01572 144000         COM 2,1         ;COMPLEMENT IT
21 01573 060302         NIOP 2
22 01574 045400         STA 1,0,3       ;STORE COMP.
23 01575 060302         NIOP 2
24 01576 025400         LDA 1,0,3       ;GET IT BACK
25 01577 060302         NIOP 2
26 01600 051400         STA 2,0,3       ;RESTORE ORIGINAL CELL
27 01601 146405         SUB 2,1,SNR     ;SKP IS CELL EXISTS
28 01602 000412         JMP MG32S       ;MEM IS SIZED
29 01603 175400         INC 3,3
30 01604 175133         MOVZL# 3,3,SNC
31 01605 000763         JMP MS.LL
32 01606 040674         STA 0,HIGHK     ;SAVE NEW TOP MEM
33 01607 030744         LDA 2,KXIST     ;EXIST MAP
34 01610 006744         JSR @MS.L1      ;SET EXIST BIT
35 01611 101001         MOV 0,0,SKP
36 01612 063077         HALT            ;CMAP GOOF!!!
37 01613 000744         JMP MSG32+1
38 01614 020666 MG32S:  LDA 0,HIGHK
39 01615 024405         LDA 1,MPF32
40 01616 030734         LDA 2,K37
41 01617 133000         ADD 1,2
42 01620 072002         DOB 2,2         ;LOG = PHYS
43 01621 002730         JMP @XMS32      ;RETURN
44 01622 076000 MPF32:  76000
```

```
10046 N3MRT
01                ;MVETA-MOVE THE EXIST MAP
02                ;INT THE AVAILABLE MAP POSITION
03 01623 054413 MVETA:  STA     3,XMVET
04 01624 030413         LDA     2,KEXMP ;A2=STR EXIST
05 01625 034413         LDA     3,KAVMP ;A3=STR AVAILABLE
06 01626 024413         LDA     1,KMX   ; FOR COUNTING
07 01627 021000         LDA     0,0,2   ;WORD
08 01630 041400         STA     0,0,3   ;TO WORD
09 01631 151400         INC     2,2
10 01632 175400         INC     3,3
11 01633 125404         INC     1,1,SZR ;SKP IS DONE ALL
12 01634 000773         JMP     .-5
13 01635 002401         JMP     @XMVET  ;
14 01636 001623 XMVET: MVETA
15 01637 027161 KEXMP: EXISM
16 01640 027171 KAVMP: AVALM
17 01641 177770 KMX:   -8.
18
19                ;U8L32-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                ;SU8R C8LIM IS USED TO CLEAR AVAILABLE BITS
25 01642 054416 U8L32:  STA     3,XU8L3 ;SAVE RETURN
26 01643 102400         SUB     0,0
27 01644 026415         LDA     1,@KNMAX        ;0 TO NMAX
28 01645 006422         JSR     @U8LIM  ;PROTECTS PROGRAMS
29 01646 022414         LDA     0,@KUP32        ;LAST ADRS IN 32K
30 01647 105000         MOV     0,1
31 01650 006417         JSR     @U8LIM  ;PROTECTS LOADER
32 01651 026412         LDA     1,@KSTSS        ;STRT SYMBOLS
33 01652 022412         LDA     0,@KSTES        ;END SYMBOLS
34 01653 006414         JSR     @U8LIM  ;PROTECTS SYMBOLS
35 01654 022411         LDA     0,@KLST8        ;STRT LINKER TABLES
36 01655 026411         LDA     1,@KLET8        ;END LINKER TABLES
37 01656 006411         JSR     @U8LIM  ;PROTECTS LINKER TABLES
38 01657 002401         JMP     @XU8L3  ;RETURN
39 01660 000000 XU8L3:  0
40 01661 001004 KNMAX:  USTNM
41 01662 001501 KUP32:  UP32L
42 01663 001002 KSTSS:  USTSS
43 01664 001003 KSTES:  USTES
44 01665 001442 KLST8:  LSYT8
45 01666 001443 KLET8:  LSET8
46 01667 002530 U8LIM:  C8LIM
```

```
10047 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 01670 054434 TINIT:  STA     3,XTINI
07 01671 102400         SUB     0,0
08 01672 040433         STA     0,NPROG ;PROG TO INIT
09 01673 026565         LDA     1,@LROGK
10 01674 034431         LDA     3,NPROG ;NEXT PROG TO INIT
11 01675 166415         SUB#    3,1,SNR ;SKP IS NOT DONE ALL
12 01676 002426         JMP     @XTINI  ;EXIT ALL PROGS INITED
13 01677 020557         LDA     0,LR.K1
14 01700 117000         ADD     0,3
15 01701 031400         LDA     2,0,3   ;GET INIT ADRS
16 01702 020135         LDA     0,PFAIS
17 01703 101024         MOVZ    0,0,SZR ;SKP IF P/F RESTART
18 01704 000407         JMP     TINIA   ;NOT RESTART
19 01705 021002         LDA     0,2,2   ;GET WAITI SWITCH
20 01706 101045         MOVO    0,0,SNR ;SKP=MAYBE DELETED
21 01707 000404         JMP     TINIA   ;RESTART NOT DELETED
22 01710 021377         LDA     0,-1,2
23 01711 101045         MOVO    0,0,SNR ;SKP=PROG WAS SELECTED
24 01712 000410         JMP     XTINI-2
25 01713 102460 TINIA:  SUBC    0,0
26 01714 041375         STA     0,-3,2  ;0 PASS K
27 01715 041376         STA     0,-2,2  ;0 ERR CTR
28 01716 041002         STA     0,2,2   ;CLEAR WAIT INT SW
29 01717 101003         MOV     0,0,SNC ;SKP IF P/FAIL RESTART
30 01720 041377         STA     0,-1,2  ;CLRS INTA WAIT CTR
31 01721 007000         JSR     @0,2    ;AND INIT THIS TEST
32 01722 010403         ISZ     NPROG   ;STEP TO NXT PROG
33 01723 000750         JMP     TINIT+3 ;AND DO AGN
34 01724 000000 XTINI:  0
35 01725 000000 NPROG:  0
36
```

```
10048 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 01726 054531 LRANP:  STA 3,LR.S3
10 01727 102400         SUB 0,0
11 01730 030526         LDA 2,LR.K1     ;START OF TEST LINKS
12 01731 035000 LPRL1:  LDA 3,0,2       ;GET TEST LINK
13 01732 175005         MOV 3,3,SNR     ;SKP IS TEST EXISTS
14 01733 000437         JMP LPRL2       ;NO ONE INTR DONE USE RAN
15 01734 025402         LDA 1,2,3       ;GET INTR SW
16 01735 044137         STA 1,ALTBL     ;SAVE FOR LATER
17 01736 125005         MOV 1,1,SNR     ;WAITING INT OR DISABLED?
18 01737 000430         JMP LPS1E       ;NO TRY NEXT TEST
19 01740 125103         MOVL 1,1,SNC    ;SKP=WAIT BIT STILL ON
20 01741 000435         JMP LPRGO       ;ENTER THIS TEST
21 01742 025777         LDA 1,-1,3      ;GET INTA ELAPSED TIMER
22 01743 125005         MOV 1,1,SNR     ;SKP IF ACTIVE
23 01744 000423         JMP LPS1E
24 01745 034142         LDA 3,RTTIM     ;GET CUR RUN TIME
25 01746 166414         SUB# 3,1,SZR    ;SKP=RUN TIME ERR
26 01747 000420         JMP LPS1E
27 01750 035000         LDA 3,0,2
28 01751 040120         STA 0,CURPR
29 01752 011777         ISZ -1,3
30 01753 011777         ISZ -1,3
31 01754 006132         JSR @LMESS      ;TIME ELAPSED MESS.
32 01755 002120         LP.TX
33 01756 024120         LDA 1,CURPR     ;PRINT PROG#
34 01757 006071         JSR @PZOCT
35 01760 024137         LDA 1,ALTBL
36 01761 006133         JSR @LPOCT
37 01762 102400         SUB 0,0
38 01763 040146         STA 0,TIMSW     ;FORCE TIME TYPE
39 01764 010147         ISZ ERTOT
40 01765 000401         JMP .+1
41 01766 000741         JMP LRANP+1
42 01767 101400 LPS1E:  INC 0,0
43 01770 151400         INC 2,2
44 01771 000740         JMP LPRL1       ;TRY NEXT TEST
45                ;NO TEST WAITING TO PROCESS INTERRUPT RANDOM SELECT
46 01772 006077 LPRL2:  JSR @ARANG
47 01773 105000         MOV 0,1
48 01774 032464         LDA 2,@LROGK    ;DIV RAN/#PROGS
49 01775 006100         JSR @ADIVI
```

```
10049 N3MRT
01                ;SEE IF NEXT RAN# FITS BETWEEN TEST ENTER LIMITS
02                ;BUT DO NOT ENTER TEST IF IT IS WAITING INTA
03 01776 040120 LPRGO:  STA 0,CURPR      ;CURRENT PROG #
04 01777 030457         LDA 2,LR.K1      ;LAST LINKR ZLOC+1
05 02000 113000         ADD 0,2          ;A2=PNTR TO PARAM ADRS
06 02001 035000         LDA 3,0,2
07 02002 054140         STA 3,PSTRT      ;START ADDRES OF PRG
08 02003 031001         LDA 2,1,2        ;GET STRT NXT PROG
09 02004 151005         MOV 2,2,SNR      ;=0 IS LAST PRG SEL
10 02005 032454         LDA 2,@LW.K9     ;AND WE USE NMAX
11 02006 050141         STA 2,PENDA      ;AS THE END OF PROG
12 02007 014141         DSZ PENDA        ;-1 FOR REAL END OF PROG
13 02010 031402         LDA 2,2,3
14 02011 151132         MOVZL# 2,2,SZC   ;C=1 IS WAITING INTR
15 02012 000760         JMP LPRL2        ;SELECT DIFF PROG
16 02013 030142         LDA 2,RTTIM
17 02014 151600         INCR 2,2
18 02015 151500         INCL 2,2
19 02016 051777         STA 2,-1,3
20                ;IF AC$3=1 DELETE ALL RANDOM SELECT DELAYS
21 02017 034060         LDA 3,RNSEL      ;SW SET?
22 02020 175004         MOV 3,3,SZR
23 02021 000411         JMP LPRGA        ;DELETE DELAYS=1
24 02022 006077         JSR @ARANG       ;GET NEW RANDOM
25 02023 034140         LDA 3,PSTRT
26 02024 025403         LDA 1,3,3        ;GET PRG LWR LIMIT
27 02025 122433         SUBZ# 1,0,SNC
28 02026 000744         JMP LPRL2
29 02027 025404         LDA 1,4,3        ;GET HIGH LIMIT
30 02030 122032         ADCZ# 1,0,SZC
31 02031 000741         JMP LPRL2
32 02032 020120 LPRGA:  LDA 0,CURPR      ;PROGRAM #
33 02033 103020         ADDZ 0,0         ;*2
34 02034 105120         MOVZL 0,1        ;*4
35 02035 127120         ADDZL 1,1        ;*16
36 02036 125120         MOVZL 1,1        ;*32
37 02037 107000         ADD 0,1          ;*34
38 02040 022422         LDA 0,@XSYTB     ;+START OF SYS TABLES
39 02041 123000         ADD 1,0
40 02042 040137         STA 0,ALTBL
41 02043 020136         LDA 0,MPSWT
42 02044 101005         MOV 0,0,SNR      ;USE MAP?
43 02045 000407         JMP .+7          ;NOPE!
44 02046 006077         JSR @ARANG
45 02047 126520         SUBZL 1,1        ;BIT 6=1 IS USER MAP B
46 02050 123700         ANDS 1,0         ;IF BIT 15 = 1
47 02051 105120         MOVZL 0,1        ;THEN AC0=1000 =USER B
48 02052 044115         STA 1,USRMP      ;
49 02053 101300         MOVS 0,0         ;
50 02054 040116         STA 0,USRSE      ;AC0 = 000001=USER B FOR STATUS
51 02055 002402         JMP @LR.S3
52 02056 000156 LR.K1: LZMAX
53 02057 000000 LR.S3: 0
54 02060 001456 LROGK:  PROGK
55 02061 001004 LW.K9: USTNM
56 02062 001442 XSYTB: LSYTB
```

```
10050 N3MRT
01                ;LSTRP-START PROGRAM
02                ;ENTER TEST SELECTED AT ITS EXECUTION ENTRY POINT
03 02063 054432 LSTRP:  STA 3,L.SS3
04 02064 060104         NIOS PRTY        ;ENABLE PARITY INTR
05 02065 030140         LDA 2,PSTRT
06 02066 021001         LDA 0,1,2        ;GET EXEC ADRS
07 02067 040427         STA 0,LS.I1
08 02070 024136         LDA 1,MPSWT
09 02071 125005         MOV 1,1,SNR      ;DON'T USE MAP IF MAPSWT=0
10 02072 002424         JMP @LS.I1
11 02073 044237         STA 1,RELLO
12 02074 044240         STA 1,RELHI
13 02075 060277         INTDS
14 02076 020143         LDA 0,STATS      ;GET STATUS
15 02077 101220         MOVZR 0,0
16 02100 101120         MOVZL 0,0        ;SHIFT OUT BIT 15
17 02101 030116         LDA 2,USRSE      ;GET USER INFO
18 02102 143000         ADD 2,0          ;ADD INTO STATUS WORD
19 02103 024414         LDA 1,LS.I2      ;GET ENABLE BITS
20 02104 100000         COM 0,0
21 02105 107400         AND 0,1
22 02106 106000         ADC 0,1          ;OR IN PGM+DCH ENABLE BITS
23 02107 065002         DOA 1,2          ;SET UP STATUS REG FOR MAP
24 02110 102400         SUB 0,0
25 02111 040232         STA 0,DCHER      ;CLR DCH ERR CTR
26 02112 060177         INTEN            ;STALL
27 02113 002403         JMP @LS.I1       ;ENTER TEST
28                ;LRETP-RETURN FROM TEST PROG CALL
29 02114 002401 LRETP:  JMP @L.SS3       ;
30 02115 000000 L.SS3: 0
31 02116 000000 LS.I1: 0
32 02117 140000 LS.I2:  140000                   ;PGM AND DCH MAP ENABLE
33 02120 005215 LP.TX:  .TXTE (<15><12>INTERRUPT WAIT ELAPSED
34 02134 005215 <15><12>PROG. NO. (
```

```
;0051 N3MRT
01                ;RANGN-RANDOM # GENERATOR
02                ;SPIN #'S OUT IN A HURRY FORGET THE MATH
03 02143 044430 RANGN:  STA 1,RN.S1
04 02144 050430         STA 2,RN.S2
05 02145 030431         LDA 2,RN.K1      ;7 FOR MASKING AT 8
06 02146 020431         LDA 0,RN.C2      ;CYCLIC CONSTANT
07 02147 024426         LDA 1,RN.C1
08 02150 133404         AND 1,2,SZR      ;ROTAT C2 EVERY 8
09 02151 000404         JMP RAN.1
10 02152 101122         MOVZL 0,0,SZC
11 02153 101400         INC 0,0
12 02154 040423         STA 0,RN.C2
13 02155 024424 RAN.1:  LDA 1,RTABL
14 02156 133000         ADD 1,2          ;TO GET NXT SUM VAR
15 02157 025000         LDA 1,0,2
16 02160 123000         ADD 1,0
17 02161 041000         STA 0,0,2        ;NEW SUM IN VAR
18 02162 024416         LDA 1,RANNM      ;LAST RAN#
19 02163 123300         ADDS 1,0
20 02164 040124         STA 0,RNAC0
21 02165 040413         STA 0,RANNM
22 02166 024405         LDA 1,RN.S1
23 02167 030405         LDA 2,RN.S2
24 02170 010405         ISZ RN.C1
25 02171 001400         JMP 0,3
26 02172 001400         JMP 0,3
27 02173 000000 RN.S1: 0
28 02174 000000 RN.S2: 0
29 02175 000000 RN.C1: 0
30 02176 000007 RN.K1: 7
31 02177 123456 RN.C2: 123456
32 02200 000000 RANNM: 0
33 02201 002202 RTABL: RTABL+1
34 02202 027247         027247
35 02203 145651         145651
36 02204 162724         162724
37 02205 071352         071352
38 02206 034565         034565
39 02207 116272         116272
40 02210 047135         047135
41 02211 113523         113523
42 02212 054411 RANG3:  STA 3,RN.S3      ;FILL AC0 TO 2 WITH RAN #'S
43 02213 004730         JSR RANGN
44 02214 111000         MOV 0,2
45 02215 004726         JSR RANGN
46 02216 105000         MOV 0,1
47 02217 004724         JSR RANGN
48 02220 044125         STA 1,RNAC1
49 02221 050126         STA 2,RNAC2
50 02222 002401         JMP @RN.S3
51 02223 000000 RN.S3: 0
```

```
;0052 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 AC0=BIT#
12                ;THIS ROUTINE IS USED BY ASCRA ESCRA AND ADMAP
13                ;TO SELECT SCRATCH AND DATA CHANNEL ASSIGNMENTS
14 02224 054441 RMSEL:  STA 3,XRMSE
15 02225 023400         LDA 0,@0,3
16 02226 025401         LDA 1,1,3
17 02227 040433         STA 0,RM.P1
18 02230 044433         STA 1,RM.P2
19 02231 100000         COM 0,0
20 02232 040432         STA 0,RM.P3
21 02233 006077 RM.L1:  JSR @ARANG       ;GET RANDOM
22 02234 030426         LDA 2,RM.P1
23 02235 151400         INC 2,2
24 02236 105000         MOV 0,1
25 02237 006100         JSR @ADIVI       ;REM IN AC0=BIT SEL
26 02240 030423         LDA 2,RM.P2
27 02241 006425         JSR @ICMPB       ;SKIP IF BIT =1
28 02242 006424         JSR @ICMPB       ;CHNG IT BK TO 0 SKP
29 02243 000415         JMP RM.FN        ;FOUND 1 EXIT
30 02244 010420         ISZ RM.P3
31 02245 000766         JMP RM.L1
32 02246 020416 RM.L2:  LDA 0,RM.P3      ;GET NXT SEQ BIT
33 02247 006417         JSR @ICMPB       ;SKP IF =1
34 02250 006416         JSR @ICMPB       ;0 BIT AND SKIP
35 02251 000407         JMP RM.FN
36 02252 024410         LDA 1,RM.P1
37 02253 010411         ISZ RM.P3
38 02254 106404         SUB 0,1,SZR      ;SKP IF SRCHED WHOLE TBL
39 02255 000771         JMP RM.L2
40 02256 034407         LDA 3,XRMSE
41 02257 001402         JMP 2,3          ;NO FIND EXIT
42 02260 010405 RM.FN:  ISZ XRMSE
43 02261 000775         JMP RM.FN-2
44 02262 000000 RM.P1: 0
45 02263 000000 RM.P2: 0
46 02264 000000 RM.P3: 0
47 02265 000000 XRMSE: 0
48 02266 002575 ICMPB: CMAPB
```

```
10053 N3MRT
01                ;DIVID-DIVIDE AC1 BY AC2
02                ;LEAVE WITH REM IN AC0
03 02267 102400 DIVID:  SUB     0,0
04 02270 054412         STA     3,DI.S3
05 02271 034412         LDA     3,DI.K1
06 02272 125120         MOVZL   1,1
07 02273 101100 DI.L1:  MOVL    0,0
08 02274 142412         SUB#    2,0,SZC
09 02275 142400         SUB     2,0
10 02276 125100         MOVL    1,1
11 02277 175404         INC     3,3,SZR
12 02300 000773         JMP     DI.L1
13 02301 002401         JMP     @DI.S3
14 02302 000000 DI.S3: 0
15 02303 177760 DI.K1: -16.
```

```
10054 N3MRT
01
02                ;CHANGE STACK POSITIONS IN MEMORY
03 02304 054452 CHSTK:  STA 3,CHSAV
04 02305 006077         JSR @ARANG
05 02306 101203         MOVR 0,0,SNC    ;BITS 14/15=00
06 02307 101202         MOVR 0,0,SZC    ;THEN CHANGE STACK
07 02310 000454         JMP SPLAY       ;PLAY WITH STACK
08                ;CHECK TO SEE IF STACK OK BEFORE CHANGE
09 02311 071201         MFSP 2
10 02312 024445         LDA 1,STADR
11 02313 050445         STA 2,NWSTK
12 02314 132414         SUB# 1,2,SZR    ;SKP=STACK OK
13 02315 004537         JSR SKERR       ;STACK IS FUNNY
14 02316 151300         MOVS 2,2
15 02317 151225         MOVZR 2,2,SNR   ;SKP IF STK NOT AT 400
16 02320 000421         JMP CHPAG       ;MOVE IT ELSEWHERE
17                ;STACK IS NOT AT 400 RELEASE 1K PAGE
18                ;AFTER RESTORING STACK TO ADRS 400
19 02321 034441         LDA 3,CHSK1     ;400
20 02322 054435         STA 3,STADR
21 02323 060277         INTDS
22 02324 075001         MTSP 3
23 02325 060177         INTEN
24 02326 074001         MTFP 3
25 02327 030414         LDA 2,CHPAG+2   ;AVALM
26 02330 020431         LDA 0,STKPG     ;PHYS PAG#
27 02331 101005         MOV 0,0,SNR     ;SHD NOT BE PAG0
28 02332 004522         JSR SKERR       ;STACK IS IN ERROR
29 02333 006430         JSR @CHSK1+1    ;CMAPB MAKE 1K AVAL
30 02334 102401         SUB 0,0,SKP     ;;BIT HAD TO GO 0 TO1
31 02335 004517         JSR SKERR       ;THIS CAN'T HAPPEN
32 02336 040423         STA 0,STKPG
33 02337 002417         JMP @CHSAV
34
35
36 02340 000037 STKLM:  37      ;LIMIT STACK TO 32K
37                ;RANDOMLY SELECT 1K OF CORE TO MOV THE STACK INTO
38 02341 004663 CHPAG:  JSR RMSEL       ;SELECTS 1K CORE
39 02342 002340         STKLM           ;WITHIN LIMIT
40 02343 027171         AVALM           ;OF AVAL CORE
41 02344 002412         JMP @CHSAV      ;NONE AVAL TO USE
42 02345 040414         STA 0,STKPG     ;SAVE PHYS PAGE
43 02346 101300         MOVS 0,0
44 02347 103120         ADDZL 0,0
45 02350 040407         STA 0,STADR
46 02351 060277         INTDS
47 02352 060001         MTFP 0
48 02353 060177         INTEN
49 02354 061001         MTSP 0
50 02355 002401         JMP @CHSAV
```

```
!0055 N3MRT
01
02 02356 000000 CHSAV:  0
03 02357 000000 STADR:  0
04 02360 000000 NWSTK: 0
05 02361 000000 STKPG:  0
06 02362 000400 CHSK1:  400
07 02363 002575         CMAPB
08
09              ;DO SOME FUNNY THINGS WITH STACK TO
10              ;TRY TO GOOF IT UP
11 02364 071201 SPLAY:  MFSP 2
12 02365 126000         ADC 1,1 ;-1
13 02366 133000         ADD 1,2
14 02367 044411         STA 1,STKIN-1   ;STK INTA FLG
15 02370 060277         INTDS
16 02371 071001         MTSP 2          ;RELOAD STACK POINTER
17 02372 061401         PSH 0
18 02373 025001         LDA 1,1,2
19 02374 060177         INTEN
20 02375 122414         SUB# 1,0,SZR    ;(AC0) GET THERE OK
21 02376 004456         JSR SKERR       ;NO! PSH ERROR
22 02377 002757         JMP @CHSAV
23
24              ;STACK OVERFLOW ROUTINE
25 02400 000000         0
26 02401 040427 STKIN:  STA 0,STSV0
27 02402 044427         STA 1,STSV1
28 02403 050427         STA 2,STSV2
29 02404 054427         STA 3,STSV3
30 02405 101200         MOVR 0,0
31 02406 040426         STA 0,STSVC
32 02407 010771         ISZ STKIN-1
33 02410 000402         JMP .+2
34 02411 000406         JMP .+6
35 02412 006441         JSR @SKERR-1                    ;PRINT ERR MESS.
36 02413 002435         STKTZ
37 02414 024000         LDA 1,0
38 02415 006435         JSR @SKERR-2
39 02416 006411         JSR @ST.K6      ;WAIT FOR OPERATOR INPUT
40 02417 034414         LDA 3,STSV3
41 02420 030412         LDA 2,STSV2
42 02421 024410         LDA 1,STSV1
43 02422 020412         LDA 0,STSVC
44 02423 101100         MOVL 0,0
45 02424 020404         LDA 0,STSV0
46 02425 060177         INTEN
47 02426 002000         JMP @0          ;RETURN
48
49 02427 004033 ST.K6:  KEY6W
50 02430 000000 STSV0: 0
51 02431 000000 STSV1: 0
52 02432 000000 STSV2: 0
53 02433 000000 STSV3: 0
54 02434 000000 STSVC: 0
55 02435 005215 STKTZ:  .TXTE !<15><12>STACK OVERFLOW ERROR @!
56       000000         .NOLOC 0
```

```
!0056 N3MRT
01
02              ;STACK IS IN ERROR SOMEHOW
03              ;MAKE ERROR TYPEOUT
04 02452 004715         POC?
05 02453 004666         MES?
06 02454 054020 SKERR:  STA 3,20
07 02455 006776         JSR @SKERR-1
08 02456 002504         STKTX
09 02457 024020         LDA 1,20        ;ADRS ERR DETECTED
10 02460 006772         JSR @SKERR-2
11 02461 024676         LDA 1,STADR     ;WHERE STACK SHD BE
12 02462 006770         JSR @SKERR-2
13 02463 024675         LDA 1,NWSTK
14 02464 006766         JSR @SKERR-2    ;WHERE IT'S AT
15              ;RESTORE STACK TO 400
16 02465 034675         LDA 3,CHSK1
17 02466 060277         INTDS
18 02467 075001         MTSP 3
19 02470 074001         MTFP 3
20 02471 060177         INTEN
21 02472 054665         STA 3,STADR
22 02473 020666         LDA 0,STKPG     ;MAKE 1K AVAL
23 02474 101005         MOV 0,0,SNR     ;IF NOT PAGE 0
24 02475 002661         JMP @CHSAV
25 02476 030645         LDA 2,CHPAG+2
26 02477 006664         JSR @CHSK1+1
27 02500 102401         SUB 0,0,SKP     ;IT'S RELEASED
28 02501 000776         JMP .-2
29 02502 040657         STA 0,STKPG
30 02503 002653         JMP @CHSAV
31 02504 005215 STKTX:  .TXTE !<15><12>STACK ERROR<15><12>
32 02513 142412 ERROR@  EXPECTED ACTUAL<15><12>!
```

```
!0057 N3MRT
01
02                ;CBLIM=CLEAR AVAILABLE BITS BETWEEN LIMITS
03                ;AC0=LOWEST ADRS    AC1=HIGHEST ADRESS
04                ;SINCE SOME USED AREAS MAY OVERLAP IN PAGES
05                ;OCCASIONALLY 2 PASSES THROUGH CMAP8 WILL BE REQ
06 02530 054426 CBLIM:  STA     3,XCBLM
07 02531 101300         MOVS    0,0       ;POS 1K
08 02532 125300         MOVS    1,1       ;FIELD BITS
09 02533 101220         MOVZR   0,0       ;FOR ADRS LIMITS
10 02534 101220         MOVZR   0,0       ;IN AC 0 AND 1
11 02535 125220         MOVZR   1,1
12 02536 125220         MOVZR   1,1
13 02537 030424         LDA     2,K37C
14 02540 143400         AND     2,0
15 02541 147400         AND     2,1
16 02542 040417         STA     0,CBLWR ;LWST 1K FLD
17 02543 044417         STA     1,CBUPR ;HGHST 1K FLD
18 02544 020415         LDA     0,CBLWR ;LWR INCS TO=UPR
19 02545 030413         LDA     2,KAVLM ;AVAILABLE MAP
20 02546 006411         JSR     @XCMP8  ;COM BIT IN AVAIL MAP
21 02547 000777         JMP     .-1     ;WENT 0-1 MAKE IT 1-0
22 02550 020411         LDA     0,CBLWR ;AC0=LST CLRED
23 02551 010410         ISZ     CBLWR   ;+1 LWR IN CASE NOT DONE
24 02552 024410         LDA     1,CBUPR
25 02553 106404         SUB     0,1,SZR ;SKP IS ALL REQ UNAVAILABLE
26 02554 000770         JMP     .-10
27 02555 002401         JMP     @XCBLM
28 02556 000000 XCBLM:  0
29 02557 002575 XCMP8:  CMAP8
30 02560 027171 KAVLM:  AVALM
31 02561 000000 CBLWR:  0
32 02562 000000 CBUPR:  0
33 02563 000037 K37C:   37
```

```
!0058 N3MRT
01                ;GETPA-GET A PHYSICAL ASSIGNMENT
02                ;AC0=ALLOCATION TABLE POSITION
03                ;RETURN WITH AC1=PHYSICAL PAGE ASSIGNMENT
04                ;SKIP EXIT IF THE ALLOC ASSIGN ISN'T ACCESS PROTECTED
05                ;THE INTEGRETY OF AC0 IS PRESERVED
06 02564 040410 GETPA:  STA     0,GPA.0
07 02565 030137         LDA     2,ALTBL ;ADRS OF ALLOCATION TABLE
08 02566 113000         ADD     0,2     ;+ WORD POSITION IN TABLE
09 02567 025002         LDA     1,2,2   ;2 PAST CTR AND DCH CONTR.
10 02570 124014         COM#    1,1,SZR ;ACCESS PROTECTED ?
11 02571 175400         INC     3,3     ;SKP EXIT PAGE ASSIGNED
12 02572 020402         LDA     0,GPA.0
13 02573 001400         JMP     0,3
14 02574 000000 GPA.0:  0
15
16                ;CMAP8-COMPLIMENT MAP BIT
17                ;COMPLIMENT THE STATE OFAMEMORY MAP BIT
18                ;THE START ADRS OF THE MAP IS INAC2
19                ;AC0 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 02575 040436 CMAP8:  STA     0,CM.S0
25 02576 044436         STA     1,CM.S1
26 02577 050436         STA     2,CM.S2
27 02600 105220         MOVZR   0,1     ;POSITION WORD # IN AC1
28 02601 125220         MOVZR   1,1
29 02602 125220         MOVZR   1,1
30 02603 125220         MOVZR   1,1
31 02604 133000         ADD     1,2     ;AC2=TABLE (MAP) ADRESS
32 02605 024431         LDA     1,K17
33 02606 107400         AND     0,1     ;AC1=#PLACES TO SHIFT
34 02607 124000         COM     1,1
35 02610 044422         STA     1,CCTR  ;FOR COUNTING SHIFTS
36 02611 126420         SUBZ    1,1     ;AC1=0  C=1
37 02612 125100         MOVL    1,1     ;POSIT BIT
38 02613 010417         ISZ     CCTR
39 02614 000776         JMP     .-2
40 02615 021000         LDA     0,0,2   ;GET MAP WORD
41 02616 050414         STA     2,CCTR  ;SV ADRS
42 02617 131000         MOV     1,2     ;FOR BIT XOR
43 02620 113525         ANDZL   0,2,SNR ;NOT 0 IN RESULT
44 02621 151002         MOV     2,2,SZC ;OR IN CARRY BIT
45 02622 175400         INC     3,3     ;IS SKIP WHEN EXIT
46 02623 107000         ADD     0,1     ;FORM REST OF
47 02624 146400         SUB     2,1     ;BIT XOR
48 02625 046405         STA     1,@CCTR ;PUT NEW WORD BACK
49 02626 020405         LDA     0,CM.S0
50 02627 024405         LDA     1,CM.S1
51 02630 030405         LDA     2,CM.S2
52 02631 001400         JMP     0,3
53 02632 000000 CCTR: 0
54 02633 000000 CM.S0: 0
55 02634 000000 CM.S1: 0
56 02635 000000 CM.S2: 0
57 02636 000017 K17:    17
```

```
!0059 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
```

```
!0060 N3MRT
01                ;ILLEGAL SUPERVISOR CALL TYPEOUT
02 02637 006132 ICALL:  JSR @LMESS
03 02640 002665         ICALT           ;TEXT
04 02641 024576         LDA 1,CD.LA
05 02642 006133         JSR @LPOCT
06 02643 020566         LDA 0,CD.S0
07 02644 024566         LDA 1,CD.S1
08 02645 030566         LDA 2,CD.S2
09 02646 006102         JSR @ERROI      ;TYPE PR# AC'S ETC
10 02647 000401         JMP .+1
11 02650 020564         LDA 0,CD.S3
12 02651 024570         LDA 1,CD.PP
13 02652 030564         LDA 2,CD.IN     ;INSTRUCTION CAUSING TRAP
14 02653 006103         JSR @ERROC
15 02654 000401         JMP .+1
16 02655 030140         LDA 2,PSTRT
17 02656 035006         LDA 3,6,2
18 02657 175120         MOVZL 3,3
19 02660 024062         LDA 1,USESW
20 02661 125004         MOV 1,1,SZR     ;USE ICALX RETURN IF USESW NOT = 0
21 02662 000420         JMP ICALX
22 02663 002401         JMP @.+1
23 02664 006033         IOV.E
24 02665 005215 ICALT:  .TXTE (<15><12>
25 02666 144411         ILLEGAL SUPER CALL AT (
26                      ;
27                      ;
28                      ;
29                      ;ICALX- TRAP RETURN IF ILLEGAL CALL BECAUSE OF USESW
30                      ;
31 02702 175220 ICALX:  MOVZR 3,3       ;AC0-1-2 = MAP REGISTERS
32 02703 054402         STA 3,ICLX.3    ;
33 02704 002401         JMP @ICLX.3
34 02705 000000 ICLX.3: 0
```

```
10061 N3MRT
01                ;SYSTEM CALL DISPATCH ROUTINE
02 02706 040523 CDISP:  STA 0,CD.S0
03 02707 044523         STA 1,CD.S1
04 02710 050523         STA 2,CD.S2
05 02711 054523         STA 3,CD.S3
06 02712 020136         LDA 0,MPSWT
07 02713 101004         MOV 0,0,SZR     ;SKIP IS DON'T USE MAP
08 02714 000403         JMP .+3
09 02715 034046         LDA 3,TPLOC     ;PC OF TRAPPED INSTRUCTION
10 02716 000440         JMP CDGLC       ;MAP WASN'T USED
11 02717 060277         INTDS
12 02720 030231         LDA 2,PRTCK     ;PROECTION OPTION EXIST?
13 02721 151005         MOV 2,2,SNR     ;SKP=EXIST
14 02722 000414         JMP CDIST       ;NO PROTECT OPTION
15 02723 063402         SKPBN 2         ;ANY DCH ERRS?
16 02724 000403         JMP .+3         ;NOT YET
17 02725 010232         ISZ DCHER       ;AT LEAST ONE HAS OCCURED
18 02726 000401         JMP .+1
19 02727 071402         DIB 2,2         ;CHECK FOR A VIOL.
20 02730 151133         MOVZL# 2,2,SNC  ;SKIP IS A VIOLATION
21 02731 000404         JMP .+4
22 02732 075403         DIB 3,3         ;GET VIOL. PC REG.
23 02733 054233         STA 3,VLPCR     ;SAVE IT
24 02734 050234         STA 2,VLDTA     ;SAVE VIOL. DATA
25 02735 060202         NIOC 2          ;CLR ANY VIOLATIONS
26 02736 060402 CDIST:  DIA 0,2         ;GET STATUS
27 02737 040143         STA 0,STATS     ;SAVE IT
28 02740 126620         SUBZR 1,1
29 02741 125220         MOVZR 1,1       ;FORM DCH ENABLE BIT USER A
30 02742 101232         MOVZR# 0,0,SZC  ;SKP IF USER A
31 02743 024501         LDA 1,CD.40
32 02744 065002         DOA 1,2         ;WRITE STATUS WORD
33 02745 151133         MOVZL# 2,2,SNC  ;SKIP IS VIOL
34 02746 000405         JMP .+5
35 02747 024476         LDA 1,CD.44     ;MASK TO USE
36 02750 133415         AND# 1,2,SNR    ;SKIP IS A I/O OR VALIDITY VIOL.
37 02751 002236         JMP @IDWCK      ;MUST BE A WRITE/DEFER VIOL.
38 02752 002235         JMP @IIOVL      ;WAS AN I/O OR VALIDITY VIOL.
39 02753 034046         LDA 3,TPLOC     ;GET TRAP PC
40 02754 060177         INTEN
41 02755 060302         NIOP 2          ;SINGLE CYCLE CMD
42 02756 021400 CDGLC:  LDA 0,0,3       ;GET CALL
43 02757 054460         STA 3,CD.LA     ;SV LGICL ADRS
44 02760 040456         STA 0,CD.IN     ;SAVE INSTRUCTION CAUSING TRAP
45 02761 030454         LDA 2,K1777
46 02762 150000         COM 2,2
47 02763 157400         AND 2,3
48 02764 054454         STA 3,CD.LP     ;SAVE LOG PAGE
49 02765 024136         LDA 1,MPSWT
50 02766 125005         MOV 1,1,SNR     ;OK TO USE MAP?
51 02767 000407         JMP .+7         ;NOPE!
52 02770 024115         LDA 1,USRMP
53 02771 137000         ADD 1,3
54 02772 075003         DOA 3,3         ;SELECT PAGE
55 02773 074403         DIA 3,3         ;READ MAP ENTRY TO GET PHYS PG #
56 02774 150220         COMZR 2,2
57 02775 157400         AND 2,3
58 02776 054443         STA 3,CD.PP     ;SAVE PHYS. PG#
59                ;NOW DETERMINE IF IT IS A VALID CALL
60 02777 030443         LDA     2,CALLS ;START OF CALLS
```

```
 0062 N3MRT
01 03000 142433         SUBZ#   2,0,SNC ;MUST BE=>
02 03001 000636         JMP     ICALL           ;ILLEGAL CALL?
03 03002 024441         LDA     1,CALLE ;LAST VALID CALL
04 03003 122032         ADCZ#   1,0,SZC ;MUST BE =<
05 03004 000633         JMP     ICALL           ;INVALID CALL?
06 03005 142640         SUBOR 2,0       ;CREATE CALL ADRS
07 03006 115220         MOVZR 0,3       ;MOV 4 R
08 03007 175220         MOVZR   3,3
09 03010 175220         MOVZR   3,3     ;(AC3)=CALL#
10 03011 020420         LDA     0,CD.S0
11 03012 024420         LDA     1,CD.S1
12 03013 030420         LDA     2,CD.S2
13 03014 007464 CD.EX:  JSR     @ASCRA,3        ;CALL JSR
14 03015 101011         MOV#    0,0,SKP ;ERROR RET
15 03016 010421         ISZ     CD.LA   ;NORMAL +1 RETURN
16 03017 010420         ISZ     CD.LA   ;TO GET PAST JSR
17 03020 034136         LDA 3,MPSWT
18 03021 175005         MOV 3,3,SNR
19 03022 000403         JMP .+3
20 03023 034143         LDA 3,STATS
21 03024 060277         INTDS
22 03025 075002         DOA 3,2         ;RESTORE MAP STATUS
23 03026 034406         LDA 3,CD.S3
24 03027 060177         INTEN
25 03030 002407         JMP     @CD.LA
26                ;ABOVE JMP RETURNS TO USER
27 03031 000000 CD.S0: 0
28 03032 000000 CD.S1: 0
29 03033 000000 CD.S2: 0
30 03034 000000 CD.S3: 0
31 03035 001777 K1777: 1777
32 03036 000000 CD.IN:  0
33 03037 000000 CD.LA: 0
34 03040 000000 CD.LP: 0
35 03041 000000 CD.PP: 0
36                CALLS:  LCALL ASCRA
37 03042 100010           ASCRA-ASCRA*1B11+100010
38                CALLE:  LCALL RDMAP
39 03043 100610           RDMAP-ASCRA*1B11+100010
40 03044 040040 CD.40:  40040
41 03045 000044 CD.44:  44
```

```
10063 N3MRT
01                ;ENTPA=ENTER PHYSICAL ASSIGNMENT
02                ;THE PHYSICAL PAGE # IN AC0 IS ENTERED INTO
03                ;THE MEMORY ALLOCATION TABLE ASSOCIATED WITH
04                ;THE TEST THAT IS CURRENTLY ACTIVE
05 03046 026137 ENTPA:  LDA     1,@ALTBL        ;GET # ENTRIES
06 03047 030137         LDA     2,ALTBL ;ADDRS OF TBL
07 03050 133000         ADD     1,2     ;+WORD #
08 03051 041002         STA     0,2,2   ;PLACE ENTRY IN TABLE
09 03052 012137         ISZ     @ALTBL  ;+1# OF ENTRIES
10 03053 001400         JMP     0,3
```

```
10064 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 03054 062401         SAVE
06 03055 102400         SUB     0,0
07 03056 040152         STA     0,SCRLO
08 03057 040153         STA     0,SCRHI
09 03060 024136         LDA     1,MPSWT ;=0 IS MAP OPT. NOT EXIST
10 03061 125005         MOV     1,1,SNR
11 03062 000477         JMP     LM.NM   ;MAP OPT NONEXIS
12 03063 020115         LDA 0,USRMP     ;ADD USER
13 03064 062002         DOB 0,2         ;MAP PAGE 0 TO ITSELF
14 03065 102400         SUB 0,0
15 03066 101400 LM.L1:  INC 0,0
16 03067 024465         LDA 1,LM.K1     ;LOAD PAGES 1-37
17 03070 111300         MOVS 0,2
18 03071 153120         ADDZL 2,2
19 03072 034115         LDA 3,USRMP
20 03073 173000         ADD 3,2
21 03074 133000         ADD 1,2         ;ADD IN 777
22 03075 072002         DOB 2,2         ;VALIDITY PROTECT THIS PAGE
23 03076 030457         LDA 2,LM.37
24 03077 112414         SUB# 0,2,SZR    ;DONE?
25 03100 000766         JMP     LM.L1
26                ;ALL PAGES BUT LOG 0(IT ='S0) ACCESS PROTECTED
27                ;MAP TEST PROGRAM TO ITSELF-MUST BE LESS THAN 1K
28 03101 024140         LDA     1,PSTRT ;START ADRS OF PROG
29 03102 121300 LM.L2:  MOVS    1,0
30 03103 101220         MOVZR   0,0     ;1K FIELD TO
31 03104 101220         MOVZR   0,0     ;LOWER 5 BITS
32 03105 024450         LDA     1,LM.37
33 03106 123400         AND     1,0
34 03107 105300         MOVS    0,1
35 03110 127120         ADDZL 1,1
36 03111 107000         ADD 0,1
37 03112 030115         LDA 2,USRMP     ;GET USER INFO
38 03113 147000         ADD 2,1         ;PUT IN MAP ENTRY WORD
39 03114 066002         DOB 1,2         ;LOG=PHYS
40                      .ENDC
41 03115 105700         INCS    0,1
42 03116 127120         ADDZL   1,1
43 03117 030141         LDA     2,PENDA ;LAST USED BY TEST
44 03120 132432         SUBZ#   1,2,SZC ;C=0 IS MAPPED LAST ALRDY
45 03121 000761         JMP     LM.L2
46                ;ENTER NEXT SEQUENCE WITH AC0=HIGHEST PAGE USED FOR TEST
```

```
!0065 N3MRT
01                ;IF TEST HAS SCRATCH AREA ASSIGNED MAP AS MUCH AS POSSIBLE
02                ;STARTING AT THE FIRST 1K ABOVE PROGRAM STORAGE
03 03122 026137           LDA     1,@ALTBL        ;# 1K FIELDS ASSIGNED
04 03123 125005           MOV     1,1,SNR
05 03124 000427           JMP     LM.DN   ;NO SCRATCH EXIT
06 03125 101400           INC     0,0
07 03126 105300           MOVS    0,1
08 03127 127120           ADDZL   1,1
09 03130 044152           STA     1,SCRLO ;START OF ACCESSABLE SCRAT
10 03131 040425           STA     0,LM.TM
11 03132 102400           SUB     0,0
12 03133 006424 LM.L3:    JSR     @LGTPA  ;NEXT PHYS PAGE ASSIGNED
13 03134 000417           JMP     LM.DN   ;DONE ALL EXIT
14 03135 030421           LDA     2,LM.TM ;AC2=LOGICAL PG. #
15 03136 151300           MOVS    2,2
16 03137 153120           ADDZL   2,2
17 03140 147000           ADD     2,1
18 03141 034115           LDA     3,USRMP
19 03142 167000           ADD     3,1
20 03143 066002           DOB     1,2     ;LOG= PHYS
21 03144 024414           LDA     1,LM.K2 ;K2=1777
22 03145 133000           ADD     1,2
23 03146 050153           STA     2,SCRHI ;NEW HI SCRATCH LIMIT
24 03147 101400           INC     0,0     ;+1 ALLOC TBL POS
25 03150 010406           ISZ     LM.TM   ;+1 LOGICAL PAGE
26 03151 151523           INCZL   2,2,SNC ;C=1 IF HI IS 77777(32K)
27 03152 000761           JMP     LM.L3
28                ;MAP OPTION IS SET UP FOR A FIRST LEVEL TEST
29                LM.DN:
30 03153 062601           RTRN
31 03154 000777 LM.K1:    777
32 03155 000037 LM.37:    37
33 03156 000000 LM.TM:    0
34 03157 002564 LGTPA:    GETPA
35 03160 001777 LM.K2:    1777
```

```
!0066 N3MRT
01                ;MAP OPTION DOES NOT EXIST
02                ;SIMPLY SET LIMITS TO SCRATCH AREA ASSIGNED
03 03161 006776 LM.NM:    JSR     @LGTPA  ;SKP=AC1 PHYS PAGE#
04 03162 000771           JMP     LM.DN   ;EXIT NO SCRATCH
05 03163 125300           MOVS    1,1
06 03164 127120           ADDZL   1,1
07 03165 044152           STA     1,SCRLO ;LOW=FIRST PHYS 1K
08 03166 006771 LM.L4:    JSR     @LGTPA  ;SKP=AC1=PHYS PG#
09 03167 000764           JMP     LM.DN   ;EXIT SCRHI ADJUSTED
10 03170 125300           MOVS    1,1
11 03171 127120           ADDZL   1,1     ;PG# POSITIONED TO PHYS
12 03172 030766           LDA     2,LM.K2
13 03173 133000           ADD     1,2
14 03174 050153           STA     2,SCRHI ;NO TEST CAN HAVE 32K IF
15 03175 101400           INC     0,0
16 03176 000770           JMP     LM.L4   ;MAP OPTION NONEXIST
```

```
!0067 N3MRT
01                ;ASSCR-ASIGN 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 03177 062401           SAVE
08 03200 054414           STA     3,AS.S3
09 03201 022137           LDA     0,@ALTBL        ;GET #1K'S ASSIGNED
10 03202 101004           MOV     0,0,SZR ;NOT=0 INVALID
11 03203 000407           JMP     AS.XT
12 03204 006411 AS.G1:    JSR     @MSEL   ;SELECT A PAGE
13 03205 001502           HIGHK           ;MAX # 1K PAGES
14 03206 027171           AVALM           ;AVAILABLE MAP
15 03207 000403           JMP     AS.XT   ;NO CORE AVAILABLE
16 03210 004636           JSR     ENTPA   ;AC0=PHYS PAGE# ENTER IT
17 03211 012403           ISZ @AS.S3
18 03212 004642 AS.XT:    JSR     LDMAP   ;LOAD MAP OPT. SET SCRLO+HI
19 03213 062601           RTRN
20 03214 000000 AS.S3: 0
21 03215 002224 MSEL: RMSEL
```

```
!0068 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 03216 062401           SAVE
08 03217 054775           STA     3,AS.S3
09 03220 022137           LDA     0,@ALTBL        ;# 1K'S ASSIGNED
10 03221 101005           MOV     0,0,SNR
11 03222 000770           JMP     AS.XT   ;CANT EXPAND 0 ASSIGNED
12 03223 024136           LDA     1,MPSWT ;=0 IS NO MAP OPT.
13 03224 125004           MOV     1,1,SZR ;SKP ON NO MAPPING
14 03225 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 03226 100400           NEG     0,0
18 03227 100000           COM     0,0     ;CALC ALLOC TBLE POS
19 03230 006436           JSR     @IGTPA  ;EXTRACT PHYS PGE #
20 03231 063077           HALT            ;#=-1 CAN'T HAPPEN
21 03232 121400           INC     1,0     ;AC0=NXT PHYS PAGE
22 03233 030753           LDA     2,AS.G1+2       ;ADRS OF AVAILABLE TBL
23 03234 006406           JSR     @EX.I1  ;CMAPB SKPS IF AVAILABLE
24 03235 101001           MOV     0,0,SKP
25 03236 000752           JMP     AS.XT-2 ;1K AVAILABLE ENTER AND REMAP
26 03237 006403           JSR     @EX.I1  ;CMAPB HAS TO SKP
27 03240 063077           HALT
28 03241 000751           JMP     AS.XT   ;EXIT NO EXPANSION
29 03242 002575 EX.I1: CMAPB
```

```
10069 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 03243 062401           SAVE
07 03244 054750           STA     3,AS.S3
08 03245 022137           LDA     0,@ALTBL        ;#1K'S ASSIGNED
09 03246 100405           NEG     0,0,SNR         ;SKP IF ANY ASSIGNED
10 03247 000743           JMP     AS.XT           ;EXIT NONE TO RELESE
11 03250 100000           COM     0,0             ;AC0=#1K'S -1
12 03251 042137           STA     0,@ALTBL        ;TO ENTER 377 LATER
13 03252 006414           JSR     @IGTPA          ;GET PHYS PAGE #
14 03253 063077           HALT                    ;ASSIGNED CAN'T=-1
15 03254 121000           MOV     1,0
16 03255 030731           LDA     2,AS.G1+2       ;A2=AVAILABLE MAP
17 03256 006764           JSR     @EX.I1          ;CMAPB MAKES 1K AVAIL
18 03257 101001           MOV     0,0,SKP
19 03260 063077           HALT                    ;BIT FOR THAT 1K HAD TO =0
20 03261 102000           ADC     0,0
21 03262 006405           JSR     @NTPA           ;PUT -1 IN ALLOCATION TBL
22 03263 016137           DSZ     @ALTBL          ;-1 # PAGES ASSIGNED
23 03264 000725           JMP     AS.XT-1         ;STILL PAGES LEFT+1 EXIT
24 03265 000725           JMP     AS.XT           ;0 MEM ALLOCATED DON'T SKIP
25 03266 002564 IGTPA:    GETPA
26 03267 003046 NTPA: ENTPA
```

```
10070 N3MRT
01                ;GOSCR - GO TO SCRATCH
02                ;ENTERED WITH AC0=LOGICAL PAGE TO
03                ;REMAP SCRATCH TO AC1=ERROR RET
04                ;AC2=ADDRESS TO START EXECUTION IN
05                ;THE REMAPPED SCRATCH
06
07 03270 040556 GOSCR:    STA 0,GO.00     ;SAVE CALL PARAMS
08 03271 044556           STA 1,GO.01
09 03272 050556           STA 2,GO.02
10 03273 054556           STA 3,GO.S3
11 03274 022561           LDA 0,@GO.K2
12 03275 040561           STA 0,GO.LP
13 03276 022561           LDA 0,@GO.K3
14 03277 040562           STA 0,GO.LA
15 03300 034136           LDA 3,MPSWT
16 03301 175005           MOV 3,3,SNR     ;SKP MAP OPT EXISTS
17 03302 002547           JMP @GO.S3      ;GO BACK TO TEST
18 03303 036137           LDA 3,@ALTBL
19 03304 175005           MOV 3,3,SNR     ;SKP SCRATCH ASSIGN
20 03305 002544           JMP @GO.S3      ;NO SCRATCH RETRN
21 03306 004560           JSR GO.CU       ;CHANGE USERS
22 03307 020152           LDA 0,SCRLO
23 03310 040543           STA 0,GO.SLO    ;SAVE SCRLO
24 03311 020153           LDA 0,SCRHI
25 03312 040542           STA 0,GO.SHI    ;SAVE SCRHI
26 03313 004440           JSR LDMP2       ;LOAD USER MAP
27 03314 102400           SUB 0,0
28 03315 024531           LDA 1,GO.00
29 03316 125300           MOVS 1,1
30 03317 127120           ADDZL 1,1       ;NEW SCRLO FOR REMAP
31 03320 044152           STA 1,SCRLO
32 03321 044237           STA 1,RELLO
33 03322 006530 GO.L1:    JSR @GO.K1      ;GET PHYS ASSIGN
34 03323 000422           JMP GO.GO       ;NO MORE AVAIL
35 03324 030522           LDA 2,GO.00
36 03325 151300           MOVS 2,2
37 03326 153120           ADDZL 2,2
38 03327 147000           ADD 2,1
39 03330 034115           LDA 3,USRMP
40 03331 167000           ADD 3,1
41 03332 066002           DOB 1,2         ;WRITE MAP ENTRY
42 03333 024417           LDA 1,GO.1K
43 03334 147000           ADD 2,1
44 03335 044153           STA 1,SCRHI     ;NEW SCR HI FOR REMAP
45 03336 044240           STA 1,RELHI
46 03337 010507           ISZ GO.00       ;+1 LOG PAGE
47 03340 101400           INC 0,0         ;+1 ALLOCATION POS
48 03341 115300           MOVS 0,3
49 03342 177120           ADDZL 3,3
50 03343 175123           MOVZL 3,3,SNC   ;LAST LOG=37 DONE
51 03344 000756           JMP GO.L1
52 03345 060277 GO.GO:    INTDS
53 03346 020143           LDA 0,STATS     ;GET SAVED STATUS
54 03347 061002           DOA 0,2         ;WRITE TO STATUS REGISTER
55 03350 060177           INTEN
56 03351 002477           JMP @GO.02
57 03352 001777 GO.1K: 1777
```

```
!0071 N3MRT
01                ;LDMP2=LOAD MAP FOR LEVEL 2
02                ;
03 03353 054420 LDMP2:  STA 3,LD.S3      ;SAVE RETURN
04 03354 020115         LDA 0,USRMP
05 03355 062002         DOB 0,2          ;MAP PAGE 0 TO ITSELF
06 03356 102400         SUB 0,0
07 03357 101400 LD.L1:  INC 0,0
08 03360 024415         LDA 1,LD.K1      ;LOAD PAGES 1 - 37
09 03361 111300         MOVS 0,2
10 03362 153120         ADDZL 2,2
11 03363 034115         LDA 3,USRMP
12 03364 173000         ADD 3,2
13 03365 133000         ADD 1,2          ;ADD IN 777
14 03366 072002         DOB 2,2          ;VALIDITY PROTECT THIS PAGE
15 03367 030405         LDA 2,LD.37
16 03370 112414         SUB# 0,2,SZR     ;DONE?
17 03371 000766         JMP LD.L1                ;NOT YET
18 03372 002401         JMP @LD.S3               ;RETURN TO CALLER
19 03373 000000 LD.S3:  0
20 03374 000037 LD.37:  37
21 03375 000777 LD.K1: 777
```

```
!0072 N3MRT
01                ;ERRRT - 2ND LEVEL ERROR RETURN
02
03 03376 054453 ERRRT:  STA 3,GO.S3
04 03377 034136         LDA 3,MPSWT
05 03400 177020         ADDZ 3,3         ;C=1 IF MAP
06 03401 034446         LDA 3,GO.01
07 03402 056455         STA 3,@GO.K3
08 03403 016454         DSZ @GO.K3       ;PREPARE CD.LA FOR ISZ IN CDISP
09 03404 175002         MOV 3,3,SZC
10 03405 000403         JMP .+3
11 03406 036452         LDA 3,@ERRK2     ;GET AC3 AT CALL
12 03407 002440         JMP @GO.01       ;NO MAP OPT GO DRECT
13 03410 040452         STA 0,ERSV0
14 03411 044452         STA 1,.ERSV1
15 03412 050452         STA 2,ERSV2
16 03413 004453         JSR GO.CU        ;CHANGE USERS
17 03414 006451         JSR @ERRK1       ;RELOAD ORIG MAP
18 03415 020445         LDA 0,ERSV0
19 03416 024445         LDA 1,.ERSV1
20 03417 030445         LDA 2,ERSV2
21 03420 036440         LDA 3,@ERRK2     ;GET AC3 AT CALL
22 03421 002430         JMP @GO.S3       ;RETURN TO TEST ERR VIA CDISP
23
24                ;RETN2 - NORMAL 2ND LEVEL END OF TEST RET
25
26 03422 010427 RETN2:  ISZ GO.S3        ;+1 RETURN ADDRESS
27 03423 034433         LDA 3,GO.LP
28 03424 056431         STA 3,@GO.K2     ;RESTORE LOGICAL
29 03425 034434         LDA 3,GO.LA      ;PAGE AND ADDRS
30 03426 056431         STA 3,@GO.K3     ;OF ORIGINAL GSCRA
31 03427 040433         STA 0,ERSV0
32 03430 044433         STA 1,.ERSV1
33 03431 050433         STA 2,ERSV2
34 03432 034136         LDA 3,MPSWT
35 03433 175005         MOV 3,3,SNR
36 03434 002415         JMP @GO.S3       ;NO MAP GO DIRECT
37 03435 004431         JSR GO.CU        ;CHANGE USERS
38 03436 020415         LDA 0,GO.SLO     ;
39 03437 040152         STA 0,SCRLO
40 03440 020414         LDA 0,GO.SHI
41 03441 040153         STA 0,SCRHI      ;RESTORE SCRLO/HI
42 03442 020420         LDA 0,ERSV0
43 03443 024420         LDA 1,.ERSV1
44 03444 030420         LDA 2,ERSV2
45 03445 002404         JMP @GO.S3       ;RETURN TO 1ST LEVEL TST
```

```
!0073 N3MRT
01
02 03446 000000 GO.00:  0
03 03447 000000 GO.01:  0
04 03450 000000 GO.02:  0
05 03451 000000 GO.S3:  0
06 03452 002564 GO.K1:  GETPA
07 03453 000000 GO.SLO: 0
08 03454 000000 GO.SHI: 0
09 03455 003040 GO.K2:  CD.LP
10 03456 000000 GO.LP:  0
11 03457 003037 GO.K3:  CD.LA
12 03460 003034 ERRK2:  CD.S3
13 03461 000000 GO.LA:  0
14
15 03462 000000 ERSV0:  0
16 03463 000000 .ERSV1:  0
17 03464 000000 ERSV2:  0
18 03465 003054 ERRK1:  LDMAP
19
20              ;GO.CU- COMPLEMENT TO OTHER USER MAP
21              ;
22 03466 102520 GO.CU:  SUBZL 0,0
23 03467 101300         MOVS 0,0
24 03470 101120         MOVZL 0,0       ;MOVE TO BIT 6
25 03471 024115         LDA 1,USRMP
26 03472 131000         MOV 1,2         ;XOR AC0 WITH AC1
27 03473 113520         ANDZL 0,2
28 03474 107000         ADD 0,1
29 03475 146400         SUB 2,1
30 03476 044115         STA 1,USRMP     ;CHANGES USER
31 03477 102520         SUBZL 0,0       ;BIT 15 = USER B
32 03500 024116         LDA 1,USRSE
33 03501 131000         MOV 1,2
34 03502 113520         ANDZL 0,2
35 03503 107000         ADD 0,1
36 03504 146400         SUB 2,1
37 03505 044116         STA 1,USRSE     ;CHANGE USER STATUS BIT
38 03506 020143         LDA 0,STATS     ;GET STATUS WORD
39 03507 101220         MOVZR 0,0
40 03510 101120         MOVZL 0,0       ;CHANGE USER BIT
41 03511 123000         ADD 1,0
42 03512 040143         STA 0,STATS
43 03513 001400         JMP 0,3         ;RETURN TO CALLER
```

```
!0074 N3MRT
01              ;SETLP - SET UP LOOP CALL HANDLER
02              ;PERFORMS SAME FUNCTION AS SETUP IN NORMAL TSTS
03              ;ENTERED VIA JSR @SETUL
04
05 03514 040437 SETLP:  STA 0,ST.S0
06 03515 044437         STA 1,ST.S1
07 03516 024437         LDA 1,ST.K1
08 03517 136414         SUB# 1,3,SZR
09 03520 000412         JMP STNMP       ;NOT AN LCALL
10 03521 022435         LDA 0,@ST.K2    ;GET LOG AORS
11 03522 040123         STA 0,ST.LA     ;FOR LOOPL
12 03523 026434         LDA 1,@ST.K3    ;AND LOG PAGE
13 03524 044122         STA 1,ST.LP
14 03525 020433 SETXI:  LDA 0,ST.LK     ;-4
15 03526 040121         STA 0,ST.LC     ;FOR LOOP RPT COUNT
16 03527 020424         LDA 0,ST.S0
17 03530 024424         LDA 1,ST.S1
18 03531 001400         JMP 0,3
19              ;LOOP SETUP WAS NOT VIA LCALL
20
21 03532 054123 STNMP:  STA 3,ST.LA
22 03533 000772         JMP SETXI
23
24              ;LOOPL - PERFORMS SAME FUNCTION AS LOOP
25              ;ENTERED VIA JSR @LLOOP
26
27 03534 010121 LOOPL:  ISZ ST.LC       ;SKIP IS FINI LOOP
28 03535 101001         MOV 0,0,SKP     ;LOOP BACK
29 03536 001400         JMP 0,3         ;CONTINUE ON
30 03537 040414         STA 0,ST.S0
31 03540 020415         LDA 0,ST.K1     ;CHK FOR
32 03541 116415         SUB# 0,3,SNR    ;SUPER CALL
33 03542 000403         JMP .+3         ;SUPER CALL
34 03543 020410         LDA 0,ST.S0     ;NOT LCALL
35 03544 002123         JMP @ST.LA      ;JUST CONTINUE
36 03545 020123         LDA 0,ST.LA     ;LOGICAL START LOOP
37 03546 042410         STA 0,@ST.K2
38 03547 020122         LDA 0,ST.LP     ;IN LOGICAL PAGE
39 03550 042407         STA 0,@ST.K3
40 03551 020402         LDA 0,ST.S0
41 03552 001400         JMP 0,3
42
43 03553 000000 ST.S0:  0
44 03554 000000 ST.S1:  0
45 03555 003015 ST.K1:  CD.EX+1
46 03556 003037 ST.K2:  CD.LA
47 03557 003040 ST.K3:  CD.LP
48 03560 177774 ST.LK:  -4.
```

```
!0075 N3MRT
01                      ;ERROH - ERROR HANDLER - PRINT ALL ERR INFO
02 03561 000010                 8.
03                      ;FIRST PRINT PRG# AND (AC'S)
04 03562 040572 ERROH:  STA 0,ER.S0
05 03563 020776         LDA 0,ERROH-1
06 03564 044571         STA 1,ER.S1
07 03565 050571         STA 2,ER.S2
08 03566 054571         STA 3,ER.S3
09 03567 030140         LDA 2,PSTRT
10 03570 011376         ISZ -2,2
11 03571 113000         ADD 0,2
12 03572 050426         STA 2,ERTIT
13 03573 101000         MOV 0,0
14 03574 010147         ISZ ERTOT       ;+1#ERROR CALLS
15 03575 101000         MOV 0,0
16 03576 020147         LDA 0,ERTOT
17 03577 024552         LDA 1,ER50.
18 03600 122414         SUB# 1,0,SZR    ;HALT AFTER FIRST 50 ERRORS
19 03601 000404         JMP .+4
20 03602 006132         JSR @LMESS
21 03603 007200         FTYTX
22 03604 006554         JSR @IWAIT
23 03605 102400         SUB 0,0
24 03606 040146         STA 0,TIMSW     ;SO TIME TYPE WILL FOLLOW
25 03607 034230         LDA 3,SWREG
26 03610 101100         MOVL 0,0
27 03611 103102         ADDL 0,0,SZC    ;SW2=1 NO TYPE
28 03612 000550         JMP EREXI
29 03613 006132         JSR @LMESS
30 03614 006220         TXT.0
31 03615 024120         LDA 1,CURPR     ;GET PROG #
32 03616 006071         JSR @LZOCT      ;PRINT IT
33 03617 006132         JSR @LMESS
34 03620 006227 ERTIT:  TXT.1           ;PRINTS TEST TITL
35 03621 006132         JSR @LMESS
36 03622 006227         TXT.1
37 03623 024531         LDA 1,ER.S0     ;PRINT AC'S
38 03624 006133         JSR @LPOCT      ;AT ERROR CALL
39 03625 024530         LDA 1,ER.S1
40 03626 006133         JSR @LPOCT
41 03627 024527         LDA 1,ER.S2
42 03630 006133         JSR @LPOCT
```

```
!0076 N3MRT
01                      ;PRINT MEM ALLOCATION ASSIGNMENTS
02 03631 006132 ERMPP:  JSR @LMESS
03 03632 006233         TXT.2
04 03633 024152         LDA 1,SCRLO
05 03634 006133         JSR @LPOCT      ;PRINT SCRATCH LIMITS
06 03635 024153         LDA 1,SCRHI
07 03636 006133         JSR @LPOCT
08 03637 024136         LDA 1,MPSWT     ;GET MAP EXIST
09 03640 125005         MOV 1,1,SNR     ;TYPE MEM ALLOC IF MAP
10 03641 000521         JMP EREXI       ;NO MAP FORGET REST OF TYPE
11 03642 020237         LDA 0,RELLO
12 03643 100015         COM# 0,0,SNR    ;SKP=RELOCATED TEST
13 03644 000407         JMP NOTRL
14 03645 006132         JSR @LMESS      ;PRINT RELLO/HI
15 03646 006403         TXT.B
16 03647 024237         LDA 1,RELLO
17 03650 006133         JSR @LPOCT
18 03651 024240         LDA 1,RELHI
19 03652 006133         JSR @LPOCT
20                      NOTRL:
21 03653 020116         LDA 0,USRSE
22 03654 101004         MOV 0,0,SZR
23 03655 000404         JMP .+4         ;USER B
24 03656 006132         JSR @LMESS
25 03657 006305         TXT.5
26 03660 000403         JMP .+3
27 03661 006132         JSR @LMESS
28 03662 006311         TXT.6
29 03663 020155         LDA 0,DCHHI
30 03664 101005         MOV 0,0,SNR
31 03665 000407         JMP ERNDC       ;DON'T PRINT DCH LOC
32 03666 006132         JSR @LMESS
33 03667 006241         TXT.3
34 03670 024154         LDA 1,DCHLO     ;PRINT DCH LIMITS
35 03671 006133         JSR @LPOCT
36 03672 024155         LDA 1,DCHHI
37 03673 006133         JSR @LPOCT
38 03674 020061 ERNDC:  LDA 0,DLTBL
39 03675 101004         MOV 0,0,SZR     ;SKP IS PRINT ALLOCATION TABLE
40 03676 000464         JMP EREXI
41 03677 020153         LDA 0,SCRHI     ;SKP=MEM ALLOCATED TO THIS TEST
42 03700 101005         MOV 0,0,SNR
43 03701 000461         JMP EREXI
44 03702 032137         LDA 2,@ALTBL
45 03703 050447         STA 2,ER.C2     ;SAVE # 1K'S ALLOCATED
46 03704 102400         SUB 0,0
47 03705 040442         STA 0,ER.C1     ;CLR COUNTER
48 03706 006132         JSR @LMESS
49 03707 006246         TXT.4
50 03710 024237         LDA 1,RELLO
51 03711 124014         COM# 1,1,SZR    ;SKP=NOT RELLOCATED TEST
52 03712 000404         JMP .+4
53 03713 006132         JSR @LMESS
54 03714 006341         TXT.9
55 03715 000403         JMP ERMPL
56 03716 006132         JSR @LMESS
57 03717 006357         TXT.A
58 03720 020427 ERMPL:  LDA 0,ER.C1     ;PRINT PHYS PAGES
59 03721 006427         JSR @ER.K4      ;ALSO IN MODULO 1K
60 03722 000440         JMP EREXI
```

```
 0077 N3MRT
01 03723 101213          MOVR# 0,0,SNC
02 03724 006070          JSR @PCRLF
03 03725 006071          JSR @LZOCT
04 03726 020421          LDA 0,ER.C1
05 03727 103120          ADDZL 0,0        ;FORM LOGICAL ADDRESS
06 03730 101300          MOVS 0,0
07 03731 024152          LDA 1,SCRLO
08 03732 107000          ADD 0,1
09 03733 040420          STA 0,ERSS0      ;SAVE DISPLACEMENT FROM SCRLO
10 03734 006133          JSR @LPOCT
11 03735 024237          LDA 1,RELLO
12 03736 124015          COM# 1,1,SNR     ;SKP=WANT TO PRINT RELOCATED ADDRESS
13 03737 000404          JMP .+4
14 03740 020413          LDA 0,ERSS0
15 03741 107000          ADD 0,1          ;RELLO+ENTRY # EXPRESSED AS LOGICAL PAGE
16 03742 006133          JSR @LPOCT
17 03743 010404          ISZ ER.C1        ;NEXT ENTRY
18 03744 014406          DSZ ER.C2        ;DONE ALL?
19 03745 000753          JMP ERMPL        ;NOPE!
20 03746 000414          JMP EREXI
21 03747 000000 ER.C1: 0
22 03750 002564 ER.K4: GETPA
23 03751 000062 ER50.: 50.
24 03752 000000 ER.C2:  0
25 03753 000000 ERSS0:  0
26 03754 000000 ER.S0: 0
27 03755 000000 ER.S1: 0
28 03756 000000 ER.S2: 0
29 03757 000000 ER.S3: 0
30 03760 004033 IWAIT: KEY6W
```

```
!0078 N3MRT
01 03761 001000          1B6
02 03762 020230 EREXI:   LDA 0,SWREG
03 03763 024776          LDA 1,EREXI-1    ;GET 1B6
04 03764 123414          AND# 1,0,SZR     ;SKP=NOT ERROR WAIT
05 03765 004446          JSR KEY6W        ;WAIT FOR TTI INPUT
06 03766 103102          ADDL 0,0,SZC     ;C=0 IS ERROR RELEASE
07 03767 010770          ISZ ER.S3
08 03770 020764 ERXXT:   LDA 0,ER.S0
09 03771 024764          LDA 1,ER.S1
10 03772 030764          LDA 2,ER.S2
11 03773 034764          LDA 3,ER.S3
12 03774 001400          JMP 0,3
13
14
15                       ;2ND OR FOLLOWING CALLS PRINT
16                       ;AC0 1 AND 2 - USED FOR TYPEOUT
17                       ;EXPANSION BY INDIVIDUAL TESTS
18
19 03775 040757 ERROE:   STA 0,ER.S0
20 03776 044757          STA 1,ER.S1
21 03777 050757          STA 2,ER.S2
22 04000 054757          STA 3,ER.S3
23 04001 020230          LDA 0,SWREG
24 04002 101100          MOVL 0,0
25 04003 103102          ADDL 0,0,SZC     ;CHECK BIT 2 TO SEE IF WANT PRINT
26 04004 000756          JMP EREXI
27 04005 006070          JSR @LCRLF
28 04006 024746          LDA 1,ER.S0
29 04007 006133          JSR @LPOCT
30 04010 024745          LDA 1,ER.S1
31 04011 006133          JSR @LPOCT
32 04012 024744          LDA 1,ER.S2
33 04013 006133          JSR @LPOCT
34 04014 000746          JMP EREXI
35
36                       ;TEXT CALL ADRS OF TEXT IS IN AC0
37                       ;CALL MUST ONLY BE MADE WHILE IN FIRST LEVEL TEST
38 04015 040737 ERTXT:   STA 0,ER.S0
39 04016 040411          STA 0,ER.TP
40 04017 044736          STA 1,ER.S1
41 04020 050736          STA 2,ER.S2
42 04021 054736          STA 3,ER.S3
43 04022 020230          LDA 0,SWREG
44 04023 101100          MOVL 0,0
45 04024 103102          ADDL 0,0,SZC     ;SKP IS OK TO TYPE
46 04025 000743          JMP ERXXT        ;EXIT TYPE DELETED
47 04026 006132          JSR @LMESS
48 04027 000000 ER.TP:   0                ;TEXT ADRS STORED HERE
49 04030 000740          JMP ERXXT
50 04031 000000          0        ;TO SAVE AC0
51 04032 000003          3        ;FOR MASK0
```

```
!0079 N3MRT
01 04033 040427 KEY6W:  STA 0,KEY.0
02 04034 054427         STA 3,KEY.3
03 04035 006132         JSR @LMESS
04 04036 007207         KEY6T               ;TYPE WAIT MESS
05 04037 020151 KEYLP:  LDA 0,LASTI         ;GET TTI INPUTTED CHARACTER
06 04040 101112         MOVL# 0,0,SZC       ;WAIT FOR TYPE IN
07 04041 000406         JMP .+6
08 04042 063610         SKPDN TTI
09 04043 000774         JMP .-4
10 04044 060610         DIAC 0,TTI
11 04045 103240         ADDOR 0,0
12 04046 040151         STA 0,LASTI         ;CLR BIT 0
13 04047 034412         LDA 3,TTOXS+1       ;=177 FOR MASK
14 04050 163400         AND 3,0             ;MASK OFF PARITY BIT
15 04051 034407         LDA 3,TTOXS         ;CK FOR SWREG REQUEST
16 04052 116414         SUB# 0,3,SZR
17 04053 000411         JMP KEY.C
18 04054 102400         SUB 0,0
19 04055 040151         STA 0,LASTI
20 04056 020404         LDA 0,KEY.0
21 04057 002404         JMP @KEY.3
22 04060 000015 TTOXS:  15
23 04061 000177         177
24 04062 000000 KEY.0:  0
25 04063 000000 KEY.3:  0
26 04064 006402 KEY.C:  JSR @KY.CK
27 04065 000752         JMP KEYLP
28 04066 001065 KY.CK:  CKODT
```

```
!0080 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 04067 040665 EPADR:  STA 0,ER.S0
11 04070 044665         STA 1,ER.S1
12 04071 050665         STA 2,ER.S2
13 04072 054665         STA 3,ER.S3
14 04073 006132         JSR @LMESS          ;PRINT HEADER
15 04074 006315         TXT.7
16 04075 024123         LDA 1,ST.LA
17 04076 006133         JSR @LPOCT          ;PRINT LOG. START OF LOOP
18 04077 030656         LDA 2,ER.S1
19 04100 146500         SUBL 2,1
20 04101 125220         MOVZR 1,1           ;GET RID OF CRY
21 04102 020654         LDA 0,ER.S2         ;PHYS STRT OF RESIDENT
22 04103 107000         ADD 0,1             ;CREATE PHYS STRT LOOP
23 04104 006133         JSR @LPOCT
24 04105 024647         LDA 1,ER.S0
25 04106 146500         SUBL 2,1            ;CREATE PHYSICAL ERR ADRS
26 04107 125220         MOVZR 1,1
27 04110 020646         LDA 0,ER.S2
28 04111 107000         ADD 0,1
29 04112 006133         JSR @LPOCT
30 04113 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 04114 040421 EPACS:  STA 0,EP.0
36 04115 044421         STA 1,EP.1
37 04116 050421         STA 2,EP.2
38 04117 054421         STA 3,EP.3
39 04120 006132         JSR @LMESS
40 04121 006333         TXT.8
41 04122 024124         LDA 1,RNAC0
42 04123 006133         JSR @LPOCT
43 04124 024125         LDA 1,RNAC1
44 04125 006133         JSR @LPOCT
45 04126 024126         LDA 1,RNAC2
46 04127 006133         JSR @LPOCT
47 04130 020405         LDA 0,EP.0
48 04131 024405         LDA 1,EP.1
49 04132 030405         LDA 2,EP.2
50 04133 034405         LDA 3,EP.3
51 04134 001400         JMP 0,3
52 04135 000000 EP.0:   0
53 04136 000000 EP.1:   0
54 04137 000000 EP.2:   0
55 04140 000000 EP.3:   0
56 04141 003762 EP.RT:  EREXI
```

```
!0081 N3MRT
01                ;*****************************************
02                ;ODT-OCTAL EDITOR
03                ;*****************************************
04 04142 000136 N136:   136
05 04143 000075 N75:    75
06 04144 004366 IMAPS:  MAP.S
07 04145 000007 N7:     7
08 04146 020777 RUBOUT: LDA     0,N7
09 04147 123420         ANDZ    1,0
10 04150 034570         LDA     3,N60
11 04151 163000         ADD     3,0             ;AC0=LAST DIGIT TYPED
12 04152 004575         JSR     SHIFT           ;ECHO AND ERASE THE CHARACTER
13                                              ;BEING RUBBED OUT
14 04153 000433         JMP     OPRIN-1
15
16 04154 044214 ODT:    STA     1,SAV1          ;SAVE THE ACCUMULATORS
17 04155 040213         STA     0,SAV0
18 04156 050215         STA     2,SAV2
19 04157 054216         STA     3,SAV3
20 04160 101200         MOVR    0,0             ;SAVE THE CARRY
21 04161 040217         STA     0,SAVCR
22 04162 102620         SUBZR   0,0
23 04163 101400         INC     0,0
24 04164 040150         STA     0,EACTV         ;PREPARE TTI/TTO FLAG
25 04165 040220         STA     0,OP.EN
26 04166 030221         LDA     2,LOPNL
27 04167 102400 CLRWT:  SUB 0,0
28 04170 040220         STA 0,OP.EN
29 04171 042753         STA     0,@IMAPS                ;CLR MAP SWITCH
30 04172 006476 WAIT:   JSR     @CR.LF          ;TYPE CR,LF
31 04173 020220         LDA 0,OP.EN
32 04174 100014         COM# 0,0,SZR
33 04175 000405         JMP WAITX
34 04176 034570         LDA     3,MAP.S
35 04177 175004         MOV     3,3,SZR         ;SKP IS DON'T USE MAP
36 04200 060302         NIOP    MAP             ;SINGLE REFERENCE
37 04201 025000         LDA     1,0,2           ;GET CONTENTS OF LOC
38 04202 176440 WAITX:  SUBO    3,3             ;AC3=0
39 04203 054561         STA     3,TEMP
40 04204 176000         ADC     3,3             ;AC3     = 177777
41 04205 054555         STA     3,SIGN
42 04206 054555         STA     3,STRAC2
```

```
!0082 N3MRT
01 04207 020151 OPRIN:  LDA     0,LASTI
02 04210 101112         MOVL#   0,0,SZC         ;SKP IS NO INPUT YET
03 04211 000404         JMP     .+4
04 04212 063610         SKPDN   TTI             ;ALSO CHECK DONE
05 04213 000774         JMP     .-4             ;INCASE ION IS OFF
06 04214 060610         DIAC    0,TTI           ;GET CHAR IF DONE = 1
07 04215 034455         LDA     3,N177
08 04216 163400         AND     3,0             ;REMOVE PARITY BIT
09 04217 040151         STA     0,LASTI         ;SAVE CHAR FOR LATER
10 04220 116415         SUB#    0,3,SNR         ;SKIP IF AC0 IS NOT 177
11 04221 000725         JMP     RUBOUT          ;OTHERWISE GO TO RUBOUT
12 04222 034440         LDA     3,N67           ;AC3=67
13 04223 116452         SUBO#   0,3,SZC         ;IF THE ASCII VALUE IS HIGHER THAN
14 04224 000404         JMP     WHERE           ;67 THEN GO TO "WHERE"
15 04225 034573         LDA     3,N57           ;AC3 = 57
16 04226 116442         SUBO    0,3,SZC         ;SKIP IF AC0 IS LESS THAN 57
17 04227 000423         JMP     DIGIT           ;OTHERWISE GO TO "DIGIT"
18 04230 010532 WHERE:  ISZ     SIGN            ;SKIP IF THE PREVIOUS SIGN WAS
19                                              ;A "+"
20 04231 124400         NEG     1,1             ;NEGATE IF IT WAS "-"
21 04232 034532         LDA     3,TEMP
22 04233 167000         ADD     3,1
23 04234 044530         STA     1,TEMP          ;STORE THE NEW VALUE
24 04235 034564         LDA     3,N12
25 04236 116415         SUB#    0,3,SNR         ;DON'T ECHO A LINE FEED
26 04237 000521         JMP     JPNXT
27 04240 034424         LDA     3,X17
28 04241 116415         SUB#    0,3,SNR         ;CONTROL "O" ?
29 04242 000730         JMP     WAIT
30 04243 034426         LDA     3,N15
31 04244 116415         SUB#    0,3,SNR         ;CHECK FOR A "CR"
32 04245 000513         JMP     JPNXT           ;CLOSE THE OPEN LOC.
33 04246 034415         LDA     3,N40
34 04247 116432         SUBZ#   0,3,SZC         ;SKP IF > 40
35 04250 000474         JMP     WHAT            ;"?"
36 04251 000422         JMP     CH.R
37 04252 010511 DIGIT:  ISZ     STRAC2
38 04253 125121         MOVZL   1,1,SKP
39 04254 126440         SUBO    1,1
40 04255 125120         MOVZL   1,1
41 04256 125120         MOVZL   1,1             ;SHIFT AC1 TO LEFT BY 3 PLACES
42 04257 166000         ADC     3,1             ;AC1=OCTAL WORD BEING INPUTED BY
43                                              ;THE OPERATOR
44 04260 004473         JSR     TPCHR           ;ECHO EVERY CHARACTER
45 04261 000726         JMP     OPRIN
```

```
!0083 N3MRT
01 04262 000067 N67:    67
02 04263 000040 N40:    40
03 04264 000017 X17:    17
04 04265 000705 JXWT:   JMP WAIT
05 04266 000124 N124:   124
06 04267 000101 N101:   101
07 04270 004505 CR.LF:  CRLFX
08 04271 000015 N15:    15
09 04272 000177 N177:   177
10 04273 034647 CH.R:   LDA 3,N136
11 04274 116405         SUB     0,3,SNR         ;SKP IF NOT ^
12 04275 000535         JMP     NXTLOC          ;OPEN PREVIOUS LOC
13 04276 004455         JSR     TPCHR
14 04277 034644         LDA     3,N75           ;=
15 04300 116405         SUB     0,3,SNR
16 04301 002525         JMP     @IQUALS         ;PRINT CURRENT ARGUMENT(AC1)
17 04302 034520         LDA     3,N121          ;AC3=121
18 04303 162015         ADC#    3,0,SNR
19 04304 002520         JMP     @IRUN           ;IF IT WAS A "R" THEN START THE
20                                              ;USERS PROGRAM
21 04305 116005         ADC     0,3,SNR
22 04306 002517         JMP     @IPRCD          ;IF IT IS A "P" THEN PROCEED
23 04307 034136         LDA     3,MPSWT
24 04310 175005         MOV     3,3,SNR         ;SKP IS CAN USE MAP
25 04311 000421         JMP     NOMAP
26 04312 034511         LDA     3,N115
27 04313 116415         SUB#    0,3,SNR         ;SKP IS NOT AN "M"
28 04314 000461         JMP     MAPIT
29 04315 034752         LDA     3,N101
30 04316 116405         SUB     0,3,SNR         ;SKP IS NOT AN "A"
31 04317 000460         JMP     MAP.A
32 04320 175405         INC     3,3,SNR         ;SKP IS NOT A "B"
33 04321 000457         JMP     MAP.B
34 04322 034744         LDA     3,N124
35 04323 116415         SUB#    0,3,SNR         ;SKP IS NOT A "T"
36 04324 000443         JMP     MAP.T
37 04325 175415         INC#    3,3,SNR         ;SKP IS NOT "U"
38 04326 000444         JMP     UNMAP
39 04327 034573         LDA     3,N105          ;IS IT AN "E"
40 04330 116415         SUB#    0,3,SNR         ;SKP IS NOPE
41 04331 000476         JMP     MAP.E           ;IT WAS AN "E"
42 04332 034466 NOMAP:  LDA     3,N57           ;AC3 = 57
43 04333 162405         SUB     3,0,SNR         ;SKIP IF AC0 IS NOT "/"
44 04334 000514         JMP     OPNLOC
45 04335 145000         MOV     2,1
46 04336 101405         INC     0,0,SNR         ;SKIP IF IT IS NOT A "."
47 04337 000643         JMP     WAITX
48 04340 126460 N60:    SUBC    1,1             ;AC1=0 AND CARRY IS REESTORED
49                                              ;TO ITS PREVIOUS VALUE
50 04341 115644         INCOR   0,3,SZR         ;SKIP IF IT WAS A "-"
51 04342 161665         INCCR   3,0,SNR         ;SKIP IF IT WAS NOT A "+"
52 04343 000642         JMP     OPRIN-2
53 04344 020554 WHAT:   LDA     0,N77           ;BITS 8-15 OF AC0=77
54 04345 004406         JSR     TPCHR           ;TYPE IT"?"
55 04346 000624         JMP     WAIT
```

```
!0084 N3MRT
01 04347 125220 SHIFT:  MOVZR   1,1
02 04350 125220         MOVZR   1,1
03 04351 125221         MOVZR   1,1,SKP
04 04352 020711         LDA     0,N40           ;PREPARE TO TYPE A SPACE
05 04353 061111 TPCHR:  DOAS    0,TTO           ;TYPE AC0
06 04354 063511         SKPBZ   TTO             ;WAIT FOR THE PRINTER
07 04355 000777         JMP     .-1
08 04356 060211         NIOC    TTO             ;CLR DNE FOR TTO
09 04357 001400         JMP     0,3             ;RETURN TO CALLER
10 04360 000452 JPNXT:  JMP     NXTLOC
11 04361 000000 TPRET:  0
12 04362 000000 SIGN:   0
13 04363 000000 STRAC2: 0
14 04364 000000 TEMP:   0
15 04365 000000 SUBRET: 0
16 04366 000000 MAP.S:  0
17 04367 004562 MAP.T:  JSR MPPRT               ;PRINT MAP ENTRY TABLES
18 04370 002401         JMP @.+1
19 04371 004167         CLRWT
20 04372 176400 UNMAP:  SUB 3,3
21 04373 054773         STA 3,MAP.S             ;SET SWITCH
22 04374 000671         JMP JXWT                ;WAIT FOR OPERATOR INPUT
23 04375 074402 MAPIT:  DIA 3,MAP
24 04376 000411         JMP MAPBB
25 04377 126401 MAP.A:  SUB 1,1,SKP
26 04400 126000 MAP.B:  ADC 1,1
27 04401 074402         DIA 3,2         ;GET CURRENT STATUS
28 04402 020514         LDA 0,K40       ;USER B BIT
29 04403 100000         COM 0,0
30 04404 117400         AND 0,3
31 04405 125004         MOV 1,1,SZR     ;SKP=USER "A"
32 04406 116000         ADC 0,3         ;OR IN BIT 10
33 04407 063477 MAPBB:  SKPBN CPU               ;SKP=ION
34 04410 175041         MOVO 3,3,SKP
35 04411 060277         INTDS
36 04412 075002         DOA 3,MAP
37 04413 036001         LDA 3,@1
38 04414 175003         MOV 3,3,SNC             ;SKP IF INTRUPTS WEREN'T ON
39 04415 060177         INTEN
40 04416 176000         ADC 3,3
41 04417 000754         JMP UNMAP+1
42 04420 000057 N57: 57
43 04421 000012 N12: 12
44 04422 000121 N121:   121
45 04423 000115 N115:   115
46 04424 004523 IRUN:   RUN
47 04425 004525 IPRCD:  PRCD
48 04426 004543 IQUALS: EQUALS
49 04427 176000 MAP.E:  ADC 3,3
50 04430 054465         STA 3,ENT.F
51 04431 000736         JMP MAP.T
```

```
10085 N3MRT
01 04432 010220 NXTLOC: ISZ     OP.EN
02 04433 000405         JMP     NTOPN           ;LOC WASN'T OPENED
03 04434 020732         LDA     0,MAP.S
04 04435 101004         MOV     0,0,SZR         ;SKP IF NOT TO USE MAP
05 04436 060302         NIOP    MAP             ;SINGLE REFERENCE
06 04437 045000         STA     1,0,2           ;RESTORE THE OPEN LOCATION
07 04440 175232 NTOPN:  MOVZR#  3,3,SZC         ;SKIP IF IT WAS NOT A "CR"
08 04441 000624         JMP     JXWT
09 04442 145400         INC     2,1             ;IF IT IS A LINE FEED THEN ADD
10                                              ;1 TO AC2
11 04443 175004         MOV 3,3,SZR             ;SKP WAS A ^
12 04444 000403         JMP .+3
13 04445 166000         ADC 3,1
14 04446 166000         ADC 3,1
15 04447 004436         JSR     CRLFX           ;TYPE A CR, LF
16 04450 152000 OPNLOC: ADC     2,2             ;AC2 = 177777
17 04451 050220         STA     2,OP.EN         ;STORE THE FLAG FOR OPEN LOC.
18 04452 151220         MOVZR   2,2
19 04453 147400         AND     2,1             ;NEGLECT ADDRESS BIT 0
20 04454 044221         STA 1,LOPNL             ;SAVE ADRS LAST OPEN'D
21 04455 004410         JSR     POCTX-1         ;TYPE AC1
22 04456 034710         LDA     3,MAP.S
23 04457 175004         MOV     3,3,SZR         ;SKP IS NOT TO USE MAP
24 04460 060302         NIOP    MAP
25 04461 025000         LDA     1,0,2
26 04462 004404         JSR     POCTX           ;TYPE THE OPEN LOCATION
27 04463 002401         JMP     @.+1
28 04464 004173         WAIT+1                  ;WAIT FOR THE OPERATOR INPUT
29
30 04465 131000         MOV     1,2
31 04466 054677 POCTX:  STA     3,SUBRET        ;SAVE THE RETURN ADDRESS
32 04467 050674         STA     2,STRAC2        ;SAVE AC2
33 04470 131000         MOV     1,2
34 04471 126621         SUBZR   1,1,SKP         ;AC1=100000
35 04472 132401         SUB     1,2,SKP
36 04473 020725         LDA     0,N57           ;AC0=57
37 04474 101400         INC     0,0
38 04475 132453 FRMDGT: SUBO#   1,2,SNC         ;IF AC2 IS LESS THAN AC1 THEN
39                                              ;GO TO PRINT THE DIGIT
40 04476 000774         JMP     .-4
41
42 04477 004650 PRIDGT: JSR     SHIFT
43 04500 125004         MOV     1,1,SZR
44 04501 000772         JMP     FRMDGT-2
45 04502 004650         JSR     TPCHR-1         ;TYPE SPACES
46 04503 030660         LDA     2,STRAC2        ;RESTORE AC2
47 04504 002661         JMP     @SUBRET
```

```
10086 N3MRT
01 04505 054660 CRLFX:  STA     3,SUBRET        ;SAVE THE RETURN ADDRESS
02 04506 020413         LDA     0,X15           ;AC0=15
03 04507 004644         JSR     TPCHR           ;TYPE A "CR"
04 04510 020711         LDA     0,N12           ;AC0=12
05 04511 004642         JSR     TPCHR           ;TYPE A LINE FEED
06 04512 020405         LDA     0,N100          ;BITS 8-15 OF AC0=100
07 04513 004640         JSR     TPCHR           ;TYPE A "@"
08 04514 002651         JMP     @SUBRET
09 04515 000000 ENT.F:  0
10 04516 000040 K40:    40
11 04517 000100 N100:   100
12 04520 000077 N77:    77
13 04521 000015 X15:    15
14 04522 000105 N105:   105
15
16 04523 062677 RUN:    IORST
17 04524 044216         STA     1,SAV3
18 04525 102400 PRCD:   SUB     0,0
19 04526 040150         STA     0,EACTV         ;RESET TTI/TTO FLAG
20 04527 040637         STA     0,MAP.S
21 04530 020213         LDA     0,SAV0
22 04531 024214         LDA     1,SAV1
23 04532 030215         LDA     2,SAV2
24 04533 036407         LDA     3,@EQUALS-1
25 04534 177005         ADD     3,3,SNR         ;SKP=TTY DELETED
26 04535 056405         STA     3,@EQUALS-1     ;CLEAR INTR WAIT
27 04536 034217         LDA     3,SAVCR
28 04537 175100         MOVL    3,3
29 04540 034216         LDA     3,SAV3
30 04541 002216         JMP     @SAV3   ;RETURN TO PGM
31
32
33
34 04542 026755         TT.00+2         ;TTY TEST WAIT SWITCH
35 04543 044405 EQUALS: STA 1,.+5
36 04544 004722         JSR POCTX
37 04545 024403         LDA 1,.+3
38 04546 002401         JMP @.+1
39 04547 004202         WAITX
40 04550 000000         0
```

