DataGeneral

DIAGNOSTIC LISTING

LISTING

096-000424-00

PROGRAM

MICRO NOVA MULTIPROGRAMMING
RELIABILITY TEST

MNMRT

TAPE

095-000424-00

ABSTRACT

THE MICRO NOVA MULTIPROGRAMMING RELIABILITY TEST CONSISTS OF A SERIES OF INDIVIDUAL PROCESSOR AND PERIPHERAL TESTS ANS A SUPERVISOR PROGRAM, THE DIAGNOSTIC LINKER. THE DIAGNOSTIC LINKER IS A PROGRAM DESIGNED TO "LINK" THE VARIETY OF SC MEMORY TESTS.

COPYRIGHT © DATA GENERAL CORPORATION, 1976
ALL RIGHTS RESERVED. PRINTED IN U.S.A.

```
 0001 MNMRT MACRO REV 03.00              13:59:36 12/03/76
01
02
03
04
05
06
07              ;*****************************************************************
08              ;
09              ; NAME: MNMORT,SR                          PART NUMBER: 094-000835
10              ;
11              ;
12              ; DESCRIPTION: MICRO NOVA MULTIPROGRAMMING RELIABILITY TEST
13              ;
14              ;
15              ; REVISION HISTORY:
16              ;
17              ;       REV.            DATE
18              ;
19              ;       00              12/03/76
20              ;
21              ;
22              ; COPYRIGHT (C) DATA GENERAL CORPORATION, 1976
23              ; ALL RIGHTS RESERVED.
24              ;*****************************************************************
25
```

```
 0002 MNMRT
01
02              ;CONDITIONAL ASSEMBLY FLAGS
03
                        .TITL  MNMRT
```

```
↓0003 MNMRT
01                ;FILE FOR MICRO-NOVA VERSION
02        000000 .DUSR MNOVA=0
03        000000 CBRUS=0
04        000001 DCUTS=1
05        000000 SCMTS=0
06        000000 ARITH=0
07        000001 FPTST=1
08        000000 MUDVT=0
09        000001 NVDSK=1
10        000001 MVDSK=1
11        000001 MTTES=1
12        000001 CATES=1
13        000001 LPTTS=1
14        000001 NXDSK=1
15        000001 MXDSK=1
16        000001 CXTES=1
17        000001 MXTES=1
18        000001 IOTST=1
19        000001 .DUSR .MAPD=1
```

```
↓0004 MNMRT
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                ;2.     MACHINE REQUIREMENTS
18                ;2.1    MICRO-NOVA PROCESSOR
19                ;2.2    8K TO 32K OF READ/WRITE MEMORY
20                ;       (MEMORY MUST BE CONTIGUOUS)
21                ;2.3    TTY INPUT/OUTPUT OR TERMINAL
22                ;2.4    PREREQUISITES
23                ;2.4.1  SOFTWARE PREREQUISITES
24                ;       THE SYSTEM SHOULD BE CAPABLE
25                ;       OF RUNNING ALL INDIVIDUAL LOGIC AND
26                ;       RELIABILITY TESTS PERTAINING TO THE
27                ;       PROCESSOR AND ITS PERIPHERAL EQUIPMENT
28                ;       BEFORE ATTEMPTING TO RUN THIS TEST
29                ;NOTE:  ALTHOUGH THIS TEST MAY AT TIMES BE USEFUL
30                ;IN DETERMING THE GO/NO GO STATUS OF AN
31                ;UNKNOWN SYSTEM, IT IS RECOMMENDED THAT:
32                ;A. ALL OTHER DIAGNOSTICS BE RUN EVEN IN THE
33                ;   EVENT THAT THIS TEST FINDS NO PROBLEMS,
34                ;B. AN ATTEMPT BE MADE TO ISOLATE ANY PROBLEMS
35                ;   FOUND BY FIRST UTILIZING THE LOWER
36                ;   LEVEL TESTS FOR MORE CONSISE ERROR REPORTS.
```

```
                ;3.       SWITCH SETTINGS
                ;3.1      AUTO-SIZE AND GO START AT 200
                ;3.2      MANUAL SELECT/DELETE TESTS START AT 202
                ;3.3      RESTART LAST TEST SELECTIONS 204
                ;3.5    IMMEDIATELY ENTER ODT START AT 210
                ;
                ;3.6    KEY ENTERED OPTIONS
                ;               ENTRIES TYPED IN SET BITS IN SWREG
                ;               FOR USE BY THE PROGRAM.
                ;
                ;       KEY 1 =1 DON'T RELEASE AND ALLOW REASSIGNMENT
                ;                OF MEMORY AFTER ERROR
                ;       KEY 2 =1 DELETE TYPEOUTS
                ;       KEY " =1 DELETE MEM ALLOCATION TABLE
                ;                FROM TYPEOUTS
                ;       KEY # =1 CAUSES THE DELETION OF THE RANDOM
                ;                WAIT STATES IN THE TTY AND LPT
                ;                TESTS.
                ;       KEY 4 =1 WILL CAUSE THE ELAPSED RUN
                ;                TIME AND ACCUMULATED ERRORS
                ;                TO BE TYPED ON THE TTY.
                ;                (NOTE: A RTC MUST EXIST)
                ;       KEY 6 =1 THE ERROR ROUTINE WILL PAUSE AFTER
                ;                EACH PHASE OF AN ERROR TYPEOUT.
                ;                TYPE ANY KEY ON DEVICE TTI TO PROCEED.
                ;
                ;       EACH KEY ENTRY COMPLEMENTS THE PREVIOUS STATE OF
                ;       SWREG BIT EXCEPT CONTROL CHARACTERS
                ;       FOLLOWING:
                ;
                ;       KEY (C)U        ENTER THE ODT EDITOR
                ;                       (SEE DESCRIPTION AT PARAGRAPH 7.0)
                ;       KEY (C)S        PRINT THE RUN STATISTICS OF EACH
                ;                       TEST.
                ;       KEY (C)D        DEFAULT MODE RESTART. SWREG
                ;                       SET TO 0.
                ;       KEY (C)R        RESTART WITHOUT RESETTING SWREG BITS.
                ;       KEY M           TYPE THE CURRENT CONTENTS OF SWREG.
                ;
                ;       WHERE (C) SIGNIFIES A CONTROL KEY.
```



```
                ;4.       OPERATING PROCEDURES
                ;4.1    LOAD THE PROGRAM VIA WHICHEVER LOAD DEVICE
                ;       AVAILABLE.
                ;       AFTER AUTOSTARTING IF TEST SELECTION
                ;       OR OPTION SELECTION IS DESIRED USE ODT(PARAGRAPH 7.)
                ;       TO ENTER THESE SELECTIONS.
                ;4.4      PROCESSOR WILL TYPE:
                ;       NAME/REV/DATE OF KEV
                ;       TOTAL #1K'S=XXX(DECIMAL) MAP OR NO MAP
                ;       PROGRAM RUN LIST
                ;       PROG# DESCRIPTION
                ;4.5      IF START WAS 200 OR 204   THE LIST OF
                ;       PROGRAMS TO BE RUN CONCURRENTLY WILL
                ;       THEN BE LISTED AND THE TEST SYSTEM
                ;       WILL AUTO START
                ;4.6      IF START WAS 202 LINKER WILL
                ;       PAUSE AT THE END OF EACH TEST
                ;       DESCRIPTION AND WAIT FOR KEYBOARD
                ;       INPUT.  TYPING IN A SPACE WILL
                ;       ENABLE THAT TEST TO BE RUN.
                ;       TYPING IN ANY OTER CHARACTER WILL
                ;       DELETE THAT TEST FROM BEING RUN
                ;
```

```
10007 MNMRT
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# +MODULO 1K + LOGICAL ADDRESS
22              ;5.1.5          CONTINUATION INFORMATION IN GROUPS
23              ;       OF 3 MEMORY LOCATIONS PERTINENT TO
24              ;       THE INDIVIDUAL TEST THAT FAILED
25              ;
26              ;5.1.6  THE CPU TESTS THAT RELOCATE/REMAP WILL
27              ;IN THEIR ERROR TYPEOUTS:
28              ;ST,LA  START/ERROR (RES.)
29              ;XXXXXX YYYYYY  ZZZZZZ
30              ;
31              ;ST,LA  THE LOGICAL START OF THE RELOCATED TEST LOOP
32              ;XXXXXX (I.E. THE LAST LCALL SETUL)
33              ;
34              ;START  THIS NUMBER INDICATES WHERE THE RESIDENT COPY
35              ;YYYYYY OF THE TEST LOOP MAY BE FOUND IN THE LISTING
36              ;
37              ;ERROR  THIS NUMBER INDICATES WHERE IN THE RESIDENT
38              ;ZZZZZZ COPY OF THE LISTING THE ERROR CALL MAY BE FOUND
39              ;       (FOR SOME VALIDITY TRAP ERRORS THIS NUMBER
40              ;       MAY NOT APPEAR TO BE VALID.)
```

```
10008 MNMRT
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 DIRECTI
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 MNMRT
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              ;5.3.5  MUL/DIV TEST
16              ;
17              ;
18              ;               MULTIPLY DIVIDE FAILURES WILL INDICATE
19              ;               EITHER MUL FOR MULTIPLY OR DIV FOR DIVIDE
20              ;                IN ADDITION, THREE SETS OF AC'S ARE TYPED
21              ;                   ORIGINAL OPERANDS
22              ;                   HARDWARE RESULT (ASSUMED TO BE INCORRECT )
23              ;                   SOFTWARE RESULT (ASSUMED TO BE CORRECT )
24              ;
25              ;
```

```
10010 MNMRT
01              ;5.4    SPECIAL CASE ERROR TYPEOUTS
02              ;
03              ;5.4.1  POWER FAIL INTERRUPT
04              ;       UPON DETECTION OF A POWER FAIL INTERRUPT
05              ;       THE LOGICAL ADRS. OF THE P.C. AT INTERRUPT
06              ;       WILL BE SAVED.
07              ;       IF AUTO-RESTART  IS ENABLED OR THE POWER
08              ;       FAIL WAS ONLY MOMENTARY, THE TEST WILL RE-
09              ;       START AS IN A START AT 206 AFTER TYPING
10              ;POWER FAIL @XXXXXX (WHERE XXXXXX IS THE PC AT INTR.)
11              ;
12              ;5.4.2  ILLEGAL SUPERVISOR CALL
13              ;
14              ;       UPON DETECTION OF A SUPERVISOR CALL
15              ;       WHICH DIDN'T MATCH THE LIST OF SUBROUTINES
16              ;       CALLS THE FOLLOWING MESSAGE WIIL BE TYPED:
17              ;
18              ;       ILLEGAL SUPER CALL AT XXXXXX
19              ;
20              ;       PROG#   NNN
21              ;
22              ;       AC'S    QQQQQQ  YYYYYY  ZZZZZZ
23              ;
24              ;       TTTTTT  WWWWWW  SSSSSS
25              ;
26              ;               WHERE XXXXXX IS THE LOGICAL ADDRESS OF THE
27              ;               SUPER CALL, TTTTTT IS AC3 CONTENTS
28              ;               AND WWWWWW IS THE PHYSICAL PAGE #,SSSSSS
29              ;               IS THE INSTRUCTION CAUSING THE SUPER-
30              ;               CALL.
```

```
10011 MNMRT
01              ;5.4.4  INTERRUPT WAIT ELAPSED
02              ;       THE PERIPHERAL DEVICE ASSOCIATED WITH THE
03              ;       PROG. NUMBER TYPED HAS NOT RESPONDED WITH
04              ;       A PROGRAM INTERRUPT FOR AN EXTENDED
05              ;       PERIOD OF TIME. THE 2ND NUMBER TYPED
06              ;       SHOULD POINT AT THE INTERRUPT HANDLER
07              ;       FOR THE DEVICE THAT FAILED
```

```
10012 MNMRT
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 IN 1K INCREMENTS
17              ;       FROM 0 TO 32K AND BUILD A 2 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              ;
```

```
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. MEMORY LOCATIONS SCRLO
16              /             AND SCRHI (SCRATCH LOW AND HIGH) ARE
17              /             SET TO INDICATE THE LIMITS OF
18              /             THE SCRATCH AREA AVAILABLE TO THE TEST.
19              /
20              /          3. DATA CHANNEL LIMITS (DCHLO AND DCHHI)
21              /             ARE CALCULATED AND ENTERED
22              /
23              /          4. THE SELECTED TEST IS ENTERED AT
24              /             ITS SPECIFIED EXECUTE ENTRY POINT
25              /
```



```
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
```

```
10015 MNMRT
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 SCRH
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.
```

```
10016 MNMRT
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. AT THE END OF EACH EXECUTION PASS SCRATCH
08              ;AREA IS RANDOMLY RELEASED OR HELD.  IF HELD, THE NEXT TIME
09              ;THE TEST IS ENTERED, THE EXECUTABLE PORTION OF THE TEST WILL
10              ;AGAIN BE RANDOMLY RELOCATED WITHIN SCRATCH FOR EXECUTION.
11              ;
12              ;6.4.6  MUL/DIV TEST
13              ;  THIS TEST WAS DERIVED FROM THE STAND ALONE
14              ;  MUL/DIV TEST.
15              ;  NO MEMORY REALLOCATING IS DONE IN THIS TEST.
```

```
10017 MNMRT
01              ;6.4.11 REAL TIME CLOCK
02              ;
03              ; THE REAL TIME CLOCK RUNS AT 416.66 HERTZ. RUNTIME ALONG
04              ;WITH ACCUMULATED ERROR COUNT ARE PRINTED AT 5 MINUTES
05              ;15 MINUTES, 30 MINUTES AND EVERY 30  MINUTES OF RUNTIME
06              ;THEREAFTER.  THIS TYPEOUT ALSO OCCURS AFTER EVERY ERROR
07              ;TYPEOUT OR IF A TTY KEY WITH SW 4=1 IS TYPED.
08              ;
09              ;6.4.12 TELETYPE TEST
10              ;
11              ;THE TELETYPE TEST PRINTS A SINGLE LINE CONSISTING OF THE
12              ;CHARACTERS SPACE TO Z.  THE TEST WILL ALSO ECHO CHARACTERS
13              ;AS TYPED.
14              ;
```

```
10018 MNMRT
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              ;
```

```
10019 MNMRT
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              ;
```

```
10020 MNMRT
01              ;
02              ;
03              ;7.6 OTHER COMMANDS
04              ;  --------------
05              ;
06              ;RUBOUT THE RUBOUT KEY IS USED TO DELETE ERRONEOUSLY TYPED
07              ;       DIGITS EACH TIME THE RUBOUT KEY IS PRESSED, THE RIGHT
08              ;       MOST DIGIT IS DELETED AND ECHOED ON THE TERMINAL.
09              ;       IF THE RUBOUT KEY IS PRESSED RIGHT AFTER OPENING A CELL
10              ;       THEN IT ALLOWS THE MODIFICATION OF THE CONTENTS AS IF
11              ;       THEY WERE TYPED JUST BEFORE THE KEY WAS PRESSED.
12              ;
13              ;P      RESTART THE EXECUTION OF THE PROGRAM AT THE LOCATION
14              ;       POINTED BY SAVED AC3 .
15              ;
16              ;ADRR   START EXECUTING THE USERS PROGRAM AT LOCATION ADR AFTER
17              ;       AN IO RESET.
18              ;
19              ;K      KILL THE STRING TYPED SO FAR. OUT RESPONDS WITH A ? AND
20              ;       THE OPEN LOCATION IS CLOSED WITHOUT MODIFICATION.
21              ; =     PRINT THE CURRENT ARGUMENT  (I.E.  TYPING ".=" WILL
22              ;       PRINT THE ADRS OF THE LAST OPENED LOCATION)
```

```
10021 MNMRT
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
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 TH
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                  /
27
28
29
```

```
10022 MNMRT
01
02                  /       .TITL   LINKR
03                  /DIAGNOSTIC PROGRAM LINKER
04          000000          .LOC 0
05 00000 011375             DIRT
06                  /CONCURRENTLY _INK 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 000202             JMP STRT2
17 00207 000206             JMP .-1
18 00210 000212 STRUDT: JSR .+2         /START ODT DIRECTLY
19 00211 000202             JMP STRT2
20 00212 003314             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 LUPNL:  0
29 00222 000000 STFLG:  0           /=1 IF PRINT STATISTICS
30
31
32          000045          .LOC 45
33 00045 000122             .BDEGG
34          000050          .LOC 50
35 00050 000000 RNSEL:  0           /=1 IF DELETE RANDOM WAIT FOR
36                                          /FOR TTY/LPT TESTS
37 00051 000000 DLTBL:  0           /=1 IF DON'T PRINT ALLOC. TBL
38 00052 000000 USESW:  0
```

```
↓0023 MNMRT
01                      ;CALL HANDLER SUBROUTINE LINK
02 000054 002514 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 000054 002647 ASCRA: ASSCR     ;ASSIGN SCRATCH
08 000055 002666 ESCRA: EXSCR     ;EXPAND SCRATCH
09 000056 002710 RSCRA: RLSCR     ;RELEASE SCRATCH
10 000057 002736 GSCRA: GOSCR
11 000060 003722 PCRLF:  CLF?     ;CAR. RETURN LINE FEED
12 000061 003640 PZOCT:  ZOC?     ;ZERO SUPRESS OCTAL
13 000062 003645 PDECI: PDE?      ;DECIMAL PRINT
14 000063 003156 ERRTX: ERTXT     ;TEXT TYPEOUT CALL
15 000064 001713 RETRN: LRETP     ;RETURN FROM TEST
16 000065 003230 ERPAD: EPADR
17 000066 003255 ERPAC: EPACS
18 000067 001742 RANG:  RANGN     ;RANDOM # GENERATOR
19 000070 002065 DIVI:  DIVID     ;INTEGR UNSIGNED DIV
20 000071 004404 EINTS:  EINTP    ;ENTER INTR SERVICE
21 000072 003036 ERROI:  ERROM    ;INIT ERROR TYPEOUTS
22 000073 003132 ERROC:  ERROE    ;APPEND TO ERR TYPEOUT
23 000074 002770 SETUL:  SETLP    ;SET UP STRT OF LOOP
24 000075 003010 LLOOP:  LOOPL    ;LOUP BACK TO SETUP
25 000076 002011 FRANG:  RANG3    ;RAN #'S TO AC'S 0,1,AND 2
26 000077 002747 ERRET:  ERRRT    ;2ND LEVEL ERR RETURN
27 000100 002751 RETU2:  RETN2    ;2ND LEVEL NO ERR RETURN
28 000101 004600 ADMAP: AMSCR     ;ASSIGN SCR TO A DCH
29 000102 004634 EDMAP: EMSCR     ;EXPAND DATA CHANNEL ASSIGN
30 000103 004652 RDMAP: RDSCR     ;RELEASE DCH MAP
```

```
↓0024 MNMRT
01                      ;TEST PARAMETER LOCATIONS
02 000104 000000 CURPR: 0
03 000105 000000 ST.LC: 0
04 000106 000000 ST.LP: 0
05 000107 000000 ST.LA: 0
06                      ;CONTENTS OF LAST 3 RANDOM AC'S FROM RANG3
07 000110 000000 RNAC0: 0
08 000111 000000 RNAC1: 0
09 000112 000000 RNAC2: 0
10                      ;LINKS TO TYPEOUT ROUTINES
11 000113 003722 ICLF?:  CLF?
12 000114 003645 IPDE?:  PDE?
13 000115 003640 IZOC?:  ZOC?
14 000116 003612 LMESS:  MES?
15        000060 LCRLF= PCRLF
16 000117 003641 LPOCT:  POC?
17        000061 LZOCT= PZOCT
18        000062 LPDEC= PDECI
19                      ; POWER FAIL AUTO RESTART LINKS
20 000120 001251 PFAIL:  PWRUP
21 000121 000000 PFAIS:  0
22 000122 000000 .BDEGG: 0          ;DTOS AUTO SWITCH
23 000123 000000          0         ;
24 000124 000000          0         ;CAT SWITCH FOR DTOS
25 000125 000000          0         ;PASS CNT FOR DTOS
26 000126 000000          0         ;DTOS STARTING ADDRESS
27 000127 000000 SWREG:  0          ;SWITCH REGISTER DEFAULTS TO 0
28 000130 000000 ALTBL: 0           ;ADRS MEM ALLOCATION TABLES
29 000131 000000 PSTRT: 0           ;FIRST LOC OF TST
30 000132 000000 PENDA: 0           ;LAST LOC
31 000133 000000 RTTIM: 0           ;ELAPSED RUNTIME IN MINUTES
32 000134 000000 STATS:  0          ;MAP STATUS REG.
33 000135 177777 UDEVI: -1
34 000136 000000 TOTPK: 0
35 000137 177777 TIMSW: -1          ;=0 IT'S TIME TO PRINT
36 000140 000000 ERTOT: 0           ;ERROR ACCUMULATOR
37 000141 000000 EACIV:  0          ;180=TTO  181=LPT
38 000142 000000 LASTI: 0           ;CHAR FROM LAST INPUT FROM TTI
39                      ;SCRATCH AREA SIZE PARAMETER LOCS FOR TEST USAGE
40 000143 000000 SCRLO: 0           ;LOWEST LOGICAL SCRATCH ADRS
41 000144 000000 SCRHI: 0           ;HIGHEST LOGICAL SCRATCH ADRS
42 000145 000000 DCHLO: 0           ;LOW LOG DCH ADRS
43 000146 000000 DCHHI: 0           ;HIGH LOG DCH ADRS
44        000147 LZMAX=DCHHI+1
45        000147 LPG0=.
```

```
10025 MNMRT
01        061401 .DIAC PSH=061401
02        061601 .DIAC PUP=061601
03        062401 .DUSR SAVE=062401
04        062601 .DUSR RTRN=062601
05        060001 .DIAC MTFP=60001
06        061001 .DIAC MTSP=61001
07        062001 .DIAC MFFP=60201
08        061201 .DIAC MFSP=61201
09        000004 .DUSR PRTY = 4
10        000047 TPAUR=47
11        000046 TPLOC=46
12        061077 .DUSR I,RST=DUA 0,CPU
```

```
10026 MNMRT
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 040420           STA 0,LAUTO       ;SET ENTRY TYPE SW
13 01013 102000           ADC 0,0
14 01014 040121           STA 0,PFAIS       ;SET NOT PWR/FA RESTART
15 01015 004405           JSR LDSPR         ;START DISPATCH
16 01016 001033           LTBL1    ;THROUGH INIT SEQ
17 01017 060177           INTEN
18 01020 004402 LRUNS:    JSR LDSPR
19 01021 001047           LTBL2    ;THROUGH RUN TABLE
20                ;DISPATCH ROUTINE
21                ;ENTER SUBROUTINES IN SEQ VIA TABLE SPEC BY (R3)
22                ;END OF EACH TABLE IS LRUNS WHICH WILL START US
23                ;BACK AT THE BEGINNING OF THE RUN TABLE
24 01022 021400 LDSPR:    LDA 0,0,3         ;ADRS OF DISPATCH TABLE
25 01023 040406           STA 0,LIDIS
26 01024 036405 LDS.1:    LDA 3,@LIDIS
27 01025 005400           JSR 0,3
28 01026 010403           ISZ LIDIS
29 01027 004431           JSR CKOUT
30 01030 000774           JMP LDS.1
31 01031 000000 LIDIS: 0
32 01032 000000 LAUTO: 0
33                ;DEFINE SYSTEM MACROS FOR INDIVIDUAL TESTS
34                        .MACRO LCALL
35                        A1=ASCRA*1B11+100010
36                %
37                        .MACRO NEXTT
38                LMEML=.
39                        .LOC LPG0
40                        A1
41                LPG0=.
42                        .LOC LMEML
43                        0       ;TEST PASS CTR
44                        0       ;TEST ERROR CTR
45                        0       ;INTERRUPT TIMEOUT SWITCH
46                %
```

```
10027 MNMRT
01
02                ;LTBL1=INIT SYSTEM DISPATCH TABLE
03                ;END OF TABLE IS LRUNS
04 01033 001306 LTBL1:  LSYSR   ;RESET SYS
05 01034 001373         GPRGK   ;GET # TESTS LOADED
06 01035 001320         LMSET   ;SET UP RUNNING CONS.
07 01036 004415         LCINT   ;INIT INTR VECTORS
08 01037 001406         LSIZE   ;SIZE MEMORY
09 01040 001465         MVETA   ;MOVE EXISM TO AVALM
10 01041 001504         UBL32   ;REMOVE USED CORE FROM AVALM
11 01042 001532         TINIT   ;INIT EACH TEST LOADED
12 01043 004713         LPRSL   ;LISTS TESTS TO BE RUN
13 01044 001017         LRUNS-1
14 01045 000000         0
15 01046 000000         0
16                ;
17                ;LTBL2=RUN SYS DISPATCH TABLE
18 01047 001570 LTBL2:  LRANP   ;RANDOM PROG SELECT
19 01050 002621         LDMAP   ;LOAD MAP OPTION
20 01051 004672         LDCHL   ;CALC DCHLO/HI
21 01052 001706         LSTRP   ;START TEST RUNNING
22 01053 001140         LSTAT   ;ADJUST RUN STATISTICA
23 01054 002103         CHSTK
24 01055 001020         LRUNS
25 01056 000000         0
26 01057 000000         0
```

```
10028 MNMRT
01                ;CKODT - CHECK IF TTI REQUESTS HAVE BEEN INPUTTED
02                ;       BY THE OPERATOR.