```
10087 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 04551 054445 MPPRT:  STA 3,MP.S3
06 04552 006132         JSR @LMESS
07 04553 004632         MPTXT
08 04554 010741         ISZ ENT.F       ;SKP="E" COMMAND
09 04555 000407         JMP MP.LP-1
10 04556 014737         DSZ ENT.F       ;RESTORE FLG
11 04557 012436         ISZ @GP.K1      ;NO SKP=ENTERED ADDR.
12 04560 000402         JMP .+2
13 04561 024447         LDA 1,GP.S2     ;SAVED AC2
14 04562 030445         LDA 2,GP.76
15 04563 133401         AND 1,2,SKP     ;AC2=ADDRESS OF PAGE TO PRINT
16 04564 152400         SUB 2,2
17 04565 145000 MP.LP:  MOV 2,1
18 04566 006133         JSR @LPOCT      ;PRINT LOGICAL PAGE #
19 04567 141000         MOV 2,0
20 04570 004427         JSR GPRMP       ;PRINT USER-A ENTRY
21 04571 020433         LDA 0,GP.B6
22 04572 143000         ADD 2,0
23 04573 004424         JSR GPRMP       ;PRINT USER-B ENTRY
24 04574 102620         SUBZR 0,0
25 04575 143000         ADD 2,0
26 04576 004421         JSR GPRMP       ;PRINT DCH-A ENTRY
27 04577 020426         LDA 0,GPB06
28 04600 143000         ADD 2,0
29 04601 004416         JSR GPRMP       ;PRINT DCH-B ENTRY
30 04602 020424         LDA 0,GP.B5
31 04603 113000         ADD 0,2
32 04604 151132         MOVZL# 2,2,SZC  ; NO SKP IS DONE ALL ENTRIES
33 04605 000405         JMP MPRTN
34 04606 050422         STA 2,GP.S2
35 04607 006070         JSR @PCRLF
36 04610 010705         ISZ ENT.F
37 04611 000754         JMP MP.LP       ;CONTINUE WITH NEXT ENTRY
38 04612 102400 MPRTN:  SUB 0,0
39 04613 040702         STA 0,ENT.F
40 04614 002402         JMP @MP.S3
41 04615 004363 GP.K1:  STRAC2
42 04616 000000 MP.S3:  0
43 04617 054412 GPRMP:  STA 3,GP.S3
44 04620 061003         DOA 0,3
45 04621 064403         DIA 1,3         ;GET ENTRY
46 04622 006133         JSR @LPOCT      ;PRINT IT
47 04623 002406         JMP @GP.S3      ;RETURN
48 04624 001000 GP.B6:  1B6
49 04625 101000 GPB06:  1B0+1B6
50 04626 002000 GP.B5:  1B5
51 04627 076000 GP.76:  76000
52 04630 000000 GP.S2:  0
53 04631 000000 GP.S3:  0
54 04632 005215 MPTXT:  .TXTE !<15><12>MAP ENTRY TABLE<15><12>LOGICAL USER-A<11>
55 04653 051525 USER-B<11>DCH-A<11>DCH-B<15><12>!
```

```
10088 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 AC0=BAD CHARACTER.
18                ; (NOT A LEGAL DIGIT OR TERMINATING CHARACTER)
19                ;EXIT AT CALL+2 UPON TERMINATING CHARACTER
20                ; WITH AC0=0, 0, 40, 12, 55
21                ; FOR NULL, TAB, SPACE, CARRIAGE RETURN, COMMA
22                ;THE ABOVE WAIT FOR TTO DONE, THEN CLEAR TTO.
23                ;"CHC?T" PRINTS ASCII CHARACTOR 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 AC0 = 40
28
29 04666 054524 MES?:   STA     3,MES?R         ;PRINT A TEXT MESSAGE
30 04667 044465         STA     1,PAC?1
31 04670 050465         STA     2,PAC?2
32 04671 010521         ISZ     MES?R
33 04672 031400         LDA     2,0,3           ;C(2) POINTS TO MESSAGE
34 04673 024463         LDA     1,P37?7         ;A 8 BIT MASK
35 04674 021000         LDA     0,0,2           ;C(2)=DATA WORD
36 04675 125112         MOVL#   1,1,SZC
37 04676 123701         ANDS    1,0,SKP
38 04677 123401         AND     1,0,SKP         ;C(0)=DATA CHARACTOR RIGHT
39 04700 151400         INC     2,2             ;INC TO NEXT WORD
40 04701 124000         COM     1,1             ;FLIP MASK
41 04702 004455         JSR     CHC?T           ;PRINT
42 04703 000771         JMP     MES?+6          ;ANOTHER
43 04704 000402         JMP     .+2
44 04705 004452 PLS?T:  JSR     CHC?T
45 04706 024446 PEX?T:  LDA     1,PAC?1
46 04707 030446         LDA     2,PAC?2
47 04710 063511         SKPBZ   TTO
48 04711 000777         JMP     .-1
49 04712 060211         NIOC    TTO
50 04713 002477         JMP     @MES?R          ;LAST
```

```
10089 N3MRT
01 04714 102401 ZOC?:   SUB     0,0,SKP
02 04715 020563 POC?:   LDA     0,PC6?0
03 04716 050437         STA     2,PAC?2
04 04717 030565         LDA     2,OCT?AB                ;PRINT C(1) IN OCTAL
05 04720 000404         JMP     .+4
06 04721 050434 PDE?:   STA     2,PAC?2
07 04722 030572         LDA     2,DEC?TB                ;PRINT C(1) IN DECIMAL
08 04723 102400         SUB     0,0
09 04724 054466         STA     3,MES?R         ;BOTH ENTRYS PRINT NUMBER
10 04725 044427         STA     1,PAC?1
11 04726 040425         STA     0,ZSU?P         ;THEN TAB TO NEXT POSITION
12 04727 050401         STA     2,.+1
13 04730 000000 DEC?OC: 0                       ;A"LDA 2,TABLE" INSTRUCTION
14 04731 010777         ISZ     .-1
15 04732 020571         LDA     0,PTB?
16 04733 151005         MOV     2,2,SNR         ;IF TABLE ENTRY=0
17 04734 000751         JMP     PLS?T           ;EXIT WITH A SPACE/TAB
18 04735 034416         LDA     3,ZSU?P         ;ZEROS SUPPRESS STUF
19 04736 102400         SUB     0,0
20 04737 146452 DEO?T:  SUBO#   2,1,SZC
21 04740 000405         JMP     DEC?P
22 04741 146400         SUB     2,1             ;FORM THE DIGIT
23 04742 034536         LDA     3,PC6?0
24 04743 101400         INC     0,0
25 04744 000773         JMP     DEO?T
26 04745 151235 DEC?P:  MOVZR#  2,2,SNR
27 04746 034532         LDA     3,PC6?0
28 04747 054404         STA     3,ZSU?P         ;C(0)=DIGIT
29 04750 163004         ADD     3,0,SZR         ;MAKE ASCII
30 04751 004406         JSR     CHC?T           ;PRINT
31 04752 000756         JMP     DEC?OC          ;GET NEXT DIGIT
32
33 04753 000000 ZSU?P:  0
34 04754 000000 PAC?1:  0
35 04755 000000 PAC?2:  0
36 04756 000377 P377:   377
37 04757 054432 CHC?T:  STA     3,CHR?T         ;PRINT C(0) RIGHT
38 04760 101315         MOVS#   0,0,SNR         ;RETURN +2 IF NULL
39 04761 001401         JMP     1,3
40 04762 034426         LDA     3,PC1?1         ;AC3 = 11
41 04763 116415         SUB#    0,3,SNR         ;SKIP IF A TAB IS NOT TO
42                                              ;BE SIMULATED
43 04764 000403         JMP     CHA?3
44 04765 004540         JSR     TYP?            ;PRINT IT
45 04766 002423         JMP     @CHR?T          ;EXIT
46 04767 004535 CHA?3:  JSR     TPS?            ;PRINT A SPACE
47 04770 020417         LDA     0,CHR?Z
48 04771 034415         LDA     3,PC7?          ;AC3 = 7
49 04772 163404         AND     3,0,SZR
50 04773 000774         JMP     CHA?3           ;SIMULATE A TABE WITH 1
51                                              ;TO 7 SPACES
52 04774 040413         STA     0,CHR?Z
53 04775 002414         JMP     @CHR?T
54
```

```
10090 N3MRT
01 04776 054414 CLF?:   STA     3,MES?R         ;SAVE RETURN
02 04777 044755         STA     1,PAC?1
03 05000 050755         STA     2,PAC?2
04 05001 020404         LDA     0,XJ15
05 05002 004755         JSR     CHC?T           ;PRINT CARRIAGE AND LF
06 05003 020515         LDA     0,PC1?2
07 05004 000701         JMP     PLS?T
08
09 05005 000015 XJ15:   15
10 05006 000007 PC7?:   7
11 05007 000000 CHR?Z:  0
12 05010 000011 PC1?1:  11
13 05011 000000 CHR?T:  0
14 05012 000000 MES?R:  0
15 05013 020505 TIC?:   LDA     0,PC1?2
16 05014 004511         JSR     TYP?
17 05015 010775 TIX?:   ISZ     MES?R
18 05016 024736 TIR?:   LDA     1,PAC?1
19 05017 034734         LDA     3,ZSU?P
20 05020 175102         MOVL    3,3,SZC
21 05021 124400         NEG     1,1
22 05022 000665         JMP     PEX?T+1
23
24 05023 102121 TIO?:   ADCZL   0,0,SKP         ;OCTAL ENTRY
25 05024 102440 TID?:   SUBO    0,0             ;DECIMAL ENTRY
26 05025 054765         STA     3,MES?R
27 05026 050727         STA     2,PAC?2         ;AC2 IS SAVED
28 05027 030471         LDA     2,PC1?2
29 05030 113000         ADD     0,2
30 05031 102440         SUBO    0,0
31 05032 040721         STA     0,ZSU?P         ;MINUS SIGN AND LEADING SPACES
32                                              ;FLAG
33 05033 034720 TIS?:   LDA     3,ZSU?P
34 05034 175004         MOV     3,3,SZR
35 05035 000760         JMP     TIX?
36 05036 054716 TIW?:   STA     3,PAC?1
37 05037 063610         SKPDN   TTI
38 05040 000777         JMP     .-1
39 05041 060610         DIAC    0,TTI
40 05042 004715         JSR     CHC?T
41 05043 034436         LDA     3,PC4?0
42 05044 116414         SUB#    0,3,SZR
43 05045 101015         MOV#    0,0,SNR
44 05046 000765         JMP     TIS?            ;SPACE, TAB, OR NULL
45 05047 024434         LDA     1,TIN?2
46 05050 106015         ADC#    0,1,SNR         ;COMMA
47 05051 000744         JMP     TIX?            ;OR
48 05052 106424         SUBZ    0,1,SZR         ;MINUS
49 05053 000405         JMP     TIM?            ;IF NEITHER THEN GO TO TIM?
50 05054 034677         LDA     3,ZSU?P
51 05055 177200         ADDR    3,3             ;COMPLEMENT SIGN
52 05056 054675         STA     3,ZSU?P
53 05057 000760         JMP     TIW?+1
54 05060 136415 TIM?:   SUB#    1,3,SNR         ;IS IT A CARRIAGE RETURN?
55 05061 000732         JMP     TIC?            ;IF CR THEN GO TO TIC?
```

```
10091 N3MRT
01 05062 024420 TIN?:   LDA     1,TIN?1
02 05063 107022         ADDZ    0,1,SZC         ;SKIP IF NOT A DIGIT
03 05064 146513         SUBL#   2,1,SNC         ;SKIP IF DIGIT
04 05065 000731         JMP     TIR?
05 05066 010665         ISZ     ZSU?P           ;OUT OF LEADING SPACES
06 05067 020665         LDA     0,PAC?1
07 05070 101120         MOVZL   0,0
08 05071 115120         MOVZL   0,3
09 05072 175120         MOVZL   3,3
10 05073 137000         ADD     1,3             ;8 OLD PAC?1'S + NEW DIGIT
11 05074 145220         MOVZR   2,1
12 05075 125232         MOVZR#  1,1,SZC         ;SKIP IF OCTAL MODE
13 05076 117000         ADD     0,3             ;ADD 2 OLD PAC?1'S
14 05077 000737         JMP     TIW?
15 05100 000060 PC6?0:  60
16 05101 000040 PC4?0:  40
17 05102 177720 TIN?1:  -60
18 05103 000055 TIN?2:  55
19 05104 030555 OCT?AB: LDA     2,.+1+.-DEC?OC
20 05105 100000         100000
21 05106 010000         10000
22 05107 001000         1000
23 05110 000100 C10?0:  100
24 05111 000010         10
25 05112 000001         1
26 05113 000000         0
27 05114 030565 DEC?TB: LDA     2,.+1+.-DEC?OC
28       000012 .RDX 10
29 05115 023420         10000
30 05116 001750         1000
31 05117 000144         100
32 05120 000012 PC1?2:  10
33 05121 000001         1
34 05122 000000         0
35       000010 .RDX    8
36
37 05123 000011 PTB?:   11
38
```

```
10092 N3MRT
01 05124 020755 TPS?:   LDA     0,PC4?0         ;PREPARE TO PRINT A SPACE
02 05125 054461 TYP?:   STA     3,TYP?R         ;TYPE THE RIGHT BYTE OF AC0
03 05126 034456         LDA     3,INT?          ;IF IT IS HERE DUE TO SWPAK
04 05127 175004         MOV     3,3,SZR         ;TTO OUTPUTS WILL BE ENABLED.
05 05130 034230         LDA     3,SWREG         ;READ THE SWITCHES
06 05131 040456         STA     0,P.MS0         ;SAVE AC0
07 05132 020457         LDA     0,P.MB5
08 05133 163404         AND     3,0,SZR         ;BIT 5 OF SWREG SET?
09 05134 020454         LDA     0,P.MB1
10 05135 040150         STA     0,EACTV         ;SET UP EACTV FOR LPT
11 05136 177100         ADDL    3,3
12 05137 175112         MOVL#   3,3,SZC         ;SKIP IF TYPEOUTS ARE NOT
13 05140 000416         JMP     PLP?T           ;SUPPRESSED
14 05141 176620         SUBZR   3,3             ;BIT 0=1
15 05142 117000         ADD     0,3
16 05143 020444         LDA     0,P.MS0         ;GET SAVED CHARACTER
17 05144 063511 TPT?Y:  SKPBZ   TTO
18 05145 000777         JMP     .-1
19 05146 000401         JMP     .+1
20 05147 063511         SKPBZ   TTO
21 05150 000774         JMP     .-4
22 05151 054150         STA 3,EACTV             ;SET UP EACTV
23 05152 061111         DOAS    0,TTO           ;OUTPUT TO TTO
24 05153 063511         SKPBZ   TTO
25 05154 000777         JMP     .-1
26 05155 060211         NIOC    TTO
27 05156 020431 PLP?T:  LDA     0,P.MS0         ;GET CHARACTER
28 05157 034424         LDA     3,P17?7
29 05160 163400         AND     3,0             ;REMOVE PARITY  BIT FOR LPT
30 05161 034150         LDA     3,EACTV
31 05162 177123         ADDZL   3,3,SNC         ;SKP IS WANT LPT OUTPUT
32 05163 000405         JMP     TPR?T           ;FORGET LPT
33 05164 061117         DOAS    0,LPT           ;OUTPUT THE CHARACTER TO LPT
34 05165 063517         SKPBZ   LPT             ;WAIT FOR LPT
35 05166 000777         JMP     .-1
36 05167 060217         NIOC    LPT
37 05170 176400 TPR?T:  SUB     3,3
38 05171 054150         STA     3,EACTV         ;CLEAR EACTV SWITCH
39 05172 034411         LDA     3,P17?7
40 05173 116043         ADCO    0,3,SNC         ;SKIP IF IT WAS A RUBOUT
41 05174 034705         LDA     3,PC4?0         ;AC3 = 40
42 05175 162432         SUBZ#   3,0,SZC         ;SKIP FOR NON PRINTING
43 05176 010611         ISZ     CHR?Z           ;CHARACTERS
44 05177 034406         LDA     3,PC1?5         ;AC3 = 15
45 05200 116445         SUBO    0,3,SNR         ;SKIP IF IT WAS NOT A "CR"
46 05201 054606         STA     3,CHR?Z         ;CLEAR HORZ POS
47 05202 002404         JMP     @TYP?R
48 05203 000177 P17?7:  177
49 05204 177777 INT?:   -1
50 05205 000015 PC1?5:  15
51 05206 000000 TYP?R:  0
52 05207 000000 P.MS0:  0
53 05210 040000 P.MB1:  181
54 05211 002000 P.MB5:  185
```

```
!0093 N3MRT
01
02                ;THIS PACKAGE IS USED TO CHANGE THE SETTINGS OF LOCATION
03                ;"SWREG" OF PAGE 0. THE PROGRAM CONTROL SHOULD ENTER "INP?"
04                ;WITH AC3 HAVING THE RETURN ADDRESS. THE CHARACTER INPUTED
05                ;BY THE OPERATOR IS ECHOED AFTER A "CR". IF THE COMMAND IS
06                ;NOT A LEGAL ONE THEN THE CONTROL IS RETURNED WITHOUT DOING
07                ;ANY THING, OTHERWISE ONE OF THE FOLLOWING COMMANDS IS
08                ;EXECUTED:
09                ;KEYS 1-9 AND A-F ARE USED TO COMPLEMENT THE CURRENT VALUE
10                ;OF BITS 1-15 OF "SWREG". IF ONE OF THESE KEYS IS HIT THE
11                ;CORRESPONDING BIT OF "SWREG" IS COMPLEMENTED AND THE CONTROL
12                ;IS RETURNED TO THE STATE PROGRAM HAD BEFORE HITTING THE KEY
13                ;TYPING OF A "0" WILL LOCK THE PROGRAM IN A SWITCH MODIFICATION
14                ;MODE IN WHICH CASE MORE THAN ONE BITS CAN BE CHANGED BEFORE
15                ;THE CONTROL IS ALLOWED TO RETURN TO THE MAIN PROGRAM. HITTING
16                ;THE "CR" KEY WILL UNLOCK THE PROGRAM FROM THIS MODE.
17                ;"^D" THIS COMMAND GIVEN AT ANY TIME WILL RESET THE "SWREG"
18                ;TO DEFAULT MODE (ALL ZEROS) AND RESTART THE PROGRAM AT ADD.
19                ;STORED IN LOCATION "INS?"
20                ;"^R" THIS COMMANND GIVEN AT ANY TIME  WILL RESTART THE PROG.
21                ;AT ADDRESS STORED IN LOCATION "INS?"
22                ;"M" THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE  CURRENT
23                ;OPERATING MODES.
24                ;BEFORE THE CONTROL IS RETURNED TO THE MAIN PROGRAM BIT 0 WILL
25                ;BE SET IF ANY OF THE OTHER BITS OF "SWREG" IS SET, OTHERWISE
26                ;IT WILL BE CLEARED
27
28                ;THIS PACKAGE EXITS WITH C(AC3) = CHARACTER TYPED IN (PARITY
29                ;STRIPPED).
30
31 05212 000136 IN1?36: 136
32 05213 000104 IN1?04: 104
33 05214 000122 IN1?22: 122
34 05215 000033 INC?33: 33
35 05216 000000 INL?K:  0
36 05217 000202 INS?:   STRT2
37 05220 000000 SVP?TB: 0
38 05221 005123 IPTB?:  PTB?
39 05222 000115 IN115:  115
```

```
!0094 N3MRT
01 05223 054454 INP?:   STA     3,INR?T         ;SAVE THE RETURN ADDRESS
02 05224 040454         STA     0,INS?0         ;SAVE AC0
03 05225 044454         STA     1,INS?1         ;AC1
04 05226 050454         STA     2,INS?2         ;AND AC2
05 05227 175200         MOVR    3,3             ;
06 05230 054454         STA     3,INS?C         ;SAVE CARRY
07 05231 176400         SUB     3,3
08 05232 054752         STA     3,INT?
09 05233 040763         STA     0,INL?K         ;"INL?K" IS NOT -1
10 05234 020151 IN0?:   LDA     0,LASTI         ;READ THE INPUT
11 05235 034746         LDA     3,P177?7        ;AC3 = 177
12 05236 163400         AND     3,0             ;GET RID OF THE PARITY BIT
13 05237 024746         LDA     1,PC1?5         ;AC1 = 15
14 05240 106415         SUB#    0,1,SNR         ;SKIP IF THE CHARACTER TYPED
15                                              ;WAS NOT "CR"
16 05241 000552         JMP     INR?
17 05242 040441         STA     0,INS?3         ;SAVE CHARACTER
18 05243 024645         LDA     1,C10?0         ;AC1 = 100
19 05244 034654         LDA     3,PC1?2         ;AC3 = 12
20 05245 116414         SUB#    0,3,SZR         ;SKIP IF IT IS A LINE FEED
21 05246 034747         LDA     3,INC?33        ;AC3 = 33
22 05247 162452         SUBO#   3,0,SZC         ;SKIP IF AC0 IS EQUAL OR MORE
23                                              ;THAN AC3
24 05250 000522         JMP     IN5?
25 05251 004654 IN1?:   JSR     TYP?            ;ECHO THE CHARACTER
26 05252 034750         LDA     3,IN115
27 05253 116405         SUB     0,3,SNR         ;CK FOR "M"
28 05254 000431         JMP     INM?
29 05255 034623         LDA     3,PC6?0         ;AC3 = 60
30 05256 152620         SUBZR   2,2             ;AC2 = 100000
31 05257 116405         SUB     0,3,SNR         ;SKIP IF THE DIGIT TYPED WAS
32                                              ;NOT 0
33 05260 000457         JMP     IN3?
34 05261 010735         ISZ     INL?K   ;IN SWREG LOOP?
35 05262 000465         JMP     IN4?+1  ;NOPE-DON'T SET SWREG BITS
36 05263 014733         DSZ     INL?K   ;RESTORE LOOP FLG
37
38 05264 151221 IN2?:   MOVZR   2,2,SKP         ;SHIFT AC2 TO RIGHT
39 05265 126520         SUBZL   1,1             ;AC1 = 1
40 05266 175405         INC     3,3,SNR
41 05267 000452         JMP     IN3?+2
42 05270 147415         AND#    2,1,SNR         ;STAY IN LOOP UNTIL ALL BITS
43                                              ;OF SWREG ARE CHECKED
44 05271 000773         JMP     IN2?
45 05272 106400         SUB     0,1             ;WHEN THE CONTROL COMES HERE
46                                              ;FOR THE FIRST TIME AC1 = 100
47 05273 135000         MOV     1,3
48 05274 151225         MOVZR   2,2,SNR
49 05275 000451         JMP     IN4?
50 05276 000767         JMP     IN2?+1
51 05277 000000 INR?T:  0
52 05300 000000 INS?0:  0
53 05301 000000 INS?1:  0
54 05302 000000 INS?2:  0
55 05303 000000 INS?3:  0
56 05304 000000 INS?C:  0
```

```
!0095 N3MRT
01 05305 126400 INM?:   SUB     1,1
02 05306 036713         LDA     3,@IPTB?
03 05307 054711         STA     3,SVP?TB        ;SAVE PTR?
04 05310 034461         LDA     3,PC.40         ;PREPARE TO PRINT A SPACE
05                                              ;AFTER EACH NUMBER
06 05311 056710         STA     3,@IPTB?
07 05312 006127         JSR     @ICLF?          ;TYPE A "CR" AND "LF"
08 05313 006130         JSR     @IPDE?          ;PRINT THE CONTENTS OF AC1
09 05314 034604         LDA     3,PC1?2         ;AC3 = 12
10 05315 125400         INC     1,1
11 05316 166452         SUBO#   3,1,SZC         ;SKIP IF AC1 IS GREATER OR EQUAL
12                                              ;TO AC3
13 05317 006450         JSR     @ITPS?          ;TYPE A SPACE
14 05320 101220         MOVZR   0,0             ;AC0 = 20
15 05321 122414         SUB#    1,0,SZR         ;SKIP AFTER TYPING # 15
16 05322 000771         JMP     INM?+6
17 05323 006127         JSR     @ICLF?
18 05324 030230         LDA     2,SWREG
19 05325 151140         MOVOL   2,2             ;AC2 HAD SWITCH SETTINGS
20 05326 126560         SUBCL   1,1             ;BRING THE CARRY BIT IN AC1
21 05327 006131         JSR     @IZOC?          ;TYPE THE CONTENTS OF AC1
22 05330 006437         JSR     @ITPS?          ;TYPE A SPACE
23 05331 151124         MOVZL   2,2,SZR         ;SKIP AFTER TYPING ALL THE 16
24                                              ;BITS
25 05332 000774         JMP     .-4
26 05333 006127         JSR     @ICLF?
27 05334 034664         LDA     3,SVP?TB
28 05335 056664         STA     3,@IPTB?        ;RESTORE PTB?
29 05336 000410         JMP     IN4?
30
31 05337 176000 IN3?:   ADC     3,3             ;AC3 = -1
32 05340 054656         STA     3,INL?K         ;LOCK IN SWITCH INPUT MODE
33 05341 024230         LDA     1,SWREG         ;READ THE CURRRENT VALUE OF
34                                              ;"SWREG"
35 05342 133414         AND#    1,2,SZR         ;TAKE XOR OF AC1 AND AC2
36 05343 146401         SUB     2,1,SKP
37 05344 147000         ADD     2,1
38 05345 044230         STA     1,SWREG         ;SAVE THE NEW VALUE OF "SWREG"
39 05346 010650 IN4?:   ISZ     INL?K           ;SKIP IF THE PROGRAM IS LOCKED
40                                              ;IN SWITCH INPUT MODE
41 05347 000444         JMP     INR?
42 05350 014646         DSZ     INL?K           ;NEVER SKIP
43 05351 102520         SUBZL   0,0
44 05352 024150         LDA     1,EACTV         ;SAVE EACTV
45 05353 040150         STA     0,EACTV         ;SET TTI ACTIVE BIT
46 05354 020151         LDA     0,LASTI         ;WHILE TTI ACTIVE IS SET
47 05355 101102         MOVL    0,0,SZC ;TTI INTR HANDLR WON'T SET SWREG BITS
48 05356 000405         JMP     .+5
49 05357 063610         SKPDN   TTI             ;WAIT FOR OPERATOR INPUT
50 05360 000774         JMP     .-4
51 05361 060610         DIAC    0,TTI
52 05362 000402         JMP     .+2
53 05363 101220         MOVZR   0,0
54 05364 040151         STA     0,LASTI
55 05365 044150         STA     1,EACTV         ;RESTORE EACTV
56 05366 000647         JMP     IN0?+1
57 05367 005124 ITPS?:  TPS?
58 05370 005125 ITYP?:  TYP?
59 05371 000040 PC.40:  40
```

```
!0096 N3MRT
01 05372 107000 IN5?:   ADD     0,1             ;AC1 = 100+ ASCII VALUE OF
02                                              ;CONTROL CHARACTER
03 05373 020617         LDA     0,IN1?36        ;AC0 = 136
04 05374 006774         JSR     @ITYP?          ;TYPE ^
05 05375 121000         MOV     1,0
06 05376 006772         JSR     @ITYP?          ;ECHO CHARACTER
07 05377 034433         LDA     3,IN?17
08 05400 116405         SUB     0,3,SNR         ;CK FOR CNTL "0"
09 05401 006427         JSR     @INODT
10 05402 034612         LDA     3,IN1?22        ;AC3 = 122
11 05403 116405         SUB     0,3,SNR         ;SKIP IF IT IS NOT ^R
12 05404 000405         JMP     IN6?
13 05405 034606         LDA     3,IN1?04        ;AC3 = 104
14 05406 116404         SUB     0,3,SZR         ;SKIP IF IT WAS A ^D
15 05407 000737         JMP     IN4?
16 05410 054230         STA     3,SWREG         ;SET SWITCHHES TO DEFAULT
17                                              ;MODE
18 05411 034606 IN6?:   LDA     3,INS?          ;AC3 = ADDRESS OF THE LOCATION
19                                              ;WHERE THE PROGRAM WILL START
20 05412 054665         STA     3,INR?T
21 05413 012416 INR?:   ISZ     @IINT?
22 05414 030230         LDA     2,SWREG
23 05415 176220         ADCZR   3,3             ;AC3 = 77777
24 05416 173404         AND     3,2,SZR         ;SKIP IF THE SWITCHES ARE SET
25                                              ;TO ALL ZERO'S
26 05417 172000         ADC     3,2
27 05420 050230         STA     2,SWREG
28 05421 020663         LDA     0,INS?C         ;RESTORE CARRY
29 05422 101100         MOVL    0,0             ;
30 05423 020655         LDA     0,INS?0
31 05424 024655         LDA     1,INS?1
32 05425 030655         LDA     2,INS?2         ;RESTORE THE ACCUMULATORS
33 05426 034655         LDA     3,INS?3         ;RETURN WITH C(AC3)=CHAR INPUT
34 05427 002650         JMP     @INR?T          ;RETURN
35 05430 004154 INODT:  ODT
36 05431 005204 IINT?:  INT?
37 05432 000117 IN?17:  117
```

```
10097 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 05433 063777 LINTR:  SKPDZ CPU         ;POWER FAIL?
13 05434 002134          JMP @PFAIL
14 05435 075401          PSH 3            ;SAVE AC3
15 05436 071401          PSH 2            ;AC2
16 05437 065401          PSH 1            ;AC1
17 05440 061401          PSH 0            ;AC0
18 05441 020000          LDA 0,0
19 05442 040465          STA 0,LOC.0
20 05443 101102          MOVL 0,0,SZC     ;CARRY
21 05444 063077          HALT             ;BIT 0 OF 0=1?
22 05445 061401          PSH 0            ;PC*2+CRY IN BIT 15
23 05446 020460          LDA 0,ILLSC      ;SET UP ILLEGAL SUPER.
24 05447 040000          STA 0,0          ;CALL IN LOC 0.
25 05450 020454          LDA 0,MSKRG
26 05451 061401          PSH 0            ;SAVE OLD MSKO
27 05452 020136          LDA 0,MPSWT
28 05453 101005          MOV 0,0,SNR      ;SKIP IS USE MAP
29 05454 000406          JMP .+6          ;IGNORE MAP
30 05455 060402          DIA 0,2          ;GET STATUS
31 05456 061401          PSH 0            ;PUT ON STACK
32 05457 101120          MOVZL 0,0        ;TURN BIT 0 OFF
33 05460 101220          MOVZR 0,0        ;
34 05461 061002          DOA 0,2          ;WRITE IT TO STATUS
35 05462 022001          LDA 0,@1         ;DEFER TO LOAD STATUS REG.
36 05463 030535          LDA 2,LK300
37 05464 034437          LDA 3,LMSKS
38 05465 061477          INTA 0
39 05466 113000          ADD 0,2
40 05467 117000          ADD 0,3
41 05470 025400          LDA 1,0,3
42 05471 035000          LDA 3,0,2
43 05472 044432          STA 1,MSKRG
44 05473 030432          LDA 2,K4
45 05474 112415          SUB# 0,2,SNR     ;WAS IT A DEV #4 INTR?
46 05475 001400          JMP 0,3          ;YEP, SERVICE IT
47 05476 066077          MSKO 1
48 05477 060177          INTEN
49 05500 005400          JSR 0,3          ;GO TO INTR SERV
```

```
10098 N3MRT
01                ;LINTD - LINKER INTERRUPT DISMISS
02                ;RETURN ADDRESS WAS GIVEN TO DEVICE
03                ;SERVICE ROUTINES VIA THE JSR 03
04
05 05501 060277 LINTD:  INTDS
06 05502 020136          LDA 0,MPSWT
07 05503 101005          MOV 0,0,SNR      ;SKIP IS USE MAP
08 05504 000403          JMP .+3
09 05505 061601          POP 0            ;RETRIEVE STATUS OFF STACK
10 05506 061002          DOA 0,2          ;WRITE IT
11 05507 065601          POP 1    ;RETRIEVE MASK
12 05510 044414          STA 1,MSKRG
13 05511 066077          MSKO 1
14 05512 061601          POP 0    ;GET CRY+2*PC
15 05513 101220          MOVZR 0,0        ;RESTORE CRY
16 05514 040413          STA 0,LOC.0
17 05515 061601          POP 0
18 05516 065601          POP 1
19 05517 071601          POP 2
20 05520 075601          POP 3
21 05521 060177          INTEN
22 05522 002405          JMP @LOC.0       ;GETS TO LOGICAL 0
23 05523 000000 LMSKS: 0
24 05524 000000 MSKRG: 0
25 05525 000004 K4:  4
26 05526 177770 ILLSC:  177770
27 05527 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 05530 054410 EINTP:  STA 3,EI.S3
34 05531 034467          LDA 3,LK300
35 05532 117000          ADD 0,3
36 05533 051400          STA 2,0,3
37 05534 034767          LDA 3,LMSKS
38 05535 117000          ADD 0,3
39 05536 045400          STA 1,0,3
40 05537 002401          JMP @EI.S3
41
42 05540 000000 EI.S3:  0
```

```
!0099 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 05541 020470 LCINT:  LDA 0,LILLI
06 05542 030456         LDA 2,LK300
07 05543 041000         STA 0,0,2        ;FILL SERVICE
08 05544 151400         INC 2,2          ;VECTORS WITH
09 05545 145300         MOVS 2,1         ;AORS ILLEGAL INTR
10 05546 125224         MOVZR 1,1,SZR
11 05547 000774         JMP LCINT+2
12 05550 071001         MTSP 2  ;STACK POINTER TO 400
13 05551 070001         MTFP 2  ;FRAME POINTER TO 400
14 05552 052450         STA 2,@LC.K1+1
15 05553 145220         MOVZR 2,1
16 05554 131220         MOVZR 1,2        ;AC2=100,AC1=200
17 05555 022444         LDA 0,@LC.K1     ;RESERVE 100 WORDS
18 05556 113000         ADD 0,2          ;ABOVE MEM ALLOC. TBLS
19 05557 052442         STA 2,@LC.K1     ;100 FOR MSKO
20 05560 125220         MOVZR 1,1        ;AC1=100
21 05561 102000         ADC 0,0
22 05562 041000         STA 0,0,2
23 05563 113000         ADD 0,2
24 05564 107004         ADD 0,1,SZR
25 05565 000775         JMP .-3
26 05566 151400         INC 2,2
27 05567 050734         STA 2,LMSKS
28 05570 020427         LDA 0,LC.K2
29 05571 040001         STA 0,1
30 05572 102400         SUB 0,0
31 05573 040731         STA 0,MSKRG      ;STRT INT MSK =0
32 05574 042427         STA 0,@LC.K1+2
33 05575 020427         LDA 0,LC.K1+3
34 05576 040003         STA 0,3          ;INIT FOR STACK INTA'S
35 05577 024117         LDA 1,PRFLG      ;PARITY EXIST?
36 05600 125005         MOV 1.1,SNR      ;SKP IS HAVE PARITY
37 05601 000405         JMP .+5          ;NO PARITY
38 05602 020423         LDA 0,LC.K7      ;ADDR OF PARITY HANDLER
39 05603 040304         STA 0,300+PRTY   ;PLACE IN TABLE
40 05604 060204         NIOC PRTY        ;CLR ANY PREVIOUS ERRS
41 05605 060104         NIOS PRTY        ;INIT PARITY INTR
```

```
!0100 N3MRT
01 05606 020420         LDA 0,LC.K8      ;ADDR OF TTI INTR HANDLER
02 05607 040310         STA 0,300+TTI    ;PUT IN INTR TBL
03 05610 020417         LDA 0,LC.K9      ;ADDR OF TTO INTR HANDLER
04 05611 040311         STA 0,300+TTO    ;PUT IN INTR TBL
05 05612 020134         LDA 0,PFAIL      ;PWR FAIL RTN ADDR
06 05613 040300         STA 0,300        ;PUT IN INTR TBL
07 05614 020414         LDA 0,LC.K6
08 05615 040317         STA 0,300+LPT    ;SET UP FOR LPT INTA
09 05616 001400         JMP 0,3
10 05617 005433 LC.K2: LINTR
11 05620 000300 LK300:  300
12 05621 001443 LC.K1:  LSETB
13 05622 002357         STADR
14 05623 002361         STKPG
15 05624 002401         STKIN
16 05625 005641 LC.K7:  PRTHL    ;PARITY INTR HANDLER ADDRESS
17 05626 027021 LC.K8: TT.TI
18 05627 027117 LC.K9: TT.TO
19 05630 005652 LC.K6:  LPT.I
```

```
10101 N3MRT
01 05631 005632 LILLI:  LILLI+1
02 05632 040144         STA 0,UDEVI
03 05633 024402         LDA 1,.+2
04 05634 123001         ADD 1,0,SKP
05 05635 060200         NIOC 0
06 05636 040401         STA 0,.+1
07 05637 060200         NIOC 0          ;CHANGED TO DEV#
08 05640 001400         JMP 0,3
09
10              ;PARITY INTERRUPT HANDLER SUBROUTINE
11 05641 022410 PRTHL:  LDA 0,@PR.R0    ;GET SAVED LOC 0
12 05642 064404         DIA 1,PRTY      ;GET MADR BITS
13 05643 071404         DIB 2,PRTY      ;GET XMADR BITS
14 05644 063077         HALT            ;PARITY ERROR OCCURRED
15 05645 060204         NIOC PRTY       ;CLEAR ERROR
16 05646 060104         NIOS PRTY       ;REENABLE INTR
17 05647 002401         JMP @PR.RT      ;RETURN TO INTERRUPT DISMISS
18 05650 005501 PR.RT:  LINTD
19 05651 005527 PR.R0:  LOC.0
20
21              ;LPT.I- LPT INTA HANDLER IF NO LPT TEST
22 05652 060217 LPT.I:  NIOC LPT
23 05653 001400         JMP 0,3 ;RETURN
```

```
10102 N3MRT
01              ;IOVAL
02              ;I/O VALIDITY TRAP HANDLER
03              ;DETERMINE LEGALITY OF TRAP ERROR OR IF NOT EXPECTED
04 05654 022501 IOVAL:  LDA 0,@ICD.0
05 05655 040567         STA 0,IOV.0
06 05656 026500         LDA 1,@ICD.1
07 05657 044566         STA 1,IOV.1
08 05660 032477         LDA 2,@ICD.2
09 05661 050565         STA 2,IOV.2
10 05662 036476         LDA 3,@ICD.3
11 05663 054560         STA 3,IOV.3
12 05664 102560         SUBCL 0,0
13 05665 040562         STA 0,IOV.4
14 05666 020233         LDA 0,VLPCR
15 05667 024234         LDA 1,VLDTA             ;VIOL. DATA
16 05670 030143         LDA 2,STATS             ;STATUS WORD
17 05671 060177         INTEN
18 05672 040556         STA 0,IOV.A
19 05673 044556         STA 1,IOV.B
20 05674 050556         STA 2,IOV.C
21 05675 036560         LDA 3,@IOV.7    ;WAS THIS A ODT VIOLATION?
22 05676 175005         MOV 3,3,SNR     ;SKIP=ODT VIOL.
23 05677 000410         JMP NTODT       ;NOT A ODT VIOL.
24 05700 101400 ETODT:  INC 0,0         ;PC + 1
25 05701 040553         STA 0,IVODT
26 05702 020542         LDA 0,IOV.0
27 05703 024542         LDA 1,IOV.1
28 05704 030542         LDA 2,IOV.2
29 05705 034536         LDA 3,IOV.3     ;RESTORE AC'S
30 05706 002546         JMP @IVODT      ;RETURN TO ODT
31 05707 034140 NTODT:  LDA 3,PSTRT
32 05710 035406         LDA 3,6,3
33 05711 175122         MOVZL 3,3,SZC   ;BIT 0=1 IS EXPECTED TRP
34 05712 000521         JMP IOV.E       ;EXPECTED TRAP RETURN
35 05713 175220         MOVZR 3,3
36 05714 054537         STA 3,IOV.R
37 05715 034234         LDA 3,VLDTA     ;VIOL. DATA REG.
38 05716 020443         LDA 0,IOV40     ;=40
39 05717 117404         AND 0,3,SZR     ;SKP IS NOT VALIDITY VIOL.
40 05720 000404         JMP .+4
41 05721 006132         JSR @LMESS      ;MUST BE I/O PROTECT ERR
42 05722 006076         IO.TX
43 05723 000403         JMP IOVPR
44 05724 006132         JSR @LMESS      ;VALIDITY PROTECT ERR
45 05725 006111         VL.TX
```

```
!0103 N3MRT
01 05726 020522 IOVPR:  LDA 0,IOV.A
02 05727 006102         JSR @ERROI      ;FINISH PRG# MAP ETC
03 05730 000401         JMP .+1
04 05731 006132         JSR @LMESS
05 05732 006057         IOTX5           ;INSTRUCTION MESSAGE
06 05733 026046         LDA 1,@TPLOC    ;GET SAVED INSTRUCTION
07 05734 006133         JSR @LPOCT      ;PRINT IT
08 05735 006132         JSR @LMESS      ;TYPE AC'S HEADER
09 05736 006263         IOTX3
10 05737 020505         LDA 0,IOV.0
11 05740 024505         LDA 1,IOV.1
12 05741 030505         LDA 2,IOV.2
13 05742 006103         JSR @ERROC      ;PRINT AC 0,1 AND 2 AT TRP
14 05743 000401         JMP .+1
15 05744 006132         JSR @LMESS
16 05745 006272         IOTX4
17 05746 020475         LDA 0,IOV.3     ;PRINT AC3 CARRY AND RETRN
18 05747 024500         LDA 1,IOV.4
19 05750 030503         LDA 2,IOV.R
20 05751 006103         JSR @ERROC
21 05752 000401         JMP .+1
22 05753 155100         MOVL 2,3
23 05754 000457         JMP IOV.E
24 05755 003031 ICD.0:  CD.S0
25 05756 003032 ICD.1:  CD.S1
26 05757 003033 ICD.2:  CD.S2
27 05760 003034 ICD.3:  CD.S3
28 05761 000040 IOV40:  40
```

```
!0104 N3MRT
01              ;DWCHK - DEFER OR WRITE CHECK TRAP
02              ;DETERMINE LEGALITY OF TRAP
03              ;OR ERROR TYPE IF NOT EXPECTED
04
05 05762 022773 DWCHK:  LDA 0,@ICD.0
06 05763 040461         STA 0,IOV.0
07 05764 026772         LDA 1,@ICD.1
08 05765 044460         STA 1,IOV.1
09 05766 032771         LDA 2,@ICD.2
10 05767 050457         STA 2,IOV.2
11 05770 036770         LDA 3,@ICD.3
12 05771 054452         STA 3,IOV.3
13 05772 102560         SUBCL 0,0
14 05773 040454         STA 0,IOV.4
15 05774 020233         LDA 0,VLPCR             ;PC REG.
16 05775 024234         LDA 1,VLDTA             ;VIOL. DATA
17 05776 030143         LDA 2,STATS             ;STATUS REG
18 05777 060177         INTEN
19 06000 040450         STA 0,IOV.A
20 06001 044450         STA 1,IOV.B
21 06002 050450         STA 2,IOV.C
22 06003 036452         LDA 3,@IOV.7
23 06004 175004         MOV 3,3,SZR
24 06005 000673         JMP ETODT
25 06006 034140         LDA 3,PSTRT
26 06007 035407         LDA 3,7,3
27 06010 175122         MOVZL 3,3,SZC   ;GET DEFER WRITE CHK RETRN
28 06011 000422         JMP IOV.E
29 06012 175220         MOVZR 3,3
30 06013 054440         STA 3,IOV.R
31 06014 125220         MOVZR 1,1
32 06015 125223         MOVZR 1,1,SNC   ;SKP IS WRITE PROTECT VIOL.
33 06016 000404         JMP .+4
34 06017 006132         JSR @LMESS
35 06020 006142         WP.TX
36 06021 000705         JMP IOVPR
37 06022 125220         MOVZR 1,1
38 06023 125223         MOVZR 1,1,SNC   ;SKP IS DEFER PROTECT ERR
39 06024 000404         JMP .+4
40 06025 006132         JSR @LMESS
41 06026 006126         DF.TX
42 06027 000677         JMP IOVPR
43 06030 006132         JSR @LMESS
44 06031 006156         AI.TX           ;MUST BE AUTO-INDEX PROTECT ERR
45 06032 000674         JMP IOVPR       ;FINISH TYPEOUT
```

```
↓0105 N3MRT
01                ;I/O OR VALIDITY TRAP RETURN
02                ;RETURN TO TEST FORCING VIOLATION
03                ;LOGICAL ADDRESS IS IN AC3 - LEFT 1
04 06033 175220 IOV.E:  MOVZR 3,3         ;AC0-1-2=MAP REGISTERS
05 06034 054407         STA 3,IOV.3
06 06035 006421         JSR @IORLM        ;RELOAD MAP IN CASE 2ND LVL
07 06036 060277         INTDS
08 06037 034143         LDA 3,STATS       ;STATUS WORD
09 06040 075002         DOA 3,2           ;WRITE IT
10 06041 060177         INTEN
11 06042 002401         JMP @IOV.3
12 06043 000000 IOV.3:  0
13 06044 000000 IOV.0:  0
14 06045 000000 IOV.1:  0
15 06046 000000 IOV.2:  0
16 06047 000000 IOV.4:  0
17 06050 000000 IOV.A:  0
18 06051 000000 IOV.B:  0
19 06052 000000 IOV.C:  0
20 06053 000000 IOV.R:  0
21 06054 000000 IVODT:  0
22 06055 004366 IOV.7:  MAP.S
23 06056 003054 IORLM: LDMAP
24 06057 005215 IOTX5: .TXTE ↓<15><12>INSTRUCTION CAUSING TRAP = ↓
25 06076 005215 IO.TX:  .TXTE ↓<15><12>
26 06077 144411         I/O PROTECT ERROR↓
27 06111 005215 VL.TX:  .TXTE ↓<15><12>
28 06112 053011         VALIDITY PROTECT ERROR↓
29 06126 005215 DF.TX:  .TXTE ↓<15><12>
30 06127 042011         DEFER PROTECT ERROR↓
31 06142 005215 WP.TX:  .TXTE ↓<15><12>
32 06143 153411         WRITE PROTECT ERROR↓
33 06156 005215 AI.TX:  .TXTE ↓<15><12>
34 06157 040411         AUTO-INDEX PROTECT ERROR↓
35 06174 005215 DCHTX:  .TXTE ↓<15><12>
36 06175 042011         DCH VIOLATION ERROR<15><12>
37 06210 151120 PROG#<11>ERRORS<15><12>↓
```

```
↓0106 N3MRT
01 06220 005215 TXT.0:  .TXTE (<15><12>PROGRAM # (
02 06227 005215 TXT.1:  .TXTE (<15><12>AC'S (
03 06233 005215 TXT.2:  .TXTE (<15><12>SCRLO/HI (
04 06241 141504 TXT.3:  .TXTE (DCHLO/HI (
05 06246 005215 TXT.4:  .TXTE (<15><12>MEM ALLOCATION TABLE<15><12>(
06 06263 005215 IOTX3:  .TXTE (<15><12>AC0<11>AC1<11>AC2(
07 06272 005215 IOTX4:  .TXTE (<15><12>AC3<11>CARRY<11>RET ADRS(
08 06305 051525 TXT.5:  .TXTE ↓USER A<11>↓
09 06311 051525 TXT.6:  .TXTE ↓USER B<11>↓
10 06315 005215 TXT.7:  .TXTE ↓<15><12>ST.LA<11>START<11>ERROR(RES.)<15><12>↓
11 06333 005215 TXT.8:  .TXTE ↓<15><12>RAN AC=  ↓
12 06341 050240 TXT.9:  .TXTE ↓ PHYS<11>LOGICAL PHYS<11>LOGICAL↓
13 06357 050240 TXT.A:  .TXTE ↓ PHYS<11>LOGICAL RELOC<11>PHYS<11>LOGICAL RELOC↓
14 06403 005215 TXT.B:  .TXTE ↓<15><12>RELLO/HI↓
15 06411 177777 PFTEX:  .TXTE ↓<177><177><177><15><12>POWER FAIL @ ↓
16
17                ;RANDCH- RANDOM SELECT DCH MAP A OR B
18                ;
19 06423 054416 RANDCH: STA 3,RND.3
20 06424 006077         JSR @ARANG        ;GET RANDOM #
21 06425 126520         SUBZL 1,1
22 06426 107404         AND 0,1,SZR       ;SKP IS USE DCH MAP A
23 06427 126621         SUBZR 1,1,SKP     ;USE DCH MAP B
24 06430 126400         SUB 1,1
25 06431 030137         LDA 2,ALTBL       ;GET ADDR OF ALLOC TBL
26 06432 021001         LDA 0,1,2         ;GET DCH ALLOC. WORD
27 06433 101120         MOVZL 0,0
28 06434 101220         MOVZR 0,0         ;DELETE BIT 0
29 06435 123000         ADD 1,0           ;ADD IN BIT 0 SELECTED
30 06436 041001         STA 0,1,2         ;PLACE IN DCH ALLOC. WORD
31 06437 010402         ISZ RND.3         ;SKIP ON EXIT
32 06440 002401         JMP @RND.3        ;RETURN
33 06441 000000 RND.3:  0
34
```

```
10107 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 06442 062401           SAVE
06 06443 054456           STA 3,AM.S3
07 06444 022137           LDA 0,@ALTBL      ;GET #1K'S ASSIGNED
08 06445 101005           MOV 0,0,SNR       ;NOT=0 INVALID NONSCR TO ASSIGN
09 06446 000441           JMP AM.XT
10 06447 030137           LDA 2,ALTBL
11 06450 025001           LDA 1,1,2
12 06451 135000           MOV 1,3           ;SAVE AC1
13 06452 125120           MOVZL 1,1
14 06453 125224           MOVZR 1,1,SZR     ;ERROR EXIT IF
15 06454 000433           JMP AM.XT         ;DCH ALREADY ASSIGNED
16 06455 020136           LDA 0,MPSWT
17 06456 101005           MOV 0,0,SNR       ;SKP IF MAPPED
18 06457 000432           JMP AM.NM         ;NO MAP
19 06460 175113           MOVL# 3,3,SNC     ;SKP IF B DCH MAP (BIT 0=1)
20 06461 000403           JMP .+3
21 06462 034447           LDA 3,AM.BB       ;B MAP DCH
22 06463 000402           JMP .+2
23 06464 034444           LDA 3,AM.AA       ;A DCH MAP
24 06465 054403           STA 3,AM.GI+2
25 06466 006440 AM.GI:    JSR @AM.RM        ;SELECT A PAGE
26 06467 006522           AM.37             ;PAGE 0 TO 37
27 06470 000000           0                 ;A OR B DCH MAP
28 06471 000416           JMP AM.XT         ;NO CORE AVAILABLE
29 06472 105300           MOVS 0,1          ;0=LOG
30 06473 125700           INCS 1,1          ;1 OPR=+1
31 06474 030137           LDA 2,ALTBL
32 06475 035001           LDA 3,1,2
33 06476 167000           ADD 3,1           ;ADD IN BIT 0 IF ON
34 06477 045001           STA 1,1,2         ;SAVE DCH
35 06500 040424           STA 0,AM.TM
36 06501 102400           SUB 0,0
37 06502 006423           JSR @AM.GA
38 06503 063077           HALT
39 06504 020420           LDA 0,AM.TM
40 06505 004561           JSR LDCHM
41 06506 012413           ISZ @AM.S3
42 06507 006414 AM.XT:    JSR @AM.K1        ;LOAD MAP OPT. SET SCRLO+HI
43 06510 062601           RTRN
```

```
10108 N3MRT
01                ;MAP OPT DOES NOT EXIST
02                ;USE FIRST 1K SCR ASSIGNED
03 06511 102400 AM.NM:    SUB 0,0
04 06512 006413           JSR @AM.GA
05 06513 063077           HALT
06 06514 125300           MOVS 1,1
07 06515 125700           INCS 1,1
08 06516 030137           LDA 2,ALTBL       ;ENTER DCH ASSIGND
09 06517 045001           STA 1,1,2
10 06520 000766           JMP AM.XT-1
11 06521 000000 AM.S3: 0
12 06522 000037 AM.37: 37
13 06523 006713 AM.K1:    LDCHL
14 06524 000000 AM.TM: 0
15 06525 002564 AM.GA: GETPA
16 06526 002224 AM.RM: RMSEL
17 06527 000777 AM777:    777     ;
18 06530 027201 AM.AA:    DCHM0             ;A DCH MAP
19 06531 027203 AM.BB:    DCHM2             ;B DCH MAP
```

```
!0109 N3MRT
01                ;EMSCR-
02                ;EXPAND DCH SCRATCH AREA
03
04                EMSCR:
05 06532 062401           SAVE
06 06533 054766           STA 3,AM.S3
07 06534 030137           LDA 2,ALTBL
08 06535 021000           LDA 0,0,2         ;AC0=#1K SCR
09 06536 025001           LDA 1,1,2         ;AC1=DCH LIMIT
10 06537 044417           STA 1,EMS.0       ;SAVE IT
11 06540 125120           MOVZL 1,1
12 06541 125220           MOVZR 1,1         ;TURN BIT 0 OFF
13 06542 034760           LDA 3,AM.37
14 06543 137400           AND 1,3           ;3=START DCH LOG
15 06544 166700           SUBS 3,1          ;1=#1K'S DCH
16 06545 122415           SUB# 1,0,SNR      ;SKIP=NOT ALL AS-DCH
17 06546 000741           JMP AM.XT         ;EXIT ALL SR-DCH
18 06547 020136           LDA 0,MPSWT
19 06550 101004           MOV 0,0,SZR       ;SKIP IF NOT MAPPED
20 06551 000410           JMP EM.UM
21 06552 125700           INCS 1,1          ;+1 #DCH 1K'S
22 06553 137000           ADD 1,3
23 06554 055001           STA 3,1,2
24 06555 000731           JMP AM.XT-1       ;SKIP EXIT 1 MORE 1K
25
26 06556 000000 EMS.0:    0
```

```
!0110 N3MRT
01
02                ;MAP OPTION EXISTS SEE IF THE NEXT
03                ;LOGICAL 1K DCH IS AVAILABLE
04 06557 000040 EM40: 40
05 06560 002575 EM.CM:    CMAPB
06 06561 137000 EM.UM:    ADD 1,3
07 06562 161000           MOV 3,0           ;(0)=NEXT HIGH DCH
08 06563 034774           LDA 3,EM40
09 06564 162415           SUB# 3,0,SNR      ;SKIP IS STILL IN 32K
10 06565 000722           JMP AM.XT
11 06566 030702           LDA 2,AM.GI+2     ;DCH BIT MAP
12 06567 006771           JSR @EM.CM        ;SKIP IF AVAIL
13 06570 006770           JSR @EM.CM        ;WASN'T SET IT=0 AND SKP
14 06571 135701           INCS 1,3,SKP      ;(3)UPPR=NEW DCH+1K
15 06572 000715           JMP AM.XT         ;EXIT CANT EXPAND UP
16 06573 040731           STA 0,AM.TM
17 06574 122400           SUB 1,0
18 06575 117000           ADD 0,3
19 06576 030137           LDA 2,ALTBL
20 06577 020757           LDA 0,EMS.0
21 06600 101103           MOVL 0,0,SNC      ;SKP IS B DCH MAP
22 06601 000403           JMP .+3
23 06602 102620           SUBZR 0,0         ;TURN ON BIT 0
24 06603 000402           JMP .+2
25 06604 102400           SUB 0,0
26 06605 117000           ADD 0,3           ;ADD IN BIT 0(ON OR OFF)
27 06606 055001           STA 3,1,2         ;RESTORE DCH ASSIGNMENTS
28 06607 121000           MOV 1,0           ;GET PHYSICAL PAGE #
29 06610 006715           JSR @AM.GA
30 06611 063077           HALT
31 06612 020712           LDA 0,AM.TM
32 06613 004453           JSR LDCHM
33 06614 000672           JMP AM.XT-1
```

```
10111 N3MRT
01                ;RDSCR - RELEASE SCRATCH FROM DCH
02                RDSCR:
03 06615 062401           SAVE
04 06616 054703           STA 3,AM.S3
05 06617 030137           LDA 2,ALTBL
06 06620 021001           LDA 0,1,2
07 06621 126620           SUBZR 1,1
08 06622 107400           AND 0,1           ;AC1 CONTAINS B DCH FLG
09 06623 044442           STA 1,RDSV0       ;SAVE IT
10 06624 101120           MOVZL 0,0
11 06625 101225           MOVZR 0,0,SNR
12 06626 000661           JMP AM.XT         ;EXIT NO DCH TO RELEASE
13 06627 126000           ADC 1,1           ;-1
14 06630 101300           MOVS 0,0
15 06631 123300           ADDS 1,0          ;#1K'S IN DCH-1
16 06632 034433           LDA 3,RDSV0       ;GET SAVED DCH FLG
17 06633 163000           ADD 3,0           ;ADD IN BIT 0(ON/OFF)
18 06634 041001           STA 0,1,2
19 06635 162400           SUB 3,0
20 06636 024664           LDA 1,AM.37       ;(AC1) AFTER AND
21 06637 107400           AND 0,1           ;=LOGICAL PAGE #
22 06640 122704           SUBS 1,0,SZR      ;AC0=#1K'S LEFT
23 06641 000405           JMP .+5
24 06642 163000           ADD 3,0           ;ADD IN BIT 0 IF ON
25 06643 041001           STA 0,1,2         ;0 DCH ASSIGNED
26 06644 162400           SUB 3,0           ;REMOVE BIT 0 IF ON
27 06645 014654           DSZ AM.S3         ;NO SKIP ON EXIT
28 06646 034136           LDA 3,MPSWT
29 06647 175005           MOV 3,3,SNR       ;SKIP IF REALLY MAPPED
30 06650 000636           JMP AM.XT-1
31 06651 123000           ADD 1,0           ;AC0=LOG PAGE #
32 06652 024655           LDA 1,AM777       ;
33 06653 004413           JSR LDCHM
34 06654 034411           LDA 3,RDSV0       ;A OR B DCH
35 06655 175103           MOVL 3,3,SNC      ;SKP IS B DCH MAP
36 06656 000403           JMP .+3
37 06657 030652           LDA 2,AM.BB       ;B DCH MAP
38 06660 000402           JMP .+2
39 06661 030647           LDA 2,AM.AA       ;A DCH MAP
40 06662 006676           JSR @EM.CM        ;MAKE 1K DCH AVAIL
41 06663 000623           JMP AM.XT-1       ;EXIT NEW DCH LIM'S
42 06664 063077           HALT    ;DCH MAP BIT WAS ALRDY 1(CAN'T HAPPEN)
43 06665 000000 RDSV0:    0
```

```
10112 N3MRT
01                ;LDCHM - LOAD DATA CHANNEL MAP
02                ;LOGICAL DCH IS IN AC0
03                ;PHYSICAL DCH IS IN AC1
04                ;USED BY ASSIGN DCH AND EXPAND DCH
05 06666 040421 LDCHM:    STA 0,LDC.0
06 06667 101300           MOVS 0,0
07 06670 103120           ADDZL 0,0
08 06671 044417           STA 1,LDC.1
09 06672 123000           ADD 1,0
10 06673 030137           LDA 2,ALTBL       ;ALLOCATION TBL
11 06674 031001           LDA 2,1,2         ;GET DCH FLG
12 06675 151103           MOVL 2,2,SNC      ;SKP IS B DCH WAS USED
13 06676 000403           JMP .+3
14 06677 024412           LDA 1,LDC.B       ;B DCH MAP
15 06700 000402           JMP .+2
16 06701 024411           LDA 1,LDC.A       ;A DCH MAP
17 06702 123000           ADD 1,0
18 06703 062002           DOB 0,2           ;WRITE MAP ENTRY
19 06704 024404           LDA 1,LDC.1
20 06705 020402           LDA 0,LDC.0
21 06706 001400           JMP 0,3
22 06707 000000 LDC.0:  0
23 06710 000000 LDC.1:  0
24 06711 101000 LDC.B:   101000   ;B DCH MAP
25 06712 100000 LDC.A:   100000   ;A DCH MAP
```

```
!0113 N3MRT
01                ;LDCHL - LOAD DATA CHANNEL LIMITS
02                ;CALCULATE DCHHI/LO FOR THIS TST
03                ;INTEGRITY OF AC'S NOT PRESERVED
04 06713 030137 LDCHL:  LDA 2,ALTBL
05 06714 025001         LDA 1,1,2         ;AC1 UPPER=#1K'S
06 06715 131000         MOV 1,2
07 06716 020604         LDA 0,AM.37       ;AC1 LOWER=1ST LOG 1K
08 06717 123400         AND 1,0
09 06720 106400         SUB 0,1
10 06721 101300         MOVS 0,0
11 06722 103120         ADDZL 0,0         ;AC0=LO LIMIT DCH
12 06723 151103         MOVL 2,2,SNC      ;SKP IS USE B DCH MAP
13 06724 000403         JMP .+3
14 06725 152620         SUBZR 2,2         ;BIT 0 = 1
15 06726 143000         ADD 2,0           ;ADD INTO LOGICAL ADDRESS
16 06727 040154         STA 0,DCHLO
17 06730 152620         SUBZR 2,2
18 06731 150000         COM 2,2
19 06732 143400         AND 2,0
20 06733 147400         AND 2,1           ;REMOVE BIT 0
21 06734 127125         ADDZL 1,1,SNR     ;AC1=MOD 1K # 1'K
22 06735 125040         MOVO 1,1          ;SET C=1 NO 1K'S
23 06736 123000         ADD 1,0           ;AC0=HIGH SCR+1
24 06737 126500         SUBL 1,1          ;AC1=1 IF DCH ASSIGNED
25 06740 122400         SUB 1,0           ;AC1=0 IF NO DCH
26 06741 040155         STA 0,DCHHI       ;HI DCH LOG LIMIT
27 06742 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 06743 026413 LPRTN:  LDA 1,@LPRSL-1
32 06744 125005         MOV 1,1,SNR
33 06745 002504         JMP @LPSV3        ;AUTO-RUN, DON'T WAIT
34 06746 024407         LDA 1,LPR60
35 06747 044151         STA 1,LASTI       ;FORCE 0 SWREG CMD
36 06750 006132         JSR @LMESS
37 06751 007160         LPRT5             ;ENTER OPTIONS MESSAGE
38 06752 006402         JSR @LPRCK        ;ENTER KEY PACKAGE
39 06753 002476         JMP @LPSV3        ;EXIT
40 06754 001065 LPRCK:  CKODT
41 06755 100060 LPR60:  100060
```

```
!0114 N3MRT
01                ;LPRSL-LINKER PROGRAM SELECT OR,
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 06756 001034         LAUTO
07 06757 054472 LPRSL:  STA 3,LPSV3
08 06760 102400         SUB 0,0
09 06761 040471         STA 0,LPRGN
10 06762 020476         LDA 0,LPRT1
11 06763 006073         JSR @ERRTX
12 06764 026467         LDA 1,@LPHIK      ;GET HIGHK PHYS.
13 06765 125400         INC 1,1           ;=MOD 1K
14 06766 006072         JSR @LPDEC        ;PRINT SUPR 0'S
15 06767 024136         LDA 1,MPSWT
16 06770 020515         LDA 0,MPXTX       ;MAP EXIST TEXT
17 06771 125005         MOV 1,1,SNR       ;SKP IF MAP EXIS
18 06772 020522         LDA 0,NMPTX       ;NO MAP TEXT
19 06773 006073         JSR @ERRTX
20 06774 026762         LDA 1,@LPRSL-1
21 06775 020524         LDA 0,LPRT2
22 06776 125004         MOV 1,1,SZR       ;SKP IF AUTO STRT
23 06777 020533         LDA 0,LPRT3       ;NOT AUTO USE OTHR HDR
24 07000 006073         JSR @ERRTX
25 07001 020447         LDA 0,LPRT4
26 07002 006073         JSR @ERRTX        ;PRT PRG HDR
27 07003 006070         JSR @PCRLF        ;CARRET LFEED TST 0
28                ;PRINT IDIVIDUAL TEST DESCRIPTIONS
29                ;GIVE OPR CHANCE TO DELETE IF LAUTO=-1
30 07004 020446 LPRLP:  LDA 0,LPRGN       ;CUR PROG #
31 07005 034442         LDA 3,LPLZM
32 07006 117000         ADD 0,3
33 07007 031400         LDA 2,0,3
34 07010 151005         MOV 2,2,SNR       ;SKP IF NOT LAST
35 07011 000732         JMP LPRTN
36 07012 105000         MOV 0,1
37 07013 021002         LDA 0,2,2         ;GET PRG WAIT SW
38 07014 101004         MOV 0,0,SZR       ;SKP IF PRG NOT WAIT
39 07015 000430         JMP LPR1E         ;DEV MUST NOT EXIST
40 07016 050441         STA 2,LPIDX
41 07017 006071         JSR @LZOCT        ;TYPE PRG #
42 07020 020437         LDA 0,LPIDX
43 07021 024435         LDA 1,LPR10
44 07022 123000         ADD 1,0           ;CALC ADRS DESC TXT
45 07023 006073         JSR @ERRTX
46 07024 026732         LDA 1,@LPRSL-1
47 07025 125005         MOV 1,1,SNR       ;SKP IS LET OPR SELECT
48 07026 000416         JMP LPR1E-1       ;CR/LF AND DO NXT PRG
```

```
!0115 N3MRT
01                ;WAIT FOR OPERATOR INPUT TO SELECT TEST
02                ;SPACE IS SELECT ANY OTHER IS DELETE
03 07027 063610           SKPDN TTI
04 07030 000777           JMP .-1
05 07031 064610           DIAC 1,TTI        ;GET CHAR
06 07032 030422           LDA 2,LPR77
07 07033 147400           AND 2,1
08 07034 030421           LDA 2,LPR40
09 07035 146415           SUB# 2,1,SNR      ;SKP IF DELETED
10 07036 000406           JMP LPR1E-1       ;SELECTED CR/LF DO NEXT
11 07037 030420           LDA 2,LPIDX
12 07040 102000           ADC 0,0
13 07041 041002           STA 0,2,2         ;SET WAIT SW IN PRG
14 07042 006132           JSR @LMESS
15 07043 007263           LPDTX+1
16 07044 006070           JSR @PCRLF        ;CR/LF
17 07045 010405 LPR1E:    ISZ LPRGN         ;+1 PROG #
18 07046 000736           JMP LPRLP         ;DO NEXT PROG
19 07047 000156 LPLZM: LZMAX
20 07050 007250 LPRT4: LPR4T
21 07051 000000 LPSV3: 0
22 07052 000000 LPRGN: 0
23 07053 001502 LPHIK: HIGHK
24 07054 000077 LPR77: 77
25 07055 000040 LPR40: 40
26 07056 000010 LPR10: 10
27 07057 000000 LPIDX: 0
28 07060 007061 LPRT1:    .+1
29 07061 005215           .TXTE !<15><12>N3MRT LONG - REV 04
30 07074 106640 <15><12>TOTAL # 1K'S= !
31 07105 007106 MPXTX:    .+1
32 07106 040515           .TXTE !MAP EXISTS!
33 07114 007115 NMPTX:    .+1
34 07115 147516           .TXTE !NO MAP!
35 07121 007122 LPRT2:    .+1
36 07122 005215           .TXTE (<15><12>TEST RUN LIST(
37 07132 007133 LPRT3:    .+1
38 07133 005215           .TXTE (<15><12>SPACE SELECTS
39 07142 120123  -ANY OTHER CHAR. DELETES(
40 07160 005215 LPRT5:    .TXTE !<15><12>ENTER OPTIONS, CR TO CONTINUE!
41 07200 005215 FTYTX:    .TXTE !<15><12>50 ERRORS<15><12>!
42 07207 005215 KEY6T:    .TXTE !<15><12>TYPE CR TO CONTINUE!
43 07222 005215 TX65K:    .TXTE !<15><12>65K PASSES TEST # !
44 07235 005215 STHDR:    .TXTE !<15><12>TST#<11>PASSES<11>ERRS<15><12>!
45 07250 005215 LPR4T:    .TXTE (<15><12>TST#<11>DESCRIPTION(
46 07262 007263 LPDTX:    .+1
47 07263 042240           .TXTE   ( DELETED(
48 07270 005215 UDEVT:    .TXTE !<15><12>UNEX. INTR. DEV# !
```

```
!0116 N3MRT
01                .MACRO SETUP
02                LCALL SETUL
03             %
04
05                .MACRO LOOP
06                LCALL LLOOP
07             %
08                .MACRO ERROR
09                JSR .+2
10                JMP .+2
11                LCALL ERRET
12             %
13                .MACRO EHALT
14                JSR .+2
15                JMP .+2
16                LCALL ERRET
17             %
```

```
!0117 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              007302 LMEML=.
11              000156          .LOC LPG0
12   00156      007305          CB.00
13              000157 LPG0=.
14              007302          .LOC LMEML
15   07302      000000          0       ;TEST PASS CTR
16   07303      000000          0       ;TEST ERROR CTR
17   07304      000000          0       ;INTERRUPT TIMEOUT SWITCH
18   07305      007324 CB.00:   CB.01   ;INIT ENTRY ADRS.
19   07306      007327          CB.02   ;EXECUTE ENTRY ADRS
20   07307      000000          0       ;NO INTR WAITS
21   07310      000000          0       ;RAND SEL LIMITS
22   07311      177777          -1      ;=ALWAYS ENTER
23   07312      176000          176000  ;EVERY PROTECT BIT ON
24   07313      007455          CB.EC   ;NO I/O VALIDITY TRAPS
25   07314      007455          CB.EC   ;NO WRITE OR DEFER TRAPS
26              007315          .TXTE (
27   07315      044303 CHKRBRD RAN.(
28              151113
29              151102
30              120104
31              040722
32              027116
33              000000
```

```
!0118 N3MRT
01                      ;INITIALIZE CHECKERBOARD TEST SEQUENCE
02   07324      102000 CB.01:  ADC 0,0
03   07325      040555         STA 0,CB.TK      ;-1 TO TEST COUNTER
04   07326      001400         JMP 0,3 ;RETURN TO LINKER TEST INIT
05                      ;
06                      ;EXECUTE ENTRY POINT
07   07327      010553 CB.02:  ISZ CB.TK        ;SKP IS NO SCRATCH
08   07330      000473         JMP CB.03        ;DO NXT IN SEQ
09                             LCALL ASCRA      ;TRY TO GET 1K
10   07331      100010         ASCRA-ASCRA*1B11+100010
11   07332      000507         JMP CB.X1        ;NONE AVAILABLE
12   07333      102000         ADC 0,0          ;-1 TO
13   07334      040557         STA 0,CB.ES      ;NO ERRSW
14                             LCALL ARANG      ;GET RAN#
15   07335      100270         ARANG-ASCRA*1B11+100010
16   07336      030545         LDA 2,CB.37
17   07337      105000         MOV 0,1
18                             LCALL ADIVI      ;REM=#1K'S TO EXPAND
19   07340      100310         ADIVI-ASCRA*1B11+100010
20   07341      100405         NEG 0,0,SNR
21   07342      000405         JMP CB.2A
22                      CB.2L: LCALL ESCRA      ;EXPAND SCRATCH 1K
23   07343      100030         ESCRA-ASCRA*1B11+100010
24   07344      000403         JMP CB.2A        ;NO MORE AVAILABLE
25   07345      101404         INC 0,0,SZR
26   07346      000775         JMP CB.2L        ;KEEP EXPANDING
```

```
!0119 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 07347 100270           ARANG-ASCRA*1B11+100010
06 07350 024152           LDA 1,SCRLO
07 07351 030153           LDA 2,SCRHI
08 07352 132400           SUB 1,2          ;# WORDS IN SCRATCH
09 07353 105000           MOV 0,1
10                        LCALL ADIVI      ;CREATE AN ADRS
11 07354 100310           ADIVI-ASCRA*1B11+100010
12 07355 024152           LDA 1,SCRLO      ;WITHIN SCRATCH
13 07356 123000           ADD 1,0          ;TO RELOCATE TO
14 07357 030525 CB.2C:    LDA 2,CB.PL      ;PROG LENGTH
15 07360 142400           SUB 2,0
16 07361 024534           LDA 1,CBPL2
17 07362 122400           SUB 1,0          ;CALC LOW REL ADRS
18 07363 024152           LDA 1,SCRLO
19 07364 122433           SUBZ# 1,0,SNC    ;SKP >LOLIMIT
20 07365 143000           ADD 2,0 ;MAKE GRTR THAN LO
21 07366 122433           SUBZ# 1,0,SNC
22 07367 000776           JMP .-2
23 07370 040515           STA 0,CB.LC      ;ADRS TO STORE TSTS
24 07371 144000 CB.RL:    COM 2,1          ;-#WORDS TO MOV
25 07372 111000           MOV 0,2          ;TO ADRS
26 07373 034513           LDA 3,CB.BG      ;FROM ADRS
27 07374 021400 CB.L2:    LDA 0,0,3
28 07375 041000           STA 0,0,2
29 07376 175400           INC 3,3
30 07377 151400           INC 2,2
31 07400 125404           INC 1,1,SZR
32 07401 000773           JMP CB.L2        ;MOV ALL TO SCRTCH
33 07402 020513           LDA 0,CBPL2
34 07403 104000           COM 0,1
35 07404 143000           ADD 2,0
36 07405 101400           INC 0,0
37 07406 041377           STA 0,-1,2       ;SET EPROG
38 07407 020476           LDA 0,CB.LC
39 07410 041376           STA 0,-2,2       ;SET BPROG
40 07411 102400           SUB 0,0
41 07412 041374           STA 0,-4,2       ;CLR PONES
42 07413 050474           STA 2,CB.EN
43                ;MOV REST OF TEST INTO SCRATCH AREA
44 07414 034500           LDA 3,CBBG2      ;STRT OF 2ND SECTION
45 07415 021400 CB.L3:    LDA 0,0,3
46 07416 041000           STA 0,0,2
47 07417 151400           INC 2,2
48 07420 175400           INC 3,3
49 07421 125404           INC 1,1,SZR
50 07422 000773           JMP CB.L3
51                ;2ND PART OF TEST IS MOVED
```

```
!0120 N3MRT
01
02                ;TEST PROGRAMS HAVE REEN MOVED TO SCRATCH AREA
03                ;EXECUTE NEXT TES IN SEQUENCE
04 07423 030457 CB.03:    LDA 2,CB.TK      ;GET TEST COUNT
05 07424 024472           LDA 1,CB.TS      ;ADRS SEQ TABLE
06 07425 133000           ADD 1,2
07 07426 035000           LDA 3,0,2        ;GET RELATIVE POSITION
08 07427 024456           LDA 1,CB.LC      ;STRT POS IN SCRATCH
09 07430 137000           ADD 1,3
10 07431 050457           STA 2,CB.TI
11 07432 054460           STA 3,CB.SE
12 07433 005400           JSR 0,3 ;***GO TO TEST ***
13 07434 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 07435 020454           LDA 0,OTOTL      ;FAST SUM RESULT
19 07436 030451           LDA 2,CB.EN
20 07437 025374           LDA 1,-4,2       ;#-1'S GENERATED
21 07440 004413           JSR CB.ER
22                CB.X1:  LCALL RSCRA      ;RELEASE 1K SCRATCH
23 07441 100050           RSCRA-ASCRA*1B11+100010
24 07442 102001           ADC 0,0,SKP      ;RET ALL RELEASED
25 07443 000776           JMP CB.X1        ;RELEASE ALL
26 07444 040436           STA 0,CB.TK      ;SET NO SCRATCH SW
27                        LCALL RETRN
28 07445 100210           RETRN-ASCRA*1B11+100010
29                ;NORMAL RETURN FROM TEST SEE IF PASS COMPLETE
30 07446 010442 CB.04:    ISZ CB.TI
31 07447 022441           LDA 0,@CB.TI     ;NEXT IN SEQ
32 07450 100005           COM 0,0,SNR      ;=-1 WAS END SEQ
33 07451 000417           JMP CB.X2        ;AND RELEASE SCRA.
34                        LCALL RETRN
35 07452 100210           RETRN-ASCRA*1B11+100010
```

```
!0121 N3MRT
01                ;PATTERN CHECK FOUND AN ERROR
02 07453 054437 CB.ER:  STA 3,CB.SE
03                      LCALL ERROI
04 07454 100350         ERROI-ASCRA*1B11+100010
05 07455 000401 CB.EC:  JMP .+1 ;CONTINUE ERROR TYPEOUT
06 07456 020446         LDA 0,CBTXT
07                      LCALL ERRTX
08 07457 100170         ERRTX-ASCRA*1B11+100010
09 07460 020422         LDA 0,CB.TK     ;TEST #
10 07461 024424         LDA 1,CB.LC     ;LOGICAL ADDRESS
11 07462 030430         LDA 2,CB.SE     ;START TEST OR E CALL
12                      LCALL ERROC
13 07463 100370         ERROC-ASCRA*1B11+100010
14 07464 000755         JMP CB.X1       ;SW0=1 RELEASE SCRATCH
15 07465 102400         SUB 0,0         ;OTHERWISE HOLD IT
16 07466 040425         STA 0,CB.ES     ;AS IS UNTIL SW0=1
17 07467 000757         JMP CB.04
18
19 07470 020230 CB.X2:  LDA 0,SWREG     ;CHECK IF RELEASEING
20 07471 103123         ADDZL 0,0,SNC   ;SCRATCH-SKP IS NOT REL.
21 07472 000445         JMP CB.05       ;RELEASE IT
22 07473 020420         LDA 0,CB.ES     ;GET ERR SWITCH
23 07474 040406         STA 0,CB.TK     ;-1 IS NO ERRS
24 07475 101004         MOV 0,0,SZR
25 07476 000441         JMP CB.05       ;AND SCRATCH IS RELEASED
26 07477 020406         LDA 0,CB.LC     ;OTHERWISE MOVE
27 07500 030404         LDA 2,CB.PL     ;PROGRAM UP AND
28 07501 000670         JMP CB.RL       ;RESTART AT TEST 0
29
30 07502 000000 CB.TK: 0
31 07503 000037 CB.37: 37
32 07504 000151 CB.PL: EPROG-BEGIN
33 07505 000000 CB.LC: 0
34 07506 007560 CB.BG: BEGIN
35 07507 000000 CB.EN: 0
36 07510 000000 CB.TI: 0
37 07511 000000 OTOTL: 0
38 07512 000000 CB.SE: 0
39 07513 000000 CB.ES: 0
40 07514 007732 CBBG2: DISTUR
41 07515 000104 CBPL2: DIRET-DISTUR
42 07516 007517 CB.TS:  CB.TS+1 ;TEST SEQUENCE TABLE
43 07517 000000         BEGIN-BEGIN
44 07520 000152         DISTUR-BEGIN
45 07521 000045         ICHECK-BEGIN
46 07522 000101         CB.FA-BEGIN
47 07523 177777         -1
48 07524 007525 CBTXT: .+1
49 07525 005215         .TXTE (<15><12>CB.TK    CB.LC   CB.SE(
```

```
!0122 N3MRT
01                ;DETERMINE IF IT IS TIME TO SWAP MEMORY
02                ;CROSSOVER CONSTANTS
03                CB.05: LCALL ARANG
04 07537 100270         ARANG-ASCRA*1B11+100010
05 07540 024557         LDA 1,C17
06 07541 123404         AND 1,0,SZR
07 07542 000677         JMP CB.X1
08 07543 020411         LDA 0,CB17
09 07544 030411         LDA 2,CB400
10 07545 106414         SUB# 0,1,SZR
11 07546 000403         JMP .+3
12 07547 020407         LDA 0,CB37
13 07550 030407         LDA 2,CB10K
14 07551 040546         STA 0,C17
15 07552 050544         STA 2,C400
16 07553 000666         JMP CB.X1
17 07554 000017 CB17: 17
18 07555 000400 CB400: 400
19 07556 000037 CB37: 37
20 07557 010000 CB10K: 10000
```

```
!0123 N3MRT
01
02                ;THE FOLLOWING SERIES OF TESTS ARE RELOCATED TO
03                ;SCRATCH AREA ASSIGNED FOR EXECUTION
04 07560 054532 BEGIN:  STA 3,RETURN
05 07561 102400         SUB 0,0
06 07562 040544         STA 0,PONES
07 07563 034152         LDA 3,SCRLO
08 07564 030527         LDA 2,C076000
09 07565 020153         LDA 0,SCRHI      ;3/0=LIMITS
10 07566 143400         AND 2,0          ;END DISTURB
11 07567 040526         STA 0,EDIST
12 07570 173400         AND 3,2
13 07571 050523         STA 2,MODUAL
14 07572 030152 IPAT:   LDA 2,SCRLO
15 07573 024523         LDA 1,C400
16 07574 020533         LDA 0,PATT       ;START PATTERN
17 07575 147404         AND 2,1,SZR
18 07576 100000 IPAT1:  COM 0,0
19 07577 024520         LDA 1,C17
20 07600 034530 FILL:   LDA 3,BPROG      ;AVOID OVERWRITE TST
21 07601 156436         SUBZ# 2,3,SEZ
22 07602 000404         JMP .+4
23 07603 034526         LDA 3,EPROG
24 07604 172433         SUBZ# 3,2,SNC    ;SKP>PROG END
25 07605 000405         JMP .+5
26 07606 041000         STA 0,0,2        ;STR PAT WORD
27 07607 034517         LDA 3,PONES
28 07610 117000         ADD 0,3
29 07611 054515         STA 3,PONES      ;ACCUMULATE CHKSUM
30 07612 151400         INC 2,2          ;FOR NXT STOR
31 07613 034506         LDA 3,C77
32 07614 133414         AND# 1,2,SZR     ;FILL FOR 16
33 07615 000763         JMP FILL
34 07616 157414         AND# 2,3,SZR     ;THEN COM PAT
35 07617 000757         JMP IPAT1        ;EVERY 64
36 07620 024153         LDA 1,SCRHI      ;TST FOR END
37 07621 125400         INC 1,1
38 07622 146434         SUBZ# 2,1,SZR
39 07623 000750         JMP IPAT+1
40 07624 002466         JMP @RETURN
```

```
!0124 N3MRT
01                ;CHECK PATTERN IN SCRATCH AREA AGAINST GENERATED
02 07625 054465 ICHECK: STA 3,RETURN
03 07626 030152         LDA 2,SCRLO      ;INIT PAT
04 07627 024467         LDA 1,C400
05 07630 020477         LDA 0,PATT
06 07631 133414         AND# 1,2,SZR
07 07632 100000 ICK:    COM 0,0
08 07633 024475         LDA 1,BPROG      ;DON'T CMP TST STOR
09 07634 146436         SUBZ# 2,1,SEZ
10 07635 000405         JMP CHECK
11 07636 024473         LDA 1,EPROG
12 07637 132436         SUBZ# 1,2,SEZ
13 07640 000402         JMP CHECK
14 07641 000404         JMP ECHECK-4
15 07642 025000 CHECK:  LDA 1,0,2        ;GET WORD FROM MEM
16 07643 106414         SUB# 0,1,SZR     ;SKP=
17 07644 006461         JSR @ERR1
18 07645 151400         INC 2,2
19 07646 024451         LDA 1,C17
20 07647 133414         AND# 1,2,SZR
21 07650 000763         JMP ICK+1
22 07651 034450 ECHECK: LDA 3,C77
23 07652 157414         AND# 2,3,SZR     ;CHK END OF LINE
24 07653 000757         JMP ICK
25 07654 024153         LDA 1,SCRHI
26 07655 125400         INC 1,1
27 07656 146434         SUBZ# 2,1,SZR    ;CHK END OF CORE
28 07657 000750         JMP ICHECK+2
29 07660 002432         JMP @RETURN
```

```
!0125 N3MRT
01
02                ;FAST CHECKSUM PATTERN SHOULD = THAT STORED
03 07661 054431 CB.FA:  STA 3,RETURN
04 07662 102400         SUB 0,0
05 07663 030152         LDA 2,SCRLO       ;BEGIN OF SCRA.
06 07664 034444         LDA 3,BPROG       ;START OF TST STORE
07 07665 156405 CB.F1:  SUB 2,3,SNR
08 07666 000407         JMP CB.F3
09 07667 174400         NEG 3,3 ;(3)=#WORDS TO ADD
10 07670 025000 CB.F2:  LDA 1,0,2         ;GET WRD
11 07671 123000         ADD 1,0           ;ACCUM SUM-1'S
12 07672 151400         INC 2,2
13 07673 175404         INC 3,3,SZR
14 07674 000774         JMP CB.F2
15 07675 034153 CB.F3:  LDA 3,SCRHI
16 07676 175400         INC 3,3
17 07677 156415         SUB# 2,3,SNR      ;DONE ALL CORE
18 07700 000403         JMP CB.F4         ;YES EXIT
19 07701 030430         LDA 2,EPROG
20 07702 000763         JMP CB.F1         ;ADD ABOVE TSTS
21 07703 024423 CB.F4:  LDA 1,PONES       ;(1)=#-1'S GEN
22 07704 122414         SUB#1,0,SZR       ;SHD BE=
23 07705 000402         JMP CB.F5         ;BUT AREN'T
24 07706 002404         JMP @RETURN
25 07707 010403 CB.F5:  ISZ RETURN        ;STEP EXIT
26 07710 042414         STA 0,@CB.WK      ;SAVE IN CASE
27 07711 000715         JMP ICHECK+1      ;CHECK CAN'T FIND ERR
28 07712 000000 RETURN: 0
29 07713 076000 C07600: 76000
30 07714 000000 MODUAL: 0
31 07715 000000 EDIST: 0
32 07716 000400 C400: 400
33 07717 000017 C17: 17
34 07720 000020 C20: 20
35 07721 000077 C77: 77
36 07722 001777 C1777: 1777
37 07723 000101 C101: 101
38 07724 007511 CB.WK: OTOTL
39 07725 007453 ERR1: CB.ER
40 07726 000000 PONES: 0
41 07727 000000 PATT: 0
42 07730 000000 BPROG: 0
43 07731 000000 EPROG: 0
```

```
!0126 N3MRT
01
02
03                ;SHUFFLE MEM BY FLOATING A BIT OR NO BIT THROUGH 16 WORD
04 07732 054760 DISTUR: STA 3,RETURN
05 07733 030152         LDA 2,SCRLO
06 07734 145000 DISTL:  MOV 2,1           ;2=START OF 16
07 07735 020763         LDA 0,C20         ;MAKE 1=END+1
08 07736 107000         ADD 0,1
09 07737 034771         LDA 3,BPROG       ;START OR "BEGIN)
10 07740 136436         SUBZ# 1,3,SEZ     ;END 16>BEG
11 07741 000411         JMP DISDO         ;NO<DO THIS 16
12 07742 034767         LDA 3,EPROG       ;GET END PRG
13 07743 172436         SUBZ# 3,2,SEZ     ;STRT 16>END PRG
14 07744 000403         JMP .+3           ;YES
15 07745 131000         MOV 1,2           ;TRY STRT NXT16
16 07746 000766         JMP DISTL
17 07747 034153         LDA 3,SCRHI
18 07750 166436         SUBZ# 3,1,SEZ     ;END 16>SCRHI
19 07751 002741         JMP @RETURN       ;YES EXIT JOB DONE
20 07752 004450 DISDO:  JSR DIXOR         ;FLT A 1 BIT 16 WORDS
21 07753 020745         LDA 0,C20
22 07754 112400         SUB 0,2           ;BACK TO START OF 16
23 07755 114400         NEG 0,3           ;SHUFFLE 16 TIMES
24 07756 021000 DISDL:  LDA 0,0,2         ;WORD 0
25 07757 025001         LDA 1,1,2
26 07760 041001         STA 0,1,2         ;GOES TO WORD 1
27 07761 021002         LDA 0,2,2
28 07762 045002         STA 1,2,2         ;1 GOES TO 2
29 07763 025003         LDA 1,3,2
30 07764 041003         STA 0,3,2         ;2 GOES TO 3
31 07765 021004         LDA 0,4,2
32 07766 045004         STA 1,4,2         ;3 GOES TO 4
33 07767 025005         LDA 1,5,2
34 07770 041005         STA 0,5,2         ;4 GOES TO 5
35 07771 021006         LDA 0,6,2
36 07772 045006         STA 1,6,2         ;5 GOES TO 6
37 07773 025007         LDA 1,7,2
38 07774 041007         STA 0,7,2         ;6 TO 7
39 07775 021010         LDA 0,10,2
40 07776 045010         STA 1,10,2        ;7 TO 10
41 07777 025011         LDA 1,11,2
42 10000 041011         STA 0,11,2        ;10 TO 11
43 10001 021012         LDA 0,12,2
44 10002 045012         STA 1,12,2        ;11 TO 12
45 10003 025013         LDA 1,13,2
46 10004 041013         STA 0,13,2        ;12 TO 13
47 10005 021014         LDA 0,14,2
48 10006 045014         STA 1,14,2        ;13 TO 14
49 10007 025015         LDA 1,15,2
50 10010 041015         STA 0,15,2        ;14 TO 15
51 10011 021016         LDA 0,16,2
52 10012 045016         STA 1,16,2        ;15 TO 16
53 10013 025017         LDA 1,17,2
54 10014 041017         STA 0,17,2        ;16 TO 17
55 10015 045000         STA 1,0,2         ;AND 17 BACK TO 0
56 10016 175404         INC 3,3,SZR       ;DONE 16 TIMES
57 10017 000737         JMP DISDL         ;WORDS NOT BACK TO ORIG YET
58 10020 004402         JSR DIXOR         ;XOR BITS BACK TO ORIG
59 10021 000713         JMP DISTL         ;DO REST OF SCRATCH
```

```
!0127 N3MRT
01                    ;COMS A SINGLE  BIT IN EACH OF NEXT 16 WORDS
02 10022 054414 DIXOR:  STA 3,DIRET     ;THEN RESTORES TO ORIG
03 10023 102520         SUBZL 0,0       ;XOR 1 BIT
04 10024 025000         LDA 1,0,2       ;GET NEXT WRD
05 10025 135000         MOV 1,3         ;START BIT XOR
06 10026 117520         ANDZL 0,3       ;SO FLT BIT PAT
07 10027 107000         ADD 0,1         ;APPEARS IN THESE
08 10030 166400         SUB 3,1         ;16 WORDS
09 10031 045000         STA 1,0,2
10 10032 151400         INC 2,2
11 10033 101124         MOVZL 0,0,SZR   ;DONE ALL - POS BIT L1
12 10034 000770         JMP DIXOR+2
13 10035 002401         JMP @DIRET
14 10036 000000 DIRET:  0
15                      .ENDC
```

```
!0128 N3MRT
01
02                  ;SC-MEMORY TEST - DEFINE PARAMETERS TO LINKR
03
04                          NEXTT  MM.00
05          010037 LMEML=.
06          000157          .LOC LPG0
07 00157 010042             MM.00
08          000160 LPG0=.
09          010037          .LOC LMEML
10 10037 000000             0       ;TEST PASS CTR
11 10040 000000             0       ;TEST ERROR CTR
12 10041 000000             0       ;INTERRUPT TIMEOUT SWITCH
13
14 10042 010062 MM.00:      MM.01                       ;INIT ENTRY
15 10043 010067             MM.02                       ;EXEC ENTRY
16 10044 000000             0                           ;WAIT SW
17 10045 000000             0                           ;
18 10046 177777             -1                          ;ALWAYS ENTER
19 10047 176000             176000                      ;EVERY PROTECT ON
20 10050 010230             MM.EC                       ;NO I/O VALIDITY TRAPS
21 10051 010230             MM.EC                       ;NO WRITE/DEFER TRAPS
22 10052 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 10062 102000 MM.01:      ADC 0,0
34 10063 040531             STA 0,MM.ES     ;CLR ERR SW
35 10064 001400             JMP 0,3         ;RETURN
36
37 10065 000067 MM.PL: MM.EN-MM.3A
38 10066 010123 MM.LC: MM.3A
```

```
!0129 N3MRT
01
02                ;EXECUTE ENTRY POINT
03                ;GET SCRATCH AREA
04 10067 020525 MM.02:  LDA 0,MM.ES             ;ERR  SWITCH
05 10070 101005         MOV 0,0,SNR             ;SKP IS NOT HAD ERROR
06 10071 000407         JMP MM.2A               ;ERR HAS OCCURRED
07                                              ;DON'T NEED TO GET SCRATCH
08 10072 102000         ADC 0,0
09 10073 040521         STA 0,MM.ES             ;CLR ERR SWITCH
10                      LCALL ASCRA             ;GET 1K SCRATCH
11 10074 100010         ASCRA=ASCRA*1B11+100010
12 10075 000520         JMP MM.X1               ;EXIT IF NONE AVAILABLE
13                MM.2L:  LCALL ESCRA           ;EXPAND SCRATCH BY 1K
14 10076 100030         ESCRA=ASCRA*1B11+100010
15 10077 000401         JMP MM.2A               ;NO MORE AVAILABLE
16
17                ;MOVE TESTS TO SCRATCH AREA
18 10100 030152 MM.2A:  LDA 2,SCRLO
19 10101 034765         LDA 3,MM.LC     ;FROM HERE
20 10102 141000         MOV 2,0
21 10103 024762         LDA 1,MM.PL
22 10104 123000         ADD 1,0
23 10105 040504         STA 0,MM.SA     ;SAVE STARTING ADDR
24 10106 124000         COM 1,1         ;-# OF WORDS TO MOVE
25 10107 021400 MM.L2:  LDA 0,0,3
26 10110 041000         STA 0,0,2
27 10111 151400         INC 2,2
28 10112 175400         INC 3,3
29 10113 125404         INC 1,1,SZR
30 10114 000773         JMP MM.L2       ;CONTINUE
31
32                ;NOW EXECUTE TESTS IN SCRATCH
33 10115 034152 MM.03:  LDA 3,SCRLO
34 10116 005400         JSR 0,3         ;GO TO SCRATCH
35 10117 020475         LDA 0,MM.ES     ;ERR SW
36 10120 101004         MOV 0,0,SZR     ;SKP IS ERR HAS OCCURRED
37 10121 000474         JMP MM.X1       ;RELEASE SCRATCH
38 10122 000500         JMP MM.X2       ;DON'T REL. SCRATCH
```

```
!0130 N3MRT
01
02                ;SCRATCH AREA HAS BEEN ASSIGNED, FILL AREA WITH BACKGND
03
04 10123 054454 MM.3A:  STA 3,MM.S3
05 10124 030465         LDA 2,MM.SA             ;FILL SCRATCH WITH MINUS 1
06 10125 034153         LDA 3,SCRHI
07 10126 126000         ADC 1,1
08 10127 045000         STA 1,0,2
09 10130 021000         LDA 0,0,2               ;CHECK  IT GOT THERE
10 10131 106434         SUBZ# 0,1,SZR
11 10132 006446         JSR @MM3ER              ;NOT -1,ERROR
12 10133 151400         INC 2,2
13 10134 156432         SUBZ# 2,3,SZC           ;DONE ALL?
14 10135 000771         JMP MM.3A+3             ;NOT YET
15
16                ;ALL SCRATCH IS FILLED, NOW TEST WITH ISZ
17 10136 030453 MM.3B:  LDA 2,MM.SA
18 10137 021000         LDA 0,0,2
19 10140 100014         COM# 0,0,SZR            ;CHECK LOC BEFORE ISZ'ING
20 10141 006440         JSR @MM3E1              ;NOT -1 BEFORE ISZ'NG
21 10142 011000         ISZ 0,2                 ;ISZ THIS LOCATION
22 10143 006437         JSR @MM3E2              ;ERROR,ISZ DIDN'T SKIP
23 10144 021000         LDA 0,0,2               ;GET CONTENTS
24 10145 101004         MOV 0,0,SZR             ;LOC CONTENTS = 0?
25 10146 006435         JSR @MM3E3              ;LOC NOT 0 AFTER ISZ
26 10147 015000         DSZ 0,2                 ;-1 TO LOC AGAIN
27 10150 000402         JMP .+2
28 10151 006433         JSR @MM3E4              ;DSZ SKIPPED-ERROR
29 10152 021000         LDA 0,0,2               ;CHECK CONTENTS
30 10153 100014         COM# 0,0,SZR            ;=-1?
31 10154 006431         JSR @MM3E5              ;LOC NOT -1 AFTER DSZ
32 10155 151400         INC 2,2
33 10156 156432         SUBZ# 2,3,SZC           ;DONE ALL?
34 10157 000760         JMP MM.3B+1             ;NOT YET
35
```

```
10131 N3MRT
01
02                ;NOW DO ISZ TEST IN REVERSE DIRECTION
03 10160 030153 MM.3C:  LDA 2,SCRHI                     ;STARTING ADDRESS
04 10161 034430         LDA 3,MM.SA                     ;FINISHING ADDRESS
05 10162 126000         ADC 1,1                         ;AC1=EXPECTED VALUE BEFORE ISZ
06 10163 021000         LDA 0,0,2                       ;CHECK LOC BEFORE DOING ISZ
07 10164 100014         COM# 0,0,SZR                    ;AC0=-1?
08 10165 006421         JSR @MM3E6                      ;LOC NOT -1 BEFORE ISZ
09 10166 011000         ISZ 0,2                         ;ISZ THIS LOCATION
10 10167 006420         JSR @MM3E7                      ;ISZ DIDN'T SKIP
11 10170 021000         LDA 0,0,2                       ;NOW CHECK LOC.
12 10171 101004         MOV 0,0,SZR                     ;LOC = 0?
13 10172 006416         JSR @MM3E8                      ;LOC NOT 0 AFTER ISZ
14 10173 133000         ADD 1,2                         ;DECREMENT AC2 BY 1
15 10174 172432         SUBZ# 3,2,SZC                   ;IS SCRHI<SCRLO?
16 10175 000766         JMP MM.3C+3                     ;NOT YET
17 10176 002401         JMP @MM.S3      ;RETURN TO MAIN TEST SECTION
18
19                ;TEST CONSTANTS
20 10177 000000 MM.S3:  0
21 10200 010223 MM3ER:  MM.ER
22 10201 010243 MM3E1:  MM.E1
23 10202 010247 MM3E2:  MM.E2
24 10203 010254 MM3E3:  MM.E3
25 10204 010260 MM3E4:  MM.E4
26 10205 010265 MM3E5:   MM.E5
27 10206 010271 MM3E6:   MM.E6
28 10207 010275 MM3E7:   MM.E7
29 10210 010302 MM3E8:   MM.E8
30 10211 000000 MM.SA: 0
31 10212 000000 MM.EN: 0
32 10213 000000 MM.SE: 0
33 10214 000000 MM.ES:  0       ;ERROR SWITCH
34
35                ;DONE ALL TESTING,RELEASE SCRATCH
36
37 10215 102000 MM.X1:  ADC 0,0
38 10216 040776         STA 0,MM.ES                     ;CLR ERR REL. SWITCH
39                      LCALL RSCRA                     ;RELEASE 1K SCRATCH
40 10217 100050         RSCRA-ASCRA*1B11+100010
41 10220 102001         ADC 0,0,SKP                     ;RET ALL RELEASED
42 10221 000774         JMP MM.X1                       ;RELEASE ALL
43                MM.X2:  LCALL RETRN                     ;RETURN TO LINKR
44 10222 100210         RETRN-ASCRA*1B11+100010
```

```
10132 N3MRT
01                ;PATTERN CHECK FOUND AN ERROR
02 10223 054770 MM.ER:  STA 3,MM.SE
03 10224 176400         SUB 3,3
04 10225 050471         STA 2,MMLOC                     ;SAVE LOC ADDR.
05 10226 054467         STA 3,MM.TK                     ;SET ERROR #
06                      LCALL ERROI
07 10227 100350         ERROI-ASCRA*1B11+100010
08 10230 000401 MM.EC:  JMP .+1
09 10231 020466         LDA 0,MMTXT
10                      LCALL ERRTX
11 10232 100170         ERRTX-ASCRA*1B11+100010
12 10233 020462         LDA 0,MM.TK                     ;ERR#
13 10234 024757         LDA 1,MM.SE                     ;ERROR CALL ADDR
14 10235 030461         LDA 2,MMLOC                     ;GET SAVED LOC
15                      LCALL ERROC
16 10236 100370         ERROC-ASCRA*1B11+100010
17 10237 000756         JMP MM.X1
18 10240 102400         SUB 0,0
19 10241 040753         STA 0,MM.ES     ;SET ERR OCCURRED SW
20 10242 000636         JMP MM.2A
21
22                ;FORWARD ISZ TST - LOC NOT -1 BEFORE ISZ
23 10243 054750 MM.E1:  STA 3,MM.SE
24 10244 126000         ADC 1,1
25 10245 176520         SUBZL 3,3
26 10246 000757         JMP MM.ER+2
27                ;FORWARD ISZ TST - ISZ DIDN'T SKIP
28 10247 054744 MM.E2:  STA 3,MM.SE
29 10250 021000         LDA 0,0,2       ;ACTUAL
30 10251 126400         SUB 1,1         ;EXPECTED = 0
31 10252 034434         LDA 3,MM.K2
32 10253 000752         JMP MM.ER+2
33
34                ;FORWARD ISZ-LOC NOT 0 AFTER ISZ
35 10254 054737 MM.E3:  STA 3,MM.SE
36 10255 126400         SUB 1,1
37 10256 034431         LDA 3,MM.K3
38 10257 000746         JMP MM.ER+2
39
40                ;DSZ TEST-DSZ SKIPPED
41 10260 054733 MM.E4:  STA 3,MM.SE
42 10261 021000         LDA 0,0,2       ;ACTUAL
43 10262 126000         ADC 1,1         ;EXPECTED = -1
44 10263 034425         LDA 3,MM.K4
45 10264 000741         JMP MM.ER+2
46
47                ;DSZ TEST-LOC NOT -1 AFTER DSZ
48 10265 054726 MM.E5: STA 3,MM.SE
49 10266 126000         ADC 1,1
50 10267 034422         LDA 3,MM.K5
51 10270 000735         JMP MM.ER+2
52
53                ;REVERSE ISZ TEST-LOC NOT -1 BEFORE ISZ
54 10271 054722 MM.E6:  STA 3,MM.SE
55 10272 126000         ADC 1,1
56 10273 034417         LDA 3,MM.K6
57 10274 000731         JMP MM.ER+2
58
```

```
!0133 N3MRT
01                ;REVERSE ISZ TEST-ISZ DIDN'T SKIP
02 10275 054716 MM.E7:   STA 3,MM.SE
03 10276 021000          LDA 0,0,2
04 10277 126400          SUB 1,1
05 10300 034413          LDA 3,MM.K7
06 10301 000724          JMP MM.ER+2
07
08                ;REVERSE ISZ TST-LOC NOT 0 AFTER ISZ
09 10302 054711 MM.E8:   STA 3,MM.SE
10 10303 126400          SUB 1,1
11 10304 034410          LDA 3,MMK10
12 10305 000720          JMP MM.EC-3
13
14 10306 000002 MM.K2: 2
15 10307 000003 MM.K3: 3
16 10310 000004 MM.K4: 4
17 10311 000005 MM.K5: 5
18 10312 000006 MM.K6: 6
19 10313 000007 MM.K7: 7
20 10314 000010 MMK10: 10
21 10315 000000 MM.TK: 0
22 10316 000000 MMLOC: 0
23 10317 010320 MMTXT:   .+1
24 10320 005215          .TXTE !<15><12>MM.TK<11>MM.SE<11>LOCATION!
```

```
!0134 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              %
08                      .MACRO XORA
09                      JSR XOR.0
10              %
11                      .MACRO XOR1
12                      JSR XOR.1
13              %
14                      .MACRO XOR2
15                      JSR XOR.2
16              %
17                      .MACRO .DIVU
18                      JSR DIVU
19              %
20                      .MACRO .MPYA
21                      JSR MPYA
22              %
23                      .MACRO RANDOM
24                      LCALL FRANG
25              %
26                      .MACRO SQRT
27                      JSR SQRT.
28              %
29                      .MACRO SQ
30                      JSR SQ.
31              %
32                      .MACRO .MPYU
33                      JSR MPYU
34              %
```

```
!0135 N3MRT
01                ;ARITH - DEFINE PARAMETERS TO LINKR
02                      NEXTT AT.00
03        010334 LMEML=.
04        000160        .LOC LPG0
05 00160 010337         AT.00
06        000161 LPG0=.
07        010334        .LOC LMEML
08 10334 000000         0       ;TEST PASS CTR
09 10335 000000         0       ;TEST ERROR CTR
10 10336 000000         0       ;INTERRUPT TIMEOUT SWITCH
11 10337 010357 AT.00:  AT.01
12 10340 010362         AT.02
13 10341 000000         0
14 10342 000000         0
15 10343 177777         -1
16 10344 176000         176000
17 10345 010522         AT.EC
18 10346 010522         AT.EC
19        010347        .TXTE (
20 10347 151101 ARITHMETIC TEST(
21        152311
22        046510
23        152305
24        141711
25        152240
26        051705
27        000324
28                ;SET NO SCRATCH ASSIGNED SWITCH
29 10357 102000 AT.01:  ADC 0,0
30 10360 040471         STA 0,AT.TK
31 10361 001400         JMP 0,3
32
```

```
!0136 N3MRT
01                ;EXECUTE ENTRY POINT GET SCRATCH IF NONE AVAIL
02
03 10362 010467 AT.02:  ISZ AT.TK       ;SKIP IS NO SCRATCH
04 10363 000467         JMP AT.03
05                      LCALL ASCRA     ;GET 1K SCRATCH
06 10364 100010         ASCRA-ASCRA*1B11+100010
07 10365 000453         JMP AT.05       ;EXIT NONE AVAILABLE
08 10366 102000         ADC 0,0
09 10367 040555         STA 0,AT.ES     ;SET NO ERR SWTCH
10                      LCALL ESCRA     ;TRY TO GET 1K MORE
11 10370 100030         ESCRA-ASCRA*1B11+100010
12 10371 000401         JMP .+1         ;BUT USE 1K IF NO MORE AVAIL
13 10372 030557         LDA 2,AT.37
14 10373 150400         NEG 2,2         ;TRY RANDOM BETWEEN LIM 37 TRYS
15                AT.2L: LCALL ARANG
16 10374 100270         ARANG-ASCRA*1B11+100010
17 10375 024153         LDA 1,SCRHI
18 10376 103000         ADD 0,0
19 10377 101220         MOVZR 0,0       ;CLR BIT 0
20 10400 122032         ADCZ# 1,0,SZC
21 10401 000411         JMP AT.2A       ;GRTR THAN HILIM
22 10402 034543         LDA 3,AT.PL
23 10403 166400         SUB 3,1         ;ENOUGH ROOM
24 10404 122032         ADCZ# 1,0,SZC   ;TO RELOCATE UP
25 10405 162400         SUB 3,0         ;NO LOWER NUMBER
26 10406 024152         LDA 1,SCRLO
27 10407 125400         INC 1,1
28 10410 122432         SUBZ# 1,0,SZC
29 10411 000404         JMP AT.2B
30 10412 151404 AT.2A:  INC 2,2,SZR
31 10413 000761         JMP AT.2L
32 10414 121000         MOV 1,0
```

```
!0137 N3MRT
01                ;MOVE TESTS TO SELECTED AREA
02
03 10415 111000 AT.28:  MOV 0,2          ;ADJUST SCRHI
04 10416 020527         LDA 0,AT.PL      ;SO THAT TESTS
05 10417 024153         LDA 1,SCRHI      ;WILL FIT
06 10420 106400         SUB 0,1          ;INTO SCRATCH
07 10421 132032         ADCZ# 1,2,SZC
08 10422 131000         MOV 1,2
09 10423 050523         STA 2,AT.LC      ;START ADRS IN SCRATCH
10 10424 024521 AT.RL:  LDA 1,AT.PL
11 10425 124000         COM 1,1          ;-# WORDS TO MOVE
12 10426 030520         LDA 2,AT.LC      ;TO
13 10427 034520         LDA 3,AT.BG      ;FROM
14 10430 021400 AT.L2:  LDA 0,0,3        ;MOVE LOOP
15 10431 041000         STA 0,0,2
16 10432 151400         INC 2,2
17 10433 175400         INC 3,3
18 10434 125404         INC 1,1,SZR
19 10435 000773         JMP AT.L2
20 10436 050512         STA 2,AT.EN
21 10437 000413         JMP AT.03
22 10440 102000 AT.05:  ADC 0,0
23 10441 040410         STA 0,AT.TK
24                AT.XI:  LCALL RETRN
25 10442 100210         RETRN-ASCRA*1B11+100010
26                ;MAP OPTION DOES NOT EXIST GO DIRECT
27 10443 020503 AT.GD:  LDA 0,AT.LC
28 10444 040507         STA 0,AT.LA
29 10445 020152         LDA 0,SCRLO
30 10446 040504         STA 0,AT.LO
31 10447 040505         STA 0,AT.LP
32 10450 002503         JMP @AT.LA
33 10451 000000 AT.TK: 0
```

```
!0138 N3MRT
01                ;TESTS HAVE BEEN MOVED TO SCRATCH
02                ;AT.03 SELECTS LOGICAL PAGE ASSIGNMENT
03                ;AND INITIATES TEST VIA GSCRA
04
05 10452 020472 AT.03:  LDA 0,AT.ES      ;GET ERR SWITCH
06 10453 101005         MOV 0,0,SNR      ;SKIP IS NO ERR
07 10454 000416         JMP AT.04        ;USE PREV ASSIGN
08                      LCALL ARANG
09 10455 100270         ARANG-ASCRA*1B11+100010
10 10456 105000         MOV 0,1
11 10457 030472         LDA 2,AT.37
12                      LCALL ADIVI
13 10460 100310         ADIVI-ASCRA*1B11+100010
14 10461 040473         STA 0,AT.LP
15 10462 024464         LDA 1,AT.LC
16 10463 030152         LDA 2,SCRLO
17 10464 146400         SUB 2,1
18 10465 111300         MOVS 0,2
19 10466 153120         ADDZL 2,2
20 10467 050463         STA 2,AT.LO
21 10470 133000         ADD 1,2
22 10471 050462         STA 2,AT.LA
23 10472 020462 AT.04:  LDA 0,AT.LP      ;REMAP SCR TO HERE
24 10473 030460         LDA 2,AT.LA      ;STARTING LOGICAL ADRS
25 10474 024422         LDA 1,ATERR      ;1=ERROR RET ADDRS
26                      LCALL GSCRA      ;GO TO SCRATCH
27 10475 100070         GSCRA-ASCRA*1B11+100010
28 10476 000745         JMP AT.GD        ;COULDNT REMAP GO DIRECT
29                ;RETURN TO NEXT LOC PASS COMPLETE NO ERRS
30 10477 020230         LDA 0,SWREG      ;CHECK IF REL. SCRATCH
31 10500 103123         ADDZL 0,0,SNC    ;SKP IS NOT REL
32 10501 000406         JMP .+6          ;RELEASE SCRATCH
33 10502 020442         LDA 0,AT.ES      ;CHK FOR PREV ERR
34 10503 101004         MOV 0,0,SZR      ;SKP ON PREV ERR
35 10504 000403         JMP .+3          ;RELEASE ASSIGN
36 10505 040744         STA 0,AT.TK
37 10506 000734         JMP AT.XI
38                      LCALL ARANG      ;IF BITS 0 AND 1=1
39 10507 100270         ARANG-ASCRA*1B11+100010
40 10510 103043         ADDO 0,0,SNC     ;RELEASE AND REMAP
41 10511 103003         ADD 0,0,SNC      ;IF EITHER =0 MAKE
42 10512 000773         JMP .-5          ;NEXT PASS SELECT NEW LP
43                AT.4A:  LCALL RSCRA
44 10513 100050         RSCRA-ASCRA*1B11+100010
45 10514 000724         JMP AT.05        ;SET NO SCRATCH
46 10515 000776         JMP AT.4A
```

```
!0139 N3MRT
01                ;ERROR IN TEST DURING EXECUTION
02
03 10516 010517 ATERR:  ATERR+1
04 10517 054424         STA 3,ATS03       ;PRINT ERR HEADERS
05                      LCALL ERROI
06 10520 100350         ERROI-ASCRA*1B11+100010
07 10521 000401         JMP .+1
08 10522 020433 AT.EC:  LDA 0,ATTX1
09                      LCALL ERRTX
10 10523 100170         ERRTX-ASCRA*1B11+100010
11 10524 020422         LDA 0,AT.LC       ;FOLLOW UP WITH
12 10525 024425         LDA 1,AT.LO       ;TEST RELOC INFO
13 10526 030425         LDA 2,AT.LA
14                      LCALL ERROC
15 10527 100370         ERROC-ASCRA*1B11+100010
16 10530 000401         JMP .+1
17                      LCALL ERPAC
18 10531 100250         ERPAC-ASCRA*1B11+100010
19 10532 020411         LDA 0,ATS03       ;CONTINUE TYPE
20 10533 024420         LDA 1,AT.LA
21 10534 030413         LDA 2,AT.BG
22                      LCALL ERPAD
23 10535 100230         ERPAD-ASCRA*1B11+100010
24 10536 000755         JMP AT.4A         ;SW0=1 RELEASE SCR
25 10537 102400         SUB 0,0
26 10540 040404         STA 0,AT.ES       ;SET ERR SW
27 10541 040710         STA 0,AT.TK       ;CLR TEST K
28 10542 000700         JMP AT.XI         ;RETURN TO LINKR
29
```

```
!0140 N3MRT
01 10543 000000 ATS03: 0
02 10544 000000 AT.ES:  0
03 10545 001361 AT.PL:  ATEND-MS1
04 10546 000000 AT.LC:  0
05 10547 010604 AT.BG:  MS1
06 10550 000000 AT.EN:  0
07 10551 000037 AT.37:  37
08 10552 000000 AT.LO:  0
09 10553 000000 AT.LA:  0
10 10554 000000 AT.LP:  0
11 10555 010556 ATTX1:  .+1
12 10556 005215         .TXTE (<15><12>AT.LC     AT.LO    AT.LA(
13 10570 010571 ATTX2:  .+1
14 10571 005215         .TXTE (<15><12>ATS03     AT.LP    E ADRS(
```

```
!0141 N3MRT
01                MS1:    SETUP           ;MISC TEST OF NEG/COM
02                        LCALL SETUL
03 10604 100410           SETUL-ASCRA*1B11+100010
04                        RANDOM
05                        LCALL FRANG
06 10605 100450           FRANG-ASCRA*1B11+100010
07 10606 104700           NEGS 0,1
08 10607 130342           COMOS 1,2,SZC
09 10610 142014           ADC# 2,0,SZR
10                        ERROR
11 10611 004402           JSR .+2
12 10612 000402           JMP .+2
13                        LCALL ERRET
14 10613 100470           ERRET-ASCRA*1B11+100010
15                        LOOP
16                        LCALL LLOOP
17 10614 100430           LLOOP-ASCRA*1B11+100010
18
19                ;MS2:
20                        SETUP           ;MISC TEST OF INC SWAPPED.
21                        LCALL SETUL
22 10615 100410           SETUL-ASCRA*1B11+100010
23                        RANDOM
24                        LCALL FRANG
25 10616 100450           FRANG-ASCRA*1B11+100010
26 10617 111700           INCS 0,2
27 10620 145323           MOVZS 2,1,SNC
28 10621 106314           ADCS# 0,1,SZR
29                        ERROR
30 10622 004402           JSR .+2
31 10623 000402           JMP .+2
32                        LCALL ERRET
33 10624 100470           ERRET-ASCRA*1B11+100010
34                        LOOP
35                        LCALL LLOOP
36 10625 100430           LLOOP-ASCRA*1B11+100010
37
```

```
!0142 N3MRT
01                ;MS3:
02                        SETUP           ;MISC NEG TEST
03                        LCALL SETUL
04 10626 100410           SETUL-ASCRA*1B11+100010
05 10627 102400           SUB 0,0
06 10630 100644           NEGOR 0,0,SZR
07                        ERROR
08 10631 004402           JSR .+2
09 10632 000402           JMP .+2
10                        LCALL ERRET
11 10633 100470           ERRET-ASCRA*1B11+100010
12 10634 100664           NEGCR 0,0,SZR
13                        ERROR
14 10635 004402           JSR .+2
15 10636 000402           JMP .+2
16                        LCALL ERRET
17 10637 100470           ERRET-ASCRA*1B11+100010
18 10640 100664           NEGCR 0,0,SZR
19                        ERROR
20 10641 004402           JSR .+2
21 10642 000402           JMP .+2
22                        LCALL ERRET
23 10643 100470           ERRET-ASCRA*1B11+100010
24                        LOOP
25                        LCALL LLOOP
26 10644 100430           LLOOP-ASCRA*1B11+100010
27
```

```
!0143 N3MRT
01                ;ROT1:
02                        SETUP           ;TEST ROTATE LEFT
03                        LCALL SETUL
04 10645 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM
06                        LCALL FRANG
07 10646 100450           FRANG-ASCRA*1B11+100010
08 10647 105000           MOV 0,1
09 10650 101100           MOVL 0,0
10 10651 111100           MOVL 0,2
11 10652 151100           MOVL 2,2
12 10653 151100           MOVL 2,2
13 10654 155100           MOVL 2,3
14 10655 175100           MOVL 3,3
15 10656 175100           MOVL 3,3
16 10657 161100           MOVL 3,0
17 10660 101100           MOVL 0,0
18 10661 101100           MOVL 0,0
19 10662 101100           MOVL 0,0
20 10663 101100           MOVL 0,0
21 10664 101100           MOVL 0,0
22 10665 101100           MOVL 0,0
23 10666 101100           MOVL 0,0
24 10667 101100           MOVL 0,0
25 10670 101100           MOVL 0,0
26 10671 106414           SUB# 0,1,SZR
27                        ERROR
28 10672 004402           JSR .+2
29 10673 000402           JMP .+2
30                        LCALL ERRET
31 10674 100470           ERRET=ASCRA*1B11+100010
32                        LOOP
33                        LCALL LLOOP
34 10675 100430           LLOOP=ASCRA*1B11+100010
```

```
!0144 N3MRT
01                ;ROT2:
02                        SETUP           ;TEST ROTATE RIGHT
03                        LCALL SETUL
04 10676 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM
06                        LCALL FRANG
07 10677 100450           FRANG-ASCRA*1B11+100010
08 10700 131000           MOV 1,2
09 10701 121200           MOVR 1,0
10 10702 101200           MOVR 0,0
11 10703 101200           MOVR 0,0
12 10704 101200           MOVR 0,0
13 10705 101200           MOVR 0,0
14 10706 101200           MOVR 0,0
15 10707 101200           MOVR 0,0
16 10710 101200           MOVR 0,0
17 10711 101200           MOVR 0,0
18 10712 101200           MOVR 0,0
19 10713 105200           MOVR 0,1
20 10714 125200           MOVR 1,1
21 10715 135200           MOVR 1,3
22 10716 175200           MOVR 3,3
23 10717 175200           MOVR 3,3
24 10720 161200           MOVR 3,0
25 10721 101200           MOVR 0,0
26 10722 112414           SUB# 0,2,SZR
27                        ERROR
28 10723 004402           JSR .+2
29 10724 000402           JMP .+2
30                        LCALL ERRET
31 10725 100470           ERRET=ASCRA*1B11+100010
32                        LOOP
33                        LCALL LLOOP
34 10726 100430           LLOOP=ASCRA*1B11+100010
```

```
!0145 N3MRT
01                ;AND0:
02                        SETUP             ;ANY NUMBER ANDED WITH
03                        LCALL SETUL
04 10727 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM            ;ITSELF SHOULD NOT
06                        LCALL FRANG
07 10730 100450           FRANG-ASCRA*1B11+100010
08 10731 131000           MOV 1,2           ;BE CHANGED.
09 10732 127700           ANDS 1,1
10 10733 127700           ANDS 1,1
11 10734 127700           ANDS 1,1
12 10735 127700           ANDS 1,1
13 10736 127700           ANDS 1,1
14 10737 135300           MOVS 1,3
15 10740 177700           ANDS 3,3
16 10741 177700           ANDS 3,3
17 10742 161300           MOVS 3,0
18 10743 103700           ANDS 0,0
19 10744 112414           SUB# 0,2,SZR
20                        ERROR
21 10745 004402           JSR .+2
22 10746 000402           JMP .+2
23                        LCALL ERRET
24 10747 100470           ERRET-ASCRA*1B11+100010
25                        LOOP
26                        LCALL LLOOP
27 10750 100430           LLOOP-ASCRA*1B11+100010
28
29                ;AND1:
30                        SETUP             ;A NUMBER ANDED WITH ITS
31                        LCALL SETUL
32 10751 100410           SETUL-ASCRA*1B11+100010
33                        RANDOM            ;COMPLIMENT SHOULD
34                        LCALL FRANG
35 10752 100450           FRANG-ASCRA*1B11+100010
36 10753 104042           COMO 0,1,SZC
37 10754 123704           ANDS 1,0,SZR      ;PRODUCE ZERO RESULT.
38                        ERROR
39 10755 004402           JSR .+2
40 10756 000402           JMP .+2
41                        LCALL ERRET
42 10757 100470           ERRET-ASCRA*1B11+100010
43                        LOOP
44                        LCALL LLOOP
45 10760 100430           LLOOP-ASCRA*1B11+100010
46
```

```
!0146 N3MRT
01                ;AND3:
02                        SETUP             ;PERFORM A AND INSTRUCTION
03                        LCALL SETUL
04 10761 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM            ;WITH THE RESULT IN AC2.
06                        LCALL FRANG
07 10762 100450           FRANG-ASCRA*1B11+100010
08 10763 176620           SUBZR 3,3         ;SIMULATE THE AND VIA
09 10764 131000           MOV 1,2           ;LOOKING FOR ADDER CARRY.
10 10765 113400           AND 0,2
11 10766 101113 AND3L:    MOVL# 0,0,SNC
12 10767 000403           JMP .+3
13 10770 125112           MOVL# 1,1,SZC
14 10771 101141           MOVOL 0,0,SKP
15 10772 101121           MOVZL 0,0,SKP
16 10773 125141           MOVOL 1,1,SKP
17 10774 125120           MOVZL 1,1
18 10775 175224           MOVZR 3,3,SZR
19 10776 000770           JMP AND3L
20 10777 106415           SUB# 0,1,SNR      ;CHECK IF AC0-1 ARE
21 11000 132414           SUB# 1,2,SZR      ;THE SAME AND IF THEY
22                        ERROR             ;ARGEE WITH INST.
23 11001 004402           JSR .+2
24 11002 000402           JMP .+2
25                        LCALL ERRET
26 11003 100470           ERRET-ASCRA*1B11+100010
27                        LOOP
28                        LCALL LLOOP
29 11004 100430           LLOOP-ASCRA*1B11+100010
```

```
!0147 N3MRT
01                ;TEST PROCESSOR VIA EXCLUSIVE OR ROUTINES.
02
03                ;X1:
04                        SETUP           ;C(AC1) IS SAVED IN C(AC2).
05                        LCALL SETUL
06 11005 100410           SETUL-ASCRA*1B11+100010
07                        RANDOM          ;AC0 IS EXCLUSIVE ORED
08                        LCALL FRANG
09 11006 100450           FRANG-ASCRA*1B11+100010
10 11007 131000           MOV 1,2         ;WITH AC1 TWICE. THE SECOND
11                        CALL            ;EXCLUSIVE OR SHOULD
12                        XORA            ;RESTORE AC1 TO ITS
13 11010 004467           JSR XOR.0
14                        CALL            ;ORIGIONAL CONTENTS.
15                        XORA
16 11011 004466           JSR XOR.0
17 11012 132414           SUB# 1,2,SZR
18                        ERROR
19 11013 004402           JSR .+2
20 11014 000402           JMP .+2
21                        LCALL ERRET
22 11015 100470           ERRET-ASCRA*1B11+100010
23                        LOOP
24                        LCALL LLOOP
25 11016 100430           LLOOP-ASCRA*1B11+100010
26
27                ;X2:
28                        SETUP           ;THE FIRST EXCLUSIVE OR
29                        LCALL SETUL
30 11017 100410           SETUL-ASCRA*1B11+100010
31                        RANDOM          ;ROUTINE EXCHANGES THE
32                        LCALL FRANG
33 11020 100450           FRANG-ASCRA*1B11+100010
34                        CALL            ;CONTENTS
35                        XOR2            ;OF AC0 AND AC1, IT ALSO
36 11021 004474           JSR XOR.2
37                        CALL            ;FORMS THE EXCLUSIVE OR
38                        XOR1            ;IN AC2. THE SECOND EXCLUSIVE
39 11022 004463           JSR XOR.1
40 11023 112414           SUB# 0,2,SZR    ;OR FORMS THE RESULT
41                        ERROR           ;OF AC0-AC1 IN AC0.
42 11024 004402           JSR .+2
43 11025 000402           JMP .+2
44                        LCALL ERRET
45 11026 100470           ERRET-ASCRA*1B11+100010
46                        LOOP
47                        LCALL LLOOP
48 11027 100430           LLOOP-ASCRA*1B11+100010
```

```
!0148 N3MRT
01                ;X3:
02                        SETUP           ;SAVE C(AC1) NEGATED IN
03                        LCALL SETUL
04 11030 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM          ;C(AC2). EXCLUSIVE OR C(AC0) TO
06                        LCALL FRANG
07 11031 100450           FRANG-ASCRA*1B11+100010
08 11032 130400           NEG 1,2         ;C(AC1). EXCLUSIVE OR THE
09                        CALL            ;RESULT BACK TO AC0.
10                        XORA            ;CHECK VIA ADDITION TO
11 11033 004444           JSR XOR.0
12                        CALL            ;COMPLIMENT OF ORIGIONAL
13                        XOR1            ;NUMBER.
14 11034 004451           JSR XOR.1
15 11035 113014           ADD# 0,2,SZR
16                        ERROR
17 11036 004402           JSR .+2
18 11037 000402           JMP .+2
19                        LCALL ERRET
20 11040 100470           ERRET-ASCRA*1B11+100010
21                        LOOP
22                        LCALL LLOOP
23 11041 100430           LLOOP-ASCRA*1B11+100010
24
25                ;X4:
26                        SETUP           ;EXCLUSIVE OR C(AC0) TO
27                        LCALL SETUL
28 11042 100410           SETUL-ASCRA*1B11+100010
29                        RANDOM          ;ALL ZEROS IN C(AC1).
30                        LCALL FRANG
31 11043 100450           FRANG-ASCRA*1B11+100010
32 11044 126400           SUB 1,1
33                        CALL
34                        XORA
35 11045 004432           JSR XOR.0
36 11046 106414           SUB# 0,1,SZR
37                        ERROR
38 11047 004402           JSR .+2
39 11050 000402           JMP .+2
40                        LCALL ERRET
41 11051 100470           ERRET-ASCRA*1B11+100010
42                        LOOP
43                        LCALL LLOOP
44 11052 100430           LLOOP-ASCRA*1B11+100010
```

```
!0149 N3MRT
01                ;X5:
02                        SETUP           ;EXCLUSIVE OR C(AC1) TO
03                        LCALL SETUL
04 11053 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM          ;ALL ZEROS IN C(AC0).
06                        LCALL FRANG
07 11054 100450           FRANG-ASCRA*1B11+100010
08 11055 102400           SUB 0,0
09                        CALL
10                        XOR1
11 11056 004427           JSR XOR.1
12 11057 106414           SUB# 0,1,SZR
13                        ERROR
14 11060 004402           JSR .+2
15 11061 000402           JMP .+2
16                        LCALL ERRET
17 11062 100470           ERRET-ASCRA*1B11+100010
18                        LOOP
19                        LCALL LLOOP
20 11063 100430           LLOOP-ASCRA*1B11+100010
21
22                ;X6:
23                        SETUP           ;EXCLUSIVE OR C(AC1) TO
24                        LCALL SETUL
25 11064 100410           SETUL-ASCRA*1B11+100010
26                        RANDOM          ;(-1) IN C(AC0).
27                        LCALL FRANG
28 11065 100450           FRANG-ASCRA*1B11+100010
29 11066 102000           ADC 0,0         ;THE COMPLIMENT OF
30                        CALL            ;THIS RESULT SHOULD
31                        XOR1            ;EQUAL C(AC1).
32 11067 004416           JSR XOR.1
33 11070 110000           COM 0,2
34 11071 146414           SUB# 2,1,SZR
35                        ERROR
36 11072 004402           JSR .+2
37 11073 000402           JMP .+2
38                        LCALL ERRET
39 11074 100470           ERRET-ASCRA*1B11+100010
40                        LOOP
41                        LCALL LLOOP
42 11075 100430           LLOOP-ASCRA*1B11+100010
43 11076 000436           JMP X7
```

```
!0150 N3MRT
01 11077 054433 XOR.0:  STA 3,XOR.4
02 11100 135000         MOV 1,3         ;EXCLUSIVE OR C(AC0),C(AC1).
03 11101 117520         ANDZL 0,3       ;RESULT IS IN C(AC1).
04 11102 107000         ADD 0,1
05 11103 166400         SUB 3,1
06 11104 002426         JMP @XOR.4
07
08 11105 054425 XOR.1:  STA 3,XOR.4
09 11106 135000         MOV 1,3         ;EXCLUSIVE OR C(AC0),C(AC1).
10 11107 117400         AND 0,3         ;RESULT IS IN C(AC0).
11 11110 174000         COM 3,3
12 11111 163400         AND 3,0
13 11112 123000         ADD 1,0
14 11113 163400         AND 3,0
15 11114 002416         JMP @XOR.4
16
17 11115 054415 XOR.2:  STA 3,XOR.4
18 11116 034415         LDA 3,XM20      ;EXCLUSIVE OR C(AC0),C(AC1).
19 11117 054412         STA 3,XORTEM    ;RESULT IN C(AC2).
20 11120 115000         MOV 0,3         ;THE CONTENTS OF AC0 AND
21 11121 137200         ADDR 1,3        ;AC1 ARE EXCHANGED.
22 11122 151200         MOVR 2,2
23 11123 101220         MOVZR 0,0
24 11124 125200         MOVR 1,1
25 11125 103200         ADDR 0,0
26 11126 010403         ISZ XORTEM
27 11127 000771         JMP XOR.2+3
28 11130 002402         JMP @XOR.4
29 11131 000000 XORTEM: 0
30 11132 000000 XOR.4: 0
31 11133 177760 XM20: -20
```

```
!0151 N3MRT
01                X7:     SETUP           ;C(AC1) IS SET EQUAL TO
02                        LCALL SETUL
03 11134 100410           SETUL-ASCRA*1B11+100010
04                        RANDOM          ;C(AC0). C(AC0) AND C(AC1)
05                        LCALL FRANG
06 11135 100450           FRANG-ASCRA*1B11+100010
07 11136 105000           MOV 0,1         ;ARE EXCLUSIVE ORED WITH
08                        CALL            ;THE RESULT GOING TO AC2.
09                        XOR2
10 11137 004756           JSR XOR.2
11 11140 151004           MOV 2,2,SZR
12                        ERROR
13 11141 004402           JSR .+2
14 11142 000402           JMP .+2
15                        LCALL ERRET
16 11143 100470           ERRET-ASCRA*1B11+100010
17                        LOOP
18                        LCALL LLOOP
19 11144 100430           LLOOP-ASCRA*1B11+100010
20
21                ;X8:
22                        SETUP           ;C(AC0) IS SET TO THE
23                        LCALL SETUL
24 11145 100410           SETUL-ASCRA*1B11+100010
25                        RANDOM          ;COMPLIMENT OF C(AC1). THE
26                        LCALL FRANG
27 11146 100450           FRANG-ASCRA*1B11+100010
28 11147 120000           COM 1,0         ;RESULT OF A EXCLUSIVE OR
29                        CALL            ;SHOULD BE ALL BITS
30                        XOR2            ;SET (-1).
31 11150 004745           JSR XOR.2
32 11151 150014           COM# 2,2,SZR
33                        ERROR
34 11152 004402           JSR .+2
35 11153 000402           JMP .+2
36                        LCALL ERRET
37 11154 100470           ERRET-ASCRA*1B11+100010
38                        LOOP
39                        LCALL LLOOP
40 11155 100430           LLOOP-ASCRA*1B11+100010
```

```
!0152 N3MRT
01                ;X9:
02                        SETUP           ;EXCLUSIVE OR ALL ONES
03                        LCALL SETUL
04 11156 100410           SETUL-ASCRA*1B11+100010
05 11157 102300           ADCS 0,0        ;TO ALL ONES. THE
06 11160 126000           ADC 1,1         ;RESULT SHOULD BE
07                        CALL            ;ALL ZEROS.
08                        XORA
09 11161 004716           JSR XOR.0
10 11162 127704           ANDS 1,1,SZR
11                        ERROR
12 11163 004402           JSR .+2
13 11164 000402           JMP .+2
14                        LCALL ERRET
15 11165 100470           ERRET-ASCRA*1B11+100010
16                        LOOP
17                        LCALL LLOOP
18 11166 100430           LLOOP-ASCRA*1B11+100010
19
20                ;X10:
21                        SETUP           ;EXCLUSIVE OR ALL ZEROS
22                        LCALL SETUL
23 11167 100410           SETUL-ASCRA*1B11+100010
24 11170 102700           SUBS 0,0        ;TO ALL ZEROS. THE
25 11171 126400           SUB 1,1         ;RESULT SHOULD BE
26                        CALL            ;ALL ZEROS IN C(AC1).
27                        XORA
28 11172 004705           JSR XOR.0
29 11173 107314           ADDS# 0,1,SZR
30                        ERROR
31 11174 004402           JSR .+2
32 11175 000402           JMP .+2
33                        LCALL ERRET
34 11176 100470           ERRET-ASCRA*1B11+100010
35                        LOOP
36                        LCALL LLOOP
37 11177 100430           LLOOP-ASCRA*1B11+100010
```

```
!0153 N3MRT
01                ;X11:
02                        SETUP             ;FORM EXCLUSIVE OF FUNCTION
03                        LCALL SETUL
04 11200 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM            ;IN C(AC2).
06                        LCALL FRANG
07 11201 100450           FRANG-ASCRA*1B11+100010
08                        CALL              ;CALL ANOTHER EXCLUSIVE
09                        XOR2              ;OR FUNCTION SEVEN TIMES.
10 11202 004713           JSR XOR.2
11                        CALL              ;THE RESULT SHOULD BE THE
12                        XORA              ;SAME AS THE FIRST XORA
13 11203 004674           JSR XOR.0
14                        CALL
15                        XORA
16 11204 004673           JSR XOR.0
17                        CALL
18                        XORA
19 11205 004672           JSR XOR.0
20                        CALL
21                        XORA
22 11206 004671           JSR XOR.0
23                        CALL
24                        XORA
25 11207 004670           JSR XOR.0
26                        CALL
27                        XORA
28 11210 004667           JSR XOR.0
29                        CALL
30                        XORA
31 11211 004666           JSR XOR.0
32 11212 132414           SUB# 1,2,SZR
33                        ERROR
34 11213 004402           JSR .+2
35 11214 000402           JMP .+2
36                        LCALL ERRET
37 11215 100470           ERRET-ASCRA*1B11+100010
38                        LOOP
39                        LCALL LLOOP
40 11216 100430           LLOOP-ASCRA*1B11+100010
41
42 11217 000402           JMP A1
43 11220 000657 XOR.L: JMP XOR.0
```

```
!0154 N3MRT
01                A1:     SETUP             ;SAVE C(AC2) IN C(AC1).
02                        LCALL SETUL
03 11221 100410           SETUL-ASCRA*1B11+100010
04                        RANDOM            ;ADD AND ADD C(AC0) TO
05                        LCALL FRANG
06 11222 100450           FRANG-ASCRA*1B11+100010
07 11223 145000           MOV 2,1           ;C(AC2). THE VALUE IN
08 11224 112400           SUB 0,2           ;AC2 SHOULD NOT BE
09 11225 113000           ADD 0,2           ;CHANGED.
10 11226 146414           SUB# 2,1,SZR
11                        ERROR
12 11227 004402           JSR .+2
13 11230 000402           JMP .+2
14                        LCALL ERRET
15 11231 100470           ERRET-ASCRA*1B11+100010
16                        LOOP
17                        LCALL LLOOP
18 11232 100430           LLOOP-ASCRA*1B11+100010
19
20                ;A2:
21                        SETUP
22                        LCALL SETUL
23 11233 100410           SETUL-ASCRA*1B11+100010
24                        RANDOM
25                        LCALL FRANG
26 11234 100450           FRANG-ASCRA*1B11+100010
27 11235 102000           ADC 0,0
28 11236 123000           ADD 1,0
29 11237 111400           INC 0,2
30 11240 146414           SUB# 2,1,SZR
31                        ERROR
32 11241 004402           JSR .+2
33 11242 000402           JMP .+2
34                        LCALL ERRET
35 11243 100470           ERRET-ASCRA*1B11+100010
36                        LOOP
37                        LCALL LLOOP
38 11244 100430           LLOOP-ASCRA*1B11+100010
```

```
;0155 N3MRT
01                ;A3:
02                        SETUP           ;INCREMENT THE VALUE IN
03                        LCALL SETUL
04 11245 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM          ;AC1 AND ADD THAT VALUE
06                        LCALL FRANG
07 11246 100450           FRANG-ASCRA*1B11+100010
08 11247 131460           INCC 1,2        ;TO (-1). THE RESULT
09 11250 102360           ADCCS 0,0       ;SHOULD BE THE ORIGIONAL
10 11251 143060           ADDC 2,0        ;NUMBER.
11 11252 106414           SUB# 0,1,SZR
12                        ERROR
13 11253 004402           JSR .+2
14 11254 000402           JMP .+2
15                        LCALL ERRET
16 11255 100470           ERRET-ASCRA*1B11+100010
17                        LOOP
18                        LCALL LLOOP
19 11256 100430           LLOOP-ASCRA*1B11+100010
20                ;A4:
21                        SETUP           ;SAVE THE C(AC1) IN C(AC0).
22                        LCALL SETUL
23 11257 100410           SETUL-ASCRA*1B11+100010
24                        RANDOM          ;A "ADDR" INSTRUCTION SHOULD
25                        LCALL FRANG
26 11260 100450           FRANG-ASCRA*1B11+100010
27 11261 121020           MOVZ 1,0        ;NOT CHANGE THE VALUE OF
28 11262 127200           ADDR 1,1        ;THE AC.
29 11263 127200           ADDR 1,1
30 11264 127200           ADDR 1,1
31 11265 127200           ADDR 1,1
32 11266 127200           ADDR 1,1
33 11267 127200           ADDR 1,1
34 11270 127200           ADDR 1,1
35 11271 127200           ADDR 1,1
36 11272 127200           ADDR 1,1
37 11273 127200           ADDR 1,1
38 11274 127200           ADDR 1,1
39 11275 127200           ADDR 1,1
40 11276 127200           ADDR 1,1
41 11277 127200           ADDR 1,1
42 11300 106454           SUB0# 0,1,SZR
43                        ERROR
44 11301 004402           JSR .+2
45 11302 000402           JMP .+2
46                        LCALL ERRET
47 11303 100470           ERRET-ASCRA*1B11+100010
48                        LOOP
49                        LCALL LLOOP
50 11304 100430           LLOOP-ASCRA*1B11+100010
```

```
;0156 N3MRT
01                ;A5:
02                        SETUP           ;THE RANDOM NUMBER IN
03                        LCALL SETUL
04 11305 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM          ;C(AC0) IS INCREMENTED VIA "INC"
06                        LCALL FRANG
07 11306 100450           FRANG-ASCRA*1B11+100010
08 11307 115400           INC 0,3         ;AND DECREMENTED VIA "ADD".
09 11310 126000           ADC 1,1         ;THE FINAL RESULT IN C(AC2)
10 11311 137000           ADD 1,3         ;SHOULD BE EQUAL TO THE
11 11312 175400           INC 3,3         ;ORIGIONAL NUMBER IN C(AC0).
12 11313 137000           ADD 1,3
13 11314 171400           INC 3,2
14 11315 133000           ADD 1,2
15 11316 151400           INC 2,2
16 11317 133000           ADD 1,2
17 11320 142414           SUB# 2,0,SZR
18                        ERROR
19 11321 004402           JSR .+2
20 11322 000402           JMP .+2
21                        LCALL ERRET
22 11323 100470           ERRET-ASCRA*1B11+100010
23                        LOOP
24                        LCALL LLOOP
25 11324 100430           LLOOP-ASCRA*1B11+100010
```

```
!0157 N3MRT
01                ;A6:
02                        SETUP             ;THE SUM OF AC0=1 IS
03                        LCALL SETUL
04 11325 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM            ;CHECKED WITH THE SIMULATED
06                        LCALL FRANG
07 11326 100450           FRANG-ASCRA*1B11+100010
08 11327 135000           MOV 1,3           ;SUM.
09 11330 117000           ADD 0,3
10 11331 054424           STA 3,ADDTEM      ;SUM VIA ADD INSTRUCTION.
11 11332 131000 A6L:      MOV 1,2
12                        CALL
13 11333 004665           JSR XOR.L                 ;SIMULATE THE ADD VIA
14 11334 143524           ANDZL 2,0,SZR     ;EXCLUSIVE OR. C(AC2)=
15 11335 000775           JMP A6L           ;RIPPLE CARRY,C(AC1)=RESULT.
16 11336 020417           LDA 0,ADDTEM
17 11337 122414           SUB# 1,0,SZR
18                        ERROR
19 11340 004402           JSR .+2
20 11341 000402           JMP .+2
21                        LCALL ERRET
22 11342 100470           ERRET-ASCRA*1B11+100010
23                        LOOP
24                        LCALL LLOOP
25 11343 100430           LLOOP-ASCRA*1B11+100010
26
27                ;A7:
28                        SETUP             ;ADDITION OF NEGATED
29                        LCALL SETUL
30 11344 100410           SETUL-ASCRA*1B11+100010
31                        RANDOM            ;NUMBERS SHOULD PRODUCE
32                        LCALL FRANG
33 11345 100450           FRANG-ASCRA*1B11+100010
34 11346 110440           NEGO 0,2          ;ZERO AND A CARRY.
35 11347 143204           ADDR 2,0,SZR
36                        ERROR
37 11350 004402           JSR .+2
38 11351 000402           JMP .+2
39                        LCALL ERRET
40 11352 100470           ERRET-ASCRA*1B11+100010
41                        LOOP
42                        LCALL LLOOP
43 11353 100430           LLOOP-ASCRA*1B11+100010
44
45 11354 101001           MOV 0,0,SKP
46 11355 177777 ADDTEM:   -1
```

```
!0158 N3MRT
01                ;A8:
02                        SETUP             ;ADD TEST.
03                        LCALL SETUL
04 11356 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM
06                        LCALL FRANG
07 11357 100450           FRANG-ASCRA*1B11+100010
08 11360 152520           SUBZL 2,2         ;=(+1)
09 11361 151300           MOVS 2,2          ;=(400)
10 11362 150400           NEG 2,2           ;=(177400)
11 11363 143700           ANDS 2,0          ;SAVE HIGH ORDER 8 BITS.
12 11364 105320           MOVZS 0,1         ;SAME 8 BITS TO C(AC1)L.
13 11365 103100           ADDL 0,0          ;MOVE C(AC0) LEFT VIA
14 11366 103100           ADDL 0,0          ;ADD SHIFT.
15 11367 103100           ADDL 0,0
16 11370 103100           ADDL 0,0
17 11371 106414           SUB# 0,1,SZR
18                        ERROR
19 11372 004402           JSR .+2
20 11373 000402           JMP .+2
21                        LCALL ERRET
22 11374 100470           ERRET-ASCRA*1B11+100010
23                        LOOP
24                        LCALL LLOOP
25 11375 100430           LLOOP-ASCRA*1B11+100010
26
```

```
10159 N3MRT
01 11376 101001          MOV 0,0,SKP
02 11377 177400 M400:    -400
03              :AR1:
04                       SETUP            ;THE ORIGIONAL CONTENTS OF
05                       LCALL SETUL
06 11400 100410          SETUL-ASCRA*1B11+100010
07                       RANDOM           ;AC1, BITS 0-7 ARE SQUARED
08                       LCALL FRANG
09 11401 100450          FRANG-ASCRA*1B11+100010
10 11402 030775          LDA 2,M400       ;VIA MULTIPLY. THE SQUARE
11 11403 133700          ANDS 1,2         ;ROOT OF THE PRODUCT SHOULD
12 11404 145000          MOV 2,1          ;EQUAL THE ORIGIONAL.
13                       CALL
14                       .MPYU            ;SEE SYSTEM REFFERANCE
15 11405 004442          JSR MPYU
16 11406 121000          MOV 1,0          ;MANUAL FOR FURTHER INFORMATION
17                       CALL             ;ON MULTIPLY/SQ ROOT
18                       SQRT             ;PROGRAMS.
19 11407 004467          JSR SQRT.
20 11410 112414          SUB# 0,2,SZR
21                       ERROR
22 11411 004402          JSR .+2
23 11412 000402          JMP .+2
24                       LCALL ERRET
25 11413 100470          ERRET-ASCRA*1B11+100010
26                       LOOP
27                       LCALL LLOOP
28 11414 100430          LLOOP-ASCRA*1B11+100010
```

```
10160 N3MRT
01              :AR2:
02                       SETUP            ;TAKE THE SQUARE ROOT
03                       LCALL SETUL
04 11415 100410          SETUL-ASCRA*1B11+100010
05                       RANDOM           ;OF A NUMBER. THE SQUARE
06                       LCALL FRANG
07 11416 100450          FRANG-ASCRA*1B11+100010
08                       CALL             ;ROOT OF THE RESULT SQUARED
09                       SQRT             ;SHOULD BE THE SAME AS THE
10 11417 004457          JSR SQRT.
11 11420 105060          MOVC 0,1         ;ORIGIONAL ROOT.
12 11421 131060          MOVC 1,2
13                       CALL
14                       .MPYU
15 11422 004425          JSR MPYU
16 11423 121000          MOV 1,0
17                       CALL             ;SQUARED NOW TAKE ROOT.
18                       SQ
19 11424 004462          JSR SQ.
20 11425 112714          SUBS# 0,2,SZR
21                       ERROR
22 11426 004402          JSR .+2
23 11427 000402          JMP .+2
24                       LCALL ERRET
25 11430 100470          ERRET-ASCRA*1B11+100010
26                       LOOP
27                       LCALL LLOOP
28 11431 100430          LLOOP-ASCRA*1B11+100010
```

```
!0161 N3MRT
01                ;AR3:
02                        SETUP           ;FIND SQUARE ROOT VIA
03                        LCALL SETUL
04 11432 100410           SETUL=ASCRA*1B11+100010
05                        RANDOM          ;DIFFIRENT SUBROUTINES.
06                        LCALL FRANG
07 11433 100450           FRANG=ASCRA*1B11+100010
08 11434 141000           MOV 2,0
09                        CALL
10                        SQRT
11 11435 004441           JSR SQRT.
12 11436 105000           MOV 0,1         ;SAVE FIRST RESULT IN AC1
13 11437 141000           MOV 2,0
14                        CALL
15                        SQ              ;RESULT IN AC0.
16 11440 004446           JSR SQ.
17 11441 106714           SUBS# 0,1,SZR
18                        ERROR
19 11442 004402           JSR .+2
20 11443 000402           JMP .+2
21                        LCALL ERRET
22 11444 100470           ERRET=ASCRA*1B11+100010
23                        LOOP
24                        LCALL LLOOP
25 11445 100430           LLOOP=ASCRA*1B11+100010
26 11446 000455           JMP AR4
```

```
!0162 N3MRT
01 11447 102460 MPYU:     SUBC 0,0        ;C(AC1)*C(AC2)
02 11450 054411 MPYA:     STA 3,.CB03     ;RESULT IN AC0,AC1.
03 11451 034411           LDA 3,.CB20     ;SEE SYSTEM REFFERANCE
04 11452 125203 .CB99:    MOVR 1,1,SNC    ;MANUAL FOR FURTHER
05 11453 101201           MOVR 0,0,SKP    ;INFORMATION.
06 11454 143220           ADDZR 2,0
07 11455 175404           INC 3,3,SZR
08 11456 000774           JMP .CB99
09 11457 125260           MOVCR 1,1
10 11460 002401           JMP @.CB03
11 11461 000000 .CB03:    0
12 11462 177760 .CB20:    -20
13
14 11463 102400 DIVI:     SUB 0,0         ;C(AC0),C(AC1)/C(AC2).
15 11464 054775 DIVU:     STA 3,.CB03     ;AC0=REMAINDER
16 11465 034775           LDA 3,.CB20     ;AC1=QUOTIENT
17 11466 125120           MOVZL 1,1       ;SEE SYSTEM REFFERANCE
18 11467 101100 .CC98:    MOVL 0,0        ;MANUAL.
19 11470 142412           SUB# 2,0,SZC
20 11471 142400           SUB 2,0
21 11472 125100           MOVL 1,1
22 11473 175404           INC 3,3,SZR
23 11474 000773           JMP .CC98
24 11475 002764           JMP @.CB03
25
```

```
!0163 N3MRT
01 11476 054423 SQRT.:  STA 3,SQ.S3
02 11477 126520         SUBZL 1,1       ;FIND SQ ROOT OF C(AC0).
03 11500 135120         MOVZL 1,3       ;SEE THE SYSTEM REFFERANCE
04 11501 122422         SUBZ 1,0,SZC    ;MANUAL.
05 11502 167001         ADD 3,1,SKP     ;RESULT IN AC0.
06 11503 121221         MOVZR 1,0,SKP   ;AC1 DESTROYED.
07 11504 000775         JMP .-3
08 11505 002414         JMP @SQ.S3
09
10 11506 054413 SQ.:    STA 3,SQ.S3
11 11507 176400         SUB 3,3         ;FIND SQ ROOT OF C(AC0).
12 11510 054412         STA 3,SQTEM     ;SAME RESULT AS PREVIOUS
13 11511 162023         ADCZ 3,0,SNC    ;TEST BUT CODE IS
14 11512 000405         JMP SQ1         ;DIFFIRENT.
15 11513 010407         ISZ SQTEM
16 11514 010406         ISZ SQTEM
17 11515 034405         LDA 3,SQTEM     ;RESULT WILL BE IN AC0.
18 11516 000773         JMP .-5
19 11517 161200 SQ1:    MOVR 3,0
20 11520 002401         JMP @SQ.S3
21
22 11521 000000 SQ.S3: 0
23 11522 000000 SQTEM:  0
```

```
!0164 N3MRT
01              AR4:    SETUP           ;THE CONTENTS OF AC2 IS
02                      LCALL SETUL
03 11523 100410         SETUL-ASCRA*1B11+100010
04              AR4L:   RANDOM          ;DIVIDED INTO AC0-1.
05                      LCALL FRANG
06 11524 100450         FRANG-ASCRA*1B11+100010
07 11525 040471         STA 0,0AC0
08 11526 044471         STA 1,0AC1
09 11527 050471         STA 2,0AC2
10 11530 142432         SUBZ# 2,0,SZC   ;THIS RESULT MULTIPLIED
11 11531 000773         JMP AR4L                ;BY AC2 SHOULD PRODUCE
12                      CALL            ;THE ORIGIONAL NUMBERS.
13                      .DIVU
14 11532 004732         JSR DIVU
15                      CALL
16                      .MPYA
17 11533 004715         JSR MPYA
18 11534 034464         LDA 3,0AC2
19 11535 156414         SUB# 2,3,SZR
20                      ERROR           ;AC2 CHANGED?
21 11536 004402         JSR .+2
22 11537 000402         JMP .+2
23                      LCALL ERRET
24 11540 100470         ERRET-ASCRA*1B11+100010
25 11541 034456         LDA 3,0AC1
26 11542 136714         SUBS# 1,3,SZR
27                      ERROR           ;AC1 CHANGED.
28 11543 004402         JSR .+2
29 11544 000402         JMP .+2
30                      LCALL ERRET
31 11545 100470         ERRET-ASCRA*1B11+100010
32 11546 034450         LDA 3,0AC0
33 11547 116714         SUBS# 0,3,SZR
34                      ERROR           ;AC0 CHANGED.
35 11550 004402         JSR .+2
36 11551 000402         JMP .+2
37                      LCALL ERRET
38 11552 100470         ERRET-ASCRA*1B11+100010
39                      LOOP
40                      LCALL LLOOP
41 11553 100430         LLOOP-ASCRA*1B11+100010
```

```
!0165 N3MRT
01                ;AR5:
02                        SETUP           ;USE INC TO FORM
03                        LCALL SETUL
04 11554 100410           SETUL-ASCRA*1B11+100010
05 11555 152440           SUB0 2,2        ;THE NUMBER 177400
06 11556 151504           INCL 2,2,SZR    ;IN AC1 AND 400 IN AC2.
07 11557 151504           INCL 2,2,SZR
08 11560 151504           INCL 2,2,SZR
09 11561 151504           INCL 2,2,SZR
10 11562 151507           INCL 2,2,SBN    ;THE "SBN/SZR" SHOULD
11 11563 151507           INCL 2,2,SBN    ;NOT CAUSE A SKIP.
12 11564 151507           INCL 2,2,SBN
13 11565 145707           INCS 2,1,SBN
14 11566 151407           INC 2,2,SBN
15 11567 151400           INC 2,2
16 11570 133014           ADD# 1,2,SZR
17                        ERROR
18 11571 004402           JSR .+2
19 11572 000402           JMP .+2
20                        LCALL ERRET
21 11573 100470           ERRET-ASCRA*1B11+100010
22                        LOOP
23                        LCALL LLOOP
24 11574 100430           LLOOP-ASCRA*1B11+100010
```

```
!0166 N3MRT
01                ;AR6:
02                        SETUP           ;USE THE INCR INSTRUCTION
03                        LCALL SETUL
04 11575 100410           SETUL-ASCRA*1B11+100010
05 11576 126420           SUBZ 1,1        ;TO FORM THE NUMBER
06 11577 125606           INCR 1,1,SEZ    ;177400. THIS NUMBER
07 11600 125606           INCR 1,1,SEZ    ;COMPLIMENTED AND SWAPPED
08 11601 125606           INCR 1,1,SEZ    ;SHOULD BE THE SAME
09 11602 125606           INCR 1,1,SEZ    ;NUMBER.
10 11603 125606           INCR 1,1,SEZ    ;THE "SEZ" SHOULD NEVER
11 11604 125606           INCR 1,1,SEZ    ;CAUSE A SKIP.
12 11605 125606           INCR 1,1,SEZ
13 11606 125606           INCR 1,1,SEZ
14 11607 130304           COMS 1,2,SZR
15 11610 132714           SUBS# 1,2,SZR
16                        ERROR
17 11611 004402           JSR .+2
18 11612 000402           JMP .+2
19                        LCALL ERRET
20 11613 100470           ERRET-ASCRA*1B11+100010
21                        LOOP
22                        LCALL LLOOP
23 11614 100430           LLOOP-ASCRA*1B11+100010
24 11615 000405           JMP .+5
25 11616 000000 OAC0: 0
26 11617 000000 OAC1: 0
27 11620 000000 OAC2: 0
28 11621 000627 MPYAL:  JMP MPYA
```

```
10167 N3MRT
01                ;AR7:
02                        SETUP             ;100000 NEGATED IS
03                        LCALL SETUL
04 11622 100410           SETUL-ASCRA*1B11+100010
05 11623 102625           SUBZR 0,0,SNR     ;STILL 100000.
06                        ERROR
07 11624 004402           JSR .+2
08 11625 000402           JMP .+2
09                        LCALL ERRET
10 11626 100470           ERRET-ASCRA*1B11+100010
11 11627 104406           NEG 0,1,SEZ       ;C(CARRY)=0
12                        ERROR
13 11630 004402           JSR .+2
14 11631 000402           JMP .+2
15                        LCALL ERRET
16 11632 100470           ERRET-ASCRA*1B11+100010
17 11633 124402           NEG 1,1,SZC
18                        ERROR
19 11634 004402           JSR .+2
20 11635 000402           JMP .+2
21                        LCALL ERRET
22 11636 100470           ERRET-ASCRA*1B11+100010
23 11637 124463           NEGC 1,1,SNC      ;C(CARRY)=1
24                        ERROR
25 11640 004402           JSR .+2
26 11641 000402           JMP .+2
27                        LCALL ERRET
28 11642 100470           ERRET-ASCRA*1B11+100010
29 11643 124407           NEG 1,1,SBN
30                        ERROR
31 11644 004402           JSR .+2
32 11645 000402           JMP .+2
33                        LCALL ERRET
34 11646 100470           ERRET-ASCRA*1B11+100010
35 11647 124465           NEGC 1,1,SNR      ;C(1)=100000
36                        ERROR
37 11650 004402           JSR .+2
38 11651 000402           JMP .+2
39                        LCALL ERRET
40 11652 100470           ERRET-ASCRA*1B11+100010
41 11653 124465           NEGC 1,1,SNR
42                        ERROR
43 11654 004402           JSR .+2
44 11655 000402           JMP .+2
45                        LCALL ERRET
46 11656 100470           ERRET-ASCRA*1B11+100010
47 11657 124407           NEG 1,1,SBN
48                        ERROR
49 11660 004402           JSR .+2
50 11661 000402           JMP .+2
51                        LCALL ERRET
52 11662 100470           ERRET-ASCRA*1B11+100010
53 11663 124407           NEG 1,1,SBN
54                        ERROR
55 11664 004402           JSR .+2
56 11665 000402           JMP .+2
57                        LCALL ERRET
58 11666 100470           ERRET-ASCRA*1B11+100010
59 11667 106414           SUB# 0,1,SZR
60                        ERROR
```

```
 0168 N3MRT
01 11670 004402           JSR .+2
02 11671 000402           JMP .+2
03                        LCALL ERRET
04 11672 100470           ERRET-ASCRA*1B11+100010
05                        LOOP
06                        LCALL LLOOP
07 11673 100430           LLOOP-ASCRA*1B11+100010
08
```

```
!0169 N3MRT
01                ;AR8:
02                        SETUP           ;NEGATION OF ZERO
03                        LCALL SETUL
04 11674 100410           SETUL-ASCRA*1B11+100010
05 11675 102440           SUBO 0,0        ;SHOULD PRODUCE ZERO
06 11676 100762           NEGCS 0,0,SZC   ;AND A CARRY.
07                        ERROR
08 11677 004402           JSR .+2
09 11700 000402           JMP .+2
10                        LCALL ERRET
11 11701 100470           ERRET-ASCRA*1B11+100010
12 11702 100762           NEGCS 0,0,SZC
13                        ERROR
14 11703 004402           JSR .+2
15 11704 000402           JMP .+2
16                        LCALL ERRET
17 11705 100470           ERRET-ASCRA*1B11+100010
18 11706 100706           NEGS 0,0,SEZ
19                        ERROR
20 11707 004402           JSR .+2
21 11710 000402           JMP .+2
22                        LCALL ERRET
23 11711 100470           ERRET-ASCRA*1B11+100010
24 11712 100706           NEGS 0,0,SEZ
25                        ERROR
26 11713 004402           JSR .+2
27 11714 000402           JMP .+2
28                        LCALL ERRET
29 11715 100470           ERRET-ASCRA*1B11+100010
30 11716 100544           NEGOL 0,0,SZR
31                        ERROR
32 11717 004402           JSR .+2
33 11720 000402           JMP .+2
34                        LCALL ERRET
35 11721 100470           ERRET-ASCRA*1B11+100010
36 11722 100403           NEG 0,0,SNC
37                        ERROR
38 11723 004402           JSR .+2
39 11724 000402           JMP .+2
40                        LCALL ERRET
41 11725 100470           ERRET-ASCRA*1B11+100010
42 11726 100644           NEGOR 0,0,SZR
43                        ERROR
44 11727 004402           JSR .+2
45 11730 000402           JMP .+2
46                        LCALL ERRET
47 11731 100470           ERRET-ASCRA*1B11+100010
48 11732 104704           NEGS 0,1,SZR
49                        ERROR
50 11733 004402           JSR .+2
51 11734 000402           JMP .+2
52                        LCALL ERRET
53 11735 100470           ERRET-ASCRA*1B11+100010
54 11736 130704           NEGS 1,2,SZR
55                        ERROR
56 11737 004402           JSR .+2
57 11740 000402           JMP .+2
58                        LCALL ERRET
59 11741 100470           ERRET-ASCRA*1B11+100010
60                        LOOP
```

```
 0170 N3MRT
01                        LCALL LLOOP
02 11742 100430           LLOOP-ASCRA*1B11+100010
```

```
!0171 N3MRT
01                ;AR9:
02                        SETUP           ;C(AC1)*0+C(AC0) SHOULD
03                        LCALL SETUL
04 11743 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM          ;PLACE AC0 IN AC1. SEE
06                        LCALL FRANG
07 11744 100450           FRANG-ASCRA*1B11+100010
08 11745 040651           STA 0,0AC0
09 11746 152400           SUB 2,2         ;SYSTEM REFFERANCE MANUAL
10                        CALL            ;FOR FURTHER INFORMATION.
11 11747 004652           JSR     MPYAL
12 11750 034646           LDA 3,0AC0
13 11751 166414           SUB# 3,1,SZR
14                        ERROR
15 11752 004402           JSR .+2
16 11753 000402           JMP .+2
17                        LCALL ERRET
18 11754 100470           ERRET-ASCRA*1B11+100010
19                        LOOP
20                        LCALL LLOOP
21 11755 100430           LLOOP-ASCRA*1B11+100010
22
23                ;AR10:
24                        SETUP           ;TEST "COM/INC" SWAPPED.
25                        LCALL SETUL
26 11756 100410           SETUL-ASCRA*1B11+100010
27 11757 102300           ADCS 0,0
28 11760 105705           INCS 0,1,SNR
29 11761 130304           COMS 1,2,SZR
30 11762 145705           INCS 2,1,SNR
31 11763 124346           COMOS 1,1,SEZ
32 11764 131707           INCS 1,2,SBN
33 11765 150304           COMS 2,2,SZR
34 11766 151704           INCS 2,2,SZR
35                        ERROR
36 11767 004402           JSR .+2
37 11770 000402           JMP .+2
38                        LCALL ERRET
39 11771 100470           ERRET-ASCRA*1B11+100010
40                        LOOP
41                        LCALL LLOOP
42 11772 100430           LLOOP-ASCRA*1B11+100010
```

```
!0172 N3MRT
01                ;AR11:
02                        SETUP           ;COMPLIMENT AND INCREMENT
03                        LCALL SETUL
04 11773 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM          ;SHOULD BE THE SAME AS
06                        LCALL FRANG
07 11774 100450           FRANG-ASCRA*1B11+100010
08 11775 110400           NEG 0,2         ;NEGATE
09 11776 104000           COM 0,1
10 11777 125400           INC 1,1
11 12000 132414           SUB# 1,2,SZR
12                        ERROR
13 12001 004402           JSR .+2
14 12002 000402           JMP .+2
15                        LCALL ERRET
16 12003 100470           ERRET-ASCRA*1B11+100010
17                        LOOP
18                        LCALL LLOOP
19 12004 100430           LLOOP-ASCRA*1B11+100010
20
21                ;AR12:
22                        SETUP           ;TEST COM LEFT AND RIGHT.
23                        LCALL SETUL
24 12005 100410           SETUL-ASCRA*1B11+100010
25                        RANDOM
26                        LCALL FRANG
27 12006 100450           FRANG-ASCRA*1B11+100010
28 12007 110100           COML 0,2
29 12010 144260           COMCR 2,1
30 12011 106714           SUBS# 0,1,SZR
31                        ERROR
32 12012 004402           JSR .+2
33 12013 000402           JMP .+2
34                        LCALL ERRET
35 12014 100470           ERRET-ASCRA*1B11+100010
36                        LOOP
37                        LCALL LLOOP
38 12015 100430           LLOOP-ASCRA*1B11+100010
```

```
10173 N3MRT
01                ;AR13:
02                        SETUP             ;FORM THE PARITY OF
03                        LCALL SETUL
04 12016 100410           SETUL-ASCRA*1B11+100010
05 12017 105020           MOVZ 0,1          ;C(AC0) IN DIFFIRENT
06 12020 176000           ADC 3,3           ;ROUTINES. CHECK THAT
07 12021 117000           ADD 0,3           ;THE RESULTS ARE EQUAL.
08 12022 163704           ANDS 3,0,SZR
09 12023 000775           JMP .-3
10 12024 102660           SUBCR 0,0         ;SAVE PARITY IN BIT 0.
11 12025 176620           SUBZR 3,3
12 12026 125102           MOVL 1,1,SZC
13 12027 101400           INC 0,0           ;BIT 15 WILL CONTAIN
14 12030 175224           MOVZR 3,3,SZR     ;THE PARITY.
15 12031 000775           JMP .-3
16 12032 115200           MOVR 0,3          ;CHECK TO SEE IF BITS
17 12033 103012           ADD# 0,0,SZC      ;ARE LIKE.
18                        ERROR
19 12034 004402           JSR .+2
20 12035 000402           JMP .+2
21                        LCALL ERRET
22 12036 100470           ERRET-ASCRA*1B11+100010
23                        LOOP
24                        LCALL LLOOP
25 12037 100430           LLOOP-ASCRA*1B11+100010
26
27                ;AR15:
28                        SETUP             ;MISC TEST OF SUB LEFT
29                        LCALL SETUL
30 12040 100410           SETUL-ASCRA*1B11+100010
31                        RANDOM            ;AND RIGHT.
32                        LCALL FRANG
33 12041 100450           FRANG-ASCRA*1B11+100010
34 12042 131000           MOV 1,2
35 12043 106500           SUBL 0,1
36 12044 127600           ANDR 1,1
37 12045 107000           ADD 0,1
38 12046 106500           SUBL 0,1
39 12047 127600           ANDR 1,1
40 12050 107000           ADD 0,1
41 12051 132414           SUB# 1,2,SZR
42                        ERROR
43 12052 004402           JSR .+2
44 12053 000402           JMP .+2
45                        LCALL ERRET
46 12054 100470           ERRET-ASCRA*1B11+100010
47                        LOOP
48                        LCALL LLOOP
49 12055 100430           LLOOP-ASCRA*1B11+100010
```

```
10174 N3MRT
01                ;AR16:
02                        SETUP             ;MISC TEST OF AND RIGHT.
03                        LCALL SETUL
04 12056 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM
06                        LCALL FRANG
07 12057 100450           FRANG-ASCRA*1B11+100010
08 12060 131040           MOVO 1,2
09 12061 113600           ANDR 0,2
10 12062 107400           AND 0,1
11 12063 151100           MOVL 2,2
12 12064 146414           SUB# 2,1,SZR
13                        ERROR
14 12065 004402           JSR .+2
15 12066 000402           JMP .+2
16                        LCALL ERRET
17 12067 100470           ERRET-ASCRA*1B11+100010
18                        LOOP
19                        LCALL LLOOP
20 12070 100430           LLOOP-ASCRA*1B11+100010
```

```
!0175 N3MRT
01                ;AR17:
02                        SETUP           ;MISC ADD SWAPPED TEST.
03                        LCALL SETUL
04 12071 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM
06                        LCALL FRANG
07 12072 100450           FRANG-ASCRA*1B11+100010
08 12073 131000           MOV 1,2
09 12074 113000           ADD 0,2
10 12075 107300           ADDS 0,1
11 12076 125300           MOVS 1,1
12 12077 132414           SUB# 1,2,SZR
13                        ERROR
14 12100 004402           JSR .+2
15 12101 000402           JMP .+2
16                        LCALL ERRET
17 12102 100470           ERRET-ASCRA*1B11+100010
18                        LOOP
19                        LCALL LLOOP
20 12103 100430           LLOOP-ASCRA*1B11+100010
21
22                ;AR18:
23                        SETUP           ;CHECK ADC LEFT.
24                        LCALL SETUL
25 12104 100410           SETUL-ASCRA*1B11+100010
26                        RANDOM
27                        LCALL FRANG
28 12105 100450           FRANG-ASCRA*1B11+100010
29 12106 131000           MOV 1,2
30 12107 112120           ADCZL 0,2
31 12110 100000           COM 0,0
32 12111 107120           ADDZL 0,1
33 12112 132414           SUB# 1,2,SZR
34                        ERROR
35 12113 004402           JSR .+2
36 12114 000402           JMP .+2
37                        LCALL ERRET
38 12115 100470           ERRET-ASCRA*1B11+100010
39                        LOOP
40                        LCALL LLOOP
41 12116 100430           LLOOP-ASCRA*1B11+100010
```

```
!0176 N3MRT
01                ;AR19:
02                        SETUP           ;TEST ADC RIGHT.
03                        LCALL SETUL
04 12117 100410           SETUL-ASCRA*1B11+100010
05                        RANDOM
06                        LCALL FRANG
07 12120 100450           FRANG-ASCRA*1B11+100010
08 12121 131000           MOV 1,2
09 12122 112220           ADCZR 0,2
10 12123 100000           COM 0,0
11 12124 107220           ADDZR 0,1
12 12125 132414           SUB# 1,2,SZR
13                        ERROR
14 12126 004402           JSR .+2
15 12127 000402           JMP .+2
16                        LCALL ERRET
17 12130 100470           ERRET-ASCRA*1B11+100010
18                        LOOP
19                        LCALL LLOOP
20 12131 100430           LLOOP-ASCRA*1B11+100010
21
22                ;AR20:
23                        SETUP           ;TEST SUB RIGHT.
24                        LCALL SETUL
25 12132 100410           SETUL-ASCRA*1B11+100010
26                        RANDOM
27                        LCALL FRANG
28 12133 100450           FRANG-ASCRA*1B11+100010
29 12134 131000           MOV 1,2
30 12135 106620           SUBZR 0,1
31 12136 100420           NEGZ 0,0
32 12137 113200           ADDR 0,2
33 12140 132414           SUB# 1,2,SZR
34                        ERROR
35 12141 004402           JSR .+2
36 12142 000402           JMP .+2
37                        LCALL ERRET
38 12143 100470           ERRET-ASCRA*1B11+100010
39                        LOOP
40                        LCALL LLOOP
41 12144 100430           LLOOP-ASCRA*1B11+100010
```

```
!0177 N3MRT
01                ARJSR:  SETUP           ;TEST THAT INDEX WITH
02                        LCALL SETUL
03 12145 100410           SETUL-ASCRA*1B11+100010
04 12146 004401           JSR .+1         ;SIGN BIT SET DOES
05 12147 171122           MOVZL 3,2,SZC   ;NOT LOAD INTO PC ON JSR.
06                        ERROR
07 12150 004402           JSR .+2
08 12151 000402           JMP .+2
09                        LCALL ERRET
10 12152 100470           ERRET-ASCRA*1B11+100010
11 12153 151240           MOVOR 2,2
12 12154 005007           JSR 7,2         ;GO TO NEXT LOCATION
13 12155 004401           JSR .+1
14 12156 165000           MOV 3,1         ;JSR SHOULD NEVER
15 12157 125112           MOVL# 1,1,SZC   ;STORE THE SIGN BIT.
16                        ERROR
17 12160 004402           JSR .+2
18 12161 000402           JMP .+2
19                        LCALL ERRET
20 12162 100470           ERRET-ASCRA*1B11+100010
21                        LOOP
22                        LCALL LLOOP
23 12163 100430           LLOOP-ASCRA*1B11+100010
24
25                AREND:  LCALL RETU2
26 12164 100510           RETU2-ASCRA*1B11+100010
27 12165 000000 ATEND: 0
```

```
!0178 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 FMTF=NIOC FPU2
10        062374          .DIAC FLODS=DO8P 0,FPU1
11        062375          .DIAC FLODD=DO8P 0,FPU2
12        062174          .DIAC FSTRS=DO8S 0,FPU1
13        062175          .DIAC FSTRD=DO8S 0,FPU2
14        060676          .DIAC FPRST=DIAC 0,FPU
15        061076          .DIAC FPLST=DOA 0,FPU
```

```
!0179 N3MRT
01                ;TEST INTERFACE TO LINKER
02                        NEXTT F.000
03        012166 LMEML=.
04        000161          .LOC LPG0
05 00161  012171          F.000
06        000162 LPG0=.
07        012166          .LOC LMEML
08 12166  000000          0         ;TEST PASS CTR
09 12167  000000          0         ;TEST ERROR CTR
10 12170  000000          0         ;INTERRUPT TIMEOUT SWITCH
11 12171  012210 F.000:   FP.01
12 12172  012224          FP.02
13 12173  000000          0
14 12174  000000          0
15 12175  177777          -1
16 12176  176000          176000
17 12177  012443          FP.EC
18 12200  012443          FP.EC
19        012201          .TXTE (
20 12201  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 12210  102000 FP.01:   ADC 0,0
31 12211  040762          STA 0,F.000+2    ;DISABLE TEST ENTER
32 12212  104400          NEG 0,1
33 12213  065076          FPLST 1          ;SETS DIAG MODE
34 12214  064676          FPRST 1          ;READ STATUS BACK
35 12215  124405          NEG 1,1,SNR      ;SKP IF FPU
36 12216  001400          JMP 0,3          ;EXIT TEST DISABLED
37 12217  040464          STA 0,FP.TK      ;SET NO SCRA SW
38 12220  102400          SUB 0,0
39 12221  040752          STA 0,F.000+2
40 12222  061076          FPLST 0
41 12223  001400          JMP 0,3
```

```
!0180 N3MRT
01                ;
02                ;EXECUTE ENTRY POINT
03 12224  010457 FP.02:   ISZ FP.TK        ;SKP IS NO SCRATCH
04 12225  000464          JMP FP.03        ;DO NXT IN SEQ
05 12226  020153          LDA 0,SCRHI
06 12227  101004          MOV 0,0,SZR
07 12230  000422          JMP FP.2A
08 12231  102400          SUB 0,0
09 12232  040456          STA 0,FP.KK      ;0 # EXTRA 1K'S
10                        LCALL ASCRA      ;TRY TO GET 1K
11 12233  100010          ASCRA=ASCRA*1B11+100010
12 12234  000565          JMP FP.05        ;NONE AVAILABLE
13 12235  102000          ADC 0,0          ;-1 TO
14 12236  040447          STA 0,FP.ES      ;NO ERRSW
15                        LCALL ARANG      ;GET RAN#
16 12237  100270          ARANG=ASCRA*1B11+100010
17 12240  030444          LDA 2,FP.37
18 12241  105000          MOV 0,1
19                        LCALL ADIVI      ;REM=#1K'S TO EXPAND
20 12242  100310          ADIVI=ASCRA*1B11+100010
21 12243  100405          NEG 0,0,SNR
22 12244  000406          JMP FP.2A
23                FP.2L:  LCALL ESCRA      ;EXPAND SCRATCH 1K
24 12245  100030          ESCRA=ASCRA*1B11+100010
25 12246  000404          JMP FP.2A        ;NO MORE AVAILABLE
26 12247  010441          ISZ FP.KK        ;+1 # EXTRA 1K'S
27 12250  101404          INC 0,0,SZR
28 12251  000774          JMP FP.2L        ;KEEP EXPANDING
```

```
10181 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 12252 030432 FP.2A:  LDA 2,FP.37
06 12253 150400         NEG 2,2
07                      LCALL ARANG     ;GET RAN #
08 12254 100270         ARANG-ASCRA*1B11+100010
09 12255 030432         LDA 2,FP255
10 12256 105000         MOV 0,1
11                      LCALL ADIVI     ;RET WITH AC0=DISPLACEMENT
12 12257 100310         ADIVI-ASCRA*1B11+100010
13 12260 024152         LDA 1,SCRLO
14 12261 125400         INC 1,1
15 12262 123000         ADD 1,0
16 12263 040545 FP.2C:  STA 0,FP.LC     ;ADRESS TO RELOC TESTS
17 12264 000425         JMP FP.03       ;INIT FIRST TEST
18                ;RELOCATE TEST TO SELECTED SCRATCH AREA
19 12265 144000 FP.RL:  COM 2,1         ;-#WORDS TO MOV
20 12266 111000         MOV 0,2         ;TO ADRS
21 12267 034543         LDA 3,FP.BG     ;FROM ADRS
22 12270 020535         LDA 0,FP.HI     ;GET HI REL SCR
23 12271 041000         STA 0,0,2
24 12272 151400         INC 2,2         ;HIGH LIMIT FOR TEST
25 12273 021400 FP.L2:  LDA 0,0,3
26 12274 041000         STA 0,0,2
27 12275 175400         INC 3,3
28 12276 151400         INC 2,2
29 12277 125404         INC 1,1,SZR
30 12300 000773         JMP FP.L2       ;MOV ALL TO SCRTCH
31 12301 050532         STA 2,FP.EN     ;SAVE LAST LOC+1
32 12302 000452         JMP FP.GO
33 12303 000000 FP.TK:  0
34 12304 000037 FP.37:  37
35 12305 000000 FP.ES:  0
36 12306 001777 FK1K:   1777
37 12307 000377 FP255:  255.
38 12310 000000 FP.KK:  0
```

```
10182 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 12311 020774 FP.03:  LDA 0,FP.ES     ;GET ERR SWITCH
08 12312 101005         MOV 0,0,SNR     ;SKIP IS NO ERR
09 12313 000441         JMP FP.GO       ;USE PREV ASSIGN
10                      LCALL ARANG
11 12314 100270         ARANG-ASCRA*1B11+100010
12 12315 105000         MOV 0,1
13 12316 030510         LDA 2,FP.40
14 12317 034771         LDA 3,FP.KK
15 12320 172400         SUB 3,2         ;DONT EXPAND ABOVE 32K
16                      LCALL ADIVI
17 12321 100310         ADIVI-ASCRA*1B11+100010
18 12322 040514         STA 0,FP.LP
19 12323 024505         LDA 1,FP.LC
20 12324 030152         LDA 2,SCRLO
21 12325 146400         SUB 2,1
22 12326 111300         MOVS 0,2
23 12327 153120         ADDZL 2,2
24 12330 050504         STA 2,FP.LO
25 12331 034757         LDA 3,FP.KK     ;RECALC SCRHI
26 12332 175300         MOVS 3,3
27 12333 177120         ADDZL 3,3
28 12334 157000         ADD 2,3
29 12335 020751         LDA 0,FK1K
30 12336 117000         ADD 0,3
31 12337 054466         STA 3,FP.HI
32 12340 133220         ADDZR 1,2
33 12341 151500         INCL 2,2
34 12342 050462         STA 2,FP.GA
35 12343 020740         LDA 0,FP.TK
36 12344 034545         LDA 3,FP.TT
37 12345 117000         ADD 0,3
38 12346 031400         LDA 2,0,3
39 12347 050463         STA 2,FP.BG
40 12350 020460         LDA 0,FP.LC
41 12351 031000         LDA 2,0,2
42 12352 054457         STA 3,FP.TP
43 12353 000712         JMP FP.RL
44 12354 020462 FP.GO:  LDA 0,FP.LP     ;REMAP SCR TO HERE
45 12355 030447         LDA 2,FP.GA     ;STARTING LOGICAL ADRS
46 12356 024461         LDA 1,FPERR     ;1=ERROR RET ADDRS
47                      LCALL GSCRA     ;GO TO SCRFPCH
48 12357 100070         GSCRA-ASCRA*1B11+100010
49 12360 000427         JMP FP.GO       ;COULDNT REMAP GO DIRECT
```

```
!0183 N3MRT
01                ;RETURN TO NEXT LOC PASS COMPLETE NO ERRS
02 12361 010450           ISZ FP.TP
03 12362 022447           LDA 0,@FP.TP      ;GET ADRS NXT TST
04 12363 100004           COM 0,0,SZR       ;=-1 IS END OF SEQ
05 12364 000437           JMP FP.XI
06 12365 020230           LDA 0,SWREG       ;CK IF REL. SCRATCH
07 12366 103123           ADDZL 0,0,SNC     ;SKP IS NOT REL.
08 12367 000406           JMP .+6
09 12370 020715           LDA 0,FP.ES       ;CHK FOR PREV ERR
10 12371 101004           MOV 0,0,SZR       ;SKP ON PREV ERR
11 12372 000403           JMP .+3           ;RELEASE ASSIGN
12 12373 040710           STA 0,FP.TK
13 12374 000427           JMP FP.XI
14                        LCALL ARANG       ;IF BITS 0 AND 1=1
15 12375 100270           ARANG-ASCRA*1B11+100010
16 12376 103043           ADDO 0,0,SNC      ;RELEASE AND REMAP
17 12377 103003           ADD 0,0,SNC       ;IF EITHER =0 MAKE
18 12400 000404           JMP FP.04                 ;NEXT PASS SELECT NEW LP
19                FP.4A:  LCALL RSCRA
20 12401 100050           RSCRA-ASCRA*1B11+100010
21 12402 000417           JMP FP.05         ;SET NO SCRFPCH
22 12403 000776           JMP FP.4A
23 12404 102400 FP.04:    SUB 0,0
24 12405 040676           STA 0,FP.TK
25 12406 000415           JMP FP.XI
26                ;MAP OPTION DOES NOT EXIST GO DIRECT
27                ;AFTER READJUSTING SCRHI PARAMETER
28 12407 020153 FP.GD:    LDA 0,SCRHI
29 12410 040415           STA 0,FP.HI
30 12411 042417           STA 0,@FP.LC
31 12412 034416           LDA 3,FP.LC
32 12413 020152           LDA 0,SCRLO
33 12414 040420           STA 0,FP.LO
34 12415 175400           INC 3,3
35 12416 175400           INC 3,3
36 12417 054405           STA 3,FP.GA
37 12420 001400           JMP 0,3
38                ;SET SCRATCH RELEASED SW AND EXIT
39 12421 102000 FP.05:    ADC 0,0
40 12422 040661           STA 0,FP.TK
41                FP.XI:  LCALL RETRN
42 12423 100210           RETRN-ASCRA*1B11+100010
43 12424 000000 FP.GA: 0
44 12425 000000 FP.HI: 0
45 12426 000040 FP.40: 040
46 12427 000000 FPS03: 0
47 12430 000000 FP.LC: 0
48 12431 000000 FP.TP: 0
49 12432 000000 FP.BG: 0
50 12433 000000 FP.EN: 0
51 12434 000000 FP.LO: 0
52 12435 000000 FP.LA: 0
53 12436 000000 FP.LP: 0
```