03 01060 020142 CKODT:  LDA 0,LASTI     ;GET LAST INPUT FROM TTI
04 01061 101103         MOVL 0,0,SNC    ;SKP IS NEW INPUT
05 01062 001400         JMP 0,3         ;NON ENTERED,EXIT
06 01063 101220         MOVZR 0,0
07 01064 030451         LDA 2,J177
08 01065 143400         AND 2,0         ;MASK FOR LOWER BITS ONLY
09 01066 040142         STA 0,LASTI
10 01067 054443         STA 3,CK,S3     ;SAVE AC3
11 01070 024444         LDA 1,N17
12 01071 106414         SUB# 0,1,SZR    ;SKP IS REQUESTED OUT
13 01072 000403         JMP CKKEY
14 01073 006436         JSR @GOUT       ;GO TO ODT EDITOR
15 01074 002436         JMP @CK,S3      ;RETURN
16 01075 006441 CKKEY:  JSR @IINP?      ;CHECK IF KEY ENTRY
17 01076 024441         LDA 1,J23       ;CK FOR CONTROL S
18 01077 136414         SUB# 1,3,SZR
19 01100 000404         JMP .+4
20 01101 126000         ADC 1,1
21 01102 044222         STA 1,STFLG     ;SET FLAG TO PRINT STATS,
22 01103 000416         JMP CK,EX       ;EXIT
23 01104 024427         LDA 1,J42
24 01105 136414         SUB# 1,3,SZR    ;CHECK FOR " KEY ENTRY
25 01106 000405         JMP .+5
26 01107 034051         LDA 3,DLTBL     ;COMPLEMENT
27 01110 174000         COM 3,3         ;STATE OF
28 01111 054051         STA 3,DLTBL     ;ALLOCATION TABLE FLAG
29 01112 000407         JMP CK,EX       ;EXIT
30 01113 125400         INC 1,1
31 01114 136414         SUB# 1,3,SZR    ;CHECK FOR # KEY ENTRY
32 01115 000404         JMP CK,EX
33 01116 034050         LDA 3,RNSEL
34 01117 174000         COM 3,3
35 01120 054050         STA 3,RNSEL     ;COM TTY/LPT RANDOM SEL FLAG
36 01121 102400 CK,EX:  SUB 0,0
37 01122 024127         LDA 1,SWREG
38 01123 127120         ADDZL 1,1
39 01124 127120         ADDZL 1,1
40 01125 125103         MOVL 1,1,SNC
41 01126 002404         JMP @CK,S3
42 01127 040137         STA 0,TIMSW     ;SET SW SO TIME WILL FOLLOW
43 01130 002402         JMP @CK,S3
44 01131 003314 GODT:   OUT
45 01132 000000 CK,S3:  0
46 01133 000042 J42:    42
47 01134 000017 N17:    17
48 01135 000177 J177:   177
49 01136 004146 IINP?:  INP?
50 01137 000023 J23:    23
```

```
↓0029 MNMRT
01                ;END OF TEST PASS, SEE IF ANY EXTRANEOUS ERRORS
02                ;NEED TO BE REPORTED
03 01140 054506 LSTAT:  STA 3,LST,3
04 01141 024135         LDA 1,UDEVI
05 01142 124015         COM# 1,1,SNR     ;SKP =UNEXP, INTA
06 01143 000406         JMP .+6          ;NONE REC'D
07 01144 006116         JSR @LMESS
08 01145 005176         UDEVT
09 01146 006061         JSR @LZUCT
10 01147 102000         ADC 0,0
11 01150 040135         STA 0,UDEVI
12 01151 034131 PRSTAT: LDA 2,PSTRT      ;PTR. TO XX,00
13 01152 011375         ISZ -3,2         ;+1 PASS CTR THIS TEST
14 01153 000406         JMP .+6          ;NOT OFLOW'D
15 01154 006116         JSR @LMESS       ;TYPE 65K HEADER
16 01155 005116         TX65K
17 01156 024104         LDA 1,CURPR
18 01157 006061         JSR @LZUCT       ;AND PROG#
19 01160 006060         JSR @LCRLF
20 01161 020222         LDA 0,STFLG      ;CHECK IF WANT PRINTOUT
21 01162 101005         MOV 0,0,SNR      ;OF STATISTICS
22 01163 000430         JMP LSXIT        ;NOBODY'S INTERESTED
23 01164 102400         SUB 0,0
24 01165 040222         STA 0,STFLG      ;CLR FLAG
25 01166 040104         STA 0,CURPR      ;START WITH 0
```

```
↓0030 MNMRT
01                ;PRINT PASSES AND ERROR COUNTS BY INDIVIDUAL TEST
02 01167 006116         JSR @LMESS
03 01170 005131         STMDR
04 01171 024104 PRSTL:  LDA 1,CURPR
05 01172 010104         ISZ CURPR        ;STEP IT TO NEXT
06 01173 034452         LDA 3,ALZMAX     ;POINTS TO TEST 0
07 01174 137000         ADD 1,3          ;NOW TO  CURRENT TEST
08 01175 031400         LDA 2,0,3        ;GET "XX,00" ADRS
09 01176 151005         MOV 2,2,SNR      ;=0 IS TYPED ALL
10 01177 000414         JMP LSXIT                ;RETURN FROM WHENCE
11 01200 021375         LDA 0,-3,2       ;GET PASS CTR
12 01201 101005         MOV 0,0,SNR      ;SKP IF TEST EXECUTED
13 01202 000767         JMP PRSTL        ;TEST NEVER ENTERED
14 01203 006061         JSR @LZUCT       ;TYPE TEST#
15 01204 025375         LDA 1,-3,2       ;GET # PASSES
16 01205 006062         JSR @LPDEC       ;PRINT IT
17 01206 025376         LDA 1,-2,2       ;GET ACCUMULATED ERRS
18 01207 125004         MOV 1,1,SZR      ;=0 DON'T PRINT
19 01210 006062         JSR @LPDEC       ;PRINT # ERRS THIS TEST
20 01211 006060         JSR @LCRLF
21 01212 000757         JMP PRSTL
22 01213 020122 LSXIT:  LDA 0,.BDEGG     ;GET DTOS SW
23 01214 101004         MOV 0,0,SZR      ;SKP=NOT DTOS RUN
24 01215 000403         JMP .+3
25 01216 034430         LDA 3,LST,3
26 01217 001400         JMP 0,3
27 01220 010136         ISZ TOTPK        ;+1 TOTAL LSTAT CALLS
28 01221 034126         LDA 3,.BDEGG+4   ;GET RETURN ADRS
29 01222 020140         LDA 0,ERTOT      ;GET # ERRS
30 01223 101004         MOV 0,0,SZR      ;SKP IF NO ERRS
31 01224 000417         JMP DTRET        ;RETURN TO DTOS
32 01225 024136         LDA 1,TOTPK      ;GET # PASSES
33 01226 030421         LDA 2,LS,NN      ;IF TOTAL = 4000
34 01227 146405         SUB 2,1,SNR      ;COUNT RUN TIME
35 01230 000403         JMP .+3
36 01231 034415         LDA 3,LST,3
37 01232 001400         JMP 0,3           ;NOT 1 PSUEDO PASS YET
38 01233 044136         STA 1,TOTPK      ;SET BACK TO 0
39 01234 014125         DSZ .BDEGG+3
40 01235 000402         JMP .+2
41 01236 000403         JMP .+3
42 01237 034407         LDA 3,LST,3
43 01240 001400         JMP 0,3          ;KEEP RUNNING
44 01241 061077         I,RST
45 01242 045776         STA 1,-2,3       ;CLEAR ERR SW,
46 01243 061077 DTRET:  I,RST
47 01244 001400         JMP 0,3          ;GO BACK TO DTOS
48 01245 000147 ALZMAX: LZMAX
49 01246 000000 LST,3:  0                ;SAVE AC3
50 01247 004000 LS,NN:  4000
51 01250 012000 LS,MM:  12000
```

```
↓0031 MNMRT
01                ;POWER FAIL AUTO RESTART/ WAIT FOR POWER UP
02 01251 020000 PWRUP:  LDA 0,0
03 01252 040427         STA 0,PWRS0       ;SAVE INTA ADRS
04 01253 063077         HALT              ;HALT THE CPU
05 01254 060277         INTDS             ;MAKE SURE ION IS OFF
06 01255 006116         JSR @LMESS
07 01256 004566         PFTEX
08 01257 024422         LDA 1,PWRS0
09 01260 006117         JSR @LPOCT
10 01261 006116         JSR @LMESS
11 01262 011202         RTTEX+1
12 01263 024133         LDA 1,RTTIM
13 01264 006062         JSR @LPDEC
14 01265 024140         LDA 1,ERTOT
15 01266 125004         MOV 1,1,SZR
16 01267 006062         JSR @LPDEC
17 01270 102400         SUB 0,0
18 01271 034412         LDA 3,AGSTRT
19 01272 054754         STA 3,LST,3
20 01273 000672         JMP PRSTL-4
21 01274 102400 GSTRT:  SUB 0,0
22 01275 040121         STA 0,PFAIS
23 01276 042406         STA 0,@ALAUTO
24 01277 006406         JSR @ALDSPR
25 01300 001033         LTBL1
26 01301 000000 PWRS0:  0
27 01302 000404         JMP LSYSR
28 01303 001274 AGSTRT: GSTRT
29 01304 001032 ALAUTO: LAUTO
30 01305 001022 ALDSPR: LDSPR
```

```
↓0032 MNMRT
01                ;LSYSR-RESET SYSTEM
02                ;SET LOGICAL PAGE 37=PHYS 37
03                ;SET MPSWT=0 NO MAP  #1'S IF MAP OPTION
04 01306 061077 LSYSR:  I.RST
05 01307 102400         SUB 0,0
06 01310 040000         STA 0,0
07 01311 040140         STA 0,ERTOT       ;CLR ACCUM ERRS
08 01312 040136         STA 0,TOTPK       ;
09 01313 040141         STA 0,EACTV
10 01314 126000         ADC 1,1
11 01315 044135         STA 1,UDEVI       ;UNEXPECTED INTERRUPT FLAG
12 01316 001400         JMP 0,3           ;EXIT NO MAPPING
13 01317 076000 LS.K2:  76000
```

```
10033 MNMRT
01
02                ;LWSET-SET UP SYSTEM FOR RUNNING
03                ;DCHMAP AND INIT MAP OPTION TRAP LOCATIONS
04 01320 022442 LWSET:  LDA 0,@LW.K1     ;GET NMAX
05 01321 040450         STA 0,LSYTB
06 01322 040450         STA 0,LSETB
07 01323 024462         LDA 1,PROGK
08 01324 124400         NEG 1,1
09 01325 044436         STA 1,LW.C1
10 01326 030444 LWS.1:  LDA 2,LSETB
11 01327 102400         SUB 0,0
12 01330 041000         STA 0,0,2        ;SET UP MEM ALLOC
13 01331 151400         INC 2,2          ;TABLES FOR 1 PROG
14 01332 041000         STA 0,0,2        ;1ST 2 WRDS =0
15 01333 151400         INC 2,2          ;NXT 16=-1
16 01334 024430         LDA 1,LW.K4
17 01335 100000         COM 0,0
18 01336 041000 LWS.2:  STA 0,0,2
19 01337 151400         INC 2,2
20 01340 125404         INC 1,1,SZR      ;STORE -1 16 TIMES
21 01341 000775         JMP LWS.2
22 01342 050430         STA 2,LSETB      ;NEW END SYS TABLES
23 01343 010420         ISZ LW.C1
24 01344 000762         JMP LWS.1        ;DO ONE MORE TABLE
25 01345 042420         STA 0,@LW.K5     ;DCH ALLOC TABLE
26 01346 042420         STA 0,@LW.K5+1   ;AVAILABLE 32K
27 01347 014423         DSZ LSETB        ;REAL END OF SYS TABLES
28 01350 024417         LDA 1,LW.K6
29 01351 030417         LDA 2,LW.K7
30 01352 100000         COM 0,0
31 01353 041000 LWS.3:  STA 0,0,2        ;CLEAR CORE EXIST MAP
32 01354 151400         INC 2,2
33 01355 125404         INC 1,1,SZR
34 01356 000775         JMP LWS.3
```

```
10034 MNMRT
01
02                ;NOW SET UP JMP @'S IN MAP TRAPS
03 01357 020053         LDA 0,ICDIS
04 01360 040047         STA 0,TPADR
05 01361 001400         JMP 0,3
06 01362 001004 LW.K1:  USTNM
07 01363 000000 LW.C1:  0
08 01364 177760 LW.K4:  -16.
09 01365 011372 LW.K5:  DCHM0
10 01366 011373         DCHM1
11 01367 177776 LW.K6:  -2
12 01370 011366 LW.K7:  EXISM
13 01371 000000 LSYTB:  0
14 01372 000000 LSETB:  0
15
16
17                ;GPRGK-GENERATE PROGRAM COUNT
18                ;THE FOLLOWING SUBROUTINE SIMPLY DETERMINES
19                ;HOW MANY TEST PROGRAMS ARE IN CORE
20                ;ALONG WITH THE DIAGNOSTIC LINKER
21                ;ZMAX-LAST LINKER ZLOC=#TESTS INTO PROGK
22 01373 024410 GPRGK:  LDA     1,KLZMX ;LAST LINKER ZLOC
23 01374 022410         LDA     0,@ISTZM         ;LAST ZPAGE FILLED
24 01375 111000         MOV     0,2
25 01376 122400         SUB     1,0      ;AC0=NUMBER TESTS
26 01377 040406         STA     0,PROGK
27 01400 102400         SUB     0,0
28 01401 041000         STA     0,0,2
29 01402 001400         JMP     0,3
30 01403 000147 KLZMX: LZMAX
31 01404 001001 ISTZM: USTZM
32 01405 000000 PROGK: 0
```

```
;0035 MNMRT
01                      ;SET UP UP32L AND HIGHK
02                      ;EXISTM=0'S
03 01400 054412 LSIZE:  STA     3,LS.S3
04 01401 004415         JSR     MSZ32   ;SIZE 0 TO 32K
05 01410 040411         STA     0,UP32L ;AC0=LAST ADRS IN 32K
06 01411 024706         LDA     1,LS.K2 ;5 BITS PHYS PAGE 37
07 01412 107400         AND     0,1
08 01413 125300         MOVS    1,1     ;AC1=LAST PHYS PAGE(32K)
09 01414 125220         MOVZR   1,1
10 01415 125220         MOVZR   1,1
11 01416 044404         STA     1,HIGHK ;IN CASE NOT 32K OR NO MAP
12 01417 000404         JMP     LSIZR
13 01420 000000 LS.S3:  0
14 01421 000000 UP32L:  0
15 01422 000000 HIGHK:  0
16                      LSIZR:
17 01423 002775         JMP @LS.S3      ;RETURN
```

```
;0036 MNMRT
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                      ;SIZES IN 1K INCR EXIT AC0=HIGHEST AVAIL ADRS.
09                      ;
10 01424 054435 MSZ32:  STA     3,XMS32 ;SAVE
11 01425 126400         SUB     1,1     ;0FOR FIRST 1K
12 01426 030431         LDA     2,K1K   ;1777 FOR END OF 1K
13 01427 133000         ADD     1,2     ;+CURRENT 1K FIELD
14 01430 025000         LDA     1,0,2   ;GET CELL
15 01431 120000         COM     1,0     ;CHNG BITS
16 01432 041000         STA     0,0,2
17 01433 021000         LDA     0,0,2   ;=COM MEM EXISTS
18 01434 122405         SUB     1,0,SNR ;AND WE'LL SKIP
19 01435 000420         JMP     M32SZ   ;LST WAS NONEXIST
20 01436 050422         STA     2,M32TEM
21 01437 045000         STA     1,0,2   ;RESTORE CELL
22 01440 141300         MOVS    2,0
23 01441 101200         MOVR    0,0
24 01442 101200         MOVR    0,0
25 01443 024417         LDA     1,K37
26 01444 123400         AND     1,0     ;AC0=PHYS PAGE #
27 01445 030416         LDA     2,KXIST ;ADRS EXIST TABLE
28 01446 006416         JSR     @MS.L1  ;SET EXIST BIT=1
29 01447 101001         MOV     0,0,SKP
30 01450 063077         HALT    ;*****CAN'T HAPPEN EXIST BIT HAD TO =0
31 01451 024407         LDA     1,M32TE ;AC1=LAST 1K TOP ADDRESS
32 01452 125400         INC     1,1
33 01453 125133         MOVZL#  1,1,SNC ;SKP IF LAST CELL =32K
34 01454 000752         JMP     MSZ32+2 ;NOT DONE SIZING
35 01455 020403 M32SZ:  LDA     0,M32TE ;AC0=HIGHEST AVAIL.
36 01456 002403         JMP     @XMS32
37 01457 001777 K1K:    1777
38 01460 000000 M32TE:  0
39 01461 000000 XMS32:  0
40 01462 000037 K37:    37
41 01463 011366 KXIST:  EXISM
42 01464 002407 MS.L1:  CMAPB
```

```
⁞0037 MNMRT
01                      ;MVETA-MOVE THE EXIST MAP
02                      ;INT THE AVAILABLE MAP POSITION
03 01465 054413 MVETA:  STA     3,XMVET
04 01466 030413         LDA     2,KEXMP ;A2=STR EXIST
05 01467 034413         LDA     3,KAVMP ;A3=STR AVAILABLE
06 01470 024413         LDA     1,KM2   ;=-2 FOR COUNTING
07 01471 021000         LDA     0,0,2   ;WORD
08 01472 041400         STA     0,0,3   ;TO WORD
09 01473 151400         INC     2,2
10 01474 175400         INC     3,3
11 01475 125404         INC     1,1,SZR ;SKP IS DONE 8
12 01476 000773         JMP     .-5
13 01477 002401         JMP     @XMVET  ;
14 01500 001465 XMVET:  MVETA
15 01501 011366 KEXMP:  EXISM
16 01502 011370 KAVMP:  AVALM
17 01503 177776 KM2:    -2
18                      ;UBL32-SET UP USUABLE SCRATCH LIMITS  IN 32K
19                      ;ALL CORE ABOVE 32K IS ASSUMED TO BE USEABLE SCRATCH
20                      ;CLEAR AVAILABLE BITS FOR THOSE AREAS USED
21                      ;SO THAT THEY WILL NOT BE ASSIGNED AS A SCRATCH AREA
22                      ;TO ANY TEST
23                      ;SUBR CBLIM IS USED TO CLEAR AVAILABLE BITS
24 01504 054416 UBL32:  STA     3,XUBL3 ;SAVE RETURN
25 01505 102400         SUB     0,0
26 01506 026415         LDA     1,@KNMAX        ;0 TO NMAX
27 01507 006422         JSR     @UBLIM  ;PROTECTS PROGRAMS
28 01510 022414         LDA     0,@KUP32        ;LAST ADRS IN 32K
29 01511 105000         MOV     0,1
30 01512 006417         JSR     @UBLIM  ;PROTECTS LOADER
31 01513 026412         LDA     1,@KSTSS        ;STRT SYMBOLS
32 01514 022412         LDA     0,@KSTES        ;END SYMBOLS
33 01515 006414         JSR     @UBLIM  ;PROTECTS SYMBOLS
34 01516 022411         LDA     0,@KLSTB        ;STRT LINKER TABLES
35 01517 026411         LDA     1,@KLETB        ;END LINKER TABLES
36 01520 006411         JSR     @UBLIM  ;PROTECTS LINKER TABLES
37 01521 002401         JMP     @XUBL3  ;RETURN
38 01522 000000 XUBL3:  0
39 01523 001004 KNMAX:  USTNM
40 01524 001421 KUP32:  UP32L
41 01525 001002 KSTSS:  USTSS
42 01526 001003 KSTES:  USTES
43 01527 001371 KLSTB:  LSYTB
44 01530 001372 KLETB:  LSETB
45 01531 002327 UBLIM:  CBLIM
```

```
⁞0038 MNMRT
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 01532 054434 TINIT:  STA     3,XTINI
07 01533 102400         SUB     0,0
08 01534 040433         STA     0,NPROG ;PROG TO INIT
09 01535 026546         LDA     1,@LROGK
10 01536 034431         LDA     3,NPROG ;NEXT PROG TO INIT
11 01537 166415         SUB#    3,1,SNR ;SKP IS NOT DONE ALL
12 01540 002426         JMP     @XTINI  ;EXIT ALL PROGS INITED
13 01541 020540         LDA     0,LR,K1
14 01542 117000         ADD     0,3
15 01543 031400         LDA     2,0,3   ;GET INIT ADRS
16 01544 020121         LDA     0,PFAIS
17 01545 101024         MOVZ    0,0,SZR ;SKP IF P/F RESTART
18 01546 000407         JMP     TINIA   ;NOT RESTART
19 01547 021002         LDA     0,2,2   ;GET WAITI SWITCH
20 01550 101045         MOVO    0,0,SNR ;SKP=MAYBE DELETED
21 01551 000404         JMP     TINIA   ;RESTART NOT DELETED
22 01552 021377         LDA     0,-1,2
23 01553 101045         MOVO    0,0,SNR ;SKP=PROG WAS SELECTED
24 01554 000410         JMP     XTINI-2
25 01555 102460 TINIA:  SUBC    0,0
26 01556 041375         STA     0,-3,2  ;0 PASS K
27 01557 041376         STA     0,-2,2  ;0 ERR CTR
28 01560 041002         STA     0,2,2   ;CLEAR WAIT INT SW
29 01561 101003         MOV     0,0,SNC ;SKP IF P/FAIL RESTART
30 01562 041377         STA     0,-1,2  ;CLRS INTA WAIT CTR
31 01563 007000         JSR     @0,2    ;AND INIT THIS TEST
32 01564 010403         ISZ     NPROG   ;STEP TO NXT PROG
33 01565 000750         JMP     TINIT+3 ;AND DO AGN
34 01566 000000 XTINI:  0
35 01567 000000 NPROG:  0
36
```

```
10039 MNMRT
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 01570 054512 LRANP:  STA 3,LR.S3
10 01571 102400         SUB 0,0
11 01572 030507         LDA 2,LR.K1       ;START OF TEST LINKS
12 01573 035000 LPRL1:  LDA 3,0,2         ;GET TEST LINK
13 01574 175005         MOV 3,3,SNR       ;SKP IS TEST EXISTS
14 01575 000433         JMP LPRL2         ;NO ONE INTR DONE USE RAN
15 01576 025402         LDA 1,2,3         ;GET INTR SW
16 01577 125005         MOV 1,1,SNR       ;WAITING INT OR DISABLED?
17 01600 000425         JMP LPS1E         ;NO TRY NEXT TEST
18 01601 125103         MOVL 1,1,SNC      ;SKP=WAIT BIT STILL ON
19 01602 000432         JMP LPRGO         ;ENTER THIS TEST
20 01603 025777         LDA 1,-1,3        ;GET INTA ELAPSED TIMER
21 01604 125005         MOV 1,1,SNR       ;SKP IF ACTIVE
22 01605 000420         JMP LPS1E
23 01606 034133         LDA 3,RTTIM       ;GET CUR RUN TIME
24 01607 166414         SUB# 3,1,SZR      ;SKP=RUN TIME ERR
25 01610 000415         JMP LPS1E
26 01611 040104         STA 0,CURPR
27 01612 006116         JSR @LMESS        ;TIME ELAPSED MESS.
28 01613 001717         LP.TX
29 01614 024104         LDA 1,CURPR       ;PRINT PROG#
30 01615 006061         JSR @PZUCT
31 01616 035000         LDA 3,0,2
32 01617 102400         SUB 0,0
33 01620 041777         STA 0,-1,3
34 01621 040137         STA 0,TIMSW       ;FORCE TIME TYPE
35 01622 010140         ISZ EXTOT
36 01623 000401         JMP .+1
37 01624 000745         JMP LRANP+1
38 01625 101400 LPS1E:  INC 0,0
39 01626 151400         INC 2,2
40 01627 000744         JMP LPRL1         ;TRY NEXT TEST
41                ;NO TEST WAITING TO PROCESS INTERRUPT RANDOM SELECT
42 01630 006067 LPRL2:  JSR @ARANG
43 01631 105000         MOV 0,1
44 01632 032451         LDA 2,@LROGK      ;DIV RAN/#PROGS
45 01633 006070         JSR @ADIVI
```

```
10040 MNMRT
01                ;SEE IF NEXT RAN# FITS BETWEEN TEST ENTER LIMITS
02                ;BUT DO NOT ENTER TEST IF IT IS WAITING INTA
03 01634 040104 LPRGO:  STA 0,CURPR       ;CURRENT PROG #
04 01635 030444         LDA 2,LR.K1       ;LAST LINKR ZLOC+1
05 01636 113000         ADD 0,2           ;A2=PNTR TO PARAM ADRS
06 01637 035000         LDA 3,0,2
07 01640 054131         STA 3,PSTRT       ;START ADDRES OF PRG
08 01641 031001         LDA 2,1,2         ;GET STRT NXT PROG
09 01642 151005         MOV 2,2,SNR       ;=0 IS LAST PRG SEL
10 01643 032441         LDA 2,@LW.K9      ;AND WE USE NMAX
11 01644 050132         STA 2,PENDA       ;AS THE END OF PROG
12 01645 014132         DSZ PENDA         ;-1 FOR REAL END OF PROG
13 01646 031402         LDA 2,2,3
14 01647 151132         MOVZL# 2,2,SZC    ;C=1 IS WAITING INTR
15 01650 000760         JMP LPRL2         ;SELECT DIFF PROG
16 01651 030133         LDA 2,RTTIM
17 01652 151600         INCR 2,2
18 01653 151500         INCL 2,2
19 01654 051777         STA 2,-1,3
20                ;IF ACS3=1 DELETE ALL RANDOM SELECT DELAYS
21 01655 034050         LDA 3,RNSEL       ;SW SET?
22 01656 175004         MOV 3,3,SZR
23 01657 000411         JMP LPRGA         ;DELETE DELAYS=1
24 01660 006067         JSR @ARANG        ;GET NEW RANDOM
25 01661 034131         LDA 3,PSTRT
26 01662 025403         LDA 1,3,3         ;GET PRG LWR LIMIT
27 01663 122433         SUBZ# 1,0,SNC
28 01664 000744         JMP LPRL2
29 01665 025404         LDA 1,4,3         ;GET HIGH LIMIT
30 01666 122032         ADCZ# 1,0,SZC
31 01667 000741         JMP LPRL2
32 01670 020104 LPRGA:  LDA 0,CURPR       ;PROGRAM #
33 01671 103020         ADDZ 0,0          ;*2
34 01672 105120         MOVZL 0,1         ;*4
35 01673 127120         ADDZL 1,1         ;*16
36 01674 107000         ADD 0,1           ;*18
37 01675 022410         LDA 0,@XSYTB      ;+START OF SYS TABLES
38 01676 123000         ADD 1,0
39 01677 040130         STA 0,ALTBL
40 01700 002402         JMP @LR.S3
41 01701 000147 LR.K1:  LZMAX
42 01702 000000 LR.S3:  0
43 01703 001405 LROGK:  PROGK
44 01704 001004 LW.K9:  USTNM
45 01705 001371 XSYTB:  LSYTB
```

```
⍳0041 MNMRT
01                ;LSTRP-START PROGRAM
02                ;ENTER TEST SELECTED AT ITS EXECUTION ENTRY POINT
03 01706 054407 LSTRP:  STA 3,L.SS3
04 01707 030131         LDA 2,PSTRT
05 01710 021001         LDA 0,1,2       ;GET EXEC ADRS
06 01711 040405         STA 0,LS.I1
07 01712 002404         JMP @LS.I1      ;ENTER TEST
08                ;LRETP-RETURN FROM TEST PROG CALL
09 01713 060177 LRETP:  INTEN
10 01714 002401         JMP @L.SS3      ;
11 01715 000000 L.SS3: 0
12 01716 000000 LS.I1: 0
13 01717 005215 LP.TX:  .TXTE (<15><12>INTERRUPT WAIT ELAPSED
14 01733 005215 <15><12>PROG. NO. (
```

```
⍳0042 MNMRT
01                ;RANGN-RANDOM # GENERATOR
02                ;SPIN #'S OUT IN A HURRY FORGET THE MATH
03 01742 044430 RANGN:  STA 1,RN.S1
04 01743 050430         STA 2,RN.S2
05 01744 030431         LDA 2,RN.K1     ;7 FOR MASKING AT 8
06 01745 020431         LDA 0,RN.C2     ;CYCLIC CONSTANT
07 01746 024426         LDA 1,RN.C1
08 01747 133404         AND 1,2,SZR     ;ROTAT C2 EVERY 8
09 01750 000404         JMP RAN.1
10 01751 101122         MOVZL 0,0,SZC
11 01752 101400         INC 0,0
12 01753 040423         STA 0,RN.C2
13 01754 024424 RAN.1:  LDA 1,RTABL
14 01755 133000         ADD 1,2         ;TO GET NXT SUM VAR
15 01756 025000         LDA 1,0,2
16 01757 123000         ADD 1,0
17 01760 041000         STA 0,0,2       ;NEW SUM IN VAR
18 01761 024416         LDA 1,RANNM     ;LAST RAN#
19 01762 123300         ADDS 1,0
20 01763 040110         STA 0,RNAC0
21 01764 040413         STA 0,RANNM
22 01765 024405         LDA 1,RN.S1
23 01766 030405         LDA 2,RN.S2
24 01767 010405         ISZ RN.C1
25 01770 001400         JMP 0,3
26 01771 001400         JMP 0,3
27 01772 000000 RN.S1: 0
28 01773 000000 RN.S2: 0
29 01774 000000 RN.C1: 0
30 01775 000007 RN.K1: 7
31 01776 123456 RN.C2: 123456
32 01777 000000 RANNM: 0
33 02000 002001 RTABL: RTABL+1
34 02001 027247         027247
35 02002 145651         145651
36 02003 162724         162724
37 02004 071352         071352
38 02005 034565         034565
39 02006 116272         116272
40 02007 047135         047135
41 02010 113523         113523
42 02011 054411 RANG3:  STA 3,RN.S3     ;FILL AC0 TO 2 WITH RAN #'S
43 02012 004730         JSR RANGN
44 02013 111000         MOV 0,2
45 02014 004726         JSR RANGN
46 02015 105000         MOV 0,1
47 02016 004724         JSR RANGN
48 02017 044111         STA 1,RNAC1
49 02020 050112         STA 2,RNAC2
50 02021 002401         JMP @RN.S3
51 02022 000000 RN.S3: 0
```

```
;0043 MNMRT
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 02023 054441 RMSEL:  STA 3,XRMSE
15 02024 023400         LDA 0,@0,3
16 02025 025401         LDA 1,1,3
17 02026 040433         STA 0,RM.P1
18 02027 044433         STA 1,RM.P2
19 02030 100000         COM 0,0
20 02031 040432         STA 0,RM.P3
21 02032 006067 RM.L1:  JSR @ARANG      ;GET RANDOM
22 02033 030426         LDA 2,RM.P1
23 02034 151400         INC 2,2
24 02035 105000         MOV 0,1
25 02036 006070         JSR @ADIVI      ;REM IN AC0=BIT SEL
26 02037 030423         LDA 2,RM.P2
27 02040 006425         JSR @ICMPB      ;SKIP IF BIT =1
28 02041 006424         JSR @ICMPB      ;CHNG IT BK TO 0 SKP
29 02042 000415         JMP RM.FN       ;FOUND 1 EXIT
30 02043 010420         ISZ RM.P3
31 02044 000766         JMP RM.L1
32 02045 020416 RM.L2:  LDA 0,RM.P3     ;GET NXT SEQ BIT
33 02046 006417         JSR @ICMPB      ;SKP IF =1
34 02047 006416         JSR @ICMPB      ;0 BIT AND SKIP
35 02050 000407         JMP RM.FN
36 02051 024410         LDA 1,RM.P1
37 02052 010411         ISZ RM.P3
38 02053 106404         SUB 0,1,SZR     ;SKP IF SRCHED WHOLE TBL
39 02054 000771         JMP RM.L2
40 02055 034407         LDA 3,XRMSE
41 02056 001402         JMP 2,3         ;NO FIND EXIT
42 02057 010405 RM.FN:  ISZ XRMSE
43 02060 000775         JMP RM.FN-2
44 02061 000000 RM.P1: 0
45 02062 000000 RM.P2: 0
46 02063 000000 RM.P3: 0
47 02064 000000 XRMSE: 0
48 02065 002407 ICMPB: CMAPB
```

```
;0044 MNMRT
01                ;DIVID-DIVIDE AC1 BY AC2
02                ;LEAVE WITH REM IN AC0
03 02066 102400 DIVID:  SUB     0,0
04 02067 054412         STA     3,DI.S3
05 02070 034412         LDA     3,DI.K1
06 02071 125120         MOVZL   1,1
07 02072 101100 DI.L1:  MOVL    0,0
08 02073 142412         SUB#    2,0,SZC
09 02074 142400         SUB     2,0
10 02075 125100         MOVL    1,1
11 02076 175404         INC     3,3,SZR
12 02077 000773         JMP     DI.L1
13 02100 002401         JMP     @DI.S3
14 02101 000000 DI.S3: 0
15 02102 177760 DI.K1: -16.
```

```
↓0045 MNMRT
01
02                ;CHANGE STACK POSITIONS IN MEMORY
03 02103 054452 CHSTK:  STA 3,CHSAV
04 02104 006067         JSR @ARANG
05 02105 101203         MOVR 0,0,SNC     ;BITS 14/15=00
06 02106 101202         MOVR 0,0,SZC     ;THEN CHANGE STACK
07 02107 000454         JMP SPLAY        ;PLAY WITH STACK
08                ;CHECK TO SEE IF STACK OK BEFORE CHANGE
09 02110 071201         MFSP 2
10 02111 024445         LDA 1,STADR
11 02112 050445         STA 2,NWSTK
12 02113 132414         SUB# 1,2,SZR     ;SKP=STACK OK
13 02114 004537         JSR SKERR        ;STACK IS FUNNY
14 02115 151300         MOVS 2,2
15 02116 151225         MOVZR 2,2,SNR    ;SKP IF STK NOT AT 400
16 02117 000421         JMP CMPAG        ;MOVE IT ELSEWHERE
17                ;STACK IS NOT AT 400 RELEASE 1K PAGE
18                ;AFTER RESTORING STACK TO ADRS 400
19 02120 034441         LDA 3,CHSK1      ;400
20 02121 054435         STA 3,STADR
21 02122 060277         INTDS
22 02123 075001         MTSP 3
23 02124 060177         INTEN
24 02125 074001         MTFP 3
25 02126 030414         LDA 2,CMPAG+2    ;AVALM
26 02127 020431         LDA 0,STKPG      ;PHYS PAG#
27 02130 101005         MOV 0,0,SNR      ;SHD NOT BE PAG0
28 02131 004522         JSR SKERR        ;STACK IS IN ERROR
29 02132 006430         JSR @CHSK1+1     ;CMAPB MAKE 1K AVAL
30 02133 102401         SUB 0,0,SKP      ;;BIT HAD TO GO 0 TO1
31 02134 004517         JSR SKERR        ;THIS CAN'T HAPPEN
32 02135 040423         STA 0,STKPG
33 02136 002417         JMP @CHSAV
34
35
36 02137 000037 STKLM:  37       ;LIMIT STACK TO 32K
37                ;RANDOMLY SELECT 1K OF CORE TO MOV THE STACK INTO
38 02140 004663 CMPAG:  JSR RMSEL        ;SELECTS 1K CORE
39 02141 002137         STKLM            ;WITHIN LIMIT
40 02142 011370         AVALM            ;OF AVAL CORE
41 02143 002412         JMP @CHSAV       ;NONE AVAL TO USE
42 02144 040414         STA 0,STKPG      ;SAVE PHYS PAGE
43 02145 101300         MOVS 0,0
44 02146 103120         ADDZL 0,0
45 02147 040407         STA 0,STADR
46 02150 060277         INTDS
47 02151 060001         MTFP 0
48 02152 060177         INTEN
49 02153 061001         MTSP 0
50 02154 002401         JMP @CHSAV
```

```
↓0046 MNMRT
01
02 02155 000000 CHSAV:  0
03 02156 000000 STADR:  0
04 02157 000000 NWSTK:  0
05 02160 000000 STKPG:  0
06 02161 000400 CHSK1:  400
07 02162 002407         CMAPB
08
09                ;DO SOME FUNNY THINGS WITH STACK TO
10                ;TRY TO GOOF IT UP
11 02163 071201 SPLAY:  MFSP 2
12 02164 126000         ADC 1,1 ;-1
13 02165 133000         ADD 1,2
14 02166 044411         STA 1,STKIN-1    ;STK INTA FLG
15 02167 060277         INTDS
16 02170 071001         MTSP 2           ;RELOAD STACK POINTER
17 02171 061401         PSH 0
18 02172 025001         LDA 1,1,2
19 02173 060177         INTEN
20 02174 122414         SUB# 1,0,SZR     ;(AC0) GET THERE OK
21 02175 004456         JSR SKERR        ;NO! PSH ERROR
22 02176 002757         JMP @CHSAV
23
24                ;STACK OVERFLOW ROUTINE
25 02177 000000         0
26 02200 040427 STKIN:  STA 0,STSV0
27 02201 044427         STA 1,STSV1
28 02202 050427         STA 2,STSV2
29 02203 054427         STA 3,STSV3
30 02204 101200         MOVR 0,0
31 02205 040426         STA 0,STSVC
32 02206 010771         ISZ STKIN-1
33 02207 000402         JMP .+2
34 02210 000406         JMP .+6
35 02211 006441         JSR @SKERR-1                    ;PRINT ERR MESS.
36 02212 002234         STKTZ
37 02213 024000         LDA 1,0
38 02214 006435         JSR @SKERR-2
39 02215 006411         JSR @ST.K6       ;WAIT FOR OPERATOR INPUT
40 02216 034414         LDA 3,STSV3
41 02217 030412         LDA 2,STSV2
42 02220 024410         LDA 1,STSV1
43 02221 020412         LDA 0,STSVC
44 02222 101100         MOVL 0,0
45 02223 020404         LDA 0,STSV0
46 02224 060177         INTEN
47 02225 002000         JMP @0           ;RETURN
48
49 02226 003174 ST.K6:  KEY6W
50 02227 000000 STSV0:  0
51 02230 000000 STSV1:  0
52 02231 000000 STSV2:  0
53 02232 000000 STSV3:  0
54 02233 000000 STSVC:  0
55 02234 005215 STKTZ:  .TXTE [<15><12>STACK OVERFLOW ERROR @]
56       000000         .NOLOC 0
```

```
10047 MNMRT
01
02                ;STACK IS IN ERROR SOMEHOW
03                ;MAKE ERROR TYPEOUT
04 02251 003641           PUC?
05 02252 003612           MES?
06 02253 054020 SKERR:    STA 3,20
07 02254 006776           JSR @SKERR-1
08 02255 002303           STKTX
09 02256 024020           LDA 1,20          ;ADRS ERR DETECTED
10 02257 006772           JSR @SKERR-2
11 02260 024676           LDA 1,STADR       ;WHERE STACK SHD BE
12 02261 006770           JSR @SKERR-2
13 02262 024675           LDA 1,NWSTK
14 02263 006766           JSR @SKERR-2      ;WHERE IT'S AT
15                ;RESTORE STACK TO 400
16 02264 034675           LDA 3,CHSK1
17 02265 060277           INTDS
18 02266 075001           MTSP 3
19 02267 074001           MTFP 3
20 02270 060177           INTEN
21 02271 054665           STA 3,STADR
22 02272 020666           LDA 0,STKPG       ;MAKE 1K AVAL
23 02273 101005           MOV 0,0,SNR       ;IF NOT PAGE 0
24 02274 002661           JMP @CHSAV
25 02275 030645           LDA 2,CMPAG+2
26 02276 006664           JSR @CHSK1+1
27 02277 102401           SUB 0,0,SKP       ;IT'S RELEASED
28 02300 000776           JMP .-2
29 02301 040657           STA 0,STKPG
30 02302 002653           JMP @CHSAV
31 02303 005215 STKTX:    .TXTE |<15><12>STACK ERROR<15><12>
32 02312 142412 ERROR:    EXPECTED ACTUAL<15><12>|
```

```
10048 MNMRT
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 CMAPB WILL BE REQ
06 02327 054426 CBLIM:    STA     3,XCBLM
07 02330 101300           MOVS    0,0       ;POS 1K
08 02331 125300           MOVS    1,1       ;FIELD BITS
09 02332 101220           MOVZR   0,0       ;FOR ADRS LIMITS
10 02333 101220           MOVZR   0,0       ;IN AC 0 AND 1
11 02334 125220           MOVZR   1,1
12 02335 125220           MOVZR   1,1
13 02336 030424           LDA     2,K37C
14 02337 143400           AND     2,0
15 02340 147400           AND     2,1
16 02341 040417           STA     0,CBLWR   ;LWST 1K FLD
17 02342 044417           STA     1,CBUPR   ;HGHST 1K FLD
18 02343 020415           LDA     0,CBLWR   ;LWR INCS TO=UPR
19 02344 030413           LDA     2,KAVLM   ;AVAILABLE MAP
20 02345 006411           JSR     @XCMPB    ;COM BIT IN AVAIL MAP
21 02346 000777           JMP     .-1       ;WENT 0-1 MAKE IT 1-0
22 02347 020411           LDA     0,CBLWR   ;AC0=LST CLRED
23 02350 010410           ISZ     CBLWR     ;+1 LWR IN CASE NOT DONE
24 02351 024410           LDA     1,CBUPR
25 02352 106404           SUB     0,1,SZR   ;SKP IS ALL REQ UNAVAILABLE
26 02353 000770           JMP     .-10
27 02354 002401           JMP     @XCBLM
28 02355 000000 XCBLM:    0
29 02356 002407 XCMPB:    CMAPB
30 02357 011370 KAVLM:    AVALM
31 02360 000000 CBLWR:    0
32 02361 000000 CBUPR:    0
33 02362 000037 K37C:     37
```

```
10049 MNMRT
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 DOES NOT =377
05                ;THE INTEGRETY OF AC0 IS PRESERVED
06 02363 040421 GETPA:  STA     0,GPA,0
07 02364 126400         SUB     1,1
08 02365 101220         MOVZR   0,0
09 02366 125100         MOVL    1,1     ;0=WORD #  1=EVEN/ODD
10 02367 124000         COM     1,1
11 02370 044415         STA     1,GETP. ;SKP SWAP IF EVEN BYTE
12 02371 030130         LDA     2,ALTBL ;ADRS OF ALLOCATION TABLE
13 02372 113000         ADD     0,2     ;+ WORD POSITION IN TABLE
14 02373 025002         LDA     1,2,2   ;2 PAST CTR AND OCM CONTR.
15 02374 010411         ISZ     GETP.   ;SKP SWAP IF EVEN BYTE
16 02375 125300         MOVS    1,1     ;SWAP ODD BYTE TO LWR
17 02376 030410         LDA     2,K.377
18 02377 147400         AND     2,1     ;MASK PHYSICAL PAGE #
19 02400 146414         SUB#    2,1,SZR ;377 IS 128K WK PROTEC
20 02401 175400         INC     3,3     ;SKP EXIT PAGE ASSIGNED
21 02402 020402         LDA     0,GPA,0
22 02403 001400         JMP     0,3
23 02404 000000 GPA,0:  0
24 02405 000000 GETP.:  0
25 02406 000377 K.377:  377
```

```
10050 MNMRT
01                ;CMAP=-COMPLIMENT MAP BIT
02                ;COMPLIMENT THE STATE OFAMEMORY MAP BIT
03                ;THE START ADRS OF THE MAP IS INAC2
04                ;AC0 CONTAINS UP TO 7 BITS OF ADDRESSWITH
05                ;BITS 12TO15=# BITS TO SHIFT LEFT
06                ;BITS 9TO11=WORD POSITION IN MAP
07                ;THOSE 7 BITS ARE THE PHYSICL PG # OF A 1K OF MEM
08                ;SKIP ON EXIT IF THE BIT IS GOING 1-0
09 02407 040436 CMAPB:  STA     0,CM.S0
10 02410 044436         STA     1,CM.S1
11 02411 050436         STA     2,CM.S2
12 02412 105220         MOVZR   0,1     ;POSITION WORD # IN AC1
13 02413 125220         MOVZR   1,1
14 02414 125220         MOVZR   1,1
15 02415 125220         MOVZR   1,1
16 02416 133000         ADD     1,2     ;AC2=TABLE (MAP) ADRESS
17 02417 024431         LDA     1,K17
18 02420 107400         AND     0,1     ;AC1=#PLACES TO SHIFT
19 02421 124000         COM     1,1
20 02422 044422         STA     1,CCTR  ;FOR COUNTING SHIFTS
21 02423 126420         SUBZ    1,1     ;AC1=0  C=1
22 02424 125100         MOVL    1,1     ;POSIT BIT
23 02425 010417         ISZ     CCTR
24 02426 000776         JMP     .-2
25 02427 021000         LDA     0,0,2   ;GET MAP WORD
26 02430 050414         STA     2,CCTR  ;SV ADRS
27 02431 131000         MOV     1,2     ;FOR BIT XOR
28 02432 113525         ANDZL   0,2,SNR ;NOT 0 IN RESULT
29 02433 151002         MOV     2,2,SZC ;OR IN CARRY BIT
30 02434 175400         INC     3,3     ;IS SKIP WHEN EXIT
31 02435 107000         ADD     0,1     ;FORM REST OF
32 02436 146400         SUB     2,1     ;BIT XOR
33 02437 046405         STA     1,@CCTR ;PUT NEW WORD BACK
34 02440 020405         LDA     0,CM.S0
35 02441 024405         LDA     1,CM.S1
36 02442 030405         LDA     2,CM.S2
37 02443 001400         JMP     0,3
38 02444 000000 CCTR: 0
39 02445 000000 CM.S0: 0
40 02446 000000 CM.S1: 0
41 02447 000000 CM.S2: 0
42 02450 000017 K17:    17
```

```
;0051 MNMRT
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                      ;ADMAP  ASSIGN DCM MAP
18                      ;EDMAP  EXPAND DCM MAP
19                      ;RDMAP  RELEASE DCM FROM MAP
```

```
;0052 MNMRT
01                      ;ILLEGAL SUPERVISOR CALL TYPEOUT
02 02451 000116 ICALL:  JSR @LMESS
03 02452 002473         ICALT           ;TEXT
04 02453 024511         LDA 1,CD.LA
05 02454 000117         JSR @LPUCT
06 02455 020501         LDA 0,CD.S0
07 02456 024501         LDA 1,CD.S1
08 02457 030501         LDA 2,CD.S2
09 02460 000072         JSR @ERRDI      ;TYPE PRM AC'S ETC
10 02461 000401         JMP .+1
11 02462 020477         LDA 0,CD.S3
12 02463 024503         LDA 1,CD.PP
13 02464 030477         LDA 2,CD.IN     ;INSTRUCTION CAUSING TRAP
14 02465 000073         JSR @ERRDC
15 02466 000401         JMP .+1
16 02467 030131         LDA 2,PSTNT
17 02470 035006         LDA 3,6,2
18 02471 175120         MOVZL 3,3
19 02472 000416         JMP ICALX
20 02473 005215 ICALT:  .TXTE (<15><12>
21 02474 144411         ILLEGAL SUPER CALL AT (
22                      ;
23                      ;
24                      ;
25                      ;ICALX- TRAP RETURN IF ILLEGAL CALL BECAUSE OF USESM
26                      ;
27 02510 175220 ICALX:  MOVZR 3,3       ;AC0-1-2 = MAP REGISTERS
28 02511 054402         STA 3,ICLX,3    ;
29 02512 002401         JMP @ICLX,3
30 02513 000000 ICLX.3: 0
```

```
10053 MNMRT
01                      ;SYSTEM CALL DISPATCH ROUTINE
02 02514 040442 CDISP:  STA 0,CD.S0
03 02515 044442         STA 1,CD.S1
04 02516 050442         STA 2,CD.S2
05 02517 054442         STA 3,CD.S3
06 02520 006451         JSR @CDODT      ;CK FOR OPERATOR REQUESTS
07 02521 034046         LDA 3,46        ;GET PC
08 02522 021400 CDGLC:  LDA 0,0,3       ;GET CALL
09 02523 054441         STA 3,CD.LA     ;SV LGICL ADRS
10 02524 040437         STA 0,CD.IN     ;SAVE INSTRUCTION CAUSING TRAP
11 02525 030435         LDA 2,K1777
12 02526 150000         COM 2,2
13 02527 157400         AND 2,3
14 02530 054435         STA 3,CD.LP     ;SAVE LOG PAGE
15 02531 054435         STA 3,CD.PP     ;SAVE PHYS. PG#
16                      ;NOW DETERMINE IF IT IS A VALID CALL
17 02532 030435         LDA     2,CALLS ;START OF CALLS
18 02533 142433         SUBZ#   2,0,SNC ;MUST BE=>
19 02534 000715         JMP     ICALL           ;ILLEGAL CALL?
20 02535 024433         LDA     1,CALLE ;LAST VALID CALL
21 02536 122032         ADCZ#   1,0,SZC ;MUST BE =<
22 02537 000712         JMP     ICALL           ;INVALID CALL?