```
!0184 N3MRT
01                ;ERROR IN TEST DURING EXECUTION
02
03 12437 012440 FPERR:    FPERR+1
04 12440 054767           STA 3,FPS03       ;PRINT ERR HEADERS
05                        LCALL ERROI
06 12441 100350           ERROI-ASCRA*1B11+100010
07 12442 000401           JMP .+1
08 12443 020454 FP.EC:    LDA 0,F.TX1
09                        LCALL ERRTX
10 12444 100170           ERRTX-ASCRA*1B11+100010
11 12445 020763           LDA 0,FP.LC       ;FOLLOW UP WITH
12 12446 024766           LDA 1,FP.LO       ;TEST RELOC INFO
13 12447 030764           LDA 2,FP.EN
14                        LCALL ERROC
15 12450 100370           ERROC-ASCRA*1B11+100010
16 12451 000401           JMP .+1
17 12452 020460           LDA 0,F.TX2
18                        LCALL ERRTX
19 12453 100170           ERRTX-ASCRA*1B11+100010
20 12454 020753           LDA 0,FPS03       ;CONTINUE TYPE
21 12455 024761           LDA 1,FP.LP       ;RELOC
22 12456 030627           LDA 2,FP.ES       ;=0 PREV ERR
23                        LCALL ERROC
24 12457 100370           ERROC-ASCRA*1B11+100010
25 12460 000401           JMP .+1
26 12461 020464           LDA 0,F.TX3
27                        LCALL ERRTX
28 12462 100170           ERRTX-ASCRA*1B11+100010
29 12463 020742           LDA 0,FP.HI       ;HI SCRATCH REL.
30 12464 024746           LDA 1,FP.BG       ;TST BEFORE MOVE
31 12465 030737           LDA 2,FP.GA       ;WHERE STARTS REL.
32                        LCALL ERROC
33 12466 100370           ERROC-ASCRA*1B11+100010
34 12467 000401           JMP .+1
35 12470 020470           LDA 0,F.TX4
36                        LCALL ERRTX
37 12471 100170           ERRTX-ASCRA*1B11+100010
38 12472 034741           LDA 3,FP.EN
39 12473 021776           LDA 0,-2,3
40 12474 024740           LDA 1,FP.LO
41 12475 111000           MOV 0,2           ;RECALCULATE ADRS
42 12476 132400           SUB 1,2           ;OF FAILED DATA
43 12477 024152           LDA 1,SCRLO       ;IN RELATION TO
44 12500 133000           ADD 1,2           ;SCRATCH AT 1ST LEVEL
45 12501 101004           MOV 0,0,SZR       ;NOT DATA ERR SKP
46 12502 025000           LDA 1,0,2
47                        LCALL ERROC
48 12503 100370           ERROC-ASCRA*1B11+100010
49 12504 000675           JMP FP.4A         ; RELEASE SCRATCH
50 12505 102400           SUB 0,0
51 12506 042407           STA 0,@FPESX      ;SET ERR SW
52 12507 042407           STA 0,@FPTKX      ;CLR TEST K
53 12510 000713           JMP FP.XI         ;RETURN TO LINKR
54 12511 012512 FP.TT:    FP.TT+1
55 12512 012574           FLT01
56 12513 012741           FLT02
57 12514 177777           -1
58 12515 012305 FPESX: FP.ES
59 12516 012303 FPTKX: FP.TK
```

```
10185 N3MRT
01 12517 012520 F.TX1: .+1
02 12520 005215         .TXTE (<15><12>FP.LC    FP.LO   FP.EN(
03 12532 012533 F.TX2: .+1
04 12533 005215         .TXTE (<15><12>FPS03    FP.LP   FP.ES(
05 12545 012546 F.TX3: .+1
06 12546 005215         .TXTE (<15><12>FP.HI    FP.BG   FP.GA(
07 12560 012561 F.TX4: .+1
08 12561 005215         .TXTE (<15><12>L ADRS   DATA    S ADRS(
```

```
10186 N3MRT
01              ;FLOATING POINT  LOAD AND STORE TEST
02              ;SINGLE PRECISION OPERANDS
03
04 12574 000144 FLT01:  FLT1E-FLT01
05 12575 004401         JSR .+1
06 12576 020776         LDA 0,FLT01
07 12577 117000         ADD 0,3
08 12600 054534         STA 3,FLT1A     ;LOAD DATA START
09 12601 024772         LDA 1,FLT01-1   ;GET UPPER LOG ADDRESS
10 12602 166640         SUBOR 3,1       ;HALF SCRATCH 2 WORDS EACH
11 12603 121220         MOVZR 1,0       ;1 FOURTH SCRATCH
12 12604 117000         ADD 0,3
13 12605 054530         STA 3,FLT1B     ;START OF STORE BUFFER
14 12606 040530         STA 0,FLT1C
15 12607 105000         MOV 0,1
16 12610 030524         LDA 2,FLT1A
17 12611 124400         NEG 1,1
18              FLT1L:  LCALL ARANG     ;FILL 1 FOURTH
19 12612 100270         ARANG-ASCRA*1B11+100010
20 12613 041000         STA 0,0,2       ;OF BUFFER
21 12614 151400         INC 2,2         ;WITH RANDOM #'S
22 12615 125404         INC 1,1,SZR
23 12616 000774         JMP FLT1L
24 12617 020515         LDA 0,FLT1A     ;FILL BUFFER
25 12620 024515         LDA 1,FLT1B     ;IST PAIR
26 12621 030515         LDA 2,FLT1C     ;MOVE 1ST TO TEMP
27 12622 150640         NEGOR 2,2       ;STORE 2ND PAIR
28 12623 151400         INC 2,2
29 12624 176520         SUBZL 3,3       ;MOVE 1ST PAIR BACK
30 12625 175120         MOVZL 3,3       ;AND STORE THEM
31 12626 062374 FLT1M:  FLODS 0         ;LOAD FIRST 2 RAN #'S
32 12627 163000         ADD 3,0
33 12630 066174         FSTRS 1         ;STORE THEM BACK
34 12631 167000         ADD 3,1
35 12632 060375         FMFT            ;SAVE IN TEMP
36 12633 000401         JMP .+1
37 12634 062374         FLODS 0         ;GET NEXT 2 RAN #'S
38 12635 000401         JMP .+1
39 12636 066174         FSTRS 1
40 12637 167000         ADD 3,1
41 12640 060275         FMTF            ;RESTORE 1ST 2 RAN #'S
42 12641 000401         JMP .+1
43 12642 066174         FSTRS 1         ;STORE THEM AGAIN
44 12643 167000         ADD 3,1
45 12644 151404         INC 2,2,SZR
46 12645 000761         JMP FLT1M
```

```
!0187 N3MRT
01                ;COMPARE SINGLE PREC STORE COMPARE
02
03 12646 030466           LDA 2,FLT1A
04 12647 034466           LDA 3,FLT1B
05 12650 024466           LDA 1,FLT1C
06 12651 124640           NEGOR 1,1
07 12652 125400           INC 1,1
08 12653 044465           STA 1,FLT1E
09 12654 021000 FLT1T:    LDA 0,0,2       ;GET FIRST WORD OF PAIR
10 12655 025400           LDA 1,0,3       ;GET FIRST STORE
11 12656 122414           SUB# 1,0,SZR    ;NOT=IS ERR
12 12657 000445           JMP FLT1D
13 12660 010453           ISZ F1WDS
14 12661 025404           LDA 1,4,3       ;GET 2ND TIME STORED
15 12662 122414           SUB# 1,0,SZR
16 12663 000441           JMP FLT1D
17 12664 014447           DSZ F1WDS
18 12665 063077           HALT
19 12666 151400           INC 2,2
20 12667 175400           INC 3,3
21 12670 021000           LDA 0,0,2       ;GET 2ND WORD OF PAIR
22 12671 025400           LDA 1,0,3       ;FIRST TIME IT STORED
23 12672 122414           SUB# 1,0,SZR
24 12673 000431           JMP FLT1D
25 12674 010437           ISZ F1WDS
26 12675 025404           LDA 1,4,3       ;2ND TIME 2ND WORD
27 12676 122414           SUB# 1,0,SZR
28 12677 000425           JMP FLT1D
29 12700 014433           DSZ F1WDS
30 12701 063077           HALT
31 12702 151400           INC 2,2
32 12703 175400           INC 3,3
33 12704 021000           LDA 0,0,2       ;FIRST WORD 2ND PAIR
34 12705 025400           LDA 1,0,3
35 12706 122414           SUB# 1,0,SZR
36 12707 000415           JMP FLT1D
37 12710 021001           LDA 0,1,2       ;2ND WORD 2ND PAIR
38 12711 025401           LDA 1,1,3
39 12712 106414           SUB# 0,1,SZR
40 12713 000407           JMP FLT1D-2
41 12714 175620           INCZR 3,3       ;+3 TO STORE
42 12715 175500           INCL 3,3        ;BUFFER ADDRESS
43 12716 175400           INC 3,3
44 12717 010421           ISZ FLT1E       ;DONE ALL SKIP
45 12720 000734           JMP FLT1T
46                        LCALL RETU2     ;TEST PASSED RETURN
47 12721 100510           RETU2-ASCRA*1B11+100010
```

```
!0188 N3MRT
01                ;ERROR RETURN FROM THIS TEST
02
03 12722 151400           INC 2,2
04 12723 175400           INC 3,3
05 12724 040413 FLT1D:    STA 0,FLT1F
06 12725 020406           LDA 0,F1WDS
07 12726 103120           ADDZL 0,0
08 12727 117000           ADD 0,3
09 12730 020407           LDA 0,FLT1F
10 12731 054406           STA 3,FLT1F     ;SAVE REAL ADRS OF ERR
11                        LCALL ERRET
12 12732 100470           ERRET-ASCRA*1B11+100010
13
14 12733 000000 F1WDS: 0
15 12734 000000 FLT1A:  0
16 12735 000000 FLT1B:  0
17 12736 000000 FLT1C:  0
18 12737 000000 FLT1F: 0
19 12740 000000 FLT1E:  0
```

```
!0189 N3MRT
01                ;FLOATING POINT  LOAD AND STORE TEST
02                ;DOUBLE PRECISION OPERANDS
03
04 12741 000166 FLT02:  FLT2E-FLT02
05 12742 004401         JSR .+1
06 12743 020776         LDA 0,FLT02
07 12744 117000         ADD 0,3
08 12745 054556         STA 3,FLT2A      ;LOAD DATA START
09 12746 024772         LDA 1,FLT02-1    ;GET UPPER LOG ADDRESS
10 12747 166640         SUBOR 3,1        ;HALF SCRATCH 2 WORDS EACH
11 12750 121220         MOVZR 1,0        ;1 FOURTH SCRATCH
12 12751 117000         ADD 0,3
13 12752 054552         STA 3,FLT2B      ;START OF STORE BUFFER
14 12753 040552         STA 0,FLT2C
15 12754 105000         MOV 0,1
16 12755 030546         LDA 2,FLT2A
17 12756 124400         NEG 1,1
18                FLT2L:  LCALL ARANG      ;FILL 1 FOURTH
19 12757 100270         ARANG-ASCRA*1B11+100010
20 12760 041000         STA 0,0,2        ;OF BUFFER
21 12761 151400         INC 2,2          ;WITH RANDOM #'S
22 12762 125404         INC 1,1,SZR
23 12763 000774         JMP FLT2L
24 12764 020537         LDA 0,FLT2A      ;FILL BUFFER
25 12765 024537         LDA 1,FLT2B      ;1ST 4
26 12766 030537         LDA 2,FLT2C      ;MOVE 1ST TO TEMP
27 12767 150640         NEGOR 2,2        ;STORE 2ND 4
28 12770 151240         MOVOR 2,2
29 12771 151400         INC 2,2
30 12772 176520         SUBZL 3,3        ;MOVE 1ST 4 BACK
31 12773 177120         ADDZL 3,3        ;AND STORE THEM
32 12774 062375 FLT2M:  FLODD 0          ;LOAD FIRST 4 RAN #'S
33 12775 163000         ADD 3,0
34 12776 066175         FSTRD 1          ;STORE THEM BACK
35 12777 167000         ADD 3,1
36 13000 060375         FMFT             ;SAVE IN TEMP
37 13001 000401         JMP .+1
38 13002 062375         FLODD 0          ;GET NEXT 4 RAN #'S
39 13003 000401         JMP .+1
40 13004 066175         FSTRD 1
41 13005 167000         ADD 3,1
42 13006 060275         FMTF             ;RESTORE 1ST 4 RAN #'S
43 13007 000401         JMP .+1
44 13010 066175         FSTRD 1          ;STORE THEM AGAIN
45 13011 167000         ADD 3,1
46 13012 151404         INC 2,2,SZR
47 13013 000761         JMP FLT2M
```

```
!0190 N3MRT
01                ;COMPARE DOUBLE PREC STORE COMPARE
02
03 13014 030507         LDA 2,FLT2A
04 13015 034507         LDA 3,FLT2B
05 13016 024507         LDA 1,FLT2C
06 13017 124640         NEGOR 1,1
07 13020 125240         MOVOR 1,1
08 13021 125400         INC 1,1
09 13022 044505         STA 1,FLT2E
10 13023 102120 FLT2T:  ADCZL 0,0
11 13024 040475         STA 0,FLT2S
12 13025 021000         LDA 0,0,2        ;GET FIRST WORD OF 4
13 13026 025400         LDA 1,0,3        ;GET FIRST STORE
14 13027 122414         SUB# 1,0,SZR     ;NOT=IS ERR
15 13030 000461         JMP FLT2D
16 13031 010471         ISZ F2WDS
17 13032 025410         LDA 1,10,3       ;GET 2ND TIME STORED
18 13033 122414         SUB# 1,0,SZR
19 13034 000455         JMP FLT2D
20 13035 014465         DSZ F2WDS
21 13036 063077         HALT
22 13037 151400         INC 2,2
23 13040 175400         INC 3,3
24 13041 021000         LDA 0,0,2        ;GET 2ND WORD OF 4
25 13042 025400         LDA 1,0,3        ;FIRST TIME IT STORED
26 13043 122414         SUB# 1,0,SZR
27 13044 000445         JMP FLT2D
28 13045 010455         ISZ F2WDS
29 13046 025410         LDA 1,10,3       ;2ND TIME 2ND WORD
30 13047 122414         SUB# 1,0,SZR
31 13050 000441         JMP FLT2D
32 13051 014451         DSZ F2WDS
33 13052 063077         HALT
34 13053 151400         INC 2,2
35 13054 175400         INC 3,3
36 13055 010444         ISZ FLT2S
37 13056 000747         JMP FLT2T+2
38 13057 102120         ADCZL 0,0
39 13060 040441         STA 0,FLT2S
```

```
!0191 N3MRT
01 13061 021000 FLT2U:  LDA 0,0,2       ;FIRST WORD 2ND 4
02 13062 025400         LDA 1,0,3
03 13063 122414         SUB# 1,0,SZR
04 13064 000425         JMP FLT2D
05 13065 021001         LDA 0,1,2       ;2ND WORD 2ND 4
06 13066 025401         LDA 1,1,3
07 13067 106414         SUB# 0,1,SZR
08 13070 000417         JMP FLT2D-2
09 13071 102520         SUBZL 0,0
10 13072 101120         MOVZL 0,0
11 13073 113000         ADD 0,2
12 13074 117000         ADD 0,3
13 13075 010424         ISZ FLT2S
14 13076 000763         JMP FLT2U       ;DO NXT 2 2ND 4
15 13077 112400         SUB 0,2
16 13100 112400         SUB 0,2
17 13101 175400         INC 3,3
18 13102 175620         INCZR 3,3       ;+5 TO STORE
19 13103 175500         INCL 3,3        ;BUFFER ADDRESS
20 13104 010423         ISZ FLT2E       ;DONE ALL SKIP
21 13105 000716         JMP FLT2T
22                      LCALL RETU2     ;TEST PASSED RETURN
23 13106 100510         RETU2=ASCRA*1811+100010
24
25              ;ERROR RETURN FROM THIS TEST
26
27 13107 151400         INC 2,2
28 13110 175400         INC 3,3
29 13111 040415 FLT2D:  STA 0,FLT2F
30 13112 020410         LDA 0,F2WDS     ;CALC REAL FAILURE
31 13113 103120         ADDZL 0,0       ;ADDRESS
32 13114 101120         MOVZL 0,0       ;COULD BE OFF BY 8
33 13115 117000         ADD 0,3
34 13116 020410         LDA 0,FLT2F
35 13117 054407         STA 3,FLT2F
36                      LCALL ERRET
37 13120 100470         ERRET=ASCRA*1811+100010
38 13121 000000 FLT2S: 0
39
40 13122 000000 F2WDS: 0
41 13123 000000 FLT2A:  0
42 13124 000000 FLT2B:  0
43 13125 000000 FLT2C:  0
44 13126 000000 FLT2F:  0
45 13127 000000 FLT2E:  0
```

```
!0192 N3MRT
01              ;       .TITL MUDVT
02              ;MULTIPLY DIVIDE TEST COMPATABLE WIITH
03              ;THE DIAGNOSTIC LINKER
04                      .MACRO STORE
05                      JSR @IXTOR
06              %
07                      .MACRO MULCK
08                      JSR @IMCK
09              %
10                      .MACRO DIVCK
11                      JSR @IDCK
12              %
13                      .MACRO DIVER
14                      JSR @EDIV
15              %
16                      .MACRO MULER
17                      JSR @EMUL
18              %
19                      NEXTT MUDIO
20        013130 LMEML=.
21        000162        .LOC LPG0
22 00162 013133         MUDIO
23        000163 LPG0=.
24        013130        .LOC LMEML
25 13130 000000         0       ;TEST PASS CTR
26 13131 000000         0       ;TEST ERROR CTR
27 13132 000000         0       ;INTERRUPT TIMEOUT SWITCH
28
```

```
!0193 N3MRT
01 13133 013152 MUDI0:  MUDI1
02 13134 013163         MDA00
03 13135 000000         0
04 13136 000000         0
05 13137 177777         -1
06 13140 176000         176000
07 13141 014030         MDTEX
08 13142 014030         MDTEX
09 13143 052515         .TXTE (MUL/DIV TEST(
10       127714
11       144504
12       120126
13       142724
14       152123
15       000000
16
17 13152 102000 MUDI1:  ADC 0,0
18 13153 040762         STA 0,MUDI0+2
19 13154 126400         SUB 1,1
20 13155 152400         SUB 2,2
21 13156 073301         DOCP 2,01
22 13157 125005         MOV 1,1,SNR      ;SKP=MUL/DIV EXISTS
23 13160 001400         JMP 0,3          ;NO MUL DIV EXIT
24 13161 040754         STA 0,MUDI0+2
25 13162 001400         JMP 0,3
```

```
!0194 N3MRT
01               MDA00:  SETUP   ;0*0=0
02                       LCALL SETUL
03 13163 100410          SETUL-ASCRA*1B11+100010
04                       STORE
05 13164 006530          JSR @IXTOR
06 13165 000000          0                ;AC0
07 13166 000000          0                ;AC1
08 13167 000000          0                ;AC2
09 13170 006525          JSR @HMUL
10                       MULCK
11 13171 006525          JSR @IMCK
12 13172 000000          0
13 13173 000000          0
14 13174 000000          0
15                       LOOP
16                       LCALL LLOOP
17 13175 100430          LLOOP-ASCRA*1B11+100010
18
19                   ;A01:
20                       SETUP
21                       LCALL SETUL
22 13176 100410          SETUL-ASCRA*1B11+100010
23                       STORE
24 13177 006515          JSR @IXTOR
25 13200 000000          0
26 13201 000000          0
27 13202 177777          -1
28 13203 006512          JSR @HMUL
29                       MULCK
30 13204 006512          JSR @IMCK
31 13205 000000          0
32 13206 000000          0
33 13207 177777          -1
34                       LOOP
35                       LCALL LLOOP
36 13210 100430          LLOOP-ASCRA*1B11+100010
37
```

```
!0195 N3MRT
01                ;A02:
02                        SETUP
03                        LCALL SETUL
04 13211 100410           SETUL-ASCRA*1B11+100010
05                        STORE
06 13212 006502           JSR @IXTOR
07 13213 000000           0
08 13214 177777           -1
09 13215 000000           0
10 13216 006477           JSR @HMUL
11                        MULCK
12 13217 006477           JSR @IMCK
13 13220 000000           0
14 13221 000000           0
15 13222 000000           0
16                        LOOP
17                        LCALL LLOOP
18 13223 100430           LLOOP-ASCRA*1B11+100010
19
20                ;A03:
21                        SETUP
22                        LCALL SETUL
23 13224 100410           SETUL-ASCRA*1B11+100010
24                        STORE
25 13225 006467           JSR @IXTOR
26 13226 177777           -1
27 13227 000000           0
28 13230 000000           0
29 13231 006464           JSR @HMUL
30                        MULCK
31 13232 006464           JSR @IMCK
32 13233 000000           0
33 13234 177777           -1
34 13235 000000           0
35                        LOOP
36                        LCALL LLOOP
37 13236 100430           LLOOP-ASCRA*1B11+100010
```

```
!0196 N3MRT
01                ;A04:
02                        SETUP
03                        LCALL SETUL
04 13237 100410           SETUL-ASCRA*1B11+100010
05                        STORE
06 13240 006454           JSR @IXTOR
07 13241 000000           0
08 13242 000001           1
09 13243 177777           -1
10 13244 006451           JSR @HMUL
11                        MULCK
12 13245 006451           JSR @IMCK
13 13246 000000           0
14 13247 177777           -1
15 13250 177777           -1
16                        LOOP
17                        LCALL LLOOP
18 13251 100430           LLOOP-ASCRA*1B11+100010
19
20                ;A05:
21                        SETUP
22                        LCALL SETUL
23 13252 100410           SETUL-ASCRA*1B11+100010
24                        STORE
25 13253 006441           JSR @IXTOR
26 13254 000000           0
27 13255 177777           -1
28 13256 000001           1
29 13257 006436           JSR @HMUL
30                        MULCK
31 13260 006436           JSR @IMCK
32 13261 000000           0
33 13262 177777           -1
34 13263 000001           1
35                        LOOP
36                        LCALL LLOOP
37 13264 100430           LLOOP-ASCRA*1B11+100010
38
```

```
!0197 N3MRT
01                ;A06:
02                        SETUP
03                        LCALL SETUL
04 13265 100410           SETUL-ASCRA*1B11+100010
05                        STORE
06 13266 006426           JSR @IXTOR
07 13267 177777           -1
08 13270 177777           -1
09 13271 000000           0
10 13272 006423           JSR @HMUL
11                        MULCK
12 13273 006423           JSR @IMCK
13 13274 000000           0
14 13275 177777           -1
15 13276 000000           0
16                        LOOP
17                        LCALL LLOOP
18 13277 100430           LLOOP-ASCRA*1B11+100010
19
20                ;A07:
21                        SETUP
22                        LCALL SETUL
23 13300 100410           SETUL-ASCRA*1B11+100010
24                        STORE
25 13301 006413           JSR @IXTOR
26 13302 000001           1
27 13303 000002           2
28 13304 000002           2
29 13305 006410           JSR @HMUL
30                        MULCK
31 13306 006410           JSR @IMCK
32 13307 000000           0
33 13310 000005           5
34 13311 000002           2
35                        LOOP
36                        LCALL LLOOP
37 13312 100430           LLOOP-ASCRA*1B11+100010
38 13313 000406           JMP MDA08
39 13314 013522 IXTOR:    SXTOR
40 13315 013745 HMUL:     XHMUL
41 13316 013466 IMCK:     MCK
42 13317 013467 IDCK:     DCK
43 13320 013757 HDIV:     XHDIV
```

```
!0198 N3MRT
01                MDA08:  SETUP   ;EXPECT A DIVIDE ERROR
02                        LCALL SETUL
03 13321 100410           SETUL-ASCRA*1B11+100010
04                        STORE           ;AND NO CHANGE TO ACS.
05 13322 006772           JSR @IXTOR
06 13323 000000           0
07 13324 000000           0
08 13325 000000           0
09 13326 006772           JSR @HDIV
10                        DIVCK
11 13327 006770           JSR @IDCK
12 13330 000000           0
13 13331 000000           0
14 13332 000000           0
15                        LOOP
16                        LCALL LLOOP
17 13333 100430           LLOOP-ASCRA*1B11+100010
18
19                ;A09:
20                        SETUP   ;EXPECT DIVIDE ERROR.
21                        LCALL SETUL
22 13334 100410           SETUL-ASCRA*1B11+100010
23                        STORE           ;NO CHANGE TO ACS.
24 13335 006757           JSR @IXTOR
25 13336 177777           -1
26 13337 177777           -1
27 13340 177777           -1
28 13341 006757           JSR @HDIV
29                        DIVCK
30 13342 006755           JSR @IDCK
31 13343 177777           -1
32 13344 177777           -1
33 13345 177777           -1
34                        LOOP
35                        LCALL LLOOP
36 13346 100430           LLOOP-ASCRA*1B11+100010
37
```

```
!0199 N3MRT
01                ;A10:
02                        SETUP   ;DIVIDE ERROR SHOULD
03                        LCALL SETUL
04 13347 100410           SETUL-ASCRA*1B11+100010
05                        STORE           ;SET THE CARRY
06 13350 006744           JSR @IXTOR
07 13351 177777           -1
08 13352 177777           -1
09 13353 177777           -1
10 13354 101020           MOVZ 0,0
11 13355 006743           JSR @HDIV
12 13356 101003           MOV 0,0,SNC
13                        DIVER
14 13357 006505           JSR @EDIV
15                        LOOP
16                        LCALL LLOOP
17 13360 100430           LLOOP-ASCRA*1B11+100010
18
19                ;A11:
20                        SETUP
21                        LCALL SETUL
22 13361 100410           SETUL-ASCRA*1B11+100010
23                        STORE
24 13362 006732           JSR @IXTOR
25 13363 000000           0
26 13364 000000           0
27 13365 000001           1
28 13366 006732           JSR @HDIV
29                        DIVCK
30 13367 006730           JSR @IDCK
31 13370 000000           0
32 13371 000000           0
33 13372 000001           1
34                        LOOP
35                        LCALL LLOOP
36 13373 100430           LLOOP-ASCRA*1B11+100010
```

```
!0200 N3MRT
01                ;A12:
02                        SETUP   ;NO DIVIDE ERROR SHOULD
03                        LCALL SETUL
04 13374 100410           SETUL-ASCRA*1B11+100010
05                        STORE           ;OCCURE AND CARRY SHOULD
06 13375 006717           JSR @IXTOR
07 13376 000000           0               ;BE A ZERO.
08 13377 000000           0
09 13400 000001           1
10 13401 101040           MOVO 0,0
11 13402 006716           JSR @HDIV
12 13403 101002           MOV 0,0,SZC
13                        DIVER
14 13404 006460           JSR @EDIV
15                        LOOP
16                        LCALL LLOOP
17 13405 100430           LLOOP-ASCRA*1B11+100010
18
19                ;A13:
20                        SETUP   ;4/2=2
21                        LCALL SETUL
22 13406 100410           SETUL-ASCRA*1B11+100010
23                        STORE
24 13407 006705           JSR @IXTOR
25 13410 000000           0
26 13411 000004           4
27 13412 000002           2
28 13413 006705           JSR @HDIV
29                        DIVCK
30 13414 006703           JSR @IDCK
31 13415 000000           0
32 13416 000002           2
33 13417 000002           2
34                        LOOP
35                        LCALL LLOOP
36 13420 100430           LLOOP-ASCRA*1B11+100010
37
```

```
!0201 N3MRT
01                  :A14:
02                          SETUP   ;CHECK REMAINDER
03                          LCALL SETUL
04 13421 100410             SETUL-ASCRA*1B11+100010
05                          STORE
06 13422 006672             JSR @IXTOR
07 13423 000000             0
08 13424 077777             77777
09 13425 100000             100000
10 13426 006672             JSR @HDIV
11                          DIVCK
12 13427 006670             JSR @IDCK
13 13430 077777             77777
14 13431 000000             0
15 13432 100000             100000
16                          LOOP
17                          LCALL LLOOP
18 13433 100430             LLOOP-ASCRA*1B11+100010
19
20                  :A15:
21                          SETUP
22                          LCALL SETUL
23 13434 100410             SETUL-ASCRA*1B11+100010
24                          STORE
25 13435 006657             JSR @IXTOR
26 13436 000000             0
27 13437 177777             -1
28 13440 177777             -1
29 13441 006657             JSR @HDIV
30                          DIVCK
31 13442 006655             JSR @IDCK
32 13443 000000             0
33 13444 000001             1
34 13445 177777             -1
35                          LOOP
36                          LCALL LLOOP
37 13446 100430             LLOOP-ASCRA*1B11+100010
```

```
!0202 N3MRT
01                  :A16:
02                          SETUP
03                          LCALL SETUL
04 13447 100410             SETUL-ASCRA*1B11+100010
05                          STORE
06 13450 006644             JSR @IXTOR
07 13451 000001             1
08 13452 000000             0
09 13453 000002             2
10 13454 006644             JSR @HDIV
11                          DIVCK
12 13455 006642             JSR @IDCK
13 13456 000000             0
14 13457 100000             100000
15 13460 000002             2
16                          LOOP
17                          LCALL LLOOP
18 13461 100430             LLOOP-ASCRA*1B11+100010
19
20                  :A17:
21 13462 002401             JMP @.+1
22 13463 013540             MTST
23 13464 013776 EDIV:       DERR
24 13465 013771 EMUL:       MERR
```

```
!0203 N3MRT
01
02 13466 101021 MCK:    MOVZ 0,0,SKP    ;CHECK MUL RESULT
03 13467 101040 DCK:    MOVO 0,0        ;CHECK DIV RESULT
04 13470 054444         STA 3,XCKRET
05 13471 035402         LDA 3,2,3
06 13472 156414         SUB# 2,3,SZR
07 13473 000411         JMP CK1         ;AC2 WRONG
08 13474 034440         LDA 3,XCKRET
09 13475 031400         LDA 2,0,3
10 13476 035401         LDA 3,1,3
11 13477 112415         SUB# 0,2,SNR    ;CHECK AC0
12 13500 136414         SUB# 1,3,SZR    ;CHECK AC1
13 13501 000403         JMP CK1         ;ERROR
14 13502 034432         LDA 3,XCKRET
15 13503 001403         JMP 3,3
16 13504 034430 CK1:    LDA 3,XCKRET
17 13505 021400         LDA 0,0,3
18 13506 025401         LDA 1,1,3
19 13507 031402         LDA 2,2,3
20 13510 040550         STA 0,OKAC
21 13511 044550         STA 1,OKMQ
22 13512 050550         STA 2,OKMD
23
24 13513 101002 CK2:    MOV 0,0,SZC     ;IF C(CARRY)=0 ITS MUL ERR
25 13514 000403         JMP .+3         ;OTHERWISE ITS A DIVIDE ER
26                      MULER
27 13515 006750         JSR @EMUL
28 13516 101001         MOV 0,0,SKP
29                      DIVER
30 13517 006745         JSR @EDIV
31 13520 034414         LDA 3,XCKRET
32 13521 001403         JMP 3,3
33
```

```
!0204 N3MRT
01 13522 021400 SXTOR:  LDA 0,0,3       ;PICK UP ARGUEMENTS
02 13523 025401         LDA 1,1,3       ;AND STORE IN ORIG
03 13524 031402         LDA 2,2,3       ;NUMBERS TABLE.
04 13525 042404         STA 0,@XOAC
05 13526 046404         STA 1,@XOMQ
06 13527 052404         STA 2,@XOMD
07 13530 001403         JMP 3,3
08 13531 013735 XOAC:   OAC
09 13532 013736 XOMQ:   OMQ
10 13533 013737 XOMD:   OMD
11 13534 000000 XCKRET: 0
12 13535 013745 IHMUL: XHMUL
13 13536 013742 IHMD: HMD
14 13537 013757 IHDIV: XHDIV
15
```

```
!0205 N3MRT
01                MTST:   SETUP   ;CHECK MULTIPLY WITH
02                        LCALL SETUL
03 13540 100410           SETUL-ASCRA*1B11+100010
04 13541 004522           JSR RAN
05 13542 006773           JSR @IHMUL
06 13543 022766           LDA 0,@XOAC
07 13544 026766           LDA 1,@XOMQ
08 13545 032766           LDA 2,@XOMD
09 13546 004537           JSR XMUL          ;PROGRAMED MULTIPLY
10 13547 036767           LDA 3,@IHMD
11 13550 156414           SUB# 2,3,SZR
12 13551 000405           JMP .+5
13 13552 030566           LDA 2,HAC
14 13553 034566           LDA 3,HMQ
15 13554 112415           SUB# 0,2,SNR
16 13555 136414           SUB# 1,3,SZR
17                        MULER             ;MULTIPLY FAILED
18 13556 006707           JSR @EMUL
19                        LOOP
20                        LCALL LLOOP
21 13557 100430           LLOOP-ASCRA*1B11+100010
22
23                DTST:   SETUP   ;CHECK DIVIDE WITH
24                        LCALL SETUL
25 13560 100410           SETUL-ASCRA*1B11+100010
26 13561 004502           JSR RAN ;RANDOM NUMBERS.
27 13562 006755           JSR @IHDIV        ;HARDWARE DIVIDE
28 13563 020552           LDA 0,OAC
29 13564 024552           LDA 1,OMQ
30 13565 030552           LDA 2,OMD
31 13566 004533           JSR XDIV          ;PROGRAMED DIVIDE
32 13567 034553           LDA 3,HMD
33 13570 156414           SUB# 2,3,SZR
34 13571 000405           JMP .+5
35 13572 030546           LDA 2,HAC
36 13573 034546           LDA 3,HMQ
37 13574 112415           SUB# 0,2,SNR
38 13575 136414           SUB# 1,3,SZR
39                        DIVER             ;DIVIDE FAILED
40 13576 006666           JSR @EDIV
41                        LOOP
42                        LCALL LLOOP
43 13577 100430           LLOOP-ASCRA*1B11+100010
```

```
!0206 N3MRT
01 13600 020454 MDTST:  LDA 0,M100        ;MULTIPLY DIVIDE TEST
02 13601 040425         STA 0,FOB
03                      SETUP
04                      LCALL SETUL
05 13602 100410         SETUL-ASCRA*1B11+100010
06 13603 004460         JSR RAN
07 13604 004553         JSR XHDIV         ;HARD DIVIDE
08 13605 040450         STA 0,D0
09 13606 044450         STA 1,D1
10 13607 050450         STA 2,D2
11 13610 004535         JSR XHMUL         ;HARD MULTIPLY
12 13611 034526         LDA 3,OMD
13 13612 156414         SUB# 2,3,SZR
14 13613 000414         JMP MDT2
15 13614 030521         LDA 2,OAC
16 13615 034521         LDA 3,OMQ
17 13616 112415         SUB# 0,2,SNR
18 13617 136414         SUB# 1,3,SZR
19 13620 000407         JMP MDT2
20                MDT1: LOOP
21                      LCALL LLOOP
22 13621 100430         LLOOP-ASCRA*1B11+100010
23 13622 010404         ISZ FOB
24 13623 000757         JMP MDTST+2
25 13624 002401         JMP @.+1
26 13625 014030         MDTEX
27 13626 000000 FOB: 0
28 13627 020506 MDT2:   LDA 0,OAC         ;EITHER MUL OR DIV
29 13630 024506         LDA 1,OMQ         ;FAILED , TYR TO FIND
30 13631 030506         LDA 2,OMD         ;WHICH ONE.
31 13632 004467         JSR XDIV
32 13633 034422         LDA 3,D0
33 13634 116414         SUB# 0,3,SZR
34 13635 000407         JMP MDT3
35 13636 034420         LDA 3,D1
36 13637 136414         SUB# 1,3,SZR
37 13640 000404         JMP MDT3
38 13641 034416         LDA 3,D2
39 13642 156415         SUB# 2,3,SNR
40 13643 000403         JMP .+3
41                MDT3: DIVER             ;ITS A DIVIDE ERROR
42 13644 006620         JSR @EDIV
43 13645 000754         JMP MDT1
44 13646 040467         STA 0,OAC
45 13647 044467         STA 1,OMQ
46 13650 050467         STA 2,OMD
47 13651 004434         JSR XMUL
48                      MULER             ;ITS A MULTIPLY ERR
49 13652 006613         JSR @EMUL
50 13653 000746         JMP MDT1
51 13654 177700 M100:   -100
52 13655 000000 D0: 0
53 13656 000000 D1: 0
54 13657 000000 D2: 0
```

```
!0207 N3MRT
01 13660 000000 OKAC: 0
02 13661 000000 OKMQ: 0
03 13662 000000 OKMD: 0
04
05 13663 054421 RAN:    STA 3,RANRET    ;GET RANDOM OPERATORS
06              RAN1:   LCALL ARANG
07 13664 100270         ARANG-ASCRA*1B11+100010
08 13665 110700         NEGS 0,2        ;FORM MQ+MD
09 13666 105120         MOVZL 0,1
10 13667 127100         ADDL 1,1
11 13670 107300         ADDS 0,1
12 13671 112415         SUB# 0,2,SNR
13 13672 000772         JMP RAN1        ;REJECT IF AC=MD
14 13673 142432         SUBZ# 2,0,SZC
15 13674 115001         MOV 0,3,SKP
16 13675 000403         JMP .+3
17 13676 141000         MOV 2,0         ;MAKE AC LESS THAN
18 13677 171000         MOV 3,2         ;MD IN ALL CASES.
19 13700 040435 RAN2:   STA 0,OAC       ;STORE IN ORIGIONAL
20 13701 044435         STA 1,OMQ       ;NUMBER BLOCK..
21 13702 050435         STA 2,OMD
22 13703 002401         JMP @RANRET
23 13704 000000 RANRET: 0
24 13705 054436 XMUL:   STA 3,MSAV      ;PROGRAMED MULTIPLY
25 13706 034436         LDA 3,M20
26 13707 125203         MOVR 1,1,SNC
27 13710 101201         MOVR 0,0,SKP
28 13711 143220         ADDZR 2,0
29 13712 175404         INC 3,3,SZR
30 13713 000774         JMP XMUL+2
31 13714 125260         MOVCR 1,1
32 13715 040743 XMUL1:  STA 0,OKAC      ;STORE RESULTS
33 13716 044743         STA 1,OKMQ
34 13717 050743         STA 2,OKMD
35 13720 002423         JMP @MSAV
36
```

```
!0208 N3MRT
01 13721 054422 XDIV:   STA 3,MSAV      ;PROGRAMED DIVIDE
02 13722 142432         SUBZ# 2,0,SZC
03 13723 001400         JMP 0,3         ;OV EXIT
04 13724 034420         LDA 3,M20
05 13725 125120         MOVZL 1,1
06 13726 101100 XDIV1:  MOVL 0,0
07 13727 142412         SUB# 2,0,SZC
08 13730 142400         SUB 2,0
09 13731 125100         MOVL 1,1
10 13732 175404         INC 3,3,SZR
11 13733 000773         JMP XDIV1
12 13734 000761         JMP XMUL1       ;STORE RESULTS.
13 13735 000000 OAC: 0
14 13736 000000 OMQ: 0
15 13737 000000 OMD: 0
16 13740 000000 HAC: 0
17 13741 000000 HMQ: 0
18 13742 000000 HMD: 0
19 13743 000000 MSAV: 0
20 13744 177760 M20: -20
```

```
!0209 N3MRT
01 13745 054423 XHMUL:  STA 3,XHRET     ;HARDWARE MULTIPLY
02 13746 073301         DOCP 2,01
03 13747 004402         JSR .+2
04 13750 013750         .
05 13751 040767         STA 0,HAC
06 13752 020776         LDA 0,.-2
07 13753 116434         SUBZ# 0,3,SZR
08 13754 000415         JMP MERR
09 13755 020763         LDA 0,HAC
10 13756 000406         JMP XHCOM
11
12 13757 054411 XHDIV:  STA 3,XHRET     ;HARDWARE DIVIDE
13 13760 176440         SUBO 3,3
14 13761 073101         DOCS 2,01
15 13762 175004         MOV 3,3,SZR
16 13763 000413         JMP DERR
17 13764 040754 XHCOM:  STA 0,HAC       ;STORE HARDWARE RESULTS.
18 13765 044754         STA 1,HMQ
19 13766 050754         STA 2,HMD
20 13767 002401         JMP @XHRET
21 13770 000000 XHRET: 0
22
23 13771 054412 MERR:   STA 3,MERET     ;MULTIPLY ERROR
24 13772 101001         MOV 0,0,SKP
25 13773 014032         MULHED          ;HEADING FOR MUL.
26 13774 020777         LDA 0,.-1
27 13775 000407         JMP GBR
28
29 13776 054405 DERR:   STA 3,MERET     ;DIVIDE ERROR
30 13777 101001         MOV 0,0,SKP
31 14000 014044         DIVHED
32 14001 020777         LDA 0,.-1
33 14002 000402         JMP GBR
34 14003 000000 MERET: 0
```

```
!0210 N3MRT
01 14004 040425 GBR:    STA 0,XHDR      ;PRINT GOOD-BAD RESULTS
02                      LCALL ERROI
03 14005 100350         ERROI-ASCRA*1B11+100010
04 14006 000403         JMP .+3
05 14007 101001         MOV 0,0,SKP
06 14010 020421         LDA 0,XHDR
07                      LCALL ERRTX
08 14011 100170         ERRTX-ASCRA*1B11+100010
09 14012 020723         LDA 0,OAC
10 14013 024723         LDA 1,OMQ
11 14014 030723         LDA 2,OMD
12                      LCALL ERROC
13 14015 100370         ERROC-ASCRA*1B11+100010
14 14016 000401         JMP .+1
15 14017 020721         LDA 0,HAC
16 14020 024721         LDA 1,HMQ
17 14021 030721         LDA 2,HMD
18                      LCALL ERROC
19 14022 100370         ERROC-ASCRA*1B11+100010
20 14023 020635         LDA 0,OKAC
21 14024 024635         LDA 1,OKMQ
22 14025 030635         LDA 2,OKMD
23                      LCALL ERROC
24 14026 100370         ERROC-ASCRA*1B11+100010
25 14027 000401         JMP .+1
26              MDTEX: LCALL RETRN
27 14030 100210         RETRN-ASCRA*1B11+100010
28 14031 000000 XHDR: 0
29 14032 005215 MULHED: .TXTE ↓<15><12>
30 14033 052515 MUL<15><12>AC0  AC1     AC2 ↓
31 14044 005215 DIVHED: .TXTE ↓<15><12>
32 14045 144504 DIV<15><12>AC0  AC1     AC2 ↓
```

```
!0211 N3MRT
01                      :                       .TITL PGDSK
02                      ; 6063/64 DISK TEST COMPATIBLE WITH
03                      ;THE DIAGNOSTIC LINKER
04
05              000026 PDDSK=26
06                      ;DEFINITIONS BLOCK TO LINKER
07                              NEXTT PD.00
08              014056 LMEML=.
09              000163          .LOC LPG0
10 00163 014061                 PD.00
11              000164 LPG0=.
12              014056          .LOC LMEML
13 14056 000000                 0       ;TEST PASS CTR
14 14057 000000                 0       ;TEST ERROR CTR
15 14060 000000                 0       ;INTERRUPT TIMEOUT SWITCH
16 14061 014154 PD.00:          PD.01
17 14062 014254                 PD.02
18 14063 000000                 0
19 14064 000000                 0
20 14065 177777                 -1
21 14066 176000                 176000
22 14067 015005                 PD.XI
23 14070 015005                 PD.XI
24 14071 030066 6063/64 DISK(PRI)!
25
26
27                      ;REPORT TO OPERATOR DISK SIZING INFORMATION
28 14102 000004                 4
29 14103 014104 PD.1R:  .+1
30 14104 020440                 LDA 0,PD3TX
31                              LCALL ERRTX
32 14105 100170                 ERRTX-ASCRA*1B11+100010
33 14106 126400                 SUB 1,1
34 14107 044441                 STA 1,PD.NS+1   ;INIT DSK CNTR
35 14110 030542 PD1RL:  LDA 2,PDTXA
36 14111 133000                 ADD 1,2
37 14112 021000                 LDA 0,0,2       ;CK IF THIS DRV EXISTS
38 14113 100015                 COM# 0,0,SNR    ;SKP=EXISTS
39 14114 000411                 JMP PD1RR       ;NON-EXISTS
40 14115 040434                 STA 0,PD.NS+2   ;SAVE SIZE FLG
41                              LCALL PCRLF     ;CR/LF
42 14116 100110                 PCRLF-ASCRA*1B11+100010
43                              LCALL PZOCT     ;DISK # 0
44 14117 100130                 PZOCT-ASCRA*1B11+100010
45 14120 024431                 LDA 1,PD.NS+2   ;TRACKS =128 OR 64
46 14121 125400                 INC 1,1
47                              LCALL PDECI     ;PRINT IT
48 14122 100150                 PDECI-ASCRA*1B11+100010
49 14123 024424                 LDA 1,PD.NS     ;# SECTORS = 32
50                              LCALL PDECI
51 14124 100150                 PDECI-ASCRA*1B11+100010
52 14125 010423 PD1RR:  ISZ PD.NS+1
53 14126 024422                 LDA 1,PD.NS+1   ;CURRENT DRV #
54 14127 030753                 LDA 2,PD.1R-1   ;=4
55 14130 146414                 SUB# 2,1,SZR    ;SKP=DONE ALL
56 14131 000757                 JMP PD1RL       ;CK NEXT DRV
57 14132 020413                 LDA 0,PD3TX+1
58 14133 040727                 STA 0,PD.00+1
59 14134 030412                 LDA 2,PD3TX+2
60 14135 024506                 LDA 1,PDMK1     ;-84
```

```
 0212 N3MRT
01 14136 102400                 SUB 0,0
02 14137 041000                 STA 0,0,2
03 14140 151400                 INC 2,2
04 14141 125404                 INC 1,1,SZR
05 14142 000775                 JMP .-3         ;CLEAR DISK CONTROL TABLES
06 14143 000511                 JMP PD.02
07 14144 015145 PD3TX:  PDTX3
08 14145 014254                 PD.02
09 14146 015172                 PDCTB
10 14147 000040 PD.NS:  32.
11 14150 000000                 0
12 14151 000000                 0       ;SAVED # OF TRACKS THIS DRIVE
```

```
!0213 N3MRT
01                ;DETERMINE IF SYSTEM HAS A 6063/64 DISK
02                ;DISABLE TEST ENTER IF NONE EXISTS
03 14152 004000           1B4
04 14153 000040           1B10              ;READY BIT
05 14154 062677 PD.01:    IORST
06 14155 054472           STA 3,PD.S3
07 14156 102000           ADC 0,0
08 14157 062026           DOB 0,PDDSK
09 14160 126520           SUBZL 1,1
10 14161 067026           DOC 1,PDDSK       ;SET ALT MODE 1
11 14162 064426           DIA 1,PDDSK
12 14163 122414           SUB# 1,0,SZR      ;SKP= DSK CONTROL
13 14164 000442           JMP PD.NC         ;NO 6063/64 DISK CONTROLLER
14                ;NOW DETERMINE HOW MANY DRIVES ARE ON LINE
15 14165 102400           SUB 0,0
16 14166 040463           STA 0,PD.ND
17 14167 030463           LDA 2,PDTXA
18 14170 133000           ADD 1,2           ;DEC ADDR IN AC2 FOR AUTO-INC LOC
19 14171 050020           STA 2,20
20 14172 024456 PD.1L:    LDA 1,PD.WB       ;WRITE BUF CMD
21 14173 032436           LDA 2,@PD.LB      ;ADDR OF QUEUE
22 14174 041000           STA 0,0,2         ;SET DRIVE TO 0
23 14175 045001           STA 1,1,2         ;PLACE CMD IN QUEUE
24 14176 051002           STA 2,2,2         ;PLACE SAFE ADDR IN QUEUE
25 14177 071126           DOAS 2,PDDSK      ;GIVE QUEUE ADR TO DSK,START OPERATION
26 14200 060026           NIO PDDSK
27 14201 060026           NIO PDDSK
28 14202 060026           NIO PDDSK         ;
29 14203 066626           DICC 1,PDDSK      ;GET STATUS BITS
30 14204 030747           LDA 2,PD.01-1
31 14205 133405           AND 1,2,SNR       ;1B10=EXIST
32 14206 000407           JMP .+7
33 14207 010442           ISZ PD.ND         ;INC # OF DISCS
34 14210 030435           LDA 2,PD.K2       ;=177
35 14211 034741           LDA 3,PD.01-2     ;=4000
36 14212 167405           AND 3,1,SNR       ;SKP=2 MEGBYTE DRIVE
37 14213 151220           MOVZR 2,2         ;=64 TRACKS=1 MEG BYTE
38 14214 151001           MOV 2,2,SKP
39 14215 152000           ADC 2,2
40 14216 052020           STA 2,@20
41 14217 024423           LDA 1,PD10K       ;=10000
42 14220 123000           ADD 1,0           ;NEXT DRIVE TO TRY
43 14221 103133           ADDZL# 0,0,SNC    ;SKP=DONE ALL
44 14222 000750           JMP PD.1L         ;TRY NEXT DRIVE
45 14223 020426           LDA 0,PD.ND
46 14224 101004           MOV 0,0,SZR       ;FIND ANY DRIVES?
47 14225 000405           JMP PD.1E         ;YEP
48 14226 102000 PD.NC:    ADC 0,0           ;NO DISKS EXIST
49 14227 040634           STA 0,PD.00+2
50 14230 002417           JMP @PD.S3
51 14231 001443 PD.LB:    LSETB
```

```
!0214 N3MRT
01                ;PAGEING DISK ON LINE,SET UP INTRA STUFF
02 14232 062677 PD.1E:    IORST
03 14233 020650           LDA 0,PD.1R
04 14234 040626           STA 0,PD.00+1     ;SO SIZE GETS TYPED
05 14235 020407           LDA 0,PD.K1
06 14236 024407           LDA 1,PD.K2
07 14237 030407           LDA 2,PD.K3
08 14240 006101           JSR @EINTS        ;ENTER INTA SERVICE PARAMS
09 14241 002406           JMP @PD.S3
10 14242 010000 PD10K:    10000
11 14243 177654 PDMK1:    -84.
12 14244 000026 PD.K1:    PDDSK
13 14245 000177 PD.K2:    177
14 14246 014673 PD.K3:    PD.IS
15 14247 000000 PD.S3:    0
16 14250 002500 PD.WB:    2500
17 14251 000000 PD.ND:    0
18 14252 014616 PDTXA:    PDDTB
```

```
!0215 N3MRT
01
02                ;EXECUTE PORTION OF DISK TEST
03 14253 014063           PD.00+2
04 14254 032777 PD.02:    LDA 2,@.-1
05 14255 151004           MOV 2,2,SZR
06 14256 001000           JMP 0,2
07                        LCALL ASCRA       ;GET 1K SCR
08 14257 100010           ASCRA-ASCRA*1B11+100010
09 14260 002457           JMP @PD.XX        ;NONE AVAIL
10                        LCALL ADMAP
11 14261 100550           ADMAP-ASCRA*1B11+100010
12 14262 002455           JMP @PD.XX
13                        LCALL ARANG
14 14263 100270           ARANG-ASCRA*1B11+100010
15 14264 105000           MOV 0,1
16 14265 030542           LDA 2,PD.K7       ;=7
17 14266 102400           SUB 0,0
18 14267 040451           STA 0,PD.KK
19                        LCALL ADIVI
20 14270 100310           ADIVI-ASCRA*1B11+100010
21 14271 100405           NEG 0,0,SNR
22 14272 000410           JMP PD.2C
23                PD.2L:  LCALL ESCRA
24 14273 100030           ESCRA-ASCRA*1B11+100010
25 14274 000406           JMP PD.2C         ;CANT EXPAND MORE
26                        LCALL EDMAP       ;ASSIGN TO DCH
27 14275 100570           EDMAP-ASCRA*1B11+100010
28 14276 000436           JMP PD.2R         ;CANT ASSIGN MORE
29 14277 010441           ISZ PD.KK         ;#1K'S +1
30 14300 101404           INC 0,0,SZR
31 14301 000772           JMP PD.2L         ;TRY 1K MORE TO ASSIGN
```

```
!0216 N3MRT
01                ;SELECT A RANDOM BUFFER START WITHIN 253 OF THE START
02                ;OF THE SECOND 256 WORD BLOCK IN
03                ;THE ASSIGNED SCRATCH AREA
04                ;FIRST 256 WORDS OF SCRATCH RESERVED FOR COMMAND QUEUE
05                PD.2C:  LCALL ARANG
06 14302 100270           ARANG-ASCRA*1B11+100010
07 14303 030443           LDA 2,PD253
08 14304 105000           MOV 0,1
09                        LCALL ADIVI
10 14305 100310           ADIVI-ASCRA*1B11+100010
11 14306 101300           MOVS 0,0
12 14307 101700           INCS 0,0          ;ADD 400 TO RANDOM START
13 14310 024152           LDA 1,SCRLO
14 14311 107000           ADD 0,1
15 14312 046430           STA 1,@PD32+1     ;PDDST=DATA START ADDRS
16 14313 034154           LDA 3,DCHLO
17 14314 171300           MOVS 3,2
18 14315 151220           MOVZR 2,2         ;FORM DCH # 1K'S ADDR
19 14316 151220           MOVZR 2,2
20 14317 052432           STA 2,@PD.12+2    ;SAVE IN PD.CA
21 14320 117000           ADD 0,3
22 14321 056422           STA 3,@PD32+2     ;PDCST=CHANNEL START ADDRS
23 14322 024416           LDA 1,PD.KK       ;#1KS
24 14323 125120           MOVZL 1,1         ;*2
25 14324 125520           INCZL 1,1         ;*4+2 FOR 1ST 1K
26 14325 046417           STA 1,@PD32+3     ;PD.BK=# BLKS/SECTORS AVAIL
27 14326 125300           MOVS 1,1
28 14327 124600           NEGR 1,1
29 14330 125400           INC 1,1
30 14331 125500           INCL 1,1
31 14332 046422           STA 1,@PD.12+5    ;=PD.WK
32 14333 000520           JMP PD.03         ;FIGURE OUT WHAT TO DO
33                PD.2R:  LCALL RSCRA       ;RELEASE 1K SCR
34 14334 100050           RSCRA-ASCRA*1B11+100010
35 14335 063077           HALT              ;NOT PROBABLE HALT
36 14336 000744           JMP PD.2C
37 14337 015005 PD.XX:    PD.XI
38 14340 000000 PD.KK:    0
39 14341 000040 PD32:     32.
40 14342 014574           PDDST
41 14343 014575           PDCST
42 14344 014566           PD.BK
43 14345 014600           PDSEC
44 14346 000375 PD253:    253.
45 14347 000012 PD.12:    12
46 14350 014614           PDEST
47 14351 014571           PD.CA
48 14352 014577           PDCYL
49 14353 014611           PDIDX
50 14354 014605           PD.WK
```

```
!0217 N3MRT
01
02                  ;NOW SELECT SECTOR TO EXERCISE
03 14355 014576             PDDRV
04                  PD.3S:  LCALL ARANG
05 14356 100270             ARANG-ASCRA*1B11+100010
06 14357 105000             MOV 0,1
07 14360 030761             LDA 2,PD32          ;32 SECTORS
08 14361 022763             LDA 0,@PD32+3       ;=PD.BK
09 14362 112400             SUB 0,2             ;32-# TO WRITE
10                                              ;=MAX POSSIBLE START SECTOR
11                          LCALL ADIVI         ;(0) WILL=SECTOR START
12 14363 100310             ADIVI-ASCRA*1B11+100010
13 14364 026760             LDA 1,@PD32+3       ;=PD.BK
14 14365 125300             MOVS 1,1            ;# SECT IN UPPER BYTE
15 14366 123000             ADD 1,0             ;CREATE SEC CONTROL WD
16 14367 042756             STA 0,@PD32+4       ;START SECTOR AND #
17 14370 122400             SUB 1,0             ;LEAVE STRT SECT. #
18                  ;CREATE DOA FOR THIS SET SECTORS
19 14371 032761             LDA 2,@PD.12+3
20 14372 153120             ADDZL 2,2           ;*4
21 14373 153120             ADDZL 2,2           ;*16
22 14374 151120             MOVZL 2,2           ;*32 FOR SEECTOR
23 14375 113000             ADD 0,2
24 14376 052757             STA 2,@PD.3S-1      ;FOR DOA'S
```

```
!0218 N3MRT
01                  ;NOW GET RANDOM DATA WORDS
02                          LCALL FRANG
03 14377 100450             FRANG-ASCRA*1B11+100010
04 14400 034430             LDA 3,PD.3G-1       ;ADDR OF PDDW1-3
05 14401 045400             STA 1,0,3           ;NEW RAN DATA WDS
06 14402 051401             STA 2,1,3
07 14403 041402             STA 0,2,3
08 14404 036747             LDA 3,@PD.12+4      ;PDIDX=ADDR OF OP TBL
09 14405 045403             STA 1,3,3           ;SAVE CONTROL
10 14406 051404             STA 2,4,3           ;WORDS NEWLY
11 14407 041405             STA 0,5,3
12 14410 171000             MOV 3,2
13                          LCALL ARANG
14 14411 100270             ARANG-ASCRA*1B11+100010
15 14412 034416             LDA 3,PD.3G-1       ;ADDR OF PDDW1
16 14413 041403             STA 0,3,3           ;NEW PDDW3
17 14414 041006             STA 0,6,2
18 14415 020562             LDA 0,PDCYL         ;GENERATED FOR
19 14416 041000             STA 0,0,2           ;THIS OP TABLE
20 14417 020561             LDA 0,PDSEC
21 14420 041001             STA 0,1,2
22 14421 020555             LDA 0,PDDRV
23 14422 041002             STA 0,2,2
24 14423 004406             JSR PD.3G           ;SET UP SCRATCH FOR WRITE
25 14424 002401             JMP @PDX05          ;START WRITE
26 14425 014630 PDX05:      PD.05
27 14426 000004 PD.K4:      4
28 14427 000007 PD.K7:      7
29
30                  ;GENERATE DATA PATTERN INTO SCRATCH AREA
31
32 14430 014601             PDDW1
33 14431 054421 PD.3G:      STA 3,PD3GR         ;SAVE RETURN ADDR
34 14432 030542             LDA 2,PDDST
35 14433 020546             LDA 0,PDDW1         ;GET 2 DATA WORDS
36 14434 024546             LDA 1,PDDW2         ;TO FIRST 2 WORDS
37 14435 041000             STA 0,0,2           ;IN THE BUFFER
38 14436 045001             STA 1,1,2
39 14437 020544             LDA 0,PDDW3
40 14440 024544             LDA 1,PDDW4
41 14441 041002             STA 0,2,2
42 14442 045003             STA 1,3,2           ;1ST 4 WORDS RAPDOM
43 14443 024542             LDA 1,PD.WK         ;GET # WORDS-4
44 14444 021000             LDA 0,0,2
45 14445 041004             STA 0,4,2
46 14446 151400             INC 2,2
47 14447 125404             INC 1,1,SZR
48 14450 000774             JMP .-4
49 14451 002401             JMP @PD3GR          ;RETRN
50 14452 000000 PD3GR:      0
```

```
!0219 N3MRT
01                ;DATA CHANNEL AND SCRATCH ARE ASSIGNED
02                ;SELECT AN AVAILABLE DRIVE ,THEN
03                ;SELECT ONE OF THE THREE OP TABLES
04                ;AND PERFORM THE NEXT OPERATION REQ
05                PD.03:  LCALL ARANG
06 14453 100270           ARANG=ASCRA*1B11+100010
07 14454 105100           MOVL 0,1
08 14455 125220           MOVZR 1,1
09 14456 030750           LDA 2,PD.K4       ;=4
10                        LCALL ADIVI       ;REM=DRIVE TO TRY
11 14457 100310           ADIVI=ASCRA*1B11+100010
12 14460 040510           STA 0,PDDNM       ;SAVE DRV #
13 14461 105000           MOV 0,1
14 14462 030533           LDA 2,PD.TB       ;DRV TBL POINTER
15 14463 133000           ADD 1,2
16 14464 021000           LDA 0,0,2         ;THIS DRV EXIST?
17 14465 100015           COM# 0,0,SNR      ;SKP=EXIST
18 14466 000765           JMP PD.03         ;NOPE TRY ANOTHER ONE
19 14467 040500           STA 0,PDSUR       ;SAVE # TRACKS THIS DRV
20                        LCALL ARANG
21 14470 100270           ARANG=ASCRA*1B11+100010
22 14471 101300           MOVS 0,0
23 14472 126520           SUBZL 1,1
24 14473 125140           MOVOL 1,1         ;AC1=3
25 14474 101220           MOVZR 0,0
26 14475 107627           ANDZR 0,1,SBN     ;3 RESULT ILLEGAL
27 14476 125101           MOVL 1,1,SKP
28 14477 000772           JMP .-6           ;TRY NEXT 2 BITS
29 14500 102400           SUB 0,0
30 14501 125004           MOV 1,1,SZR
31 14502 020725           LDA 0,PD.K7       ;=7
32 14503 125224           MOVZR 1,1,SZR
33 14504 103000           ADD 0,0
34 14505 030515           LDA 2,PDTTL
35 14506 024462           LDA 1,PDDNM       ;DRV #
36 14507 133000           ADD 1,2
37 14510 031000           LDA 2,0,2         ;GET ADDR OF THIS DRV'S OP TBLS
38 14511 050462           STA 2,PDTBL       ;SAVE IT
39 14512 113000           ADD 0,2           ;ADDR IN AC2 OF THIS OP TBL
40 14513 025000           LDA 1,0,2
41 14514 044463           STA 1,PDCYL       ;GET CURRENT CYLINDER
42 14515 021001           LDA 0,1,2
43 14516 040462           STA 0,PDSEC       ;SECTOR AND # SECTORS
44 14517 021002           LDA 0,2,2         ;DRV+TRK+SECT
45 14520 040456           STA 0,PDDRV
46 14521 021003           LDA 0,3,2
47 14522 040457           STA 0,PDDW1       ;DATA WORD 1
48 14523 021004           LDA 0,4,2
49 14524 040456           STA 0,PDDW2       ;DATA WORD 2
50 14525 021005           LDA 0,5,2         ;DATA WORD 3
51 14526 040455           STA 0,PDDW3
52 14527 021006           LDA 0,6,2         ;DATA WORD 4
53 14530 040454           STA 0,PDDW4
54 14531 050460           STA 2,PDIDX       ;SAVE ADDRS CONTOL WORD
55 14532 125005           MOV 1,1,SNR       ;SKIP = CYL SEL
56 14533 000407           JMP PD.3L         ;SELECT CYL AND WRITE
57                        LCALL ARANG
58 14534 100270           ARANG=ASCRA*1B11+100010
59 14535 101233           MOVZR# 0,0,SNC    ;SKP=DO DATA VERIFY
60 14536 000471           JMP PD.04         ;DO READ
```

```
 0220 N3MRT
01 14537 004672           JSR PD.3G         ;SET UP FOR VERIFY
02 14540 126520           SUBZL 1,1         ;DATA VERIFY FLAG
03 14541 000470           JMP PD.06         ;DO DATA VERIFY
```

```
!0221 N3MRT
01                ;SELECT A RANDOM CYL NOT IN USE AND WRITE IT
02                PD.3L:  LCALL ARANG       ;RANDOM SEL CYL
03 14542 100270           ARANG-ASCRA*1B11+100010
04 14543 105000           MOV 0,1
05 14544 030423           LDA 2,PDSUR       ;CYLINDER SELECTED
06                        LCALL ADIVI       ;MUST NOT BE
07 14545 100310           ADIVI-ASCRA*1B11+100010
08 14546 101231           MOVZR# 0,0,SKP    ;SKP=USE ALL # SZC=EVEN SNC=ODD
09 14547 000773           JMP PD.3L         ;DUAL. PROC. DELETE CYL.
10 14550 103240           ADDOR 0,0         ;1B0 SO #0 IS VALID
11 14551 034422           LDA 3,PDTBL       ;ALREADY IN USE
12 14552 031400           LDA 2,0,3         ;AND CYL 0
13 14553 142415           SUB # 2,0,SNR     ;IS NOT EXERCISED
14 14554 000766           JMP PD.3L
15 14555 031407           LDA 2,7,3
16 14556 142415           SUB# 2,0,SNR
17 14557 000763           JMP PD.3L
18 14560 031416           LDA 2,16,3
19 14561 142415           SUB# 2,0,SNR
20 14562 000760           JMP PD.3L
21 14563 040414           STA 0,PDCYL       ;CYL SEL OK TO USE
22 14564 002401           JMP @.+1
23 14565 014356           PD.3S
24 14566 000000 PD.BK:    0
25 14567 000000 PDSUR:    0
26 14570 000000 PDDNM:    0
27 14571 000000 PD.CA:    0
28 14572 000037 PD37:     37
29 14573 000000 PDTBL:    0
30 14574 000000 PDDST:    0
31 14575 000000 PDCST:    0
32 14576 000000 PDDRV:    0
33 14577 000000 PDCYL:    0
34 14600 000000 PDSEC:    0
35 14601 000000 PDDW1:    0
36 14602 000000 PDDW2:    0
37 14603 000000 PDDW3:    0
38 14604 000000 PDDW4:    0
39 14605 000000 PD.WK:    0
40 14606 000000 PD.CO:    0                 ;0=READ,1=WRITE
41 14607 000000 PD.OC:    0                 ;SECT
42 14610 000000 PDSTA:    0                 ;STATUS READ
43 14611 000000 PDIDX:    0
44 14612 000400 PD.WR:    1B7
45 14613 000040 PDHOE:    1B10
46 14614 000000 PDEST:    0
47 14615 014616 PD.TB:    .+1
48 14616 000004 PDDTB:    .BLK 4
49 14622 014623 PDTTL:    .+1
50 14623 015172           PDCTB
51 14624 015217           PDCTB+21.
52 14625 015244           PDCTB+42.
53 14626 015271           PDCTB+63.
```

```
!0222 N3MRT
01                ;PD.06 IS VERIFY DATA ENTER
02                ;PD.05 IS WRITE ENTER
03                ;CALCULATE DISK CONTROL WORDS
04                ;AND PLACE IN COMMAND QUEUE
05                ;ND.04 IS READ ENTER
06 14627 126001 PD.04:    ADC 1,1,SKP       ;-1 TO STRTT READ
07 14630 126400 PD.05:    SUB 1,1           ;0 TO STRT WRITE
08 14631 044755 PD.06:    STA 1,PD.CO       ;1 TO VERIFY
09 14632 024746           LDA 1,PDSEC
10 14633 020737           LDA 0,PD37
11 14634 123400           AND 1,0
12 14635 106700           SUBS 0,1          ;AC1=START SECTOR
13 14636 030730           LDA 2,PD.BK       ;(2)=# 256 WD BLK AVAIL
14 14637 146432           SUBZ# 2,1,SZC
15 14640 145000           MOV 2,1           ;AC1=# SECTORS
16 14641 044746           STA 1,PD.OC       ;SAVE # SECTORS
17 14642 131300           MOVS 1,2          ;TO EXERCISE
18 14643 150600           NEGR 2,2
19 14644 151400           INC 2,2
20 14645 151500           INCL 2,2
21 14646 050737           STA 2,PD.WK       ;BUFFER L - 4 WORDS
22 14647 102120           ADCZL 0,0         ;INTA EXPECTED FLG
23 14650 042420           STA 0,@PD.0X
24 14651 020742           LDA 0,PDHOE       ;HALT ON ERR BIT
25 14652 034740           LDA 3,PD.WR       ;BIT 7
26 14653 010733           ISZ PD.CO         ;SKP =READ
27 14654 000402           JMP .+2           ;MUST BE WRITE OR VERIFY
28 14655 000405           JMP .+5
29 14656 030730           LDA 2,PD.CO       ;2=VERIFY
30 14657 151233           MOVZR# 2,2,SNC    ;SKP=WRITE=1
31 14660 175120           MOVZL 3,3         ;DATA VERIFY BIT
32 14661 163000           ADD 3,0           ;ADD IN WRITE/VERIFY BIT
33 14662 040534           STA 0,PDCMD       ;SAVE IN CMD WORD
34 14663 024724           LDA 1,PD.OC       ;# SECTORS
35 14664 006405           JSR @PDICQ        ;BUILD COMMAND QUEUE
36 14665 034154           LDA 3,DCHLO
37 14666 075126           DOAS 3,PDDSK
38                        LCALL RETRN       ;START DISK ,EXIT
39 14667 100210           RETRN-ASCRA*1B11+100010
40 14670 014063 PD.0X:    PD.00+2
41 14671 015056 PDICQ:    PDCMQ    ;ADDR OF CMD QUEUE BUILDER
```

```
!0223 N3MRT
01 14672 000026                 PDDSK
02                      ;DISK INTERRUPT SERVICE DISPATCHER
03 14673 062426 PD.IS:  DIC 0,PDDSK
04 14674 032774         LDA 2,@PD.0X
05 14675 151132         MOVZL# 2,2,SZC  ;EXPECTED INTR?
06 14676 000405         JMP .+5
07 14677 020773         LDA 0,PD.IS-1
08 14700 040144         STA 0,UDEVI     ;SET UNEXPECTED INTR FLG
09 14701 060226         NIOC PDDSK
10 14702 001400         JMP 0,3
11 14703 040705         STA 0,PDSTA
12 14704 101132         MOVZL# 0,0,SZC  ;CK FOR STATUS ERR
13 14705 000405         JMP PD.SE
14 14706 020410         LDA 0,PDCOM
15 14707 060226 PD.EX:  NIOC PDDSK
16 14710 042760         STA 0,@PD.0X    ;VERIFY DATA RETURN
17 14711 001400         JMP 0,3         ;DISMISS INTA
18 14712 020530 PD.SE:  LDA 0,PDSTE
19 14713 000774         JMP PD.EX       ;STATUS ERR EXIT
```

```
!0224 N3MRT
01
02                      ;READ OR WRITE OK NO STAT ERRS
03                      ;COMPARE DATA IN BUFFER
04                      ;AGAINST WHAT'S SUPPOSED TO BE THERE
05 14714 014605         PD.WK
06 14715 014601         PDDW1
07 14716 014717 PDCOM:  .+1
08 14717 034776         LDA 3,PDCOM-1   ;PDDW1 ADDR
09 14720 030654         LDA 2,PDDST     ;START ADRS
10 14721 021400         LDA 0,0,3       ;PDDW1
11 14722 025000         LDA 1,0,2       ;COMPARE
12 14723 122414         SUB# 1,0,SZR    ;FIRST 4 WORDS
13 14724 000474         JMP PDERR       ;AGAINST EVERY
14 14725 021401         LDA 0,1,3       ;PDDW2
15 14726 025001         LDA 1,1,2
16 14727 122414         SUB# 1,0,SZR
17 14730 000470         JMP PDERR
18 14731 021402         LDA 0,2,3       ;PDDW3
19 14732 025002         LDA 1,2,2
20 14733 106414         SUB# 0,1,SZR
21 14734 000464         JMP PDERR
22 14735 021403         LDA 0,3,3       ;PDDW4
23 14736 025003         LDA 1,3,2
24 14737 106414         SUB# 0,1,SZR
25 14740 000460         JMP PDERR
26 14741 036753         LDA 3,@PDCOM-2  ;PD.WK ,#WORDS - 4
27 14742 021000         LDA 0,0,2
28 14743 025004         LDA 1,4,2
29 14744 122414         SUB# 1,0,SZR
30 14745 000453         JMP PDERR
31 14746 055000         STA 3,0,2       ;CLR BUFFER NXT RD
32 14747 151400         INC 2,2         ;BUMP ADRS
33 14750 175404         INC 3,3,SZR     ;SKP DONEALL
34 14751 000771         JMP .-7
35                      LCALL ARANG     ;BITS 1 AND 15=1
36 14752 100270         ARANG-ASCRA*1B11+100010
37 14753 101232         MOVZR# 0,0,SZC  ;RELEASE THIS
38 14754 103123         ADDZL 0,0,SNC   ;OP TABLE
39 14755 000403         JMP .+3         ;EITHER = 0 KEEP IT
40 14756 102400 PD.X0:  SUB 0,0
41 14757 042436         STA 0,@PDIIX
42                      LCALL ARANG
43 14760 100270         ARANG-ASCRA*1B11+100010
44 14761 101102         MOVL 0,0,SZC    ;NEXT 2 BITS =1?
45 14762 101103         MOVL 0,0,SNC    ;IS RELEASE THIS BUFFER
46 14763 002434         JMP @PD.SX      ;SELECT NEW STRT BUFFER
47 14764 032420 PD.X3:  LDA 2,@PDHIK    ;=HIGHK=# 1K'S
48 14765 024136         LDA 1,MPSWT
49 14766 125005         MOV 1,1,SNR     ;SKP=MAPPED
50 14767 000416         JMP PD.XI       ;RELEASE IF UNMAPPED
51 14770 024413         LDA 1,PD.37
52 14771 146032         ADCZ# 2,1,SZC   ;SKP=>32K
53 14772 000413         JMP PD.XI
54                      PD.X4:  LCALL RDMAP     ;REL DCH
55 14773 100610         RDMAP-ASCRA*1B11+100010
56 14774 101001         MOV 0,0,SKP
57 14775 000776         JMP PD.X4
58                      PD.X5:  LCALL RSCRA
59 14776 100050         RSCRA-ASCRA*1B11+100010
60 14777 101001         MOV 0,0,SKP
```

```
 0225 N3MRT
01 15000 000776          JMP PD.X5
02 15001 002401          JMP @.+1
03 15002 014257          PD.02+3
04 15003 000037 PD.37:   37
05 15004 001502 PDHIK:   HIGHK
```

```
10226 N3MRT
01               PD.XI:  LCALL RDMAP     ;REL DCH MAP
02 15005 100610          RDMAP-ASCRA*1B11+100010
03 15006 101001          MOV 0,0,SKP
04 15007 000776          JMP PD.XI
05               PD.X2:  LCALL RSCRA
06 15010 100050          RSCRA-ASCRA*1B11+100010
07 15011 102401          SUB 0,0,SKP
08 15012 000776          JMP PD.X2
09 15013 042655          STA 0,@PD.0X
10                       LCALL RETRN
11 15014 100210          RETRN-ASCRA*1B11+100010
12 15015 114611 PDIIX:  @PDIDX
13 15016 000000 PDCMD:   0
14 15017 014302 PD.SX:  PD.2C
15                       ;ERROR IN DATA COMPARE
16 15020 054430 PDERR:   STA 3,PD.SA
17                       LCALL ERROI
18 15021 100350          ERROI-ASCRA*1B11+100010
19 15022 000401          JMP .+1
20 15023 020474          LDA 0,PDTX1
21                       LCALL ERRTX
22 15024 100170          ERRTX-ASCRA*1B11+100010
23 15025 020423          LDA 0,PD.SA
24 15026 026425          LDA 1,@PDDSX
25 15027 032426          LDA 2,@PDXCA
26                       LCALL ERROC
27 15030 100370          ERROC-ASCRA*1B11+100010
28 15031 000401          JMP .+1
29 15032 020500          LDA 0,PDTX2
30                       LCALL ERRTX
31 15033 100170          ERRTX-ASCRA*1B11+100010
32 15034 022420          LDA 0,@PDSTX
33 15035 026414          LDA 1,@PD.CX
34 15036 032414          LDA 2,@PD.0X
35                       LCALL ERROC
36 15037 100370          ERROC-ASCRA*1B11+100010
37 15040 000716          JMP PD.X0       ;RELEASE IMMED SW15=1
38 15041 000723          JMP PD.X3
```

```
!0227 N3MRT
01                ;NON REC DISK STATUS ERROR
02 15042 015043 PDSTE:  .+1
03 15043 032411         LDA 2,@PDSTX
04 15044 145000         MOV 2,1
05 15045 141000         MOV 2,0
06 15046 036406         LDA 3,@PDSTX
07 15047 000751         JMP PDERR
08 15050 000000 PD.SA: 0
09 15051 014576 PD.CX: PDDRV
10 15052 014606 PD.OX: PD.CO
11 15053 014574 PDDSX: PDDST
12 15054 014610 PDSTX: PDSTA
13 15055 014571 PDXCA: PD.CA
14
15                ;THIS SUBROUTINE BUILDS THE COMMAND QUEUE USED
16                ; IN WRITE/VERIFY AND READ OPERATIONS.
17 15056 054432 PDCMQ:  STA 3,PDCQR
18 15057 044432         STA 1,PDNNS     ;SAVE # SECTORS
19 15060 034152         LDA 3,SCRLO     ;SCRLO=STARTING ADDRESS OF QUE
20 15061 054432         STA 3,PDSVS
21 15062 022767         LDA 0,@PD.CX    ;=PDDRV=DRIVE+TRK+SECT
22 15063 024733         LDA 1,PDCMD     ;READ/VERIFY/WRITE+EXT ADDR BITS
23 15064 032430         LDA 2,@PD.XT    ;=PDCST=DATA ADDR START
24 15065 034424 PDCQL:  LDA 3,PDNNS     ;LAST BLK?
25 15066 175224         MOVZR 3,3,SZR   ;SKP=LST CMD BLK
26 15067 000403         JMP .+3
27 15070 034422         LDA 3,PD.HL     ;HALT BIT
28 15071 167000         ADD 3,1         ;ADD IN HALT BIT
29 15072 034421         LDA 3,PDSVS
30 15073 041400         STA 0,0,3       ;DRV+TRK+SEC
31 15074 045401         STA 1,1,3       ;CMD+MAP MODE+EXT ADDR
32 15075 051402         STA 2,2,3       ;DATA ADDR
33 15076 030420         LDA 2,PD.K5     ;=5
34 15077 157000         ADD 2,3         ;INCR QUEUE ADDR BY 5
35 15100 054413         STA 3,PDSVS     ;SAVE IT
36 15101 031775         LDA 2,-3,3      ;GET AC2
37 15102 101400         INC 0,0         ;NEXT SECTOR
38 15103 034412         LDA 3,PD400     ;=400
39 15104 173020         ADDZ 3,2        ;ADD 400 TO DATA ADDR,
40 15105 014404         DSZ PDNNS       ;SKP=DONE ALL
41 15106 000757         JMP PDCQL       ;DO NEXT CMD BLOCK
42 15107 002401         JMP @PDCQR      ;RETURN
43 15110 000000 PDCQR:  0
44 15111 000000 PDNNS:  0
45 15112 000100 PD.HL:  100
46 15113 000000 PDSVS:  0
47 15114 014575 PD.XT: PDCST
48 15115 000400 PD400:  400
49 15116 000005 PD.K5: 5
50
51 15117 015120 PDTX1: .+1
52 15120 005215         .TXTE (<15><12>PD.SA<11>PDDST<11>PD.CA(
53 15132 015133 PDTX2: .+1
54 15133 005215         .TXTE (<15><12>PDSTA<11>PDADR<11>PD.CO(
55 15145 005215 PDTX3:  .TXTE   1<15><12>DEV# 26 6063/64 DISK<15><12>
56 15161 051504 DSK #<11>#TRACK<11>#SEC!
57
58                ;4-3/7 WORD OPERATION TABLES
59 15172 000124 PDCTB:  .BLK 84.
```

```
!0228 N3MRT
01                ;DCUTS - DEFINE PARAMETERS TO LINKR
02                ;DCU 50 TEST TO BE RUN BY DCU 50
03                ;(ARITHMETIC TEST THROUGH DATA CHANNEL)
04                        NEXTT DC.00
05        015316 LMEML=.
06        000164          .LOC LPG0
07 00164 015321           DC.00
08        000165 LPG0=.
09        015316          .LOC LMEML
10 15316 000000           0       ;TEST PASS CTR
11 15317 000000           0       ;TEST ERROR CTR
12 15320 000000           0       ;INTERRUPT TIMEOUT SWITCH
13 15321 015337 DC.00:    DC.01
14 15322 015364           DC.02
15 15323 000000           0
16 15324 000000           0
17 15325 177777           -1
18 15326 176000           176000
19 15327 015460           DC.05
20 15330 015460           DC.05
21        015331          .TXTE (
22 15331 141504 DCU 50 TEST(
23        120125
24        030065
25        152240
26        051705
27        000324
28                ;SET NO SCRATCH ASSIGNED SWITCH
29 15337 020421 DC.01:    LDA 0,DCUDV     ;GET DEVICE #
30 15340 101004           MOV 0,0,SZR     ;SKP NO DEVICE # INSERTED
31 15341 000404           JMP .+4         ;ENTER INTERRUPT SERVICE
32 15342 102000           ADC 0,0
33 15343 040760           STA 0,DC.00+2
34 15344 001400           JMP 0,3
35 15345 030416           LDA 2,DCUDV+3
36 15346 025000           LDA 1,0,2       ;UPDATE THE I/O OUT'S ETC.
37 15347 125005           MOV 1,1,SNR     ;TO REFLECT THE CURRENT DCU
38 15350 000405           JMP .+5         ;DEVICE CODE
39 15351 107000           ADD 0,1
40 15352 045005           STA 1,DCDOA-DCDOT,2     ;STA'S UPDATED IO
41 15353 151400           INC 2,2         ;STEP TO  NEXT INSTR.
42 15354 000772           JMP .-6         ;0'S = END OF TABLE
43 15355 024404           LDA 1,DCUDV+1
44 15356 030404           LDA 2,DCUDV+2
45 15357 002101           JMP @EINTS      ;ENTER DCU IN INTR. VECTORS
46
47 15360 000000 DCUDV:    0       ;ENTER DCU DEVICE # HERE TO RUN
48 15361 007777           7777            ;INTERRUPT MASK
49 15362 015537           DCUIS           ;ADRS OF INTERRUPT SERVICE
50 15363 015466           DCDOT           ;ADDRESS OF I/O CMD'S
```

```
!0229 N3MRT
01                ;EXECUTE ENTRY POINT GET SCRATCH IF NONE AVAIL
02
03 15364 030737 DC.02:  LDA 2,DC.00+2
04 15365 151004         MOV 2,2,SZR       ;SKP IF NO INTR. TO PROCESS
05 15366 001000         JMP 0,2
06 15367 020152         LDA 0,SCRLO
07 15370 101004         MOV 0,0,SZR       ;SKP IF NO SCRATCH ASSIGNED
08 15371 000453         JMP DC.RL
09                      LCALL ASCRA       ;GET 1K SCRATCH
10 15372 100010         ASCRA=ASCRA*1B11+100010
11 15373 000465         JMP DC.05         ;EXIT NONE AVAILABLE
12                      LCALL ADMAP       ;GET DCH MAP ASSIGNED
13 15374 100550         ADMAP=ASCRA*1B11+100010
14 15375 000463         JMP DC.05         ;NO DCH AVAILABLE
15 15376 020154         LDA 0,DCHLO       ;DCU CANNOT USE 0 TO 1777
16 15377 101005         MOV 0,0,SNR       ;SKP OK DCH
17 15400 000460         JMP DC.05         ;TRY AGAIN NEXT TIME
18 15401 102000         ADC 0,0
19 15402 040532         STA 0,DC.ES       ;SET NO ERR SWTCH
20                      LCALL ESCRA       ;TRY TO GET 1K MORE
21 15403 100030         ESCRA=ASCRA*1B11+100010
22 15404 000406         JMP DC.2L-2               ;BUT USE 1K IF NO MORE AVAIL
23                      LCALL EDMAP       ;EXPAND THE DCH MAP 1K
24 15405 100570         EDMAP=ASCRA*1B11+100010
25 15406 000402         JMP .+2 ;CAN'T EXPAND DATA CHANNEL RELEASE 1K
26
27 15407 000403         JMP DC.2L-2
28                      LCALL RSCRA       ;GET RID OF 1K SCRATCH
29 15410 100050         RSCRA=ASCRA*1B11+100010
30 15411 063077         HALT              ;CAN'T HAPPEN NO SCR. TO RELEASE!
31 15412 030524         LDA 2,DC.37
32 15413 150400         NEG 2,2           ;TRY RANDOM BETWEEN LIM 37 TRYS
33              DC.2L:  LCALL ARANG
34 15414 100270         ARANG=ASCRA*1B11+100010
35 15415 024153         LDA 1,SCRHI
36 15416 103000         ADD 0,0
37 15417 101220         MOVZR 0,0         ;CLR BIT 0
38 15420 122032         ADCZ# 1,0,SZC
39 15421 000411         JMP DC.2A         ;GRTR THAN HILIM
40 15422 034513         LDA 3,DC.PL
41 15423 166400         SUB 3,1           ;ENOUGH ROOM
42 15424 122032         ADCZ# 1,0,SZC     ;TO RELOCATE UP
43 15425 162400         SUB 3,0           ;NO LOWER NUMBER
44 15426 024152         LDA 1,SCRLO
45 15427 125400         INC 1,1
46 15430 122432         SUBZ# 1,0,SZC
47 15431 000404         JMP DC.2B
48 15432 151404 DC.2A:  INC 2,2,SZR
49 15433 000761         JMP DC.2L
50 15434 121000         MOV 1,0
```

```
!0230 N3MRT
01
02
03                ;MOVE TESTS TO SELECTED AREA
04
05 15435 111000 DC.2B:  MOV 0,2           ;ADJUST SCRHI
06 15436 020477         LDA 0,DC.PL       ;SO THAT TESTS
07 15437 024153         LDA 1,SCRHI       ;WILL FIT
08 15440 106400         SUB 0,1           ;INTO SCRATCH
09 15441 132032         ADCZ# 1,2,SZC
10 15442 131000         MOV 1,2
11 15443 050516         STA 2,DC.LC       ;START ADRS IN SCRATCH
12 15444 024471 DC.RL:  LDA 1,DC.PL
13 15445 124000         COM 1,1           ;-# WORDS TO MOVE
14 15446 030513         LDA 2,DC.LC       ;TO
15 15447 034514         LDA 3,DC.BG       ;FROM
16 15450 021400 DC.L2:  LDA 0,0,3         ;MOVE LOOP
17 15451 041000         STA 0,0,2
18 15452 151400         INC 2,2
19 15453 175400         INC 3,3
20 15454 125404         INC 1,1,SZR
21 15455 000773         JMP DC.L2
22 15456 050506         STA 2,DC.EN
23 15457 000420         JMP DC.03
24              DC.05:  LCALL RDMAP
25 15460 100610         RDMAP=ASCRA*1B11+100010
26 15461 000401         JMP .+1
27                      LCALL RSCRA
28 15462 100050         RSCRA=ASCRA*1B11+100010
29 15463 000402         JMP .+2
30 15464 000774         JMP DC.05
31                      LCALL RETRN
32 15465 100210         RETRN=ASCRA*1B11+100010
33
34                ;THE FOLLOWING LIST OF I/O INSTR.'S ARE
35                ;UPDATED TO THE CURRENT DCU DEVICE CODE
36                ;AND THEN UTILIZED BY THE TEST
37 15466 061000 DCDOT:  DOA 0,0
38 15467 066000         DOB 1,0
39 15470 060200         NIOC 0
40 15471 061400         DIB 0,0
41 15472 000000         0                 ;END OF TABLE
42 15473 000000 DCDOA:  0
43 15474 000000 DCDOB:  0
44 15475 000000 DCNIC:  0
45 15476 000000 DCDIB:  0
```

```
!0231 N3MRT
01
02
03                ;ARITHMETIC TEST HAS BEEN MOVED TO SCRATCH
04                ;SET UP AND INITIATE THE DCU TO START
05                ;EXECUTING CODE VIA THE DATA CHANNEL
06 15477 020774 DC.03:  LDA 0,DCDOA
07 15500 040422         STA 0,DC.DA      ;DOA 0,DCU FOR EXECUTE
08 15501 040455         STA 0,DC.IA      ;AND IN INTR. HNDLER
09 15502 020772         LDA 0,DCDOB
10 15503 040416         STA 0,DC.DB      ;DOB 1,DCU OUTPUTS ADRS.
11                      LCALL ARANG      ;PRIME ARITH RANGEN
12 15504 100270         ARANG=ASCRA*1B11+100010
13 15505 030454         LDA 2,DC.LC
14 15506 041053         STA 0,DCRNN-DCMS1,2      ;NEW RNDOM TO SCRATCH
15 15507 024152         LDA 1,SCRLO
16 15510 132400         SUB 1,2
17 15511 024154         LDA 1,DCHLO      ;DATA CHAN. SAME PHYSICAL
18 15512 147000         ADD 2,1          ;BUT DIFF LOGICAL
19 15513 044453         STA 1,DC.LA      ;LOGIICAL START OF TEST
20 15514 102000         ADC 0,0
21 15515 042416         STA 0,@DC.IF     ;WAITING INTERRUPT
22 15516 101120         MOVZL 0,0
23 15517 040441         STA 0,DC.PK      ;CONTINUE TEST A SECOND TIME
24 15520 100400         NEG 0,0          ;++1 TELLS DCU TO START
25 15521 063077 DC.DB:  HALT             ;DOB 1,DCU FOR ADRS.
26 15522 063077 DC.DA:  HALT             ;DOA 0,DCU STARTS IT
27                      LCALL RETRN      ;GO BACK TO LINKR
28 15523 100210         RETRN=ASCRA*1B11+100010
29 15524 015525 DC.04:  .+1      ;PASS COMPLETE NO ERRORS
30 15525 102400         SUB 0,0
31 15526 042405         STA 0,@DC.IF     ;CLEAR INTR. SW.
32 15527 020405         LDA 0,DC.ES      ;PREVIOUS ERR SW.
33 15530 101004         MOV 0,0,SZR      ;SKP IF PREV. ERR.
34 15531 000727         JMP DC.05        ;RELEASE SCRATCH
35 15532 000712         JMP DC.RL        ;MOV TEST TRY AGN.
36
37 15533 015323 DC.IF:  DC.00+2
38 15534 000000 DC.ES:  0
39 15535 001311 DC.PL:  DCEND=DCMS1
40 15536 000037 DC.37:  37
```

```
!0232 N3MRT
01
02                ;DCU INTERRUPT HANDLER
03                ;MAKE SURE STOPPED PC IS NOT AT ADRS 0
04                ;ISSUE A CONTINUE IF THIS IS NOT 2ND HALT
05 15537 020737 DCUIS:  LDA 0,DCDIB
06 15540 040403         STA 0,DC.DI
07 15541 020734         LDA 0,DCNIC
08 15542 040402         STA 0,.+2
09 15543 063077 DC.DI:  HALT             ;DIB 0,DCU GET'S P.C. FROM DCU
10 15544 063077         HALT             ;NIOC DCU       ;CLRS INTR.
11 15545 024423         LDA 1,DCERT
12 15546 101224         MOVZR 0,0,SZR    ;SKP=ERR HALT
13 15547 000403         JMP .+3          ;NO ERR. CHK. FOR CONTINUE
14 15550 046763         STA 1,@DC.IF     ;ENTER TO PROCESS INTA
15 15551 001400         JMP 0,3
16 15552 024752         LDA 1,DC.04      ;NO ERR RETURN
17 15553 010405         ISZ DC.PK        ;SKP=2ND PASS
18 15554 102521         SUBZL 0,0,SKP
19 15555 000773         JMP .-5          ;EXIT DCU DONE 2 PASSES
20 15556 063077 DC.IA:  HALT             ;DOA 0,DCU      ;CONTINUE
21 15557 001400         JMP 0,3          ;DISMISS INTA
22 15560 000000 DC.PK:  0
23 15561 000000 DC.LC:  0
24 15562 000000 DCS03:  0
25 15563 015701 DC.BG:  DCMS1
26 15564 000000 DC.EN:  0
27 15565 000000 DC.LO:  0
28 15566 000000 DC.LA:  0
29 15567 000000 DC.LP:  0
```