23 02540 142640         SUBOR 2,0       ;CREATE CALL ADRS
24 02541 115220         MOVZR 0,3       ;MOV 4 R
25 02542 175220         MOVZR   3,3
26 02543 175220         MOVZR   3,3     ;(AC3)=CALL#
27 02544 020412         LDA     0,CD.S0
28 02545 024412         LDA     1,CD.S1
29 02546 030412         LDA     2,CD.S2
30 02547 007454 CD.EX:  JSR     @ASCRA,3        ;CALL JSR
31 02550 101011         MOV#    0,0,SKP ;ERROR RET
32 02551 010413         ISZ     CD.LA   ;NORMAL +1 RETURN
33 02552 010412         ISZ     CD.LA   ;TO GET PAST JSR
34 02553 034406         LDA 3,CD.S3
35 02554 060177         INTEN
36 02555 002407         JMP     @CD.LA
37                      ;ABOVE JMP RETURNS TO USER
38 02556 000000 CD.S0: 0
39 02557 000000 CD.S1: 0
40 02560 000000 CD.S2: 0
41 02561 000000 CD.S3: 0
42 02562 001777 K1777: 1777
43 02563 000000 CD.IN:  0
44 02564 000000 CD.LA: 0
45 02565 000000 CD.LP: 0
46 02566 000000 CD.PP: 0
47                      CALLS:  LCALL ASCRA
48 02567 100010                 ASCRA=ASCRA*1B11+100010
49                      CALLE:  LCALL RUMAP
50 02570 100570                 RUMAP=ASCRA*1B11+100010
51 02571 001060 CDODT:  CKODT
```

```
10054 MNMRT
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 02572 054426 ENTPA:  STA     3,XNTPA
06 02573 026130         LDA     1,@ALTBL        ;GET # ENTRIES
07 02574 152400         SUB     2,2
08 02575 125220         MOVZR   1,1     ;AC1 =WORD #
09 02576 151100         MOVL    2,2     ;AC2=EVEN/ODD BYTE
10 02577 150000         COM     2,2     ;SKP/NSKP EVEN BYTE
11 02600 050416         STA     2,ENT.2 ;NSKP/SKP ODD BYTE
12 02601 030130         LDA     2,ALTBL ;ADRS OF ALLOC TBLE
13 02602 133000         ADD     1,2     ;+ WORD #
14 02603 025002         LDA     1,2,2   ;+2 TO GET BY CTRS
15 02604 010412         ISZ     ENT.2   ;SKP IF EV BYTE
16 02605 125300         MOVS    1,1     ;GET ODD BYTE LOW
17 02606 034411         LDA     3,ENT.K ;177400 MSK UPR BYTE
18 02607 167400         AND     3,1
19 02610 107300         ADDS    0,1     ;PHYS PG# ENTERED
20 02611 010405         ISZ     ENT.2   ;SKP IS ODD BYTE
21 02612 125300         MOVS    1,1     ;REPO EVEN BYTE
22 02613 045002         STA     1,2,2   ;PUT ENTRY BACK
23 02614 012130         ISZ     @ALTBL  ;+1# OF ENTRIES
24 02615 002403         JMP     @XNTPA
25 02616 000000 ENT.2: 0
26 02617 177400 ENT.K: 177400
27 02620 000000 XNTPA: 0
```

```
10055 MNMRT
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 02021 062401           SAVE
06 02022 102400           SUB     0,0
07 02023 040143           STA     0,SCRLO
08 02024 040144           STA     0,SCRHI
09 02025 000404           JMP LM.NM
```

```
10056 MNMRT
01                ;MAP OPTION IS SET UP FOR A FIRST LEVEL TEST
02                LM.ON:
03 02026 062601           RTRN
04 02027 002363 LGTPA: GETPA
05 02030 001777 LM.K2:  1777
```

```
10057 MNMRT
01                ;MAP OPTION DOES NOT EXIST
02                ;SIMPLY SET LIMITS TO SCRATCH AREA ASSIGNED
03 02631 006776 LM.NM:  JSR     @LGTPA  ;SKP=AC1 PHYS PAGE#
04 02632 000774         JMP     LM.ON   ;EXIT NO SCRATCH
05 02633 125300         MOVS    1,1
06 02634 127120         ADDZL   1,1
07 02635 044143         STA     1,SCRLO ;LOW=FIRST PHYS 1K
08 02636 006771 LM.L4:  JSR     @LGTPA  ;SKP=AC1=PHYS PG#
09 02637 000767         JMP     LM.ON   ;EXIT SCRHI ADJUSTED
10 02640 125300         MOVS    1,1
11 02641 127120         ADDZL   1,1     ;PG# POSITIONED TO PHYS
12 02642 030766         LDA     2,LM.K2
13 02643 133000         ADD     1,2
14 02644 050144         STA     2,SCRHI ;NO TEST CAN HAVE 32K IF
15 02645 101400         INC     0,0
16 02646 000770         JMP     LM.L4   ;MAP OPTION NONEXIST
```

```
10058 MNMRT
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 02647 062401           SAVE
08 02650 054414           STA     3,AS.S3
09 02651 022130           LDA     0,@ALTBL        ;GET #1K'S ASSIGNED
10 02652 101004           MOV     0,0,SZR ;NOT=0 INVALID
11 02653 000407           JMP     AS.XT
12 02654 006411 AS.G1:    JSR     @MSEL   ;SELECT A PAGE
13 02655 001422           HIGHK           ;MAX # 1K PAGES
14 02656 011370           AVALM           ;AVAILABLE MAP
15 02657 000403           JMP     AS.XT   ;NO CORE AVAILABLE
16 02660 004712           JSR     ENTPA   ;AC0=PHYS PAGE# ENTER IT
17 02661 012403           ISZ @AS.S3
18 02662 004737 AS.XT:    JSR     LDMAP   ;LOAD MAP OPT, SET SCRLO+HI
19 02663 062601           RTRN
20 02664 000000 AS.S3: 0
21 02665 002023 MSEL: RMSEL
```

```
10059 MNMRT
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 02666 062401           SAVE
08 02667 054775           STA     3,AS,S3
09 02670 022130           LDA     0,@ALTBL        ;# 1K'S ASSIGNED
10 02671 101005           MOV     0,0,SNR
11 02672 000770           JMP     AS,XT   ;CANT EXPAND 0 ASSIGNED
12                ;MAPPING OPTION DOES NOT EXIST ASSIGN NEXT SEQ 1K
13                ;UNLESS IT IS ALREADY BEING USED
14 02673 100400           NEG     0,0
15 02674 100000           COM     0,0     ;CALC ALLOC TBLE POS
16 02675 006437           JSR     @IGTPA  ;EXTRACT PHYS PGE #
17 02676 063077           HALT            ;#=377 CAN'T HAPPEN
18 02677 121400           INC     1,0     ;AC0=NXT PHYS PAGE
19 02700 030756           LDA     2,AS,G1+2       ;ADRS OF AVAILABLE TBL
20 02701 006406           JSR     @EX.I1  ;CMAPB SKPS IF AVAILABLE
21 02702 101001           MOV     0,0,SKP
22 02703 000755           JMP     AS,XT-2 ;1K AVAILABLE ENTER AND REMAP
23 02704 006403           JSR     @EX.I1  ;CMAPB HAS TO SKP
24 02705 063077           HALT
25 02706 000754           JMP     AS,XT   ;EXIT NO EXPANSION
26 02707 002407 EX.I1: CMAPB
```

```
10060 MNMRT
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 02710 062401           SAVE
07 02711 054753           STA     3,AS,S3
08 02712 022130           LDA     0,@ALTBL        ;#1K'S ASSIGNED
09 02713 100405           NEG     0,0,SNR         ;SKP IF ANY ASSIGNED
10 02714 000746           JMP     AS,XT           ;EXIT NONE TO RELESE
11 02715 100000           COM     0,0             ;AC0=#1K'S -1
12 02716 042130           STA     0,@ALTBL        ;TO ENTER 377 LATER
13 02717 006415           JSR     @IGTPA          ;GET PHYS PAGE #
14 02720 063077           HALT                    ;ASSIGNED CAN'T=377
15 02721 121000           MOV     1,0
16 02722 030734           LDA     2,AS,G1+2       ;A2=AVAILABLE MAP
17 02723 006764           JSR     @EX.I1          ;CMAPB MAKES 1K AVAIL
18 02724 101001           MOV     0,0,SKP
19 02725 063077           HALT                    ;BIT FOR THAT 1K HAD TO =0
20 02726 020405           LDA     0,EX.K1
21 02727 006406           JSR     @NTPA           ;PUT 377 IN ALLOCATION TBL
22 02730 016130           DSZ     @ALTBL          ;-1 # PAGES ASSIGNED
23 02731 000730           JMP     AS,XT-1         ;STILL PAGES LEFT+1 EXIT
24 02732 000730           JMP     AS,XT           ;0 MEM ALLOCATED DON'T SKIP
25 02733 000377 EX.K1: 377
26 02734 002363 IGTPA:  GETPA
27 02735 002572 NTPA: ENTPA
```

```
↓0061 MNMRT
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 02736 040421 GOSCR:  STA 0,GO.00      ;SAVE CALL PARAMS
08 02737 044421          STA 1,GO.01
09 02740 050421          STA 2,GO.02
10 02741 054421          STA 3,GO.S3
11 02742 022421          LDA 0,@GO.K2
12 02743 040421          STA 0,GO.LP
13 02744 022421          LDA 0,@GO.K3
14 02745 040422          STA 0,GO.LA
15 02746 002414          JMP @GO.S3       ;GO BACK TO TEST, NO MAP OPTION
```

```
↓0062 MNMRT
01                ;ERRRT - 2ND LEVEL ERROR RETURN
02
03 02747 054413 ERRRT:  STA 3,GO.S3
04 02750 002410          JMP @GO.01       ;RETURN TO TEST ERR
05
06                ;RETN2 - NORMAL 2ND LEVEL END OF TEST RET
07
08 02751 010411 RETN2:  ISZ GO.S3        ;+1 RETURN ADDRESS
09 02752 034412          LDA 3,GO.LP
10 02753 056410          STA 3,@GO.K2     ;RESTORE LOGICAL
11 02754 034413          LDA 3,GO.LA      ;PAGE AND ADDRS
12 02755 056410          STA 3,@GO.K3     ;OF ORIGINAL GSCRA
13 02756 002404          JMP @GO.S3       ;RETURN TO 1ST LEVEL TST
```

```
10063 MNMRT
01
02
03 02757 000000 GO.00:  0
04 02760 000000 GO.01:  0
05 02761 000000 GO.02:  0
06 02762 000000 GO.S3:  0
07 02763 002565 GO.K2:  CO.LP
08 02764 000000 GO.LP:  0
09 02765 002564 GO.K3:  CO.LA
10 02766 002561 ERRK2:  CO.S3
11 02767 000000 GO.LA:  0
12
13
```

```
10064 MNMRT
01                  ;SETLP - SET UP LOOP CALL HANDLER
02                  ;PERFORMS SAME FUNCTION AS SETUP IN NORMAL TSTS
03                  ;ENTERED VIA JSR @SETUL
04
05 02770 040437 SETLP:  STA 0,ST.S0
06 02771 044437         STA 1,ST.S1
07 02772 024437         LDA 1,ST.K1
08 02773 136414         SUB# 1,3,SZR
09 02774 000412         JMP STNMP           ;NOT AN LCALL
10 02775 022435         LDA 0,@ST.K2        ;GET LOG ADRS
11 02776 040107         STA 0,ST.LA         ;FOR LOOPL
12 02777 020434         LDA 1,@ST.K3        ;AND LOG PAGE
13 03000 044106         STA 1,ST.LP
14 03001 020433 SETXI:  LDA 0,ST.LK         ;-4
15 03002 040105         STA 0,ST.LC         ;FOR LOOP RPT COUNT
16 03003 020424         LDA 0,ST.S0
17 03004 024424         LDA 1,ST.S1
18 03005 001400         JMP 0,3
19                  ;LOOP SETUP WAS NOT VIA LCALL
20
21 03006 054107 STNMP:  STA 3,ST.LA
22 03007 000772         JMP SETXI
23
24                  ;LOOPL - PERFORMS SAME FUNCTION AS LOOP
25                  ;ENTERED VIA JSR @LLOOP
26
27 03010 010105 LOOPL:  ISZ ST.LC           ;SKIP IS FINI LOOP
28 03011 101001         MOV 0,0,SKP         ;LOOP BACK
29 03012 001400         JMP 0,3             ;CONTINUE ON
30 03013 040414         STA 0,ST.S0
31 03014 020415         LDA 0,ST.K1         ;CHK FOR
32 03015 116415         SUB# 0,3,SNR        ;SUPER CALL
33 03016 000403         JMP .+3             ;SUPER CALL
34 03017 020410         LDA 0,ST.S0         ;NOT LCALL
35 03020 002107         JMP @ST.LA          ;JUST CONTINUE
36 03021 020107         LDA 0,ST.LA         ;LOGICAL START LOOP
37 03022 042410         STA 0,@ST.K2
38 03023 020106         LDA 0,ST.LP         ;IN LOGICAL PAGE
39 03024 042407         STA 0,@ST.K3
40 03025 020402         LDA 0,ST.S0
41 03026 001400         JMP 0,3
42
43 03027 000000 ST.S0:  0
44 03030 000000 ST.S1:  0
45 03031 002550 ST.K1:  CO.EX+1
46 03032 002564 ST.K2:  CO.LA
47 03033 002565 ST.K3:  CO.LP
48 03034 177774 ST.LK:  -4.
```

```
↓0065 MNMRT
01                ;ERROH - ERROR HANDLER - PRINT ALL ERR INFO
02 03035 000010           8.
03                ;FIRST PRINT PRG# AND (AC'S)
04 03036 040514 ERROH:    STA 0,ER.S0
05 03037 020776           LDA 0,ERROH-1
06 03040 044513           STA 1,ER.S1
07 03041 050513           STA 2,ER.S2
08 03042 054513           STA 3,ER.S3
09 03043 030131           LDA 2,PSTRT
10 03044 011376           ISZ -2,2
11 03045 113000           ADD 0,2
12 03046 050426           STA 2,ERTIT
13 03047 101000           MOV 0,0
14 03050 010140           ISZ ERTOT       ;+1#ERROR CALLS
15 03051 101000           MOV 0,0
16 03052 020140           LDA 0,ERTOT
17 03053 024442           LDA 1,ER50.
18 03054 122414           SUB# 1,0,SZR    ;HALT AFTER FIRST 50 ERRORS
19 03055 000404           JMP .+4
20 03056 006116           JSR @LMESS
21 03057 005077           FTYTX
22 03060 004514           JSR KEY6W
23 03061 102400           SUB 0,0
24 03062 040137           STA 0,TIMSW     ;SO TIME TYPE WILL FOLLOW
25 03063 034127           LDA 3,SWREG
26 03064 101100           MOVL 0,0
27 03065 103102           AUDL 0,0,SZC    ;SW2=1 NO TYPE
28 03066 000431           JMP EREXI
29 03067 006116           JSR @LMESS
30 03070 004520           TXT.0
31 03071 024104           LDA 1,CURPR     ;GET PROG #
32 03072 006061           JSR @LZOCT      ;PRINT IT
33 03073 006116           JSR @LMESS
34 03074 004527 ERTIT:    TXT.1           ;PRINTS TEST TITL
35 03075 006116           JSR @LMESS
36 03076 004527           TXT.1
37 03077 024453           LDA 1,ER.S0     ;PRINT AC'S
38 03100 006117           JSR @LPOCT      ;AT ERROR CALL
39 03101 024452           LDA 1,ER.S1
40 03102 006117           JSR @LPOCT
41 03103 024451           LDA 1,ER.S2
42 03104 006117           JSR @LPOCT
```

```
↓0066 MNMRT
01                ;PRINT MEM ALLOCATION ASSIGNMENTS
02 03105 006116 ERMPP:    JSR @LMESS
03 03106 004533           TXT.2
04 03107 024143           LDA 1,SCRLO
05 03110 006117           JSR @LPOCT      ;PRINT SCRATCH LIMITS
06 03111 024144           LDA 1,SCRHI
07 03112 006117           JSR @LPOCT
08 03113 000404           JMP EREXI       ;NO MAP FORGET REST OF TYPE
09 03114 000000 ER.C1:    0
10 03115 000062 ER50.:    50.
```

```
!0067 MNMRT
01 03116 001000          186
02 03117 020127 EHEXI:   LDA 0,SWREG
03 03120 024776          LDA 1,EHEXI-1    ;GET 186
04 03121 123414          AND# 1,0,SZR     ;SKP=NOT ERROR WAIT
05 03122 004452          JSR KEY6W        ;WAIT FOR TTI INPUT
06 03123 103102          ADDL 0,0,SZC     ;C=0 IS ERROR RELEASE
07 03124 010431          ISZ ER.S3
08 03125 020425 EHXXT:   LDA 0,ER.S0
09 03126 024425          LDA 1,ER.S1
10 03127 030425          LDA 2,ER.S2
11 03130 034425          LDA 3,ER.S3
12 03131 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 03132 040420 EHROE:   STA 0,ER.S0
20 03133 044420          STA 1,ER.S1
21 03134 050420          STA 2,ER.S2
22 03135 054420          STA 3,ER.S3
23 03136 020127          LDA 0,SWREG
24 03137 101100          MOVL 0,0
25 03140 103102          ADDL 0,0,SZC     ;CHECK BIT 2 TO SEE IF WANT PRINT
26 03141 000756          JMP EHEXI
27 03142 006060          JSR @LCHLF
28 03143 024407          LDA 1,ER.S0
29 03144 006117          JSR @LPOCT
30 03145 024406          LDA 1,ER.S1
31 03146 006117          JSR @LPOCT
32 03147 024405          LDA 1,ER.S2
33 03150 006117          JSR @LPOCT
34 03151 000746          JMP EHEXI
35
36 03152 000000 ER.S0:   0
37 03153 000000 ER.S1:   0
38 03154 000000 ER.S2:   0
39 03155 000000 ER.S3:   0
40                       ;TEXT CALL ADRS OF TEXT IS IN AC0
41                       ;CALL MUST ONLY BE MADE WHILE IN FIRST LEVEL TEST
42 03156 040774 ERTXT:   STA 0,ER.S0
43 03157 040411          STA 0,ER.TP
44 03160 044773          STA 1,ER.S1
45 03161 050773          STA 2,ER.S2
46 03162 054773          STA 3,ER.S3
47 03163 020127          LDA 0,SWREG
48 03164 101100          MOVL 0,0
49 03165 103102          ADDL 0,0,SZC     ;SKP IS OK TO TYPE
50 03166 000737          JMP EHXXT        ;EXIT TYPE DELETED
51 03167 006116          JSR @LMESS
52 03170 000000 ER.TP:   0                ;TEXT ADRS STORED HERE
53 03171 000734          JMP EHXXT
54 03172 000000          0        ;TO SAVE AC0
55 03173 000003          3        ;FOR MASKO
```

```
!0068 MNMRT
01 03174 040432 KEY6W:   STA 0,KEY.0
02 03175 054432          STA 3,KEY.3
03 03176 006116          JSR @LMESS
04 03177 005106          KEY6T            ;TYPE WAIT MESS
05 03200 020142          LDA 0,LASTI      ;GET TTI INPUTTED CHARACTER
06 03201 101102          MOVL 0,0,SZC     ;WAIT FOR TYPE IN
07 03202 000406          JMP .+6
08 03203 063610          SKPDN TTI
09 03204 000774          JMP .-4
10 03205 101000          MOV 0,0
11 03206 060610          DIAC 0,TTI
12 03207 000402          JMP .+2
13 03210 101220          MOVZR 0,0
14 03211 040142          STA 0,LASTI      ;CLR BIT 0
15 03212 034412          LDA 3,TTOXS+2
16 03213 163400          AND 3,0          ;MASK OFF PARITY BIT
17 03214 040142          STA 0,LASTI
18 03215 034406          LDA 3,TTOXS+1    ;CK FOR ODT REQUEST
19 03216 116415          SUB# 0,3,SNR
20 03217 006406          JSR @IODT        ;GO TO EDITOR
21 03220 020406          LDA 0,KEY,0
22 03221 002406          JMP @KEY,3
23 03222 000000 TTOXS:   0
24 03223 000017          17
25 03224 000177          177
26 03225 003314 IODT:    ODT
27 03226 000000 KEY,0:   0
28 03227 000000 KEY,3:   0
```

```
10069 MNMRT
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                ;(S1.LA)-(AC1)+(AC2)=START LOOP IN RESIDENT
08                ;(AC0)-(AC1)+(AC2)"MAYBE"= ADRS OF ERROR
09
10 03230 040722 EPADR:  STA 0,ER.S0
11 03231 044722          STA 1,ER.S1
12 03232 050722          STA 2,ER.S2
13 03233 054722          STA 3,ER.S3
14 03234 006116          JSR @LMESS       ;PRINT HEADER
15 03235 004541          TXT.7
16 03236 024107          LDA 1,ST.LA
17 03237 006117          JSR @LPUCT       ;PRINT LOG. START OF LOOP
18 03240 030713          LDA 2,ER.S1
19 03241 146500          SUBL 2,1
20 03242 125220          MOVZR 1,1        ;GET RID OF CRY
21 03243 020711          LDA 0,ER.S2      ;PHYS STRT OF RESIDENT
22 03244 107000          ADD 0,1          ;CREATE PHYS STRT LOOP
23 03245 006117          JSR @LPUCT
24 03246 024704          LDA 1,ER.S0
25 03247 146500          SUBL 2,1         ;CREATE PHYSICAL ERR ADRS
26 03250 125220          MOVZR 1,1
27 03251 020703          LDA 0,ER.S2
28 03252 107000          ADD 0,1
29 03253 006117          JSR @LPUCT
30 03254 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 03255 040421 EPACS:  STA 0,EP.0
36 03256 044421          STA 1,EP.1
37 03257 050421          STA 2,EP.2
38 03260 054421          STA 3,EP.3
39 03261 006116          JSR @LMESS
40 03262 004560          TXT.8
41 03263 024110          LDA 1,RNAC0
42 03264 006117          JSR @LPUCT
43 03265 024111          LDA 1,RNAC1
44 03266 006117          JSR @LPUCT
45 03267 024112          LDA 1,RNAC2
46 03270 006117          JSR @LPUCT
47 03271 020405          LDA 0,EP.0
48 03272 024405          LDA 1,EP.1
49 03273 030405          LDA 2,EP.2
50 03274 034405          LDA 3,EP.3
51 03275 001400          JMP 0,3
52 03276 000000 EP.0: 0
53 03277 000000 EP.1: 0
54 03300 000000 EP.2: 0
55 03301 000000 EP.3: 0
56 03302 003117 EP.RT: EREXI
```

```
10070 MNMRT
01                ;ODT-OCTAL EDITOR
02 03303 000136 N136:    136
03 03304 000075 N75:     75
04
05 03305 000007 N7:      7
06 03306 020777 RUBOUT:  LDA     0,N7
07 03307 123420          ANDZ    1,0
08 03310 034536          LDA     3,N80
09 03311 163000          ADD     3,0              ;AC0=LAST DIGIT TYPED
10 03312 004543          JSR     SHIFT            ;ECHO AND ERASE THE CHARACTER
11                                                ;BEING RUBBED OUT
12 03313 000421          JMP     OPRIN-1
13
14 03314 044214 OUT:     STA     1,SAV1           ;SAVE THE ACCUMULATORS
15 03315 040213          STA     0,SAV0
16 03316 050215          STA     2,SAV2
17 03317 054216          STA     3,SAV3
18 03320 101200          MOVR    0,0              ;SAVE THE CARRY
19 03321 040217          STA     0,SAVCR
20 03322 102620          SUBZR   0,0
21 03323 040141          STA     0,EACTV          ;PREPARE TTI/TTO FLAG
22 03324 040220          STA     0,OP.EN
23 03325 030221          LDA     2,LOPNL
24 03326 006473 WAIT:    JSR     @CR.LF           ;TYPE CR,LF
25 03327 025000          LDA     1,0,2            ;GET CONTENTS OF LOC
26 03330 176440 WAITX:   SUBO    3,3              ;AC3=0
27 03331 054542          STA     3,TEMP
28 03332 176000          ADC     3,3              ;AC3     = 177777
29 03333 054536          STA     3,SIGN
30 03334 054536          STA     3,STRAC2
```

```
10071 MNMRT
01 03335 020142 OPRIN:  LDA     0,LASTI
02 03336 101102         MOVL    0,0,SZC         ;SKP IS NO INPUT YET
03 03337 000406         JMP     .+6
04 03340 063610         SKPDN   TTI             ;ALSO CHECK DONE
05 03341 000774         JMP     .-4             ;INCASE ION IS OFF
06 03342 101000         MOV     0,0
07 03343 060610         DIAC    0,TTI           ;GET CHAR IF DONE = 1
08 03344 000402         JMP     .+2
09 03345 101220         MOVZR   0,0
10 03346 040142         STA     0,LASTI         ;CLEAR BIT ZERO OF LASTI
11 03347 034454         LDA     3,N177
12 03350 163400         AND     3,0             ;REMOVE PARITY BIT
13 03351 116415         SUB#    0,3,SNR         ;SKIP IF AC0 IS NOT 177
14 03352 000734         JMP     RUBOUT          ;OTHERWISE GO TO RUBOUT
15 03353 034440         LDA     3,N67           ;AC3=67
16 03354 116452         SUB0#   0,3,SZC         ;IF THE ASCII VALUE IS HIGHER TH
17 03355 000404         JMP     WHERE           ;67 THEN GO TO "WHERE"
18 03356 034517         LDA     3,N57           ;AC3 = 57
19 03357 116442         SUB0    0,3,SZC         ;SKIP IF AC0 IS LESS THAN 57
20 03360 000423         JMP     DIGIT           ;OTHERWISE GO TO "DIGIT"
21 03361 010510 WHERE:  ISZ     SIGN            ;SKIP IF THE PREVIOUS SIGN WAS
22                                              ;A "+"
23 03362 124400         NEG     1,1             ;NEGATE IF IT WAS "-"
24 03363 034510         LDA     3,TEMP
25 03364 167000         ADD     3,1
26 03365 044506         STA     1,TEMP          ;STORE THE NEW VALUE
27 03366 034510         LDA     3,N12
28 03367 116415         SUB#    0,3,SNR         ;DON'T ECHO A LINE FEED
29 03370 000477         JMP     JPNXT
30 03371 034424         LDA     3,X17
31 03372 116415         SUB#    0,3,SNR         ;CONTROL "O" ?
32 03373 000733         JMP     WAIT
33 03374 034426         LDA     3,N15
34 03375 116415         SUB#    0,3,SNR         ;CHECK FOR A "CR"
35 03376 000471         JMP     JPNXT           ;CLOSE THE OPEN LOC.
36 03377 034415         LDA     3,N40
37 03400 116432         SUBZ#   0,3,SZC         ;SKP IF > 40
38 03401 000451         JMP     WHAT            ;"?"
39 03402 000422         JMP     CH.R
40 03403 010467 DIGIT:  ISZ     STRAC2
41 03404 125121         MOVZL   1,1,SKP
42 03405 126440         SUB0    1,1
43 03406 125120         MOVZL   1,1
44 03407 125120         MOVZL   1,1             ;SHIFT AC1 TO LEFT BY 3 PLACES
45 03410 160000         ADC     3,1             ;AC1=OCTAL WORD BEING INPUTED BY
46                                              ;THE OPERATOR
47 03411 004450         JSR     TPCHR           ;ECHO EVERY CHARACTER
48 03412 000723         JMP     OPRIN
49
```

```
10072 MNMRT
01 03413 000067 N67:    67
02 03414 000040 N40:    40
03 03415 000017 X17:    17
04 03416 000710 JXWT:   JMP WAIT
05 03417 000125 N125:   125
06 03420 000101 N101:   101
07 03421 003551 CR.LF:  CRLFX
08 03422 000015 N15:    15
09 03423 000177 N177:   177
10 03424 034457 CH.R:   LDA 3,N136
11 03425 116405         SUB     0,3,SNR         ;SKP IF NOT ^
12 03426 000456         JMP     NXTLOC          ;OPEN PREVIOUS LOC
13 03427 004432         JSR     TPCHR
14 03430 034454         LDA     3,N75           ;=
15 03431 116405         SUB     0,3,SNR
16 03432 002451         JMP     @IQUALS         ;PRINT CURRENT ARGUMENT(AC1)
17 03433 034444         LDA     3,N121          ;AC3=121
18 03434 162015         ADC#    3,0,SNR
19 03435 002444         JMP     @IRUN           ;IF IT WAS A "R" THEN START THE
                                                ;USERS PROGRAM
20
21 03436 116005         ADC     0,3,SNR
22 03437 002443         JMP     @IPRCD          ;IF IT IS A "P" THEN PROCEED
23 03440 034435 NOMAP:  LDA     3,N57           ;AC3 = 57
24 03441 162405         SUB     3,0,SNR         ;SKIP IF AC0 IS NOT "/"
25 03442 000455         JMP     OPNLOC
26 03443 145000         MOV     2,1
27 03444 101405         INC     0,0,SNR         ;SKIP IF IT IS NOT A "."
28 03445 000663         JMP     WAITX
29 03446 126460 N60:    SUBC    1,1             ;AC1=0 AND CARRY IS REESTORED
30                                              ;TO ITS PREVIOUS VALUE
31 03447 115644         INCUR   0,3,SZR         ;SKIP IF IT WAS A "-"
32 03450 161665         INCCR   3,0,SNR         ;SKIP IF IT WAS NOT A "+"
33 03451 000662         JMP     OPRIN-2
34 03452 020511 WHAT:   LDA     0,N77           ;BITS 8-15 OF AC0=77
35 03453 004406         JSR     TPCHR           ;TYPE IT"?"
36 03454 000652         JMP     WAIT
```

```
↓0073 MNMRT
01 03455 125220 SHIFT:  MOVZR   1,1
02 03456 125220         MOVZR   1,1
03 03457 125221         MOVZR   1,1,SKP
04 03460 020734         LDA     0,N40          ;PREPARE TO TYPE A SPACE
05 03461 061111 TPCHR:  DOAS    0,TTO          ;TYPE AC0
06 03462 063511         SKPBZ   TTO            ;WAIT FOR THE PRINTER
07 03463 000777         JMP     .-1
08 03464 101000         MOV     0,0
09 03465 060211         NIOC    TTO            ;CLR DNE FOR TTO
10 03466 001400         JMP     0,3            ;RETURN TO CALLER
11 03467 004415 JPNXT:  JMP     NXTLOC
12 03470 000000 TPRET:  0
13 03471 000000 SIGN:   0
14 03472 000000 STRAC2: 0
15 03473 000000 TEMP:   0
16 03474 000000 SUBRET: 0
17 03475 000057 N57: 57
18 03476 000012 N12: 12
19 03477 000121 N121:   121
20 03500 000115 N115:   115
21 03501 003565 IRUN:   RUN
22 03502 003567 IPRCD:  PRCD
23 03503 003604 IQUALS: EQUALS
```

```
↓0074 MNMRT
01 03504 010220 NXTLOC: ISZ     OP.EN
02 03505 000402         JMP     NTOPN          ;LOC WASN'T OPENED
03 03506 045000         STA     1,0,2          ;RESTORE THE OPEN LOCATION
04 03507 175232 NTOPN:  MOVZR#  3,3,SZC        ;SKIP IF IT WAS NOT A "CR"
05 03510 000706         JMP     JXMT
06 03511 145400         INC     2,1            ;IF IT IS A LINE FEED THEN ADD
07                                             ;1 TO AC2
08 03512 175004         MOV 3,3,SZR            ;SKP WAS A ^
09 03513 000403         JMP .+3
10 03514 166000         ADC 3,1
11 03515 166000         ADC 3,1
12 03516 004433         JSR     CRLFX          ;TYPE A CR, LF
13 03517 152000 OPNLOC: ADC     2,2            ;AC2 = 177777
14 03520 050220         STA     2,OP.EN        ;STORE THE FLAG FOR OPEN LOC.
15 03521 151220         MOVZR   2,2
16 03522 147400         AND     2,1            ;NEGLECT ADDRESS BIT 0
17 03523 044221         STA 1,LUPNL            ;SAVE ADRS LAST OPEN'D
18 03524 004405         JSR     POCTX-1        ;TYPE AC1
19 03525 025000         LDA     1,0,2
20 03526 004404         JSR     POCTX          ;TYPE THE OPEN LOCATION
21 03527 002401         JMP     @.+1
22 03530 003327         WAIT+1                 ;WAIT FOR THE OPERATOR INPUT
23
24 03531 131000         MOV     1,2
25 03532 054742 POCTX:  STA     3,SUBRET       ;SAVE THE RETURN ADDRESS
26 03533 050737         STA     2,STRAC2       ;SAVE AC2
27 03534 131000         MOV     1,2
28 03535 126621         SUBZR   1,1,SKP        ;AC1=100000
29 03536 132401         SUB     1,2,SKP
30 03537 020736         LDA     0,N57          ;AC0=57
31 03540 101400         INC     0,0
32 03541 132453 FRMDGT: SUBO#   1,2,SNC        ;IF AC2 IS LESS THAN AC1 THEN
33                                             ;GO TO PRINT THE DIGIT
34 03542 000774         JMP     .-4
35
36 03543 004712 PRIDGT: JSR     SHIFT
37 03544 125004         MOV     1,1,SZR
38 03545 000772         JMP     FRMDG1-2
39 03546 004712         JSR     TPCHR-1        ;TYPE SPACES
40 03547 030723         LDA     2,STRAC2       ;RESTORE AC2
41 03550 002724         JMP     @SUBRET
42
```

```
10075 MNMRT
01 03551 054723 CRLFX:  STA     3,SUBRET        ;SAVE THE RETURN ADDRESS
02 03552 020412         LDA     0,X15           ;AC0=15
03 03553 004706         JSR     TPCHR           ;TYPE A "CR"
04 03554 020722         LDA     0,N12           ;AC0=12
05 03555 004704         JSR     TPCHR           ;TYPE A LINE FEED
06 03556 020404         LDA     0,N100          ;BITS 8-15 OF AC0=100
07 03557 004702         JSR     TPCHR           ;TYPE A "@"
08 03560 002714         JMP     @SUBRET
09
10 03561 000040 K40:    40
11 03562 000100 N100:   100
12 03563 000077 N77:    77
13 03564 000015 X15:    15
14
15 03565 061077 RUN:    I.RST
16 03566 044216         STA     1,SAV3
17 03567 122400 PRCD:   SUB     0,0
18 03570 040141         STA     0,EACTV         ;RESET TTI/TTO FLAG
19 03571 020213         LDA     0,SAV0
20 03572 024214         LDA     1,SAV1
21 03573 030215         LDA     2,SAV2
22 03574 036407         LDA     3,@EQUALS-1
23 03575 177005         ADD     3,3,SNR         ;SKP=TTY DELETED
24 03576 056405         STA     3,@EQUALS-1     ;CLEAR INTR WAIT
25 03577 034217         LDA     3,SAVCR
26 03600 175100         MOVL    3,3
27 03601 034216         LDA     3,SAV3
28 03602 002216         JMP     @SAV3   ;RETURN TO PGM
29
30
31
32 03603 011213         TT.00+2         ;TTY TEST WAIT SWITCH
33 03604 044405 EQUALS: STA 1,.+5
34 03605 004725         JSR POCTX
35 03606 024403         LDA 1,.+3
36 03607 002401         JMP @.+1
37 03610 003330         WAITX
38 03611 000000         0
```

```
10076 MNMRT
01              ;FILENAME=TTYIO
02
03              ;TELETYPE INTERRUPT PACKAGE
04              ;AC1,AC2 SAVED
05              ;"MES?" PRINTS ASCII MESSAGES AS SPECIFIED BY ASSEMBLER
06              ;"CLF?" PRINTS A CARRIAGE RETURN
07              ;"POC?" PRINTS C(1) IN OCTAL
08              ;"ZOC?" PRINTS C(1) IN OCTAL, LEADING ZEROS SUPPRESSED
09              ;"POD?" PRINTS C(1) IN DECIMAL, LEADING ZEROS SUPPRESSED,
10              ;THE ABOVE THREE ARE FOLLOWED BY A TAB UNLESS LOCATION PTB? IS
11              ;ALTERED IN WHICH CASE CONTENTS OF PTB? WILL BE PRINTED AFTER
12              ;THE NUMBER.
13              ;"TIO?" ACCEPTS OCTAL, AND
14              ;"TID?" ACCEPTS DECIMAL SINGLE PRECISION SIGNED INTEGERS
15              ;INTO AC1 FROM THE TTI.  LEADING NULLS, TABS,
16              ;AND SPACES ARE IGNORED.  A 16 BIT UNSIGNED INTEGER IS
17              ;FORMED, THEN NEGATED IF A MINUS SIGN IS TYPED.
18              ;EXIT AT CALL+1 IF INPUT ERROR WITH AC0=BAD CHARACTER.
19              ; (NOT A LEGAL DIGIT OR TERMINATING CHARACTER)
20              ;EXIT AT CALL+2 UPON TERMINATING CHARACTER
21              ; WITH AC0=0, 0, 40, 12, 55
22              ; FOR NULL, TAB, SPACE, CARRIAGE RETURN, COMMA
23              ;THE ABOVE WAIT FOR TTO DONE, THEN CLEAR TTO.
24              ;"CHC?T" PRINTS ASCII CHARACTOR IN C(0)R; C(0)L MUST BE 0.
25              ;EXITS CALL +2 IF C(0)R=0; SIMULATES TAB
26              ;"TYP?" PRINTS C(0)R, EXITS AT CALL+1.  REPLACE "TYP?" WITH
27              ;INTERRUPT TYP? IF DESIRED.
28              ;"TPS?" PRINTS A SPACE AND EXITS AT CALL+1 WITH AC0 = 40
29
30 03612 054524 MES?:   STA     3,MES?R         ;PRINT A TEXT MESSAGE
31 03613 044465         STA     1,PAC?1
32 03614 050465         STA     2,PAC?2
33 03615 010521         ISZ     MES?R
34 03616 031400         LDA     2,0,3           ;C(2) POINTS TO MESSAGE
35 03617 024463         LDA     1,PJ777         ;A 8 BIT MASK
36 03620 021000         LDA     0,0,2           ;C(2)=DATA WORD
37 03621 125112         MOVL#   1,1,SZC
38 03622 123701         ANDS    1,0,SKP
39 03623 123401         AND     1,0,SKP         ;C(0)=DATA CHARACTOR RIGHT
40 03624 151400         INC     2,2             ;INC TO NEXT WORD
41 03625 124000         COM     1,1             ;FLIP MASK
42 03626 004455         JSR     CHC?T           ;PRINT
43 03627 000771         JMP     MES?+6          ;ANOTHER
44 03630 000402         JMP     .+2
45 03631 004452 PLS?T:  JSR     CHC?T
46 03632 024446 PEX?T:  LDA     1,PAC?1
47 03633 030446         LDA     2,PAC?2
48 03634 063511         SKPBZ   TTO
49 03635 000777         JMP     .-1
50 03636 060211         NIOC    TTO
51 03637 002477         JMP     @MES?R          ;LAST
```

```
10077 MNMRT
03640 102401 ZOC?:   SUB     0,0,SKP
03641 020563 POC?:   LDA     0,PC6?0
03642 050437         STA     2,PAC?2
03643 030565         LDA     2,OCT?AB          ;PRINT C(1) IN OCTAL
03644 000404         JMP     .+4
03645 050434 PDE?:   STA     2,PAC?2
03646 030572         LDA     2,DEC?TB          ;PRINT C(1) IN DECIMAL
03647 102400         SUB     0,0
03650 054466         STA     3,MES?R           ;BOTH ENTRYS PRINT NUMBER
03651 044427         STA     1,PAC?1
03652 040425         STA     0,ZSU?P           ;THEN TAB TO NEXT POSITION
03653 050401         STA     2,.+1
03654 000000 DEC?OC: 0                         ;A"LDA 2,TABLE" INSTRUCTION
03655 010777         ISZ     .-1
03656 020571         LDA     0,PTB?
03657 151005         MOV     2,2,SNR           ;IF TABLE ENTRY=0
03660 000751         JMP     PLS?T             ;EXIT WITH A SPACE/TAB
03661 034416         LDA     3,ZSU?P           ;ZEROS SUPPRESS STUF
03662 102400         SUB     0,0
03663 146452 DEO?T:  SUBO#   2,1,SZC
03664 000405         JMP     DEC?P
03665 146400         SUB     2,1               ;FORM THE DIGIT
03666 034536         LDA     3,PC6?0
03667 101400         INC     0,0
03670 000773         JMP     DEO?T
03671 151235 DEC?P:  MOVZR#  2,2,SNR
03672 034532         LDA     3,PC6?0
03673 054404         STA     3,ZSU?P           ;C(0)=DIGIT
03674 163004         ADD     3,0,SZR           ;MAKE ASCII
03675 004406         JSR     CHC?T             ;PRINT
03676 000756         JMP     DEC?OC            ;GET NEXT DIGIT
03677 000000 ZSU?P:  0
03700 000000 PAC?1:  0
03701 000000 PAC?2:  0
03702 000377 P377?:  377
03703 054432 CHC?T:  STA     3,CHR?T           ;PRINT C(0) RIGHT
03704 101315         MOVS#   0,0,SNR           ;RETURN +2 IF NULL
03705 001401         JMP     1,3
03706 034426         LDA     3,PC1?1           ;AC3 = 11
03707 116415         SUB#    0,3,SNR           ;SKIP IF A TAB IS NOT TO
                                               ;BE SIMULATED
03710 000403         JMP     CHA?3
03711 004541         JSR     TYP?              ;PRINT IT
03712 002423         JMP     @CHR?T            ;EXIT
03713 004536 CHA?3:  JSR     TPS?              ;PRINT A SPACE
03714 020417         LDA     0,CHR?Z
03715 034415         LDA     3,PC7?            ;AC3 = 7
03716 163404         AND     3,0,SZR
03717 000774         JMP     CHA?3             ;SIMULATE A TABE WITH 1
                                               ;TO 7 SPACES
03720 040413         STA     0,CHR?Z
03721 002414         JMP     @CHR?T
```

```
10078 MNMRT
03722 054414 CLF?:   STA     3,MES?R           ;SAVE RETURN
03723 044755         STA     1,PAC?1
03724 050755         STA     2,PAC?2
03725 020404         LDA     0,XJ15
03726 004755         JSR     CHC?T             ;PRINT CARRIAGE AND LF
03727 020515         LDA     0,PC1?2
03730 000701         JMP     PLS?T

03731 000015 XJ15:   15
03732 000007 PC7?:   7
03733 000000 CHR?Z:  0
03734 000011 PC1?1:  11
03735 000000 CHR?T:  0
03736 000000 MES?R:  0
03737 020505 TIC?:   LDA     0,PC1?2
03740 004512         JSR     TYP?
03741 010775 TIX?:   ISZ     MES?R
03742 024736 TIR?:   LDA     1,PAC?1
03743 034734         LDA     3,ZSU?P
03744 175102         MOVL    3,3,SZC
03745 124400         NEG     1,1
03746 000665         JMP     PEX?T+1

03747 102121 TIO?:   ADC2L   0,0,SKP           ;OCTAL ENTRY
03750 102440 TID?:   SUBO    0,0               ;DECIMAL ENTRY
03751 054705         STA     3,MES?R
03752 050727         STA     2,PAC?2           ;AC2 IS SAVED
03753 030471         LDA     2,PC1?2
03754 113000         ADD     0,2
03755 102440         SUBO    0,0
03756 040721         STA     0,ZSU?P           ;MINUS SIGN AND LEADING SPACES
                                               ;FLAG
03757 034720 TIS?:   LDA     3,ZSU?P
03760 175004         MOV     3,3,SZR
03761 000760         JMP     TIX?
03762 054716 TIW?:   STA     3,PAC?1
03763 063610         SKPDN   TTI
03764 000777         JMP     .-1
03765 060610         DIAC    0,TTI
03766 004715         JSR     CHC?T
03767 034436         LDA     3,PC42?
03770 116414         SUB#    0,3,SZR
03771 101015         MOV#    0,0,SNR
03772 000765         JMP     TIS?              ;SPACE, TAB, OR NULL
03773 024434         LDA     1,TIN?2
03774 106015         ADC#    0,1,SNR           ;COMMA
03775 000744         JMP     TIX?              ;OR
03776 106424         SUBZ    0,1,SZR           ;MINUS
03777 000405         JMP     TIM?              ;IF NEITHER THEN GO TO TIM?
04000 034677         LDA     3,ZSU?P
04001 177200         ADDR    3,3               ;COMPLEMENT SIGN
04002 054675         STA     3,ZSU?P
04003 000760         JMP     TIW?+1
04004 136415 TIM?:   SUB#    1,3,SNR           ;IS IT A CARRIAGE RETURN?
04005 000732         JMP     TIC?              ;IF CR THEN GO TO TIC?
```

```
10079 MNMRT
01 04005 024420 TIN?:   LDA     1,TIN?1
02 04006 107022         ADDZ    0,1,SZC         ;SKIP IF NOT A DIGIT
03 04010 146513         SUBL#   2,1,SNC         ;SKIP IF DIGIT
04 04011 000731         JMP     TIR?
05 04012 010665         ISZ     ZSU?P           ;OUT OF LEADING SPACES
06 04013 020665         LDA     0,PAC?1
07 04014 101120         MOVZL   0,0
08 04015 115120         MOVZL   0,3
09 04016 175120         MOVZL   3,3
10 04017 137000         ADD     1,3             ;3 OLD PAC?1'S + NEW DIGIT
11 04020 145220         MOVZR   2,1
12 04021 125232         MOVZR#  1,1,SZC         ;SKIP IF OCTAL MODE
13 04022 117000         ADD     0,3             ;ADD 2 OLD PAC?1'S
14 04023 000737         JMP     TIN?
15 04024 000060 PC6?0:  60
16 04025 000040 PC4?0:  40
17 04026 177720 TIN?1:  -50
18 04027 000055 TIN?2:  55
19 04030 030555 OCT?AB: LDA     2,.+1+.-OCT?OC
20 04031 100000         100000
21 04032 010000         10000
22 04033 001000         1000
23 04034 000100 C100:   100
24 04035 000010         10
25 04036 000001         1
26 04037 000000         0
27 04040 030565 DEC?TB: LDA     2,.+1+.-DEC?OC
28       000012 .RDX 10
29 04041 023420         10000
30 04042 001750         1000
31 04043 000144         100
32 04044 000012 PC12:   10
33 04045 000001         1
34 04046 000000         0
35       000010 .RDX    8
36
37 04047 000011 PT8?:   11
38
```

```
10080 MNMRT
01 04050 011213 P?TTO:  TT.00+2
02 04051 020754 TPS?:   LDA     0,PC4?0         ;PREPARE TO PRINT A SPACE
03 04052 054452 TYP?:   STA     3,TYP?R         ;TYPE THE RIGHT BYTE OF AC0
04 04053 034447         LDA     3,INT?          ;IF IT IS HERE DUE TO SWPAK
05 04054 175004         MOV     3,3,SZR         ;TTO OUTPUTS WILL BE ENABLED.
06 04055 034127         LDA     3,SWREG         ;READ THE SWITCHES
07 04056 040447         STA     0,P.MS0         ;SAVE AC0
08 04057 177100         ADDL    3,3
09 04060 175112         MOVL#   3,3,SZC         ;SKIP IF TYPEOUTS ARE NOT
10 04061 000417         JMP     PLP?T           ;SUPPRESSED
11 04062 176620         SUBZR   3,3             ;BIT 0=1
12 04063 163000         ADD     3,0
13 04064 040141         STA     0,EACTV         ;SET UP EACTV FOR TTO
14 04065 020440         LDA     0,P.MS0         ;GET SAVED CHARACTER
15 04066 063511 TPT?Y:  SKPBZ   TTO
16 04067 000777         JMP     .-1
17 04070 000401         JMP     .+1
18 04071 063511         SKPBZ   TTO
19 04072 000775         JMP     .-3
20 04073 061111         DOAS    0,TTO           ;OUTPUT TO TTO
21 04074 063511         SKPBZ   TTO
22 04075 000777         JMP     .-1
23 04076 101000         MOV     0,0
24 04077 060211         NIOC    TTO
25 04100 020425 PLP?T:  LDA     0,P.MS0         ;GET CHARACTER
26 04101 034420         LDA     3,P177
27 04102 163000         AND     3,0             ;REMOVE PARITY  BIT FOR LPT
28 04103 176400 TPR?T:  SUB     3,3
29 04104 054141         STA     3,EACTV         ;CLEAR EACTV SWITCH
30 04105 036743         LDA     3,@P?TTO        ;CK TTO INTR WAIT SWITCH
31 04106 177005         ADD     3,3,SNR         ;SKIP IS TTO TST DELETED
32 04107 056741         STA     3,@P?TTO        ;CLEAR INTR WAIT SWITCH
33 04110 034411         LDA     3,P177
34 04111 116043         ADCO    0,3,SNC         ;SKIP IF IT WAS A RUBOUT
35 04112 034713         LDA     3,PC4?0         ;AC3 = 40
36 04113 162432         SUBZ#   3,0,SZC         ;SKIP FOR NON PRINTING
37 04114 010617         ISZ     CHR?Z           ;CHARACTERS
38 04115 034406         LDA     3,PC15          ;AC3 = 15
39 04116 116445         SUBO    0,3,SNR         ;SKIP IF IT WAS NOT A "CR"
40 04117 054614         STA     3,CHR?Z         ;CLEAR HORZ POS
41 04120 002404         JMP     @TYP?R
42 04121 000177 P177:   177
43 04122 177777 INT?:   -1
44 04123 000015 PC15:   15
45 04124 000000 TYP?R:  0
46 04125 000000 P.MS0:  0
47 04126 040000 P.MB1:  1B1
48 04127 002000 P.MB5:  1B5
```

```
↓0081 MNMRT
01
02                ;FILENAME= SWRPACK
03
04                ;THIS PACKAGE IS USED TO CHANGE THE SETTINGS OF LOCATION
05                ;"SWREG" OF PAGE 0. THE PROGRAM CONTROL SHOULD ENTER "INP?"
06                ;WITH AC3 HAVING THE RETURN ADDRESS. THE CHARACTER INPUTED
07                ;BY THE OPERATOR IS ECHOED AFTER A "CR". IF THE COMMAND IS
08                ;NOT A LEGAL ONE THEN THE CONTROL IS RETURNED WITHOUT DOING
09                ;ANY THING, OTHERWISE ONE OF THE FOLLOWING COMMANDS IS
10                ;EXECUTED:
11                ;KEYS 1-9 AND A-F ARE USED TO COMPLEMENT THE CURRENT VALUE
12                ;OF BITS 1-15 OF "SWREG". IF ONE OF THESE KEYS IS HIT THE
13                ;CORRESPONDING BIT OF "SWREG" IS COMPLEMENTED AND THE CONTROL
14                ;IS RETURNED TO THE STATE PROGRAM HAD BEFORE HITTING THE KEY
15                ;TYPING OF A "0" WILL LOCK THE PROGRAM IN A SWITCH MODIFICATION
16                ;MODE IN WHICH CASE MORE THAN ONE BITS CAN BE CHANGED BEFORE
17                ;THE CONTROL IS ALLOWED TO RETURN TO THE MAIN PROGRAM. HITTING
18                ;THE "CR" KEY WILL UNLOCK THE PROGRAM FROM THIS MODE.
19                ;"AD" THIS COMMAND GIVEN AT ANY TIME WILL RESET THE "SWREG"
20                ;TO DEFAULT MODE (ALL ZEROS) AND RESTART THE PROGRAM AT ADD.
21                ;STORED IN LOCATION "INS?"
22                ;"AR" THIS COMMANND GIVEN AT ANY TIME  WILL RESTART THE PROG.
23                ;AT ADDRESS STORED IN LOCATION "INS?"
24                ;"M" THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE  CURRENT
25                ;OPERATING MODES.
26                ;BEFORE THE CONTROL IS RETURNED TO THE MAIN PROGRAM BIT 0 WILL
27                ;BE SET IF ANY OF THE OTHER BITS OF "SWREG" IS SET, OTHERWISE
28                ;IT WILL BE CLEARED
29
30                ;THIS PACKAGE EXITS WITH C(AC3) = CHARACTER TYPED IN (PARITY
31                ;STRIPPED).
32
33 04130 000136 IN1?3G: 136
34 04131 000104 IN1?D4: 104
35 04132 000122 IN1?2?: 122
36 04133 000033 INC?33: 33
37 04134 000000 INL?K:  0
38 04135 000000 INR?T:  0
39 04136 000000 INS?0:  0
40 04137 000000 INS?1:  0
41 04140 000000 INS?2:  0
42 04141 000000 INS?3:  0
43 04142 000000 INS?C:  0
44 04143 000202 INS?:   STRT2
45 04144 000000 SVP?TB: 0
46 04145 004047 IPTB?:  PTB?
```

```
↓0082 MNMRT
01 04146 054767 INP?:   STA     3,INR?T         ;SAVE THE RETURN ADDRESS
02 04147 044767         STA     0,INS?0         ;SAVE AC0
03 04150 044767         STA     1,INS?1         ;AC1
04 04151 050767         STA     2,INS?2         ;AND AC2
05 04152 175200         MOVR    3,3             ;
06 04153 054767         STA     3,INS?C         ;SAVE CARRY
07 04154 176400         SUB     3,3
08 04155 054745         STA     3,INT?
09 04156 040756         STA     0,INL?K         ;"INL?K" IS NOT -1
10 04157 020142 IN0?:   LDA     0,LASTI         ;READ THE INPUT
11 04160 034741         LDA     3,P177?         ;AC3 = 177
12 04161 163400         AND     3,0             ;GET RID OF THE PARITY BIT
13 04162 024741         LDA     1,PC17S         ;AC1 = 15
14 04163 106415         SUB#    0,1,SNR         ;SKIP IF THE CHARACTER TYPED
15                                              ;WAS NOT "CR"
16 04164 000531         JMP     INR?
17 04165 044754         STA     0,INS?3         ;SAVE CHARACTER
18 04166 024646         LDA     1,C100?         ;AC1 = 100
19 04167 034655         LDA     3,PC1?2         ;AC3 = 12
20 04170 110414         SUB#    0,3,SZR         ;SKIP IF IT IS A LINE FEED
21 04171 034742         LDA     3,INC?33        ;AC3 = 33
22 04172 162452         SUBO#   3,0,SZC         ;SKIP IF AC0 IS EQUAL OR MORE
                                                ;THAN AC3
23
24 04173 000504         JMP     IN5?
25 04174 004656 IN1?:   JSR     TYP?            ;ECHO THE CHARACTER
26 04175 034627         LDA     3,PC6?0         ;AC3 = 60
27 04176 152620         SUBZR   2,2             ;AC2 = 100000
28 04177 116405         SUB     0,3,SNR         ;SKIP IF THE DIGIT TYPED WAS
29                                              ;NOT 0
30 04200 000447         JMP     IN3?