```
!0233 N3MRT
01
02                ;ERROR IN TEST DURING EXECUTION
03
04 15570 015571 DCERT:  .+1      ;ENTER LINK
05 15571 030770         LDA 2,DC.LC      ;ADRS RELOCATED IN SCRATCH
06 15572 021047         LDA 0,DCAC0-DCMS1,2
07 15573 025050         LDA 1,DCAC1-DCMS1,2
08 15574 035052         LDA 3,DCAC3-DCMS1,2
09 15575 031051         LDA 2,DCAC2-DCMS1,2
10                      LCALL ERROI
11 15576 100350         ERROI-ASCRA*1B11+100010
12 15577 000401         JMP .+1
13 15600 020441         LDA 0,DCTX1
14                      LCALL ERRTX
15 15601 100170         ERRTX-ASCRA*1B11+100010
16 15602 030757         LDA 2,DC.LC
17 15603 021053         LDA 0,DCRNN-DCMS1,2
18 15604 025054         LDA 1,DCRN1-DCMS1,2
19 15605 031055         LDA 2,DCRN2-DCMS1,2
20                      LCALL ERROC
21 15606 100370         ERROC-ASCRA*1B11+100010
22 15607 000401         JMP .+1
23 15610 020443         LDA 0,DCTX2
24                      LCALL ERRTX
25 15611 100170         ERRTX-ASCRA*1B11+100010
26 15612 030747         LDA 2,DC.LC
27 15613 021056         LDA 0,DCLOR-DCMS1,2
28 15614 025057         LDA 1,DCLPK-DCMS1,2
29 15615 031052         LDA 2,DCAC3-DCMS1,2
30                      LCALL ERROC
31 15616 100370         ERROC-ASCRA*1B11+100010
32 15617 000401         JMP .+1
33 15620 020446         LDA 0,DCTX3
34                      LCALL ERRTX
35 15621 100170         ERRTX-ASCRA*1B11+100010
36 15622 030737         LDA 2,DC.LC
37 15623 034743         LDA 3,DC.LA
38 15624 021056         LDA 0,DCLOR-DCMS1,2
39 15625 025052         LDA 1,DCAC3-DCMS1,2
40 15626 162400         SUB 3,0
41 15627 166400         SUB 3,1
42 15630 030733         LDA 2,DC.BG
43 15631 143000         ADD 2,0
44 15632 147000         ADD 2,1
45 15633 131000         MOV 1,2
46 15634 105000         MOV 0,1
47 15635 161000         MOV 3,0
48                      LCALL ERROC
49 15636 100370         ERROC-ASCRA*1B11+100010
50 15637 000666         JMP DC.04+1
51 15640 000665         JMP DC.04+1
```

```
!0234 N3MRT
01 15641 015642 DCTX1:  .+1
02 15642 005215         .TXTE !<15><12>RANDOM AC0,1,2!
03 15653 015654 DCTX2:  .+1
04 15654 005215         .TXTE !<15><12>DCLOR<11>DCLPK<11>DCLER!
05 15666 015667 DCTX3:  .+1
06 15667 005215         .TXTE !<15><12>DC.LA<11>DC.LC<11>ERROR!
07
08
09                ;DEFINE CALL LINKS FOR THE TEST TO RUN IN DCU MODE
10        006100 DCSET=JSR @100
11        006101 DCRAN=JSR @101
12        006102 DCERR=JSR @102
13        006103 DCLOO=JSR @103
14        006104 DCALL=JSR @104
15        006105 DCRET=JSR @105
16
17                ;SET UP THE CALL LINKS IN PAGE 0 OF THE DCU
18                ;AFTER THE TEST HAS STARTED
19 15701 004411 DCMS1:  JSR .+11
20 15702 000021         DSETC-.
21 15703 000056         DRANC-.
22 15704 000027         DERRC-.
23 15705 000023         DLOOC-.
24 15706 000067         DALLC-.
25 15707 000036         DRETC-.
26 15710 000000         0        ;END OF TABLE
27 15711 000100         100      ;TO LOAD PAGE 0 LINKS
28 15712 030777         LDA 2,.-1
29 15713 021400         LDA 0,0,3
30 15714 101005         MOV 0,0,SNR
31 15715 000467         JMP DCG01        ;ALL LINKS SETUP START TEST
32 15716 163000         ADD 3,0
33 15717 041000         STA 0,0,2
34 15720 151400         INC 2,2
35 15721 175400         INC 3,3
36 15722 000771         JMP .-7
```

```
10235 N3MRT
01
02
03                      ;SETUP SUBROUTINE FOR THE DCU
04 15723 054434 DSETC:  STA 3,DCLOR
05 15724 034403         LDA 3,.+3
06 15725 054433         STA 3,DCLPK
07 15726 002431         JMP @DCLOR
08 15727 177774         -4
09
10                      ;LOOP SUBROUTINE FOR THE DCU TEST
11 15730 010430 DLOOC:  ISZ DCLPK
12 15731 002426         JMP @DCLOR
13 15732 001400         JMP 0,3
14                      ;ERROR SUBROUTINE FOR THE DCU TEST
15 15733 040415 DERRC:  STA 0,DCAC0
16 15734 044415         STA 1,DCAC1
17 15735 050415         STA 2,DCAC2
18 15736 102000         ADC 0,0
19 15737 117000         ADD 0,3
20 15740 054413         STA 3,DCAC3
21 15741 020405         LDA 0,DRETC+1
22 15742 040000         STA 0,0
23 15743 060176         NIOS 76
24 15744 000000         JMP 0
25                      ;END OF TEST HALT WILL LEAVE DCACX'S SET TO 0
26                      ;ERROR IN DCU ARITH TEST IS INDICATED BY NONZERO DCAC3
27 15745 060176 DRETC:  NIOS 76          ;INTERRUPT HOST
28 15746 063077         HALT             ;AND STOP DCU
29 15747 000732         JMP DCMS1        ;CONTINUE WAS SEEN
30 15750 000000 DCAC0:  0
31 15751 000000 DCAC1:  0
32 15752 000000 DCAC2:  0
33 15753 000000 DCAC3:  0
34 15754 000000 DCRNN:  0
35 15755 000000 DCRN1:  0
36 15756 000000 DCRN2:  0
37 15757 000000 DCLOR:  0
38 15760 000000 DCLPK:  0
39                      ;SHORT RANDOM # GENERATOR FOR DCU ARITH
40 15761 020773 DRANC:  LDA     0,DCRNN ;FILL AC0, 1 AND WITH RANDOM
41 15762 024773         LDA     1,DCRN1
42 15763 030773         LDA     2,DCRN2
43 15764 125045         MOVO    1,1,SNR
44 15765 024772         LDA     1,DCLOR
45 15766 133100         ADDL    1,2
46 15767 143160         ADDCL   2,0
47 15770 125200         MOVR    1,1
48 15771 040763         STA     0,DCRNN
49 15772 044763         STA     1,DCRN1
50 15773 050763         STA     2,DCRN2
51 15774 001400         JMP     0,3
```

```
10236 N3MRT
01
02                      ;DCU CALL SUBROUTINE USED TO LINK
03                      ;TO SPECIAL SUBROUTINS THAT CAN' BE REACHED BY DIRECT JSR'S
04                      ;WITHIN THE DCU ARITHMETIC TEST
05 15775 040753 DALLC:  STA 0,DCAC0
06 15776 021400         LDA 0,0,3
07 15777 163000         ADD 3,0
08 16000 040751         STA 0,DCAC1
09 16001 175400         INC 3,3
10 16002 020746         LDA 0,DCAC0
11 16003 002746         JMP @DCAC1
12                      ;THE FOLLOWING MACRO'S DEFINE THE CALL PARAMETERS
13                      ;UTILIZED BY THE INDIVIDUAL TESTS
14                              .MACRO DXRA
15                              DXR.0-.
16                      %
17                              .MACRO DXR2
18                              DXR.2-.
19                      %
20                              .MACRO DXR1
21                              DXR.1-.
22                      %
23                              .MACRO DCMPU
24                              D.MPU-.
25                      %
26                              .MACRO DCMPA
27                              D.MPA-.
28                      %
29                              .MACRO DCSQR
30                              D.SQR-.
31                      %
32                              .MACRO DCSQ
33                              D.SQ.-.
34                      %
35                              .MACRO DCDIV
36                              D.DVU-.
37                      %
```

```
!0237 N3MRT
01 16004 020407 DCG01:  LDA 0,DCEJM
02 16005 040000         STA 0,0          ;PLACE HALT IN LOC 0
03 16006 006100         DCSET            ;MISC TEST OF NEG/COM
04 16007 006101         DCRAN
05 16010 104700         NEGS 0,1
06 16011 130342         COMOS 1,2,SZC
07 16012 142014         ADC# 2,0,SZR
08 16013 006102 DCEJM:  DCERR
09 16014 006103         DCLOO
10
11                ;MS2:
12 16015 006100         DCSET            ;MISC TEST OF INC SWAPPED.
13 16016 006101         DCRAN
14 16017 111700         INCS 0,2
15 16020 145323         MOVZS 2,1,SNC
16 16021 106314         ADCS# 0,1,SZR
17 16022 006102         DCERR
18 16023 006103         DCLOO
19
20                ;MS3:
21 16024 006100         DCSET            ;MISC NEG TEST
22 16025 102400         SUB 0,0
23 16026 100644         NEGOR 0,0,SZR
24 16027 006102         DCERR
25 16030 100664         NEGCR 0,0,SZR
26 16031 006102         DCERR
27 16032 100664         NEGCR 0,0,SZR
28 16033 006102         DCERR
29 16034 006103         DCLOO
30
```

```
!0238 N3MRT
01
02                ;ROT1:
03 16035 006100         DCSET            ;TEST ROTATE LEFT
04 16036 006101         DCRAN
05 16037 105000         MOV 0,1
06 16040 101100         MOVL 0,0
07 16041 111100         MOVL 0,2
08 16042 151100         MOVL 2,2
09 16043 151100         MOVL 2,2
10 16044 155100         MOVL 2,3
11 16045 175100         MOVL 3,3
12 16046 175100         MOVL 3,3
13 16047 161100         MOVL 3,0
14 16050 101100         MOVL 0,0
15 16051 101100         MOVL 0,0
16 16052 101100         MOVL 0,0
17 16053 101100         MOVL 0,0
18 16054 101100         MOVL 0,0
19 16055 101100         MOVL 0,0
20 16056 101100         MOVL 0,0
21 16057 101100         MOVL 0,0
22 16060 101100         MOVL 0,0
23 16061 106414         SUB# 0,1,SZR
24 16062 006102         DCERR
25 16063 006103         DCLOO
26
27                ;ROT2:
28 16064 006100         DCSET            ;TEST ROTATE RIGHT
29 16065 006101         DCRAN
30 16066 131000         MOV 1,2
31 16067 121200         MOVR 1,0
32 16070 101200         MOVR 0,0
33 16071 101200         MOVR 0,0
34 16072 101200         MOVR 0,0
35 16073 101200         MOVR 0,0
36 16074 101200         MOVR 0,0
37 16075 101200         MOVR 0,0
38 16076 101200         MOVR 0,0
39 16077 101200         MOVR 0,0
40 16100 101200         MOVR 0,0
41 16101 105200         MOVR 0,1
42 16102 125200         MOVR 1,1
43 16103 135200         MOVR 1,3
44 16104 175200         MOVR 3,3
45 16105 175200         MOVR 3,3
46 16106 161200         MOVR 3,0
47 16107 101200         MOVR 0,0
48 16110 112414         SUB# 0,2,SZR
49 16111 006102         DCERR
50 16112 006103         DCLOO
```

```
!0239 N3MRT
01
02                ;AND0:
03 16113 006100           DCSET             ;ANY NUMBER ANDED WITH
04 16114 006101           DCRAN             ;ITSELF SHOULD NOT
05 16115 131000           MOV 1,2           ;BE CHANGED.
06 16116 127700           ANDS 1,1
07 16117 127700           ANDS 1,1
08 16120 127700           ANDS 1,1
09 16121 127700           ANDS 1,1
10 16122 127700           ANDS 1,1
11 16123 135300           MOVS 1,3
12 16124 177700           ANDS 3,3
13 16125 177700           ANDS 3,3
14 16126 161300           MOVS 3,0
15 16127 103700           ANDS 0,0
16 16130 112414           SUB# 0,2,SZR
17 16131 006102           DCERR
18 16132 006103           DCLOO
19
20                ;AND1:
21 16133 006100           DCSET             ;A NUMBER ANDED WITH ITS
22 16134 006101           DCRAN             ;COMPLIMENT SHOULD
23 16135 104042           COMO 0,1,SZC
24 16136 123704           ANDS 1,0,SZR      ;PRODUCE ZERO RESULT.
25 16137 006102           DCERR
26 16140 006103           DCLOO
27
28                ;AND3:
29 16141 006100           DCSET             ;PERFORM A AND INSTRUCTION
30 16142 006101           DCRAN             ;WITH THE RESULT IN AC2.
31 16143 176620           SUBZR 3,3         ;SIMULATE THE AND VIA
32 16144 131000           MOV 1,2           ;LOOKING FOR ADDER CARRY.
33 16145 113400           AND 0,2
34 16146 101113 AND.L:    MOVL# 0,0,SNC
35 16147 000403           JMP .+3
36 16150 125112           MOVL# 1,1,SZC
37 16151 101141           MOVOL 0,0,SKP
38 16152 101121           MOVZL 0,0,SKP
39 16153 125141           MOVOL 1,1,SKP
40 16154 125120           MOVZL 1,1
41 16155 175224           MOVZR 3,3,SZR
42 16156 000770           JMP AND.L
43 16157 106415           SUB# 0,1,SNR      ;CHECK IF AC0-1 ARE
44 16160 132414           SUB# 1,2,SZR      ;THE SAME AND IF THEY
45 16161 006102           DCERR             ;ARGEE WITH INST.
46 16162 006103           DCLOO
```

```
!0240 N3MRT
01                ;TEST PROCESSOR VIA EXCLUSIVE OR ROUTINES.
02
03                ;X1:
04 16163 006100           DCSET             ;C(AC1) IS SAVED IN C(AC2).
05 16164 006101           DCRAN             ;AC0 IS EXCLUSIVE ORED
06 16165 131000           MOV 1,2           ;WITH AC1 TWICE. THE SECOND
07 16166 006104           DCALL             ;EXCLUSIVE OR SHOULD
08                        DXRA              ;RESTORE AC1 TO ITS
09 16167 000063           DXR.0-.
10 16170 006104           DCALL             ;ORIGIONAL CONTENTS.
11                        DXRA
12 16171 000061           DXR.0-.
13 16172 132414           SUB# 1,2,SZR
14 16173 006102           DCERR
15 16174 006103           DCLOO
16
17                ;X2:
18 16175 006100           DCSET             ;THE FIRST EXCLUSIVE OR
19 16176 006101           DCRAN             ;ROUTINE EXCHANGES THE
20 16177 006104           DCALL             ;CONTENTS
21                        DXR2              ;OF AC0 AND AC1, IT ALSO
22 16200 000070           DXR.2-.
23 16201 006104           DCALL             ;FORMS THE EXCLUSIVE OR
24                        DXR1              ;IN AC2. THE SECOND EXCLUSIVE
25 16202 000056           DXR.1-.
26 16203 112414           SUB# 0,2,SZR      ;OR FORMS THE RESULT
27 16204 006102           DCERR             ;OF AC0-AC1 IN AC0.
28 16205 006103           DCLOO
29
30                ;X3:
31 16206 006100           DCSET             ;SAVE C(AC1) NEGATED IN
32 16207 006101           DCRAN             ;C(AC2). EXCLUSIVE OR C(AC0) TO
33 16210 130400           NEG 1,2           ;C(AC1). EXCLUSIVE OR THE
34 16211 006104           DCALL             ;RESULT BACK TO AC0.
35                        DXRA              ;CHECK VIA ADDITION TO
36 16212 000040           DXR.0-.
37 16213 006104           DCALL             ;COMPLIMENT OF ORIGIONAL
38                        DXR1              ;NUMBER.
39 16214 000044           DXR.1-.
40 16215 113014           ADD# 0,2,SZR
41 16216 006102           DCERR
42 16217 006103           DCLOO
43
44                ;X4:
45 16220 006100           DCSET             ;EXCLUSIVE OR C(AC0) TO
46 16221 006101           DCRAN             ;ALL ZEROS IN C(AC1).
47 16222 126400           SUB 1,1
48 16223 006104           DCALL
49                        DXRA
50 16224 000026           DXR.0-.
51 16225 106414           SUB# 0,1,SZR
52 16226 006102           DCERR
53 16227 006103           DCLOO
```

```
!0241 N3MRT
01
02 16230 006100                ;X5:
03 16231 006101                        DCSET           ;EXCLUSIVE OR C(AC1) TO
04 16232 102400                        DCRAN           ;ALL ZEROS IN C(AC0).
05 16233 006104                        SUB 0,0
06                                     DCALL
07 16234 000024                        DXR1
08 16235 106414                        DXR.1-.
09 16236 006102                        SUB# 0,1,SZR
10 16237 006103                        DCERR
11                                     DCLOO
12
13 16240 006100                ;X6:
14 16241 006101                        DCSET           ;EXCLUSIVE OR C(AC1) TO
15 16242 102000                        DCRAN           ;(-1) IN C(AC0).
16 16243 006104                        ADC 0,0         ;THE COMPLIMENT OF
17                                     DCALL           ;THIS RESULT SHOULD
18 16244 000014                        DXR1            ;EQUAL C(AC1).
19 16245 110000                        DXR.1-.
20 16246 146414                        COM 0,2
21 16247 006102                        SUB# 2,1,SZR
22 16250 006103                        DCERR
23 16251 000436                        DCLOO
24                                     JMP X7.
25 16252 054433 DXR.0:         STA 3,DXR.4
26 16253 135000                        MOV 1,3         ;EXCLUSIVE OR C(AC0),C(AC1).
27 16254 117520                        ANDZL 0,3       ;RESULT IS IN C(AC1).
28 16255 107000                        ADD 0,1
29 16256 166400                        SUB 3,1
30 16257 002426                        JMP @DXR.4
31
32 16260 054425 DXR.1:         STA 3,DXR.4
33 16261 135000                        MOV 1,3         ;EXCLUSIVE OR C(AC0),C(AC1).
34 16262 117400                        AND 0,3         ;RESULT IS IN C(AC0).
35 16263 174000                        COM 3,3
36 16264 163400                        AND 3,0
37 16265 123000                        ADD 1,0
38 16266 163400                        AND 3,0
39 16267 002416                        JMP @DXR.4
40
41 16270 054415 DXR.2:         STA 3,DXR.4
42 16271 034415                        LDA 3,XM.20     ;EXCLUSIVE OR C(AC0),C(AC1).
43 16272 054412                        STA 3,DXRTEM    ;RESULT IN C(AC2).
44 16273 115000                        MOV 0,3         ;THE CONTENTS OF AC0 AND
45 16274 137200                        ADDR 1,3        ;AC1 ARE EXCHANGED.
46 16275 151200                        MOVR 2,2
47 16276 101220                        MOVZR 0,0
48 16277 125200                        MOVR 1,1
49 16300 103200                        ADDR 0,0
50 16301 010403                        ISZ DXRTEM
51 16302 000771                        JMP DXR.2+3
52 16303 002402                        JMP @DXR.4
53 16304 000000 DXRTEM: 0
54 16305 000000 DXR.4: 0
55 16306 177760 XM.20: -20
```

```
!0242 N3MRT
01
02 16307 006100 X7.:           DCSET           ;C(AC1) IS SET EQUAL TO
03 16310 006101                DCRAN           ;C(AC0). C(AC0) AND C(AC1)
04 16311 105000                MOV 0,1         ;ARE EXCLUSIVE ORED WITH
05 16312 006104                DCALL           ;THE RESULT GOING TO AC2.
06                             DXR2
07 16313 177755                DXR.2-.
08 16314 151004                MOV 2,2,SZR
09 16315 006102                DCERR
10 16316 006103                DCLOO
11
12                     ;X8:
13 16317 006100                DCSET           ;C(AC0) IS SET TO THE
14 16320 006101                DCRAN           ;COMPLIMENT OF C(AC1). THE
15 16321 120000                COM 1,0         ;RESULT OF A EXCLUSIVE OR
16 16322 006104                DCALL           ;SHOULD BE ALL BITS
17                             DXR2            ;SET (-1).
18 16323 177745                DXR.2-.
19 16324 150014                COM# 2,2,SZR
20 16325 006102                DCERR
21 16326 006103                DCLOO
22                     ;X9:
23 16327 006100                DCSET           ;EXCLUSIVE OR ALL ONES
24 16330 102300                ADCS 0,0        ;TO ALL ONES. THE
25 16331 126000                ADC 1,1         ;RESULT SHOULD BE
26 16332 006104                DCALL           ;ALL ZEROS.
27                             DXRA
28 16333 177717                DXR.0-.
29 16334 127704                ANDS 1,1,SZR
30 16335 006102                DCERR
31 16336 006103                DCLOO
32
33                     ;X10:
34 16337 006100                DCSET           ;EXCLUSIVE OR ALL ZEROS
35 16340 102700                SUBS 0,0        ;TO ALL ZEROS. THE
36 16341 126400                SUB 1,1         ;RESULT SHOULD BE
37 16342 006104                DCALL           ;ALL ZEROS IN C(AC1).
38                             DXRA
39 16343 177707                DXR.0-.
40 16344 107314                ADDS# 0,1,SZR
41 16345 006102                DCERR
42 16346 006103                DCLOO
```

```
!0243 N3MRT
01
02                ;X11:
03 16347 006100           DCSET             ;FORM EXCLUSIVE OF FUNCTION
04 16350 006101           DCRAN             ;IN C(AC2).
05 16351 006104           DCALL             ;CALL ANOTHER EXCLUSIVE
06                        DXR2              ;OR FUNCTION SEVEN TIMES.
07 16352 177716           DXR.2-.
08 16353 006104           DCALL             ;THE RESULT SHOULD BE THE
09                        DXRA              ;SAME AS THE FIRST XORA
10 16354 177676           DXR.0-.
11 16355 006104           DCALL
12                        DXRA
13 16356 177674           DXR.0-.
14 16357 006104           DCALL
15                        DXRA
16 16360 177672           DXR.0-.
17 16361 006104           DCALL
18                        DXRA
19 16362 177670           DXR.0-.
20 16363 006104           DCALL
21                        DXRA
22 16364 177666           DXR.0-.
23 16365 006104           DCALL
24                        DXRA
25 16366 177664           DXR.0-.
26 16367 006104           DCALL
27                        DXRA
28 16370 177662           DXR.0-.
29 16371 132414           SUB# 1,2,SZR
30 16372 006102           DCERR
31 16373 006103           DCLOO
32
```

```
!0244 N3MRT
01                ;A1:
02 16374 006100           DCSET             ;SAVE C(AC2) IN C(AC1).
03 16375 006101           DCRAN             ;ADD AND ADD C(AC0) TO
04 16376 145000           MOV 2,1           ;C(AC2). THE VALUE IN
05 16377 112400           SUB 0,2           ;AC2 SHOULD NOT BE
06 16400 113000           ADD 0,2           ;CHANGED.
07 16401 146414           SUB# 2,1,SZR
08 16402 006102           DCERR
09 16403 006103           DCLOO
10
11                ;A2:
12 16404 006100           DCSET
13 16405 006101           DCRAN
14 16406 102000           ADC 0,0
15 16407 123000           ADD 1,0
16 16410 111400           INC 0,2
17 16411 146414           SUB# 2,1,SZR
18 16412 006102           DCERR
19 16413 006103           DCLOO
20                ;A3:
21 16414 006100           DCSET             ;INCREMENT THE VALUE IN
22 16415 006101           DCRAN             ;AC1 AND ADD THAT VALUE
23 16416 131460           INCC 1,2          ;TO (-1). THE RESULT
24 16417 102360           ADCCS 0,0         ;SHOULD BE THE ORIGIONAL
25 16420 143060           ADDC 2,0          ;NUMBER.
26 16421 106414           SUB# 0,1,SZR
27 16422 006102           DCERR
28 16423 006103           DCLOO
29                ;A4:
30 16424 006100           DCSET             ;SAVE THE C(AC1) IN C(AC0).
31 16425 006101           DCRAN             ;A "ADDR" INSTRUCTION SHOULD
32 16426 121020           MOVZ 1,0          ;NOT CHANGE THE VALUE OF
33 16427 127200           ADDR 1,1          ;THE AC.
34 16430 127200           ADDR 1,1
35 16431 127200           ADDR 1,1
36 16432 127200           ADDR 1,1
37 16433 127200           ADDR 1,1
38 16434 127200           ADDR 1,1
39 16435 127200           ADDR 1,1
40 16436 127200           ADDR 1,1
41 16437 127200           ADDR 1,1
42 16440 127200           ADDR 1,1
43 16441 127200           ADDR 1,1
44 16442 127200           ADDR 1,1
45 16443 127200           ADDR 1,1
46 16444 127200           ADDR 1,1
47 16445 106454           SUBO# 0,1,SZR
48 16446 006102           DCERR
49 16447 006103           DCLOO
```

```
10245 N3MRT
01
02                      ;A5:
03 16450 006100                 DCSET           ;THE DCRAN NUMBER IN
04 16451 006101                 DCRAN           ;C(AC0) IS INCREMENTED VIA "INC"
05 16452 115400                 INC 0,3         ;AND DECREMENTED VIA "ADD".
06 16453 126000                 ADC 1,1         ;THE FINAL RESULT IN C(AC2)
07 16454 137000                 ADD 1,3         ;SHOULD BE EQUAL TO THE
08 16455 175400                 INC 3,3         ;ORIGIONAL NUMBER IN C(AC0).
09 16456 137000                 ADD 1,3
10 16457 171400                 INC 3,2
11 16460 133000                 ADD 1,2
12 16461 151400                 INC 2,2
13 16462 133000                 ADD 1,2
14 16463 142414                 SUB# 2,0,SZR
15 16464 006102                 DCERR
16 16465 006103                 DCLOO
17
18                      ;A6:
19 16466 006100                 DCSET           ;THE SUM OF AC0-1 IS
20 16467 006101                 DCRAN           ;CHECKED WITH THE SIMULATED
21 16470 135000                 MOV 1,3         ;SUM.
22 16471 117000                 ADD 0,3
23 16472 054421                 STA 3,AD.DT     ;SUM VIA ADD INSTRUCTION.
24 16473 131000 A6L.:           MOV 1,2
25 16474 006104                 DCALL
26                              DXRA            ;SIMULATE THE ADD VIA
27 16475 177555                 DXR.0-.
28 16476 143524                 ANDZL 2,0,SZR   ;EXCLUSIVE OR. C(AC2)=
29 16477 000774                 JMP A6L.                ;RIPPLE CARRY,C(AC1)=RESULT.
30 16500 020413                 LDA 0,AD.DT
31 16501 122414                 SUB# 1,0,SZR
32 16502 006102                 DCERR
33 16503 006103                 DCLOO
34
35                      ;A7:
36 16504 006100                 DCSET           ;ADDITION OF NEGATED
37 16505 006101                 DCRAN           ;NUMBERS SHOULD PRODUCE
38 16506 110440                 NEGO 0,2        ;ZERO AND A CARRY.
39 16507 143204                 ADDR 2,0,SZR
40 16510 006102                 DCERR
41 16511 006103                 DCLOO
42
43 16512 101001                 MOV 0,0,SKP
44 16513 177777 AD.DT:          -1
```

```
10246 N3MRT
01
02                      ;A8:
03 16514 006100                 DCSET           ;ADD TEST.
04 16515 006101                 DCRAN
05 16516 152520                 SUBZL 2,2       ;=(+1)
06 16517 151300                 MOVS 2,2        ;=(400)
07 16520 150400                 NEG 2,2         ;=(177400)
08 16521 143700                 ANDS 2,0        ;SAVE HIGH ORDER 8 BITS.
09 16522 105320                 MOVZS 0,1       ;SAME 8 BITS TO C(AC1)L.
10 16523 103100                 ADDL 0,0        ;MOVE C(AC0) LEFT VIA
11 16524 103100                 ADDL 0,0        ;ADD SHIFT.
12 16525 103100                 ADDL 0,0
13 16526 103100                 ADDL 0,0
14 16527 106414                 SUB# 0,1,SZR
15 16530 006102                 DCERR
16 16531 006103                 DCLOO
17
18
19
20 16532 101001                 MOV 0,0,SKP
21 16533 177400 M400.:          -400
22                      ;AR1:
23 16534 006100                 DCSET           ;THE ORIGIONAL CONTENTS OF
24 16535 006101                 DCRAN           ;AC1, BITS 0-7 ARE SQUARED
25 16536 030775                 LDA 2,M400.     ;VIA MULTIPLY. THE SQUARE
26 16537 133700                 ANDS 1,2        ;ROOT OF THE PRODUCT SHOULD
27 16540 145000                 MOV 2,1         ;EQUAL THE ORIGIONAL.
28 16541 006104                 DCALL
29                              DCMPU           ;SEE SYSTEM REFFERANCE
30 16542 000042                 D.MPU-.
31 16543 121000                 MOV 1,0         ;MANUAL FOR FURTHER INFORMATION
32 16544 006104                 DCALL           ;ON MULTIPLY/SQ ROOT
33                              DCSQR           ;PROGRAMS.
34 16545 000066                 D.SQR-.
35 16546 112414                 SUB# 0,2,SZR
36 16547 006102                 DCERR
37 16550 006103                 DCLOO
```

```
!0247 N3MRT
01
02                ;AR2:
03 16551 006100           DCSET             ;TAKE THE SQUARE ROOT
04 16552 006101           DCRAN             ;OF A NUMBER. THE SQUARE
05 16553 006104           DCALL             ;ROOT OF THE RESULT SQUARED
06                        DCSQR             ;SHOULD BE THE SAME AS THE
07 16554 000057           D.SQR-.
08 16555 105060           MOVC 0,1          ;ORIGIONAL ROOT.
09 16556 131060           MOVC 1,2
10 16557 006104           DCALL
11                        DCMPU
12 16560 000024           D.MPU-.
13 16561 121000           MOV 1,0
14 16562 006104           DCALL             ;SQUARED NOW TAKE ROOT.
15                        DCSQ
16 16563 000060           D.SQ.-.
17 16564 112714           SUBS# 0,2,SZR
18 16565 006102           DCERR
19 16566 006103           DCLOO
20
21                ;AR3:
22 16567 006100           DCSET             ;FIND SQUARE ROOT VIA
23 16570 006101           DCRAN             ;DIFFIRENT SUBROUTINES.
24 16571 141000           MOV 2,0
25 16572 006104           DCALL
26                        DCSQR
27 16573 000040           D.SQR-.
28 16574 105000           MOV 0,1           ;SAVE FIRST RESULT IN AC1
29 16575 141000           MOV 2,0
30 16576 006104           DCALL
31                        DCSQ              ;RESULT IN AC0.
32 16577 000044           D.SQ.-.
33 16600 106714           SUBS# 0,1,SZR
34 16601 006102           DCERR
35 16602 006103           DCLOO
36 16603 000455           JMP AR4.
```

```
!0248 N3MRT
01
02 16604 102460 D.MPU:    SUBC 0,0          ;C(AC1)*C(AC2)
03 16605 054411 D.MPA:    STA 3,C.B03       ;RESULT IN AC0,AC1.
04 16606 034411           LDA 3,C.B20       ;SEE SYSTEM REFFERANCE
05 16607 125203 C.B99:    MOVR 1,1,SNC      ;MANUAL FOR FURTHER
06 16610 101201           MOVR 0,0,SKP      ;INFORMATION.
07 16611 143220           ADDZR 2,0
08 16612 175404           INC 3,3,SZR
09 16613 000774           JMP C.B99
10 16614 125260           MOVCR 1,1
11 16615 002401           JMP @C.B03
12 16616 000000 C.B03:    0
13 16617 177760 C.B20:    -20
14
15 16620 102400 D.DVI:    SUB 0,0           ;C(AC0),C(AC1)/C(AC2).
16 16621 054775 D.DVU:    STA 3,C.B03       ;AC0=REMAINDER
17 16622 034775           LDA 3,C.B20       ;AC1=QUOTIENT
18 16623 125120           MOVZL 1,1         ;SEE SYSTEM REFFERANCE
19 16624 101100 C.C98:    MOVL 0,0          ;MANUAL.
20 16625 142412           SUB# 2,0,SZC
21 16626 142400           SUB 2,0
22 16627 125100           MOVL 1,1
23 16630 175404           INC 3,3,SZR
24 16631 000773           JMP C.C98
25 16632 002764           JMP @C.B03
26
27
28 16633 054423 D.SQRT.:        STA 3,D.SQS3
29 16634 126520           SUBZL 1,1         ;FIND D.SQ ROOT OF C(AC0).
30 16635 135120           MOVZL 1,3         ;SEE THE SYSTEM REFFERANCE
31 16636 122422           SUBZ 1,0,SZC      ;MANUAL.
32 16637 167001           ADD 3,1,SKP       ;RESULT IN AC0.
33 16640 121221           MOVZR 1,0,SKP     ;AC1 DESTROYED.
34 16641 000775           JMP .-3
35 16642 002414           JMP @D.SQS3
36
37 16643 054413 D.SQ.:    STA 3,D.SQS3
38 16644 176400           SUB 3,3           ;FIND D.SQ ROOT OF C(AC0).
39 16645 054412           STA 3,D.SQTEM     ;SAME RESULT AS PREVIOUS
40 16646 162023           ADCZ 3,0,SNC      ;TEST BUT CODE IS
41 16647 000405           JMP D.SQ1                 ;DIFFIRENT.
42 16650 010407           ISZ D.SQTEM
43 16651 010406           ISZ D.SQTEM
44 16652 034405           LDA 3,D.SQTEM     ;RESULT WILL BE IN AC0.
45 16653 000773           JMP .-5
46 16654 161200 D.SQ1:    MOVR 3,0
47 16655 002401           JMP @D.SQS3
48
49 16656 000000 D.SQS3: 0
50 16657 000000 D.SQTEM: 0
```

```
!0249 N3MRT
01
02 16660 006100 AR4.:   DCSET           ;THE CONTENTS OF AC2 IS
03 16661 006101 AR4L.:  DCRAN           ;DIVIDED INTO AC0-1.
04 16662 040461         STA 0,OAC.0
05 16663 044461         STA 1,OAC.1
06 16664 050461         STA 2,OAC.2
07 16665 142432         SUBZ# 2,0,SZC   ;THIS RESULT MULTIPLIED
08 16666 000773         JMP AR4L.               ;BY AC2 SHOULD PRODUCE
09 16667 006104         DCALL           ;THE ORIGIONAL NUMBERS.
10                      DCDIV
11 16670 177731         D.DVU-.
12 16671 006104         DCALL
13                      DCMPA
14 16672 177713         D.MPA-.
15 16673 034452         LDA 3,OAC.2
16 16674 156414         SUB# 2,3,SZR
17 16675 006102         DCERR           ;AC2 CHANGED?
18 16676 034446         LDA 3,OAC.1
19 16677 136714         SUBS# 1,3,SZR
20 16700 006102         DCERR           ;AC1 CHANGED.
21 16701 034442         LDA 3,OAC.0
22 16702 116714         SUBS# 0,3,SZR
23 16703 006102         DCERR           ;AC0 CHANGED.
24 16704 006103         DCLOO
25
26              ;AR5:
27 16705 006100         DCSET           ;USE INC TO FORM
28 16706 152440         SUBO 2,2        ;THE NUMBER 177400
29 16707 151504         INCL 2,2,SZR    ;IN AC1 AND 400 IN AC2.
30 16710 151504         INCL 2,2,SZR
31 16711 151504         INCL 2,2,SZR
32 16712 151504         INCL 2,2,SZR
33 16713 151507         INCL 2,2,SBN    ;THE "SBN/SZR" SHOULD
34 16714 151507         INCL 2,2,SBN    ;NOT CAUSE A SKIP.
35 16715 151507         INCL 2,2,SBN
36 16716 145707         INCS 2,1,SBN
37 16717 151407         INC 2,2,SBN
38 16720 151400         INC 2,2
39 16721 133014         ADD# 1,2,SZR
40 16722 006102         DCERR
41 16723 006103         DCLOO
```

```
!0250 N3MRT
01
02              ;AR6:
03 16724 006100         DCSET           ;USE THE INCR INSTRUCTION
04 16725 126420         SUBZ 1,1        ;TO FORM THE NUMBER
05 16726 125606         INCR 1,1,SEZ    ;177400. THIS NUMBER
06 16727 125606         INCR 1,1,SEZ    ;COMPLIMENTED AND SWAPPED
07 16730 125606         INCR 1,1,SEZ    ;SHOULD BE THE SAME
08 16731 125606         INCR 1,1,SEZ    ;NUMBER.
09 16732 125606         INCR 1,1,SEZ    ;THE "SEZ" SHOULD NEVER
10 16733 125606         INCR 1,1,SEZ    ;CAUSE A SKIP.
11 16734 125606         INCR 1,1,SEZ
12 16735 125606         INCR 1,1,SEZ
13 16736 130304         COMS 1,2,SZR
14 16737 132714         SUBS# 1,2,SZR
15 16740 006102         DCERR
16 16741 006103         DCLOO
17 16742 000404         JMP .+4
18 16743 000000 OAC.0: 0
19 16744 000000 OAC.1: 0
20 16745 000000 OAC.2: 0
21
22              ;AR7:
23 16746 006100         DCSET           ;100000 NEGATED IS
24 16747 102625         SUBZR 0,0,SNR   ;STILL 100000.
25 16750 006102         DCERR
26 16751 104406         NEG 0,1,SEZ     ;C(CARRY)=0
27 16752 006102         DCERR
28 16753 124402         NEG 1,1,SZC
29 16754 006102         DCERR
30 16755 124463         NEGC 1,1,SNC    ;C(CARRY)=1
31 16756 006102         DCERR
32 16757 124407         NEG 1,1,SBN
33 16760 006102         DCERR
34 16761 124465         NEGC 1,1,SNR    ;C(1)=100000
35 16762 006102         DCERR
36 16763 124465         NEGC 1,1,SNR
37 16764 006102         DCERR
38 16765 124407         NEG 1,1,SBN
39 16766 006102         DCERR
40 16767 124407         NEG 1,1,SBN
41 16770 006102         DCERR
42 16771 106414         SUB# 0,1,SZR
43 16772 006102         DCERR
44 16773 006103         DCLOO
```

```
10251 N3MRT
01                ;AR8:
02 16774 006100           DCSET             ;NEGATION OF ZERO
03 16775 102440           SUBO 0,0          ;SHOULD PRODUCE ZERO
04 16776 100762           NEGCS 0,0,SZC     ;AND A CARRY.
05 16777 006102           DCERR
06 17000 100762           NEGCS 0,0,SZC
07 17001 006102           DCERR
08 17002 100706           NEGS 0,0,SEZ
09 17003 006102           DCERR
10 17004 100706           NEGS 0,0,SEZ
11 17005 006102           DCERR
12 17006 100544           NEGOL 0,0,SZR
13 17007 006102           DCERR
14 17010 100403           NEG 0,0,SNC
15 17011 006102           DCERR
16 17012 100644           NEGOR 0,0,SZR
17 17013 006102           DCERR
18 17014 104704           NEGS 0,1,SZR
19 17015 006102           DCERR
20 17016 130704           NEGS 1,2,SZR
21 17017 006102           DCERR
22 17020 006103           DCLOO
23
24                ;AR9:
25 17021 006100           DCSET             ;C(AC1)*0+C(AC0) SHOULD
26 17022 006101           DCRAN             ;PLACE AC0 IN AC1. SEE
27 17023 040720           STA 0,0AC.0
28 17024 152400           SUB 2,2           ;SYSTEM REFFERANCE MANUAL
29 17025 006104           DCALL             ;FOR FURTHER INFORMATION.
30                        DCMPA
31 17026 177557           D.MPA-.
32 17027 034714           LDA 3,0AC.0
33 17030 166414           SUB# 3,1,SZR
34 17031 006102           DCERR
35 17032 006103           DCLOO
36
37                ;AR10:
38 17033 006100           DCSET             ;TEST "COM/INC" SWAPPED.
39 17034 102300           ADCS 0,0
40 17035 105705           INCS 0,1,SNR
41 17036 130304           COMS 1,2,SZR
42 17037 145705           INCS 2,1,SNR
43 17040 124346           COMOS 1,1,SEZ
44 17041 131707           INCS 1,2,SBN
45 17042 150304           COMS 2,2,SZR
46 17043 151704           INCS 2,2,SZR
47 17044 006102           DCERR
48 17045 006103           DCLOO
```

```
10252 N3MRT
01                ;AR11:
02 17046 006100           DCSET             ;COMPLIMENT AND INCREMENT
03 17047 006101           DCRAN             ;SHOULD BE THE SAME AS
04 17050 110400           NEG 0,2           ;NEGATE
05 17051 104000           COM 0,1
06 17052 125400           INC 1,1
07 17053 132414           SUB# 1,2,SZR
08 17054 006102           DCERR
09 17055 006103           DCLOO
10
11                ;AR12:
12 17056 006100           DCSET             ;TEST COM LEFT AND RIGHT.
13 17057 006101           DCRAN
14 17060 110100           COML 0,2
15 17061 144260           COMCR 2,1
16 17062 106714           SUBS# 0,1,SZR
17 17063 006102           DCERR
18 17064 006103           DCLOO
19
20                ;AR13:
21 17065 006100           DCSET             ;FORM THE PARITY OF
22 17066 105020           MOVZ 0,1          ;C(AC0) IN DIFFIRENT
23 17067 176000           ADC 3,3           ;ROUTINES. CHECK THAT
24 17070 117000           ADD 0,3           ;THE RESULTS ARE EQUAL.
25 17071 163704           ANDS 3,0,SZR
26 17072 000775           JMP .-3
27 17073 102660           SUBCR 0,0         ;SAVE PARITY IN BIT 0.
28 17074 176620           SUBZR 3,3
29 17075 125102           MOVL 1,1,SZC
30 17076 101400           INC 0,0           ;BIT 15 WILL CONTAIN
31 17077 175224           MOVZR 3,3,SZR     ;THE PARITY.
32 17100 000775           JMP .-3
33 17101 115200           MOVR 0,3          ;CHECK TO SEE IF BITS
34 17102 103012           ADD# 0,0,SZC      ;ARE LIKE.
35 17103 006102           DCERR
36 17104 006103           DCLOO
37
38                ;AR15:
39 17105 006100           DCSET             ;MISC TEST OF SUB LEFT
40 17106 006101           DCRAN             ;AND RIGHT.
41 17107 131000           MOV 1,2
42 17110 106500           SUBL 0,1
43 17111 127600           ANDR 1,1
44 17112 107000           ADD 0,1
45 17113 106500           SUBL 0,1
46 17114 127600           ANDR 1,1
47 17115 107000           ADD 0,1
48 17116 132414           SUB# 1,2,SZR
49 17117 006102           DCERR
50 17120 006103           DCLOO
```

```
10253 N3MRT
01                ;AR16:
02 17121 006100           DCSET             ;MISC TEST OF AND RIGHT.
03 17122 006101           DCRAN
04 17123 131040           MOVO 1,2
05 17124 113600           ANDR 0,2
06 17125 107400           AND 0,1
07 17126 151100           MOVL 2,2
08 17127 146414           SUB# 2,1,SZR
09 17130 006102           DCERR
10 17131 006103           DCLOO
11
12                ;AR17:
13 17132 006100           DCSET             ;MISC ADD SWAPPED TEST.
14 17133 006101           DCRAN
15 17134 131000           MOV 1,2
16 17135 113000           ADD 0,2
17 17136 107300           ADDS 0,1
18 17137 125300           MOVS 1,1
19 17140 132414           SUB# 1,2,SZR
20 17141 006102           DCERR
21 17142 006103           DCLOO
22
23                ;AR18:
24 17143 006100           DCSET             ;CHECK ADC LEFT.
25 17144 006101           DCRAN
26 17145 131000           MOV 1,2
27 17146 112120           ADCZL 0,2
28 17147 100000           COM 0,0
29 17150 107120           ADDZL 0,1
30 17151 132414           SUB# 1,2,SZR
31 17152 006102           DCERR
32 17153 006103           DCLOO
```

```
10254 N3MRT
01                ;AR19:
02 17154 006100           DCSET             ;TEST ADC RIGHT.
03 17155 006101           DCRAN
04 17156 131000           MOV 1,2
05 17157 112220           ADCZR 0,2
06 17160 100000           COM 0,0
07 17161 107220           ADDZR 0,1
08 17162 132414           SUB# 1,2,SZR
09 17163 006102           DCERR
10 17164 006103           DCLOO
11
12                ;AR20:
13 17165 006100           DCSET             ;TEST SUB RIGHT.
14 17166 006101           DCRAN
15 17167 131000           MOV 1,2
16 17170 106620           SUBZR 0,1
17 17171 100420           NEGZ 0,0
18 17172 113200           ADDR 0,2
19 17173 132414           SUB# 1,2,SZR
20 17174 006102           DCERR
21 17175 006103           DCLOO
22
23
24 17176 006100 DCJSR:    DCSET             ;TEST THAT INDEX WITH
25 17177 004401           JSR .+1           ;SIGN BIT SET DOES
26 17200 171122           MOVZL 3,2,SZC     ;NOT LOAD INTO PC ON JSR.
27 17201 006102           DCERR
28 17202 151240           MOVOR 2,2
29 17203 005004           JSR 4,2           ;GO TO NEXT LOCATION
30 17204 004401           JSR .+1
31 17205 165000           MOV 3,1           ;JSR SHOULD NEVER
32 17206 125112           MOVL# 1,1,SZC     ;STORE THE SIGN BIT.
33 17207 006102           DCERR
34 17210 006103           DCLOO
35
36 17211 006105           DCRET             ;END OF TEST
37 17212 000000 DCEND: 0
```

```
!0255 N3MRT
01                      ;       .TITL NVDSK
02                      ;MULTIPLE NOVA DISK TEST
03                      ;THE DIAGNOSTIC LINKER
04
05          000020 NDDSK=20
06                      ;DEFINITIONS BLOCK TO LINKER
07                              NEXTT ND.00
08          017213 LMEML=.
09          000165              .LOC LPG0
10 00165 017216                 ND.00
11          000166 LPG0=.
12          017213              .LOC LMEML
13 17213 000000                 0       ;TEST PASS CTR
14 17214 000000                 0       ;TEST ERROR CTR
15 17215 000000                 0       ;INTERRUPT TIMEOUT SWITCH
16 17216 017262 ND.00:          ND.01
17 17217 017351                 ND.02   *
18 17220 000000                 0
19 17221 000000                 0
20 17222 177777                 -1
21 17223 176000                 176000
22 17224 020037                 ND.XI
23 17225 020037                 ND.XI
24          017226              .TXTE !
25 17226 147516 NOVA DISK(PRI)!
26          040526
27          042240
28          051711
29          024113
30          151120
31          124711
32          000000
33
34
35                      ;REPORT TO OPERATOR HIGHEST DISK ADRS FOUND
36 17236 017237 ND.1R:  .+1
37 17237 020416         LDA 0,ND3TX
38                      LCALL ERRTX
39 17240 100170         ERRTX-ASCRA*1B11+100010
40 17241 024475         LDA 1,ND.ND     ;HIGH DISK ADRS
41                      LCALL PZOCT     ;PRINT IT
42 17242 100130         PZOCT-ASCRA*1B11+100010
43 17243 020413         LDA 0,ND3TX+1
44 17244 040753         STA 0,ND.00+1
45 17245 030412         LDA 2,ND3TX+2
46 17246 024473         LDA 1,NDMK1
47 17247 102400         SUB 0,0
48 17250 041000         STA 0,0,2
49 17251 151400         INC 2,2
50 17252 125404         INC 1,1,SZR
51 17253 000775         JMP .-3         ;CLEAR DISK CONTROL TABLE
52 17254 000475         JMP ND.02
53 17255 020161 ND3TX:  NDTX3
54 17256 017351         ND.02
55 17257 020106         NDCTB
```

```
!0256 N3MRT
01                      ;DETERMINE IF SYSTEM HAS A FIXED HEAD DISK
02                      ;DISABLE TEST ENTER IF NONE EXISTS
03 17260 001000                 186
04 17261 000004                 1B13            ;NON-EXIST BIT
05 17262 062677 ND.01:  IORST
06 17263 054462         STA 3,ND.S3
07 17264 102220         ADCZR 0,0
08 17265 062020         DOB 0,NDDSK
09 17266 065420         DIB 1,NDDSK
10 17267 122404         SUB 1,0,SZR     ;SKP= DSK CONTROL
11 17270 000424         JMP ND.1E-3     ;NO NOVA DISK CONTROL
12                      ;NOW DETERMINE MAXIMUM AVAILABLE DISK ADRS
13 17271 034767         LDA 3,ND.01-2   ;TO COUNT DOWN ADRS
14 17272 020445         LDA 0,NDMAX
15 17273 026454 ND.1L:  LDA 1,@NDETB
16 17274 066020         DOB 1,NDDSK     ;SET CA=BUFFER ADRS
17 17275 061120         DOAS 0,NDDSK    ;TRY TO READ
18 17276 126400         SUB 1,1
19 17277 060000         NIO 0
20 17300 125405         INC 1,1,SNR     ;COUNT A WHILE
21 17301 000404         JMP .+4
22 17302 063520         SKPBZ NDDSK     ;SKP=READ DONE
23 17303 000774         JMP .-4
24 17304 101001         MOV 0,0,SKP     ;SKP FIRST "C" PULSE
25 17305 060220         NIOC NDDSK
26 17306 064620         DIAC 1,NDDSK    ;GET STATUS BITS
27 17307 030752         LDA 2,ND.01-1
28 17310 147405         AND 2,1,SNR     ;1B12=NONEXISTENT
29 17311 000406         JMP ND.1E       ;THIS ONE EXISTS
30 17312 162422         SUBZ 3,0,SZC    ;HALVE DISK ADRS
31 17313 000760         JMP ND.1L       ;TRY AGAIN
32 17314 102000         ADC 0,0         ;NO DISKS EXIST
33 17315 040703         STA 0,ND.00+2
34 17316 002427         JMP @ND.S3
35 17317 040417 ND.1E:  STA 0,ND.ND     ;HIGH DISK ADRS
36 17320 101220         MOVZR 0,0
37 17321 101220         MOVZR 0,0
38 17322 101220         MOVZR 0,0
39 17323 101220         MOVZR 0,0       ;DIV BY 16 SEECTORS
40 17324 101400         INC 0,0
41 17325 040413         STA 0,NDSUR     ;=#AVAIL FOR  RAN. SELECT
42 17326 062677         IORST
43 17327 020707         LDA 0,ND.1R
44 17330 040667         STA 0,ND.00+1   ;SO SIZE GETS TYPED
45 17331 020411         LDA 0,ND.K1
46 17332 024411         LDA 1,ND.K2
47 17333 030411         LDA 2,ND.K3
48 17334 006101         JSR @EINTS      ;ENTER INTA SERVICE PARAMS
49 17335 002410         JMP @ND.S3
50 17336 000000 ND.ND:  0       ;# OF DISKS
51 17337 017777 NDMAX:  17777   ;MAX POSSIBLE DISK
52 17340 000000 NDSUR:  0       ;# SECTORS/16
53 17341 177753 NDMK1: -21.
54 17342 000020 ND.K1: NDDSK
55 17343 000177 ND.K2: 177
56 17344 017703 ND.K3: ND.IS
57 17345 000000 ND.S3: 0
58 17346 020106 NDTBA: NDCTB
59 17347 001443 NDETB:  LSETB
```

```
!0257 N3MRT
01                ;EXECUTE PORTION OF DISK TEST
02 17350 017424           ND.KK
03 17351 030647 ND.02:    LDA 2,ND.00+2
04 17352 151004           MOV 2,2,SZR
05 17353 001000           JMP 0,2
06                        LCALL ASCRA       ;GET 1K SCR
07 17354 100010           ASCRA-ASCRA*1B11+100010
08 17355 002446           JMP @ND.XX        ;NONE AVAIL
09                        LCALL ADMAP
10 17356 100550           ADMAP-ASCRA*1B11+100010
11 17357 002444           JMP @ND.XX
12                        LCALL ARANG
13 17360 100270           ARANG-ASCRA*1B11+100010
14 17361 105000           MOV 0,1
15 17362 152520           SUBZL 2,2
16 17363 151140           MOVOL 2,2         ;=3
17 17364 102400           SUB 0,0           .
18 17365 042763           STA 0,@ND.02-1
19                        LCALL ADIVI
20 17366 100310           ADIVI-ASCRA*1B11+100010
21 17367 100405           NEG 0,0,SNR
22 17370 000410           JMP ND.2C
23                ND.2L:  LCALL ESCRA
24 17371 100030           ESCRA-ASCRA*1B11+100010
25 17372 000406           JMP ND.2C         ;CANT EXPAND MORE
26                        LCALL EDMAP       ;ASSIGN TO DCH
27 17373 100570           EDMAP-ASCRA*1B11+100010
28 17374 000424           JMP ND.2R         ;CANT ASSIGN MORE
29 17375 012753           ISZ @ND.02-1      ;#1K'S +1
30 17376 101404           INC 0,0,SZR
31 17377 000772           JMP ND.2L         ;TRY 1K MORE TO ASSIGN
```

```
!0258 N3MRT
01                ;SELECT A RANDOM BUFFER START WITHIN FIRST 256 OF
02                ;THE ASSIGNED SCRATCH AREA
03                ND.2C:  LCALL ARANG
04 17400 100270           ARANG-ASCRA*1B11+100010
05 17401 030424           LDA 2,ND252
06 17402 105000           MOV 0,1
07                        LCALL ADIVI
08 17403 100310           ADIVI-ASCRA*1B11+100010
09 17404 024152           LDA 1,SCRLO
10 17405 107000           ADD 0,1
11 17406 046421           STA 1,@ND16+1     ;DATA START ADDRS
12 17407 024154           LDA 1,DCHLO
13 17410 107000           ADD 0,1
14 17411 046417           STA 1,@ND16+2     ;CHANNEL START ADDRS
15 17412 024412           LDA 1,ND.KK       ;#1KS
16 17413 127120           ADDZL 1,1         ;*4
17 17414 125620           INCZR 1,1
18 17415 125500           INCL 1,1          ;+3 FOR 1ST 1K
19 17416 046413           STA 1,@ND16+3     ;# BLOKS/SECTORS AVAIL
20 17417 000507           JMP ND.03         ;FIGURE OU WHAT TO DO
21                ND.2R:  LCALL RSCRA       ;RELEASE 1K SCR
22 17420 100050           RSCRA-ASCRA*1B11+100010
23 17421 063077           HALT              ;NOT PROBABLE HALT
24 17422 000756           JMP ND.2C
25 17423 020037 ND.XX:    ND.XI
26 17424 000000 ND.KK:    0
27 17425 000374 ND252:    252.
28 17426 000020 ND16:     16.
29 17427 017621           NDDST
30 17430 017622           NDCST
31 17431 017613           ND.BK
32 17432 017625           NDSEC
```

```
10259 N3MRT
01                ;NOW SELECT SECTOR TO EXERCISE
02 17433 017632           ND.WK
03                ND.3S:  LCALL ARANG
04 17434 100270           ARANG-ASCRA*1B11+100010
05 17435 105000           MOV 0,1
06 17436 030770           LDA 2,ND16        ;16 SECTORS
07 17437 022772           LDA 0,@ND16+3
08 17440 112400           SUB 0,2           ;-# TO WRITE
09                        LCALL ADIVI       ;(0) WILL=SECTOR START
10 17441 100310           ADIVI-ASCRA*1B11+100010
11 17442 024551           LDA 1,ND.BK
12 17443 125300           MOVS 1,1
13 17444 123000           ADD 1,0           ;CREATE SEC CONTROL WD
14 17445 042765           STA 0,@ND16+4     ;START SECTOR AND #
15 17446 122400           SUB 1,0           ;LEAVE STRT SECT. #
16 17447 124600           NEGR 1,1
17 17450 125400           INC 1,1           .
18 17451 125500           INCL 1,1
19 17452 046761           STA 1,@ND.3S-1    ;# TIMES TO REGEN 4 WDS
20                ;CREATE DOA FOR THIS SET SECTORS
21 17453 030551           LDA 2,NDCYL
22 17454 151120           MOVZL 2,2         ;*2
23 17455 153120           ADDZL 2,2         ;*8
24 17456 151100           MOVL 2,2          ;*16 FOR SEECTOR
25 17457 113000           ADD 0,2
26 17460 050543           STA 2,NDDRV       ;FOR DOA'S
```

```
10260 N3MRT
01                ;NOW GET RANDOM DATA WORDS
02                        LCALL FRANG
03 17461 100450           FRANG-ASCRA*1B11+100010
04 17462 044544           STA 1,NDDW1       ;NEW RAN DATA WDS
05 17463 050544           STA 2,NDDW2
06 17464 040544           STA 0,NDDW3
07 17465 034552           LDA 3,NDIDX
08 17466 045403           STA 1,3,3         ;SAVE CONTROL
09 17467 051404           STA 2,4,3         ;WORDS NEWLY
10 17470 041405           STA 0,5,3
11 17471 171000           MOV 3,2
12                        LCALL ARANG
13 17472 100270           ARANG-ASCRA*1B11+100010
14 17473 040536           STA 0,NDDW4
15 17474 041006           STA 0,6,2
16 17475 020527           LDA 0,NDCYL       ;GENERATED FOR
17 17476 041000           STA 0,0,2         ;THIS OP TABLE
18 17477 020526           LDA 0,NDSEC
19 17500 041001           STA 0,1,2
20 17501 020522           LDA 0,NDDRV
21 17502 041002           STA 0,2,2
22                ;GENERATE DATA PATTERN INTO SCRATCH AREA
23
24 17503 030516 ND.3G:    LDA 2,NDDST
25 17504 020522           LDA 0,NDDW1       ;GET 2 DATA WORDS
26 17505 024522           LDA 1,NDDW2       ;TO FIRST 2 WORDS
27 17506 041000           STA 0,0,2         ;IN THE BUFFER
28 17507 045001           STA 1,1,2
29 17510 020520           LDA 0,NDDW3
30 17511 024520           LDA 1,NDDW4
31 17512 041002           STA 0,2,2
32 17513 045003           STA 1,3,2         ;1ST 4 WORDS RANDOM
33 17514 024516           LDA 1,ND.WK       ;GET # WORDS-4
34 17515 021000           LDA 0,0,2
35 17516 041004           STA 0,4,2
36 17517 151400           INC 2,2
37 17520 125404           INC 1,1,SZR
38 17521 000774           JMP .-4
39 17522 024402           LDA 1,ND.WC       ;WRITE COMMAND
40 17523 000516           JMP ND.05         ;START SEEK AND WRITE
41 17524 177400 ND.WC:    -400
```

```
10261 N3MRT
01                ;DATA CHANNEL AND SCRATCH ARE ASSIGNED
02                ;SELECT ONE OF THE THREE OP TABLES
03                ;AND PERFORM THE NEXT OPERATION REQ
04 17525 000007           7
05                ND.03:  LCALL ARANG
06 17526 100270           ARANG-ASCRA*1B11+100010
07 17527 101300           MOVS 0,0
08 17530 126520           SUBZL 1,1
09 17531 125140           MOVOL 1,1         ;AC1=3
10 17532 101220           MOVZR 0,0
11 17533 107627           ANDZR 0,1,SBN     ;3 RESULT ILLEGAL
12 17534 125101           MOVL 1,1,SKP
13 17535 000772           JMP .-6           ;TRY NEXT 2 BITS
14 17536 102400           SUB 0,0
15 17537 125004           MOV 1,1,SZR
16 17540 020765           LDA 0,ND.03-1
17 17541 125224           MOVZR 1,1,SZR     .
18 17542 103000           ADD 0,0
19 17543 030455           LDA 2,NDTBL
20 17544 113000           ADD 0,2
21 17545 025000           LDA 1,0,2
22 17546 044456           STA 1,NDCYL       ;GET CURRENT CYLINDER
23 17547 021001           LDA 0,1,2
24 17550 040455           STA 0,NDSEC       ;SECTOR AND # SECTORS
25 17551 021002           LDA 0,2,2         ;DRIVE AND SURFACE
26 17552 040451           STA 0,NDDRV
27 17553 021003           LDA 0,3,2
28 17554 040452           STA 0,NDDW1       ;DATA WORD 1
29 17555 021004           LDA 0,4,2
30 17556 040451           STA 0,NDDW2       ;DATA WORD 2
31 17557 021005           LDA 0,5,2         ;DATA WORD 3
32 17560 040450           STA 0,NDDW3
33 17561 021006           LDA 0,6,2         ;DATA WORD 4
34 17562 040447           STA 0,NDDW4
35 17563 050454           STA 2,NDIDX       ;SAVE ADDRS CONTOL WORD
36 17564 125004           MOV 1,1,SZR       ;SKIP NO CYL SEL
37 17565 000453           JMP ND.04         ;CYL ENABLE READ IT
```

```
10262 N3MRT
01                ND.3L:  LCALL ARANG       ;RANDOM SEL CYL
02 17566 100270           ARANG-ASCRA*1B11+100010
03 17567 105000           MOV 0,1
04 17570 032422           LDA 2,@ND.BK-1    ;CYLINDER SELECTED
05                        LCALL ADIVI       ;MUST NOT BE
06 17571 100310           ADIVI-ASCRA*1B11+100010
07 17572 101231           MOVZR# 0,0,SKP    ;SKP=USE ALL # SZC=EVEN SNC=ODD
08 17573 000773           JMP ND.3L         ;DUAL. PROC. DELETE CYL.
09 17574 103240           ADDOR 0,0         ;1B0 SO #0 IS VALID
10 17575 034423           LDA 3,NDTBL       ;ALREADY IN USE
11 17576 031400           LDA 2,0,3         ;AND CYL 0
12 17577 142415           SUB # 2,0,SNR     ;IS NOT EXERCISED
13 17600 000766           JMP ND.3L
14 17601 031407           LDA 2,7,3
15 17602 142415           SUB# 2,0,SNR
16 17603 000763           JMP ND.3L
17 17604 031416           LDA 2,16,3
18 17605 142415           SUB# 2,0,SNR
19 17606 000760           JMP ND.3L
20 17607 040415           STA 0,NDCYL       ;CYL SEL OK TO USE
21 17610 002401           JMP @.+1
22 17611 017434           ND.3S
23 17612 017340           NDSUR
24 17613 000000 ND.BK:    0
25
26
27 17614 000037 ND37: 37
28 17615 000014 ND12: 12.
29 17616 000020 ND20: 20
30 17617 000017 ND17: 17
31 17620 020106 NDTBL:    NDCTB
32 17621 000000 NDDST:    0
33 17622 000000 NDCST:    0
34 17623 000000 NDDRV:    0
35 17624 000000 NDCYL:    0
36 17625 000000 NDSEC:    0
37 17626 000000 NDDW1:    0
38 17627 000000 NDDW2:    0
39 17630 000000 NDDW3:    0
40 17631 000000 NDDW4:    0
41 17632 000000 ND.WK:    0
42 17633 001000 ND.SC:    1000
43 17634 000000 ND.CO:    0                 ;COMMAND + CYL #
44 17635 000000 ND.OC:    0                 ;SECT SIDE-SECK
45 17636 000000 NDSTA:    0                 ;STATUS READ
46 17637 000000 NDIDX:    0
```

```
!0263 N3MRT
01                ;AC1=READ OR WRITE COMMAND
02                ;CALCULATE DISK CONTROL WORDS AND
03                ;START SEEK ND.04 IS READ ENTER
04
05 17640 126001 ND.04:   ADC 1,1,SKP      ;-1 TO STRTT READ
06 17641 126400 ND.05:   SUB 1,1          ;0 TO STRT WRITE
07 17642 044772          STA 1,ND.CO      ;READ OR WRITE
08 17643 024762          LDA 1,NDSEC
09 17644 020750          LDA 0,ND37
10 17645 123400          AND 1,0
11 17646 106700          SUBS 0,1
12 17647 030744 ND.5S:   LDA 2,ND.BK      ;(2)=# 256 WD BLK AVAIL
13 17650 146432          SUBZ# 2,1,SZC
14 17651 145000          MOV 2,1          ;AC1=# SECTORS
15 17652 044763          STA 1,ND.OC      ;FOR INTA COUNTING
16 17653 131300          MOVS 1,2         ;TO EXERCISE
17 17654 150600          NEGR 2,2         *
18 17655 151400          INC 2,2
19 17656 151500          INCL 2,2
20 17657 050753          STA 2,ND.WK      ;BUFFER L - 4 WORDS
21 17660 020454 ND.5G:   LDA 0,NDRDI      ;READ INTA HNDLER
22 17661 042416          STA 0,@ND.0X
23 17662 024741          LDA 1,NDDRV
24 17663 044415          STA 1,NDDNX      ;INCR. BY INTA'S
25 17664 030736          LDA 2,NDCST      ;CHANL STRT ADRS
26 17665 050414          STA 2,NDCNX      ;FOR NEXT INTA
27 17666 072020          DOB 2,NDDSK      ;AND CONTROL
28 17667 010745          ISZ ND.CO        ;SKP=STRT READ
29 17670 000403          JMP ND.5W        ;DO WRITE INSTEAD
30 17671 065120          DOAS 1,NDDSK
31                       LCALL RETRN      ;START SEEK EXIT
32 17672 100210          RETRN=ASCRA*1B11+100010
33 17673 020446 ND.5W:   LDA 0,NDWRI
34 17674 042403          STA 0,@ND.0X
35 17675 065320          DOAP 1,NDDSK
36                       LCALL RETRN
37 17676 100210          RETRN=ASCRA*1B11+100010
38 17677 017220 ND.0X:   ND.00+2
39 17700 000000 NDDNX:   0        ;NEXT SECTOR
40 17701 000000 NDCNX:   0        ;NEXT CHANNEL STRT
```

```
!0264 N3MRT
01                ;DISK INTERRUPT SERVICE DISPATCHER
02 17702 000020          NDDSK
03 17703 060420 ND.IS:   DIA 0,NDDSK
04 17704 040732          STA 0,NDSTA
05 17705 032772          LDA 2,@ND.0X
06 17706 151132          MOVZL# 2,2,SZC   ;SKP=UNEXPECTED INTRP
07 17707 000405          JMP .+5
08 17710 020772          LDA 0,ND.IS-1
09 17711 040144          STA 0,UDEVI
10 17712 060220          NIOC NDDSK
11 17713 001400          JMP 0,3
12 17714 101200          MOVR 0,0         ;SAVE ERR IN CRY
13 17715 020763          LDA 0,NDDNX
14 17716 101400          INC 0,0
15 17717 024762          LDA 1,NDCNX
16 17720 125300          MOVS 1,1
17 17721 125700          INCS 1,1         ;+400 NEXT CA
18 17722 151002          MOV 2,2,SZC      ;SKP=ERR STATUS
19 17723 000407          JMP ND.SE
20 17724 014711          DSZ ND.OC        ;SKP=DONE ALL
21 17725 001000          JMP 0,2
22 17726 020417          LDA 0,NDCOM
23 17727 060220          NIOC NDDSK
24 17730 042747          STA 0,@ND.0X     ;VERIFY DATA RETURN
25 17731 001400          JMP 0,3          ;DISMISS INTA
26 17732 020540 ND.SE:   LDA 0,NDSTE
27 17733 000774          JMP ND.SE-3      ;STATUS ERR EXIT
28                ;READ INTA ;RESTART
29 17734 117735 NDRDI:   @.+1
30 17735 061120          DOAS 0,NDDSK
31 17736 040742          STA 0,NDDNX
32 17737 044742          STA 1,NDCNX
33 17740 001400          JMP 0,3
34                ;WRITE INTA HANDLER RESTART
35 17741 117742 NDWRI:   @.+1
36 17742 066020          DOB 1,NDDSK
37 17743 061320          DOAP 0,NDDSK
38 17744 000772          JMP NDRDI+2      ;DISMISS INTA
```

```
!0265 N3MRT
01                      ;READ OR WRITE OK NO STAT ERRS
02                      ;COMPARE DATA IN BUFFER
03                      ;AGAINST WHAT'S SUPPOSED TO BE THERE
04 17745 017746 NDCOM:  .+1
05 17746 176400         SUB 3,3
06 17747 030652         LDA 2,NDDST     ;START ADRS
07 17750 020656         LDA 0,NDDW1
08 17751 025000         LDA 1,0,2       ;COMPARE
09 17752 122414         SUB# 1,0,SZR    ;FIRST 4 WORDS
10 17753 000475         JMP NDERR       ;AGAINST EVERY
11 17754 175400         INC 3,3         ;OTHER IN BUFFER
12 17755 020652         LDA 0,NDDW2
13 17756 025001         LDA 1,1,2
14 17757 122414         SUB# 1,0,SZR
15 17760 000470         JMP NDERR
16 17761 020647         LDA 0,NDDW3
17 17762 025002         LDA 1,2,2       .
18 17763 175400         INC 3,3
19 17764 106414         SUB# 0,1,SZR
20 17765 000463         JMP NDERR
21 17766 175400         INC 3,3
22 17767 020642         LDA 0,NDDW4
23 17770 025003         LDA 1,3,2
24 17771 106414         SUB# 0,1,SZR
25 17772 000456         JMP NDERR
26 17773 034637         LDA 3,ND.WK     ;#WORDS - 4
27 17774 021000         LDA 0,0,2
28 17775 025004         LDA 1,4,2
29 17776 122414         SUB# 1,0,SZR
30 17777 000451         JMP NDERR
31 20000 055000         STA 3,0,2       ;CLR BUFFER NXT RD
32 20001 151400         INC 2,2         ;BUMP ADRS
33 20002 175404         INC 3,3,SZR     ;SKP DONEALL
34 20003 000771         JMP .-7
35                      LCALL ARANG     ;BITS 1 AND 15=1
36 20004 100270         ARANG-ASCRA*1B11+100010
37 20005 101232         MOVZR# 0,0,SZC  ;RELEASE THIS
38 20006 103123         ADDZL 0,0,SNC   ;OP TABLE
39 20007 000403         JMP .+3         ;EITHER = 0 KEEP IT
40 20010 102400 ND.X0:  SUB 0,0                 '
41 20011 042626         STA 0,@NDIDX
42                      LCALL ARANG
43 20012 100270         ARANG-ASCRA*1B11+100010
44 20013 101102         MOVL 0,0,SZC    ;NEXT 2 BITS =1?
45 20014 101103         MOVL 0,0,SNC    ;IS RELEASE THIS BUFFER
46 20015 002432         JMP @ND.SX      ;SELECT NEW STRT BUFFER
47 20016 022417 ND.X3:  LDA 0,@NDHIK
48 20017 024136         LDA 1,MPSWT
49 20020 125005         MOV 1,1,SNR
50 20021 000416         JMP ND.XI
51 20022 024414         LDA 1,ND.37
52 20023 106032         ADCZ# 0,1,SZC   ;SKP IF HIGHK >= 32K
53 20024 000413         JMP ND.XI
54                ND.X4: LCALL RDMAP
55 20025 100610         RDMAP-ASCRA*1B11+100010
56 20026 101001         MOV 0,0,SKP
57 20027 000776         JMP ND.X4
58                ND.X5: LCALL RSCRA
59 20030 100050         RSCRA-ASCRA*1B11+100010
60 20031 101001         MOV 0,0,SKP
```

```
 0266 N3MRT
01 20032 000776         JMP ND.X5
02 20033 002401         JMP @.+1
03 20034 017354         ND.02+3
04 20035 001502 NDHIK:  HIGHK
05 20036 000037 ND.37:  37
```

```
!0267 N3MRT
01                ND.XI:  LCALL RDMAP
02 20037 100610           RDMAP-ASCRA*1B11+100010
03 20040 101001           MOV 0,0,SKP
04 20041 000776           JMP ND.XI
05                ND.X2:  LCALL RSCRA
06 20042 100050           RSCRA-ASCRA*1B11+100010
07 20043 102401           SUB 0,0,SKP
08 20044 000776           JMP ND.X2
09 20045 042632           STA 0,@ND.0X
10                        LCALL RETRN
11 20046 100210           RETRN-ASCRA*1B11+100010
12 20047 017400 ND.SX: ND.2C
13                ;ERROR IN DATA COMPARE
14 20050 054430 NDERR:    STA 3,ND.SA
15                        LCALL ERROI
16 20051 100350           ERROI-ASCRA*1B11+100010
17 20052 000401           JMP .+1
18 20053 020460           LDA 0,NDTX1
19                        LCALL ERRTX
20 20054 100170           ERRTX-ASCRA*1B11+100010
21 20055 020423           LDA 0,ND.SA
22 20056 026425           LDA 1,@NDDSX
23 20057 032425           LDA 2,@NDCSX
24                        LCALL ERROC
25 20060 100370           ERROC-ASCRA*1B11+100010
26 20061 000401           JMP .+1
27 20062 020464           LDA 0,NDTX2
28                        LCALL ERRTX
29 20063 100170           ERRTX-ASCRA*1B11+100010
30 20064 022421           LDA 0,@NDSTX
31 20065 026414           LDA 1,@ND.CX
32 20066 032414           LDA 2,@ND.0X
33                        LCALL ERROC
34 20067 100370           ERROC-ASCRA*1B11+100010
35 20070 000720           JMP ND.X0       ;RELEASE IMMED
36 20071 000725           JMP ND.X3
37                ;NON REC DISK STATUS ERROR
38 20072 020073 NDSTE:    .+1
39 20073 061420           DIB 0,NDDSK
40 20074 066420           DIC 1,NDDSK
41 20075 032410           LDA 2,@NDSTX
42 20076 036407           LDA 3,@NDSTX
43 20077 000751           JMP NDERR
44 20100 000000 ND.SA: 0
45 20101 017623 ND.CX: NDDRV
46 20102 017634 ND.OX: ND.CO
47 20103 017621 NDDSX: NDDST
48 20104 017622 NDCSX: NDCST
49 20105 017636 NDSTX: NDSTA
50                ;3/7 WORD OPERATION TABLES
51 20106 000025 NDCT8:    .BLK 21.
52 20133 020134 NDTX1: .+1
53 20134 005215           .TXTE (<15><12>ND.SA    NDDST   NDCST(
54 20146 020147 NDTX2: .+1
55 20147 005215           .TXTE (<15><12>NDSTA    NDADR   ND.CO(
56 20161 005215 NDTX3:    .TXTE   (<15><12>DEV# 20 HIGH DISK ADDRESS= (
```

```
!0268 N3MRT
01                ;       .TITL MVDSK
02                ;MOVABLE HEAD DISK TEST COMPATABLE WITH
03                ;THE DIAGNOSTIC LINKER
04
05        000033 MHDSK=33
06                ;DEFINITIONS BLOCK TO LINKER
07                        NEXTT MH.00
08        020200 LMEML=.
09        000166          .LOC LPG0
10 00166 020203           MH.00
11        000167 LPG0=.
12        020200          .LOC LMEML
13 20200 000000           0       ;TEST PASS CTR
14 20201 000000           0       ;TEST ERROR CTR
15 20202 000000           0       ;INTERRUPT TIMEOUT SWITCH
16 20203 020231 MH.00:            MH.01
17 20204 020341                   MH.02
18 20205 000000                   0
19 20206 000000                   0
20 20207 177777                   -1
21 20210 176000                   176000
22 20211 021665                   MH.XI
23 20212 021665                   MH.XI
24        020213                  .TXTE 1
25 20213 147515 MOVING HEAD DISK(PRI)1
26        144526
27        043516
28        044240
29        040705
30        120104
31        144504
32        045523
33        050050
34        144722
35        000251
36
```

```
!0269 N3MRT
01                ;EITHER NO DISK CONTROL,ADAPTER IS OFF
02                ;OR THERE ARE NO DISKS ON LINE
03 20226 102000 MHNCT:  ADC 0,0
04 20227 040756         STA 0,MH.00+2
05 20230 001400         JMP 0,3         ;TEST DELETED
06                ;DETERMINE IF SYSTEM HAS A MOVING HEAD DISK
07                ;DISABLE TEST ENTER IF NONE EXISTS
08 20231 102000 MH.01:  ADC 0,0
09 20232 062033         DOB 0,MHDSK     ;SEE IF C.A. REG
10 20233 065433         DIB 1,MHDSK
11 20234 101220         MOVZR 0,0
12 20235 122415         SUB# 1,0,SNR    ;CHECK BITS 1-15
13 20236 000404         JMP .+4
14 20237 124014         COM# 1,1,SZR    ;SKP IS CONTROLLLER EXISTS
15 20240 000766         JMP MHNCT       ;NO DISK CONTROL
16 20241 000402         JMP .+2         ;CAN USE B DCH MAP
17 20242 126400         SUB 1,1         ;CAN'T USE B DCH MAP
18 20243 044470         STA 1,MH.BF     ;SET FLAG FOR B DCH MAP
19 20244 102400         SUB 0,0
20 20245 024460         LDA 1,MHMK1
21 20246 030464         LDA 2,MHTBA     ;CLR ACTIVE TABLS
22 20247 041000         STA 0,0,2
23 20250 151400         INC 2,2
24 20251 125404         INC 1,1,SZR
25 20252 000775         JMP .-3
26 20253 062033         DOB 0,MHDSK
27                ;DETERMINE HOW MANY DISKS ON SYSTEM
28 20254 102400         SUB 0,0
29 20255 105240         MOVOR 0,1
30 20256 125220         MOVZR 1,1       ;TO INCR. DISK#
31 20257 040550         STA 0,MH.ND     ;0 # DISKS
32 20260 030454         LDA 2,MHTXA     ;TO SET EXIST SWS
33 20261 050020         STA 2,20        ;BY DOUBLE@
34 20262 130700 MH.1L:  NEGS 1,2
35 20263 151424         INCZ 2,2,SZR
36 20264 000777         JMP .-1
37 20265 063333         DOCP 0,MHDSK    ;SELECT CURRENT DISK
38 20266 151103         MOVL 2,2,SNC    ;STALL AWHILE
39 20267 000777         JMP .-1
40 20270 070633         DIAC 2,MHDSK    ;GET STATUS ,
41 20271 153300         ADDS 2,2        ;1B9=DSK RDY
42 20272 151103         MOVL 2,2,SNC    ;SKP=1B9
43 20273 000402         JMP .+2         ;THIS DISK NONEXIS
44 20274 010533         ISZ MH.ND       ;+1#DISKS
45 20275 152402         SUB 2,2,SZC     ;SKP=DRIVE EXISTS
46 20276 152000         ADC 2,2         ;SET NO DRIVE SWITCH
47 20277 052020         STA 2,@20       ;INTO DESCRIPTOR TABLE
48 20300 123023         ADDZ 1,0,SNC    ;SKP=4 DSKS
49 20301 000761         JMP MH.1L       ;NOT DONE ALL
```

```
!0270 N3MRT
01                ;STALL FOR AWHILE, THEN DETERMINE THE
02                ;CHARACTERISTICS OF EACH INDIVIDUAL DISK
03                ;BUILDING A DISK DESCRIPTOR TABLE
04 20302 062677         IORST
05 20303 130700         NEGS 1,2
06 20304 151404         INC 2,2,SZR
07 20305 000777         JMP .-1
08 20306 052431         STA 2,@MHTXB+2
09 20307 052427         STA 2,@MHTXB+1
10 20310 020414         LDA 0,MHMK1-1
11 20311 040673         STA 0,MH.00+1   ;SO DESCRIPTION IS TYPED
12 20312 020515         LDA 0,MH.ND     ;=# OF DISKS ON LINE
13 20313 101005         MOV 0,0,SNR     ;SKP= AT LEAST ONE
14 20314 000712         JMP MHNCT       ;DID NOT FIND ANY
15 20315 020411         LDA 0,MH.K1
16 20316 024411         LDA 1,MH.K2     ;INT MASK BITS
17 20317 030411         LDA 2,MH.K3     ;INT SERV DIRECTOR
18 20320 054411         STA 3,MH.S3
19 20321 006101         JSR @EINTS
20 20322 002401         JMP @.+1
21 20323 021223         MHFSB           ;FIND SEEK BIT
22 20324 021474         MH.1R           ;ADRS TO TYPE DESCRIP.
23 20325 177574 MHMK1:  MHDP0-MHDRE
24 20326 000033 MH.K1:  MHDSK
25 20327 000777 MH.K2:  777
26 20330 021020 MH.K3:  MH.IS
27 20331 000000 MH.S3:  0
28 20332 022145 MHTBA:  MHDP0
29 20333 000000 MH.BF:   0              ;-1 IF CAN USE "B" DCH MAP
30 20334 121555 MHTXA:  @MHDPT
31 20335 000102 MHTXB:  .TXTE (B(
32 20336 021303         MHDRX
33 20337 021304         MHDRX+1
```

```
!0271 N3MRT
01                ;EXECUTE PORTION OF DISK TEST
02 20340 020421           MH.KK
03 20341 030644 MH.02:    LDA 2,MH.00+2
04 20342 151004           MOV 2,2,SZR
05 20343 001000           JMP 0,2
06                        LCALL ASCRA       ;GET 1K SCR
07 20344 100010           ASCRA-ASCRA*1B11+100010
08 20345 002453           JMP @MH.XX        ;NONE AVAIL
09 20346 020765           LDA 0,MH.BF       ;GET B FLG
10 20347 100004           COM 0,0,SZR       ;SKP IS USE "B" MAP FEATURE
11 20350 000403           JMP .+3           ;CAN'T USE IT
12                        LCALL RNDMP       ;SELECT A/B MAP
13 20351 100530           RNDMP-ASCRA*1B11+100010
14 20352 101000           MOV 0,0
15                        LCALL ADMAP
16 20353 100550           ADMAP-ASCRA*1B11+100010
17 20354 002444           JMP @MH.XX         .
18                        LCALL ARANG
19 20355 100270           ARANG-ASCRA*1B11+100010
20 20356 105000           MOV 0,1
21 20357 152520           SUBZL 2,2
22 20360 153100           ADDL 2,2          ;=4
23 20361 102400           SUB 0,0
24 20362 042756           STA 0,@MH.02-1
25                        LCALL ADIVI
26 20363 100310           ADIVI-ASCRA*1B11+100010
27 20364 100405           NEG 0,0,SNR
28 20365 000410           JMP MH.2C
29                MH.2L:  LCALL ESCRA
30 20366 100030           ESCRA-ASCRA*1B11+100010
31 20367 000406           JMP MH.2C         ;CANT EXPAND MORE
32                        LCALL EDMAP       ;ASSIGN TO DCH
33 20370 100570           EDMAP-ASCRA*1B11+100010
34 20371 000424           JMP MH.2R         ;CANT ASSIGN MORE
35 20372 012746           ISZ @MH.02-1      ;#1K'S +1
36 20373 101404           INC 0,0,SZR
37 20374 000772           JMP MH.2L         ;TRY 1K MORE TO ASSIGN
```

```
!0272 N3MRT
01                ;SELECT A RANDOM BUFFER START WITHIN FIRST 248 OF
02                ;THE ASSIGNED SCRATCH AREA
03                MH.2C:  LCALL ARANG
04 20375 100270           ARANG-ASCRA*1B11+100010
05 20376 030432           LDA 2,MH248
06 20377 105000           MOV 0,1
07                        LCALL ADIVI
08 20400 100310           ADIVI-ASCRA*1B11+100010
09 20401 024152           LDA 1,SCRLO
10 20402 107000           ADD 0,1
11 20403 046420           STA 1,@MH24+1     ;DATA START ADDRS
12 20404 024154           LDA 1,DCHLO
13 20405 107000           ADD 0,1
14 20406 046416           STA 1,@MH24+2     ;CHANNEL START ADDRS
15 20407 024412           LDA 1,MH.KK       ;#1KS
16 20410 127120           ADDZL 1,1         ;*4
17 20411 125620           INCZR 1,1
18 20412 125500           INCL 1,1          ;+3 FOR 1ST 1K
19 20413 046412           STA 1,@MH24+3     ;# BLOKS/SECTORS AVAIL
20 20414 000520           JMP MH.03         ;FIGURE OU WHAT TO DO
21                MH.2R:  LCALL RSCRA       ;RELEASE 1K SCR
22 20415 100050           RSCRA-ASCRA*1B11+100010
23 20416 063077           HALT              ;NOT PROBABLE HALT
24 20417 000756           JMP MH.2C
25 20420 021665 MH.XX: MH.XI
26 20421 000000 MH.KK:    0
27 20422 000030 MH24:     24.
28 20423 020710           MHDST
29 20424 020711           MHCST
30 20425 020644           MH.BK
31 20426 020715           MHSEC
32 20427 000000 MH.ND:    0
33 20430 000370 MH248:    248.
```

```
!0273 N3MRT
01                ;NOW SELECT SECTOR TO EXERCISE
02 20431 020703           MHSCY
03 20432 020726           MH.WK
04                MH.3S:  LCALL ARANG
05 20433 100270           ARANG-ASCRA*1B11+100010
06 20434 105100           MOVL 0,1
07 20435 125220           MOVZR 1,1         ;080 NO OFLOW
08 20436 022767           LDA 0,@MH24+3     ;# 256 WD BLKS AVAIL.
09 20437 032772           LDA 2,@MH.3S-2    ;# SECTORS CYL.
10 20440 142432           SUBZ# 2,0,SZC     ;IF NOT MORE IN CYL
11 20441 141000           MOV 2,0           ;USE # PER CYLINDER
12 20442 112405           SUB 0,2,SNR       ;#PER CYL-#AVAIL
13 20443 151400           INC 2,2           ;DIV BY 0 IS OFLOW
14                        LCALL ADIVI       ;(0) WILL=SECTOR START
15 20444 100310           ADIVI-ASCRA*1B11+100010
16 20445 026760           LDA 1,@MH.ND-2
17 20446 032763           LDA 2,@MH.3S-2    .
18 20447 146432           SUBZ# 2,1,SZC     ;SKP=MORE IN CYLINDER
19 20450 145000           MOV 2,1           ;ONLY FILL CYL.'S WORTH
20 20451 131300           MOVS 1,2
21 20452 125300           MOVS 1,1
22 20453 127120           ADDZL 1,1         ;GET OUT OF STRT. SEC'S WAY
23 20454 123000           ADD 1,0           ;CREATE SEC CONTROL WD
24 20455 042751           STA 0,@MH24+4     ;START SECTOR AND #
25 20456 150600           NEGR 2,2
26 20457 151400           INC 2,2
27 20460 151500           INCL 2,2
28 20461 052751           STA 2,@MH.3S-1    ;# TIMES TO REGEN 4 WDS
```

```
!0274 N3MRT
01                ;NOW GET RANDOM DATA WORDS
02                        LCALL FRANG
03 20462 100450           FRANG-ASCRA*1B11+100010
04 20463 034743           LDA 3,MH.ND-1
05 20464 045402           STA 1,MHDW1-MHSEC,3      ;NEW RAN DATA WDS
06 20465 051403           STA 2,MHDW2-MHSEC,3
07 20466 041404           STA 0,MHDW3-MHSEC,3
08                        LCALL ARANG
09 20467 100270           ARANG-ASCRA*1B11+100010
10                ;ENTER THESE SELECTIONS INTO OPER. TABLE
11 20470 032440           LDA 2,@MH.WC+1
12 20471 126400           SUB 1,1
13 20472 034734           LDA 3,MH.ND-1
14 20473 041405           STA 0,MHDW4-MHSEC,3      ;STR 4TH RANDOM WRD
15 20474 045401           STA 1,1,3        ;CLEAR ERR CTR. THIS OPTBL
16 20475 045406           STA 1,MHRSC-MHSEC,3      ;CLR SINGLE SECTOR SEL.
17 20476 025776           LDA 1,-2,3
18 20477 021777           LDA 0,-1,3
19 20500 041000           STA 0,0,2
20 20501 151400           INC 2,2
21 20502 175400           INC 3,3
22 20503 125404           INC 1,1,SZR
23 20504 000773           JMP .-5
24                ;GENERATE DATA PATTERN INTO SCRATCH AREA
25
26 20505 034721 MH.3G:    LDA 3,MH.ND-1
27 20506 031773           LDA 2,MHDST-MHSEC,3
28 20507 021402           LDA 0,MHDW1-MHSEC,3      ;GET 2 DATA WORDS
29 20510 025403           LDA 1,MHDW2-MHSEC,3      ;TO FIRST 2 WORDS
30 20511 041000           STA 0,0,2        ;IN THE BUFFER
31 20512 045001           STA 1,1,2
32 20513 021404           LDA 0,MHDW3-MHSEC,3
33 20514 025405           LDA 1,MHDW4-MHSEC,3
34 20515 041002           STA 0,2,2
35 20516 045003           STA 1,3,2        ;1ST 4 WORDS RANDOM
36 20517 025411           LDA 1,MH.WK-MHSEC,3      ;GET # WORDS-4
37 20520 021000           LDA 0,0,2
38 20521 041004           STA 0,4,2
39 20522 151400           INC 2,2
40 20523 125404           INC 1,1,SZR
41 20524 000774           JMP .-4
42 20525 002401           JMP @.+1
43 20526 020734           MH.05    ;START SEEK AND WRITE
44 20527 177400 MH.WC:    -400
45 20530 020733           MHIDX
```

```
!0275 N3MRT
01 20531 000004 MH.44:  4
02              ;DATA CHANNEL AND SCRATCH ARE ASSIGNED
03              ;SELECT ONE OF AVAILABLE DRIVES, THEN
04              ;SELECT ONE OF THE THREE OP TABLES
05              ;AND PERFORM THE NEXT OPERATION REQ
06 20532 021536          MHGDS
07 20533 000010          10
08              MH.03:  LCALL ARANG      ;SELECT ONE OF AVAIL DRV'S
09 20534 100270          ARANG-ASCRA*1B11+100010
10 20535 105100          MOVL 0,1
11 20536 125220          MOVZR 1,1       ;0B0, NO DIV OFLOW
12 20537 030772          LDA 2,MH.44     ;# AVAILABLE DRIVES
13                       LCALL ADIVI     ;REM= DRIVE SELECTED
14 20540 100310          ADIVI-ASCRA*1B11+100010
15 20541 040563          STA 0,MHDNM
16 20542 101220          MOVZR 0,0
17 20543 126660          SUBCR 1,1        .
18 20544 101220          MOVZR 0,0
19 20545 125200          MOVR 1,1        ;DRV# POSITIONED FOR DOC'S
20 20546 044557          STA 1,MHDRV
21 20547 020555          LDA 0,MHDNM     ;CURRENT DRIVE#
22 20550 006762          JSR @MH.03-2    ;GET DRV DESCRIPTOR TBLS.
23 20551 020676          MHLSC
24 20552 022777          LDA 0,@.-1      ;GET LAST SECTOR WORD
25 20553 100015          COM# 0,0,SNR    ;NO SKP=NO DRIVE
26 20554 000760          JMP MH.03       ;TRY NEW RANDOM
27 20555 050532          STA 2,MHTBL     ;SAVE CYL. I.D. ADRS
28                       LCALL ARANG
29 20556 100270          ARANG-ASCRA*1B11+100010
30 20557 101300          MOVS 0,0
31 20560 126520          SUBZL 1,1
32 20561 125140          MOVOL 1,1       ;AC1=3
33 20562 101220          MOVZR 0,0
34 20563 107627          ANDZR 0,1,SBN   ;3 RESULT ILLEGAL
35 20564 125101          MOVL 1,1,SKP
36 20565 000772          JMP .-6         ;TRY NEXT 2 BITS
37 20566 102400          SUB 0,0
38 20567 125004          MOV 1,1,SZR
39 20570 020743          LDA 0,MH.03-1
40 20571 125224          MOVZR 1,1,SZR
41 20572 103000          ADD 0,0
42 20573 030514          LDA 2,MHTBL
43 20574 113000          ADD 0,2
44 20575 050536          STA 2,MHIDX     ;SAVE SELECTED OP TBL. ADRS
45 20576 034514          LDA 3,MHCST+1
46 20577 024514          LDA 1,MHCST+2
47 20600 021000          LDA 0,0,2
48 20601 041400          STA 0,0,3
49 20602 151400          INC 2,2
50 20603 175400          INC 3,3
51 20604 125404          INC 1,1,SZR
52 20605 000773          JMP .-5
53 20606 024506          LDA 1,MHCYL     ;GET CYL SEEK WORD
54 20607 125004          MOV 1,1,SZR     ;SKP=NON SEL. WRITE
55 20610 000440          JMP MH.4A       ;ALRDY ACTIVE, READ IT
```