31
32 04201 151221 IN2?:   MOVZR   2,2,SKP         ;SHIFT AC2 TO RIGHT
33 04202 126520         SUBZL   1,1             ;AC1 = 1
34 04203 175405         INC     3,3,SNR
35 04204 000445         JMP     IN3?+2
36 04205 147415         ANDO#   2,1,SNR         ;STAY IN LOOP UNTIL ALL BITS
37                                              ;OF SWREG ARE CHECKED
38 04206 000773         JMP     IN2?
39 04207 106400         SUB     0,1             ;WHEN THE CONTROL COMES HERE
40                                              ;FOR THE FIRST TIME AC1 = 100
41 04210 135000         MOV     1,3
42 04211 151225         MOVZR   2,2,SNR
43 04212 000444         JMP     IN4?
44 04213 024710         LDA     1,PC17S         ;AC1 = 15
45 04214 167004         AUD     3,1,SZR         ;SKIP IF THE COMMAND WAS "M"
46 04215 000765         JMP     IN2?+1
47
```

```
10083 MNMNT
01 04216 036727 INM?:   LDA     3,@IPTB?        ;SAVE PTB?
02 04217 054725         STA     3,SVP?TB
03 04220 034456         LDA     3,PC,40         ;PREPARE TO PRINT A SPACE
04                                              ;AFTER EACH NUMBER
05 04221 056724         STA     3,@IPTB?
06 04222 000113         JSR     @ICLF?          ;TYPE A "CR" AND "LF"
07 04223 000114         JSR     @IPDE?          ;PRINT THE CONTENTS OF AC1
08 04224 034620         LDA     3,PC172         ;AC3 = 12
09 04225 125400         INC     1,1
10 04226 166452         SUB0#   3,1,SZC         ;SKIP IF AC1 IS GREATER OR EQUAL
11                                              ;TO AC3
12 04227 000445         JSR     @ITPS?          ;TYPE A SPACE
13 04230 101220         MOVZR   0,0             ;AC0 = 20
14 04231 122414         SUB#    1,0,SZR         ;SKIP AFTER TYPING # 15
15 04232 000771         JMP     INM?+5
16 04233 000113         JSR     @ICLF?
17 04234 030127         LDA     2,SWREG
18 04235 151140         MOVOL   2,2             ;AC2 HAD SWITCH SETTINGS
19 04236 126500         SUBCL   1,1             ;BRING THE CARRY BIT IN AC1
20 04237 000115         JSR     @IZUC?          ;TYPE THE CONTENTS OF AC1
21 04240 000434         JSR     @ITPS?          ;TYPE A SPACE
22 04241 151124         MOVZL   2,2,SZR         ;SKIP AFTER TYPING ALL THE 16
23                                              ;BITS
24 04242 000774         JMP     .-4
25 04243 000113         JSR     @ICLF?
26 04244 034700         LDA     3,SVP?TB
27 04245 056700         STA     3,@IPTB?        ;RESTORE PTB?
28 04246 000410         JMP     IN4?
29
30 04247 176000 IN3?:   ADC     3,3             ;AC3 = -1
31 04250 054664         STA     3,INL?K         ;LOCK IN SWITCH INPUT MODE
32 04251 024127         LDA     1,SWREG         ;READ THE CURRRENT VALUE OF
33                                              ;"SWREG"
34 04252 133414         AND#    1,2,SZR         ;TAKE XOR OF AC1 AND AC2
35 04253 146401         SUB     2,1,SKP
36 04254 147000         ADD     2,1
37 04255 044127         STA     1,SWREG         ;SAVE THE NEW VALUE OF "SWREG"
38 04256 010656 IN4?:   ISZ     INL?K           ;SKIP IF THE PROGRAM IS LOCKED
39                                              ;IN SWITCH INPUT MODE
40 04257 000436         JMP     INR?
41 04260 014654         DSZ     INL?K           ;NEVER SKIP
42 04261 020142         LDA     0,LASTI
43 04262 101102         MOVL    0,0,SZC
44 04263 000406         JMP     .+6
45 04264 063610         SKPDN   TTI             ;WAIT FOR OPERATOR INPUT
46 04265 000774         JMP     .-4
47 04266 101000         MOV     0,0
48 04267 060610         DIAC    0,TTI
49 04270 000402         JMP     .+2
50 04271 101220         MOVZR   0,0
51 04272 040142         STA     0,LASTI
52 04273 000665         JMP     IN0?+1
53 04274 004051 ITPS?:  TPS?
54 04275 004052 ITYP?:  TYP?
55 04276 000040 PC,40:  40
```

```
10084 MNMNT
01 04277 107000 IN5?:   ADC     0,1             ;AC1 = 100+ ASCII VALUE OF
                                                ;CONTROL CHARACTER
02
03 04300 020630         LDA     0,IN1?36        ;AC0 = 136
04 04301 000774         JSR     @ITYP?          ;TYPE ^
05 04302 121000         MOV     1,0
06 04303 000772         JSR     @ITYP?
07 04304 034626         LDA     3,IN1?22        ;AC3 = 122
08 04305 116405         SUB     0,3,SNR         ;SKIP IF IT IS NOT ^R
09 04306 000405         JMP     IN6?
10 04307 034622         LDA     3,IN1?04        ;AC3 = 104
11 04310 116404         SUB     0,3,SZR         ;SKIP IF IT WAS A ^D
12 04311 000745         JMP     IN4?
13 04312 054127         STA     3,SWREG         ;SET SWITCHHES TO DEFAULT
                                                ;MODE
14
15 04313 034630 IN6?:   LDA     3,INS?          ;AC3 = ADDRESS OF THE LOCATION
                                                ;WHERE THE PROGRAM WILL START
16
17 04314 054621         STA     3,INR?T
18 04315 010605 INR?:   ISZ     INT?
19 04316 030127         LDA     2,SWREG
20 04317 176220         ADCZR   3,3             ;AC3 = 77777
21 04320 173404         AND     3,2,SZR         ;SKIP IF THE SWITCHES ARE SET
                                                ;TO ALL ZERO'S
22
23 04321 172000         ADC     3,2
24 04322 050127         STA     2,SWREG
25 04323 020617         LDA     0,INS?C         ;RESTORE CARRY
26 04324 101100         MOVL    0,0             ;
27 04325 020611         LDA     0,INS?0
28 04326 024611         LDA     1,INS?1         ;RESTORE THE ACCUMULATORS
29 04327 030611         LDA     2,INS?2         ;RETURN WITH C(AC3)=CHAR INPUT
30 04330 034611         LDA     3,INS?3         ;RETURN
31 04331 002604         JMP     @INR?T
```

```
10085 MNMRT
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                LINTR:
13 04332 075401           PSH 3             ;SAVE AC3
14 04333 071401           PSH 2             ;AC2
15 04334 065401           PSH 1             ;AC1
16 04335 061401           PSH 0             ;AC0
17 04336 020000           LDA 0,0
18 04337 101102           MOVL 0,0,SZC      ;CARRY
19 04340 063077           HALT              ;BIT 0 OF 0=1?
20 04341 061401           PSH 0             ;PC+2+CRY IN BIT 15
21 04342 020440           LDA 0,MSKRG
22 04343 061401           PSH 0             ;SAVE OLD MSKO
23 04344 030521           LDA 2,LK300
24 04345 034434           LDA 3,LMSKS
25 04346 061477           INTA 0
26 04347 100015           COM# 0,0,SNR              ;CHECK FOR POWERFAIL
27 04350 002120           JMP @PFAIL                ;INTA GAVE -1 MUST BE POWER FAIL
28 04351 113000           ADD 0,2
29 04352 117000           ADD 0,3
30 04353 025400           LDA 1,0,3
31 04354 035000           LDA 3,0,2
32 04355 044425           STA 1,MSKRG
33 04356 030425           LDA 2,K4
34 04357 112415           SUB# 0,2,SNR      ;WAS IT A DEV #4 INTR?
35 04360 001403           JMP 0,3           ;YEP, SERVICE IT
36 04361 063077           MSKO 1
37 04362 063177           INTEN
38 04363 005400           JSR 0,3           ;GO TO INTR SERV
```

```
10086 MNMRT
01                ;LINTD - LINKER INTERRUPT DISMISS
02                ;RETURN ADDRESS WAS GIVEN TO DEVICE
03                ;SERVICE ROUTINES VIA THE JSR 03
04
05 04364 063277 LINTD:    INTDS
06 04365 065601           POP 1     ;RETRIEVE MASK
07 04366 044414           STA 1,MSKRG
08 04367 063077           MSKO 1
09 04370 061601           POP 0     ;GET CRY+2+PC
10 04371 101220           MOVZR 0,0         ;RESTORE CRY
11 04372 040000           STA 0,0
12 04373 061601           POP 0
13 04374 065601           POP 1
14 04375 071601           POP 2
15 04376 075601           POP 3
16 04377 063177           INTEN
17 04400 002000           JMP @0    ;GETS TO LOGICAL 0
18 04401 000000 LMSKS: 0
19 04402 000000 MSKRG: 0
20 04403 000004 K4:  4
21                ;EINTP, - ENTER INTERRUPT SERVICE PARAMETERS
22                ;(AC0)=DEV# JSR @EINTS
23                ;(AC1)=MSKO
24                ;(AC2)=ADDRESS OF DEV INTR SERV
25
26 04404 054410 EINTP:    STA 3,EI.S3
27 04405 034400           LDA 3,LK300
28 04406 117000           ADD 0,3
29 04407 051400           STA 2,0,3
30 04410 034771           LDA 3,LMSKS
31 04411 117000           ADD 0,3
32 04412 045400           STA 1,0,3
33 04413 002401           JMP @EI.S3
34
35 04414 000000 EI.S3:    0
```

```
;0087 MNMRT
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 04415 020460 LCINT:  LDA 0,LILLI
06 04416 030447         LDA 2,LK300
07 04417 041000         STA 0,0,2       ;FILL SERVICE
08 04420 151400         INC 2,2         ;VECTORS WITH
09 04421 145300         MOVS 2,1        ;ADRS ILLEGAL INTR
10 04422 125224         MOVZR 1,1,SZR
11 04423 000774         JMP LCINT+2
12 04424 071001         MTSP 2  ;STACK POINTER TO 400
13 04425 070001         MTFP 2  ;FRAME POINTER TO 400
14 04426 052441         STA 2,@LC.K1+1
15 04427 145220         MOVZR 2,1
16 04430 131220         MOVZR 1,2       ;AC2=100,AC1=200
17 04431 022435         LDA 0,@LC.K1    ;RESERVE 100 WORDS
18 04432 113000         ADD 0,2         ;ABOVE MEM ALLOC. TBLS
19 04433 052433         STA 2,@LC.K1    ;100 FOR MSKO
20 04434 125220         MOVZR 1,1       ;AC1=100
21 04435 102000         ADC 0,0
22 04436 041000         STA 0,0,2
23 04437 113000         ADD 0,2
24 04440 107004         ADD 0,1,SZR
25 04441 000775         JMP .-3
26 04442 151400         INC 2,2
27 04443 050736         STA 2,LMSKS
28 04444 020420         LDA 0,LC.K2
29 04445 040001         STA 0,1
30 04446 102400         SUB 0,0
31 04447 040733         STA 0,MSKRG     ;STRT INT MSK =0
32 04450 042420         STA 0,@LC.K1+2
33 04451 020420         LDA 0,LC.K1+3
34 04452 040003         STA 0,3         ;INIT FOR STACK INTA'S
```

```
;0088 MNMRT
01 04453 020417         LDA 0,LC.K8     ;ADDR OF TTI INTR HANDLER
02 04454 040310         STA 0,300+TTI   ;PUT IN INTR TBL
03 04455 020416         LDA 0,LC.K9     ;ADDR OF TTO INTR HANDLER
04 04456 040311         STA 0,300+TTO   ;PUT IN INTR TBL
05 04457 020120         LDA 0,PFAIL     ;PWR FAIL RTN ADDR
06 04460 040300         STA 0,300       ;PUT IN INTR TBL
07 04461 020413         LDA 0,LC.CN     ;CONSULE INTR HANDLER
08 04462 040304         STA 0,300+4
09 04463 001400         JMP 0,3
10 04464 004332 LC.K2: LINTR
11 04465 000300 LK300:  300
12 04466 001372 LC.K1:  LSETB
13 04467 002156         STADR
14 04470 002160         STKPG
15 04471 002200         STKIN
16 04472 011257 LC.K8: TT.TI
17 04473 011324 LC.K9: TT.TO
18 04474 004505 LC.CN:  MNCON
```

```
;0089 MNMRT
01 04475 004476 LILLI:  LILLI+1
02 04476 040135         STA 0,UDEVI
03 04477 024402         LDA 1,.+2
04 04500 123001         ADD 1,0,SKP
05 04501 060200         NIOC 0
06 04502 040401         STA 0,.+1
07 04503 060200         NIOC 0          ;CHANGED TO DEV#
08 04504 001400         JMP 0,3
09
10 04505 065601 MNCON:  POP 1
11 04506 044674         STA 1,MSKRG
12 04507 061601         POP 0
13 04510 101220         MOVZR 0,0
14 04511 040000         STA 0,0
15 04512 061601         POP 0
16 04513 065601         POP 1
17 04514 071601         POP 2
18 04515 075601         POP 3           ;UNSAVE AC'S
19 04516 002401         JMP @.+1
20 04517 177776         @77776
```

```
;0090 MNMRT
01 04520 005215 TXT.0:  .TXTE (<15><12>
02 04521 151120 PROGRAM # (
03 04527 005215 TXT.1:  .TXTE (<15><12>
04 04530 141501 AC'S (
05 04533 005215 TXT.2:  .TXTE (<15><12>
06 04534 141523 SCRLO/HI (
07 04541 005215 TXT.7:  .TXTE !<15><12>ST.LA    START / ERROR(RES.)<15><12>!
08 04560 005215 TXT.8:  .TXTE !<15><12>RAN AC=  !
09 04566 177777 PFTEX:  .TXTE !<177><177><177><15><12>POWER FAIL @ !
10
```

```
10091 MNMRT
01                ;AMSCR-ASIGN A SCRATCH AREA TO DCM
02                ;NO SKIP IF MEMORY ASSIGNED
03                ;OR NO SCRATCH AREA AVAILABLE TO ASSIGN
04                AMSCR:
05 04600 062401           SAVE
06 04601 054426           STA 3,AM.S3
07 04602 022130           LDA 0,@ALTBL      ;GET #1K'S ASSIGNED
08 04603 101005           MOV 0,0,SNR       ;NOT=0 INVALID NONSCR TO ASSIGN
09 04604 000411           JMP AM.XT
10 04605 030130           LDA 2,ALTBL
11 04606 025001           LDA 1,1,2
12 04607 133000           MOV 1,3           ;SAVE AC1
13 04610 125120           MOVZL 1,1
14 04611 125224           MOVZR 1,1,SZR     ;ERROR EXIT IF
15 04612 000403           JMP AM.XT         ;DCM ALREADY ASSIGNED
16 04613 000404           JMP AM.NM         ;NO MAP
17 04614 012413           ISZ @AM.S3
18 04615 006414 AM.XT:    JSR @AM.K1        ;LOAD MAP OPT. SET SCRLO+HI
19 04616 062601           RTRN
20                ;MAP OPT DOES NOT EXIST
21                ;USE FIRST 1K SCR ASSIGNED
22 04617 102400 AM.NM:    SUB 0,0
23 04620 006413           JSR @AM.GA
24 04621 063077           HALT
25 04622 125300           MOVS 1,1
26 04623 125700           INCS 1,1
27 04624 030130           LDA 2,ALTBL       ;ENTER DCM ASSIGND
28 04625 045001           STA 1,1,2
29 04626 000766           JMP AM.XT-1
30 04627 000000 AM.S3:    0
31 04630 000037 AM.37:    37
32 04631 004672 AM.K1:    LDCML
33 04632 000000 AM.TM:    0
34 04633 002363 AM.GA:    GETPA
```

```
10092 MNMRT
01                ;EMSCR-
02                ;EXPAND DCM SCRATCH AREA
03
04                EMSCR:
05 04634 062401           SAVE
06 04635 054772           STA 3,AM.S3
07 04636 030130           LDA 2,ALTBL
08 04637 021000           LDA 0,0,2         ;AC0=#1K SCR
09 04640 025001           LDA 1,1,2         ;AC1=DCM LIMIT
10 04641 034767           LDA 3,AM.37
11 04642 137400           AND 1,3           ;3=START DCM LOG
12 04643 166700           SUBS 3,1          ;1=#1K'S DCM
13 04644 122415           SUB# 1,0,SNR      ;SKIP=NOT ALL AS-DCM
14 04645 000750           JMP AM.XT         ;EXIT ALL SR-DCM
15 04646 125700           INCS 1,1          ;+1 #DCM 1K'S
16 04647 137000           ADD 1,3
17 04650 055001           STA 3,1,2
18 04651 000743           JMP AM.XT-1       ;SKIP EXIT 1 MORE 1K
19
```

```
10093 MNMRT
01                ;RDSCR - RELEASE SCRATCH FROM DCH
02                RDSCR:
03 04652 062401           SAVE
04 04653 054754           STA 3,AM,S3
05 04654 030130           LDA 2,ALTBL
06 04655 021001           LDA 0,1,2
07 04656 000737           JMP AM,XT         ;EXIT NO DCH TO RELEASE
08 04657 126000           ADC 1,1           ;-1
09 04660 101300           MOVS 0,0
10 04661 123300           ADDS 1,0          ;#1K'S IN DCH-1
11 04662 041001           STA 0,1,2
12 04663 024745           LDA 1,AM,37       ;(AC1) AFTER AND
13 04664 107400           AND 0,1           ;=LOGICAL PAGE #
14 04665 122704           SUBS 1,0,SZR      ;AC0=#1K'S LEFT
15 04666 041001           STA 0,1,2         ;0 DCH ASSIGNED
16 04667 014740           DSZ AM,S3         ;NO SKIP ON EXIT
17 04670 000724           JMP AM,XT-1       ;EXIT NEW DCH LIMITS
18 04671 063077           HALT    ;DCH MAP BIT WAS ALRDY 1(CAN'T HAPPEN)
```

```
10094 MNMRT
01                ;LDDCHL - LOAD DATA CHANNEL LIMITS
02                ;CALCULATE DCHHI/LO FOR THIS TST
03                ;INTEGRITY OF AC'S NOT PRESERVED
04 04672 030130 LDCHL:    LDA 2,ALTBL
05 04673 025001           LDA 1,1,2         ;AC1 UPPER=#1K'S
06 04674 131000           MOV 1,2
07 04675 020733           LDA 0,AM,37       ;AC1 LOWER=1ST LOG 1K
08 04676 123400           AND 1,0
09 04677 106400           SUB 0,1
10 04700 101300           MOVS 0,0
11 04701 103120           ADDZL 0,0         ;AC0=LO LIMIT DCH
12 04702 040145           STA 0,DCHLO
13 04703 127125           ADDZL 1,1,SNR     ;AC1=MOD 1K # 1'K
14 04704 125040           MOVO 1,1          ;SET C=1 NO 1K'S
15 04705 123000           ADD 1,0           ;AC0=HIGH SCR+1
16 04706 126500           SUBL 1,1          ;AC1=1 IF DCH ASSIGNED
17 04707 122400           SUB 1,0           ;AC1=0 IF NO DCH
18 04710 040146           STA 0,DCHHI       ;HI DCH LOG LIMIT
19 04711 001400           JMP 0,3
```

```
;0095 MNMRT
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 04712 001032                 LAUTO
07 04713 054465 LPRSL:  STA 3,LPSV3
08 04714 102400                 SUB 0,0
09 04715 040464                 STA 0,LPRGN
10 04716 020471                 LDA 0,LPRT1
11 04717 006063                 JSR @ERRTX
12 04720 020462                 LDA 1,@LPHIK    ;GET HIGHK PHYS.
13 04721 125400                 INC 1,1         ;=MOD 1K
14 04722 006062                 JSR @LPDEC      ;PRINT SUPR 0'S
15 04723 020767                 LDA 1,@LPRSL-1
16 04724 020512                 LDA 0,LPRT2
17 04725 125004                 MOV 1,1,SZR     ;SKP IF AUTO STRT
18 04726 020523                 LDA 0,LPRT3     ;NOT AUTO USE OTHR HDR
19 04727 006063                 JSR @ERRTX
20 04730 020447                 LDA 0,LPRT4
21 04731 006063                 JSR @ERRTX      ;PRT PRG HDR
22 04732 006060                 JSR @PCRLF      ;CARRET LFEED TST 0
23                      ;NEXT PAGE PRINT INDIVIDUAL PROGRAM DESCRIPTIONS
```

```
;0096 MNMRT
01                      ;PRINT IDIVIDUAL TEST DESCRIPTIONS
02                      ;GIVE OPR CHANCE TO DELETE IF LAUTO=-1
03 04733 020446 LPRLP:  LDA 0,LPRGN     ;CUR PROG #
04 04734 034442         LDA 3,LPLZM
05 04735 117000         ADD 0,3
06 04736 031400         LDA 2,0,3
07 04737 151005         MOV 2,2,SNR     ;SKP IF NOT LAST
08 04740 002440         JMP @LPSV3
09 04741 105000         MOV 0,1
10 04742 021002         LDA 0,2,2       ;GET PRG WAIT SW
11 04743 101004         MOV 0,0,SZR     ;SKP IF PRG NOT WAIT
12 04744 000430         JMP LPR1E       ;DEV MUST NOT EXIST
13 04745 050441         STA 2,LPIDX
14 04746 006061         JSR @LZUCT      ;TYPE PRG #
15 04747 020437         LDA 0,LPIDX
16 04750 024435         LDA 1,LPR10
17 04751 123000         ADD 1,0         ;CALC ADRS DESC TXT
18 04752 006063         JSR @ERRTX
19 04753 020737         LDA 1,@LPRSL-1
20 04754 125005         MOV 1,1,SNR     ;SKP IS LET OPR SELECT
21 04755 000416         JMP LPR1E-1     ;CR/LF AND DO NXT PRG
```

```
↓0097 MNMRT
01                      ;WAIT FOR OPERATOR INPUT TO SELECT TEST
02                      ;SPACE IS SELECT ANY OTHER IS DELETE
03 04756 063610         SKPDN TTI
04 04757 000777         JMP .-1
05 04760 064610         DIAC 1,TTI      ;GET CHAR
06 04761 030422         LDA 2,LPR77
07 04762 147400         AND 2,1
08 04763 030421         LDA 2,LPR40
09 04764 146415         SUB# 2,1,SNR    ;SKP IF DELETED
10 04765 000406         JMP LPR1E-1     ;SELECTED CR/LF DO NEXT
11 04766 030420         LDA 2,LPIDX
12 04767 102000         ADC 0,0
13 04770 041002         STA 0,2,2       ;SET WAIT SW IN PRG
14 04771 006116         JSR @LMESS
15 04772 005171         LPDTX+1
16 04773 006060         JSR @PCRLF      ;CR/LF
17 04774 010405 LPR1E:  ISZ LPRGN       ;+1 PROG #
18 04775 000736         JMP LPRLP       ;DO NEXT PROG
19 04776 000147 LPLZM: LZMAX
20 04777 005156 LPRT4: LPR4T
21 05000 000000 LPSV3: 0
22 05001 000000 LPRGN: 0
23 05002 001422 LPHIK: HIGHK
24 05003 000077 LPR77: 77
25 05004 000040 LPR40: 40
26 05005 000010 LPR10: 10
27 05006 000000 LPIDX: 0
28 05007 005010 LPRT1: .+1
29 05010 005215         .TXTE |<15><12>MNMRT REV. 00  09/03/76
30 05024 100466 <15><12>TOTAL # 1K'S = |
31 05036 005037 LPRT2:  .+1
32 05037 005215         .TXTE (<15><12>PROGRAM RUN LIST(
33 05051 005052 LPRT3:  .+1
34 05052 005215         .TXTE (<15><12>SPACE SELECTS
35 05061 120123  -ANY OTHER CHAR. DELETES(
36 05077 005215 FTYTX:  .TXTE |<15><12>50 ERRORS<15><12>|
37 05106 005215 KEY6T:  .TXTE |<15><12>TYPE ANY KEY|
38 05116 005215 TX65K:  .TXTE |<15><12>65K PASSES TEST # |
39 05131 005215 STHDR:  .TXTE |<15><12>PROG#    PASSES  ERRS<15><12>|
40 05144 005215         .TXTE |<15><12>UNEX. INTR. DEV# |
41 05156 005215 LPR4T:  .TXTE (<15><12>PRG#     DESCRIPTION(
42 05170 005171 LPDTX:  .+1
43 05171 042240         .TXTE   ( DELETED(
44 05176 005215 UUEVT:  .TXTE |<15><12>UNEX. INTR. DEV# |
```

```
↓0098 MNMRT
01                      .MACRO SETUP
02                      LCALL SETUL
03                  %
04
05                      .MACRO LOOP
06                      LCALL .LOOP
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                  %
```

```
10099 MNMRT
01                      ;       .TITL CHRDS
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                      ;DEFNS TO LINKER PARAMETER FILE FOLLOWS
09                              NEXIT CB.00
10              005210 LMEML=.
11              000147          .LOC LPG0
12 00147 005213         CB.00
13              000150 LPG0=.
14              005210          .LOC LMEML
15 05210 000000                 0       ;TEST PASS CTR
16 05211 000000                 0       ;TEST ERROR CTR
17 05212 000000                 0       ;INTERRUPT TIMEOUT SWITCH
18 05213 005232 CB.00:  CB.01   ;INIT ENTRY ADRS.
19 05214 005235         CB.02   ;EXECUTE ENTRY ADRS
20 05215 000000                 0       ;NO INTR WAITS
21 05216 000000                 0       ;RAND SEL LIMITS
22 05217 177777                 -1      ;=ALWAYS ENTER
23 05220 176000                 176000  ;EVERY PROTECT BIT ON
24 05221 005363                 CB.EC   ;NO I/O VALIDITY TRAPS
25 05222 005363                 CB.EC   ;NO WRITE OR DEFER TRAPS
26              005223          .TXTE (
27 05223 044303 CHKRBRD RAN.(
28              151113
29              151102
30              120104
31              040722
32              027116
33              000000
```

```
10100 MNMRT
01                      ;INITIALIZE CHECKERBOARD TEST SEQUENCE
02 05232 102000 CB.01:  ADC 0,0
03 05233 040555         STA 0,CB.TK     ;-1 TO TEST COUNTER
04 05234 001400         JMP 0,3 ;RETURN TO LINKER TEST INIT
05                      ;
06                      ;EXECUTE ENTRY POINT
07 05235 010553 CB.02:  ISZ CB.TK       ;SKP IS NO SCRATCH
08 05236 000473         JMP CB.03       ;DO NXT IN SEQ
09                      LCALL ASCRA     ;TRY TO GET 1K
10 05237 100010         ASCRA=ASCRA*1B11+100010
11 05240 000507         JMP CB.X1       ;NONE AVAILABLE
12 05241 102000         ADC 0,0         ;-1 TO
13 05242 040557         STA 0,CB.ES     ;NO ERRSW
14                      LCALL ARANG     ;GET RAN#
15 05243 100270         ARANG=ASCRA*1B11+10001C
16 05244 030545         LDA 2,CB.37
17 05245 105000         MOV 0,1
18                      LCALL ADIVI     ;REM=#1K'S TO EXPAND
19 05246 100310         ADIVI=ASCRA*1B11+100010
20 05247 100405         NEG 0,0,SNR
21 05250 000405         JMP CB.2A
22                      LCALL ESCRA     ;EXPAND SCRATCH 1K
23 05251 100030 CB.2L:  ESCRA=ASCRA*1B11+100010
24 05252 000403         JMP CB.2A       ;NO MORE AVAILABLE
25 05253 101404         INC 0,0,SZR
26 05254 000775         JMP CB.2L       ;KEEP EXPANDING
```

```
10101 MNMRT
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 05255 100270           ARANG=ASCRA*1B11+100010
06 05256 024143           LDA 1,SCRLO
07 05257 030144           LDA 2,SCRHI
08 05260 132400           SUB 1,2         ;# WORDS IN SCRATCH
09 05261 105000           MOV 0,1
10                        LCALL ADIVI     ;CREATE AN ADRS
11 05262 100310           ADIVI=ASCRA*1B11+100010
12 05263 024143           LDA 1,SCRLO     ;WITHIN SCRATCH
13 05264 123000           ADD 1,0         ;TO RELOCATE TO
14 05265 030525 CB.2C:    LDA 2,Cb.PL     ;PROG LENGTH
15 05266 142400           SUB 2,0
16 05267 024534           LDA 1,CBPL2
17 05270 122400           SUB 1,0         ;CALC LOW REL ADRS
18 05271 024143           LDA 1,SCRLO
19 05272 122433           SUBZ# 1,0,SNC   ;SKP >LOLIMIT
20 05273 143000           ADD 2,0 ;MAKE GRTR THAN LO
21 05274 122433           SUBZ# 1,0,SNC
22 05275 000776           JMP .-2
23 05276 040515           STA 0,Cb.LC     ;ADRS TO STORE TSTS
24 05277 144000 CB.RL:    COM 2,1         ;-#WORDS TO MOV
25 05300 111000           MOV 0,2         ;TO ADRS
26 05301 034513           LDA 3,Cb.BG     ;FROM ADRS
27 05302 021400 CB.L2:    LDA 0,0,3
28 05303 041000           STA 0,0,2
29 05304 175400           INC 3,3
30 05305 151400           INC 2,2
31 05306 125404           INC 1,1,SZR
32 05307 000773           JMP CB.L2       ;MOV ALL TO SCRTCH
33 05310 020513           LDA 0,CBPL2
34 05311 104000           COM 0,1
35 05312 143000           ADD 2,0
36 05313 101400           INC 0,0
37 05314 041377           STA 0,-1,2      ;SET EPROG
38 05315 020476           LDA 0,CB.LC
39 05316 041376           STA 0,-2,2      ;SET BPROG
40 05317 102400           SUB 0,0
41 05320 041374           STA 0,-4,2      ;CLR PONES
42 05321 050474           STA 2,CB.EN
43                ;MOV REST OF TEST INTO SCRATCH AREA
44 05322 034500           LDA 3,CBBG2     ;STRT OF 2ND SECTION
45 05323 021400 CB.L3:    LDA 0,0,3
46 05324 041000           STA 0,0,2
47 05325 151400           INC 2,2
48 05326 175400           INC 3,3
49 05327 125404           INC 1,1,SZR
50 05330 000773           JMP CB.L3
51                ;2ND PART OF TEST IS MOVED
```

```
10102 MNMRT
01
02                ;TEST PROGRAMS HAVE BEEN MOVED TO SCRATCH AREA
03                ;EXECUTE NEXT TES IN SEQUENCE
04 05331 030457 CB.03:    LDA 2,CB.TK     ;GET TEST COUNT
05 05332 024472           LDA 1,CB.TS     ;ADRS SEQ TABLE
06 05333 133000           ADD 1,2
07 05334 035000           LDA 3,0,2       ;GET RELATIVE POSITION
08 05335 024456           LDA 1,CB.LC     ;STRT POS IN SCRATCH
09 05336 137000           ADD 1,3
10 05337 050457           STA 2,CB.TI
11 05340 054460           STA 3,CB.SE
12 05341 004400           JSR 0,3 ;***GO TO TEST ***
13 05342 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 05343 020454           LDA 0,CTOTL     ;FAST SUM RESULT
19 05344 030451           LDA 2,CB.EN
20 05345 025374           LDA 1,-4,2      ;#-1'S GENERATED
21 05346 004413           JSR CB.ER
22                CB.X1:  LCALL RSCRA     ;RELEASE 1K SCRATCH
23 05347 100050           RSCRA=ASCRA*1B11+100010
24 05350 102001           ADC 0,0,SKP     ;RET ALL RELEASED
25 05351 000776           JMP CB.X1       ;RELEASE ALL
26 05352 040436           STA 0,CB.TK     ;SET NO SCRATCH SW
27                        LCALL RETRN
28 05353 100210           RETRN=ASCRA*1B11+100010
29                ;NORMAL RETURN FROM TEST SEE IF PASS COMPLETE
30 05354 010442 CB.04:    ISZ CB.TI
31 05355 022441           LDA 0,@CB.TI    ;NEXT IN SEQ
32 05356 100005           COM 0,0,SNR     ;=-1 WAS END SEQ
33 05357 000417           JMP CB.X2       ;AND RELEASE SCRA.
34                        LCALL RETRN
35 05360 100210           RETRN=ASCRA*1B11+100010
```

```
10103 MNMRT
01                ;PATTERN CHECK FOUND AN ERROR
02 05361 054437 CB.ER:  STA 3,CB.SE
03                      LCALL ERRDI
04 05362 100350         ERRDI=ASCRA+1B11+100010
05 05363 000401 CB.EC:  JMP .+1 ;CONTINUE ERROR TYPEOUT
06 05364 020446         LDA 0,CBTXT
07                      LCALL ERRTX
08 05365 100170         ERRTX=ASCRA+1B11+100010
09 05366 020422         LDA 0,CB.TK     ;TEST #
10 05367 024424         LDA 1,CB.LC     ;LOGICAL ADDRESS
11 05370 030430         LDA 2,CB.SE     ;START TEST OR E CALL
12                      LCALL ERRDC
13 05371 100370         ERRDC=ASCRA+1B11+100010
14 05372 000755         JMP CB.X1       ;SW0=1 RELEASE SCRATCH
15 05373 102400         SUB 0,0         ;OTHERWISE HOLD IT
16 05374 040425         STA 0,CB.ES     ;AS IS UNTIL SW0=1
17 05375 000757         JMP CB.04
18
19 05376 020127 CB.X2:  LDA 0,SWREG     ;CHECK IF RELEASEING
20 05377 103123         ADDZL 0,0,SNC   ;SCRATCH=SKP IS NOT REL.
21 05400 000445         JMP CB.05       ;RELEASE IT
22 05401 020420         LDA 0,CB.ES     ;GET ERR SWITCH
23 05402 040406         STA 0,CB.TK     ;-1 IS NO ERRS
24 05403 101004         MOV 0,0,SZR
25 05404 000441         JMP CB.05       ;AND SCRATCH IS RELEASED
26 05405 020406         LDA 0,CB.LC     ;OTHERWISE MOVE
27 05406 030404         LDA 2,CB.PL     ;PROGRAM UP AND
28 05407 000670         JMP CB.RL       ;RESTART AT TEST 0
29
30 05410 000000 CB.TK:  0
31 05411 000037 CB.37:  37
32 05412 000151 CB.PL:  EPROG-BEGIN
33 05413 000000 CB.LC:  0
34 05414 005406 CB.BG:  BEGIN
35 05415 000000 CB.EN:  0
36 05416 000000 CB.TI:  0
37 05417 000000 DTOTL:  0
38 05420 000000 CB.SE:  0
39 05421 000000 CB.ES:  0
40 05422 005640 CBBG2:  DISTUR
41 05423 000104 CBPL2:  DIRET-DISTUR
42 05424 005425 CB.TS:   CB.TS+1 ;TEST SEQUENCE TABLE
43 05425 000000          BEGIN-BEGIN
44 05426 000152          DISTUR-BEGIN
45 05427 000045          ICHECK-BEGIN
46 05430 000101          CB.FA-BEGIN
47 05431 177777          -1
48 05432 005433 CBTXT:  .+1
49 05433 005215          .TXTE (<15><12>CB.TK    CB.LC   CB.SE(
```

```
10104 MNMRT
01                ;DETERMINE IF IT IS TIME TO SWAP MEMORY
02                ;CROSSOVER CONSTANTS
03                CB.05: LCALL ARANG
04 05445 100270          ARANG=ASCRA+1B11+100010
05 05446 024557          LDA 1,C17
06 05447 123404          AND 1,0,SZR
07 05450 000677          JMP CB.X1
08 05451 020411          LDA 0,CB17
09 05452 030411          LDA 2,CB400
10 05453 100414          SUB# 0,1,SZR
11 05454 000403          JMP .+3
12 05455 020407          LDA 0,CB37
13 05456 030407          LDA 2,CB10K
14 05457 040546          STA 0,C17
15 05460 050544          STA 2,C400
16 05461 000666          JMP CB.X1
17 05462 000017 CB17:  17
18 05463 000400 CB400:  400
19 05464 000037 CB37:  37
20 05465 010000 CB10K:  10000
```

```
10105 MNMRT
01
02                ;THE FOLLOWING SERIES OF TESTS ARE RELOCATED TO
03                ;SCRATCH AREA ASSIGNED FOR EXECUTION
04 05466 054532 BEGIN:  STA 3,RETURN
05 05467 102400         SUB 0,0
06 05470 040544         STA 0,PUNES
07 05471 034143         LDA 3,SCRLO
08 05472 030527         LDA 2,C076000
09 05473 020144         LDA 0,SCRHI       ;3/0=LIMITS
10 05474 143400         AND 2,0           ;END DISTURB
11 05475 040526         STA 0,EDIST
12 05476 173400         AND 3,2
13 05477 050523         STA 2,MODUAL
14 05500 030143 IPAT:   LDA 2,SCRLO
15 05501 024523         LDA 1,C400
16 05502 020533         LDA 0,PATT        ;START PATTERN
17 05503 147404         AND 2,1,SZR
18 05504 100000 IPAT1:  COM 0,0
19 05505 024520         LDA 1,C17
20 05506 034530 FILL:   LDA 3,BPROG       ;AVOID OVERWRITE TST
21 05507 156436         SUBZ# 2,3,SEZ
22 05510 000404         JMP .+4
23 05511 034526         LDA 3,EPROG
24 05512 172433         SUBZ# 3,2,SNC     ;SKP>PROG END
25 05513 000405         JMP .+5
26 05514 041000         STA 0,0,2         ;STR PAT WORD
27 05515 034517         LDA 3,PUNES
28 05516 117000         ADD 0,3
29 05517 054515         STA 3,PUNES       ;ACCUMULATE CHKSUM
30 05520 151400         INC 2,2           ;FOR NXT STOR
31 05521 034506         LDA 3,C77
32 05522 133414         AND# 1,2,SZR      ;FILL FOR 16
33 05523 000763         JMP FILL
34 05524 157414         AND# 2,3,SZR      ;THEN COM PAT
35 05525 000757         JMP IPAT1         ;EVERY 64
36 05526 024144         LDA 1,SCRHI       ;TST FOR END
37 05527 125400         INC 1,1
38 05530 146434         SUBZ# 2,1,SZR
39 05531 000750         JMP IPAT+1
40 05532 002466         JMP @RETURN
```

```
10106 MNMRT
01                ;CHECK PATTERN IN SCRATCH AREA AGAINST GENERATED
02 05533 054465 ICHECK: STA 3,RETURN
03 05534 030143         LDA 2,SCRLO       ;INIT PAT
04 05535 024467         LDA 1,C400
05 05536 020477         LDA 0,PATT
06 05537 133414         AND# 1,2,SZR
07 05540 100000 ICK:    COM 0,0
08 05541 024475         LDA 1,BPROG       ;DON'T CMP TST STOR
09 05542 146436         SUBZ# 2,1,SEZ
10 05543 000405         JMP CHECK
11 05544 024473         LDA 1,EPROG
12 05545 132436         SUBZ# 1,2,SEZ
13 05546 000402         JMP CHECK
14 05547 000404         JMP ECHECK+4
15 05550 020000 CHECK:  LDA 1,0,2         ;GET WORD FROM MEM
16 05551 106414         SUB# 0,1,SZR      ;SKP=
17 05552 006461         JSR @ERR1
18 05553 151400         INC 2,2
19 05554 024451         LDA 1,C17
20 05555 133414         AND# 1,2,SZR
21 05556 000763         JMP ICK+1
22 05557 034450 ECHECK: LDA 3,C77
23 05560 157414         AND# 2,3,SZR      ;CHK END OF LINE
24 05561 000757         JMP ICK
25 05562 024144         LDA 1,SCRHI
26 05563 125400         INC 1,1
27 05564 146434         SUBZ# 2,1,SZR     ;CHK END OF CORE
28 05565 000750         JMP ICHECK+2
29 05566 002432         JMP @RETURN
```

```
;0107 MNMRT
01
02                ;FAST CHECKSUM PATTERN SHOULD = THAT STORED
03 05567 054431 CB.FA:  STA 3,RETURN
04 05570 102400         SUB 0,0
05 05571 034143         LDA 2,SCRLO       ;BEGIN OF SCRA.
06 05572 034444         LDA 3,BPROG       ;START OF TST STORE
07 05573 156405 CB.F1:  SUB 2,3,SNR
08 05574 000407         JMP CB.F3
09 05575 174400         NEG 3,3 ;(3)=#WORDS TO ADD
10 05576 025000 CB.F2:  LDA 1,0,2         ;GET WRD
11 05577 123000         ADD 1,0           ;ACCUM SUM-1'S
12 05600 151400         INC 2,2
13 05601 175404         INC 3,3,SZR
14 05602 000774         JMP CB.F2
15 05603 034144 CB.F3:  LDA 3,SCRHI
16 05604 175420         INC 3,3
17 05605 156415         SUB# 2,3,SNR      ;DONE ALL CORE
18 05606 000403         JMP CB.F4         ;YES EXIT
19 05607 034430         LDA 2,EPROG
20 05610 000763         JMP CB.F1         ;ADD ABOVE TSTS
21 05611 024423 CB.F4:  LDA 1,PONES       ;(1)=#-1'S GEN
22 05612 122414         SUB#1,0,SZR       ;SHD BE=
23 05613 000402         JMP CB.F5         ;BUT AREN'T
24 05614 000404         JMP @RETURN
25 05615 410403 CB.F5:  ISZ RETURN        ;STEP EXIT
26 05616 042414         STA 0,@CB.WK      ;SAVE IN CASE
27 05617 000715         JMP ICHECK+1      ;CHECK CAN'T FIND ERR
28 05620 000000 RETURN: 0
29 05621 076000 C07600: 76000
30 05622 000000 MODUAL: 0
31 05623 000000 EDIST: 0
32 05624 000400 C400: 400
33 05625 000017 C17: 17
34 05626 000020 C20: 20
35 05627 000077 C77: 77
36 05630 001777 C1777: 1777
37 05631 000101 C101: 101
38 05632 005417 CB.WK: OTUTL
39 05633 005361 ERR1: CB.ER
40 05634 000000 PONES: 0
41 05635 000000 PATT: 0
42 05636 000000 BPROG: 0
43 05637 000000 EPROG: 0
```

```
;0108 MNMRT
01
02
03                ;SHUFFLE MEM BY FLOATING A BIT OR NO BIT THROUGH 16 WORD
04 05640 054700 DISTUR: STA 3,RETURN
05 05641 034143         LDA 2,SCRLO
06 05642 145000 DISTL:  MOV 2,1           ;2=START OF 16
07 05643 020763         LDA 0,C20         ;MAKE 1=END+1
08 05644 107000         ADD 0,1
09 05645 034771         LDA 3,BPROG       ;START OR "BEGIN)
10 05646 136436         SUBZ# 1,3,SEZ     ;END 16>BEG
11 05647 000411         JMP DISDO         ;NO<DO THIS 16
12 05650 034707         LDA 3,EPROG       ;GET END PRG
13 05651 172436         SUBZ# 3,2,SEZ     ;STRT 16>END PRG
14 05652 000403         JMP .+3           ;YES
15 05653 131000         MOV 1,2           ;TRY STRT NXT16
16 05654 000766         JMP DISTL
17 05655 034144         LDA 3,SCRHI
18 05656 166436         SUBZ# 3,1,SEZ     ;END 16>SCRHI
19 05657 002741         JMP @RETURN       ;YES EXIT JOB DONE
20 05660 004450 DISDO:  JSR DIXOR         ;FLT A 1 BIT 16 WORDS
21 05661 020745         LDA 0,C20
22 05662 112400         SUB 0,2           ;BACK TO START OF 16
23 05663 114400         NEG 0,3           ;SHUFFLE 16 TIMES
24 05664 021000 DISDL:  LDA 0,0,2         ;WORD 0
25 05665 025001         LDA 1,1,2
26 05666 041001         STA 0,1,2         ;GOES TO WORD 1
27 05667 021002         LDA 0,2,2
28 05670 045002         STA 1,2,2         ;1 GOES TO 2
29 05671 025003         LDA 1,3,2
30 05672 041003         STA 0,3,2         ;2 GOES TO 3
31 05673 021004         LDA 0,4,2
32 05674 045004         STA 1,4,2         ;3 GOES TO 4
33 05675 025005         LDA 1,5,2
34 05676 041005         STA 0,5,2         ;4 GOES TO 5
35 05677 021006         LDA 0,6,2
36 05700 045006         STA 1,6,2         ;5 GOES TO 6
37 05701 025007         LDA 1,7,2
38 05702 041007         STA 0,7,2         ;6 TO 7
39 05703 021010         LDA 0,10,2
40 05704 045010         STA 1,10,2        ;7 TO 10
41 05705 025011         LDA 1,11,2
42 05706 041011         STA 0,11,2        ;10 TO 11
43 05707 021012         LDA 0,12,2
44 05710 045012         STA 1,12,2        ;11 TO 12
45 05711 025013         LDA 1,13,2
46 05712 041013         STA 0,13,2        ;12 TO 13
47 05713 021014         LDA 0,14,2
48 05714 045014         STA 1,14,2        ;13 TO 14
49 05715 025015         LDA 1,15,2
50 05716 041015         STA 0,15,2        ;14 TO 15
51 05717 021016         LDA 0,16,2
52 05720 045016         STA 1,16,2        ;15 TO 16
53 05721 025017         LDA 1,17,2
54 05722 041017         STA 0,17,2        ;16 TO 17
55 05723 045000         STA 1,0,2         ;AND 17 BACK TO 0
56 05724 175404         INC 3,3,SZR       ;DONE 16 TIMES
57 05725 000737         JMP DISDL         ;WORDS NOT BACK TO ORIG YET
58 05726 004402         JSR DIXOR         ;XOR BITS BACK TO ORIG
59 05727 000713         JMP DISTL         ;DO REST OF SCRATCH
```

```
↓0109 MNMRT
01                      ;COMS A SINGLE  BIT IN EACH OF NEXT 16 WORDS
02 05730 054414 DIXOR:  STA 3,DIRET     ;THEN RESTORES TO ORIG
03 05731 102520         SUBZL 0,0       ;XOR 1 BIT
04 05732 025000         LDA 1,0,2       ;GET NEXT WRD
05 05733 135000         MOV 1,3         ;START BIT XOR
06 05734 117520         ANDZL 0,3       ;SO FLT BIT PAT
07 05735 107000         ADD 0,1         ;APPEARS IN THESE
08 05736 166400         SUB 3,1         ;16 WORDS
09 05737 045000         STA 1,0,2
10 05740 151400         INC 2,2
11 05741 101124         MOVZL 0,0,SZR   ;DONE ALL - POS BIT L1
12 05742 000770         JMP DIXOR+2
13 05743 002401         JMP @DIRET
14 05744 000000 DIRET:  0
15                      .ENDC
```

```
↓0110 MNMRT
01
02                  ;SC-MEMORY TEST - DEFINE PARAMETERS TO LINKR
03
04                          NEXTT  MM.00
05          005745 LMEML=.
06          000150          .LOC LPG0
07 00150 005750             MM.00
08          000151 LPG0=.
09          005745          .LOC LMEML
10 05745 000000             0           ;TEST PASS CTR
11 05746 000000             0           ;TEST ERROR CTR
12 05747 000000             0           ;INTERRUPT TIMEOUT SWITCH
13
14 05750 005770 MM.00:      MM.01                           ;INIT ENTRY
15 05751 005773             MM.02                           ;EXEC ENTRY
16 05752 000000             0                               ;WAIT SW
17 05753 000000             0                               ;
18 05754 177777             -1                              ;ALWAYS ENTER
19 05755 176000             176000                          ;EVERY PROTECT ON
20 05756 006137             MM.EC                           ;NO I/O VALIDITY TRAPS
21 05757 006137             MM.EC                           ;NO WRITE/DEFER TRAPS
22 05760 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 05770 001400 MM.01:  JMP 0,3     ;RETURN TO CALLER
34
35
36
37
38
39 05771 000067 MM.PL:  MM.EN-MM.3A
40 05772 006032 MM.LC:  MM.3A
```

```
10111 MNMRT
01
02                ;EXECUTE ENTRY POINT
03                ;GET SCRATCH AREA
04 05773 020127 MM.02:  LDA 0,SWREG             ;CHECK IF RELEASEING SCRATCH
05 05774 103123         ADDZL 0,0,SNC           ;SKP IS NOT REL.
06 05775 000404         JMP .+4                 ;RELEASING SCRATCH
07 05776 020525         LDA 0,MM.ES             ;ERR  SWITCH
08 05777 101005         MOV 0,0,SNR             ;SKP IS NOT HAD ERROR
09 06000 000407         JMP MM.2A               ;ERR HAS OCCURRED
10                                              ;DON'T NEED TO GET SCRATCH
11 06001 102000         ADC 0,0
12 06002 040521         STA 0,MM.ES             ;CLR ERR SWITCH
13                      LCALL ASCRA             ;GET 1K SCRATCH
14 06003 100010         ASCRA=ASCRA+1B11+100010
15 06004 000520         JMP MM.X1               ;EXIT IF NONE AVAILABLE
16                MM.2L:  LCALL ESCRA             ;EXPAND SCRATCH BY 1K
17 06005 100030         ESCRA=ASCRA+1B11+100010
18 06006 000401         JMP MM.2A               ;NO MORE AVAILABLE
19
20                ;MOVE TESTS TO SCRATCH AREA
21 06007 030143 MM.2A:  LDA 2,SCRLO
22 06010 034762         LDA 3,MM.LC     ;FROM HERE
23 06011 141000         MOV 2,0
24 06012 024757         LDA 1,MM.PL
25 06013 123000         ADD 1,0
26 06014 040504         STA 0,MM.SA     ;SAVE STARTING ADDR
27 06015 124000         COM 1,1         ;-# OF WORDS TO MOVE
28 06016 021400 MM.L2:  LDA 0,0,3
29 06017 041000         STA 0,0,2
30 06020 151400         INC 2,2
31 06021 175400         INC 3,3
32 06022 125404         INC 1,1,SZR
33 06023 000773         JMP MM.L2       ;CONTINUE
34
35                ;NOW EXECUTE TESTS IN SCRATCH
36 06024 034143 MM.03:  LDA 3,SCRLO
37 06025 005400         JSR 0,3         ;GO TO SCRATCH
38 06026 020475         LDA 0,MM.ES     ;ERR SW
39 06027 101004         MOV 0,0,SZR     ;SKP IS ERR HAS OCCURRED
40 06030 000474         JMP MM.X1       ;RELEASE SCRATCH
41 06031 000500         JMP MM.X2       ;DON'T REL. SCRATCH
```

```
10112 MNMRT
01
02                ;SCRATCH AREA HAS BEEN ASSIGNED, FILL AREA WITH BACKGND
03
04 06032 054454 MM.3A:  STA 3,MM.33
05 06033 030465         LDA 2,MM.SA             ;FILL SCRATCH WITH MINUS
06 06034 034144         LDA 3,SCRMI
07 06035 126000         ADC 1,1
08 06036 040000         STA 1,0,2
09 06037 021000         LDA 0,0,2               ;CHECK  IT GOT THERE
10 06040 106434         SUBZ# 0,1,SZR
11 06041 006446         JSR @MM3ER              ;NOT -1,ERROR
12 06042 151400         INC 2,2
13 06043 156432         SUBZ# 2,3,SZC           ;DONE ALL?