```
!0276 N3MRT
01              ;SELECT A RANDOM CYLINDER NOT CURRENTLY USED
02              ;BY ANY OF THE OP TABLES ON THIS DISK
03              MH.3L:  LCALL ARANG      ;RANDOM SEL CYL
04 20611 100270          ARANG-ASCRA*1B11+100010
05 20612 105000          MOV 0,1
06 20613 030473          LDA 2,MHNCY     ;CYLINDER SELECTED
07                       LCALL ADIVI     ;MUST NOT BE
08 20614 100310          ADIVI-ASCRA*1B11+100010
09 20615 101231          MOVZR# 0,0,SKP  ;SKP=USE ALL # SZC=EVEN SNC=ODD
10 20616 000773          JMP MH.3L       ;DUAL. PROC. DELETE CYL.
11 20617 034466          LDA 3,MHSKM
12 20620 164000          COM 3,1
13 20621 107400          AND 0,1         ;HIGH CYL# BIT SOME DRV'S
14 20622 127120          ADDZL 1,1       ;POS. FOR SEEK
15 20623 163400          AND 3,0         ;LOW SEEK CYL#
16 20624 123000          ADD 1,0
17 20625 034457          LDA 3,MHSKB     ;SEEK BIT
18 20626 163000          ADD 3,0         ;CHNG. TO SEEK CYL#
19 20627 034460          LDA 3,MHTBL     ;ALREADY IN USE
20 20630 031400          LDA 2,0,3       ;AND CYL 0
21 20631 142415          SUB # 2,0,SNR   ;IS NOT EXERCISED
22 20632 000757          JMP MH.3L
23 20633 031410          LDA 2,10,3
24 20634 142415          SUB# 2,0,SNR
25 20635 000754          JMP MH.3L
26 20636 031420          LDA 2,20,3
27 20637 142415          SUB# 2,0,SNR
28 20640 000751          JMP MH.3L
29 20641 040453          STA 0,MHCYL     ;CYL SEL OK TO USE
30 20642 002401          JMP @.+1
31 20643 020433          MH.3S
32 20644 000000 MH.BK:  0
33 20645 000017 MH17:   17
34 20646 177404          -252.
35 20647 002000          185
```

```
!0277 N3MRT
01                      ;RANDOM SELECT TO UTILIZE CONTIGUOS SECTORS
02                      ;OR SINGLE SECTOR AREA OF THIS CYLINDER
03                      MH.4A:  LCALL ARANG     ;GET RANDOM #
04  20650 100270                ARANG-ASCRA*1B11+100010
05  20651 105300                MOVS 0,1        ;SEE WHITCH BYTE IS LARGER
06  20652 122023                ADCZ 1,0,SNC    ;SKP=USE SINGLE CYLINDER
07  20653 000463                JMP MH.04       ;USE ALL POS. SECTORS
08  20654 020447                LDA 0,MHRSC     ;GET SINGLE SECTOR WORD
09  20655 040440                STA 0,MHSEC     ;IN CASE ALREADY WRITTEN
10  20656 101004                MOV 0,0,SZR     ;SKP=NO SECTOR SELECTED
11  20657 000457                JMP MH.04       ;ALRDY WRITTEN, READ IT
12  20660 030423                LDA 2,MHSCY     ;# SECTORS PER CYL
13                              LCALL ADIVI     ;REM= SECTOR SELECTED
14  20661 100310                ADIVI-ASCRA*1B11+100010
15  20662 024765                LDA 1,MH17+2    ;1 SECTOR
16  20663 115400                INC 0,3         ;IF=LAST SECTOR
17  20664 156415                SUB# 2,3,SNR    ;2 IS FORCE EOC ON OPERATIONS
18  20665 123000                ADD 1,0
19  20666 123000                ADD 1,0
20  20667 040434                STA 0,MHRSC
21  20670 040425                STA 0,MHSEC
22  20671 034442                LDA 3,MHIDX
23  20672 041407                STA 0,7,3       ;UPDATE CONTROL TABLE
24  20673 020753                LDA 0,MH17+1    ;=-252.
25  20674 040432                STA 0,MH.WK     ;TO GEN 1 BLOCK'S WORTH
26  20675 000610                JMP MH.3G       ;GENERATE DATA START WRITE
```

```
!0278 N3MRT
01  20676 000000 MHLSC:  0         ;LST SECTOR*20
02  20677 000000 MHHDP:  0         ;HEAD POSITION BIT
03  20700 000000 MHNSC:  0         ;# OF SECTORS 1 SIDE
04  20701 000000 MHLHD:  0         ;LAST HD.# POS FOR DOC'S
05  20702 000000 MHNHD:  0         ;# OF HEADS(SURFACES)
06  20703 000000 MHSCY:  0         ;# OF SECTORS PER CYLINDER
07  20704 000000 MHSKB:  0         ;"SEEK" COMMAND BIT
08  20705 000000 MHSKM:  0         ;SEEK CYL# MASK LOW BITS
09  20706 000000 MHNCY:  0         ;# OF CYLINDERS THIS DISK
10        000011 MHDBL=.-MHLSC     ;# OF WORDS IN ABOVE
11  20707 022156 MHTBL:  MHDP0+MHDBL      ;PTR TO 3 DRV. OP TBLS
12  20710 000000 MHDST:  0
13  20711 000000 MHCST:  0
14  20712 020714         MHCYL
15  20713 177770         -MHCTL
16  20714 000000 MHCYL:  0         ;RANDOM CYL# PLUS SEEK
17  20715 000000 MHSEC:  0         ;#SEC'S*1B5+SECTOR # LOW BITS
18  20716 000000         0         ;ERROR CTR. THIS OP TABLE
19  20717 000000 MHDW1:  0         ;RANDOM DATA WORD 1
20  20720 000000 MHDW2:  0         ;WORD 2
21  20721 000000 MHDW3:  0         ;RANDOM WORD 3
22  20722 000000 MHDW4:  0         ;RANDOM WORD 4
23  20723 000000 MHRSC:  0         ;SINGLE RANDOM SECTOR
24        000010 MHCTL=.-MHCYL     ;LENGTH OF OPERATION TBL
25  20724 000000 MHDNM:  0         ;CURRENT DRIVE#
26  20725 000000 MHDRV:  0         ;DRV# POS. FOR DOC
27  20726 000000 MH.WK:  0
28  20727 001000 MH.SC:  1000
29  20730 000000 MH.CO:  0                 ;COMMAND + CYL #
30  20731 000000 MH.OC:  0                 ;SECT SIDE-SECK
31  20732 000000 MHSTA:  0                 ;STATUS READ
32  20733 000000 MHIDX:  0
```

```
!0279 N3MRT
01                ;CREATE DISK DOA'S AND DOC'S
02                ;MH.05 IS WRITE ENTER SUB SEEK/2 FROM SEEK
03                ;MH.04 IS READ ENTER SUB SEEK FROM SEEK
04 20734 020750 MH.05:  LDA 0,MHSKB       ;GET SEEK BIT
05 20735 101221         MOVZR 0,0,SKP     ;DIV BY 2
06 20736 020746 MH.04:  LDA 0,MHSKB
07 20737 024755         LDA 1,MHCYL       ;CYL# WITH SEEK IN IT
08 20740 106400         SUB 0,1           ;CREATE READ OR WRITE
09 20741 044767         STA 1,MH.CO       ;READ OR WRITE
10 20742 024753         LDA 1,MHSEC
11 20743 020453         LDA 0,MH.0X+1     ;TO MASK START SECTOR
12 20744 123400         AND 1,0
13 20745 106700         SUBS 0,1
14 20746 125220         MOVZR 1,1         ;POSITION# OF SECTORS
15 20747 125220         MOVZR 1,1         ;INTO BIT 15
16 20750 030730         LDA 2,MHNSC       ;# OF SECTORS 1 SIDE
17 20751 044760         STA 1,MH.0C       .
18 20752 105000         MOV 0,1           ;CALCULATE HEAD AND
19                      LCALL ADIVI       ;START SECTOR #
20 20753 100310         ADIVI=ASCRA*1B11+100010
21 20754 135000         MOV 1,3           ;AC1=HEAD #
22 20755 024754         LDA 1,MH.0C       ;REM IN AC0=START SECTOR
23 20756 103120 MH.5S:  ADDZL 0,0         ;SIDE AND
24 20757 103120         ADDZL 0,0         ;SECTOR POSITIONED
25 20760 030717         LDA 2,MHHDP       ;HEAD POSITION BIT
26 20761 151220         MOVZR 2,2
27 20762 175120         MOVZL 3,3         ;HD# OVER 1 BIT
28 20763 151224         MOVZR 2,2,SZR     ;SKP=HD# POSITIONED
29 20764 000776         JMP .-2
30 20765 163000         ADD 3,0           ;HD# INTO THE DOC
31 20766 030656         LDA 2,MH.BK       ;(2)=# 256 WD BLK AVAIL
32 20767 146432         SUBZ# 2,1,SZC
33 20770 145000         MOV 2,1           ;AC1=# SECTORS
34 20771 131300         MOVS 1,2          ;TO EXERCISE
35 20772 124400         NEG 1,1
36 20773 034652         LDA 3,MH17
37 20774 167400         AND 3,1
38 20775 123000         ADD 1,0
39 20776 024727         LDA 1,MHDRV
40 20777 123000         ADD 1,0           ;+DRIVE AND SURF.
41 21000 040731         STA 0,MH.0C       ;=SECTOR SIDE SECT K
42 21001 150400         NEG 2,2
43 21002 102520         SUBZL 0,0
44 21003 103100         ADDL 0,0
45 21004 113000         ADD 0,2
46 21005 050721         STA 2,MH.WK       ;BUFFER L - 4 WORDS
47 21006 020424 MH.5G:  LDA 0,MHSKI
48 21007 042406         STA 0,@MH.0X
49 21010 020721         LDA 0,MH.0C       ;DRV# SURF/SEC AND COUNT
50 21011 063233         DOCC 0,MHDSK
51 21012 024702         LDA 1,MHCYL
52 21013 065333         DOAP 1,MHDSK
53                      LCALL RETRN       ;START SEEK EXIT
54 21014 100210         RETRN=ASCRA*1B11+100010
55 21015 020205 MH.0X:  MH.00+2
56 21016 001777         1777
```

```
!0280 N3MRT
01                ;INTERRUPT SERVICE DISPATCHER
02 21017 000033         MHDSK
03 21020 060433 MH.IS:  DIA 0,MHDSK
04 21021 040711         STA 0,MHSTA
05 21022 032773         LDA 2,@MH.0X
06 21023 151132         MOVZL# 2,2,SZC
07 21024 001000         JMP 0,2
08
09 21025 020772         LDA 0,MH.IS-1
10 21026 040144         STA 0,UDEVI
11 21027 060233         NIOC MHDSK
12 21030 001400         JMP 0,3
13                ;PROCESS SEEK INTERRUPT
14 21031 004000         184
15 21032 121033 MHSKI:  @.+1
16 21033 024776         LDA 1,MHSKI-1     ;SEEK DONE BIT
17 21034 123414         AND# 1,0,SZR      ;SKP=NOT THIS DRV
18 21035 000405         JMP .+5           ;FOUND SEEK DONE
19 21036 125120         MOVZL 1,1
20 21037 125133         MOVZL# 1,1,SNC    ;SKP NOW IS ERROR
21 21040 000774         JMP MHSKI+2       ;TRY NEXT DRV
22 21041 000414         JMP MHRSK         ;ERROR RESEEK
23 21042 123400         AND 1,0           ;(0)=SINGLE BIT SEEK DONE
24 21043 061033         DOA 0,MHDSK
25 21044 024415         LDA 1,MHRWI       ;READ/WRITE ADRS
26 21045 046750         STA 1,@MH.0X
27 21046 020662 MHRDO:  LDA 0,MH.CO       ;COM# CYL#
28 21047 024662         LDA 1,MH.0C       ;SECTOR SIDE SECK
29 21050 030641         LDA 2,MHCST       ;CHANNEL ADRS
30 21051 072033         DOB 2,MHDSK
31 21052 067033         DOC 1,MHDSK
32 21053 061133         DOAS 0,MHDSK
33 21054 001400         JMP 0,3
34 21055 061033 MHRSK:  DOA 0,MHDSK
35 21056 024420         LDA 1,MHST.
36 21057 046736         STA 1,@MH.0X
37 21060 001400         JMP 0,3
38
39                ;PROCESS READ OR WRITE INTR
40
41 21061 121062 MHRWI:  @.+1
42 21062 105223         MOVZR 0,1,SNC     ;SKP DSK STATUS ERROR
43 21063 000407         JMP MHFOK
44 21064 065433         DIB 1,MHDSK
45 21065 046415         STA 1,@MHST.+4
46 21066 024410         LDA 1,MHST.
47 21067 046726         STA 1,@MH.0X
48 21070 061033         DOA 0,MHDSK
49 21071 001400         JMP 0,3
50 21072 061033 MHFOK:  DOA 0,MHDSK
51 21073 024404         LDA 1,MHST.+1
52 21074 046721         STA 1,@MH.0X
53 21075 001400         JMP 0,3
54 21076 021722 MHST.:  MHSTE
55 21077 021574         MHCOM
56 21100 021420         MHSE0
57 21101 021432         MHWDN
58 21102 021773         MHDIB
59 21103 001400 MH.14:  1400
60 21104 001760         1760
```

```
 0281 N3MRT
01 21105 001443          LSETB
02 21106 000016 MH.16:   16
```

```
10282 N3MRT
01                ;DETERMINE THE NUMBER OF SECTORS ON A SIDE
02                ;READ 2 SECTORS AT A TIME UNTILTHE SECTOR #
03                ;INCREMENTS BY MORE THAN 2 OR EOC SETS TO=1
04 21107 102400           SUB 0,0
05 21110 006770 MHSCQ:    JSR @MHST.+2      ;SEEK CYL 0
06 21111 024572           LDA 1,MHDRX       ;DRIVE # FOR DOC
07 21112 030774           LDA 2,MH.16
08 21113 147000           ADD 2,1
09 21114 107000           ADD 0,1           ;CREATE DOC
10 21115 040571           STA 0,.MHLSC      ;SAVE IN CASE LAST
11 21116 143000           ADD 2,0
12 21117 101400           INC 0,0           ;= NEXT SECTOR#
13 21120 101400           INC 0,0
14 21121 067233           DOCC 1,MHDSK      ;DISK# SECTOR AND -1
15 21122 032763           LDA 2,@MH.14+2    ;PROG. END BUFFER
16 21123 072033           DOB 2,MHDSK
17 21124 152400           SUB 2,2
18 21125 071133           DOAS 2,MHDSK      ;START READ
19 21126 006753           JSR @MHST.+3      ;WAIT FOR DONE
20 21127 000020           1B11              ;END BIT IS LEGAL
21 21130 072433           DIC 2,MHDSK
22 21131 024753           LDA 1,MH.14+1     ;SECTOR/HEAD MASK
23 21132 147400           AND 2,1
24 21133 030774           LDA 2,.-4         ;TO+1 SECTOR AG.
25 21134 113000           ADD 0,2
26 21135 132414           SUB# 1,2,SZR      ;SKP=HEAD INCREMENTED
27 21136 000404           JMP .+4           ;FOUND HEAD END
28 21137 070633           DIAC 2,MHDSK      ;GET STATUS
29 21140 151213           MOVR# 2,2,SNC     ;SKP=ASSUME EOC ERR
30 21141 000747           JMP MHSCQ         ;TRY NEXT SECTOR
31 21142 030741           LDA 2,MH.14
32 21143 147400           AND 2,1           ;CLEAR ALL BUT HEAD BIT
33 21144 044543           STA 1,.MHHDP      ;SAVE HEAD POS BIT
34 21145 101220           MOVZR 0,0
35 21146 101220           MOVZR 0,0
36 21147 101220           MOVZR 0,0         ;SAVE THE # OF SECTORS
37 21150 101220           MOVZR 0,0         ;POSITION IN BIT 15
38 21151 040537           STA 0,.MHNSC
```

```
!0283 N3MRT
01                ;NOW DETERMINE THE NUMBER OF SIDES PER CYLINDER
02                ;READ TWO SECTORS STARTING AT THE LAST SECTOR
03                ;ON EACH SIDE UNTIL EOC IS SET TO 1
04 21152 102400           SUB 0,0          ;START HEAD 0
05 21153 006725 MHHDQ:    JSR @MHST.+2     ;SEEK CYL#0
06 21154 024527           LDA 1,MHDRX      ;DRV# POS FOR DOC
07 21155 030731           LDA 2,MH.16      ;TO READ 2 SECTORS
08 21156 147000           ADD 2,1
09 21157 030527           LDA 2,.MHLSC     ;LAST SECTOR #
10 21160 147000           ADD 2,1
11 21161 107000           ADD 0,1          ;+ HEAD #
12 21162 067233           DOCC 1,MHDSK
13 21163 032722           LDA 2,@MH.14+2
14 21164 072033           DOB 2,MHDSK
15 21165 152400           SUB 2,2
16 21166 071133           DOAS 2,MHDSK     ;START READ
17 21167 006712           JSR @MHST.+3     ;WAIT READ DONE
18 21170 000020           1811             ;END ONLY LEGAL ERR
19 21171 024516           LDA 1,.MHHDP     ;HEAD POSITION BIT
20 21172 070433           DIA 2,MHDSK
21 21173 151222           MOVZR 2,2,SZC    ;SKP=NO ERR YET
22 21174 000403           JMP .+3          ;ASSUME EOC
23 21175 123000           ADD 1,0          ;+1 HEAD #
24 21176 000755           JMP MHHDQ        ;QUERY HEAD AGAIN
25 21177 040512           STA 0,.MHLHD     ;LAST HEAD # FOR DOC
26 21200 125220           MOVZR 1,1
27 21201 101220           MOVZR 0,0
28 21202 125224           MOVZR 1,1,SZR    ;POS. HD# TO BIT 15
29 21203 000776           JMP .-2
30 21204 101400           INC 0,0          ;=# HEADS THIS DISK
31 21205 040505           STA 0,.MHNHD     ;SAVED FOR SELECT
32 21206 126400           SUB 1,1
33 21207 030501           LDA 2,.MHNSC     ;# SECTORS PER SIDE
34 21210 147000           ADD 2,1
35 21211 100400           NEG 0,0
36 21212 100004           COM 0,0,SZR      ;#HDS-1=0 ACUMLATE
37 21213 000775           JMP .-3          ;# SECTORS PER CYLINDER
38 21214 044477           STA 1,.MHSCY
39 21215 062677           IORST
40 21216 002401           JMP @.+1
41 21217 021454           MHSTD
42 21220 021456           MHSTD+2
43 21221 001000 MH.B6:    1B6
```

```
!0284 N3MRT
01 21222 021536           MHGDS
02                ;DETERMINE WHERE THE SEEK BIT IS IN THE DOA INSTR.
03                ;IF 186 ACTS LIKE A DISK WRITE, SEEK IS 185
04 21223 020461 MHFSB:    LDA 0,MHDRX+1    ;CURRENT DRIVE#
05 21224 006776           JSR @MH.B6+1     ;GET DESCRIPTOR TABLE
06 21225 021306           .MHLS            ;INTO WORK AREA
07 21226 010460           ISZ .MHLS        ;SKP=NO SUCH DRIVE
08 21227 101001           MOV 0,0,SKP
09 21230 002770           JMP @MH.B6-1     ;SEE IF LAST DRIVE
10 21231 020770           LDA 0,MH.B6
11 21232 024451           LDA 1,MHDRX      ;CURRENT DOC FOR DRV
12 21233 067233           DOCC 1,MHDSK
13 21234 176400           SUB 3,3
14 21235 076033           DOB 3,MHDSK
15 21236 061133           DOAS 0,MHDSK
16 21237 176120           ADCZL 3,3        ;=-2
17 21240 175404           INC 3,3,SZR      ;STALL WAIT FOR DCH
18 21241 000777           JMP .-1
19 21242 075433           DIB 3,MHDSK      ;SEE IF ADRS INCR.'D
20 21243 062677           IORST
21 21244 175004           MOV 3,3,SZR      ;SKP=186 IS SEEK
22 21245 176401           SUB 3,3,SKP
23 21246 000403           JMP .+3
24 21247 056436           STA 3,@.MHBF     ;CAN'T USE B DCH MAP
25 21250 101120           MOVZL 0,0        ;NO, IT'S 185
26 21251 040443           STA 0,.MHSKB     ;SAVE SEEK BIT
27 21252 114240           COMOR 0,3
28 21253 175404           INC 3,3,SZR      ;STALL REST OF IORST
29 21254 000777           JMP .-1
30 21255 115220           MOVZR 0,3
31 21256 117000           ADD 0,3          ;CREATE RECALIBRATE
32 21257 067233           DOCC 1,MHDSK
33 21260 075333           DOAP 3,MHDSK
34 21261 004551           JSR MHWDN
35 21262 000000           0                ;WAIT DONE NO ERRS
36 21263 074433           DIA 3,MHDSK
37 21264 075233           DOAC 3,MHDSK
38 21265 114640           NEGOR 0,3        ;CREATE CYL# MASK
39 21266 174000           COM 3,3          ;IN CASE CYL# SPLIT
40 21267 054426           STA 3,.MHSM      ;SAVE SEEK LOW MASK
41 21270 054416           STA 3,.MHLS      ;FOR OK STATUS READ
42 21271 117000           ADD 0,3 ;SEEK TO LAST POS.
43 21272 067233           DOCC 1,MHDSK     ;SELECT DRV
44 21273 075333           DOAP 3,MHDSK     ;TRY CYL#=MASK
45 21274 004536           JSR MHWDN        ;WAIT FOR DONE
46 21275 000040           1B10             ;SEEK ERR ALLOWED
47 21276 064433           DIA 1,MHDSK
48 21277 101220           MOVZR 0,0        ;POSITION HIGH ORDER SEEK BIT
49 21300 125212           MOVR# 1,1,SZC    ;NO SEEK ERR=SKP
50 21301 101220           MOVZR 0,0        ;REAL SEEK WITHIN MASK
51 21302 000415           JMP MHCYQ-2
```

```
10285 N3MRT
01 21303 000000 MHDRX:  0
02 21304 000000         0
03 21305 020333 .MHBF:  MH.BF
04 21306 000000 .MHLS:  0
05 21307 000000 .MHHD:  0
06 21310 000000 .MHNS:  0
07 21311 000000 .MHLH:  0
08 21312 000000 .MHNH:  0
09 21313 000000 .MHSC:  0
10 21314 000000 .MHSK:  0
11 21315 000000 .MHSM:  0
12 21316 000000 .MHNC:  0
```

```
10286 N3MRT
01                      ;NOW PERFORM A SERIES OF SEEKS ADDING ONE BIT AT
02                      ;A TIME INTO THE CYLINDER # STARTING FROM HIGH AND
03                      ;WORKING TO THE LOW ORDER BIT. IF SEEK ERROR, TAKE
04                      ;THE BIT OUT OF THE CYLINDER # AND MOVE TO THE NEXT
05                      ;LEAST SIGNIFICANT BIT. THIS SEQUENCE SHOULD ALLOW
06                      ;US TO DETERMINE THE LAST VALID CYLINDER NUMBER
07 21317 176400                 SUB 3,3
08 21320 054776                 STA 3,.MHNCY    ;CURRENT CYL#
09 21321 065233 MHCYQ:  DOAC 1,MHDSK
10 21322 030773                 LDA 2,.MHSM     ;LOW ORDER SEEK MASK
11 21323 150400                 NEG 2,2
12 21324 151404                 INC 2,2,SZR     ;KILL SOME TME
13 21325 000777                 JMP .-1
14 21326 030755                 LDA 2,MHDRX
15 21327 034422                 LDA 3,MH.17
16 21330 173000                 ADD 3,2         ;TO READ 1 SECTOR
17 21331 073233                 DOCC 2,MHDSK    ;SELECT DRIVE
18 21332 125213                 MOVR# 1,1,SNC   ;SKP=ERROR LAST SEEK
19 21333 000420                 JMP MHCYX       ;OK TO SEEK AGN.
20 21334 024760                 LDA 1,.MHSKB    ;SEEK BIT
21 21335 131220                 MOVZR 1,2
22 21336 133000                 ADD 1,2         ;CREATE RECAL INSTR.
23 21337 071333                 DOAP 2,MHDSK    ;START IT
24 21340 064433                 DIA 1,MHDSK
25 21341 125202                 MOVR 1,1,SZC
26 21342 000776                 JMP .-2         ;WAIT SEEK ERR TO CLR
27 21343 004467                 JSR MHWDN       ;WAIT RECAL DONE
28 21344 000000                 0               ;NO ERRS ALLOWED
29 21345 064633                 DIAC 1,MHDSK
30 21346 152400                 SUB 2,2
31 21347 050737                 STA 2,.MHLS     ;TO READ FROM CYL#0
32 21350 000751                 JMP MHCYQ       ;STALL CHECK ERR AGN.
33 21351 000017 MH.17:  17
34 21352 001443                 LSETB
```

```
!0287 N3MRT
01 21353 101005 MHCYX:  MOV 0,0,SNR
02 21354 000436         JMP MHCYE       ;DONE ALL BITS IN CYL#
03 21355 024731         LDA 1,.MHLS     ;LAST OK CYL #
04 21356 032774         LDA 2,@MH.17+1
05 21357 072033         DOB 2,MHDSK     ;LOAD THE CA
06 21360 065133         DOAS 1,MHDSK    ;START A DISK READ
07 21361 004451         JSR MHWDN       ;WAIT READ WRITE DONE
08 21362 000004         1B13            ;CHKWRD LEGAL ERRS
09 21363 064433         DIA 1,MHDSK
10 21364 065033         DOA 1,MHDSK             ;CLEAR STATUS
11 21365 024730         LDA 1,.MHSM     ;SEEK MASK
12 21366 130000         COM 1,2
13 21367 034727         LDA 3,.MHNCY
14 21370 117000         ADD 0,3         ;ADD 1 BIT TO CYL#
15 21371 167400         AND 3,1         ;LOW ORDER CYL# BITS
16 21372 173400         AND 3,2         ;HIGH ORDER BIT
17 21373 153120         ADDZL 2,2       .
18 21374 147000         ADD 2,1         ;CYL# FOR DOAP
19 21375 044711         STA 1,.MHLS     ;FOR NEXT RD. IF OK STATUS
20 21376 030716         LDA 2,.MHSKB    ;+SEEK BIT
21 21377 147000         ADD 2,1         ;SEEK+CYL#
22 21400 065333         DOAP 1,MHDSK    ;START SEEK
23 21401 054715         STA 3,.MHNCY    ;SAVE CYL #
24 21402 004430         JSR MHWDN       ;WAIT SEEK DONE OR ERR
25 21403 000040         1B10            ;ONLY SEEK ERR VALID
26 21404 064633         DIAC 1,MHDSK
27 21405 034711         LDA 3,.MHNCY
28 21406 125212         MOVR# 1,1,SZC   ;SKP=NO SEEK ERROR
29 21407 116400         SUB 0,3         ;ERR IS REMOVE BIT AG.
30 21410 101220         MOVZR 0,0       ;SKP IS CYL# FORMED
31 21411 000707         JMP MHCYQ-1     ;QUERY NEXT CYL#
```

```
!0288 N3MRT
01 21412 010704 MHCYE:  ISZ .MHNCY      ;=#CYL.'S INC. 0
02 21413 002401         JMP @.+1
03 21414 021107         MHSCQ-1
04              ;SEEK FOR CYL#0
05 21415 000004         4
06 21416 174000         174000
07 21417 000000         0       ;SAVE PC
08 21420 054777 MHSE0:  STA 3,.-1
09 21421 034662         LDA 3,MHDRX
10 21422 077233         DOCC 3,MHDSK
11 21423 034671         LDA 3,.MHSK
12 21424 075333         DOAP 3,MHDSK
13 21425 004405         JSR MHWDN       ;WAIT SEEK DONE
14 21426 000000         0               ;NO ERRS ALLOWED
15 21427 074433         DIA 3,MHDSK
16 21430 075233         DOAC 3,MHDSK
17 21431 002766         JMP @MHSE0-1
18              ;WAIT FOR ANY DONE OR ERROR FLG
19              ;CALL+1 CONTAINS ANY LEGAL ERROR BITS
20 21432 050417 MHWDN:  STA 2,MHSAVR
21 21433 044417         STA 1,MHSAVR+1
22 21434 030762         LDA 2,MHSE0-2   ;R/W DONE SEEK DONE AND ERR
23 21435 064433         DIA 1,MHDSK
24 21436 147415         AND# 2,1,SNR    ;WAIT FOR A DONE
25 21437 000776         JMP .-2
26 21440 064433         DIA 1,MHDSK
27 21441 125213         MOVR# 1,1,SNC   ;SKP=ERROR BIT
28 21442 000404         JMP .+4
29 21443 031400         LDA 2,0,3       ;GET LEGAL ERR BIT
30 21444 147415         AND# 2,1,SNR    ;SKP IF LEGAL ERR
31 21445 063077         HALT            ;DISK ERR NOT EXPECTED
32 21446 030403         LDA 2,MHSAVR
33 21447 024403         LDA 1,MHSAVR+1
34 21450 001401         JMP 1,3         ;RETURN TO CALL
35 21451 000000 MHSAVR: 0
36 21452 000000         0
37 21453 040000         181
38              ;THIS DRIVE HAS BEEN SIZED SAVE THE DATA ASSOCIATED WITH IT
39 21454 020630 MHSTD:  LDA 0,MHDRX+1
40 21455 004507         JSR MHSDS       ;SAV DESCR. INDRIVE TABLE
41 21456 020626         LDA 0,MHDRX+1
42 21457 024736         LDA 1,MHSE0-3
43 21460 101400         INC 0,0
44 21461 122415         SUB# 1,0,SNR    ;SKP=NOT DONE ALL DRIVES
45 21462 002452         JMP @MHRTX+3
46 21463 040621         STA 0,MHDRX+1   ;NEXT DRV#
47 21464 024767         LDA 1,MHSTD-1
48 21465 020616         LDA 0,MHDRX
49 21466 107000         ADD 0,1
50 21467 044614         STA 1,MHDRX     ;DRV# FOR DOC INSTR.
51 21470 002401         JMP @.+1
52 21471 021223         MHFSB   ;DO NEXT DRIVE
```

```
!0289 N3MRT
01                ;REPORT TO THE OPERATOR THE MAGNITUDE AND
02                ;SPECIFICS OF THE DISK SYSTEM
03 21472 020341           MH.02
04 21473 020204           MH.00+1
05 21474 020776 MH.1R:    LDA 0,MH.1R-2
06 21475 042776           STA 0,@MH.1R-1   ;TO ENTER AT EXECUTE
07 21476 102400           SUB 0,0
08 21477 040605           STA 0,MHDRX+1
09 21500 020431           LDA 0,MHRTX
10                        LCALL ERRTX      ;TYPE HEADER
11 21501 100170           ERRTX-ASCRA*1B11+100010
12 21502 020602 MH1RL:    LDA 0,MHDRX+1    ;GET DRV #
13 21503 004433           JSR MHGDS        ;DESCRIPTOR TABLE
14 21504 021306           .MHLS            ;INTO WORK AREA
15 21505 022777           LDA 0,@.-1       ;GET SECTOR WORD
16 21506 100005           COM 0,0,SNR      ;SKP=DRIVE EXISTS
17 21507 000413           JMP MH1RR        ;DRIVE NON-EXIS
18                        LCALL PCRLF      ;CAR RET/L.F.
19 21510 100110           PCRLF-ASCRA*1B11+100010
20 21511 026424           LDA 1,@MHRTX+4   ;DISK #
21                        LCALL PDECI      ;PRINT IT
22 21512 100150           PDECI-ASCRA*1B11+100010
23 21513 024603           LDA 1,.MHNCY
24                        LCALL PDECI      ;PRINT # CYLINDERS
25 21514 100150           PDECI-ASCRA*1B11+100010
26 21515 030767           LDA 2,MH1RL+2
27 21516 025002           LDA 1,.MHNSC-.MHLS,2     ;# SECTORS PER SIDE
28                        LCALL PDECI
29 21517 100150           PDECI-ASCRA*1B11+100010
30 21520 025004           LDA 1,.MHNHD-.MHLS,2     ;# OF SURFACES
31                        LCALL PDECI      ;PRINT # SURFACES
32 21521 100150           PDECI-ASCRA*1B11+100010
33 21522 030762 MH1RR:    LDA 2,MH1RL+2
34 21523 011376           ISZ MHDRX+1-.MHLS,2
35 21524 021376           LDA 0,MHDRX+1-.MHLS,2
36 21525 024670           LDA 1,MHSE0-3
37 21526 122415           SUB# 1,0,SNR
38 21527 002404           JMP @MHRTX+2
39 21530 000752           JMP MH1RL
40 21531 022116 MHRTX:    MHTXR
41 21532 020427           MH.ND
42 21533 020341           MH.02
43 21534 120331           @MH.S3
44 21535 021304           MHDRX+1
```

```
!0290 N3MRT
01                ;MOVE THE DEFINITIONS TABLE FOR THIS DISK
02                ;INTO THE PROGRAM WORK TABLE AREA
03 21536 175400 MHGDS:    INC 3,3
04 21537 054415           STA 3,MHGD3
05 21540 035777           LDA 3,-1,3
06 21541 030414           LDA 2,MHDPT      ;POINTS TO DRIVE PTR. TABLE
07 21542 113000           ADD 0,2
08 21543 031000           LDA 2,0,2        ;TABLE PTR. THIS DRIVE
09 21544 024417 MHCDS:    LDA 1,MHWEL      ;WORK AREA LENGTH
10 21545 021000           LDA 0,0,2
11 21546 041400           STA 0,0,3
12 21547 151400           INC 2,2
13 21550 175400           INC 3,3
14 21551 125404           INC 1,1,SZR
15 21552 000773           JMP .-5          ;MOVE THE WHOLE TABLE
16 21553 002401           JMP @MHGD3
17 21554 000000 MHGD3:    0        ;SAVE PC
18 21555 021556 MHDPT:    .+1
19 21556 022145           MHDP0            ;DRIVE 0 PTR
20 21557 022206           MHDP1            ;DRV 1 PTR
21 21560 022247           MHDP2            ;DRIVE 2 TABLES
22 21561 022310           MHDP3            ;AND DRIVE 3 TABLES
23 21562 020676 MHWEP:    MHLSC            ;LAST SECTOR SIDE
24 21563 177767 MHWEL:    -MHDBL           ;DISK DEF. BLOCK LENGTH
25                ;MOVE THE DEFINITIONS FOR THIS DRIVE INTO
26                ;THE DRIVE TABLE SAVING ALL USEABLE INFO
27 21564 054770 MHSDS:    STA 3,MHGD3
28 21565 030770           LDA 2,MHDPT
29 21566 113000           ADD 0,2
30 21567 035000           LDA 3,0,2
31 21570 030402           LDA 2,.+2
32 21571 000753           JMP MHCDS        ;MOVE DEF. BLK TO DRV TABLE
33 21572 021306           .MHLSC
```

```
!0291 N3MRT
01                      ;READ OR WRITE OK NO STAT ERRS
02                      ;COMPARE DATA IN BUFFER
03                      ;AGAINST WHAT'S SUPPOSED TO BE THERE
04 21573 020717                 MHDW1
05 21574 034777 MHCOM:  LDA 3,MHCOM-1   ;ADRS OF DATA WORDS
06 21575 031771         LDA 2,-7,3      ;START ADRS
07 21576 021400         LDA 0,0,3
08 21577 025000         LDA 1,0,2       ;COMPARE
09 21600 122414         SUB# 1,0,SZR    ;FIRST 4 WORDS
10 21601 000477         JMP MHERR       ;AGAINST EVERY
11 21602 175400         INC 3,3         ;OTHER IN BUFFER
12 21603 021400         LDA 0,0,3
13 21604 025001         LDA 1,1,2
14 21605 122414         SUB# 1,0,SZR
15 21606 000472         JMP MHERR
16 21607 175400         INC 3,3
17 21610 021400         LDA 0,0,3        •
18 21611 025002         LDA 1,2,2
19 21612 106414         SUB# 0,1,SZR
20 21613 000465         JMP MHERR
21 21614 175400         INC 3,3
22 21615 021400         LDA 0,0,3
23 21616 025003         LDA 1,3,2
24 21617 106414         SUB# 0,1,SZR
25 21620 000460         JMP MHERR
26 21621 035404         LDA 3,4,3       ;#WORDS - 4
27 21622 021000         LDA 0,0,2
28 21623 025004         LDA 1,4,2
29 21624 122414         SUB# 1,0,SZR
30 21625 000453         JMP MHERR
31 21626 055000         STA 3,0,2       ;CLR BUFFER NXT RD
32 21627 151400         INC 2,2         ;BUMP ADRS
33 21630 175404         INC 3,3,SZR     ;SKP DONEALL
34 21631 000771         JMP .-7
35                      LCALL ARANG     ;BITS 1 AND 15=1
36 21632 100270         ARANG=ASCRA*1B11+100010
37 21633 101232         MOVZR# 0,0,SZC  ;RELEASE THIS
38 21634 103123         ADDZL 0,0,SNC   ;OP TABLE
39 21635 000403         JMP .+3         ;EITHER = 0 KEEP IT
40 21636 102400 MHERL:  SUB 0,0                  ,
41 21637 042440         STA 0,@MH.SX+2
42                      LCALL ARANG
43 21640 100270         ARANG=ASCRA*1B11+100010
44 21641 101102         MOVL 0,0,SZC    ;NEXT 2 BITS =1?
45 21642 101103         MOVL 0,0,SNC    ;IS RELEASE THIS BUFFER
46 21643 002432         JMP @MH.SX      ;SELECT NEW STRT BUFFER
47 21644 032420 MH.X3:  LDA 2,@MHHIK
48 21645 020136         LDA 0,MPSWT
49 21646 101005         MOV 0,0,SNR
50 21647 000416         JMP MH.XI
51 21650 024413         LDA 1,MH.37
52 21651 146032         ADCZ# 2,1,SZC   ;SKP IF HIGHK >=32K
53 21652 000413         JMP MH.XI
54                      LCALL RDMAP
55 21653 100610         RDMAP=ASCRA*1B11+100010
56 21654 101001         MOV 0,0,SKP
57 21655 000776         JMP MH.X4
58                      LCALL RSCRA
59 21656 100050         RSCRA=ASCRA*1B11+100010
60 21657 101001         MOV 0,0,SKP
```

```
 0292 N3MRT
01 21660 000776         JMP MH.X5
02 21661 002401         JMP @.+1
03 21662 020344         MH.02+3
04 21663 000037 MH.37:  37
05 21664 001502 MHHIK:  HIGHK
```

```
10293 N3MRT
01                MH.XI:  LCALL RDMAP
02 21665 100610           RDMAP=ASCRA*1B11+100010
03 21666 101001           MOV 0,0,SKP
04 21667 000776           JMP MH.XI
05                MH.X2:  LCALL RSCRA
06 21670 100050           RSCRA=ASCRA*1B11+100010
07 21671 102401           SUB 0,0,SKP
08 21672 000776           JMP MH.X2
09 21673 042403           STA 0,@MH.SX+1
10                        LCALL RETRN
11 21674 100210           RETRN=ASCRA*1B11+100010
12 21675 020375 MH.SX:    MH.2C
13 21676 020205           MH.00+2
14 21677 120733           @MHIDX
15 21700 054464 MHERR:    STA 3,MH.SA     ;ERROR IN DATA COMPARE
16                        LCALL ERROI
17 21701 100350           ERROI=ASCRA*1B11+100010
18 21702 000401           JMP .+1
19 21703 020471           LDA 0,MHTX1
20                        LCALL ERRTX
21 21704 100170           ERRTX=ASCRA*1B11+100010
22 21705 020457           LDA 0,MH.SA
23 21706 026462           LDA 1,@MHDSX
24 21707 032462           LDA 2,@MHCSX
25                        LCALL ERROC
26 21710 100370           ERROC=ASCRA*1B11+100010
27 21711 000401           JMP .+1
28 21712 020475           LDA 0,MHTX2
29                        LCALL ERRTX
30 21713 100170           ERRTX=ASCRA*1B11+100010
31 21714 022456           LDA 0,@MHSTX
32 21715 026451           LDA 1,@MH.CX
33 21716 032451           LDA 2,@MH.OX
34                        LCALL ERROC
35 21717 100370           ERROC=ASCRA*1B11+100010
36 21720 000503           JMP MHRCL       ;RECALIBRATE DRV
37 21721 000502           JMP MHRCL
38 21722 020451 MHSTE:    LDA 0,MHDIB     ;DISK STATUS ERROR
39 21723 032440           LDA 2,@MH.BX            ;GET B FLG
40 21724 150005           COM 2,2,SNR             ;SKP IS NOT USING B DCH MAP
41 21725 000410           JMP MHNBS               ;USING B MAP DON'T CHECK B.S. FLG
42 21726 032440           LDA 2,@MH.CX
43 21727 026436           LDA 1,@MHSKX    ;CHECK IF DOING A WRITE
44 21730 125220           MOVZR 1,1       ;OPERATION
45 21731 133405           AND 1,2,SNR     ;SKP=WRITE
46 21732 000403           JMP MHNBS       ;NOT WRITE,DON'T CK FOR BSF
47 21733 101112           MOVL# 0,0,SZC
48 21734 000417           JMP MHBSF
49 21735 066433 MHNBS:    DIC 1,MHDSK
50 21736 032434           LDA 2,@MHSTX
51 21737 034413           LDA 3,MH.25+1   ;GET EOC BIT
52 21740 157424           ANDZ 2,3,SZR    ;SKP=NOT EOC STATUS
53 21741 101040           MOVO 0,0
54 21742 036430           LDA 3,@MHSTX
55 21743 101003           MOV 0,0,SNC     ;SKP IF ERR WAS EOC
56 21744 000734           JMP MHERR
57 21745 030404           LDA 2,MH.25     ;1 SECTOR -4 WORDS
58 21746 034625           LDA 3,MHCOM-1
59 21747 051407           STA 2,MH.WK-MHDW1,3     ;SO COMPARE STOPS
60 21750 000624           JMP MHCOM
```

```
 0294 N3MRT
01 21751 177404 MH.25:  -252.
02 21752 000020          1B11
```

```
10295 N3MRT
01                ;ERROR WAS BAD SECTOR FLAG
02 21753 020513 MHBSF:  LDA 0,MHTX3
03                      LCALL ERRTX
04 21754 100170         ERRTX-ASCRA*1B11+100010
05 21755 022411         LDA 0,@MH.CX
06 21756 066433         DIC 1,MHDSK
07 21757 030414         LDA 2,MHDIB
08                      LCALL ERROC
09 21760 100370         ERROC-ASCRA*1B11+100010
10 21761 000401         JMP .+1
11 21762 000654         JMP MHERL
12 21763 020333 MH.BX:  MH.BF
13 21764 000000 MH.SA: 0
14 21765 020704 MHSKX:  MHSKB
15 21766 020730 MH.CX: MH.CO
16 21767 020731 MH.OX: MH.OC
17 21770 020710 MHDSX: MHDST
18 21771 020711 MHCSX: MHCST
19 21772 020732 MHSTX: MHSTA
20 21773 000000 MHDIB:  0
21 21774 021775 MHTX1: .+1
22 21775 005215         .TXTE [<15><12>MH.SA    MHDST   MHCST[
23 22007 022010 MHTX2: .+1
24 22010 005215         .TXTE [<15><12>MHSTA    MHDOA   MHDOC[
```

```
10296 N3MRT
01                ;RECALIBRATE THE DRIVE THEN RETRY IF
02                ;LESS THAN 4 ERRS THIS OP TABLE
03 22022 021636         MHERL
04 22023 020424 MHRCL:  LDA 0,MHRCI
05 22024 042652         STA 0,@MH.SX+1
06 22025 030652         LDA 2,MH.SX+2   ;ADRS MHIDX
07 22026 035000         LDA 3,0,2       ;OP. TABLE PTR
08 22027 011402         ISZ 2,3         ;+1 ERR TRY CTR
09 22030 021402         LDA 0,2,3
10 22031 101220         MOVZR 0,0
11 22032 101224         MOVZR 0,0,SZR   ;NO SKP=<4 RETRY'S
12 22033 002767         JMP @MHRCL-1    ;RELEASE THIS BADDY
13 22034 022733         LDA 0,@MH.OX    ;GET FOR DOC
14 22035 025351         LDA 1,MHSKB-MHIDX,2     ;GET SEEK BIT
15 22036 131220         MOVZR 1,2
16 22037 036727         LDA 3,@MH.CX    ;GET READ/WRITE DOA
17 22040 157400         AND 2,3         ;SAVE RD/WRT BIT
18 22041 054405         STA 3,MHRCI-1   ;SAVE IT
19 22042 133000         ADD 1,2         ;CREATE RECAL INSTR.
20 22043 063233         DOCC 0,MHDSK
21 22044 071333         DOAP 2,MHDSK    ;START RECAL
22                      LCALL RETRN
23 22045 100210         RETRN-ASCRA*1B11+100010
24                ;PROCESS RECALIBRATE INTERRUPT
25 22046 000000         0       ;SAVE FOR READ OR WRITE BIT
26 22047 122050 MHRCI:  @.+1
27 22050 061233         DOAC 0,MHDSK
28 22051 024405         LDA 1,.+5
29 22052 101202         MOVR 0,0,SZC    ;SKP=NO ERR
30 22053 024404         LDA 1,.+4       ;TYPE AGAIN
31 22054 046622         STA 1,@MH.SX+1
32 22055 001400         JMP 0,3         ;DISMISS INTA
33 22056 022060         .+2             ;TO TRY AGAIN
34 22057 021722         MHSTE           ;TYPE STATUS ERR
35                ;DETERMINE IF REDOING A READ OR WRITE
36 22060 020766         LDA 0,MHRCI-1   ;=0 IS READ
37 22061 101004         MOV 0,0,SZR     ;NOT =0 IS WRITE
38 22062 002402         JMP @.+2        ;REGEN DATA "WRITE"
39 22063 002402         JMP @.+2        ;JUST RE-READ
40 22064 020505         MH.3G
41 22065 021006         MH.5G
42 22066 022067 MHTX3:  .+1
43 22067 005215         .TXTE   ¦<15><12>DEV #33 BAD SECTOR FLAG
44 22103 106507 <15><12> MHDOA  MHDIC   MHDIB¦
45 22116 005215 MHTXR:  .TXTE ¦<15><12>DEV.#33 M.H.DISK<15><12>
46 22130 051504 DSK#    #CYLS   #SEC/S  #SURF.¦
47                ;DISK DEFINITIONS TABLE 1 FOR EACH DISK
48 22145 000041 MHDP0:  .BLK MHDBL+MHCTL+MHCTL+MHCTL
49 22206 000041 MHDP1:  .BLK MHDBL+MHCTL+MHCTL+MHCTL
50 22247 000041 MHDP2:  .BLK MHDBL+MHCTL+MHCTL+MHCTL
51 22310 000041 MHDP3:  .BLK MHDBL+MHCTL+MHCTL+MHCTL
52 22351 000000 MHDRE:  0       ;END OF DISK TABLES
```

```
!0297 N3MRT
01        000000 .NOLOC PZDSK
02               ;       .TITL PZDSK
03               ;MOVABLE HEAD 6060/61 DISK TEST COMPATABLE WITH
04               ;THE DIAGNOSTIC LINKER
05
06        000027 ZBDSK=27
07               ;DEFINITIONS BLOCK TO LINKER
08                       NEXTT Z8.00
09        022352 LMEML=.
10        000167         .LOC LPG0
11 00167  022355         Z8.00
12        000170 LPG0=.
13        022352         .LOC LMEML
14 22352  000000         0       ;TEST PASS CTR
15 22353  000000         0       ;TEST ERROR CTR
16 22354  000000         0       ;INTERRUPT TIMEOUT SWITCH
17 22355  022402 Z8.00:          Z8.01    .
18 22356  022503                 Z8.02
19 22357  000000                 0
20 22360  000000                 0
21 22361  177777                 -1
22 22362  000077                 77
23 22363  023576                 Z8.XI
24 22364  023576                 Z8.XI
25        022365                 .TXTE !
26 22365  030066 6060/61 DISK(PRI)!
27        030066
28        033257
29        120261
30        144504
31        045523
32        050050
33        144722
34        000251
```

```
!0298 N3MRT
01               ;EITHER NO DISK CONTROL,ADAPTER IS OFF
02               ;OR THERE ARE NO DISKS ON LINE
03 22376  102000 Z8NCT:  ADC 0,0
04 22377  040760         STA 0,Z8.00+2
05 22400  001400         JMP 0,3         ;TEST DELETED
06               ;DETERMINE IF SYSTEM HAS A MOVING HEAD DISK
07               ;DISABLE TEST ENTER IF NONE EXISTS
08 22401  000777         777
09 22402  020777 Z8.01:  LDA 0,.-1       ;GET CYL TRIAL BITS
10 22403  063027         DOC 0,ZBDSK     ;WRITE CYL #
11 22404  066427         DIC 1,ZBDSK     ;READ CYL #
12 22405  106404         SUB 0,1,SZR     ;SKIP=DSK CNTL EXISTS
13 22406  000770         JMP Z8NCT       ;DON'T RUN THIS TEST
14 22407  102400         SUB 0,0
15 22410  024456         LDA 1,Z8MK1
16 22411  030464         LDA 2,Z8TBA     ;CLR ACTIVE TABLS
17 22412  041000         STA 0,0,2
18 22413  151400         INC 2,2
19 22414  125404         INC 1,1,SZR
20 22415  000775         JMP .-3
21 22416  062027         DOB 0,ZBDSK
22               ;DETERMINE HOW MANY DISKS ON SYSTEM
23 22417  102400         SUB 0,0
24 22420  024453         LDA 1,Z8.K4     ;=BIT 10
25 22421  040544         STA 0,Z8.ND     ;0 # DISKS
26 22422  030452         LDA 2,Z8TXA     ;TO SET EXIST SWS
27 22423  050020         STA 2,20        ;BY DOUBLE@
28 22424  130700 Z8.1L:  NEGS 1,2
29 22425  151404         INC 2,2,SZR     ;STALL FOR AWHILE
30 22426  000777         JMP .-1
31 22427  061027         DOA 0,ZBDSK     ;SELECT CURRENT DRIVE
32 22430  060027         NIO ZBDSK
33 22431  060027         NIO ZBDSK
34 22432  071627         DIBC 2,ZBDSK    ;CHECK FOR RDY
35 22433  153120         ADDZL 2,2
36 22434  153123         ADDZL 2,2,SNC   ;SKIP IS RDY
37 22435  000402         JMP .+2
38 22436  010527         ISZ Z8.ND       ;INC # OF DRIVES
39 22437  152402         SUB 2,2,SZC     ;SKP=DRV EXISTS
40 22440  152000         ADC 2,2         ;DELETE THIS DRIVE
41 22441  052020         STA 2,@20       ;
42 22442  123000         ADD 1,0
43 22443  111300         MOVS 0,2
44 22444  151123         MOVZL 2,2,SNC   ;SKP=DONE 4 DRIVES
45 22445  000757         JMP Z8.1L
```

```
!0299 N3MRT
01                ;STALL FOR AWHILE, THEN DETERMINE THE
02                ;CHARACTERISTICS OF EACH INDIVIDUAL DISK
03                ;BUILDING A DISK DESCRIPTOR TABLE
04 22446 152400           SUB 2,2
05 22447 052431           STA 2,@ZBTXB+2
06 22450 052427           STA 2,@ZBTXB+1
07 22451 020414           LDA 0,Z8MK1-1
08 22452 040704           STA 0,Z8.00+1    ;SO DESCRIPTION IS TYPED
09 22453 020512           LDA 0,Z8.ND      ;=# OF DISKS ON LINE
10 22454 101005           MOV 0,0,SNR      ;SKP= AT LEAST ONE
11 22455 000721           JMP Z8NCT        ;DID NOT FIND ANY
12 22456 020411           LDA 0,Z8.K1
13 22457 024411           LDA 1,Z8.K2      ;INT MASK BITS
14 22460 030411           LDA 2,Z8.K3      ;INT SERV DIRECTOR
15 22461 054411           STA 3,Z8.S3
16 22462 006101           JSR @EINTS
17 22463 002401           JMP @.+1          .
18 22464 023257           Z8FS8            ;FIND # OF CYL'S
19 22465 023401           Z8.1R            ;ADRS TO TYPE DESCRIP.
20 22466 177574 Z8MK1:  Z8DP0-Z8DRE
21 22467 000027 Z8.K1:  Z8DSK
22 22470 000777 Z8.K2:  777
23 22471 023151 Z8.K3:  Z8.IS
24 22472 000000 Z8.S3:  0
25 22473 000040 Z8.K4:   1B10
26
27
28 22474 123462 Z8TXA:  @Z8DPT
29 22475 024072 Z8TBA:   Z8DP0
30 22476 000102 Z8TXB:   .TXTE (8(
31 22477 023314           Z8DRX
32 22500 023315           Z8DRX+1
33 22501 000077 Z8TXC:   .TXTE (?(
```

```
!0300 N3MRT
01
02
03                ;EXECUTE PORTION OF DISK TEST
04 22502 022557           Z8.KK
05 22503 030654 Z8.02:    LDA 2,Z8.00+2
06 22504 151004           MOV 2,2,SZR
07 22505 001000           JMP 0,2
08                        LCALL ASCRA      ;GET 1K SCR
09 22506 100010           ASCRA-ASCRA*1B11+100010
10 22507 002447           JMP @Z8.XX       ;NONE AVAIL
11                Z8.2J:  LCALL ADMAP      ;ASSIGN TO DCH
12 22510 100550           ADMAP-ASCRA*1B11+100010
13 22511 002445           JMP @Z8.XX
14                        LCALL ARANG
15 22512 100270           ARANG-ASCRA*1B11+100010
16 22513 105000           MOV 0,1
17 22514 152520           SUBZL 2,2
18 22515 153100           ADDL 2,2         ;=4
19 22516 151120           MOVZL 2,2        ;=8
20 22517 102400           SUB 0,0
21 22520 042762           STA 0,@Z8.02-1
22                        LCALL ADIVI
23 22521 100310           ADIVI-ASCRA*1B11+100010
24 22522 100405           NEG 0,0,SNR
25 22523 000410           JMP Z8.2C
26                Z8.2L:  LCALL ESCRA
27 22524 100030           ESCRA-ASCRA*1B11+100010
28 22525 000406           JMP Z8.2C        ;CANT EXPAND MORE
29                        LCALL EDMAP      ;ASSIGN TO DCH
30 22526 100570           EDMAP-ASCRA*1B11+100010
31 22527 000424           JMP Z8.2R        ;CANT ASSIGN MORE
32 22530 012752           ISZ @Z8.02-1     ;#1K'S +1
33 22531 101404           INC 0,0,SZR
34 22532 000772           JMP Z8.2L        ;TRY 1K MORE TO ASSIGN
```

```
!0301 N3MRT
01                ;SELECT A RANDOM BUFFER START WITHIN FIRST 248 OF
02                ;THE ASSIGNED SCRATCH AREA
03                ZB.2C:  LCALL ARANG
04 22533 100270           ARANG-ASCRA*1B11+100010
05 22534 030432           LDA 2,ZB248
06 22535 105000           MOV 0,1
07                        LCALL ADIVI
08 22536 100310           ADIVI-ASCRA*1B11+100010
09 22537 024152           LDA 1,SCRLO
10 22540 107000           ADD 0,1
11 22541 046420           STA 1,@ZB24+1    ;DATA START ADDRS
12 22542 024154           LDA 1,DCHLO
13 22543 107000           ADD 0,1
14 22544 046416           STA 1,@ZB24+2    ;CHANNEL START ADDRS
15 22545 024412           LDA 1,ZB.KK      ;#1KS
16 22546 127120           ADDZL 1,1        ;*4
17 22547 125620           INCZR 1,1        *
18 22550 125500           INCL 1,1         ;+3 FOR 1ST 1K
19 22551 046412           STA 1,@ZB24+3    ;# BLOKS/SECTORS AVAIL
20 22552 000517           JMP ZB.03        ;FIGURE OU WHAT TO DO
21                ZB.2R:  LCALL RSCRA      ;RELEASE 1K SCR
22 22553 100050           RSCRA-ASCRA*1B11+100010
23 22554 063077           HALT             ;NOT PROBABLE HALT
24 22555 000756           JMP ZB.2C
25 22556 023576 ZB.XX: ZB.XI
26 22557 000000 ZB.KK:  0
27 22560 000030 ZB24:   24.
28 22561 023034           ZBDST
29 22562 023035           ZBCST
30 22563 022773           ZB.BK
31 22564 023041           ZBSEC
32 22565 000000 ZB.ND:  0
33 22566 000370 ZB248:  248.
```

```
!0302 N3MRT
01
02                ;NOW SELECT SECTOR TO EXERCISE
03 22567 023027           ZBSCY
04 22570 023052           ZB.WK
05                Z8.3S:  LCALL ARANG
06 22571 100270           ARANG-ASCRA*1B11+100010
07 22572 105100           MOVL 0,1
08 22573 125220           MOVZR 1,1        ;0B0 NO OFLOW
09 22574 022767           LDA 0,@ZB24+3    ;# 256 WD BLKS AVAIL.
10 22575 032772           LDA 2,@Z8.3S-2   ;# SECTORS CYL.
11 22576 142432           SUBZ# 2,0,SZC    ;IF NOT MORE IN CYL
12 22577 141000           MOV 2,0          ;USE # PER CYLINDER
13 22600 112405           SUB 0,2,SNR      ;#PER CYL-#AVAIL
14 22601 151400           INC 2,2          ;DIV BY 0 IS OFLOW
15                        LCALL ADIVI      ;(0) WILL=SECTOR START
16 22602 100310           ADIVI-ASCRA*1B11+100010
17 22603 026760           LDA 1,@ZB.ND-2
18 22604 032763           LDA 2,@Z8.3S-2
19 22605 146432           SUBZ# 2,1,SZC    ;SKP=MORE IN CYLINDER
20 22606 145000           MOV 2,1          ;ONLY FILL CYL.'S WORTH
21 22607 131300           MOVS 1,2
22 22610 125300           MOVS 1,1
23 22611 127120           ADDZL 1,1        ;GET OUT OF STRT. SEC'S WAY
24 22612 123000           ADD 1,0          ;CREATE SEC CONTROL WD
25 22613 042751           STA 0,@ZB24+4    ;START SECTOR AND #
26 22614 150600           NEGR 2,2
27 22615 151400           INC 2,2
28 22616 151500           INCL 2,2
29 22617 052751           STA 2,@Z8.3S-1   ;# TIMES TO REGEN 4 WDS
```

```
!0303 N3MRT
01                ;NOW GET RANDOM DATA WORDS
02                        LCALL FRANG
03 22620 100450           FRANG-ASCRA*1B11+100010
04 22621 034743           LDA 3,ZB.ND-1
05 22622 045402           STA 1,ZBDW1-ZBSEC,3     ;NEW RAN DATA WDS
06 22623 051403           STA 2,ZBDW2-ZBSEC,3
07 22624 041404           STA 0,ZBDW3-ZBSEC,3
08                        LCALL ARANG
09 22625 100270           ARANG-ASCRA*1B11+100010
10                ;ENTER THESE SELECTIONS INTO OPER. TABLE
11 22626 032440           LDA 2,@ZB.WC+1
12 22627 126400           SUB 1,1
13 22630 034734           LDA 3,ZB.ND-1
14 22631 041405           STA 0,ZBDW4-ZBSEC,3     ;STR 4TH RANDOM WRD
15 22632 045401           STA 1,1,3               ;CLEAR ERR CTR. THIS OPTBL
16 22633 045406           STA 1,ZBRSC-ZBSEC,3     ;CLR SINGLE SECTOR SEL.
17 22634 025776           LDA 1,-2,3
18 22635 021777           LDA 0,-1,3
19 22636 041000           STA 0,0,2
20 22637 151400           INC 2,2
21 22640 175400           INC 3,3
22 22641 125404           INC 1,1,SZR
23 22642 000773           JMP .-5
24                ;GENERATE DATA PATTERN INTO SCRATCH AREA
25
26 22643 034721 ZB.3G:    LDA 3,ZB.ND-1
27 22644 031773           LDA 2,ZBDST-ZBSEC,3
28 22645 021402           LDA 0,ZBDW1-ZBSEC,3     ;GET 2 DATA WORDS
29 22646 025403           LDA 1,ZBDW2-ZBSEC,3     ;TO FIRST 2 WORDS
30 22647 041000           STA 0,0,2               ;IN THE BUFFER
31 22650 045001           STA 1,1,2
32 22651 021404           LDA 0,ZBDW3-ZBSEC,3
33 22652 025405           LDA 1,ZBDW4-ZBSEC,3
34 22653 041002           STA 0,2,2
35 22654 045003           STA 1,3,2               ;1ST 4 WORDS RANDOM
36 22655 025411           LDA 1,ZB.WK-ZBSEC,3     ;GET # WORDS-4
37 22656 021000           LDA 0,0,2
38 22657 041004           STA 0,4,2
39 22660 151400           INC 2,2
40 22661 125404           INC 1,1,SZR
41 22662 000774           JMP .-4
42 22663 002401           JMP @.+1
43 22664 023063           ZB.05           ;START SEEK AND WRITE
44 22665 177400 ZB.WC:    -400
45 22666 023057           ZBIDX
```

```
!0304 N3MRT
01                ;DATA CHANNEL AND SCRATCH ARE ASSIGNED
02                ;SELECT ONE OF AVAILABLE DRIVES, THEN
03                ;SELECT ONE OF THE THREE OP TABLES
04                ;AND PERFORM THE NEXT OPERATION REQ
05 22667 023443           ZBGDS
06 22670 000010           10
07                ZB.03:  LCALL ARANG     ;SELECT ONE OF AVAIL DRV'S
08 22671 100270           ARANG-ASCRA*1B11+100010
09 22672 105100           MOVL 0,1
10 22673 125220           MOVZR 1,1       ;0B0, NO DIV OFLOW
11 22674 030774           LDA 2,ZB.03-1   ; 10
12 22675 151220           MOVZR 2,2       ;NOW  ='S 4 MAX # OF DRIVES
13                        LCALL ADIVI     ;REM= DRIVE SELECTED
14 22676 100310           ADIVI-ASCRA*1B11+100010
15 22677 040551           STA 0,ZBDNM
16 22700 105320           MOVZS 0,1       ;POSITION DRIVE # TO BITS
17 22701 125200           MOVR 1,1        ;9 AND 10 FOR DOA'S
18 22702 125200           MOVR 1,1
19 22703 125200           MOVR 1,1        ;DRV# POSITIONED FOR DOA'S
20 22704 044545           STA 1,ZBDRV
21 22705 020543           LDA 0,ZBDNM     ;CURRENT DRIVE#
22 22706 006761           JSR @ZB.03-2    ;GET DRV DESCRIPTOR TBLS.
23 22707 023022           ZBLSC
24 22710 022777           LDA 0,@.-1      ;GET LAST SECTOR WORD
25 22711 100015           COM# 0,0,SNR    ;NO SKP=NO DRIVE
26 22712 000757           JMP ZB.03       ;TRY NEw RANDOM
27 22713 050520           STA 2,ZBTBL     ;SAVE CYL. I.D. ADRS
28                        LCALL ARANG
29 22714 100270           ARANG-ASCRA*1B11+100010
30 22715 101300           MOVS 0,0
31 22716 126520           SUBZL 1,1
32 22717 125140           MOVOL 1,1       ;AC1=3
33 22720 101220           MOVZR 0,0
34 22721 107627           ANDZR 0,1,SBN   ;3 RESULT ILLEGAL
35 22722 125101           MOVL 1,1,SKP
36 22723 000772           JMP .-6         ;TRY NEXT 2 BITS
37 22724 102400           SUB 0,0
38 22725 125004           MOV 1,1,SZR
39 22726 020742           LDA 0,ZB.03-1
40 22727 125224           MOVZR 1,1,SZR
41 22730 103000           ADD 0,0
42 22731 030502           LDA 2,ZBTBL
43 22732 113000           ADD 0,2
44 22733 050524           STA 2,ZBIDX     ;SAVE SELECTED OP TBL. ADRS
45 22734 034502           LDA 3,ZBCST+1
46 22735 024502           LDA 1,ZBCST+2
47 22736 021000           LDA 0,0,2
48 22737 041400           STA 0,0,3
49 22740 151400           INC 2,2
50 22741 175400           INC 3,3
51 22742 125404           INC 1,1,SZR
52 22743 000773           JMP .-5
53 22744 024474           LDA 1,ZBCYL     ;GET CYL SEEK WORD
54 22745 125004           MOV 1,1,SZR     ;SKP=NON SEL. WRITE
55 22746 000431           JMP ZB.4A       ;ALRDY ACTIVE, READ IT
```

```
10305 N3MRT
01                ;SELECT A RANDOM CYLINDER NOT CURRENTLY USED
02                ;BY ANY OF THE OP TABLES ON THIS DISK
03                ZB.3L:  LCALL ARANG      ;RANDOM SEL CYL
04 22747 100270           ARANG-ASCRA*1B11+100010
05 22750 105000           MOV 0,1
06 22751 030461           LDA 2,ZBNCY      ;CYLINDER SELECTED
07                        LCALL ADIVI      ;MUST NOT BE
08 22752 100310           ADIVI-ASCRA*1B11+100010
09 22753 101231           MOVZR# 0,0,SKP   ;SKP=USE ALL # SZC=EVEN SNC=ODD
10 22754 000773           JMP ZB.3L        ;DUAL. PROC. DELETE CYL.
11 22755 103240           ADDOR 0,0        ;1B0 SO CYL# 0 GETS USED ALSO
12 22756 034455           LDA 3,ZBT8L      ;ALREADY IN USE
13 22757 031400           LDA 2,0,3        ;AND CYL 0
14 22760 142415           SUB # 2,0,SNR    ;IS NOT EXERCISED
15 22761 000766           JMP ZB.3L
16 22762 031410           LDA 2,10,3
17 22763 142415           SUB# 2,0,SNR      •
18 22764 000763           JMP ZB.3L
19 22765 031420           LDA 2,20,3
20 22766 142415           SUB# 2,0,SNR
21 22767 000760           JMP ZB.3L
22 22770 040450           STA 0,ZBCYL      ;CYL SEL OK TO USE
23 22771 002401           JMP @.+1
24 22772 022571           ZB.3S
25 22773 000000 ZB.8K:    0
26 22774 000017 ZB17: 17
27 22775 177404           -252.
28 22776 002000           185
29
```

```
10306 N3MRT
01                ;RANDOM SELECT TO UTILIZE CONTIGUOS SECTORS
02                ;OR SINGLE SECTOR AREA OF THIS CYLINDER
03                ZB.4A:  LCALL ARANG      ;GET RANDOM #
04 22777 100270           ARANG-ASCRA*1B11+100010
05 23000 105300           MOVS 0,1         ;SEE WHITCH BYTE IS LARGER
06 23001 122023           ADCZ 1,0,SNC     ;SKP=USE SINGLE CYLINDER
07 23002 000463           JMP ZB.04        ;USE ALL POS. SECTORS
08 23003 020444           LDA 0,ZBRSC      ;GET SINGLE SECTOR WORD
09 23004 040435           STA 0,ZBSEC      ;IN CASE ALREADY WRITTEN
10 23005 101004           MOV 0,0,SZR      ;SKP=NO SECTOR SELECTED
11 23006 000457           JMP ZB.04        ;ALRDY WRITTEN, READ IT
12 23007 030420           LDA 2,ZBSCY      ;# SECTORS PER CYL
13                        LCALL ADIVI      ;REM= SECTOR SELECTED
14 23010 100310           ADIVI-ASCRA*1B11+100010
15 23011 024765           LDA 1,ZB17+2     ;1 SECTOR
16 23012 123000           ADD 1,0
17 23013 040434           STA 0,ZBRSC
18 23014 040425           STA 0,ZBSEC
19 23015 034442           LDA 3,ZBIDX
20 23016 041407           STA 0,7,3        ;UPDATE CONTROL TABLE
21 23017 020756           LDA 0,ZB17+1                .
22 23020 040432           STA 0,ZB.WK      ;TO GEN 1 BLOCK'S WORTH
23 23021 000622           JMP ZB.3G        ;GENERATE DATA START WRITE
```

```
!0307 N3MRT
01 23022 000000 ZBLSC:  0          ;LST SECTOR*20
02 23023 000000 ZBHDP:  0          ;HEAD POSITION BIT
03 23024 000000 ZBNSC:  0          ;# OF SECTORS 1 SIDE
04 23025 000000 ZBLHD:  0          ;LAST HD.# POS FOR DOC'S
05 23026 000000 ZBNHD:  0          ;# OF HEADS(SURFACES)
06 23027 000000 ZBSCY:  0          ;# OF SECTORS PER CYLINDER
07 23030 000000 ZBSKB:  0          ;"SEEK" COMMAND BIT
08 23031 000000 ZBSKM:  0          ;SEEK CYL# MASK LOW BITS
09 23032 000000 ZBNCY:  0          ;# OF CYLINDERS THIS DISK
10       000011 ZBDBL=.-ZBLSC      ;# OF WORDS IN ABOVE
11 23033 024103 ZBTBL:  ZBDP0+ZBDBL       ;PTR TO 3 DRV. OP TBLS
12 23034 000000 ZBDST:  0
13 23035 000000 ZBCST:  0
14 23036 023040         ZBCYL
15 23037 177770         -ZBCTL
16 23040 000000 ZBCYL:  0          ;RANDOM CYL# PLUS SEEK
17 23041 000000 ZBSEC:  0          ;#SEC'S*1@5+SECTOR # LOW BITS
18 23042 000000         0          ;ERROR CTR. THIS OP TABLE
19 23043 000000 ZBDW1:  0          ;RANDOM DATA WORD 1
20 23044 000000 ZBDW2:  0          ;WORD 2
21 23045 000000 ZBDW3:  0          ;RANDOM WORD 3
22 23046 000000 ZBDW4:  0          ;RANDOM WORD 4
23 23047 000000 ZBRSC:  0          ;SINGLE RANDOM SECTOR
24       000010 ZBCTL=.-ZBCYL      ;LENGTH OF OPERATION TBL
25 23050 000000 ZBDNM:  0          ;CURRENT DRIVE#
26 23051 000000 ZBDRV:  0          ;DRV# POS. FOR DOC
27 23052 000000 ZB.WK:  0
28 23053 001000 ZB.SC:  1000
29 23054 000000 ZB.CO:  0                  ;COMMAND + CYL #
30 23055 000000 ZB.OC:  0                  ;SECT SIDE-SECK
31 23056 000000 ZBSTA:  0                  ;STATUS READ
32 23057 000000 ZBIDX:  0
33 23060 000037 ZB37:   37
```

```
!0308 N3MRT
01              ;CREATE DISK DOA'S AND DOC'S
02              ;ZB.05 IS WRITE ENTER SUB SEEK/2 FROM SEEK
03              ;ZB.04 IS READ ENTER SUB SEEK FROM SEEK
04 23061 001777         1777
05 23062 003400         1688              ;DISK WRITE COMMAND
06 23063 020777 ZB.05:  LDA 0,.-1         ;GET SEEK BIT
07 23064 101001         MOV 0,0,SKP       ;DIV BY 2
08 23065 102400 ZB.04:  SUB 0,0           ;0'S= DISK READ
09 23066 024763         LDA 1,ZBDRV       ;DRV# FOR DOA'S
10 23067 123000         ADD 1,0
11 23070 040764         STA 0,ZB.CO       ;READ OR WRITE
12 23071 024750         LDA 1,ZBSEC
13 23072 020767         LDA 0,ZB.05-2     ;1777 TO MASK START SECTOR
14 23073 123400         AND 1,0
15 23074 106700         SUBS 0,1
16 23075 125220         MOVZR 1,1         ;POSITION# OF SECTORS
17 23076 125220         MOVZR 1,1         ;INTO BIT 15
18 23077 030725         LDA 2,ZBNSC       ;# OF SECTORS 1 SIDE
19 23100 044755         STA 1,ZB.OC
20 23101 105000         MOV 0,1           ;CALCULATE HEAD AND
21                      LCALL ADIVI       ;START SECTOR #
22 23102 100310         ADIVI-ASCRA*1B11+100010
23 23103 135000         MOV 1,3           ;AC1=HEAD #
24 23104 024751         LDA 1,ZB.OC       ;REM IN AC0=START SECTOR
25 23105 103120 ZB.5S:  ADDZL 0,0         ;SIDE AND
26 23106 103120         ADDZL 0,0         ;SECTOR POSITIONED
27 23107 101120         MOVZL 0,0
28 23110 030713         LDA 2,ZBHDP       ;HEAD POSITION BIT
29 23111 151220         MOVZR 2,2
30 23112 175120         MOVZL 3,3         ;HD# OVER 1 BIT
31 23113 151224         MOVZR 2,2,SZR     ;SKP=HD# POSITIONED
32 23114 000776         JMP .-2
33 23115 163000         ADD 3,0           ;HD# INTO THE DOC
34 23116 030655         LDA 2,ZB.BK       ;(2)=# 256 WD BLK AVAIL
35 23117 146432         SUBZ# 2,1,SZC
36 23120 145000         MOV 2,1           ;AC1=# SECTORS
37 23121 131300         MOVS 1,2          ;TO EXERCISE
38 23122 124400         NEG 1,1
39 23123 034735         LDA 3,ZB37
40 23124 167400         AND 3,1
41 23125 123000         ADD 1,0
42 23126 040727         STA 0,ZB.OC       ;=SECTOR SIDE SECT K
43 23127 150400         NEG 2,2
44 23130 102520         SUBZL 0,0
45 23131 103100         ADDL 0,0
46 23132 113000         ADD 0,2
47 23133 050717         STA 2,ZB.WK       ;BUFFER L - 4 WORDS
48 23134 020427 ZB.5G:  LDA 0,ZBSKI
49 23135 042411         STA 0,@ZB.0X
50 23136 020713         LDA 0,ZBDRV
51 23137 024671         LDA 1,ZBSKB
52 23140 107000         ADD 0,1
53 23141 065027         DOA 1,ZBDSK       ;SELECT DRIVE FOR SEEK
54 23142 020676         LDA 0,ZBCYL       ;CYL# WITH 180 FOR CYL 0
55 23143 103240         ADDOR 0,0         ;0B0
56 23144 063327         DOCP 0,ZBDSK      ;START SSEEK
57                      LCALL RETRN       ;START SEEK EXIT
58 23145 100210         RETRN-ASCRA*1B11+100010
59 23146 022357 ZB.0X:  ZB.00+2
```

```
10309 N3MRT                                                                    0310 N3MRT
01                                                                             01 23235 023501          ZBCOM
02 23147 000377          377     ;TO MASK ERROR BITS IN DIA
03               ;INTERRUPT SERVICE DISPATCHER
04 23150 000027          ZBDSK
05 23151 060427 ZB.IS:   DIA 0,ZBDSK
06 23152 040704          STA 0,ZBSTA
07 23153 032773          LDA 2,@ZB.0X
08 23154 151132          MOVZL# 2,2,SZC  ;EXPECTED INTR?
09 23155 001000          JMP 0,2
10 23156 020772          LDA 0,ZB.IS-1
11 23157 040144          STA 0,UDEVI
12 23160 060227          NIOC ZBDSK
13 23161 001400          JMP 0,3
14
15               ;PROCESS SEEK INTERRUPT
16 23162 002000          185
17 23163 123164 ZBSKI:   @.+1            .
18 23164 024776          LDA 1,ZBSKI-1   ;SEEK DONE BIT
19 23165 123414          AND# 1,0,SZR    ;SKP=NOT THIS DRV
20 23166 000405          JMP .+5         ;FOUND SEEK DONE
21 23167 125120          MOVZL 1,1
22 23170 127133          ADDZL# 1,1,SNC  ;SKP NOW IS ERROR
23 23171 000774          JMP ZBSKI+2     ;TRY NEXT DRV
24 23172 000415          JMP ZBRSK       ;ERROR RESEEK
25 23173 024754          LDA 1,ZB.IS-2   ;377 TO MASK ERR BITS FROM DIA
26
27 23174 123414          AND# 1,0,SZR    ;SKP=NO DISK ERRS
28 23175 000412          JMP ZBRSK       ;RESTART SEEK AFTER TYPING
29 23176 024426          LDA 1,ZBRWI     ;READ/WRITE ADRS
30 23177 046747          STA 1,@ZB.0X
31 23200 020654 ZBRDO:   LDA 0,ZB.CO     ;COM# DRV#
32 23201 024654          LDA 1,ZB.OC     ;SECTOR SIDE SECK
33 23202 030633          LDA 2,ZBCST     ;CHANNEL ADRS
34 23203 061027          DOA 0,ZBDSK
35 23204 072027          DOB 2,ZBDSK
36 23205 067127          DOCS 1,ZBDSK
37 23206 001400          JMP 0,3
38 23207 024425 ZBRSK:   LDA 1,ZBST.
39 23210 046736          STA 1,@ZB.0X
40 23211 024412          LDA 1,ZBRWI-1   ;76000  TOO MA$K DONE BITS
41 23212 123400          AND 1,0
42 23213 071427          DIB 2,ZBDSK
43 23214 052417          STA 2,@ZB.CO+1
44 23215 030613          LDA 2,ZBSKB
45 23216 143000          ADD 2,0
46 23217 030632          LDA 2,ZBDRV
47 23220 143000          ADD 2,0
48 23221 061227          DOAC 0,ZBDSK    ;CLEAR FLAGS RESERVE DRV
49 23222 001400          JMP 0,3
50 23223 076000          76000
51               ;PROCESS READ OR WRITE INTR
52 23224 123225 ZBRWI:   @.+1
53 23225 000401          JMP .+1
54 23226 105222          MOVZR 0,1,SZC   ;SKP NO DSK STATUS ERROR
55 23227 000760          JMP ZBRSK
56 23230 024405 ZBFOK:   LDA 1,ZBST.+1
57 23231 000757          JMP ZBRSK+1
58 23232 000000 ZB.CO:   0
59 23233 023675          ZBDIB
60 23234 023641 ZBST.:   ZBSTE
```

```
!0311 N3MRT
01                ;SET THE NUMBER OF SECTORS ON A SIDE
02 23236 102400           SUB 0,0
03 23237 024723 ZBSCQ:    LDA 1,ZBSKI-1    ;BIT 5 IS HD. POSITION BIT
04 23240 044461           STA 1,.ZBHDP     ;INTO CONTROL TABLE
05 23241 020507           LDA 0,ZBKNA      ;24 SECTORS ON A SURFACE
06 23242 040460           STA 0,.ZBNSC     ;INTO CONTROL TABLE
07                ;NOW SET THE NUMBER OF SIDES PER CYLINDER
08 23243 102400           SUB 0,0          ;START HEAD 0
09 23244 020505 ZBHDQ:    LDA 0,ZBKNB      ;6060/61 HAS 19 SURFACES
10 23245 040457           STA 0,.ZBNHD     ;SAVED FOR SELECT
11 23246 126400           SUB 1,1
12 23247 030453           LDA 2,.ZBNSC     ;# SECTORS PER SIDE
13 23250 147000           ADD 2,1
14 23251 100400           NEG 0,0
15 23252 100004           COM 0,0,SZR      ;#HDS-1=0 ACUMLATE
16 23253 000775           JMP .-3          ;# SECTORS PER CYLINDER
17 23254 044451           STA 1,.ZBSCY     •
18 23255 002401           JMP @.+1
19 23256 023332           ZBSTD
20
```

```
!0312 N3MRT
01                ;DETERMINE IF DISK IS HIGH OR LOW DENSITY TYPE
02 23257 020436 ZBFSH:    LDA 0,ZBDRX+1    ;CURRENT DRIVE#
03 23260 004563           JSR ZBGDS        ;GET DESCRIPTOR TABLE
04 23261 023320           .ZBLS            ;INTO WORK AREA
05 23262 010436           ISZ .ZBLS        ;SKP=NO SUCH DRIVE
06 23263 101001           MOV 0,0,SKP
07 23264 002432           JMP @ZBDRX+2     ;SEE IF LAST DRIVE
08 23265 020465           LDA 0,ZBKNC      ;SEEK BIT=BIT 7
09 23266 040440           STA 0,.ZBSKB     ;SAVE SEEK BIT
10 23267 024425           LDA 1,ZBDRX      ;DRIVE # FOR DOA
11 23270 115220           MOVZR 0,3        ;CREATE RECAL CMD
12 23271 137000           ADD 1,3          ;ADD IN DRV
13 23272 075327           DOAP 3,ZBDSK     ;RECAL CMD
14 23273 004463           JSR ZBWDN        ;WAIT FOR DONE
15 23274 000000           0                ;NO ERRS ALLOWED
16 23275 074427           DIA 3,ZBDSK      ;GET STATUS
17 23276 175120           MOVZL 3,3
18 23277 075227           DOAC 3,ZBDSK     ;CLR DONE FLG
19 23300 030417           LDA 2,ZBDRX+3    ;=777
20 23301 107000           ADD 0,1
21 23302 065027           DOA 1,ZBDSK      ;ISSUE SEEK CMD
22 23303 073327           DOCP 2,ZBDSK     ;SET CYL # TO TRY
23 23304 004452           JSR ZBWDN        ;WAIT FOR DONE
24 23305 000000           0
25 23306 065627           DIBC 1,ZBDSK     ;GET STATUS
26 23307 020445           LDA 0,ZBKND+1    ;# OF CYL IN DUEL DENSITY
27 23310 125202           MOVR 1,1,SZC     ;SKIP IF ERR SEEK
28 23311 020442           LDA 0,ZBKND      ;WAS NOT DUEL DEN. DSK
29 23312 040416           STA 0,.ZBNCY
30 23313 000723           JMP ZBSCQ-1      ;LOAD SEC'S/SURFACES
31 23314 000000 ZBDRX:    0                ;DRIVE # FOR DOA
32 23315 000000           0                ;CURRENT DRIVE #
33 23316 023334           ZBSTD+2
34 23317 000777           777
35 23320 000000 .ZBLS:    0
36 23321 000000 .ZBHD:    0
37 23322 000000 .ZBNS:    0
38 23323 000000 .ZBLH:    0
39 23324 000000 .ZBNH:    0
40 23325 000000 .ZBSC:    0
41 23326 000000 .ZBSK:    0
42 23327 000000 .ZBSM:    0
43 23330 000000 .ZBNC:    0
```

```
10313 N3MRT
01                ;THIS DRIVE HAS BEEN SIZED SAVE THE DATA ASSOCIATED WITH IT
02 23331 000004           4
03 23332 020763 ZBSTD:    LDA 0,ZBDRX+1
04 23333 004536           JSR ZBSDS         ;SAV DESCR. INDRIVE TABLE
05 23334 020761           LDA 0,ZBDRX+1
06 23335 024774           LDA 1,ZBSTD-1
07 23336 101400           INC 0,0
08 23337 122415           SUB# 1,0,SNR      ;SKP=NOT DONE ALL DRIVES
09 23340 002502           JMP @ZBRTX+3
10 23341 040754           STA 0,ZBDRX+1     ;NEXT DRV#
11 23342 024413           LDA 1,ZBKND+2     ;=40
12 23343 020751           LDA 0,ZBDRX
13 23344 107000           ADD 0,1
14 23345 044747           STA 1,ZBDRX       ;DRV# FOR DOA INSTR.
15 23346 002401           JMP @.+1
16 23347 023257           ZBFSB             ;DO NEXT DRIVE
17 23350 000030 ZBKNA:    24.               .
18 23351 000023 ZBKNB:    19.
19 23352 000400 ZBKNC:    187
20 23353 000633 ZBKND:    411.
21 23354 001457           815.
22 23355 000040           40
23
24                ;WAIT FOR DONE, CALLER+1 CONTAINS ANY
25                ;LEGAL ERROR STATUS BITS
26 23356 050420 ZBWDN:    STA 2,ZBSV2
27 23357 044420           STA 1,ZBSV2+1
28 23360 030415           LDA 2,ZBSV2-1
29 23361 064427           DIA 1,ZBDSK       ;GET STATUS WORD
30 23362 147415           AND# 2,1,SNR      ;CK FOR DONE
31 23363 000776           JMP .-2
32 23364 064427           DIA 1,ZBDSK       ;CK FOR ERRS
33 23365 125233           MOVZR# 1,1,SNC    ;SKP IF ERR FLG SET
34 23366 000404           JMP .+4
35 23367 031400           LDA 2,0,3         ;ANY LEGAL ER BITS?
36 23370 147415           AND# 2,1,SNR      ;SKP IF ERR WAS LEGAL ONE
37 23371 063077           HALT              ;UNEXPECTED STATUS ERR
38 23372 030404           LDA 2,ZBSV2
39 23373 024404           LDA 1,ZBSV2+1
40 23374 001401           JMP 1,3           ;RETURN TO CALLER
41 23375 036000           36000             ;DRV DONE MSK
42 23376 000000 ZBSV2:    0
43 23377 000000           0
```

```
10314 N3MRT
01
02                ;REPORT TO THE OPERATOR THE MAGNITUDE AND
03                ;SPECIFICS OF THE DISK SYSTEM
04 23400 022356           ZB.00+1
05 23401 020440 ZB.1R:    LDA 0,ZBRTX+2
06 23402 042776           STA 0,@ZB.1R-1    ;TO ENTER AT EXECUTE
07 23403 102400           SUB 0,0
08 23404 042432           STA 0,@ZBRTX-1
09 23405 020432           LDA 0,ZBRTX
10                        LCALL ERRTX       ;TYPE HEADER
11 23406 100170           ERRTX-ASCRA*1B11+100010
12 23407 022427 ZB1RL:    LDA 0,@ZBRTX-1    ;GET DRV #
13 23410 004433           JSR ZBGDS         ;DESCRIPTOR TABLE
14 23411 023320           .ZBLS             ;INTO WORK AREA
15 23412 022777           LDA 0,@.-1        ;GET SECTOR WORD
16 23413 100005           COM 0,0,SNR       ;SKP=DRIVE EXISTS
17 23414 000413           JMP ZB1RR         ;DRIVE NON-EXIS
18                        LCALL PCRLF       ;CAR RET/L.F.
19 23415 100110           PCRLF-ASCRA*1B11+100010
20 23416 026420           LDA 1,@ZBRTX-1    ;DISK #
21                        LCALL PDECI       ;PRINT IT
22 23417 100150           PDECI-ASCRA*1B11+100010
23 23420 024710           LDA 1,.ZBNCY
24                        LCALL PDECI       ;PRINT # CYLINDERS
25 23421 100150           PDECI-ASCRA*1B11+100010
26 23422 030767           LDA 2,ZB1RL+2
27 23423 025002           LDA 1,.ZBNSC-.ZBLS,2    ;# SECTORS PER SIDE
28                        LCALL PDECI
29 23424 100150           PDECI-ASCRA*1B11+100010
30 23425 025004           LDA 1,.ZBNHD-.ZBLS,2    ;# OF SURFACES
31                        LCALL PDECI       ;PRINT # SURFACES
32 23426 100150           PDECI-ASCRA*1B11+100010
33 23427 030762 ZB1RR:    LDA 2,ZB1RL+2
34 23430 012406           ISZ @ZBRTX-1
35 23431 022405           LDA 0,@ZBRTX-1
36 23432 024677           LDA 1,ZBSTD-1
37 23433 122415           SUB# 1,0,SNR
38 23434 002405           JMP @ZBRTX+2
39 23435 000752           JMP ZB1RL
40 23436 023315           ZBDRX+1
41 23437 024041 ZBRTX:    ZBTXR
42 23440 022565           ZB.ND
43 23441 022503           ZB.02
44 23442 122472           @ZB.S3
```

```
!0315 N3MRT
01
02                ;MOVE THE DEFINITIONS TABLE FOR THIS DISK
03                ;INTO THE PROGRAM WORK TABLE AREA
04 23443 175400 ZBGDS:  INC 3,3
05 23444 054415         STA 3,ZBGD3
06 23445 035777         LDA 3,-1,3
07 23446 030414         LDA 2,ZBDPT       ;POINTS TO DRIVE PTR. TABLE
08 23447 113000         ADD 0,2
09 23450 031000         LDA 2,0,2         ;TABLE PTR. THIS DRIVE
10 23451 024417 ZBCDS:  LDA 1,ZBWEL       ;WORK AREA LENGTH
11 23452 021000         LDA 0,0,2
12 23453 041400         STA 0,0,3
13 23454 151400         INC 2,2
14 23455 175400         INC 3,3
15 23456 125404         INC 1,1,SZR
16 23457 000773         JMP .-5           ;MOVE THE WHOLE TABLE
17 23460 002401         JMP @ZBGD3        .
18 23461 000000 ZBGD3:  0                 ;SAVE PC
19 23462 023463 ZBDPT:  .+1
20 23463 024072         ZBDP0             ;DRIVE 0 PTR
21 23464 024133         ZBDP1             ;DRV 1 PTR
22 23465 024174         ZBDP2             ;DRIVE 2 TABLES
23 23466 024235         ZBDP3             ;AND DRIVE 3 TABLES
24 23467 023022 ZBWEP:  ZBLSC             ;LAST SECTOR SIDE
25 23470 177767 ZBWEL:  -ZBDBL            ;DISK DEF. BLOCK LENGTH
26                ;MOVE THE DEFINITIONS FOR THIS DRIVE INTO
27                ;THE DRIVE TABLE SAVING ALL USEABLE INFO
28 23471 054770 ZBSDS:  STA 3,ZBGD3
29 23472 030770         LDA 2,ZBDPT
30 23473 113000         ADD 0,2
31 23474 035000         LDA 3,0,2
32 23475 030402         LDA 2,.+2
33 23476 000753         JMP ZBCDS         ;MOVE DEF. BLK TO DRV TABLE
34 23477 023320         .ZBLSC
```

```
!0316 N3MRT
01
02                ;READ OR WRITE OK NO STAT ERRS
03                ;COMPARE DATA IN BUFFER
04                ;AGAINST WHAT'S SUPPOSED TO BE THERE
05 23500 023043         ZBDW1
06 23501 034777 ZBCOM:  LDA 3,ZBCOM-1     ;ADRS OF DATA WORDS
07 23502 031771         LDA 2,-7,3        ;START ADRS
08 23503 021400         LDA 0,0,3
09 23504 025000         LDA 1,0,2         ;COMPARE
10 23505 122414         SUB# 1,0,SZR      ;FIRST 4 WORDS
11 23506 000503         JMP ZBERR         ;AGAINST EVERY
12 23507 175400         INC 3,3           ;OTHER IN BUFFER
13 23510 021400         LDA 0,0,3
14 23511 025001         LDA 1,1,2
15 23512 122414         SUB# 1,0,SZR
16 23513 000476         JMP ZBERR
17 23514 175400         INC 3,3
18 23515 021400         LDA 0,0,3
19 23516 025002         LDA 1,2,2
20 23517 106414         SUB# 0,1,SZR
21 23520 000471         JMP ZBERR
22 23521 175400         INC 3,3
23 23522 021400         LDA 0,0,3
24 23523 025003         LDA 1,3,2
25 23524 106414         SUB# 0,1,SZR
26 23525 000464         JMP ZBERR
27 23526 035404         LDA 3,4,3         ;#WORDS - 4
28 23527 021000         LDA 0,0,2
29 23530 025004         LDA 1,4,2
30 23531 122414         SUB# 1,0,SZR
31 23532 000457         JMP ZBERR
32 23533 055000         STA 3,0,2         ;CLR BUFFER NXT RD
33 23534 151400         INC 2,2           ;BUMP ADRS
34 23535 175404         INC 3,3,SZR       ;SKP DONEALL
35 23536 000771         JMP .-7
36                      LCALL ARANG       ;BITS 1 AND 15=1
37 23537 100270         ARANG-ASCRA*1B11+100010
38 23540 101232         MOVZR# 0,0,SZC    ;RELEASE THIS
39 23541 103123         ADDZL 0,0,SNC     ;OP TABLE
40 23542 000403         JMP .+3           ;EITHER = 0 KEEP IT
41 23543 102400 ZBERL:  SUB 0,0
42 23544 042444         STA 0,@ZB.SX+2
43 23545 022466         LDA 0,@ZBKNE      ;DRV# POSITIONED FOR COM'S
44 23546 024466         LDA 1,ZBKNF       ;788 FOR RELEASE COMMAND
45 23547 123000         ADD 1,0
46 23550 061327         DOAP 0,ZBDSK
47                      LCALL ARANG
48 23551 100270         ARANG-ASCRA*1B11+100010
49 23552 101102         MOVL 0,0,SZC      ;NEXT 2 BITS =1?
50 23553 101103         MOVL 0,0,SNC      ;IS RELEASE THIS BUFFER
51 23554 002432         JMP @ZB.SX        ;SELECT NEW STRT BUFFER
52 23555 022417 ZB.X3:  LDA 0,@ZBHIK
53 23556 024136         LDA 1,MPSWT
54 23557 125005         MOV 1,1,SNR
55 23560 000416         JMP ZB.XI
56 23561 024414         LDA 1,ZB.37
57 23562 106032         ADCZ# 0,1,SZC     ;SKP IF >=32K
58 23563 000413         JMP ZB.XI
59                ZB.X4:  LCALL RDMAP
60 23564 100610         RDMAP=ASCRA*1B11+100010
```

```
 0317 N3MRT
01 23565 101001          MOV 0,0,SKP
02 23566 000776          JMP ZB.X4
03               ZB.X5:  LCALL RSCRA
04 23567 100050          RSCRA-ASCRA*1B11+100010
05 23570 101001          MOV 0,0,SKP
06 23571 000776          JMP ZB.X5
07 23572 002401          JMP @.+1
08 23573 022506          ZB.02+3
09 23574 001502 ZBHIK:   HIGHK
10 23575 000037 ZB.37:   37
```

```
!0318 N3MRT
01               ZB.XI:  LCALL RDMAP
02 23576 100610          RDMAP-ASCRA*1B11+100010
03 23577 101001          MOV 0,0,SKP
04 23600 000776          JMP ZB.XI
05               ZB.X2:  LCALL RSCRA
06 23601 100050          RSCRA-ASCRA*1B11+100010
07 23602 102401          SUB 0,0,SKP
08 23603 000776          JMP ZB.X2
09 23604 042403          STA 0,@ZB.SX+1
10                       LCALL RETRN
11 23605 100210          RETRN-ASCRA*1B11+100010
12 23606 022533 ZB.SX: ZB.2C
13 23607 022357          ZB.00+2
14 23610 123057          @ZBIDX
15 23611 054466 ZBERR:   STA 3,ZB.SA      ;ERROR IN DATA COMPARE
16                       LCALL ERROI
17 23612 100350          ERROI-ASCRA*1B11+100010
18 23613 000401          JMP .+1
19 23614 020471          LDA 0,ZBTX1
20                       LCALL ERRTX
21 23615 100170          ERRTX-ASCRA*1B11+100010
22 23616 022417          LDA 0,@ZBKNG     ;CURRENT CYL #
23 23617 026463          LDA 1,@ZBDSX
24 23620 032463          LDA 2,@ZBCSX
25                       LCALL ERROC
26 23621 100370          ERROC-ASCRA*1B11+100010
27 23622 000401          JMP .+1
28 23623 020475          LDA 0,ZBTX2
29                       LCALL ERRTX
30 23624 100170          ERRTX-ASCRA*1B11+100010
31 23625 022457          LDA 0,@ZBSTX
32 23626 026452          LDA 1,@ZB.CX
33 23627 032452          LDA 2,@ZB.OX
34                       LCALL ERROC
35 23630 100370          ERROC-ASCRA*1B11+100010
36 23631 000401          JMP .+1          ;RECALIBRATE DRV
37 23632 000502          JMP ZBRCL
38 23633 023051 ZBKNE:   ZBDRV
39 23634 001600 ZBKNF:   788
40 23635 023040 ZBKNG:   ZBCYL
41 23636 000100 ZBKNH:   189
42 23637 000200 ZBKNI:   200
```

```
10319 N3MRT
01 23640 001000          1B6
02 23641 032443 ZBSTE:   LDA 2,@ZBSTX
03 23642 024776          LDA 1,ZBSTE-1    ;GET BIT 6
04 23643 036435          LDA 3,@ZB.CX     ;DOA
05 23644 137405          AND 1,3,SNR      ;SKP=WRITE CMD
06 23645 000404          JMP ZBNBS        ;NOT WRITE CMD,DON'T CK FOR BSF
07 23646 024770          LDA 1,ZBKNH      ;1B9 BAD SECTOR FLAG
08 23647 133414          AND# 1,2,SZR     ;SKP=NOT A  FLAGGED SECTOR
09 23650 000415          JMP ZBBSF        ;TYPE BAD SECTOR FOUND
10 23651 066427 ZBNBS:   DIC 1,ZBDSK
11 23652 020423          LDA 0,ZBDIB
12 23653 034764          LDA 3,ZBKNI      ;1B11 END OF CYL..
13 23654 157424          ANDZ 2,3,SZR     ;SKP=NOT EOC STATUS
14 23655 101040          MOVO 0,0
15 23656 036426          LDA 3,@ZBSTX
16 23657 101003          MOV 0,0,SNC      ;SKP IF ERR WAS EOC
17 23660 000731          JMP ZBERR
18 23661 030415          LDA 2,ZBDIB+1    ;1 SECTOR -4 WORDS
19 23662 034616          LDA 3,ZBCOM-1
20 23663 051407          STA 2,ZB.WK-ZBDW1,3      ;SO COMPARE STOPS
21 23664 000615          JMP ZBCOM
22 23665 020520 ZBBSF:   LDA 0,ZBTX3
23                       LCALL ERRTX      ;BAD SECTOR FLAG ETC.
24 23666 100170          ERRTX-ASCRA*1B11+100010
25 23667 022411          LDA 0,@ZB.CX
26 23670 066427          DIC 1,ZBDSK
27 23671 030404          LDA 2,ZBDIB
28                       LCALL ERROC
29 23672 100370          ERROC-ASCRA*1B11+100010
30 23673 000401          JMP .+1
31 23674 000647          JMP ZBERL        ;RELEASE CTRL TABLE
32 23675 000000 ZBDIB:   0
33 23676 177404          -252.
34 23677 000000 ZB.SA:   0
35 23700 023054 ZB.CX:   ZB.CO
36 23701 023055 ZB.OX:   ZB.OC
37 23702 023034 ZBDSX:   ZBDST
38 23703 023035 ZBCSX:   ZBCST
39 23704 023056 ZBSTX:   ZBSTA
40 23705 023706 ZBTX1:   .+1
41 23706 005215          .TXTE (<15><12>CYL #     ZBDST   ZBCST(
42 23720 023721 ZBTX2:   .+1
43 23721 005215          .TXTE (<15><12>ZBSTA     ZBDOA   ZBDOC(
```

```
10320 N3MRT
01                ;RECALIBRATE THE DRIVE THEN RETRY IF
02                ;LESS THAN 4 ERRS THIS OP TABLE
03 23733 023543          ZBERL
04 23734 020423 ZBRCL:   LDA 0,ZBRCI
05 23735 042652          STA 0,@ZB.SX+1
06 23736 030652          LDA 2,ZB.SX+2    ;ADRS ZBIDX
07 23737 035000          LDA 3,0,2        ;OP. TABLE PTR
08 23740 011402          ISZ 2,3          ;+1 ERR TRY CTR
09 23741 021402          LDA 0,2,3
10 23742 101220          MOVZR 0,0
11 23743 101224          MOVZR 0,0,SZR    ;NO SKP=<4 RETRY'S
12 23744 002767          JMP @ZBRCL-1     ;RELEASE THIS BADDY
13 23745 025351          LDA 1,ZBSKB-ZBIDX,2      ;GET SEEK BIT
14 23746 131220          MOVZR 1,2
15 23747 036731          LDA 3,@ZB.CX     ;GET READ/WRITE DOA
16 23750 137400          AND 1,3          ;SAVE RD/WRT BIT
17 23751 054405          STA 3,ZBRCI-1    ;SAVE IT
18 23752 026661          LDA 1,@ZBKNE     ;
19 23753 133000          ADD 1,2          ;CREATE RECAL INSTR.
20 23754 071327          DOAP 2,ZBDSK     ;START RECAL
21                       LCALL RETRN
22 23755 100210          RETRN-ASCRA*1B11+100010
23                ;PROCESS RECALIBRATE INTERRUPT
24 23756 000000          0        ;SAVE FOR READ OR WRITE BIT
25 23757 123760 ZBRCI:   @.+1
26 23760 024455          LDA 1,ZBKNJ      ;377 TO MASK ERROR BITS
27 23761 123415          AND# 1,0,SNR
28 23762 000403          JMP .+3          ;NO STATUS ERR
29 23763 002453          JMP @ZBKNL       ;SETUP STATUS TYPEOUT
30 23764 024453          LDA 1,ZBKNM      ;76000 TO MASK DONE BITS
31 23765 123400          AND 1,0
32 23766 032645          LDA 2,@ZBKNE     ;DRV# POSITIONED
33 23767 143000          ADD 2,0
34 23770 024450          LDA 1,ZBKNO      ;2B8 RESERVE DRV
35 23771 123000          ADD 1,0
36 23772 061227          DOAC 0,ZBDSK
37 23773 024403          LDA 1,.+3
38 23774 046613          STA  1,@ZB.SX+1
39 23775 001400          JMP 0,3
40 23776 023777          .+1
41                ;DETERMINE IF REDOING A READ OR WRITE
42 23777 020757          LDA 0,ZBRCI-1    ;=0 IS READ
43 24000 101004          MOV 0,0,SZR      ;NOT =0 IS WRITE
44 24001 002402          JMP @.+2         ;REGEN DATA "WRITE"
45 24002 002402          JMP @.+2         ;JUST RE-READ
46 24003 022643          ZB.3G
47 24004 023134          ZB.5G
48 24005 024006 ZBTX3:   .+1
49 24006 005215          .TXTE !<15><12>DEV #27 BAD SECTOR FLAG<15><12>
50 24023 055012 ZBDOA<11>ZBDIC<11>ZBDIB!
51 24035 000377 ZBKNJ:   377
52 24036 023207 ZBKNL:   ZBRSK
53 24037 076000 ZBKNM:   76000
54 24040 000400 ZBKNO:   2B8
```

```
!0321 N3MRT
01 24041 005215 ZBTXR:  .TXTE !<15><12>DEV.#27 6060/61 DISK<15><12>
02 24055 051504 DSK#    #CYLS   #SEC/S  #SURF.!
03              ;DISK DEFINITIONS TABLE 1 FOR EACH DISK
04 24072 000041 ZBDP0:  .BLK ZBDBL+ZBCTL+ZBCTL+ZBCTL
05 24133 000041 ZBDP1:  .BLK ZBDBL+ZBCTL+ZBCTL+ZBCTL
06 24174 000041 ZBDP2:  .BLK ZBDBL+ZBCTL+ZBCTL+ZBCTL
07 24235 000041 ZBDP3:  .BLK ZBDBL+ZBCTL+ZBCTL+ZBCTL
08 24276 000000 ZBDRE:  0       ;END OF DISK TABLES
```

```
!0322 N3MRT
01              ;       .TITL MTTES
02              ;MAGNETIC TAPE TEST COMPATABLE
03              ;WITH THE DIAGNOSTIC LINKER
04
05              ;DEFINITIONS  BLOCK TO LINKER
06                      NEXTT MT.00
07      024277 LMEML=.
08      000170          .LOC LPG0
09 00170 024302         MT.00
10      000171 LPG0=.
11      024277          .LOC LMEML
12 24277 000000         0       ;TEST PASS CTR
13 24300 000000         0       ;TEST ERROR CTR
14 24301 000000         0       ;INTERRUPT TIMEOUT SWITCH
15 24302 024327 MT.00:          MT.01
16 24303 024423                 MT.02
17 24304 000000                 0
18 24305 000000                 0
19 24306 177777                 -1
20 24307 176000                 176000
21 24310 025237                 MT.XI
22 24311 025237                 MT.XI
23      024312                  .TXTE !
24 24312 040515 MAGNETIC TAPE TEST(PRI)!
25      047107
26      152305
27      141711
28      152240
29      050101
30      120305
31      142724
32      152123
33      050050
34      144722
35      000251
```

```
10323 N3MRT
01                ;MAGNETIC TAPE TEST INITIALIZE
02                ;DETERMINE IF AN MTA CONTROL AND
03                ;A DRIVE 0 EXIST
04
05 24326 000004 MTB13:  1B13
06 24327 102000 MT.01:  ADC 0,0
07 24330 062022         DOB 0,MTA
08 24331 065422         DIB 1,MTA         ;READ ADDRS BACK
09 24332 125005         MOV 1,1,SNR       ;SKIP IF MTA CONTROL
10 24333 000415         JMP MT.1A         ;NO MAG TAPE
11 24334 126400         SUB 1,1
12 24335 065022 MT.1X:  DOA 1,MTA         ;DRIV0 TO REGA
13 24336 060022         NIO MTA           ;KILL SOME TIME
14 24337 070422         DIA 2,MTA         ;GET STATUS
15 24340 020766         LDA 0,MTB13
16 24341 113414         AND# 0,2,SZR      ;SKP=NOT WRITE LOCKED
17 24342 000406         JMP MT.1A
18 24343 151232         MOVZR# 2,2,SZC    ;=1 IS READY
19 24344 000413         JMP MT.1B
20 24345 153100         ADDL 2,2
21 24346 151122         MOVZL 2,2,SZC     ;=1 IS REWINDING
22 24347 000410         JMP MT.1B
23 24350 125400 MT.1A:  INC 1,1           ;+1 DRIVE#
24 24351 030446         LDA 2,MT10
25 24352 146414         SUB# 2,1,SZR              ;SKP=TRYED 8
26 24353 000762         JMP MT.1X
27 24354 102000         ADC 0,0
28 24355 040727         STA 0,MT.00+2
29 24356 001400         JMP 0,3
30
```

```
10324 N3MRT
01                ;MAG TAPE CONTROL AND A DRIVE 0 EXIST
02                ;ENABLE TAPE TEST AND ENTER INTERRUPT VECTORS
03 24357 020411 MT.1B:  LDA 0,MT.1C
04 24360 040723         STA 0,MT.00+1     ;TO INIT REWIND
05 24361 126400         SUB 1,1
06 24362 046405         STA 1,@MT.TB
07 24363 020435         LDA 0,MT.K1       ;DEV #
08 24364 024435         LDA 1,MT.K2       ;IO MSK
09 24365 030435         LDA 2,MT.K3       ;INTERRUPT DIRECTOR
10 24366 002101         JMP @EINTS        ;ENTER INT VECTORS
11 24367 024603 MT.TB:  MTCTB
12 24370 024371 MT.1C:  .+1
13 24371 102400         SUB 0,0
14 24372 061022         DOA 0,MTA
15 24373 060022         NIO MTA
16 24374 070422         DIA 2,MTA
17 24375 151233         MOVZR# 2,2,SNC    ;SKP=DRV READY
18 24376 000407         JMP MT1CX         ;NOT RDY, DO NEXT
19 24377 024416         LDA 1,MTMSK
20 24400 133414         AND# 1,2,SZR      ;TAPE REWINDING OR AT LD PT
21 24401 000404         JMP MT1CX                 ;YES
22 24402 024415         LDA 1,MT10
23 24403 107000         ADD 0,1           ;CRREATE REWIND
24 24404 065122         DOAS 1,MTA
25 24405 101400 MT1CX:  INC 0,0
26 24406 024411         LDA 1,MT10
27 24407 122414         SUB# 1,0,SZR      ;SKP=TRIED 8
28 24410 000762         JMP MT.1C+2       ;DO NEXT DRIVE
29 24411 060222         NIOC MTA
30 24412 020404         LDA 0,MTMSK+1
31 24413 040670         STA 0,MT.00+1
32 24414 000407         JMP MT.02
33 24415 020200 MTMSK:  1B2+1B8           ;REWIND+LD. PT BITS
34 24416 024423         MT.02
35 24417 000010 MT10:   10
36 24420 000022 MT.K1:  MTA
37 24421 000077 MT.K2:  77
38 24422 024715 MT.K3:  MT.ID
```

```
10325 N3MRT
01
02                ;EXECUTE ENTRY POINT
03                ;MT.00+2=ADDRESS PROCESS OR=0 NO SCRATCH
04
05 24423 030661 MT.02:  LDA 2,MT.00+2    ;GET INTR SW
06 24424 151004         MOV 2,2,SZR      ;SKIP=NO SCRATCH
07 24425 001000         JMP 0,2          ;GO TO SEVICE
08                      LCALL ASCRA      ;GET 1K SCR
09 24426 100010         ASCRA-ASCRA*1B11+100010
10 24427 002445         JMP @MT.XX       ;NO SCRATCH AVAIL
11                      LCALL ADMAP      ;ASSIGN IT TO DCH
12 24430 100550         ADMAP-ASCRA*1B11+100010
13 24431 002443         JMP @MT.XX       ;NO DCH AVAIL
14                      LCALL ARANG
15 24432 100270         ARANG-ASCRA*1B11+100010
16 24433 105000         MOV 0,1
17 24434 030442         LDA 2,MT.6
18 24435 102400         SUB 0,0
19 24436 040537         STA 0,MT.KK      ;0 # 1K'S EXTRA
20                      LCALL ADIVI
21 24437 100310         ADIVI-ASCRA*1B11+100010
22 24440 100405         NEG 0,0,SNR      ;(0)=# 1K'S TO EXPAND
23 24441 000410         JMP MT.2C
24                MT.2L:  LCALL ESCRA      ;TRY TO GET 1K
25 24442 100030         ESCRA-ASCRA*1B11+100010
26 24443 000406         JMP MT.2C        ;NO MORE AVAZC
27                      LCALL EDMAP      ;ALSO EXPND DCH 1K
28 24444 100570         EDMAP-ASCRA*1B11+100010
29 24445 000424         JMP MT.2R        ;CANT EXPAND DCH
30 24446 010527         ISZ MT.KK        ;#1 #1K'S ASSIGNED
31 24447 101404         INC 0,0,SZR
32 24450 000772         JMP MT.2L
```

```
10326 N3MRT
01                ;SOME AMOUNT (1K TO 6K) SCRATCH AND DCH
02                ;ARE ASSIGNED TO THIS TEST
03                ;PICK START BUFFER WITHIN FIRST 400 WORDS
04
05                MT.2C:  LCALL ARANG
06 24451 100270         ARANG-ASCRA*1B11+100010
07 24452 105000         MOV 0,1
08 24453 030424         LDA 2,MT256
09                      LCALL ADIVI
10 24454 100310         ADIVI-ASCRA*1B11+100010
11 24455 024152         LDA 1,SCRLO
12 24456 107000         ADD 0,1
13 24457 044521         STA 1,MTDST      ;DATA START ADDRESS
14 24460 024154         LDA 1,DCHLO
15 24461 107000         ADD 0,1
16 24462 044517         STA 1,MTCST      ;CHANNEL START ADDRESS
17 24463 024512         LDA 1,MT.KK      ;# 1K'S AVAIL
18 24464 127120         ADDZL 1,1        ;*4
19 24465 020410         LDA 0,MT.3
20 24466 107000         ADD 0,1          ;AC1=# RECORDS THAT FIT
21 24467 044507         STA 1,MT.BK      ;INTO AVAILABLE BUFFER
22 24470 000412         JMP MT.03        ;DETERMINE NEXT OP
23
24                ;RELEASE 1K THAT COULDNT BE ASSIGNED TO DCH
25
26                MT.2R:  LCALL RSCRA
27 24471 100050         RSCRA-ASCRA*1B11+100010
28 24472 063077         HALT             ;NOT PROBABLE HALT
29 24473 000756         JMP MT.2C
30
31 24474 025237 MT.XX:  MT.XI
32 24475 000003 MT.3:   3
33 24476 000006 MT.6:   6
34 24477 000400 MT256:  256.
35 24500 000000 MTDKX:  0        ;TRY FIND CTR
```

```
!0327 N3MRT
01                ;DATA CHANNEL AND SCRATCH ARE ASSIGNED
02                ;DETERMINE WHAT TO DO WITH DRIVE
03
04 24501 000004 MTX13:  1B13
05 24502 022472 MT.03:  LDA 0,@MT.TT
06 24503 100644         NEGOR 0,0,SZR    ;SKIP=FIND NEW DRIVE
07 24504 000425         JMP MT.3A
08 24505 030772         LDA 2,MT256
09 24506 050772         STA 2,MTDKX
10 24507 030710         LDA 2,MT10
11                      LCALL ARANG      ;GET RND #
12 24510 100270         ARANG-ASCRA*1B11+100010
13 24511 105000         MOV 0,1          ;FOR DIVIDE
14                      LCALL ADIVI
15 24512 100310         ADIVI-ASCRA*1B11+100010
16 24513 061022         DOA 0,MTA
17 24514 060022         NIO MTA          ;KILL SOME TIME
18 24515 064422         DIA 1,MTA
19 24516 125233         MOVZR# 1,1,SNC   ;15=1 IS DRV RDY
20 24517 000404         JMP .+4          ;DRIVE NOT READY
21 24520 030761         LDA 2,MTX13      ;WRITE LOCK BIT
22 24521 147405         AND 2,1,SNR      ;SKP IS WRITE LOCKED
23 24522 000404         JMP MT.3A-3      ;OK TO WRITE
24 24523 014755         DSZ MTDKX        ;CANT USE DRV
25 24524 000763         JMP MT.03+5      ;TRY ANOTHER
26 24525 002747         JMP @MT.XX       ;STILL REWINDING EXIT
27 24526 101040         MOVO 0,0         ;CRY=1 IS WRITE
28 24527 040562         STA 0,MTDRV      ;SAVE DRIVE#
29 24530 012444         ISZ @MT.TT       ;SET WRITE STATE
```

```
!0328 N3MRT
01 24531 101003 MT.3A:  MOV 0,0,SNC      ;WRITE SELECT CARRY=1
02 24532 000515         JMP MT.3R        ;TAPE IS IN READ MODE
03                      LCALL ARANG
04 24533 100270         ARANG-ASCRA*1B11+100010
05 24534 105000         MOV 0,1
06 24535 030507         LDA 2,MT64.
07 24536 020440         LDA 0,MT.BK
08 24537 112400         SUB 0,2
09                      LCALL ADIVI      ;WRITE AT LEAST
10 24540 100310         ADIVI-ASCRA*1B11+100010
11 24541 024435         LDA 1,MT.BK      ;ONE BUFFER FULL
12 24542 123000         ADD 1,0
13 24543 030431         LDA 2,MT.TT
14 24544 041002         STA 0,2,2        ;# BLOCKS TO WRITE
15 24545 041001         STA 0,1,2        ;# BLOCKS BKSP/READ
16 24546 125300         MOVS 1,1         ;# BLOCKS *400
17 24547 124600         NEGR 1,1         ;-FOR WC
18 24550 125400         INC 1,1          ;TAKE OFF 4
19 24551 125500         INCL 1,1
20 24552 044425         STA 1,MT.BL      ;BUFFER LENGTH
21                      LCALL FRANG
22 24553 100450         FRANG-ASCRA*1B11+100010
23 24554 040432         STA 0,MTDW1      ;GET 4 RAN#'S
24 24555 044432         STA 1,MTDW2      ;FOR DATA
25 24556 050432         STA 2,MTDW3
26                      LCALL ARANG
27 24557 100270         ARANG-ASCRA*1B11+100010
28 24560 040431         STA 0,MTDW4
29 24561 030417         LDA 2,MTDST
30 24562 034420         LDA 3,MTDWX
31 24563 126120         ADCZL 1,1
32 24564 125120         MOVZL 1,1        ;1=-4
33 24565 021400         LDA 0,0,3        ;MOV TO FIRST 4 WDS
34 24566 041000         STA 0,0,2        ;IN DATA BUFFER
35 24567 175400         INC 3,3
36 24570 151400         INC 2,2
37 24571 125404         INC 1,1,SZR
38 24572 000773         JMP .-5
39 24573 000421         JMP MTFIL
```

```
!0329 N3MRT
01
02 24574 024603 MT.TT: MTCTB
03 24575 000000 MT.KK: 0          ;# 1K'S ASSIGNED
04 24576 000000 MT.BK: 0          ;# TAPE BLOCKS AVAILABLE
05 24577 000000 MT.BL: 0          ;# WORDS IN BUFFER
06 24600 000000 MTDST: 0          ;DATA START ADDRESS
07 24601 000000 MTCST: 0          ;SAME BUT FOR DCH
08                ;TEST CONTROL TABLE
09 24602 024606 MTDWX: MTDW1
10 24603 000000 MTCTB: 0          ;MODE CONTROL REG
11 24604 000000          0        ;WRITE RECORD CTR
12 24605 000000          0        ;READ BKSPA CTR
13 24606 000000 MTDW1: 0
14 24607 000000 MTDW2: 0
15 24610 000000 MTDW3: 0
16 24611 000000 MTDW4: 0
17 24612 000000 MT.CA: 0          ;CURRENT BLOCK ADRS
18 24613 000000 MT.RK: 0          ;# RECORDS IN BUFFER
```

```
!0330 N3MRT
01                ;FIRST 4 WORDS IN BUFFER ARE FILLED
02                ;MOVE THEM TO REST OF BUFFER
03                ;AND START WRITING
04 24614 024763 MTFIL:  LDA 1,MT.BL
05 24615 021374         LDA 0,-4,2
06 24616 041000         STA 0,0,2
07 24617 151400         INC 2,2
08 24620 125404         INC 1,1,SZR
09 24621 000774         JMP .-4
10 24622 044470         STA 1,MTDRV+1
11 24623 020543         LDA 0,MTMK1
12 24624 063022         DOC 0,MTA         ;LOAD WC
13 24625 024754         LDA 1,MTCST       ;CHANNEL START
14 24626 066022         DOB 1,MTA         ;LOAD CA
15 24627 044464         STA 1,MTDRV+2
16 24630 030503         LDA 2,MT.WI
17 24631 052571         STA 2,@MT00X      ;WRITE INT PROCESSOR
18 24632 030413         LDA 2,MT50        ;MT WRITE COMMAND
19 24633 020456         LDA 0,MTDRV       ;DRIVE #
20 24634 113000         ADD 0,2           ;CREATE WRITE COMMAND
21 24635 020741         LDA 0,MT.BK
22 24636 040755         STA 0,MT.RK       ;# RECORDS 1 PASS BUFFER
23 24637 102000         ADC 0,0           ;SET BKSPACE SW
24 24640 040450         STA 0,MT8SW
25 24641 044751         STA 1,MT.CA       ;START RECORD ADRS
26 24642 071122         DOAS 2,MTA
27                      LCALL RETRN
28 24643 100210         RETRN-ASCRA*1B11+100010
29 24644 000100 MT64.:  64.
30 24645 000050 MT50:   50
31 24646 000000 MT.EK: 0          ;READ RETRY CTR
```

```
!0331 N3MRT
01                ;DRIVE IS IN READ STATE
02                ;READ AS MANY RECORDS AS POSSIBLE
03                ;INTO THE DATA BUFFER
04                ;AFTER BACKSPACING OVER ALL WRITTEN
05 24647 030725 MT.3R:  LDA 2,MT.TT
06 24650 025002         LDA 1,2,2
07 24651 020725         LDA 0,MT.BK
08 24652 122432         SUBZ# 1,0,SZC
09 24653 121000         MOV 1,0          ;(0)=#RECORDS TO READ
10 24654 040722         STA 0,MT.BK
11 24655 040736         STA 0,MT.RK
12 24656 020723         LDA 0,MTCST
13 24657 040733         STA 0,MT.CA
14 24660 062022         DOB 0,MTA        ;LOAD ADDRESS
15 24661 025002         LDA 1,2,2
16 24662 124400         NEG 1,1
17 24663 067022         DOC 1,MTA        ;WORD COUNT
18 24664 030522         LDA 2,MT.WB
19 24665 020423         LDA 0,MTBSW
20 24666 100044         COMO 0,0,SZR
21 24667 000414         JMP MTRDG
22 24670 052532         STA 2,@MT00X     ;BKSPA INTERRUPT HANDLER
23 24671 040417         STA 0,MTBSW
24 24672 042415         STA 0,@MTBSW-1   ;=MTRDS
25 24673 101500         INCL 0,0
26 24674 010416         ISZ MTDRV+1      ;SKP = RETRY
27 24675 040751         STA 0,MT.EK      ;3 FOR RETRY CTR
28 24676 020467         LDA 0,MT40       ;BKSPA DRV 0
29 24677 024412         LDA 1,MTDRV
30 24700 123000         ADD 1,0          ;CREATE BKSPA THIS DRV
31 24701 061122         DOAS 0,MTA       ;START BKSPA
32                      LCALL RETRN
33 24702 100210         RETRN-ASCRA*1B11+100010
34 24703 004505 MTRDG:  JSR MT.WB+2
35 24704 152400         SUB 2,2
36 24705 050564         STA 2,MTRDS
37                      LCALL RETRN
38 24706 100210         RETRN-ASCRA*1B11+100010
39 24707 025071         MTRDS
40 24710 000000 MTBSW:  0
41 24711 000000 MTDRV:  0        ;CURRENT DRIVE#
42 24712 000000         0        ;-1 = RETRY ON READ DATA ERRORS
43 24713 000000         0        ;CA START FOR WRITE OPERATIONS
```

```
!0332 N3MRT
01                ;INTERRUPT DISPATCHER
02 24714 000022         MTA
03 24715 032505 MT.ID:  LDA 2,@MT00X
04 24716 151132         MOVZL# 2,2,SZC
05 24717 000405         JMP .+5
06 24720 020774         LDA 0,MT.ID-1
07 24721 040144         STA 0,UDEVI
08 24722 060222         NIOC MTA
09 24723 001400         JMP 0,3
10 24724 060622         DIAC 0,MTA
11 24725 040442         STA 0,MTSTA
12 24726 101133         MOVZL# 0,0,SNC   ;SKIP IS ERR FLAG
13 24727 001001         JMP 1,2          ;PROCESS NORML RET
14 24730 103132         ADDZL# 0,0,SZC   ;IS IT DATA LATE?
15 24731 000517         JMP MT.RE+2      ;TYPE ALL DATA LATES
16 24732 001000         JMP 0,2          ;PROCESS ERROR
17
18                ;WRITE INTERRUPT SERVICE ROUTINE
19 24733 124734 MT.WI:  @.+1
20 24734 000435         JMP MT.WE
21 24735 016433         DSZ @MTWTX       ;-1 RECORDS TO WRITE
22 24736 000404         JMP MT.WM        ;MORE TO DO
23 24737 020535         LDA 0,MT.04      ;SET UP BACKSPACE
24 24740 042462         STA 0,@MT00X     ;COMPARE WRITE BUFFER
25 24741 001400         JMP 0,3
26                ;CONTINUE WRITING RECORDS
27 24742 020650 MT.WM:  LDA 0,MT.CA
28 24743 101300         MOVS 0,0
29 24744 101700         INCS 0,0         ;+400 TO ADRS
30 24745 040645         STA 0,MT.CA
31 24746 024633         LDA 1,MTCST
32 24747 014644         DSZ MT.RK        ;SKIP DONE FULL BUFF
33 24750 000405         JMP .+5          ;NOT AT EOB YET
34 24751 044641         STA 1,MT.CA      ;RESET ADDRS TO START
35 24752 121000         MOV 1,0
36 24753 024623         LDA 1,MT.BK      ;#BLOCKS IN BUFFER
37 24754 044637         STA 1,MT.RK
38 24755 062022 MT.WN:  DOB 0,MTA        ;LOAD CA
39 24756 024410         LDA 1,MTMK1
40 24757 067022         DOC 1,MTA        ;-256 WK
41 24760 030665         LDA 2,MT50       ;WRITE COM
42 24761 024730         LDA 1,MTDRV
43 24762 133000         ADD 1,2          ;CREATE WRITE+DRV#
44 24763 071122         DOAS 2,MTA       ;GO AGAIN
45 24764 001400         JMP 0,3
46 24765 000040 MT40: 40
47 24766 177400 MTMK1: -256.
48 24767 000000 MTSTA: 0
49 24770 024604 MTWTX: MTCTB+1
50
```

```
!0333 N3MRT
01                ;ERROR STATUS DURING WRITE
02                ;BACK UP AND TRY AGAIN IF NOT EOT
03                ;REWIND RESTART IF EOT
04
05 24771 105300 MT.WE:  MOVS 0,1
06 24772 125200         MOVR 1,1
07 24773 125202         MOVR 1,1,SZC     ;SKIP NOT EOT
08 24774 000457         JMP MTREW        ;=EOT REWIND
09 24775 102000 MT8K1:  ADC 0,0
10 24776 063022         DOC 0,MTA                ;-1 TO WC
11 24777 020407         LDA 0,MT.WB
12 25000 042422         STA 0,@MT00X
13 25001 030764         LDA 2,MT40       ;SPACE REVRS
14 25002 024707         LDA 1,MTDRV
15 25003 133000         ADD 1,2
16 25004 071122         DOAS 2,MTA
17 25005 001400         JMP 0,3
18                ;PROCESS BACKSPACE COMPLETE INTERRUPTS
19                ;BACKSPACE OVER ALL WRITTEN RESTART READ
20                ;BACKSPACE 1 RECORD  COMPLETE-RESTART WRITE/READ
21 25006 125007 MT.WB:  @.+1
22 25007 000441         JMP MT.RE+2      ;BACKSP ERR REWIND
23 25010 022413         LDA 0,@MT.EE     ;MT.CA
24 25011 024722         LDA 1,MT.WI
25 25012 046410         STA 1,@MT00X     ;SET UP WRITE INTR
26 25013 026406         LDA 1,@MT.88
27 25014 125222         MOVZR 1,1,SZC    ;SKP IS READ MODE
28 25015 000740         JMP MT.WN        ;WRITING
29 25016 024406         LDA 1,MT.RI      ;SET UP READ INTR
30 25017 046403         STA 1,@MT00X
31 25020 000420         JMP MT.RC
32 25021 024603 MT.88:  MTCTB
33 25022 024304 MT00X: MT.00+2
```

```
!0334 N3MRT
01                ;READ INTERRUPT PROCESSOR
02
03 25023 024612 MT.EE:  MT.CA
04 25024 125025 MT.RI:  @.+1
05 25025 000421         JMP MT.RE        ;READ ERROR STATUS
06 25026 010443         ISZ MTRDS        ;+1 #BLKS RD
07 25027 022774         LDA 0,@MT.EE
08 25030 101300         MOVS 0,0         ;CA + 400
09 25031 101700         INCS 0,0
10 25032 042771         STA 0,@MT.EE
11 25033 016440         DSZ @MT.RX       ;READ TO FILL BUFFER
12 25034 000404         JMP MT.RC        ;NO READ 1 MORE
13 25035 024507         LDA 1,MT.05
14 25036 046764         STA 1,@MT00X     ;COMPARE BUFFER
15 25037 001400         JMP 0,3          ;NEXT NORMAL ENTRY
16 25040 062022 MT.RC:  DOB 0,MTA
17 25041 024725         LDA 1,MTMK1      ;READ
18 25042 067022         DOC 1,MTA        ;NEXT BLOCK
19 25043 030646         LDA 2,MTDRV              ;IN SEQUENCE
20 25044 071122         DOAS 2,MTA
21 25045 001400         JMP 0,3
22
23 25046 016424 MT.RE:  DSZ @MT.KX       ;RE-READ 3 TRYS
24 25047 000726         JMP MT8K1        ;NO TRY AGAIN
25 25050 020413         LDA 0,MT.NR      ;NON REC READ TYPEOUT
26 25051 042751         STA 0,@MT00X     ;NEXT NORMAL ENTRY
27 25052 001400         JMP 0,3
28
29
30                ;REWIND DRIVE CLEAR MODE
31
32 25053 020407 MTREW:  LDA 0,MT.10      ;REW
33 25054 024635         LDA 1,MTDRV
34 25055 123000         ADD 1,0          ;CREATE REWIND
35 25056 061122         DOAS 0,MTA       ;TO CONTROL/DRV 0
36 25057 102400         SUB 0,0
37 25060 042406         STA 0,@MTTTX     ;SET TAPE REWINDING
38 25061 000770         JMP MTREW-2
39 25062 000010 MT.10: 10
40 25063 025173 MT.NR: MTDER
41 25064 024606 MT.DX: MTDW1
42 25065 024612         MTDW4+1
43 25066 024603 MTTTX: MTCTB
44 25067 024600 MTDSX: MTDST
45 25070 024576 MT8KX: MT.8K
46 25071 000000 MTRDS: 0         ;# BLOCKS READ
47 25072 024646 MT.KX:  MT.EK
48 25073 024613 MT.RX:  MT.RK
```

```
10335 N3MRT
01                ;WRITE PASS COMPLETE RECHECK DATA FOR VALIDITY
02                ;READ PASS ENTERS AFTER CHECKING FOR 7 TRACK
03 25074 025075 MT.04:  .+1
04 25075 034767         LDA 3,MT.DX       ;START OF DATA RAN#'S
05 25076 032771         LDA 2,@MTDSX      ;START OF BUFFER
06 25077 021400         LDA 0,0,3         ;GET NEXT OF 4
07 25100 025000         LDA 1,0,2         ;GET ONE OF FIRST 4
08 25101 122414         SUB# 1,0,SZR      ;SHD BE=
09 25102 000471         JMP MTDER         ;ERROR IN ONE OF 4
10 25103 151400         INC 2,2           ;BUMP ADRS
11 25104 175400         INC 3,3
12 25105 020760         LDA 0,MT.DX+1
13 25106 116414         SUB# 0,3,SZR      ;SKP IS DONE FIRST 4
14 25107 000770         JMP MT.04+3
15 25110 036760         LDA 3,@MTBKX      ;#400 WORD BLOCKS
16 25111 175300         MOVS 3,3          ;*400
17 25112 174600         NEGR 3,3
18 25113 175400         INC 3,3           ;4 LESS COMPARES
19 25114 175500         INCL 3,3          ;ARE NEEDED
20 25115 021374         LDA 0,-4,2        ;GET WORD VALIDATED
21 25116 025000         LDA 1,0,2         ;GET WORD NOT VALIDATED
22 25117 106414         SUB# 0,1,SZR      ;SKP=
23 25120 000453         JMP MTDER         ;DATA ERROR
24 25121 055374         STA 3,-4,2        ;CLR BUFFER FOR NXT RD
25 25122 151400         INC 2,2           ;STP ADRS
26 25123 175404         INC 3,3,SZR       ;SKP DONE ALL
27 25124 000771         JMP .-7
28 25125 022741         LDA 0,@MTTTX      ;GET MODE SW
29 25126 101223         MOVZR 0,0,SNC     ;SKP=CHNG WRITE TO RD
30 25127 000403         JMP .+3
31 25130 012736         ISZ @MTTTX
32 25131 002412         JMP @MTFLX
33 25132 020737         LDA 0,MTRDS
34 25133 026407         LDA 1,@MTRDX      ;# BLKS TO RD
35 25134 106420         SUBZ 0,1          ;-#BLKS RD
36 25135 046405         STA 1,@MTRDX      ;REM BLKS TO RD
37 25136 125004         MOV 1,1,SZR       ;NONE LEFT RELEASE BUFFER
38 25137 002404         JMP @MTFLX        ;RESELECT BUFFER START
39 25140 046726         STA 1,@MTTTX      ;SET TO SELECT NEW DRIVE
40 25141 000476         JMP MT.XI
41 25142 024605 MTRDX: MTCTB+2
42 25143 024451 MTFLX: MT.2C     ;RESELECT IN FIRST 256
```

```
10336 N3MRT
01                ;READ COMPLETE RE-ENTER TAPE TEST
02 25144 025145 MT.05:  .+1               ;IF 7 TRACK DRIVE
03 25145 020724         LDA 0,MTRDS
04 25146 042722         STA 0,@MTBKX      ;#BLKS TO COMPARE
05 25147 022421         LDA 0,@MT.RV
06 25150 061022         DOA 0,MTA
07 25151 060422         DIA 0,MTA         ;CLR UNUSED BITS
08 25152 101300         MOVS 0,0
09 25153 103122         ADDZL 0,0,SZC     ;SKP IF 7 TRACK
10 25154 000721         JMP MT.04+1       ;9 TRACK COMP WD FOR WD
11 25155 030707         LDA 2,MT.DX       ;START 4 RAN#'S
12 25156 034707         LDA 3,MT.DX+1     ;END OF SAME
13 25157 024410         LDA 1,MT7TM       ;7 TRACK DATAMASK
14 25160 021000         LDA 0,0,2
15 25161 123400         AND 1,0           ;CLR ANY EXTRA BITS
16 25162 041000         STA 0,0,2
17 25163 151400         INC 2,2
18 25164 156414         SUB# 2,3,SZR
19 25165 000773         JMP .-5
20 25166 000707         JMP MT.04+1       ;NOW COMPARE
21 25167 037477 MT7TM: 37477
22 25170 024711 MT.RV:  MTDRV
```

```
!0337 N3MRT
01                ;TAPE DATA ERROR START TYPEOUTS
02
03 25171 125022           @MT00X
04 25172 024767           MTSTA
05                MTDER:  LCALL ERROI
06 25173 100350           ERROI-ASCRA*1B11+100010
07 25174 000401           JMP .+1
08 25175 020474           LDA 0,MTTX2
09                        LCALL ERRTX
10 25176 100170           ERRTX-ASCRA*1B11+100010
11 25177 020672           LDA 0,MTRDS
12 25200 026454           LDA 1,@MTEKX+5
13 25201 032454           LDA 2,@MTEKX+6
14                        LCALL ERROC
15 25202 100370           ERROC-ASCRA*1B11+100010
16 25203 000401           JMP .+1
17 25204 020452           LDA 0,MTTX1
18                        LCALL ERRTX
19 25205 100170           ERRTX-ASCRA*1B11+100010
20 25206 022660           LDA 0,@MTTTX    ;GET MODE,1=WRITE,2=READ
21 25207 026441           LDA 1,@MTEKX+1  ;GET DRIVE #
22 25210 127300           ADDS 1,1        ;TO BITS 4,5,6
23 25211 123000           ADD 1,0         ;FOR TYPE OUT
24 25212 026760           LDA 1,@MTDER-1  ;GET LAST STATUS
25 25213 032434           LDA 2,@MTEKX    ;GET # RETRYS
26                        LCALL ERROC
27 25214 100370           ERROC-ASCRA*1B11+100010
28 25215 000401           JMP .+1
29 25216 022754           LDA 0,@MTDER-1  ;GET STATUS
30 25217 101102           MOVL 0,0,SZC    ;SKP=NOT STATUS ERROR
31 25220 000416           JMP MT.XI-1
32 25221 020650           LDA 0,MTRDS     ;# BLKS READ
33 25222 101005           MOV 0,0,SNR     ;SKP IS READ SOME
34 25223 000413           JMP MT.XI-1
35 25224 026423           LDA 1,@MTEKX    ;GET RETRY CTR
36 25225 125005           MOV 1,1,SNR     ;SKP = NOT RETRYED YET
37 25226 000410           JMP MT.XI-1
38 25227 016420           DSZ @MTEKX      ;
39 25230 042712           STA 0,@MTRDX    ;SET # TO BACKUP/READ
40 25231 102000           ADC 0,0
41 25232 042417           STA 0,@MTEKX+2  ;SET BACKUP SWITCH
42 25233 042417           STA 0,@MTEKX+3  ;SKP RETRT CTR RELOAD
43 25234 002417           JMP @MTEKX+4    ;BACKUP/RE-READ
```

```
!0338 N3MRT
01 25235 025053           MTREW
02 25236 006777           JSR @.-1        ;BACK TRY AGAIN
03                MT.XI:  LCALL RDMAP     ;RELEASE ALL DCH
04 25237 100610           RDMAP-ASCRA*1B11+100010
05 25240 102401           SUB 0,0,SKP
06 25241 000776           JMP MT.XI
07                MT.X2:  LCALL RSCRA     ;RELEASE ALL SCR
08 25242 100050           RSCRA-ASCRA*1B11+100010
09 25243 101001           MOV 0,0,SKP
10 25244 000776           JMP MT.X2
11 25245 042724           STA 0,@MTDER-2  ;CLR SCR ASSIGNED
12                        LCALL RETRN
13 25246 100210           RETRN-ASCRA*1B11+100010
14
15 25247 024646 MTEKX: MT.EK
16 25250 024711           MTDRV
17 25251 024710           MTBSW
18 25252 024712           MTDRV+1
19 25253 024647           MT.3R
20 25254 024713           MTDRV+2
21 25255 024601           MTCST
22 25256 025257 MTTX1: .+1
23 25257 005215           .TXTE (<15><12>
24 25260 147515 MODE      STATUS  MT.EK(
25 25271 025272 MTTX2:    .+1
26 25272 005215           .TXTE !<15><12>#READ<11>W/DOB<11>LAST/DOB!
```

```
!0339 N3MRT
01                      ;       .TITL CATES
02                      ;CASSETTE TAPE TEST COMPATABLE
03                      ;WITH THE DIAGNOSTIC LINKER
04
05                      ;DEFINITIONS  BLOCK TO LINKER
06              000034 CAA=34   ;DEFINE DEVICE CODE
07                              NEXTT CA.00
08              025306 LMEML=.
09              000171          .LOC LPG0
10 00171        025311          CA.00
11              000172 LPG0=.
12              025306          .LOC LMEML
13 25306        000000          0       ;TEST PASS CTR
14 25307        000000          0       ;TEST ERROR CTR
15 25310        000000          0       ;INTERRUPT TIMEOUT SWITCH
16 25311        025335 CA.00:           CA.01
17 25312        025432                  CA.02
18 25313        000000                  0
19 25314        000000                  0
20 25315        177777                  -1
21 25316        176000                  176000
22 25317        026255                  CA.XI
23 25320        026255                  CA.XI
24              025321                  .TXTE !
25 25321        040703 CASSETTE TAPE TEST(PRI)!
26              051523
27              152305
28              142724
29              152240
30              050101
31              120305
32              142724
33              152123
34              050050
35              144722
36              000251
```

```
!0340 N3MRT
01                      ;MAGNETIC TAPE TEST INITIALIZE
02                      ;DETERMINE IF AN CAA CONTROL AND
03                      ;A DRIVE 0 EXIST
04
05 25335        102000 CA.01:   ADC 0,0
06 25336        062034          DOB 0,CAA
07 25337        065434          DIB 1,CAA       ;READ ADDRS BACK
08 25340        125005          MOV 1,1,SNR     ;SKIP IF CAA CONTROL
09 25341        000415          JMP CA.1A       ;NO MAG TAPE
10 25342        126400          SUB 1,1
11 25343        065034 CA.1X:   DOA 1,CAA       ;DRIVO TO REGA
12 25344        060034          NIO CAA         ;KILL SOME TIME
13 25345        070434          DIA 2,CAA       ;GET STATUS
14 25346        020463          LDA 0,CA.K3+1
15 25347        113414          AND# 0,2,SZR    ;SKP=NOT WRITE LOCKED
16 25350        000406          JMP CA.1A
17 25351        151232          MOVZR# 2,2,SZC  ;=1 IS READY
18 25352        000413          JMP CA.1B
19 25353        153100          ADDL 2,2
20 25354        151122          MOVZL 2,2,SZC   ;=1 IS REWINDING
21 25355        000410          JMP CA.1B
22 25356        125400 CA.1A:   INC 1,1         ;+1 DRIVE#
23 25357        030446          LDA 2,CA10
24 25360        146414          SUB# 2,1,SZR            ;SKP=TRYED 8
25 25361        000762          JMP CA.1X
26 25362        102000          ADC 0,0
27 25363        040730          STA 0,CA.00+2
28 25364        001400          JMP 0,3
```

```
!0341 N3MRT
01                ;MAG TAPE CONTROL AND A DRIVE 0 EXIST
02                ;ENABLE TAPE TEST AND ENTER INTERRUPT VECTORS
03 25365 020410 CA.1B:  LDA 0,CA.1C
04 25366 040724         STA 0,CA.00+1    ;TO INIT REWIND
05 25367 126400         SUB 1,1
06 25370 046434         STA 1,@CAMSK+2
07 25371 020435         LDA 0,CA.K1      ;DEV #
08 25372 024435         LDA 1,CA.K2      ;IO MSK
09 25373 030435         LDA 2,CA.K3      ;INTERRUPT DIRECTOR
10 25374 002101         JMP @EINTS       ;ENTER INT VECTORS
11 25375 025376 CA.1C:  .+1
12 25376 102400         SUB 0,0
13 25377 061034         DOA 0,CAA
14 25400 060034         NIO CAA
15 25401 070434         DIA 2,CAA
16 25402 151233         MOVZR# 2,2,SNC   ;SKP=DRV READY
17 25403 000407         JMP CA1CX        ;NOT RDY, DO NEXT
18 25404 024416         LDA 1,CAMSK
19 25405 133414         AND# 1,2,SZR     ;TAPE REWINDING OR AT LD PT
20 25406 000404         JMP CA1CX                ;YES
21 25407 024416         LDA 1,CA10
22 25410 107000         ADD 0,1          ;CRREATE REWIND
23 25411 065134         DOAS 1,CAA
24 25412 101400 CA1CX:  INC 0,0
25 25413 024412         LDA 1,CA10
26 25414 122414         SUB# 1,0,SZR     ;SKP=TRIED 8
27 25415 000762         JMP CA.1C+2      ;DO NEXT DRIVE
28 25416 060234         NIOC CAA
29 25417 020404         LDA 0,CAMSK+1
30 25420 040672         STA 0,CA.00+1
31 25421 000411         JMP CA.02
32 25422 020200 CAMSK:  1B2+1B8          ;REWIND+LD. PT BITS
33 25423 025432         CA.02
34 25424 025611         CACTB
35
36 25425 000010 CA10:   10
37 25426 000034 CA.K1:  CAA
38 25427 000077 CA.K2:  77
39 25430 025723 CA.K3:  CA.ID
40 25431 000004         1B13
```

```
!0342 N3MRT
01
02                ;EXECUTE ENTRY POINT
03                ;CA.00+2=ADDRESS PROCESS OR=0 NO SCRATCH
04
05 25432 030661 CA.02:  LDA 2,CA.00+2    ;GET INTR SW
06 25433 151004         MOV 2,2,SZR      ;SKIP=NO SCRATCH
07 25434 001000         JMP 0,2          ;GO TO SEVICE
08                      LCALL ASCRA      ;GET 1K SCR
09 25435 100010         ASCRA-ASCRA*1B11+100010
10 25436 002445         JMP @CA.XX       ;NO SCRATCH AVAIL
11                      LCALL ADMAP      ;ASSIGN IT TO DCH
12 25437 100550         ADMAP-ASCRA*1B11+100010
13 25440 002443         JMP @CA.XX       ;NO DCH AVAIL
14                      LCALL ARANG
15 25441 100270         ARANG-ASCRA*1B11+100010
16 25442 105000         MOV 0,1
17 25443 030442         LDA 2,CA.6
18 25444 102400         SUB 0,0
19 25445 040536         STA 0,CA.KK      ;0 # 1K'S EXTRA
20                      LCALL ADIVI
21 25446 100310         ADIVI-ASCRA*1B11+100010
22 25447 100405         NEG 0,0,SNR      ;(0)=# 1K'S TO EXPAND
23 25450 000410         JMP CA.2C
24              CA.2L:  LCALL ESCRA      ;TRY TO GET 1K
25 25451 100030         ESCRA-ASCRA*1B11+100010
26 25452 000406         JMP CA.2C        ;NO MORE AVAZC
27                      LCALL EDMAP      ;ALSO EXPND DCH 1K
28 25453 100570         EDMAP-ASCRA*1B11+100010
29 25454 000424         JMP CA.2R        ;CANT EXPAND DCH
30 25455 010526         ISZ CA.KK        ;#1 #1K'S ASSIGNED
31 25456 101404         INC 0,0,SZR
32 25457 000772         JMP CA.2L
33
34
```

```
!0343 N3MRT
01                ;SOME AMOUNT (1K TO 6K) SCRATCH AND DCH
02                ;ARE ASSIGNED TO THIS TEST
03                ;PICK START BUFFER WITHIN FIRST 400 WORDS
04
05                CA.2C:  LCALL ARANG
06 25460 100270           ARANG-ASCRA*1B11+100010
07 25461 105000           MOV 0,1
08 25462 030424           LDA 2,CA256
09                        LCALL ADIVI
10 25463 100310           ADIVI-ASCRA*1B11+100010
11 25464 024152           LDA 1,SCRLO
12 25465 107000           ADD 0,1
13 25466 044520           STA 1,CADST     ;DATA START ADDRESS
14 25467 024154           LDA 1,DCHLO
15 25470 107000           ADD 0,1
16 25471 044516           STA 1,CACST     ;CHANNEL START ADDRESS
17 25472 024511           LDA 1,CA.KK     ;# 1K'S AVAIL
18 25473 127120           ADDZL 1,1       ;*4
19 25474 020410           LDA 0,CA.3
20 25475 107000           ADD 0,1         ;AC1=# RECORDS THAT FIT
21 25476 044506           STA 1,CA.BK     ;INTO AVAILABLE BUFFER
22 25477 000411           JMP CA.03       ;DETERMINE NEXT OP
23
24                ;RELEASE 1K THAT COULDNT BE ASSIGNED TO DCH
25
26                CA.2R:  LCALL RSCRA
27 25500 100050           RSCRA-ASCRA*1B11+100010
28 25501 063077           HALT            ;NOT PROBABLE HALT
29 25502 000756           JMP CA.2C
30
31 25503 026255 CA.XX:    CA.XI
32 25504 000003 CA.3:     3
33 25505 000006 CA.6:     6
34 25506 000400 CA256:    256.
35 25507 000000 CADKX:    0       ;TRY FIND CTR
```

```
!0344 N3MRT
01                ;DATA CHANNEL AND SCRATCH ARE ASSIGNED
02                ;DETERMINE WHAT TO DO WITH DRIVE
03
04 25510 022472 CA.03:    LDA 0,@CA.TT
05 25511 100644           NEGOR 0,0,SZR   ;SKIP=FIND NEW DRIVE
06 25512 000425           JMP CA.3A
07 25513 030773           LDA 2,CA256
08 25514 050773           STA 2,CADKX
09 25515 030710           LDA 2,CA10
10                        LCALL ARANG     ;GET RND #
11 25516 100270           ARANG-ASCRA*1B11+100010
12 25517 105000           MOV 0,1         ;FOR DIVIDE
13                        LCALL ADIVI
14 25520 100310           ADIVI-ASCRA*1B11+100010
15 25521 061034           DOA 0,CAA
16 25522 060034           NIO CAA         ;KILL SOME TIME
17 25523 064434           DIA 1,CAA
18 25524 125233           MOVZR# 1,1,SNC  ;15=1 IS DRV RDY
19 25525 000404           JMP .+4         ;DRIVE NOT READY
20 25526 030703           LDA 2,CA.K3+1   ;WRITE LOCK BIT
21 25527 147405           AND 2,1,SNR     ;SKP IS WRITE LOCKED
22 25530 000404           JMP CA.3A-3     ;OK TO WRITE
23 25531 014756           DSZ CADKX       ;CANT USE DRV
24 25532 000763           JMP CA.03+5     ;TRY ANOTHER
25 25533 002750           JMP @CA.XX      ;STILL REWINDING EXIT
26 25534 101040           MOVO 0,0        ;CRY=1 IS WRITE
27 25535 040562           STA 0,CADRV     ;SAVE DRIVE#
28 25536 012444           ISZ @CA.TT      ;SET WRITE STATE
```

```
!0345 N3MRT
01 25537 101003 CA.3A:  MOV 0,0,SNC      ;WRITE SELECT CARRY=1
02 25540 000515         JMP CA.3R        ;TAPE IS IN READ MODE
03                      LCALL ARANG
04 25541 100270         ARANG-ASCRA*1B11+100010
05 25542 105000         MOV 0,1
06 25543 030507         LDA 2,CA64.
07 25544 020440         LDA 0,CA.BK
08 25545 112400         SUB 0,2
09                      LCALL ADIVI      ;WRITE AT LEAST
10 25546 100310         ADIVI-ASCRA*1B11+100010
11 25547 024435         LDA 1,CA.BK      ;ONE BUFFER FULL
12 25550 123000         ADD 1,0
13 25551 030431         LDA 2,CA.TT
14 25552 041002         STA 0,2,2        ;# BLOCKS TO WRITE
15 25553 041001         STA 0,1,2        ;# BLOCKS BKSP/READ
16 25554 125300         MOVS 1,1         ;# BLOCKS *400
17 25555 124600         NEGR 1,1         ;-FOR WC
18 25556 125400         INC 1,1          ;TAKE OFF 4
19 25557 125500         INCL 1,1
20 25560 044425         STA 1,CA.BL      ;BUFFER LENGTH
21                      LCALL FRANG
22 25561 100450         FRANG-ASCRA*1B11+100010
23 25562 040432         STA 0,CADW1      ;GET 4 RAN#'S
24 25563 044432         STA 1,CADW2      ;FOR DATA
25 25564 050432         STA 2,CADW3
26                      LCALL ARANG
27 25565 100270         ARANG-ASCRA*1B11+100010
28 25566 040431         STA 0,CADW4
29 25567 030417         LDA 2,CADST
30 25570 034420         LDA 3,CADWX
31 25571 126120         ADCZL 1,1
32 25572 125120         MOVZL 1,1        ;1=-4
33 25573 021400         LDA 0,0,3        ;MOV TO FIRST 4 WDS
34 25574 041000         STA 0,0,2        ;IN DATA BUFFER
35 25575 175400         INC 3,3
36 25576 151400         INC 2,2
37 25577 125404         INC 1,1,SZR
38 25600 000773         JMP .-5
39 25601 000421         JMP CAFIL
40 25602 025611 CA.TT:  CACTB
41 25603 000000 CA.KK:  0        ;# 1K'S ASSIGNED
42 25604 000000 CA.BK:  0        ;# TAPE BLOCKS AVAILABLE
43 25605 000000 CA.BL:  0        ;# WORDS IN BUFFER
44 25606 000000 CADST:  0        ;DATA START ADDRESS
45 25607 000000 CACST:  0        ;SAME BUT FOR DCH
46              ;TEST CONTROL TABLE
47 25610 025614 CADWX:  CADW1
48 25611 000000 CACTB:  0        ;MODE CONTROL REG
49 25612 000000          0       ;WRITE RECORD CTR
50 25613 000000          0       ;READ BKSPA CTR
51 25614 000000 CADW1:  0
52 25615 000000 CADW2:  0
53 25616 000000 CADW3:  0
54 25617 000000 CADW4:  0
55 25620 000000 CA.CA:  0        ;CURRENT BLOCK ADRS
56 25621 000000 CA.RK:  0        ;# RECORDS IN BUFFER
```

```
!0346 N3MRT
01              ;FIRST 4 WORDS IN BUFFER ARE FILLED
02              ;MOVE THEM TO REST OF BUFFER
03              ;AND START WRITING
04 25622 024763 CAFIL:  LDA 1,CA.BL
05 25623 021374         LDA 0,-4,2
06 25624 041000         STA 0,0,2
07 25625 151400         INC 2,2
08 25626 125404         INC 1,1,SZR
09 25627 000774         JMP .-4
10 25630 044470         STA 1,CADRV+1
11 25631 020541         LDA 0,CAMK1
12 25632 063034         DOC 0,CAA        ;LOAD WC
13 25633 024754         LDA 1,CACST      ;CHANNEL START
14 25634 066034         DOB 1,CAA        ;LOAD CA
15 25635 044464         STA 1,CADRV+2
16 25636 030501         LDA 2,CA.WI
17 25637 052570         STA 2,@CA00X     ;WRITE INT PROCESSOR
18 25640 030413         LDA 2,CA50       ;CA WRITE COMMAND
19 25641 020456         LDA 0,CADRV      ;DRIVE #
20 25642 113000         ADD 0,2          ;CREATE WRITE COMMAND
21 25643 020741         LDA 0,CA.BK
22 25644 040755         STA 0,CA.RK      ;# RECORDS 1 PASS BUFFER
23 25645 102000         ADC 0,0          ;SET BKSPACE SW
24 25646 040450         STA 0,CABSW
25 25647 044751         STA 1,CA.CA      ;START RECORD ADRS
26 25650 071134         DOAS 2,CAA
27                      LCALL RETRN
28 25651 100210         RETRN-ASCRA*1B11+100010
29 25652 000100 CA64.:  64.
30 25653 000050 CA50:   50
31 25654 000000 CA.EK:  0        ;READ RETRY CTR
```

```
!0347 N3MRT
01                      ;DRIVE IS IN READ STATE
02                      ;READ AS MANY RECORDS AS POSSIBLE
03                      ;INTO THE DATA BUFFER
04                      ;AFTER BACKSPACING OVER ALL WRITTEN
05 25655 030725 CA.3R:  LDA 2,CA.TT
06 25656 025002         LDA 1,2,2
07 25657 020725         LDA 0,CA.BK
08 25660 122432         SUBZ# 1,0,SZC
09 25661 121000         MOV 1,0             ;(0)=#RECORDS TO READ
10 25662 040722         STA 0,CA.BK
11 25663 040736         STA 0,CA.RK
12 25664 020723         LDA 0,CACST
13 25665 040733         STA 0,CA.CA
14 25666 062034         DOB 0,CAA           ;LOAD ADDRESS
15 25667 025002         LDA 1,2,2
16 25670 124400         NEG 1,1
17 25671 067034         DOC 1,CAA           ;WORD COUNT
18 25672 030522         LDA 2,CA.WB
19 25673 020423         LDA 0,CABSW
20 25674 100044         COMO 0,0,SZR
21 25675 000414         JMP CARDG
22 25676 052531         STA 2,@CA00X        ;BKSPA INTERRUPT HANDLER
23 25677 040417         STA 0,CABSW
24 25700 042415         STA 0,@CABSW-1      ;=CARDS
25 25701 101500         INCL 0,0
26 25702 010416         ISZ CADRV+1         ;SKP = RETRY
27 25703 040751         STA 0,CA.EK         ;3 FOR RETRY CTR
28 25704 020465         LDA 0,CA40          ;BKSPA DRV 0
29 25705 024412         LDA 1,CADRV
30 25706 123000         ADD 1,0             ;CREATE BKSPA THIS DRV
31 25707 061134         DOAS 0,CAA          ;START BKSPA
32                      LCALL RETRN
33 25710 100210         RETRN-ASCRA*1B11+100010
34 25711 004505 CARDG:  JSR CA.WB+2
35 25712 152400         SUB 2,2
36 25713 050565         STA 2,CARDS
37                      LCALL RETRN
38 25714 100210         RETRN-ASCRA*1B11+100010
39 25715 026100         CARDS
40 25716 000000 CABSW:  0
41 25717 000000 CADRV:  0           ;CURRENT DRIVE#
42 25720 000000         0           ;-1=RETRY ON READ DATA ERROR
43 25721 000000         0           ;CA START FOR WRITE OPERATIONS
```

```
!0348 N3MRT
01                      ;INTERRUPT DISPATCHER
02 25722 000034         CAA
03 25723 032504 CA.ID:  LDA 2,@CA00X
04 25724 151132         MOVZL# 2,2,SZC
05 25725 000405         JMP .+5
06 25726 020774         LDA 0,CA.ID-1
07 25727 040144         STA 0,UDEVI
08 25730 060234         NIOC CAA
09 25731 001400         JMP 0,3
10 25732 060634         DIAC 0,CAA
11 25733 040440         STA 0,CASTA
12 25734 101133         MOVZL# 0,0,SNC  ;SKIP IS ERR FLAG
13 25735 001001         JMP 1,2         ;PROCESS NORML RET
14 25736 001000         JMP 0,2         ;PROCESS ERROR
15
16                      ;WRITE INTERRUPT SERVICE ROUTINE
17 25737 125740 CA.WI:  @.+1
18 25740 000435         JMP CA.WE
19 25741 016433         DSZ @CAWTX      ;-1 RECORDS TO WRITE
20 25742 000404         JMP CA.WM       ;MORE TO DO
21 25743 020540         LDA 0,CA.04     ;SET UP BACKSPACE
22 25744 042463         STA 0,@CA00X    ;COMPARE WRITE BUFFER
23 25745 001400         JMP 0,3
24                      ;CONTINUE WRITING RECORDS
25 25746 020652 CA.WM:  LDA 0,CA.CA
26 25747 101300         MOVS 0,0
27 25750 101700         INCS 0,0        ;+400 TO ADRS
28 25751 040647         STA 0,CA.CA
29 25752 024635         LDA 1,CACST
30 25753 014646         DSZ CA.RK       ;SKIP DONE FULL BUFF
31 25754 000405         JMP .+5         ;NOT AT EOB YET
32 25755 044643         STA 1,CA.CA     ;RESET ADDRS TO START
33 25756 121000         MOV 1,0
34 25757 024625         LDA 1,CA.BK     ;#BLOCKS IN BUFFER
35 25760 044641         STA 1,CA.RK
36 25761 062034 CA.WN:  DOB 0,CAA       ;LOAD CA
37 25762 024410         LDA 1,CAMK1
38 25763 067034         DOC 1,CAA       ;-256 WK
39 25764 030667         LDA 2,CA50      ;WRITE COM
40 25765 024732         LDA 1,CADRV
41 25766 133000         ADD 1,2         ;CREATE WRITE+DRV#
42 25767 071134         DOAS 2,CAA      ;GO AGAIN
43 25770 001400         JMP 0,3
44 25771 000040 CA40: 40
45 25772 177400 CAMK1: -256.
46 25773 000000 CASTA: 0
47 25774 025612 CAWTX: CACTB+1
48
```

```
10349 N3MRT
01                ;ERROR STATUS DURING WRITE
02                ;BACK UP AND TRY AGAIN IF NOT EOT
03                ;REWIND RESTART IF EOT
04
05 25775 105300 CA.WE:   MOVS 0,1
06 25776 125200          MOVR 1,1
07 25777 125202          MOVR 1,1,SZC     ;SKIP NOT EOT
08 26000 000462          JMP CAREW        ;=EOT REWIND
09 26001 103132          ADDZL# 0,0,SZC
10 26002 000455          JMP CA.BE
11 26003 102000 CABK1:   ADC 0,0
12 26004 063034          DOC 0,CAA                ;-1 TO WC
13 26005 020407          LDA 0,CA.WB
14 26006 042421          STA 0,@CA00X
15 26007 030762          LDA 2,CA40       ;SPACE REVRS
16 26010 024707          LDA 1,CADRV
17 26011 133000          ADD 1,2
18 26012 071134          DOAS 2,CAA
19 26013 001400          JMP 0,3
20                ;PROCESS BACKSPACE COMPLETE INTERRUPTS
21                ;BACKSPACE OVER ALL WRITTEN RESTART READ
22                ;BACKSPACE 1 RECORD  COMPLETE=RESTART WRITE/READ
23 26014 126015 CA.WB:   @.+1
24 26015 000440          JMP CA.RE+2      ;BACKSP ERR REWIND
25 26016 022463          LDA 0,@CA.CX     ;CA.CA
26 26017 024720          LDA 1,CA.WI
27 26020 046407          STA 1,@CA00X     ;SET UP WRITE INTR
28 26021 026407          LDA 1,@CA00X+1
29 26022 125222          MOVZR 1,1,SZC    ;SKP IS READ MODE
30 26023 000736          JMP CA.WN        ;WRITING
31 26024 024405          LDA 1,CA.RI      ;SET UP READ INTR
32 26025 046402          STA 1,@CA00X
33 26026 000417          JMP CA.RC
34 26027 025313 CA00X:   CA.00+2
35 26030 025611          CACTB
```

```
10350 N3MRT
01                ;READ INTERRUPT PROCESSOR
02
03 26031 126032 CA.RI:   @.+1
04 26032 000421          JMP CA.RE        ;READ ERROR STATUS
05 26033 010445          ISZ CARDS        ;+1 #BLKS RD
06 26034 022445          LDA 0,@CA.CX
07 26035 101300          MOVS 0,0         ;CA + 400
08 26036 101700          INCS 0,0
09 26037 042442          STA 0,@CA.CX
10 26040 016442          DSZ @CA.CX+1     ;READ TO FILL BUFFER
11 26041 000404          JMP CA.RC        ;NO READ 1 MORE
12 26042 024511          LDA 1,CA.05
13 26043 046764          STA 1,@CA00X     ;COMPARE BUFFER
14 26044 001400          JMP 0,3          ;NEXT NORMAL ENTRY
15 26045 062034 CA.RC:   DOB 0,CAA
16 26046 024724          LDA 1,CAMK1      ;READ
17 26047 067034          DOC 1,CAA        ;NEXT BLOCK
18 26050 030647          LDA 2,CADRV              ;IN SEQUENCE
19 26051 071134          DOAS 2,CAA
20 26052 001400          JMP 0,3
21
22 26053 103132 CA.RE:   ADDZL# 0,0,SZC   ;EOT?
23 26054 000403          JMP CA.BE
24 26055 016523          DSZ @CAEKX       ;RE READ 3 TRYS
25 26056 000725          JMP CABK1        ;NO TRY AGAIN
26 26057 020413 CA.BE:   LDA 0,CA.NR      ;NON REC READ TYPEOUT
27 26060 042747          STA 0,@CA00X     ;NEXT NORMAL ENTRY
28 26061 001400          JMP 0,3
29
30
31                ;REWIND DRIVE CLEAR MODE
32
33 26062 020407 CAREW:   LDA 0,CA.10      ;REW
34 26063 024634          LDA 1,CADRV
35 26064 123000          ADD 1,0          ;CREATE REWIND
36 26065 061134          DOAS 0,CAA       ;TO CONTROL/DRV 0
37 26066 102400          SUB 0,0
38 26067 042406          STA 0,@CATTX     ;SET TAPE REWINDING
39 26070 000770          JMP CAREW-2
40 26071 000010 CA.10: 10
41 26072 026211 CA.NR: CADER
42 26073 025614 CA.DX: CADW1
43 26074 025620          CADW4+1
44 26075 025611 CATTX: CACTB
45 26076 025606 CADSX: CADST
46 26077 025604 CABKX: CA.BK
47 26100 000000 CARDS: 0         ;# BLOCKS READ
48 26101 025620 CA.CX:   CA.CA
49 26102 025621          CA.RK
```

```
10351 N3MRT
01                ;WRITE PASS COMPLETE RECHECK DATA FOR VALIDITY
02                ;READ PASS ENTERS AFTER CHECKING FOR 7 TRACK
03 26103 026104 CA.04:  .+1
04 26104 034767         LDA 3,CA.DX       ;START OF DATA RAN#'S
05 26105 032771         LDA 2,@CADSX      ;START OF BUFFER
06 26106 021400         LDA 0,0,3         ;GET NEXT OF 4
07 26107 025000         LDA 1,0,2         ;GET ONE OF FIRST 4
08 26110 122414         SUB# 1,0,SZR      ;SHD BE=
09 26111 000500         JMP CADER         ;ERROR IN ONE OF 4
10 26112 151400         INC 2,2           ;BUMP ADRS
11 26113 175400         INC 3,3
12 26114 020760         LDA 0,CA.DX+1
13 26115 116414         SUB# 0,3,SZR      ;SKP IS DONE FIRST 4
14 26116 000770         JMP CA.04+3
15 26117 036760         LDA 3,@CABKX      ;#400 WORD BLOCKS
16 26120 175300         MOVS 3,3          ;*400
17 26121 174600         NEGR 3,3
18 26122 175400         INC 3,3           ;4 LESS COMPARES
19 26123 175500         INCL 3,3          ;ARE NEEDED
20 26124 021374         LDA 0,-4,2        ;GET WORD VALIDATED
21 26125 025000         LDA 1,0,2         ;GET WORD NOT VALIDATED
22 26126 106414         SUB# 0,1,SZR      ;SKP=
23 26127 000462         JMP CADER         ;DATA ERROR
24 26130 055374         STA 3,-4,2        ;CLR BUFFER FOR NXT RD
25 26131 151400         INC 2,2           ;STP ADRS
26 26132 175404         INC 3,3,SZR       ;SKP DONE ALL
27 26133 000771         JMP .-7
28 26134 022741         LDA 0,@CATTX      ;GET MODE SW
29 26135 101223         MOVZR 0,0,SNC     ;SKP=CHNG WRITE TO RD
30 26136 000403         JMP .+3
31 26137 012736         ISZ @CATTX
32 26140 002412         JMP @CAFLX
33 26141 020737         LDA 0,CARDS
34 26142 026407         LDA 1,@CARDX      ;# BLKS TO RD
35 26143 106420         SUBZ 0,1          ;-#BLKS RD
36 26144 046405         STA 1,@CARDX      ;REM BLKS TO RD
37 26145 125004         MOV 1,1,SZR       ;NONE LEFT RELEASE BUFFER
38 26146 002404         JMP @CAFLX        ;RESELECT BUFFER START
39 26147 046726         STA 1,@CATTX      ;SET TO SELECT NEW DRIVE
40 26150 000505         JMP CA.XI
41 26151 025613 CARDX: CACTB+2
42 26152 025460 CAFLX: CA.2C     ;RESELECT IN FIRST 256
```

```
10352 N3MRT
01                ;READ COMPLETE RE-ENTER TAPE TEST
02 26153 026154 CA.05:  .+1               ;IF 7 TRACK DRIVE
03 26154 020724         LDA 0,CARDS
04 26155 042722         STA 0,@CABKX      ;#BLKS TO COMPARE
05 26156 022423         LDA 0,@CAEKX+1
06 26157 061034         DOA 0,CAA
07 26160 060434         DIA 0,CAA         ;CLR UNUSED BITS
08 26161 101300         MOVS 0,0
09 26162 103122         ADDZL 0,0,SZC     ;SKP IF 7 TRACK
10 26163 000721         JMP CA.04+1       ;9 TRACK COMP WD FOR WD
11 26164 030707         LDA 2,CA.DX       ;START 4 RAN#'S
12 26165 034707         LDA 3,CA.DX+1     ;END OF SAME
13 26166 024410         LDA 1,CA7TM       ;7 TRACK DATAMASK
14 26167 021000         LDA 0,0,2
15 26170 123400         AND 1,0           ;CLR ANY EXTRA BITS
16 26171 041000         STA 0,0,2
17 26172 151400         INC 2,2
18 26173 156414         SUB# 2,3,SZR
19 26174 000773         JMP .-5
20 26175 000707         JMP CA.04+1       ;NOW COMPARE
21 26176 037477 CA7TM:  37477
22 26177 025717         CADRV
23 26200 025654 CAEKX: CA.EK
24 26201 025717         CADRV
25 26202 025716         CABSW
26 26203 025720         CADRV+1
27 26204 025655         CA.3R
28 26205 025721         CADRV+2
29 26206 025607         CACST
```

```
!0353 N3MRT
01                ;TAPE DATA ERROR START TYPEOUTS
02
03 26207 126027           @CA00X
04 26210 025773           CASTA
05                CADER:  LCALL ERROI
06 26211 100350           ERROI-ASCRA*1B11+100010
07 26212 000401           JMP .+1
08 26213 020465           LDA 0,CATX2
09                        LCALL ERRTX
10 26214 100170           ERRTX-ASCRA*1B11+100010
11 26215 020663           LDA 0,CARDS
12 26216 026767           LDA 1,@CAEKX+5
13 26217 032767           LDA 2,@CAEKX+6
14                        LCALL ERROC
15 26220 100370           ERROC-ASCRA*1B11+100010
16 26221 000401           JMP .+1
17 26222 020443           LDA 0,CATX1
18                        LCALL ERRTX
19 26223 100170           ERRTX-ASCRA*1B11+100010
20 26224 022651           LDA 0,@CATTX    ;GET MODE,1=WRITE,2=READ
21 26225 026754           LDA 1,@CAEKX+1  ;GET DRIVE #
22 26226 127300           ADDS 1,1
23 26227 123000           ADD 1,0         ;FOR TYPE OUT
24 26230 026760           LDA 1,@CADER-1  ;GET STATUS
25 26231 032747           LDA 2,@CAEKX    ;GET # RETRIES
26                        LCALL ERROC
27 26232 100370           ERROC-ASCRA*1B11+100010
28 26233 000401           JMP .+1
29 26234 022754           LDA 0,@CADER-1  ;GET STATUS
30 26235 101102           MOVL 0,0,SZC    ;SKP IS NOT STATUS ER
31 26236 000416           JMP CA.XI-1
32 26237 020641           LDA 0,CARDS     ;# BLKS READ
33 26240 101005           MOV 0,0,SNR     ;SKP IS READ SOME
34 26241 000413           JMP CA.XI-1
35 26242 026736           LDA 1,@CAEKX    ;GET RETRY CTR
36 26243 125005           MOV 1,1,SNR     ;SKP=NOT RETRIED YET
37 26244 000410           JMP CA.XI-1
38 26245 016733           DSZ @CAEKX      ;
39 26246 042703           STA 0,@CARDX    ;SET # TO BACKUP/READ
40 26247 102000           ADC 0,0
41 26250 042732           STA 0,@CAEKX+2  ;SET BACKUP SW
42 26251 042732           STA 0,@CAEKX+3  ;RELOAD RETRY CTR
43 26252 002732           JMP @CAEKX+4    ;BACKUP/RE-READ
```

```
!0354 N3MRT
01 26253 026062           CAREW
02 26254 006777           JSR @.-1        ;REWIND START OVER
03                CA.XI:  LCALL RDMAP     ;RELEASE ALL DCH
04 26255 100610           RDMAP-ASCRA*1B11+100010
05 26256 102401           SUB 0,0,SKP
06 26257 000776           JMP CA.XI
07                CA.X2:  LCALL RSCRA     ;RELEASE ALL SCR
08 26260 100050           RSCRA-ASCRA*1B11+100010
09 26261 101001           MOV 0,0,SKP
10 26262 000776           JMP CA.X2
11 26263 042724           STA 0,@CADER-2  ;CLR SCR ASSIGNED
12                        LCALL RETRN
13 26264 100210           RETRN-ASCRA*1B11+100010
14
15 26265 026266 CATX1: .+1
16 26266 005215           .TXTE (<15><12>MODE     STATUS  CA.EK(
17 26300 026301 CATX2:  .+1
18 26301 005215           .TXTE !<15><12>#READ<11>W/DOB<11>LAST/DOB!
```

```
!0355 N3MRT
01                ;       .TITL LPTTS
02                ;LINE PRINTER TEST COMPATABLE WITH THE DIAGNOSTIC LINKER
03                        NEXTT LP.00
04        026315 LMEML=.
05        000172          .LOC LPG0
06 00172 026320           LP.00
07        000173 LPG0=.
08        026315          .LOC LMEML
09 26315 000000           0       ;TEST PASS CTR
10 26316 000000           0       ;TEST ERROR CTR
11 26317 000000           0       ;INTERRUPT TIMEOUT SWITCH
12 26320 026337 LP.00:    LP.01
13 26321 026360           LP.02
14 26322 000000           0
15 26323 001750           1000.
16 26324 003244           1700.
17 26325 176000           176000
18 26326 026374           LP.XI
19 26327 026374           LP.XI
20        026330          .TXTE (
21 26330 144714 LINE PRINTER(
22        142516
23        050240
24        144722
25        152116
26        151305
27        000000
28                ;INITIALIZE LINE PRINTER TEST
29 26337 102000 LP.01:    ADC 0,0
30 26340 040762           STA 0,LP.00+2
31 26341 064417           DIA 1,LPT
32 26342 125223           MOVZR 1,1,SNC   ;LPT ON LINE
33 26343 001400           JMP 0,3         ;NO DON'T TEST IT
34 26344 102400           SUB 0,0
35 26345 040412           STA 0,LPCRF     ;CLR CARRET OUT
36 26346 040754           STA 0,LP.00+2
37 26347 040436           STA 0,LP.LK
38 26350 020404           LDA 0,LP.K1     ;DEV #
39 26351 024404           LDA 1,LP.K2     ;INTR MSK'
40 26352 030404           LDA 2,LP.K3     ;INTR SERV ADRS
41 26353 002101           JMP @EINTS      ;PASSED TO INTA SERVICE
42 26354 000017 LP.K1:    LPT
43 26355 000017 LP.K2:    17
44 26356 026516 LP.K3:    LPINT
45 26357 000000 LPCRF:    0
```

```
!0356 N3MRT
01                ;EXECUTE PORTION OF LPT TEST
02                ;ENTERED AT RANDOM INTERVALS
03 26360 020425 LP.02:    LDA 0,LP.LK     ;GET LINE CTR
04 26361 101004           MOV 0,0,SZR     ;TIME FOR FORM FEED
05 26362 000425           JMP LP.2A       ;NO
06 26363 020774           LDA 0,LPCRF
07 26364 101005           MOV 0,0,SNR
08 26365 000410           JMP LPCRO
09 26366 102400           SUB 0,0
10 26367 040770           STA 0,LPCRF     ;CR HAS BEEN DONE
11 26370 020416           LDA 0,LP.FF
12 26371 126000 LPFFO:    ADC 1,1
13 26372 044730           STA 1,LP.00+2   ;SET WAITING INT
14 26373 061117           DOAS 0,LPT
15                LP.XI:  LCALL RETRN
16 26374 100210           RETRN-ASCRA*1B11+100010
17
18 26375 020472 LPCRO:    LDA 0,LP.CR
19 26376 010761           ISZ LPCRF
20 26377 061117           DOAS 0,LPT
21 26400 063517           SKPBZ LPT
22 26401 000777           JMP .-1
23 26402 000756           JMP LP.02
24 26403 000133 LP.ZK: 133        ;Z+1
25 26404 000040 LP.40: 40         ;7 BIT SPACE
26 26405 000000 LP.LK: 0          ;LINE COUNTER
27 26406 000014 LP.FF: 14         ;FORM FEED CODE
28 26407 101404 LP.2A:    INC 0,0,SZR
29 26410 000411           JMP LP.2B       ;START NEXT LINE
30                        LCALL ARANG     ;RAN#
31 26411 100270           ARANG-ASCRA*1B11+100010
32 26412 105000           MOV 0,1
33 26413 102400           SUB 0,0
34                ;RANDOMLY SELECT 10 TO 60 LINES TO PRINT
35 26414 030422           LDA 2,LP.50
36                        LCALL ADIVI
37 26415 100310           ADIVI-ASCRA*1B11+100010
38 26416 024421           LDA 1 LP.10
39 26417 123000           ADD 1,0
40 26420 040765           STA 0,LP.LK
```

```
:0357 N3MRT
01                ;START PRINTING SPACE TO Z
02                ;WITH RANDOM STALLS AT EVERY 1 TO 9 LINES
03                LP.28:  LCALL ARANG
04 26421 100270           ARANG-ASCRA*1B11+100010
05 26422 105000           MOV 0,1
06 26423 102400           SUB 0,0
07 26424 030413           LDA 2,LP.10
08                        LCALL ADIVI
09 26425 100310           ADIVI-ASCRA*1B11+100010
10 26426 040412           STA 0,LPSTALL
11 26427 020755           LDA 0,LP.40
12 26430 040405           STA 0,LP.CK
13 26431 102000           ADC 0,0
14 26432 040670           STA 0,LP.00+2   ;SET WAITING INTA
15 26433 004406           JSR LPFILL
16                        LCALL RETRN
17 26434 100210           RETRN-ASCRA*1B11+100010
18 26435 000000 LP.CK:  0
19 26436 000062 LP.50:  50.
20 26437 000012 LP.10:  10.
21 26440 000000 LPSTALL:  0       ;STALL BY LINE CTR
```

```
:0358 N3MRT
01                ;FILL PRINTER BUFFER UNTIL PRINT CYCLE
02                ;OR FORCE LINE FEED IF Z AND MORE LINES TO PRINT
03 26441 020774 LPFILL:  LDA 0,LP.CK
04 26442 024741           LDA 1,LP.ZK
05 26443 106415           SUB# 0,1,SNR
06 26444 000417           JMP LPCR2
07 26445 061117           DOAS 0,LPT
08 26446 010767           ISZ LP.CK       ;ADVANCE TO NEXT CHAR
09 26447 126120           ADCZL 1,1       ;ZONE DETECT LOOPS
10 26450 152120           ADCZL 2,2
11 26451 063417           SKPBN LPT
12 26452 000767           JMP LPFILL
13 26453 151404           INC 2,2,SZR
14 26454 000775           JMP .-3
15                ;WAIT FOR BUFFER READY OR DETECT START OF PRINT
16                ;WHEN PRINTER ZONE IS FULL
17 26455 063517           SKPBZ LPT
18 26456 152121           ADCZL 2,2,SKP
19 26457 000762           JMP LPFILL
20 26460 125404           INC 1,1,SZR
21 26461 000773           JMP .-5
22 26462 001400           JMP 0,3         ;END OF ZONE EXIT
23 26463 020404 LPCR2:  LDA 0,LP.CR
24 26464 010673           ISZ LPCRF
25 26465 061117           DOAS 0,LPT
26 26466 001400           JMP 0,3
27 26467 000015 LP.CR:  15
```

```
!0359 N3MRT
01                ;PRINT BUFFER IS FILLED TO "Z"
02                ;START LINE FEED ON ITS WAY
03                ;OR FORCE TO FORM FEED IF LAST LINE THIS PAGE
04 26470 014715 LPEND:  DSZ LP.LK
05 26471 000404         JMP .+4          ;DO A LINE FEED
06 26472 102220         ADCZR 0,0
07 26473 040627         STA 0,LP.00+2    ;FORCE A FORM FEED
08 26474 000416         JMP LP.LF-2
09 26475 020417         LDA 0,LP.LF
10 26476 061117         DOAS 0,LPT       ;PRINT REST OF LINE
11 26477 063517         SKPBZ LPT
12 26500 000777         JMP .-1
13 26501 000415         JMP LPINT
14 26502 020703 LPCRI:  LDA 0,LP.LK      ;INTA  WAS FROM A CR
15 26503 101005         MOV 0,0,SNR
16 26504 000404         JMP .+4 ;DECIDE IF LF OR FF
17 26505 102400         SUB 0,0
18 26506 040651         STA 0,LPCRF
19 26507 000761         JMP LPEND
20 26510 102220         ADCZR 0,0
21 26511 040611         STA 0,LP.00+2
22 26512 060217         NIOC LPT
23 26513 001400         JMP 0,3
24 26514 000012 LP.LF: 12
25 26515 026322 LP002: LP.00+2
26
27                ;INTERRUPT SERV DISPATCH FOR LPT
28 26516 020150 LPINT:  LDA 0,EACTV
29 26517 103102         ADDL 0,0,SZC     ;SKP=LPT
30 26520 000772         JMP LP.LF-2      ;ACTV ERROR
31 26521 020636         LDA 0,LPCRF
32 26522 101004         MOV 0,0,SZR
33 26523 000757         JMP LPCRI
34 26524 020661         LDA 0,LP.LK
35 26525 101004         MOV 0,0,SZR      ;=0 WAS F/F
36 26526 000407         JMP .+7
37 26527 102400         SUB 0,0
38 26530 042765         STA 0,@LP002
39 26531 102000         ADC 0,0
40 26532 040653         STA 0,LP.LK      ;F/F DONE SW
41 26533 060217         NIOC LPT         ;CLR DONE
42 26534 001400         JMP 0,3 ;DISMISS INTR
43 26535 020700         LDA 0,LP.CK
44 26536 024645         LDA 1,LP.ZK
45 26537 106414         SUB# 0,1,SZR
46 26540 000701         JMP LPFILL       ;STILL FILLING BUFFER
47                ;MUST BE INTR FROM LINE FEED
48 26541 020643         LDA 0,LP.40
49 26542 040673         STA 0,LP.CK
50 26543 014675         DSZ LPSTALL      ;SKP FOR MOMENTARY STALL
51 26544 000675         JMP LPFILL
52 26545 102220         ADCZR 0,0
53 26546 042747         STA 0,@LP002
54 26547 060217         NIOC LPT
55 26550 001400         JMP 0,3
```

```
!0360 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        026551 LMEML=.
11        000173          .LOC LPG0
12 00173 026554           RT.00
13        000174 LPG0=.
14        026551          .LOC LMEML
15 26551 000000           0       ;TEST PASS CTR
16 26552 000000           0       ;TEST ERROR CTR
17 26553 000000           0       ;INTERRUPT TIMEOUT SWITCH
18 26554 026574 RT.00:    RT.01
19 26555 026623           RT.02
20 26556 000000           0
21 26557 000000           0
22 26560 177777           -1
23 26561 176000           176000
24 26562 026646           RT.03
25 26563 026646           RT.03
26 26564 142722           .TXTE (REAL TIME CLOCK(
27        146101
28        152240
29        046711
30        120305
31        146303
32        141717
33        000113
```

```
!0361 N3MRT
01                ;DETERMINE WHETHER OR NOT A REAL TIME CLOCK EXISTS
02                ;ENABLE OR DISABLE TEST ACCORDINGLY
03 26574 060277 RT.01:  INTDS
04 26575 102000         ADC 0,0
05 26576 040760         STA 0,RT.00+2   ;DISABLES TEST
06 26577 060114         NIOS RTC        ;TURN CLOCK ON
07 26600 063514         SKPBZ RTC       ;SKIP MAYBE NO CLOCK
08 26601 000402         JMP .+2         ;CLOCK EXISTS
09 26602 063714         SKPDZ RTC       ;TRY FOR DONE =1
10 26603 000402         JMP .+2         ;CLOCK EXISTS
11                ;NO RTC ON SYSTEM EXIT WITH CLOCK DISABLED
12 26604 001400         JMP 0,3
13 26605 102400         SUB 0,0
14 26606 040750         STA 0,RT.00+2   ;ENABLE CLOCK TEST
15 26607 020413         LDA 0,RT.02-1
16 26610 040745         STA 0,RT.00+1
17 26611 020405         LDA 0,RT.K1
18 26612 024405         LDA 1,RT.K2
19 26613 030405         LDA 2,RT.K3
20 26614 060214         NIOC RTC        ;TURN CLOCK OFF
21 26615 002101         JMP @EINTS
22 26616 000014 RT.K1: RTC
23 26617 000007 RT.K2: 7
24 26620 026706 RT.K3: RT.ID
25 26621 000005 RTFIV: 5
```

```
!0362 N3MRT
01                ;START CLOCK TEST IS NOT DELETED
02 26622 026623         RT.02
03 26623 020424 RT.02:  LDA 0,RT.K4
04 26624 040731         STA 0,RT.00+1
05 26625 102420         SUBZ 0,0
06 26626 101500         INCL 0,0
07 26627 024450         LDA 1,RT.K5
08 26630 044450         STA 1,RTSEC     ;TO COUNT 1 SECOND
09 26631 024445         LDA 1,RT.K6
10 26632 044447         STA 1,RTMIN     ;60 SEC'S =1MIN.
11 26633 024766         LDA 1,RTFIV
12 26634 131120         MOVZL 1,2
13 26635 044445         STA 1,RTCTR     ;TO COUNT DOWN 1ST
14 26636 050445         STA 2,RTCTR+1   ;2ND AFTER 10 MORE
15 26637 133000         ADD 1,2
16 26640 126400         SUB 1,1
17 26641 044142         STA 1,RTTIM
18 26642 126000         ADC 1,1
19 26643 044146         STA 1,TIMSW     ;SET RT=0 INHIBIT TIME
20 26644 050440         STA 2,RTCTR+2   ;3RD AT HALF HOUR
21 26645 061114         DOAS 0,RTC      ;TURN CLOCK ON 1K HZ
22                RT.03:  LCALL RETRN
23 26646 100210         RETRN-ASCRA*1B11+100010
24 26647 026650 RT.K4: RT.04
25 26650 102400 RT.04:  SUB 0,0
26 26651 040700         STA 0,RT.00-3   ;SO NO 65K TYPEOUT
27 26652 020146         LDA 0,TIMSW
28 26653 101004         MOV 0,0,SZR     ;PRINT TIME
29 26654 000772         JMP RT.03       ;NOT YET
30 26655 102000         ADC 0,0
31 26656 040146         STA 0,TIMSW     ;RESET INH. SW
32 26657 020454         LDA 0,RTTEX
33                      LCALL ERRTX
34 26660 100170         ERRTX-ASCRA*1B11+100010
35 26661 102400         SUB 0,0
36 26662 024142         LDA 1,RTTIM
37 26663 030413         LDA 2,RT.K6
38                      LCALL ADIVI
39 26664 100310         ADIVI-ASCRA*1B11+100010
40 26665 111000         MOV 0,2         ;SAVE MINS
41                      LCALL PDECI     ;PRINT HRS
42 26666 100150         PDECI-ASCRA*1B11+100010
43 26667 145000         MOV 2,1
44                      LCALL PDECI     ;ELAPSED TIME IN MINUTES
45 26670 100150         PDECI-ASCRA*1B11+100010
46 26671 024147         LDA 1,ERTOT     ;# ERR TYPEOUTS
47 26672 125005         MOV 1,1,SNR
48 26673 000753         JMP RT.03
49                      LCALL PDECI     ;PRINT ERROR TOTAL
50 26674 100150         PDECI-ASCRA*1B11+100010
51 26675 000751         JMP RT.03
```

```
!0363 N3MRT
01 26676 000074 RT.K6: 60.
02 26677 001750 RT.K5: 1000.
03 26700 000000 RTSEC: 0
04 26701 000000 RTMIN: 0
05 26702 000000 RTCTR: 0
06 26703 000000          0
07 26704 000000          0
08 26705 000036          30.
09 26706 060114 RT.ID:   NIOS RTC
10 26707 014771          DSZ RTSEC        ;1 SECOND
11 26710 000422          JMP RTSTR        ;NO
12 26711 020766          LDA 0,RT.K5
13 26712 040766          STA 0,RTSEC
14 26713 014766          DSZ RTMIN        ;1 MINUTE
15 26714 000416          JMP RTSTR        ;NO
16 26715 010142          ISZ RTTIM        ;BUMP ELAPSED MINUTES
17 26716 024760          LDA 1,RT.K6
18 26717 044762          STA 1,RTMIN      ;RES. MIN. CTR
19 26720 014762          DSZ RTCTR        ;TIME TO PRINT
20 26721 000411          JMP RTSTR        ;NOT YET
21 26722 126400          SUB 1,1
22 26723 044146          STA 1,TIMSW      ;CLR PR. INH. SW
23 26724 020757          LDA 0,RTCTR+1
24 26725 024757          LDA 1,RTCTR+2
25 26726 030757          LDA 2,RTCTR+3
26 26727 040753          STA 0,RTCTR
27 26730 044753          STA 1,RTCTR+1
28 26731 050753          STA 2,RTCTR+2
29               RTSTR:
30 26732 001400          JMP 0,3          ;RESTRT CLK DISMISS INTR
31 26733 026734 RTTEX: .+1
32 26734 005215          .TXTE !<15><12>R/T(HRS,MINS,ERTOT)= !
```

```
!0364 N3MRT
01                ;       .TITL TTTES
02                        NEXTT TT.00
03       026750 LMEML=.
04       000174           .LOC LPG0
05 00174 026753           TT.00
06       000175 LPG0=.
07       026750           .LOC LMEML
08 26750 000000           0        ;TEST PASS CTR
09 26751 000000           0        ;TEST ERROR CTR
10 26752 000000           0        ;INTERRUPT TIMEOUT SWITCH
11 26753 026770 TT.00:    TT.01
12 26754 027013           TT.02
13 26755 000000           0        ;WAIT FOR INTERRUPT SW
14 26756 177700           -64.
15 26757 177716           -50.
16 26760 176000           176000
17 26761 027017           TT.XI
18 26762 027017           TT.XI
19       026763           .TXTE (
20 26763 152324 TTY TEST(
21       120131
22       142724
23       152123
24       000000
25                ;INITIALIZE TTY TEST
26 26770 102400 TT.01:    SUB 0,0
27 26771 040566           STA 0,TT.CK      ;CLR CHAR COUNT
28 26772 040763           STA 0,TT.00+2    ;CLR WAIT INTR
29 26773 020425           LDA 0,TT.03      ;INTR ADRS
30 26774 054460           STA 3,TT.S3
31 26775 020411           LDA 0,TT.K1
32 26776 024411           LDA 1,TT.K2
33 26777 030411           LDA 2,TT.K3
34 27000 006101           JSR @EINTS       ;ENTER KYBORD INT SERV
35 27001 020410           LDA 0,TT.K4
36 27002 030410           LDA 2,TT.K5
37 27003 006101           JSR @EINTS       ;ENTER TTO INT SERV
38 27004 034450           LDA 3,TT.S3
39 27005 001400           JMP 0,3
40 27006 000010 TT.K1: TTI
41 27007 000003 TT.K2: 3
42 27010 027021 TT.K3: TT.TI
43 27011 000011 TT.K4: TTO
44 27012 027117 TT.K5: TT.TO
45                ;START TTY PRINTING
46 27013 102620 TT.02:    SUBZR 0,0
47 27014 040741           STA 0,TT.00+2    ;SET WAITING INT
48 27015 020536           LDA 0,TT.CR
49 27016 061111           DOAS 0,TTO       ;OUT CARG RET
50                TT.XI:  LCALL RETRN
51 27017 100210           RETRN-ASCRA*1B11+100010
```

```
!0365 N3MRT
01                ;TTY INTR SERVICE
02 27020 027021 TT.03:  .+1
03 27021 054433 TT.TI:  STA 3,TT.S3
04 27022 060610 TT.RD:  DIAC 0,TTI
05 27023 030434         LDA 2,TT177
06 27024 143400         AND 2,0
07 27025 024150         LDA 1,EACTV
08 27026 125202         MOVR 1,1,SZC
09 27027 000421         JMP TT.EX
10 27030 024151         LDA 1,LASTI     ;CK IF LAST TTI INTA SERVICED
11 27031 125113         MOVL# 1,1,SNC   ;SKP IS NOT SERVICED
12 27032 000437         JMP TT.RE
13 27033 147400         AND 2,1         ;MASK OFF PARITY BIT
14 27034 106404         SUB 0,1,SZR     ;SAME CHAR TWICE ?
15 27035 000434         JMP TT.RE       ;NOPE
16 27036 024420         LDA 1,TT.17
17 27037 106415         SUB#0,1,SNR     ;SKP IF NOT CONTROL O
18 27040 000423         JMP EODT        ;GO DIRECT TO ODT
19 27041 024412         LDA 1,TTI04     ;CONTROL "D"
20 27042 122415         SUB# 1,0,SNR    ;SKIP IS NOT CONTROL D
21 27043 000422         JMP TT.P?       ;KEY INPUT SERVICE
22 27044 024411         LDA 1,TT.22     ;CK FOR CONTROL R
23 27045 122415         SUB# 1,0,SNR    ;SKP IS NOT CONTROL R
24 27046 000417         JMP TT.P?       ;KEY INPUT SERVICE
25 27047 000422         JMP TT.RE
26 27050 103240 TT.EX:  ADDOR 0,0
27 27051 040151         STA 0,LASTI     ;SAVE CHAR FOR LATER
28 27052 002402 TT.DI:  JMP @TT.S3
29 27053 000004 TTI04:  4
30 27054 000000 TT.S3:  0
31 27055 000022 TT.22:  22
32 27056 000017 TT.17:  17
33 27057 000177 TT177:  177
34 27060 000100 TT100:  100
35 27061 000060 TT.60:  60
36 27062 004154 PODT:   ODT
37 27063 006777 EODT:   JSR @PODT
38 27064 002770         JMP @TT.S3      ;RETURN TO DISMIS INTR
```

```
!0366 N3MRT
01
02 27065 040151 TT.P?:  STA 0,LASTI     ;PLACE CHARACTER IN LASTI
03 27066 006402         JSR @IN.P?
04 27067 002765         JMP @TT.S3
05 27070 005223 IN.P?:  INP?
06
07                ;NOW CHECK FOR SWREG COMMANDS
08 27071 034770 TT.RE:  LDA 3,TT.60
09 27072 116405         SUB 0,3,SNR     ;SKIP IF NOT 60
10 27073 000755         JMP TT.EX
11 27074 152620         SUBZR 2,2       ;NOW CK FOR KEYS 1-9,A-F
12 27075 024763         LDA 1,TT100
13 27076 151221 TT.SC:  MOVZR 2,2,SKP
14 27077 126520         SUBZL 1,1
15 27100 175405         INC 3,3,SNR     ;SKIP IF NOT FOUND IT YET
16 27101 000410         JMP TT.SS       ;FOUND IT, SET SWREG BIT
17 27102 133415         AND# 1,2,SNR    ;DONE THIS GROUP YET
18 27103 000773         JMP TT.SC       ;NOT YET
19 27104 106400         SUB 0,1 ;SET UP FOR KEY A-F NOW
20 27105 135000         MOV 1,3
21 27106 151225         MOVZR 2,2,SNR   ;DONE ALL ?
22 27107 000741         JMP TT.EX       ;YEP
23 27110 000767         JMP TT.SC+1     ;NOT YET
24 27111 024230 TT.SS:  LDA 1,SWREG
25 27112 133414         AND# 1,2,SZR
26 27113 146401         SUB 2,1,SKP     ;XOR SWREG BIT
27 27114 147000         ADD 2,1
28 27115 044230         STA 1,SWREG     ;PLACE BACK IN SWREG
29 27116 000732         JMP TT.EX
```

```
!0367 N3MRT
01                ;TTY OUTPUT INTR HANDLER
02 27117 020440 TT.TO:  LDA 0,TT.CK
03 27120 060211         NIOC TTO
04 27121 054733         STA 3,TT.S3
05 27122 034150         LDA 3,EACTV
06 27123 175102         MOVL 3,3,SZC    ;SKP IF NOT ETYPE
07 27124 002730         JMP @TT.S3      ;THROW DONE AWAY
08 27125 034630         LDA 3,TT.00+2   ;WAITING TTO INTR?
09 27126 024432         LDA 1,TT.80     ;=180
10 27127 136404         SUB 1,3,SZR     ;SKP IS TTO TEST OUTPUT
11 27130 002724         JMP @TT.S3      ;THROW DNE AWAY
12 27131 101004         MOV 0,0,SZR     ;SKP FOR LINE FEED
13 27132 000406         JMP TT.04       ;INTO PRINTING CHAR
14 27133 020421         LDA 0,TT.LF
15 27134 061111         DOAS 0,TTO
16 27135 020421         LDA 0,TT.SP     ;GET SPACE
17 27136 040421         STA 0,TT.CK     ;TO NXT OUT
18 27137 000713         JMP TT.DI       ;DISMISS INTR
19                ;OUTPUT SPACE TO Z
20 27140 024415 TT.04:  LDA 1,TT.EN
21 27141 122415         SUB# 1,0,SNR    ;SKP IS NOT Z YET
22 27142 000404         JMP TT.05
23 27143 061111         DOAS 0,TTO
24 27144 010413         ISZ TT.CK
25 27145 000705         JMP TT.DI
26                ;ALL CHARACTERS PRINTED STOP INTERRUPTS
27 27146 102400 TT.05:  SUB 0,0
28 27147 040606         STA 0,TT.00+2
29 27150 060211         NIOC TTO
30 27151 040406         STA 0,TT.CK
31 27152 000700         JMP TT.DI
32 27153 000215 TT.CR:  215
33 27154 000212 TT.LF:  212
34 27155 000333 TT.EN:  333
35 27156 000240 TT.SP:  240
36 27157 000000 TT.CK:  0
37 27160 100000 TT.80:   180
```

```
!0368 N3MRT
01 27161 000000 EXISM:  0
02 27162 000007         .BLK 7
03 27171 000000 AVALM:  0
04 27172 000007         .BLK 7
05 27201 000000 DCHM0:  0       ;DCH A MAP
06 27202 000000 DCHM1:  0
07 27203 000000 DCHM2:  0       ;DCH B MAP
08 27204 000000 DCHM3:  0
09 27205 000000 LSYSE:  0
                        .TITL N3MRT
11 27206 031516 DIRT:   .TXTE !N3MRT L  04!
12 27214 000000         0
13 27215 000210         STRODT
14 27216 167771         167771
15 27217 070707         070707
16 27220 070707         070707
17 27221 070707         070707
18 27222 070707         070707
19 27223 000000         0
20 27224 047503         .TXT !COPYRIGHT (C) DGC,1973,1974,1975,1976,1977
21 27251 040440  ALL RIGHTS RESERVED!
22 01000 000000         0
23 01001 000175         LPG0
24 01002 000000         0
25 01003 000000         0
26 01004 027205         LSYSE
27                      .END
```

0369 N3MRT

A1	011221	153/42	154/01					
A6L	011332	157/11	157/15					
A6L.	016473	245/24	245/29					
ADDTE	011355	157/10	157/16	157/46				
ADIVI	000100	32/19	48/49	52/25	118/19	119/11	138/13	180/20
		181/12	182/17	215/20	216/10	217/12	219/11	221/07
		257/20	258/08	259/10	262/06	271/26	272/08	273/15
		275/14	276/08	277/14	279/20	300/23	301/08	302/16
		304/14	305/08	306/14	308/22	325/21	326/10	327/15
		328/10	342/21	343/10	344/14	345/10	356/37	357/09
		362/39						
ADMAP	000112	33/01	215/11	229/13	257/10	271/16	300/12	325/12
		342/12						
AD.DT	016513	245/23	245/30	245/44				
AGSTR	001303	39/22	39/32					
AI.TX	006156	104/44	105/33					
ALAUT	001304	39/27	39/33					
ALDSP	001305	39/28	39/34					
ALTBL	000137	33/29	48/16	48/35	49/40	58/07	63/05	63/06
		63/09	65/03	67/09	68/09	69/08	69/12	69/22
		70/18	76/44	106/25	107/07	107/10	107/31	108/08
		109/07	110/19	111/05	112/10	113/04		
ALZMA	001242	38/06	38/23					
AM777	006527	108/17	111/32					
AMSCR	006442	33/01	107/04					
AM.37	006522	107/26	108/12	109/13	111/20	113/07		
AM.AA	006530	107/23	108/18	111/39				
AM.BB	006531	107/21	108/19	111/37				
AM.GA	006525	107/37	108/04	108/15	110/29			
AM.GI	006466	107/24	107/25	110/11	111/34			
AM.K1	006523	107/42	108/13					
AM.NM	006511	107/18	108/03					
AM.RM	006526	107/25	108/16					
AM.S3	006521	107/06	107/41	107/42	107/44	108/11	109/06	111/04
		111/27						
AM.TM	006524	107/35	107/39	108/14	110/16	110/31		
AM.XT	006507	107/09	107/15	107/28	107/42	108/10	109/17	109/24
		110/10	110/15	110/33	111/12	111/30	111/41	
AND3L	010766	146/11	146/19					
AND.L	016146	239/34	239/42					
AR4	011523	161/26	164/01					
AR4L	011524	164/04	164/11					
AR4L.	016661	249/03	249/08					
AR4.	016660	247/36	249/02					
ARANG	000077	32/18	48/46	49/24	49/44	52/21	54/04	106/20
		118/15	119/05	122/04	136/16	138/09	138/39	180/16
		181/08	182/11	183/15	186/19	189/19	207/07	215/14
		216/06	217/05	218/14	219/06	219/21	219/58	221/03
		224/36	224/43	229/34	231/12	257/13	258/04	259/04
		260/13	261/06	262/02	265/36	265/43	271/19	272/04
		273/05	274/09	275/09	275/29	276/04	277/04	291/36
		291/43	300/15	301/04	302/06	303/09	304/08	304/29
		305/04	306/04	316/37	316/48	325/15	326/06	327/12
		328/04	328/27	342/15	343/06	344/11	345/04	345/27
		356/31	357/04					
AREND	012164	177/25						
ARITH	000000	2/06	2/24	9/02	20/01	134/01		
ARJSR	012145	177/01						
ASCRA	000064	32/07	60/01	62/13	62/37	62/39	118/10	118/15

0370 N3MRT

118/19	118/23	119/05	119/11	120/23	120/28	120/35
121/04	121/08	121/13	122/04	129/11	129/14	131/40
131/44	132/07	132/11	132/16	136/06	136/11	136/16
137/25	138/09	138/13	138/27	138/39	138/44	139/06
139/10	139/15	139/18	139/23	141/03	141/06	141/14
141/17	141/22	141/25	141/33	141/36	142/04	142/11
142/17	142/23	142/26	143/04	143/07	143/31	143/34
144/04	144/07	144/31	144/34	145/04	145/07	145/24
145/27	145/32	145/35	145/42	145/45	146/04	146/07
146/26	146/29	147/06	147/09	147/22	147/25	147/30
147/33	147/45	147/48	148/04	148/07	148/20	148/23
148/28	148/31	148/41	148/44	149/04	149/07	149/17
149/20	149/25	149/28	149/39	149/42	151/03	151/06
151/16	151/19	151/24	151/27	151/37	151/40	152/04
152/15	152/18	152/23	152/34	152/37	153/04	153/07
153/37	153/40	154/03	154/06	154/15	154/18	154/23
154/26	154/35	154/38	155/04	155/07	155/16	155/19
155/23	155/26	155/47	155/50	156/04	156/07	156/22
156/25	157/04	157/07	157/22	157/25	157/30	157/33
157/40	157/43	158/04	158/07	158/22	158/25	159/06
159/09	159/25	159/28	160/04	160/07	160/25	160/28
161/04	161/07	161/22	161/25	164/03	164/06	164/24
164/31	164/38	164/41	165/04	165/21	165/24	166/04
166/20	166/23	167/04	167/10	167/16	167/22	167/28
167/34	167/40	167/46	167/52	167/58	168/04	168/07
169/04	169/11	169/17	169/23	169/29	169/35	169/41
169/47	169/53	169/59	170/02	171/04	171/07	171/18
171/21	171/26	171/39	171/42	172/04	172/07	172/16
172/19	172/24	172/27	172/35	172/38	173/04	173/22
173/25	173/30	173/33	173/46	173/49	174/04	174/07
174/17	174/20	175/04	175/07	175/17	175/20	175/25
175/28	175/38	175/41	176/04	176/07	176/17	176/20
176/25	176/28	176/38	176/41	177/03	177/10	177/20
177/23	177/26	180/11	180/16	180/20	180/24	181/08
181/12	182/11	182/17	182/48	183/15	183/20	183/42
184/06	184/10	184/15	184/19	184/24	184/28	184/33
184/37	184/48	186/19	187/47	188/12	189/19	191/23
191/37	194/03	194/17	194/22	194/36	195/04	195/18
195/23	195/37	196/04	196/18	196/23	196/37	197/04
197/18	197/23	197/37	198/03	198/17	198/22	198/36
199/04	199/17	199/22	199/36	200/04	200/17	200/22
200/36	201/04	201/18	201/23	201/37	202/04	202/18
205/03	205/21	205/25	205/43	206/05	206/22	207/07
210/03	210/08	210/13	210/19	210/24	210/27	211/32
211/42	211/44	211/48	211/51	215/08	215/11	215/14
215/20	215/24	215/27	216/06	216/10	216/34	217/05
217/12	218/03	218/14	219/06	219/11	219/21	219/58
221/03	221/07	222/39	224/36	224/43	224/55	224/59
226/02	226/06	226/11	226/18	226/22	226/27	226/31
226/36	229/10	229/13	229/21	229/24	229/29	229/34
230/25	230/28	230/32	231/12	231/28	233/11	233/15
233/21	233/25	233/31	233/35	233/49	255/39	255/42
257/07	257/10	257/13	257/20	257/24	257/27	258/04
258/08	258/22	259/04	259/10	260/03	260/13	261/06
262/02	262/06	263/32	263/37	265/36	265/43	265/55
265/59	267/02	267/06	267/11	267/16	267/20	267/25
267/29	267/34	271/07	271/13	271/16	271/19	271/26
271/30	271/33	272/04	272/08	272/22	273/05	273/15

CATES	000000	2/14	2/24	12/01	339/01			
CATTX	026075	350/38	350/44	351/28	351/31	351/39	353/20	
CATX1	026265	353/17	354/15					
CATX2	026300	353/08	354/17					
CAWTX	025774	348/19	348/47					
CA.00	025311	339/10	339/16	340/27	341/04	341/30	342/05	349/34
CA.01	025335	339/16	340/05					
CA.02	025432	339/17	341/31	341/33	342/05			
CA.03	025510	343/22	344/04	344/24				
CA.04	026103	348/21	351/03	351/14	352/10	352/20		
CA.05	026153	350/12	352/02					
CA.10	026071	350/33	350/40					
CA.1A	025356	340/09	340/16	340/22				
CA.1B	025365	340/18	340/21	341/03				
CA.1C	025375	341/03	341/11	341/27				
CA.1X	025343	340/11	340/25					
CA.2C	025460	342/23	342/26	343/05	343/29	351/42		
CA.2L	025451	342/24	342/32					
CA.2R	025500	342/29	343/26					
CA.3	025504	343/19	343/32					
CA.3A	025537	344/06	344/22	345/01				
CA.3R	025655	345/02	347/05	352/27				
CA.6	025505	342/17	343/33					
CA.BE	026057	349/10	350/23	350/26				
CA.BK	025604	343/21	345/07	345/11	345/42	346/21	347/07	347/10
		348/34	350/46					
CA.BL	025605	345/20	345/43	346/04				
CA.CA	025620	345/55	346/25	347/13	348/25	348/28	348/32	350/48
CA.CX	026101	349/25	350/06	350/09	350/10	350/48		
CA.DX	026073	350/42	351/04	351/12	352/11	352/12		
CA.EK	025654	346/31	347/27	352/23				
CA.ID	025723	341/39	348/03	348/06				
CA.K1	025426	341/07	341/37					
CA.K2	025427	341/08	341/38					
CA.K3	025430	340/14	341/09	341/39	344/20			
CA.KK	025603	342/19	342/30	343/17	345/41			
CA.NR	026072	350/26	350/41					
CA.RC	026045	349/33	350/11	350/15				
CA.RE	026053	349/24	350/04	350/22				
CA.RI	026031	349/31	350/03					
CA.RK	025621	345/56	346/22	347/11	348/30	348/35	350/49	
CA.TT	025602	344/04	344/28	345/13	345/40	347/05		
CA.WB	026014	347/18	347/34	349/13	349/23			
CA.WE	025775	348/18	349/05					
CA.WI	025737	346/16	348/17	349/26				
CA.WM	025746	348/20	348/25					
CA.WN	025761	348/36	349/30					
CA.X2	026260	354/07	354/10					
CA.XI	026255	339/22	339/23	343/31	351/40	353/31	353/34	353/37
		354/03	354/06					
CA.XX	025503	342/10	342/13	343/31	344/25			
CB10K	007557	122/13	122/20					
CB17	007554	122/08	122/17					
CB37	007556	122/12	122/19					
CB400	007555	122/09	122/18					
CBBG2	007514	119/44	121/40					
CBLIM	002530	46/46	57/06					
CBLWR	002561	57/16	57/18	57/22	57/23	57/31		

CBPL2	007515	119/16	119/33	121/41				
CBRDS	000000	2/03	2/24	8/20	18/03	117/01		
CBTXT	007524	121/06	121/48					
CBUPR	002562	57/17	57/24	57/32				
CB.00	007305	117/12	117/18					
CB.01	007324	117/18	118/02					
CB.02	007327	117/19	118/07					
CB.03	007423	118/08	120/04					
CB.04	007446	120/13	120/30	121/17				
CB.05	007537	121/21	121/25	122/03				
CB.2A	007347	118/21	118/24	119/04				
CB.2C	007357	119/14						
CB.2L	007343	118/22	118/26					
CB.37	007503	118/16	121/31					
CB.8G	007506	119/26	121/34					
CB.EC	007455	117/24	117/25	121/05				
CB.EN	007507	119/42	120/19	121/35				
CB.ER	007453	120/21	121/02	125/39				
CB.ES	007513	118/13	121/16	121/22	121/39			
CB.F1	007665	125/07	125/20					
CB.F2	007670	125/10	125/14					
CB.F3	007675	125/08	125/15					
CB.F4	007703	125/18	125/21					
CB.F5	007707	125/23	125/25					
CB.FA	007661	121/46	125/03					
CB.L2	007374	119/27	119/32					
CB.L3	007415	119/45	119/50					
CB.LC	007505	119/23	119/38	120/08	121/10	121/26	121/33	
CB.PL	007504	119/14	121/27	121/32				
CB.RL	007371	119/24	121/28					
CB.SE	007512	120/11	121/02	121/11	121/38			
CB.TI	007510	120/10	120/30	120/31	121/36			
CB.TK	007502	118/03	118/07	120/04	120/26	121/09	121/23	121/30
CB.TS	007516	120/05	121/42					
CB.WK	007724	125/26	125/38					
CB.X1	007441	118/11	120/22	120/25	121/14	122/07	122/16	
CB.X2	007470	120/33	121/19					
CCTR	002632	58/35	58/38	58/41	58/48	58/53		
CDGLC	002756	61/10	61/42					
CDISP	002706	32/02	60/01	61/02				
CDIST	002736	61/14	61/26					
CD.40	003044	61/31	62/40					
CD.44	003045	61/35	62/41					
CD.EX	003014	60/01	60/13	62/13	74/45			
CD.IN	003036	60/13	61/44	62/32				
CD.LA	003037	60/01	60/04	60/05	61/43	62/15	62/16	62/25
		62/33	73/11	74/46				
CD.LP	003040	60/01	60/05	61/48	62/34	73/09	74/47	
CD.PP	003041	60/01	60/12	61/58	62/35			
CD.S0	003031	60/01	60/06	61/02	62/10	62/27	103/24	
CD.S1	003032	60/01	60/07	61/03	62/11	62/28	103/25	
CD.S2	003033	60/01	60/08	61/04	62/12	62/29	103/26	
CD.S3	003034	60/01	60/11	61/05	62/23	62/30	73/12	103/27
CHA?3	004767	89/43	89/46	89/50				
CHC?T	004757	88/41	88/44	89/30	89/37	90/05	90/40	
CHECK	007642	105/24	124/10	124/13	124/15			
CHPAG	002341	54/16	54/25	54/38	56/25			
CHR?T	005011	89/37	89/45	89/53	90/13			

ERPAC	000076		32/17	139/18				
ERPAD	000075		32/16	139/23				
ERR1	007725		124/17	125/39				
ERRET	000107		32/26	116/18	141/14	141/33	142/11	142/17	142/23
			143/31	144/31	145/24	145/42	146/26	147/22	147/45
			148/20	148/41	149/17	149/39	151/16	151/37	152/15
			152/34	153/37	154/15	154/35	155/16	155/47	156/22
			157/22	157/40	158/22	159/25	160/25	161/22	164/24
			164/31	164/38	165/21	166/20	167/10	167/16	167/22
			167/28	167/34	167/40	167/46	167/52	167/58	168/04
			169/11	169/17	169/23	169/29	169/35	169/41	169/47
			169/53	169/59	171/18	171/39	172/16	172/35	173/22
			173/46	174/17	175/17	175/38	176/17	176/38	177/10
			177/20	188/12	191/37				
ERRK1	003465		72/17	72/36	73/18				
ERRK2	003460		72/11	72/21	73/12				
ERROC	000103		32/22	60/14	103/13	103/20	121/13	132/16	139/15
			184/15	184/24	184/33	184/48	210/13	210/19	210/24
			226/27	226/36	233/21	233/31	233/49	267/25	267/34
			293/26	293/35	295/09	318/26	318/35	319/29	337/15
			337/27	353/15	353/27				
ERROE	003775		32/22	78/19					
ERROH	003562		32/21	75/04	75/05				
ERROI	000102		32/21	60/09	103/02	121/04	132/07	139/06	184/06
			210/03	226/18	233/11	267/16	293/17	318/17	337/06
			353/06						
ERROR	000025	MC	116/08	116/18	141/10	141/29	142/07	142/13	142/19
			143/27	144/27	145/20	145/38	146/22	147/18	147/41
			148/16	148/37	149/13	149/35	151/12	151/33	152/11
			152/30	153/33	154/11	154/31	155/12	155/43	156/18
			157/18	157/36	158/18	159/21	160/21	161/18	164/20
			164/27	164/34	165/17	166/16	167/06	167/12	167/18
			167/24	167/30	167/36	167/42	167/48	167/54	167/60
			169/07	169/13	169/19	169/25	169/31	169/37	169/43
			169/49	169/55	171/14	171/35	172/12	172/31	173/18
			173/42	174/13	175/13	175/34	176/13	176/34	177/06
			177/16						
ERRRT	003376		32/26	72/03					
ERRTX	000073		32/14	114/11	114/19	114/24	114/26	114/45	121/08
			132/11	139/10	184/10	184/19	184/28	184/37	210/08
			211/32	226/22	226/31	233/15	233/25	233/35	255/39
			267/20	267/29	289/11	293/21	293/30	295/04	314/11
			318/21	318/30	319/24	337/10	337/19	353/10	353/19
			362/34						
ERSS0	003753		77/09	77/14	77/25				
ERSV0	003462		72/13	72/18	72/31	72/42	73/15		
ERSV2	003464		72/15	72/20	72/33	72/44	73/17		
ERTIT	003620		75/12	75/34					
ERTOT	000147		33/37	37/25	38/23	39/18	40/08	48/39	75/14
			75/16	362/46					
ERTXT	004015		32/14	78/38					
ERXXT	003770		78/08	78/46	78/49				
ER.C1	003747		76/47	76/58	77/04	77/17	77/21		
ER.C2	003752		76/45	77/18	77/24				
ER.K4	003750		76/59	77/22					
ER.S0	003754		75/04	75/37	77/26	78/08	78/19	78/28	78/38
			80/10	80/24					
ER.S1	003755		75/06	75/39	77/27	78/09	78/20	78/30	78/40

			80/11	80/18				
ER.S2	003756		75/07	75/41	77/28	78/10	78/21	78/32	78/41
			80/12	80/21	80/27				
ER.S3	003757		75/08	77/29	78/07	78/11	78/22	78/42	80/13
ER.TP	004027		78/39	78/48					
ESCRA	000065		32/08	118/23	129/14	136/11	180/24	215/24	229/21
			257/24	271/30	300/27	325/25	342/25		
ETODT	005700		102/24	104/24					
EXISM	027161		42/14	44/45	46/15	368/01			
EXSCR	003216		32/08	68/06					
EX.I1	003242		68/23	68/26	68/29	69/17			
F1WDS	012733		187/13	187/17	187/25	187/29	188/06	188/14	
F2WDS	013122		190/16	190/20	190/28	190/32	191/30	191/40	
FILL	007600		123/20	123/33					
FK1K	012306		181/36	182/29					
FLT01	012574		184/55	186/04	186/06	186/09			
FLT02	012741		184/56	189/04	189/06	189/09			
FLT1A	012734		186/08	186/16	186/24	187/03	188/15		
FLT1B	012735		186/13	186/25	187/04	188/16			
FLT1C	012736		186/14	186/26	187/05	188/17			
FLT1D	012724		187/12	187/16	187/24	187/28	187/36	187/40	188/05
FLT1E	012740		186/04	187/08	187/44	188/19			
FLT1F	012737		188/05	188/09	188/10	188/18			
FLT1L	012612		186/18	186/23					
FLT1M	012626		186/31	186/46					
FLT1T	012654		187/09	187/45					
FLT2A	013123		189/08	189/16	189/24	190/03	191/41		
FLT2B	013124		189/13	189/25	190/04	191/42			
FLT2C	013125		189/14	189/26	190/05	191/43			
FLT2D	013111		190/15	190/19	190/27	190/31	191/04	191/08	191/29
FLT2E	013127		189/04	190/09	191/20	191/45			
FLT2F	013126		191/29	191/34	191/35	191/44			
FLT2L	012757		189/18	189/23					
FLT2M	012774		189/32	189/47					
FLT2S	013121		190/11	190/36	190/39	191/13	191/38		
FLT2T	013023		190/10	190/37	191/21				
FLT2U	013061		191/01	191/14					
FOB	013626		206/02	206/23	206/27				
FP255	012307		181/09	181/37					
FPERR	012437		182/46	184/03					
FPESX	012515		184/51	184/58					
FPS03	012427		183/46	184/04	184/20				
FPTKX	012516		184/52	184/59					
FPTST	000000		2/07	2/24	9/15	20/14	178/01		
FP.01	012210		179/11	179/30					
FP.02	012224		179/12	180/03					
FP.03	012311		180/04	181/17	182/07				
FP.04	012404		183/18	183/23					
FP.05	012421		180/12	183/21	183/39				
FP.2A	012252		180/07	180/22	180/25	181/05			
FP.2C	012263		181/16						
FP.2L	012245		180/23	180/28					
FP.37	012304		180/17	181/05	181/34				
FP.40	012426		182/13	183/45					
FP.4A	012401		183/19	183/22	184/49				
FP.BG	012432		181/21	182/39	183/49	184/30			
FP.EC	012443		179/17	179/18	184/08				
FP.EN	012433		181/31	183/50	184/13	184/38			

INR?	005413	94/16	95/41	96/21				
INR?T	005277	94/01	94/51	96/20	96/34			
INS?	005217	93/36	96/18					
INS?0	005300	94/02	94/52	96/30				
INS?1	005301	94/03	94/53	96/31				
INS?2	005302	94/04	94/54	96/32				
INS?3	005303	94/17	94/55	96/33				
INS?C	005304	94/06	94/56	96/28				
INT?	005204	92/03	92/49	94/08	96/36			
IN.P?	027070	366/03	366/05					
IN?17	005432	96/07	96/37					
IORLM	006056	105/06	105/23					
IOTST	000001	2/22	2/24	12/23	26/01	33/28	40/08	51/22
		97/47	363/30					
IOTX3	006263	103/09	106/06					
IOTX4	006272	103/16	106/07					
IOTX5	006057	103/05	105/24					
IOV40	005761	102/38	103/28					
IOVAL	005654	31/43	102/04					
IOVPR	005726	102/43	103/01	104/36	104/42	104/45		
IOV.0	006044	102/04	102/05	102/26	103/10	104/05	104/06	105/13
IOV.1	006045	102/04	102/07	102/27	103/11	104/05	104/08	105/14
IOV.2	006046	102/04	102/09	102/28	103/12	104/05	104/10	105/15
IOV.3	006043	102/04	102/11	102/29	103/17	104/05	104/12	105/05
		105/11	105/12					
IOV.4	006047	102/13	103/18	104/14	105/16			
IOV.7	006055	102/21	104/22	105/22				
IOV.A	006050	102/18	103/01	104/19	105/17			
IOV.B	006051	102/19	104/20	105/18				
IOV.C	006052	102/20	104/21	105/19				
IOV.E	006033	60/23	102/34	103/23	104/28	105/04		
IOV.R	006053	102/36	103/19	104/30	105/20			
IO.TX	006076	102/42	105/25					
IPAT	007572	123/14	123/39					
IPAT1	007576	123/18	123/35					
IPDE?	000130	33/18	95/08					
IPRCD	004425	83/22	84/47					
IPTB?	005221	93/38	95/02	95/06	95/28			
IQUAL	004426	83/16	84/48					
IRUN	004424	83/19	84/46					
ISTZM	001455	43/08	43/16					
ITPS?	005367	95/13	95/22	95/57				
ITYP?	005370	95/58	96/04	96/06				
IVODT	006054	102/25	102/30	105/21				
IWAIT	003760	75/22	77/30					
IXTOR	013314	194/05	194/24	195/06	195/25	196/06	196/25	197/06
		197/25	197/39	198/05	198/24	199/06	199/24	200/06
		200/24	201/06	201/25	202/06			
IZOC?	000131	33/19	95/21					
J177	001144	36/11	36/50					
J23	001146	36/15	36/52					
J42	001143	36/21	36/49					
JPNXT	004360	82/26	82/32	84/10				
JXWT	004265	83/04	84/22	85/08				
K17	002636	58/32	58/57					
K1777	003035	60/01	61/45	62/31				
K1K	001555	44/23	44/47					
K37	001552	44/31	44/44	45/01	45/40			

K37C	002563	57/13	57/33					
K4	005525	97/44	98/25					
K40	004516	84/28	86/10					
KAVLM	002560	57/19	57/30					
KAVMP	001640	46/05	46/16					
KEXMP	001637	46/04	46/15					
KEY6T	007207	79/04	115/42					
KEY6W	004033	55/49	77/30	78/05	79/01			
KEYLP	004037	79/05	79/27					
KEY.0	004062	79/01	79/20	79/24				
KEY.3	004063	79/02	79/21	79/25				
KEY.C	004064	79/17	79/26					
KLETB	001666	46/36	46/45					
KLSTB	001665	46/35	46/44					
KLZMX	001454	43/07	43/15					
KMX	001641	46/06	46/17	46/18				
KNMAX	001661	46/27	46/40					
KSTES	001664	46/33	46/43					
KSTSS	001663	46/32	46/42					
KUP32	001662	46/29	46/41					
KXIST	001553	44/33	44/45	45/01	45/33			
KY.CK	004066	79/26	79/28					
LASTI	000151	33/39	36/07	36/13	40/11	79/05	79/12	79/19
		82/01	82/09	94/10	95/46	95/54	113/35	365/10
		365/27	366/02					
LAUTO	001034	34/14	34/35	39/33	114/06			
LCALL	000000	MC 34/41	62/36	62/38	116/18	118/09	118/14	118/18
		118/22	119/04	119/10	120/22	120/27	120/34	121/03
		121/07	121/12	122/03	129/10	129/13	131/39	131/43
		132/06	132/10	132/15	136/05	136/10	136/15	137/24
		138/08	138/12	138/26	138/38	138/43	139/05	139/09
		139/14	139/17	139/22	141/02	141/05	141/13	141/16
		141/21	141/24	141/32	141/35	142/03	142/10	142/16
		142/22	142/25	143/03	143/06	143/30	143/33	144/03
		144/06	144/30	144/33	145/03	145/06	145/23	145/26
		145/31	145/34	145/41	145/44	146/03	146/06	146/25
		146/28	147/05	147/08	147/21	147/24	147/29	147/32
		147/44	147/47	148/03	148/06	148/19	148/22	148/27
		148/30	148/40	148/43	149/03	149/06	149/16	149/19
		149/24	149/27	149/38	149/41	151/02	151/05	151/15
		151/18	151/23	151/26	151/36	151/39	152/03	152/14
		152/17	152/22	152/33	152/36	153/03	153/06	153/36
		153/39	154/02	154/05	154/14	154/17	154/22	154/25
		154/34	154/37	155/03	155/06	155/15	155/18	155/22
		155/25	155/46	155/49	156/03	156/06	156/21	156/24
		157/03	157/06	157/21	157/24	157/29	157/32	157/39
		157/42	158/03	158/06	158/21	158/24	159/05	159/08
		159/24	159/27	160/03	160/06	160/24	160/27	161/03
		161/06	161/21	161/24	164/02	164/05	164/23	164/30
		164/37	164/40	165/03	165/20	165/23	166/03	166/19
		166/22	167/03	167/09	167/15	167/21	167/27	167/33
		167/39	167/45	167/51	167/57	168/03	168/06	169/03
		169/10	169/16	169/22	169/28	169/34	169/40	169/46
		169/52	169/58	170/01	171/03	171/06	171/17	171/20
		171/25	171/38	171/41	172/03	172/06	172/15	172/18
		172/23	172/26	172/34	172/37	173/03	173/21	173/24
		173/29	173/32	173/45	173/48	174/03	174/06	174/16
		174/19	175/03	175/06	175/16	175/19	175/24	175/27

	175/37	175/40	176/03	176/06	176/16	176/19	176/24
	176/27	176/37	176/40	177/02	177/09	177/19	177/22
	177/25	180/10	180/15	180/19	180/23	181/07	181/11
	182/10	182/16	182/47	183/14	183/19	183/41	184/05
	184/09	184/14	184/18	184/23	184/27	184/32	184/36
	184/47	186/18	187/46	188/11	189/18	191/22	191/36
	194/02	194/16	194/21	194/35	195/03	195/17	195/22
	195/36	196/03	196/17	196/22	196/36	197/03	197/17
	197/22	197/36	198/02	198/16	198/21	198/35	199/03
	199/16	199/21	199/35	200/03	200/16	200/21	200/35
	201/03	201/17	201/22	201/36	202/03	202/17	205/02
	205/20	205/24	205/42	206/04	206/21	207/06	210/02
	210/07	210/12	210/18	210/23	210/26	211/31	211/41
	211/43	211/47	211/50	215/07	215/10	215/13	215/19
	215/23	215/26	216/05	216/09	216/33	217/04	217/11
	218/02	218/13	219/05	219/10	219/20	219/57	221/02
	221/06	222/38	224/35	224/42	224/54	224/58	226/01
	226/05	226/10	226/17	226/21	226/26	226/30	226/35
	229/09	229/12	229/20	229/23	229/28	229/33	230/24
	230/27	230/31	231/11	231/27	233/10	233/14	233/20
	233/24	233/30	233/34	233/48	255/38	255/41	257/06
	257/09	257/12	257/19	257/23	257/26	258/03	258/07
	258/21	259/03	259/09	260/02	260/12	261/05	262/01
	262/05	263/31	263/36	265/35	265/42	265/54	265/58
	267/01	267/05	267/10	267/15	267/19	267/24	267/28
	267/33	271/06	271/12	271/15	271/18	271/25	271/29
	271/32	272/03	272/07	272/21	273/04	273/14	274/02
	274/08	275/08	275/13	275/28	276/03	276/07	277/03
	277/13	279/19	279/53	289/10	289/18	289/21	289/24
	289/28	289/31	291/35	291/42	291/54	291/58	293/01
	293/05	293/10	293/16	293/20	293/25	293/29	293/34
	295/03	295/08	296/22	300/08	300/11	300/14	300/22
	300/26	300/29	301/03	301/07	301/21	302/05	302/15
	303/02	303/08	304/07	304/13	304/28	305/03	305/07
	306/03	306/13	308/21	308/57	314/10	314/18	314/21
	314/24	314/28	314/31	316/36	316/47	316/59	317/03
	318/01	318/05	318/10	318/16	318/20	318/25	318/29
	318/34	319/23	319/28	320/21	325/08	325/11	325/14
	325/20	325/24	325/27	326/05	326/09	326/26	327/11
	327/14	328/03	328/09	328/21	328/26	330/27	331/32
	331/37	337/05	337/09	337/14	337/18	337/26	338/03
	338/07	338/12	342/08	342/11	342/14	342/20	342/24
	342/27	343/05	343/09	343/26	344/10	344/13	345/03
	345/09	345/21	345/26	346/27	347/32	347/37	353/05
	353/09	353/14	353/18	353/26	354/03	354/07	354/12
	356/15	356/30	356/36	357/03	357/08	357/16	362/22
	362/33	362/38	362/41	362/44	362/49	364/50	
LCINT 005541	35/07	99/05	99/11				
LCRLF 000070	33/21	37/34	38/20	78/27			
LC.K1 005621	99/14	99/17	99/19	99/32	99/33	100/12	
LC.K2 005617	99/28	100/10					
LC.K6 005630	100/07	100/19					
LC.K7 005625	99/38	100/16					
LC.K8 005626	100/01	100/17					
LC.K9 005627	100/03	100/18					
LDCHL 006713	35/20	108/13	113/04				
LDCHM 006666	107/40	110/32	111/33	112/05			
LDC.0 006707	112/05	112/20	112/22				

LDC.1 006710	112/08	112/19	112/23				
LDC.A 006712	112/16	112/25					
LDC.B 006711	112/14	112/24					
LDMAP 003054	35/19	64/04	67/18	73/18	105/23		
LDMP2 003353	70/26	71/03					
LDSPR 001024	34/17	34/20	34/27	39/34			
LDS.1 001026	34/29	34/33					
LD.37 003374	71/15	71/20					
LD.K1 003375	71/08	71/21					
LD.L1 003357	71/07	71/17					
LD.S3 003373	71/03	71/18	71/19				
LGTPA 003157	65/12	65/34	66/03	66/08			
LIDIS 001033	34/28	34/29	34/31	34/34			
LILLI 005631	99/05	101/01					
LINKR 001010	31/11	31/13	34/10	34/39			
LINTD 005501	97/01	98/05	101/18				
LINTR 005433	97/01	97/12	100/10				
LK300 005620	97/01	97/36	98/34	99/06	100/11		
LLOOP 000105	32/24	141/17	141/36	142/26	143/34	144/34	145/27
	145/45	146/29	147/25	147/48	148/23	148/44	149/20
	149/42	151/19	151/40	152/18	152/37	153/40	154/18
	154/38	155/19	155/50	156/25	157/25	157/43	158/25
	159/28	160/28	161/25	164/41	165/24	166/23	168/07
	170/02	171/21	171/42	172/19	172/38	173/25	173/49
	174/20	175/20	175/41	176/20	176/41	177/23	194/17
	194/36	195/18	195/37	196/18	196/37	197/18	197/37
	198/17	198/36	199/17	199/36	200/17	200/36	201/18
	201/37	202/18	205/21	205/43	206/22		
LMEML 026750	117/10	117/14	128/05	128/09	135/03	135/07	179/03
	179/07	192/20	192/24	211/08	211/12	228/05	228/09
	255/08	255/12	268/08	268/12	297/09	297/13	322/07
	322/11	339/08	339/12	355/04	355/08	360/10	360/14
	364/03	364/07					
LMESS 000132	33/20	37/07	37/16	37/30	38/02	39/10	39/14
	48/31	60/02	75/20	75/29	75/33	75/35	76/02
	76/14	76/24	76/27	76/32	76/48	76/53	76/56
	78/47	79/03	80/14	80/39	87/06	102/37	102/41
	102/44	103/04	103/08	103/15	104/31	104/34	104/40
	104/43	113/36	115/14				
LMSKS 005523	97/01	97/37	98/23	98/37	99/27		
LM.37 003155	64/23	64/32	65/32				
LM.DN 003153	65/05	65/13	65/29	66/04	66/09		
LM.K1 003154	64/12	64/16	65/31				
LM.K2 003160	65/21	65/35	66/12				
LM.L1 003066	64/12	64/15	64/25				
LM.L2 003102	64/29	64/45					
LM.L3 003133	65/12	65/27					
LM.L4 003166	66/08	66/16					
LM.NM 003161	64/09	64/11	66/03				
LM.TM 003156	65/10	65/14	65/25	65/33			
LOC. 015360	5/05	18/38	33/04				
LOC.0 005527	97/19	98/16	98/22	98/27	101/19		
LOOP 000023 MC	116/05	141/15	141/34	142/24	143/32	144/32	145/25
	145/43	146/27	147/23	147/46	148/21	148/42	149/18
	149/40	151/17	151/38	152/16	152/35	153/38	154/16
	154/36	155/17	155/48	156/23	157/23	157/41	158/23
	159/26	160/26	161/23	164/39	165/22	166/21	168/05
	169/60	171/19	171/40	172/17	172/36	173/23	173/47

	174/18	175/18	175/39	176/18	176/39	177/21	194/15
	194/34	195/16	195/35	196/16	196/35	197/16	197/35
	198/15	198/34	199/15	199/34	200/15	200/34	201/16
	201/35	202/16	205/19	205/41	206/20		
LOOPL 003534	32/24	74/27					
LOPNL 000221	31/28	81/26	85/20				
LP002 026515	359/25	359/38	359/53				
LPCR2 026463	358/06	358/23					
LPCRF 026357	355/35	355/45	356/06	356/10	356/19	358/24	359/18
	359/31						
LPCRI 026502	359/14	359/33					
LPCRO 026375	356/08	356/18					
LPDEC 000072	33/24	38/16	38/19	39/17	39/20	114/14	
LPDTX 007262	115/15	115/46					
LPEND 026470	359/04	359/19					
LPFFO 026371	356/12						
LPFIL 026441	357/15	358/03	358/12	358/19	359/46	359/51	
LPGO 000175	33/46	117/11	117/13	128/06	128/08	135/04	135/06
	179/04	179/06	192/21	192/23	211/09	211/11	228/06
	228/08	255/09	255/11	268/09	268/11	297/10	297/12
	322/08	322/10	339/09	339/11	355/05	355/07	360/11
	360/13	364/04	364/06	368/23			
LPHIK 007053	114/12	115/23					
LPIDX 007057	114/40	114/42	115/11	115/27			
LPINT 026516	355/44	359/13	359/28				
LPLZM 007047	114/31	115/19					
LPOCT 000133	33/22	39/13	48/36	60/05	75/38	75/40	75/42
	76/05	76/07	76/17	76/19	76/35	76/37	77/10
	77/16	78/29	78/31	78/33	80/17	80/23	80/29
	80/42	80/44	80/46	87/18	87/46	103/07	
LPR10 007056	114/43	115/26					
LPR1E 007045	114/39	114/48	115/10	115/17			
LPR40 007055	115/08	115/25					
LPR4T 007250	115/20	115/45					
LPR60 006755	113/34	113/41					
LPR77 007054	115/06	115/24					
LPRCK 006754	113/38	113/40					
LPRGA 002032	49/23	49/32					
LPRGN 007052	114/09	114/30	115/17	115/22			
LPRGO 001776	48/20	49/03					
LPRL1 001731	48/12	48/44					
LPRL2 001772	48/14	48/46	49/15	49/28	49/31		
LPRLP 007004	114/30	115/18					
LPRSL 006757	35/12	113/31	114/07	114/20	114/46		
LPRT1 007060	114/10	115/28	115/31				
LPRT2 007121	114/21	115/35					
LPRT3 007132	114/23	115/37					
LPRT4 007050	114/25	115/20					
LPRT5 007160	113/37	115/40					
LPRTN 006743	113/31	114/35					
LPS1E 001767	48/18	48/23	48/26	48/42			
LPSTA 026440	357/10	357/21	359/50				
LPSV3 007051	113/33	113/39	114/07	115/21			
LPTTS 000000	2/15	2/24	12/15	25/01	355/01		
LPT.I 005652	100/19	101/22					
LP.00 026320	355/06	355/12	355/30	355/36	356/13	357/14	359/07
	359/21	359/25					
LP.01 026337	355/12	355/29					

LP.02 026360	355/13	356/03	356/23				
LP.10 026437	356/38	357/07	357/20				
LP.2A 026407	356/05	356/28					
LP.2B 026421	356/29	357/03					
LP.40 026404	356/25	357/11	359/48				
LP.50 026436	356/35	357/19					
LP.CK 026435	357/12	357/18	358/03	358/08	359/43	359/49	
LP.CR 026467	356/18	358/23	358/27				
LP.FF 026406	356/11	356/27					
LP.K1 026354	355/38	355/42					
LP.K2 026355	355/39	355/43					
LP.K3 026356	355/40	355/44					
LP.LF 026514	359/08	359/09	359/24	359/30			
LP.LK 026405	355/37	356/03	356/26	356/40	359/04	359/14	359/34
	359/40						
LP.TX 002120	48/32	50/33					
LP.XI 026374	355/18	355/19	356/15				
LP.ZK 026403	356/24	358/04	359/44				
LRANP 001726	35/18	48/09	48/41				
LRETP 002114	32/15	50/29					
LROGK 002060	47/09	48/48	49/54				
LRUNS 001022	34/20	35/13	35/24				
LR.K1 002056	47/13	48/11	49/04	49/52			
LR.S3 002057	48/09	49/51	49/53				
LSETB 001443	41/06	41/10	41/22	41/29	42/16	44/11	46/45
	100/12	213/51	256/59	281/01	286/34		
LSIZE 001457	35/08	43/23					
LSIZR 001503	43/33	43/36	43/39	43/43			
LSTAT 001150	35/22	37/03					
LSTRP 002063	35/21	50/03					
LST.3 001243	37/03	38/22	38/23	38/24	39/23		
LSXIT 001241	37/37	38/10	38/22	38/23			
LSYSE 027205	368/09	368/26					
LSYSR 001306	35/04	39/31	40/04				
LSYTB 001442	41/05	42/15	46/44	49/56			
LS.I1 002116	50/07	50/10	50/27	50/31			
LS.I2 002117	50/19	50/32					
LS.K2 001364	40/15	40/28	40/51	43/26			
LS.NN 001244	38/23	38/25	38/26				
LS.S3 001500	43/23	43/40	43/43	43/45	43/48		
LT8L1 001040	34/18	35/04	39/29				
LT8L2 001054	34/21	35/18					
LWSET 001365	35/06	41/04					
LWS.1 001373	41/10	41/24					
LWS.2 001403	41/18	41/21					
LWS.3 001422	41/33	41/36					
LW.C1 001432	41/09	41/23	42/07				
LW.K1 001431	41/04	42/06					
LW.K4 001433	41/16	42/08					
LW.K5 001434	41/25	41/26	41/27	41/28	42/09		
LW.K6 001440	41/30	42/13	42/14				
LW.K7 001441	41/31	42/14					
LW.K9 002061	49/10	49/55					
LZMAX 000156	33/45	38/23	43/15	49/52	115/19		
LZOCT 000071	33/23	37/09	37/19	37/33	38/14	75/32	77/03
	114/41						
L.SS3 002115	50/03	50/29	50/30				
M100 013654	206/01	206/51					

M20	013744	207/25	208/04	208/20				
M32SZ	001546	44/19	44/40					
M32TE	001550	44/12	44/20	44/37	44/40	44/42		
M400	011377	159/02	159/10					
M400.	016533	246/21	246/25					
MAPBB	004407	84/24	84/33					
MAPIT	004375	83/28	84/23					
MAP.A	004377	83/31	84/25					
MAP.B	004400	83/33	84/26					
MAP.E	004427	83/41	84/49					
MAP.S	004366	81/06	81/34	84/16	84/21	85/03	85/22	86/20
		105/22						
MAP.T	004367	83/36	84/17	84/51				
MCK	013466	197/41	203/02					
MDA00	013163	193/02	194/01					
MDA08	013321	197/38	198/01					
MDT1	013621	206/20	206/43	206/50				
MDT2	013627	206/14	206/19	206/28				
MDT3	013644	206/34	206/37	206/41				
MDTEX	014030	193/07	193/08	206/26	210/26			
MDTST	013600	206/01	206/24					
MEQ12	000000	2/02	3/36	3/37	16/17	16/18	42/13	42/14
		45/12	46/17	46/18	368/02	368/03	368/04	368/05
MERET	014003	209/23	209/29	209/34				
MERR	013771	202/24	209/08	209/23				
MES?	004666	33/20	56/05	88/29	88/42			
MES?R	005012	88/29	88/32	88/50	89/09	90/01	90/14	90/17
		90/26						
MG32S	001614	45/01	45/12	45/13	45/28	45/38		
MH17	020645	276/33	277/15	277/24	279/36			
MH1RL	021502	289/12	289/26	289/33	289/39			
MH1RR	021522	289/17	289/33					
MH24	020422	272/11	272/14	272/19	272/27	273/08	273/24	
MH248	020430	272/05	272/33					
MHBSF	021753	293/48	295/02					
MHCDS	021544	290/09	290/32					
MHCOM	021574	280/55	291/05	293/58	293/60			
MHCST	020711	272/29	275/45	275/46	278/13	280/29	295/18	
MHCSX	021771	293/24	295/18					
MHCTL	000010	278/15	278/24	296/48	296/49	296/50	296/51	
MHCYE	021412	287/02	288/01					
MHCYL	020714	275/53	276/29	278/14	278/16	278/24	279/07	279/51
MHCYQ	021321	284/51	286/09	286/32	287/31			
MHCYX	021353	286/19	287/01					
MHDBL	000011	278/10	278/11	290/24	296/48	296/49	296/50	296/51
MHDIB	021773	280/58	293/38	295/07	295/20			
MHDNM	020724	275/15	275/21	278/25				
MHDP0	022145	270/23	270/28	278/11	290/19	296/48		
MHDP1	022206	290/20	296/49					
MHDP2	022247	290/21	296/50					
MHDP3	022310	290/22	296/51					
MHDPT	021555	270/30	290/06	290/18	290/28			
MHDRE	022351	270/23	296/52					
MHDRV	020725	275/20	278/26	279/39				
MHDRX	021303	270/32	270/33	282/06	283/06	284/04	284/11	285/01
		286/14	288/09	288/39	288/41	288/46	288/48	288/50
		289/08	289/12	289/34	289/35	289/44		
MHDSK	000033	268/05	269/09	269/10	269/26	269/37	269/40	270/24

		279/50	279/52	280/02	280/03	280/11	280/24	280/30
		280/31	280/32	280/34	280/44	280/48	280/50	282/14
		282/16	282/18	282/21	282/28	283/12	283/14	283/16
		283/20	284/12	284/14	284/15	284/19	284/32	284/33
		284/36	284/37	284/43	284/44	284/47	286/09	286/17
		286/23	286/24	286/29	287/05	287/06	287/09	287/10
		287/22	287/26	288/10	288/12	288/15	288/16	288/23
		288/26	293/49	295/06	296/20	296/21	296/27	
MHDST	020710	272/28	274/27	278/12	295/17			
MHDSX	021770	293/23	295/17					
MHDW1	020717	274/05	274/28	278/19	291/04	293/59		
MHDW2	020720	274/06	274/29	278/20				
MHDW3	020721	274/07	274/32	278/21				
MHDW4	020722	274/14	274/33	278/22				
MHERL	021636	291/40	295/11	296/03				
MHERR	021700	291/10	291/15	291/20	291/25	291/30	293/15	293/56
MHFOK	021072	280/43	280/50					
MHFSB	021223	270/21	284/04	288/52				
MHGD3	021554	290/04	290/16	290/17	290/27			
MHGDS	021536	275/06	284/01	289/13	290/03			
MHHDP	020677	278/02	279/25					
MHHDQ	021153	283/05	283/24					
MHHIK	021664	291/47	292/05					
MHIDX	020733	274/45	275/44	277/22	278/32	293/14	296/14	
MHLHD	020701	278/04						
MHLSC	020676	275/23	278/01	278/10	290/23			
MHMK1	020325	269/20	270/10	270/23				
MHNBS	021735	293/41	293/46	293/49				
MHNCT	020226	269/03	269/15	270/14				
MHNCY	020706	276/06	278/09					
MHNHD	020702	278/05						
MHNSC	020700	278/03	279/16					
MHRCI	022047	296/04	296/18	296/26	296/36			
MHRCL	022023	293/36	293/37	296/04	296/12			
MHRDO	021046	280/27						
MHRSC	020723	274/16	277/08	277/20	278/23			
MHRSK	021055	280/22	280/34					
MHRTX	021531	288/45	289/09	289/20	289/38	289/40		
MHRWI	021061	280/25	280/41					
MHSAV	021451	288/20	288/21	288/32	288/33	288/35		
MHSCQ	021110	282/05	282/30	288/03				
MHSCY	020703	273/02	277/12	278/06				
MHSDS	021564	288/40	290/27					
MHSE0	021420	280/56	288/08	288/17	288/22	288/42	289/36	
MHSEC	020715	272/31	274/05	274/06	274/07	274/14	274/16	274/27
		274/28	274/29	274/32	274/33	274/36	277/09	277/21
		278/17	279/10					
MHSKB	020704	276/17	278/07	279/04	279/06	295/14	296/14	
MHSKI	021032	279/47	280/15	280/16	280/21			
MHSKM	020705	276/11	278/08					
MHSKX	021765	293/43	295/14					
MHSTA	020732	278/31	280/04	295/19				
MHSTD	021454	283/41	283/42	288/39	288/47			
MHSTE	021722	280/54	293/38	296/34				
MHSTX	021772	293/31	293/50	293/54	295/19			
MHST.	021076	280/35	280/45	280/46	280/51	280/54	282/05	282/19
		283/05	283/17					
MHTBA	020332	269/21	270/28					

0391 N3MRT

MHTBL	020707	275/27	275/42	276/19	278/11			
MHTX1	021774	293/19	295/21					
MHTX2	022007	293/28	295/23					
MHTX3	022066	295/02	296/42					
MHTXA	020334	269/32	270/30					
MHTXB	020335	270/08	270/09	270/31				
MHTXR	022116	289/40	296/45					
MHWDN	021432	280/57	284/34	284/45	286/27	287/07	287/24	288/13
		288/20						
MHWEL	021563	290/09	290/24					
MHWEP	021562	290/23						
MH.00	020203	268/10	268/16	269/04	270/11	271/03	279/55	289/04
		293/13						
MH.01	020231	268/16	269/08					
MH.02	020341	268/17	271/03	271/24	271/35	289/03	289/42	292/03
MH.03	020534	272/20	275/08	275/22	275/26	275/39		
MH.04	020736	277/07	277/11	279/06				
MH.05	020734	274/43	279/04					
MH.0X	021015	279/11	279/48	279/55	280/05	280/26	280/36	280/47
		280/52						
MH.14	021103	280/59	282/15	282/22	282/31	283/13		
MH.16	021106	281/02	282/07	283/07				
MH.17	021351	286/15	286/33	287/04				
MH.1L	020262	269/34	269/49					
MH.1R	021474	270/22	289/05	289/06				
MH.25	021751	293/51	293/57	294/01				
MH.2C	020375	271/28	271/31	272/03	272/24	293/12		
MH.2L	020366	271/29	271/37					
MH.2R	020415	271/34	272/21					
MH.37	021663	291/51	292/04					
MH.3G	020505	274/26	277/26	296/40				
MH.3L	020611	276/03	276/10	276/22	276/25	276/28		
MH.3S	020433	273/04	273/09	273/17	273/28	276/31		
MH.44	020531	275/01	275/12					
MH.4A	020650	275/55	277/03					
MH.5G	021006	279/47	296/41					
MH.5S	020756	279/23						
MH.B6	021221	283/43	284/05	284/09	284/10			
MH.BF	020333	269/18	270/29	271/09	285/03	295/12		
MH.BK	020644	272/30	276/32	279/31				
MH.BX	021763	293/39	295/12					
MH.CO	020730	278/29	279/09	280/27	295/15			
MH.CX	021766	293/32	293/42	295/05	295/15	296/16		
MH.IS	021020	270/26	280/03	280/09				
MH.K1	020326	270/15	270/24					
MH.K2	020327	270/16	270/25					
MH.K3	020330	270/17	270/26					
MH.KK	020421	271/02	272/15	272/26				
MH.ND	020427	269/31	269/44	270/12	272/32	273/16	274/04	274/13
		274/26	289/41					
MH.OC	020731	278/30	279/17	279/22	279/41	279/49	280/28	295/16
MH.OX	021767	293/33	295/16	296/13				
MH.S3	020331	270/18	270/27	289/43				
MH.SA	021764	293/15	293/22	295/13				
MH.SC	020727	278/28						
MH.SX	021675	291/41	291/46	293/09	293/12	296/05	296/06	296/31
MH.WC	020527	274/11	274/44					
MH.WK	020726	273/03	274/36	277/25	278/27	279/46	293/59	

0392 N3MRT

MH.X2	021670	293/05	293/08					
MH.X3	021644	291/47						
MH.X4	021653	291/54	291/57					
MH.X5	021656	291/58	292/01					
MH.XI	021665	268/22	268/23	272/25	291/50	291/53	293/01	293/04
MH.XX	020420	271/08	271/17	272/25				
MM3E1	010201	130/20	131/22					
MM3E2	010202	130/22	131/23					
MM3E3	010203	130/25	131/24					
MM3E4	010204	130/28	131/25					
MM3E5	010205	130/31	131/26					
MM3E6	010206	131/08	131/27					
MM3E7	010207	131/10	131/28					
MM3E8	010210	131/13	131/29					
MM3ER	010200	130/11	131/21					
MMK10	010314	133/11	133/20					
MMLOC	010316	132/04	132/14	133/22				
MMTXT	010317	132/09	133/23					
MM.00	010042	128/07	128/14					
MM.01	010062	128/14	128/33					
MM.02	010067	128/15	129/04					
MM.03	010115	129/33						
MM.2A	010100	129/06	129/15	129/18	132/20			
MM.2L	010076	129/13						
MM.3A	010123	128/37	128/38	130/04	130/14			
MM.3B	010136	130/17	130/34					
MM.3C	010160	131/03	131/16					
MM.E1	010243	131/22	132/23					
MM.E2	010247	131/23	132/28					
MM.E3	010254	131/24	132/35					
MM.E4	010260	131/25	132/41					
MM.E5	010265	131/26	132/48					
MM.E6	010271	131/27	132/54					
MM.E7	010275	131/28	133/02					
MM.E8	010302	131/29	133/09					
MM.EC	010230	128/20	128/21	132/08	133/12			
MM.EN	010212	128/37	131/31					
MM.ER	010223	131/21	132/02	132/26	132/32	132/38	132/45	132/51
		132/57	133/06					
MM.ES	010214	128/34	129/04	129/09	129/35	131/33	131/38	132/19
MM.K2	010306	132/31	133/14					
MM.K3	010307	132/37	133/15					
MM.K4	010310	132/44	133/16					
MM.K5	010311	132/50	133/17					
MM.K6	010312	132/56	133/18					
MM.K7	010313	133/05	133/19					
MM.L2	010107	129/25	129/30					
MM.LC	010066	128/38	129/19					
MM.PL	010065	128/37	129/21					
MM.S3	010177	130/04	131/17	131/20				
MM.SA	010211	129/23	130/05	130/17	131/04	131/30		
MM.SE	010213	131/32	132/02	132/13	132/23	132/28	132/35	132/41
		132/48	132/54	133/02	133/09			
MM.TK	010315	132/05	132/12	133/21				
MM.X1	010215	129/12	129/37	131/37	131/42	132/17		
MM.X2	010222	129/38	131/43					
MODUA	007714	123/13	125/30					
MPF32	001622	45/01	45/14	45/17	45/39	45/44		

```
MT.XX  024474        325/10  325/13  326/31  327/26
MUDI0  013133        192/22  193/01  193/18  193/24
MUDI1  013152        193/01  193/17
MUDVT  000000          2/08    2/24    9/41   20/33  192/01
MULCK  000062  MC    192/07  194/10  194/29  195/11  195/30  196/11  196/30
                     197/11  197/30
MULER  000070  MC    192/16  203/26  205/17  206/48
MULHE  014032        209/25  210/29
MVDSK  000000          2/11    2/24   11/01   23/01  268/01
MVETA  001623         35/09   46/03   46/14
MXDSK  000001          2/18    2/24
MXTES  000001          2/21    2/24
N100   004517         86/06   86/11
N101   004267         83/06   83/29
N105   004522         83/39   86/14
N115   004423         83/26   84/45
N12    004421         82/24   84/43   86/04
N121   004422         83/17   84/44
N124   004266         83/05   83/34
N136   004142         81/04   83/10
N15    004271         82/30   83/08
N177   004272         82/07   83/09
N40    004263         82/33   83/02   84/04
N57    004420         82/15   83/42   84/42   85/36
N60    004340         81/10   83/48
N67    004262         82/12   83/01
N7     004145         81/07   81/08
N75    004143         81/05   83/14
N77    004520         83/53   86/12
ND12   017615        262/28
ND16   017426        258/11  258/14  258/19  258/28  259/06  259/07  259/14
ND17   017617        262/30
ND20   017616        262/29
ND252  017425        258/05  258/27
ND37   017614        262/27  263/09
ND3TX  017255        255/37  255/43  255/45  255/53
NDCNX  017701        263/26  263/40  264/15  264/32
NDCOM  017745        264/22  265/04
NDCST  017622        258/30  262/33  263/25  267/48
NDCSX  020104        267/23  267/48
NDCTB  020106        255/55  256/58  262/31  267/51
NDCYL  017624        259/21  260/16  261/22  262/20  262/35
NDDNX  017700        263/24  263/39  264/13  264/31
NDDRV  017623        259/26  260/20  261/26  262/34  263/23  267/45
NDDSK  000020        255/05  256/08  256/09  256/16  256/17  256/22  256/25
                     256/26  256/54  263/27  263/30  263/35  264/02  264/03
                     264/10  264/23  264/30  264/36  264/37  267/39  267/40
NDDST  017621        258/29  260/24  262/32  265/06  267/47
NDDSX  020103        267/22  267/47
NDDW1  017626        260/04  260/25  261/28  262/37  265/07
NDDW2  017627        260/05  260/26  261/30  262/38  265/12
NDDW3  017630        260/06  260/29  261/32  262/39  265/16
NDDW4  017631        260/14  260/30  261/34  262/40  265/22
NDERR  020050        265/10  265/15  265/20  265/25  265/30  267/14  267/43
NDETB  017347        256/15  256/59
NDHIK  020035        265/47  266/04
NDIDX  017637        260/07  261/35  262/46  265/41
NDMAX  017337        256/14  256/51
```



```
NDMK1  017341        255/46  256/53
NDRDI  017734        263/21  264/29  264/38
NDSEC  017625        258/32  260/18  261/24  262/36  263/08
NDSTA  017636        262/45  264/04  267/49
NDSTE  020072        264/26  267/38
NDSTX  020105        267/30  267/41  267/42  267/49
NDSUR  017340        256/41  256/52  262/23
NDTBA  017346        256/58
NDTBL  017620        261/19  262/10  262/31
NDTX1  020133        267/18  267/52
NDTX2  020146        267/27  267/54
NDTX3  020161        255/53  267/56
NDWRI  017741        263/33  264/35
ND.00  017216        255/10  255/16  255/44  256/33  256/44  257/03  263/38
ND.01  017262        255/16  256/05  256/13  256/27
ND.02  017351        255/17  255/52  255/54  257/03  257/18  257/29  266/03
ND.03  017526        258/20  261/05  261/16
ND.04  017640        261/37  263/05
ND.05  017641        260/40  263/06
ND.0X  017677        263/22  263/34  263/38  264/05  264/24  267/09
ND.1E  017317        256/11  256/29  256/35
ND.1L  017273        256/15  256/31
ND.1R  017236        255/36  256/43
ND.2C  017400        257/22  257/25  258/03  258/24  267/12
ND.2L  017371        257/23  257/31
ND.2R  017420        257/28  258/21
ND.37  020036        265/51  266/05
ND.3G  017503        260/24
ND.3L  017566        262/01  262/08  262/13  262/16  262/19
ND.3S  017434        259/03  259/19  262/22
ND.5G  017660        263/21
ND.5S  017647        263/12
ND.5W  017673        263/29  263/33
ND.8K  017613        258/31  259/11  262/04  262/24  263/12
ND.CO  017634        262/43  263/07  263/28  267/46
ND.CX  020101        267/31  267/45
ND.IS  017703        256/56  264/03  264/08
ND.K1  017342        256/45  256/54
ND.K2  017343        256/46  256/55
ND.K3  017344        256/47  256/56
ND.KK  017424        257/02  258/15  258/26
ND.ND  017336        255/40  256/35  256/50
ND.OC  017635        262/44  263/15  264/20
ND.OX  020102        267/32  267/46
ND.S3  017345        256/06  256/34  256/49  256/57
ND.SA  020100        267/14  267/21  267/44
ND.SC  017633        262/42
ND.SE  017732        264/19  264/26  264/27
ND.SX  020047        265/46  267/12
ND.WC  017524        260/39  260/41
ND.WK  017632        259/02  260/33  262/41  263/20  265/26
ND.X0  020010        265/40  267/35
ND.X2  020042        267/05  267/08
ND.X3  020016        265/47  267/36
ND.X4  020025        265/54  265/57
ND.X5  020030        265/58  266/01
ND.XI  020037        255/22  255/23  258/25  265/50  265/53  267/01  267/04
ND.XX  017423        257/08  257/11  258/25
```

```
NEXTT 000003  MC   34/44  117/09  128/04  135/02  179/02  192/19  211/07
                  228/04  255/07  268/07  297/08  322/06  339/07  355/03
                  360/09  364/02
NMPTX 007114       114/18  115/31  115/33
NOMAP 004332        83/25   83/42
NOTRL 003653        76/13   76/20
NOVA  000001         2/02    2/06    2/24   14/24   45/01   53/16   60/01
                    97/01   99/17   99/20  104/31  105/24  115/29  115/31
                   116/18  368/20
NPROG 001725        47/08   47/10   47/32   47/35
NTODT 005707       102/23  102/31
NTOPN 004440        85/02   85/07
NTPA  003267        69/21   69/26
NVDSK 000000         2/10    2/24   10/18   22/11  255/01
NWSTK 002360        54/11   55/04   56/13
NXDSK 000001         2/17    2/24
NXTLO 004432        83/12   84/10   85/01
OAC   013735       204/08  205/28  206/15  206/28  206/44  207/19  208/13
                   210/09
OAC0  011616       164/07  164/32  166/25  171/08  171/12
OAC1  011617       164/08  164/25  166/26
OAC2  011620       164/09  164/18  166/27
OAC.0 016743       249/04  249/21  250/18  251/27  251/32
OAC.1 016744       249/05  249/18  250/19
OAC.2 016745       249/06  249/15  250/20
OCT?A 005104        89/04   91/19
ODT   004154        31/20   36/44   81/16   96/35  365/36
OKAC  013660       203/20  207/01  207/32  210/20
OKMD  013662       203/22  207/03  207/34  210/22
OKMQ  013661       203/21  207/02  207/33  210/21
OMD   013737       204/10  205/30  206/12  206/30  206/46  207/21  208/15
                   210/11
OMQ   013736       204/09  205/29  206/16  206/29  206/45  207/20  208/14
                   210/10
OPNLO 004450        83/44   85/16
OPRIN 004207        81/14   82/01   82/45   83/52
OP.EN 000220        31/27   81/25   81/28   81/31   85/01   85/17
OTOTL 007511       120/18  121/37  125/38
P17?7 005203        92/28   92/39   92/48   94/11
P37?7 004756        88/34   89/36
PAC?1 004754        88/30   88/45   89/10   89/34   90/02   90/18   90/36
                    91/06
PAC?2 004755        88/31   88/46   89/03   89/06   89/35   90/03   90/27
PATT  007727       123/16  124/05  125/41
PC1?1 005010        89/40   90/12
PC1?2 005120        90/06   90/15   90/28   91/32   94/19   95/09
PC1?5 005205        92/44   92/50   94/13
PC4?0 005101        90/41   91/16   92/01   92/41
PC6?0 005100        89/02   89/23   89/27   91/15   94/29
PC7?  005006        89/48   90/10
PCRLF 000070        32/11   33/21   77/02   87/35  114/27  115/16  211/42
                   289/19  314/19
PC.40 005371        95/04   95/59
PD10K 014242       213/41  214/10
PD1RL 014110       211/35  211/56
PD1RR 014125       211/39  211/52
PD253 014346       216/07  216/44
PD32  014341       216/15  216/22  216/26  216/39  217/07  217/08  217/13
```



```
                   217/16
PD37  014572       221/28  222/10
PD3GR 014452       218/33  218/49  218/50
PD3TX 014144       211/30  211/57  211/59  212/07
PD400 015115       227/38  227/48
PDCMD 015016       222/33  226/13  227/22
PDCMQ 015056       222/41  227/17
PDCOM 014716       223/14  224/07  224/08  224/26
PDCQL 015065       227/24  227/41
PDCQR 015110       227/17  227/42  227/43
PDCST 014575       216/41  221/31  227/47
PDCTB 015172       212/09  221/50  221/51  221/52  221/53  227/59
PDCYL 014577       216/48  218/18  219/41  221/21  221/33
PDDNM 014570       219/12  219/35  221/26
PDDRV 014576       217/03  218/22  219/45  221/32  227/09
PDDSK 000026       211/05  213/08  213/10  213/11  213/25  213/26  213/27
                   213/28  213/29  214/12  222/37  223/01  223/03  223/09
                   223/15
PDDST 014574       216/40  218/34  221/30  224/09  227/11
PDDSX 015053       226/24  227/11
PDDTB 014616       214/18  221/48
PDDW1 014601       218/32  218/35  219/47  221/35  224/06
PDDW2 014602       218/36  219/49  221/36
PDDW3 014603       218/39  219/51  221/37
PDDW4 014604       218/40  219/53  221/38
PDECI 000072        32/13   33/24   37/21  211/48  211/51  289/22  289/25
                   289/29  289/32  314/22  314/25  314/29  314/32  362/42
                   362/45  362/50
PDERR 015020       224/13  224/17  224/21  224/25  224/30  226/16  227/07
PDEST 014614       216/46  221/46
PDE?  004721        32/13   33/18   89/06
PDHIK 015004       224/47  225/05
PDHOE 014613       221/45  222/24
PDICQ 014671       222/35  222/41
PDIDX 014611       216/49  219/54  221/43  226/12
PDIIX 015015       224/41  226/12
PDMK1 014243       211/60  214/11
PDNNS 015111       227/18  227/24  227/40  227/44
PDSEC 014600       216/43  218/20  219/43  221/34  222/09
PDSTA 014610       221/42  223/11  227/12
PDSTE 015042       223/18  227/02
PDSTX 015054       226/32  227/03  227/06  227/12
PDSUR 014567       219/19  221/05  221/25
PDSVS 015113       227/20  227/29  227/35  227/46
PDTBL 014573       219/38  221/11  221/29
PDTTL 014622       219/34  221/49
PDTX1 015117       226/20  227/51
PDTX2 015132       226/29  227/53
PDTX3 015145       212/07  227/55
PDTXA 014252       211/35  213/17  214/18
PDX05 014425       218/25  218/26
PDXCA 015055       226/25  227/13
PD.00 014061       211/10  211/16  211/58  213/49  214/04  215/03  222/40
PD.01 014154       211/16  213/05  213/30  213/35
PD.02 014254       211/17  212/06  212/08  215/04  225/03
PD.03 014453       216/32  219/05  219/18
PD.04 014627       219/60  222/06
PD.05 014630       218/26  222/07
```

X17	004264		82/27	83/03					
X7	011134		149/43	151/01					
X7.	016307		241/23	242/02					
XCBLM	002556		57/06	57/27	57/28				
XCKRE	013534		203/04	203/08	203/14	203/16	203/31	204/11	
XCMPB	002557		57/20	57/29					
XDIV	013721		205/31	206/31	208/01				
XDIV1	013726		208/06	208/11					
XHCOM	013764		209/10	209/17					
XHDIV	013757		197/43	204/14	206/07	209/12			
XHDR	014031		210/01	210/06	210/28				
XHMUL	013745		197/40	204/12	206/11	209/01			
XHRET	013770		209/01	209/12	209/20	209/21			
XJ15	005005		90/04	90/09					
XM20	011133		150/18	150/31					
XMS32	001551		44/10	44/41	44/43	45/01	45/08	45/43	
XMUL	013705		205/09	206/47	207/24	207/30			
XMUL1	013715		207/32	208/12					
XMVET	001636		46/03	46/13	46/14				
XM.20	016306		241/42	241/55					
XOAC	013531		204/04	204/08	205/06				
XOMD	013533		204/06	204/10	205/08				
XOMQ	013532		204/05	204/09	205/07				
XOR1	000040	MC	134/11	147/38	148/13	149/10	149/31		
XOR2	000042	MC	134/14	147/35	151/09	151/30	153/09		
XORA	000036	MC	134/08	147/12	147/15	148/10	148/34	152/08	152/27
			153/12	153/15	153/18	153/21	153/24	153/27	153/30
XORTE	011131		150/19	150/26	150/29				
XOR.0	011077		147/13	147/16	148/11	148/35	150/01	152/09	152/28
			153/13	153/16	153/19	153/22	153/25	153/28	153/31
			153/43						
XOR.1	011105		147/39	148/14	149/11	149/32	150/08		
XOR.2	011115		147/36	150/17	150/27	151/10	151/31	153/10	
XOR.4	011132		150/01	150/06	150/08	150/15	150/17	150/28	150/30
XOR.L	011220		153/43	157/13					
XRMSE	002265		52/14	52/40	52/42	52/47			
XSYTB	002062		49/38	49/56					
XTINI	001724		47/06	47/12	47/24	47/34			
XUBL3	001660		46/25	46/38	46/39				
ZB17	022774		305/26	306/15	306/21				
ZB1RL	023407		314/12	314/26	314/33	314/39			
ZB1RR	023427		314/17	314/33					
ZB24	022560		301/11	301/14	301/19	301/27	302/09	302/25	
ZB248	022566		301/05	301/33					
ZB37	023060		307/33	308/39					
ZB8SF	023665		319/09	319/22					
ZBCDS	023451		315/10	315/33					
ZBCOM	023501		310/01	316/06	319/19	319/21			
ZBCST	023035		301/29	304/45	304/46	307/13	309/33	319/38	
ZBCSX	023703		318/24	319/38					
ZBCTL	000010		307/15	307/24	321/04	321/05	321/06	321/07	
ZBCYL	023040		304/53	305/22	307/14	307/16	307/24	308/54	318/40
ZBDBL	000011		307/10	307/11	315/25	321/04	321/05	321/06	321/07
ZBDIB	023675		309/59	319/11	319/18	319/27	319/32		
ZBDNM	023050		304/15	304/21	307/25				
ZBDP0	024072		299/20	299/29	307/11	315/20	321/04		
ZBDP1	024133		315/21	321/05					
ZBDP2	024174		315/22	321/06					

ZBDP3	024235	315/23	321/07					
ZBDPT	023462	299/28	315/07	315/19	315/29			
ZBDRE	024276	299/20	321/08					
ZBDRV	023051	304/20	307/26	308/09	308/50	309/46	318/38	
ZBDRX	023314	299/31	299/32	312/02	312/07	312/10	312/19	312/31
		313/03	313/05	313/10	313/12	313/14	314/40	
ZBDSK	000027	297/06	298/10	298/11	298/21	298/31	298/32	298/33
		298/34	299/21	308/53	308/56	309/04	309/05	309/12
		309/34	309/35	309/36	309/42	309/48	312/13	312/16
		312/18	312/21	312/22	312/25	313/29	313/32	316/46
		319/10	319/26	320/20	320/36			
ZBDST	023034	301/28	303/27	307/12	319/37			
ZBDSX	023702	318/23	319/37					
ZBDW1	023043	303/05	303/28	307/19	316/05	319/20		
ZBDW2	023044	303/06	303/29	307/20				
ZBDW3	023045	303/07	303/32	307/21				
ZBDW4	023046	303/14	303/33	307/22				
ZBERL	023543	316/41	319/31	320/03				
ZBERR	023611	316/11	316/16	316/21	316/26	316/31	318/15	319/17
ZBFOK	023230	309/56						
ZBFSB	023257	299/18	312/02	313/16				
ZBGD3	023461	315/05	315/17	315/18	315/28			
ZBGDS	023443	304/05	312/03	314/13	315/04			
ZBHDP	023023	307/02	308/28					
ZBHDQ	023244	311/09						
ZBHIK	023574	316/52	317/09					
ZBIDX	023057	303/45	304/44	306/19	307/32	318/14	320/13	
ZBKNA	023350	311/05	313/17					
ZBKNB	023351	311/09	313/18					
ZBKNC	023352	312/08	313/19					
ZBKND	023353	312/26	312/28	313/11	313/20			
ZBKNE	023633	316/43	318/38	320/18	320/32			
ZBKNF	023634	316/44	318/39					
ZBKNG	023635	318/22	318/40					
ZBKNH	023636	318/41	319/07					
ZBKNI	023637	318/42	319/12					
ZBKNJ	024035	320/26	320/51					
ZBKNL	024036	320/29	320/52					
ZBKNM	024037	320/30	320/53					
ZBKNO	024040	320/34	320/54					
ZBLHD	023025	307/04						
ZBLSC	023022	304/23	307/01	307/10	315/24			
ZBMK1	022466	298/15	299/07	299/20				
ZBNBS	023651	319/06	319/10					
ZBNCT	022376	298/03	298/13	299/11				
ZBNCY	023032	305/06	307/09					
ZBNHD	023026	307/05						
ZBNSC	023024	307/03	308/18					
ZBRCI	023757	320/04	320/17	320/25	320/42			
ZBRCL	023734	318/37	320/04	320/12				
ZBRDO	023200	309/31						
ZBRSC	023047	303/16	306/08	306/17	307/23			
ZBRSK	023207	309/24	309/28	309/38	309/55	309/57	320/52	
ZBRTX	023437	313/09	314/05	314/08	314/09	314/12	314/20	314/34
		314/35	314/38	314/41				
ZBRWI	023224	309/29	309/40	309/52				
ZBSCQ	023237	311/03	312/30					
ZBSCY	023027	302/03	306/12	307/06				

ZBSDS 023471	313/04	315/28						
ZBSEC 023041	301/31	303/05	303/06	303/07	303/14	303/16	303/27	
	303/28	303/29	303/32	303/33	303/36	306/09	306/18	
	307/17	308/12						
ZBSKB 023030	307/07	308/51	309/44	320/13				
ZBSKI 023163	308/48	309/17	309/18	309/23	311/03			
ZBSKM 023031	307/08							
ZBSTA 023056	307/31	309/06	319/39					
ZBSTD 023332	311/19	312/33	313/03	313/06	314/36			
ZBSTE 023641	309/60	319/02	319/03					
ZBSTX 023704	318/31	319/02	319/15	319/39				
ZBST. 023234	309/38	309/56	309/60					
ZBSV2 023376	313/26	313/27	313/28	313/38	313/39	313/42		
ZBTBA 022475	298/16	299/29						
ZBTBL 023033	304/27	304/42	305/12	307/11				
ZBTX1 023705	318/19	319/40						
ZBTX2 023720	318/28	319/42						
ZBTX3 024005	319/22	320/48						
ZBTXA 022474	298/26	299/28						
ZBTXB 022476	299/05	299/06	299/30					
ZBTXC 022501	299/33							
ZBTXR 024041	314/41	321/01						
ZBWDN 023356	312/14	312/23	313/26					
ZBWEL 023470	315/10	315/25						
ZBWEP 023467	315/24							
ZB.00 022355	297/11	297/17	298/04	299/08	300/05	308/59	314/04	
	318/13							
ZB.01 022402	297/17	298/09						
ZB.02 022503	297/18	300/05	300/21	300/32	314/43	317/08		
ZB.03 022671	301/20	304/07	304/11	304/22	304/26	304/39		
ZB.04 023065	306/07	306/11	308/08					
ZB.05 023063	303/43	308/06	308/13					
ZB.0X 023146	308/49	308/59	309/07	309/30	309/39			
ZB.1L 022424	298/28	298/45						
ZB.1R 023401	299/19	314/05	314/06					
ZB.2C 022533	300/25	300/28	301/03	301/24	318/12			
ZB.2J 022510	300/11							
ZB.2L 022524	300/26	300/34						
ZB.2R 022553	300/31	301/21						
ZB.37 023575	316/56	317/10						
ZB.3G 022643	303/26	306/23	320/46					
ZB.3L 022747	305/03	305/10	305/15	305/18	305/21			
ZB.3S 022571	302/05	302/10	302/18	302/29	305/24			
ZB.4A 022777	304/55	306/03						
ZB.5G 023134	308/48	320/47						
ZB.5S 023105	308/25							
ZB.BK 022773	301/30	305/25	308/34					
ZB.C0 023232	309/43	309/58						
ZB.CO 023054	307/29	308/11	309/31	319/35				
ZB.CX 023700	318/32	319/04	319/25	319/35	320/15			
ZB.IS 023151	299/23	309/05	309/10	309/25				
ZB.K1 022467	299/12	299/21						
ZB.K2 022470	299/13	299/22						
ZB.K3 022471	299/14	299/23						
ZB.K4 022473	298/24	299/25						
ZB.KK 022557	300/04	301/15	301/26					
ZB.ND 022565	298/25	298/38	299/09	301/32	302/17	303/04	303/13	
	303/26	314/42						

ZB.OC 023055	307/30	308/19	308/24	308/42	309/32	319/36		
ZB.OX 023701	318/33	319/36						
ZB.S3 022472	299/15	299/24	314/44					
ZB.SA 023677	318/15	319/34						
ZB.SC 023053	307/28							
ZB.SX 023606	316/42	316/51	318/09	318/12	320/05	320/06	320/38	
ZB.WC 022665	303/11	303/44						
ZB.WK 023052	302/04	303/36	306/22	307/27	308/47	319/20		
ZB.X2 023601	318/05	318/08						
ZB.X3 023555	316/52							
ZB.X4 023564	316/59	317/02						
ZB.X5 023567	317/03	317/06						
ZB.XI 023576	297/23	297/24	301/25	316/55	316/58	318/01	318/04	
ZB.XX 022556	300/10	300/13	301/25					
ZOC? 004714	32/12	33/19	89/01					
ZSU?P 004753	89/11	89/18	89/28	89/33	90/19	90/31	90/33	
	90/50	90/52	91/05					
.BDEG 000223	31/31	31/49	38/23					
.CB03 011461	162/02	162/10	162/11	162/15	162/24			
.CB20 011462	162/03	162/12	162/16					
.CB99 011452	162/04	162/08						
.CC98 011467	162/18	162/23						
.DIVU 000044 MC	134/17	164/13						
.ERSV 003463	72/14	72/19	72/32	72/43	73/16			
.MH8F 021305	284/24	285/03						
.MHHD 021307	282/33	283/19	285/05					
.MHLH 021311	283/25	285/07						
.MHLS 021306	282/10	283/09	284/06	284/07	284/41	285/04	286/31	
	287/03	287/19	289/14	289/27	289/30	289/34	289/35	
	290/33							
.MHNC 021316	285/12	286/08	287/13	287/23	287/27	288/01	289/23	
.MHNH 021312	283/31	285/08	289/30					
.MHNS 021310	282/38	283/33	285/06	289/27				
.MHSC 021313	283/38	285/09						
.MHSK 021314	284/26	285/10	286/20	287/20	288/11			
.MHSM 021315	284/40	285/11	286/10	287/11				
.MPYA 000046 MC	134/20	164/16						
.MPYU 000056 MC	134/32	159/14	160/14					
.ZBHD 023321	311/04	312/36						
.ZBLH 023323	312/38							
.ZBLS 023320	312/04	312/05	312/35	314/14	314/27	314/30	315/34	
.ZBNC 023330	312/29	312/43	314/23					
.ZBNH 023324	311/10	312/39	314/30					
.ZBNS 023322	311/06	311/12	312/37	314/27				
.ZBSC 023325	311/17	312/40						
.ZBSK 023326	312/09	312/41						
.ZBSM 023327	312/42							