14 06044 000771         JMP MM.3A+3             ;NOT YET
15
16                ;ALL SCRATCH IS FILLED, NOW TEST WITH ISZ
17 06045 030453 MM.3B:  LDA 2,MM.SA
18 06046 021000         LDA 0,0,2
19 06047 100014         COM# 0,0,SZR            ;CHECK LOC BEFORE ISZ'IN
20 06050 006440         JSR @MM3E1              ;NOT -1 BEFORE ISZ'NG
21 06051 011000         ISZ 0,2                 ;ISZ THIS LOCATION
22 06052 006437         JSR @MM3E2              ;ERROR,ISZ DIDN'T SKIP
23 06053 021000         LDA 0,0,2               ;GET CONTENTS
24 06054 101004         MOV 0,0,SZR             ;LOC CONTENTS = 0?
25 06055 006435         JSR @MM3E3              ;LOC NOT 0 AFTER ISZ
26 06056 015000         DSZ 0,2                 ;-1 TO LOC AGAIN
27 06057 000402         JMP .+2
28 06060 006433         JSR @MM3E4              ;DSZ SKIPPED-ERROR
29 06061 021000         LDA 0,0,2               ;CHECK CONTENTS
30 06062 100014         COM# 0,0,SZR            ;=-1?
31 06063 006431         JSR @MM3E5              ;LOC NOT -1 AFTER DSZ
32 06064 151400         INC 2,2
33 06065 156432         SUBZ# 2,3,SZC           ;DONE ALL?
34 06066 000760         JMP MM.3B+1             ;NOT YET
35
```

```
↓0113 MNMRT
01
02                      ;NOW DO ISZ TEST IN REVERSE DIRECTION
03 06067 030144 MM.3C:  LDA 2,SCRHI                     ;STARTING ADDRESS
04 06070 034430         LDA 3,MM.SA                     ;FINISHING ADDRESS
05 06071 126000         ADC 1,1                         ;AC1=EXPECTED VALUE BEFO
06 06072 021000         LDA 0,0,2                       ;CHECK LOC BEFORE DOING
07 06073 100014         COM# 0,0,SZR                    ;AC0=-1?
08 06074 006421         JSR @MM3E6                      ;LOC NOT -1 BEFORE ISZ
09 06075 011000         ISZ 0,2                         ;ISZ THIS LOCATION
10 06076 006420         JSR @MM3E7                      ;ISZ DIDN'T SKIP
11 06077 021000         LDA 0,0,2                       ;NOW CHECK LOC.
12 06100 101004         MOV 0,0,SZR                     ;LOC = 0?
13 06101 006416         JSR @MM3E8                      ;LOC NOT 0 AFTER ISZ
14 06102 133000         ADD 1,2                         ;DECREMENT AC2 BY 1
15 06103 172432         SUBZ# 3,2,SZC                   ;IS SCRHI<SCRLO?
16 06104 000766         JMP MM.3C+3                     ;NOT YET
17 06105 002401         JMP @MM.S3      ;RETURN TO MAIN TEST SECTION
18
19                      ;TEST CONSTANTS
20 06106 000000 MM.S3:  0
21 06107 006132 MM3ER:  MM.ER
22 06110 006152 MM3E1:  MM.E1
23 06111 006156 MM3E2:  MM.E2
24 06112 006163 MM3E3:  MM.E3
25 06113 006167 MM3E4:  MM.E4
26 06114 006174 MM3E5:   MM.E5
27 06115 006200 MM3E6:   MM.E6
28 06116 006204 MM3E7:   MM.E7
29 06117 006211 MM3E8:   MM.E8
30 06120 000000 MM.SA:  0
31 06121 000000 MM.EN:  0
32 06122 000000 MM.SE:  0
33 06123 000000 MM.ES:   0       ;ERROR SWITCH
34
35                      ;DONE ALL TESTING,RELEASE SCRATCH
36
37 06124 102000 MM.X1:  ADC 0,0
38 06125 040776         STA 0,MM.ES                     ;CLR ERR REL. SWITCH
39                      LCALL RSCRA                     ;RELEASE 1K SCRATCH
40 06126 100050         RSCRA=ASCRA+1B11+100010
41 06127 102001         ADC 0,0,SKP                     ;RET ALL RELEASED
42 06130 000774         JMP MM.X1                       ;RELEASE ALL
43                      LCALL RETRN                     ;RETURN TO LINKR
44 06131 100210         RETRN=ASCRA+1B11+100010
```

```
↓0114 MNMRT
01                      ;PATTERN CHECK FOUND AN ERROR
02 06132 054770 MM.ER:  STA 3,MM.SE
03 06133 176400         SUB 3,3
04 06134 050472         STA 2,MMLOC                     ;SAVE LOC ADDR.
05 06135 054470         STA 3,MM.TK                     ;SET ERROR #
06                      LCALL ERROI
07 06136 100350         ERROI=ASCRA+1B11+100010
08 06137 000401 MM.EC:  JMP .+1
09 06140 102400         SUB 0,0
10 06141 040762         STA 0,MM.ES             ;SET ERROR OCCURRED SWITCH
11 06142 020465         LDA 0,MMTXT
12                      LCALL ERRTX
13 06143 100170         ERRTX=ASCRA+1B11+100010
14 06144 020461         LDA 0,MM.TK                     ;ERR#
15 06145 024755         LDA 1,MM.SE                     ;ERROR CALL ADDR
16 06146 030460         LDA 2,MMLOC                     ;GET SAVED LOC
17                      LCALL ERROC
18 06147 100370         ERROC=ASCRA+1B11+100010
19 06150 000754         JMP MM.X1
20 06151 000636         JMP MM.2A
21
22                      ;FORWARD ISZ TST = LOC NOT -1 BEFORE ISZ
23 06152 054750 MM.E1:  STA 3,MM.SE
24 06153 126000         ADC 1,1
25 06154 034441         LDA 3,MM.K1
26 06155 000757         JMP MM.ER+2
27                      ;FORWARD ISZ TST = ISZ DIDN'T SKIP
28 06156 054744 MM.E2:  STA 3,MM.SE
29 06157 021000         LDA 0,0,2       ;ACTUAL
30 06160 126400         SUB 1,1         ;EXPECTED = 0
31 06161 034435         LDA 3,MM.K2
32 06162 000752         JMP MM.ER+2
33
34                      ;FORWARD ISZ=LOC NOT 0 AFTER ISZ
35 06163 054737 MM.E3:  STA 3,MM.SE
36 06164 126400         SUB 1,1
37 06165 034432         LDA 3,MM.K3
38 06166 000746         JMP MM.ER+2
39
40                      ;DSZ TEST-DSZ SKIPPED
41 06167 054733 MM.E4:  STA 3,MM.SE
42 06170 021000         LDA 0,0,2       ;ACTUAL
43 06171 126000         ADC 1,1         ;EXPECTED = -1
44 06172 034426         LDA 3,MM.K4
45 06173 000741         JMP MM.ER+2
46
47                      ;DSZ TEST-LOC NOT -1 AFTER DSZ
48 06174 054726 MM.E5:  STA 3,MM.SE
49 06175 126000         ADC 1,1
50 06176 034423         LDA 3,MM.K5
51 06177 000735         JMP MM.ER+2
52
53                      ;REVERSE ISZ TEST-LOC NOT -1 BEFORE ISZ
54 06200 054722 MM.E6:  STA 3,MM.SE
55 06201 126000         ADC 1,1
56 06202 034420         LDA 3,MM.K6
57 06203 000731         JMP MM.ER+2
58
```

```
!0115 MNMRT
01                ;REVERSE ISZ TEST-ISZ DIDN'T SKIP
02 06204 054716 MM.E7:  STA 3,MM.SE
03 06205 021000         LDA 0,0,2
04 06206 126400         SUB 1,1
05 06207 034414         LDA 3,MM.K7
06 06210 000724         JMP MM.ER+2
07
08                ;REVERSE ISZ TST=LOC NOT 0 AFTER ISZ
09 06211 054711 MM.E8:  STA 3,MM.SE
10 06212 126400         SUB 1,1
11 06213 034411         LDA 3,MMK10
12 06214 000720         JMP MM.EC-3
13
14 06215 000001 MM.K1: 1
15 06216 000002 MM.K2: 2
16 06217 000003 MM.K3: 3
17 06220 000004 MM.K4: 4
18 06221 000005 MM.K5: 5
19 06222 000006 MM.K6: 6
20 06223 000007 MM.K7: 7
21 06224 000010 MMK10: 10
22 06225 000000 MM.TK: 0
23 06226 000000 MMLOC: 0
24 06227 005230 MMTXT:  .+1
25 06230 005215         .TXTE [<15><12>MM.TK<11>MM.SE<11>LOCATION]
```

```
!0116 MNMRT
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                %
```

```
10117 MNMRT
01                      ;ARITH - DEFINE PARAMETERS TO LINKR
02                              NEXTT AT,00
03              006244 LMEML=.
04              000151          .LOC LPG0
05 00151        006247          AT,00
06              000152 LPG0=.
07              006244          .LOC LMEML
08 06244        000000          0       ;TEST PASS CTR
09 06245        000000          0       ;TEST ERROR CTR
10 06246        000000          0       ;INTERRUPT TIMEOUT SWITCH
11 06247        006267 AT,00:   AT,01
12 06250        006272          AT,02
13 06251        000000          0
14 06252        000000          0
15 06253        177777          -1
16 06254        176000          176000
17 06255        006432          AT,EC
18 06256        006432          AT,EC
19              006257          .TXTE {
20 06257        151101 ARITHMETIC TEST(
21              152311
22              046510
23              152305
24              141711
25              152240
26              051705
27              000324
28                      ;SET NO SCRATCH ASSIGNED SWITCH
29 06267 102000 AT,01:  ADC 0,0
30 06270 040471         STA 0,AT,TK
31 06271 001400         JMP 0,3
32
```

```
10118 MNMRT
01                      ;EXECUTE ENTRY POINT GET SCRATCH IF NONE AVAIL
02
03 06272 010467 AT,02:  ISZ AT,TK       ;SKIP IS NO SCRATCH
04 06273 000467          JMP AT,03
05                       LCALL ASCRA     ;GET 1K SCRATCH
06 06274 100010          ASCRA=ASCRA*1B11+100010
07 06275 000453          JMP AT,05       ;EXIT NONE AVAILABLE
08 06276 102000          ADC 0,0
09 06277 040555          STA 0,AT,ES     ;SET NO ERR SWTCH
10                       LCALL ESCRA     ;TRY TO GET 1K MORE
11 06300 100030          ESCRA=ASCRA*1B11+100010
12 06301 000401          JMP .+1         ;BUT USE 1K IF NO MORE AVAIL
13 06302 036557          LDA 2,AT,37
14 06303 150400          NEG 2,2         ;TRY RANDOM BETWEEN LIM 37 TRYS
15              AT,2L:   LCALL ARANG
16 06304 100270          ARANG=ASCRA*1B11+100010
17 06305 024144          LDA 1,SCRHI
18 06306 103000          ADD 0,0
19 06307 101220          MOVZR 0,0       ;CLR BIT 0
20 06310 122032          ADCZ# 1,0,SZC
21 06311 000411          JMP AT,2A       ;GRTR THAN HILIM
22 06312 034543          LDA 3,AT,PL
23 06313 166400          SUB 3,1         ;ENOUGH ROOM
24 06314 122032          ADCZ# 1,0,SZC   ;TO RELOCATE UP
25 06315 162400          SUB 3,0         ;NO LOWER NUMBER
26 06316 024143          LDA 1,SCRLO
27 06317 125400          INC 1,1
28 06320 122432          SUBZ# 1,0,SZC
29 06321 000404          JMP AT,2B
30 06322 151404 AT,2A:   INC 2,2,SZR
31 06323 000761          JMP AT,2L
32 06324 121000          MOV 1,0
```

```
10119 MNMRT
01                ;MOVE TESTS TO SELECTED AREA
02
03 06325 111000 AT.2B:  MOV 0,2           ;ADJUST SCRHI
04 06326 020527         LDA 0,AT.PL       ;SO THAT TESTS
05 06327 024144         LDA 1,SCRHI       ;WILL FIT
06 06330 106400         SUB 0,1           ;INTO SCRATCH
07 06331 132032         ADCZ# 1,2,SZC
08 06332 131000         MOV 1,2
09 06333 050523         STA 2,AT.LC       ;START ADRS IN SCRATCH
10 06334 024521 AT.MV:  LDA 1,AT.PL
11 06335 124000         COM 1,1           ;-# WORDS TO MOVE
12 06336 030520         LDA 2,AT.LC       ;TO
13 06337 034520         LDA 3,AT.BG       ;FROM
14 06340 021400 AT.L2:  LDA 0,0,3         ;MOVE LOOP
15 06341 041000         STA 0,0,2
16 06342 151400         INC 2,2
17 06343 175400         INC 3,3
18 06344 125404         INC 1,1,SZR
19 06345 000773         JMP AT.L2
20 06346 050512         STA 2,AT.EN
21 06347 000413         JMP AT.03
22 06350 102000 AT.05:  ADC 0,0
23 06351 040410         STA 0,AT.TK
24              AT.XI:  LCALL RETRN
25 06352 100210         RETRN=ASCRA*1B11+100010
26                ;MAP OPTION DOES NOT EXIST GO DIRECT
27 06353 020503 AT.GD:  LDA 0,AT.LC
28 06354 040507         STA 0,AT.LA
29 06355 020143         LDA 0,SCRLO
30 06356 040504         STA 0,AT.LO
31 06357 040505         STA 0,AT.LP
32 06360 002503         JMP @AT.LA
33 06361 000000 AT.TK:  0
```

```
10120 MNMRT
01                ;TESTS HAVE BEEN MOVED TO SCRATCH
02                ;AT.03 SELECTS LOGICAL PAGE ASSIGNMENT
03                ;AND INITIATES TEST VIA GSCRA
04
05 06362 020472 AT.03:  LDA 0,AT.ES       ;GET ERR SWITCH
06 06363 101005         MOV 0,0,SNR       ;SKIP IS NO ERR
07 06364 000416         JMP AT.04         ;USE PREV ASSIGN
08                      LCALL ARANG
09 06365 100270         ARANG=ASCRA*1B11+100010
10 06366 105000         MOV 0,1
11 06367 030472         LDA 2,AT.37
12                      LCALL ADIVI
13 06370 100310         ADIVI=ASCRA*1B11+100010
14 06371 040473         STA 0,AT.LP
15 06372 024464         LDA 1,AT.LC
16 06373 030143         LDA 2,SCRLO
17 06374 146400         SUB 2,1
18 06375 111300         MOVS 0,2
19 06376 153120         ADDZL 2,2
20 06377 050463         STA 2,AT.LO
21 06400 133000         ADD 1,2
22 06401 050462         STA 2,AT.LA
23 06402 020462 AT.04:  LDA 0,AT.LP       ;REMAP SCR TO HERE
24 06403 030460         LDA 2,AT.LA       ;STARTING LOGICAL ADRS
25 06404 024422         LDA 1,ATERR       ;1=ERROR RET ADDRS
26                      LCALL GSCRA       ;GO TO SCRATCH
27 06405 100070         GSCRA=ASCRA*1B11+100010
28 06406 000745         JMP AT.GD         ;COULDNT REMAP GO DIRECT
29                ;RETURN TO NEXT LOC PASS COMPLETE NO ERRS
30 06407 020127         LDA 0,SWREG       ;CHECK IF REL. SCRATCH
31 06410 103123         ADDZL 0,0,SNC     ;SKP IS NOT REL
32 06411 000406         JMP .+6           ;RELEASE SCRATCH
33 06412 020442         LDA 0,AT.ES       ;CHK FOR PREV ERR
34 06413 101004         MOV 0,0,SZR       ;SKP ON PREV ERR
35 06414 000403         JMP .+3           ;RELEASE ASSIGN
36 06415 040744         STA 0,AT.TK
37 06416 000734         JMP AT.XI
38                      LCALL ARANG       ;IF BITS 0 AND 1=1
39 06417 100270         ARANG=ASCRA*1B11+100010
40 06420 103043         ADDU 0,0,SNC      ;RELEASE AND REMAP
41 06421 103003         ADD 0,0,SNC       ;IF EITHER =0 MAKE
42 06422 000773         JMP .-5           ;NEXT PASS SELECT NEW LP
43              AT.4A:  LCALL RSCRA
44 06423 100050         RSCRA=ASCRA*1B11+100010
45 06424 000724         JMP AT.05         ;SET NO SCRATCH
46 06425 000776         JMP AT.4A
```

```
10121 MNMRT
01                ;ERROR IN TEST DURING EXECUTION
02
03 06426 006427 ATERR:  ATERR+1
04 06427 054424         STA 3,ATS03       ;PRINT ERR HEADERS
05                      LCALL ERROI
06 06430 100350         ERROI=ASCRA+1B11+100010
07 06431 000401         JMP .+1
08 06432 020433 AT.EC:  LDA 0,ATTX1
09                      LCALL ERRTX
10 06433 100170         ERRTX=ASCRA+1B11+100010
11 06434 020422         LDA 0,AT.LC       ;FOLLOW UP WITH
12 06435 024425         LDA 1,AT.LO       ;TEST RELOC INFO
13 06436 030425         LDA 2,AT.LA
14                      LCALL ERROC
15 06437 100370         ERROC=ASCRA+1B11+100010
16 06440 000401         JMP .+1
17                      LCALL ERPAC
18 06441 100250         ERPAC=ASCRA+1B11+100010
19 06442 020411         LDA 0,ATS03       ;CONTINUE TYPE
20 06443 024420         LDA 1,AT.LA
21 06444 030413         LDA 2,AT.BG
22                      LCALL ERPAD
23 06445 100230         ERPAD=ASCRA+1B11+100010
24 06446 000755         JMP AT.4A         ;SW0=1 RELEASE SCR
25 06447 102400         SUB 0,0
26 06450 040404         STA 0,AT.ES       ;SET ERR SW
27 06451 040710         STA 0,AT.TK       ;CLR TEST K
28 06452 000700         JMP AT.XI         ;RETURN TO LINKR
29
```

```
10122 MNMRT
01 06453 000000 ATS03: 0
02 06454 000000 AT.ES:  0
03 06455 001361 AT.PL:  ATEND-MS1
04 06456 000000 AT.LC:  0
05 06457 006514 AT.BG:  MS1
06 06460 000000 AT.EN:  0
07 06461 000037 AT.37:  37
08 06462 000000 AT.LO:  0
09 06463 000000 AT.LA:  0
10 06464 000000 AT.LP:  0
11 06465 006466 ATTX1:  .+1
12 06466 005215         .TXTE (<15><12>AT.LC    AT.LO   AT.LA(
13 06500 006501 ATTX2:  .+1
14 06501 005215         .TXTE (<15><12>ATS03    AT.LP   E ADRS(
```

```
10123 MNMRT
01                MS1:    SETUP           ;MISC TEST OF NEG/COM
02                        LCALL SETUL
03 06514 100410           SETUL=ASCRA*1B11+100010
04                        RANDOM
05                        LCALL FRANG
06 06515 100450           FRANG=ASCRA*1B11+100010
07 06516 104700           NEGS 0,1
08 06517 130342           COMOS 1,2,SZC
09 06520 142014           ADC# 2,0,SZR
10                        ERROR
11 06521 004402           JSR .+2
12 06522 000402           JMP .+2
13                        LCALL ERRET
14 06523 100470           ERRET=ASCRA*1B11+100010
15                        LOOP
16                        LCALL LLOOP
17 06524 100430           LLOOP=ASCRA*1B11+100010
18
19                ;MS2:
20                        SETUP           ;MISC TEST OF INC SWAPPED.
21                        LCALL SETUL
22 06525 100410           SETUL=ASCRA*1B11+100010
23                        RANDOM
24                        LCALL FRANG
25 06526 100450           FRANG=ASCRA*1B11+100010
26 06527 111700           INCS 0,2
27 06530 145323           MOVZS 2,1,SNC
28 06531 106314           ADCS# 0,1,SZR
29                        ERROR
30 06532 004402           JSR .+2
31 06533 000402           JMP .+2
32                        LCALL ERRET
33 06534 100470           ERRET=ASCRA*1B11+100010
34                        LOOP
35                        LCALL LLOOP
36 06535 100430           LLOOP=ASCRA*1B11+100010
37
```

```
10124 MNMRT
01                ;MS3:
02                        SETUP           ;MISC NEG TEST
03                        LCALL SETUL
04 06536 100410           SETUL=ASCRA*1B11+100010
05 06537 102400           SUB 0,0
06 06540 100644           NEGCR 0,0,SZR
07                        ERROR
08 06541 004402           JSR .+2
09 06542 000402           JMP .+2
10                        LCALL ERRET
11 06543 100470           ERRET=ASCRA*1B11+100010
12 06544 100664           NEGCR 0,0,SZR
13                        ERROR
14 06545 004402           JSR .+2
15 06546 000402           JMP .+2
16                        LCALL ERRET
17 06547 100470           ERRET=ASCRA*1B11+100010
18 06550 100664           NEGCR 0,0,SZR
19                        ERROR
20 06551 004402           JSR .+2
21 06552 000402           JMP .+2
22                        LCALL ERRET
23 06553 100470           ERRET=ASCRA*1B11+100010
24                        LOOP
25                        LCALL LLOOP
26 06554 100430           LLOOP=ASCRA*1B11+100010
27
```

```
10125 MNMRT
01                ;ROT1:
02                        SETUP             ;TEST ROTATE LEFT
03                        LCALL SETUL
04 06555 100410           SETUL=ASCRA*1B11+100010
05                        RANDOM
06                        LCALL FRANG
07 06556 100450           FRANG=ASCRA*1B11+100010
08 06557 105000           MOV 0,1
09 06560 101100           MOVL 0,0
10 06561 111100           MOVL 0,2
11 06562 151100           MOVL 2,2
12 06563 151100           MOVL 2,2
13 06564 155100           MOVL 2,3
14 06565 175100           MOVL 3,3
15 06566 175100           MOVL 3,3
16 06567 161100           MOVL 3,0
17 06570 101100           MOVL 0,0
18 06571 101100           MOVL 0,0
19 06572 101100           MOVL 0,0
20 06573 101100           MOVL 0,0
21 06574 101100           MOVL 0,0
22 06575 101100           MOVL 0,0
23 06576 101100           MOVL 0,0
24 06577 101100           MOVL 0,0
25 06600 101100           MOVL 0,0
26 06601 100414           SUB# 0,1,SZR
27                        ERROR
28 06602 004402           JSR .+2
29 06603 000402           JMP .+2
30                        LCALL ERRET
31 06604 100470           ERRET=ASCRA*1B11+100010
32                        LOOP
33                        LCALL LLOOP
34 06605 100430           LLOOP=ASCRA*1B11+100010
```

```
10126 MNMRT
01                ;ROT2:
02                        SETUP             ;TEST ROTATE RIGHT
03                        LCALL SETUL
04 06606 100410           SETUL=ASCRA*1B11+100010
05                        RANDOM
06                        LCALL FRANG
07 06607 100450           FRANG=ASCRA*1B11+100010
08 06610 131000           MOV 1,2
09 06611 121200           MOVR 1,0
10 06612 101200           MOVR 0,0
11 06613 101200           MOVR 0,0
12 06614 101200           MOVR 0,0
13 06615 101200           MOVR 0,0
14 06616 101200           MOVR 0,0
15 06617 101200           MOVR 0,0
16 06620 101200           MOVR 0,0
17 06621 101200           MOVR 0,0
18 06622 101200           MOVR 0,0
19 06623 105200           MOVR 0,1
20 06624 125200           MOVR 1,1
21 06625 135200           MOVR 1,3
22 06626 175200           MOVR 3,3
23 06627 175200           MOVR 3,3
24 06630 161200           MOVR 3,0
25 06631 101200           MOVR 0,0
26 06632 112414           SUB# 0,2,SZR
27                        ERROR
28 06633 004402           JSR .+2
29 06634 000402           JMP .+2
30                        LCALL ERRET
31 06635 100470           ERRET=ASCRA*1B11+100010
32                        LOOP
33                        LCALL LLOOP
34 06636 100430           LLOOP=ASCRA*1B11+100010
```

```
!0127 MNMRT
01                ;AND0:
02                        SETUP           ;ANY NUMBER ANDED WITH
03                        LCALL SETUL
04 06637 100410           SETUL=ASCRA*1B11+100010
05                        RANDOM          ;ITSELF SHOULD NOT
06                        LCALL FRANG
07 06640 100450           FRANG=ASCRA*1B11+100010
08 06641 131000           MOV 1,2         ;BE CHANGED.
09 06642 127700           ANDS 1,1
10 06643 127700           ANDS 1,1
11 06644 127700           ANDS 1,1
12 06645 127700           ANDS 1,1
13 06646 127700           ANDS 1,1
14 06647 135300           MOVS 1,3
15 06650 177700           ANDS 3,3
16 06651 177700           ANDS 3,3
17 06652 161300           MOVS 3,0
18 06653 103700           ANDS 0,0
19 06654 112414           SUB# 0,2,SZR
20                        ERROR
21 06655 004402           JSR .+2
22 06656 000402           JMP .+2
23                        LCALL ERRET
24 06657 100470           ERRET=ASCRA*1B11+100010
25                        LOOP
26                        LCALL LLOUP
27 06660 100430           LLOUP=ASCRA*1B11+100010
28
29                ;AND1:
30                        SETUP           ;A NUMBER ANDED WITH ITS
31                        LCALL SETUL
32 06661 100410           SETUL=ASCRA*1B11+100010
33                        RANDOM          ;COMPLIMENT SHOULD
34                        LCALL FRANG
35 06662 100450           FRANG=ASCRA*1B11+100010
36 06663 104042           COMO 0,1,SZC
37 06664 123704           ANDS 1,0,SZR    ;PRODUCE ZERO RESULT.
38                        ERROR
39 06665 004402           JSR .+2
40 06666 000402           JMP .+2
41                        LCALL ERRET
42 06667 100470           ERRET=ASCRA*1B11+100010
43                        LOOP
44                        LCALL LLOUP
45 06670 100430           LLOUP=ASCRA*1B11+100010
46
```

```
!0128 MNMRT
01                ;AND3:
02                        SETUP           ;PERFORM A AND INSTRUCTION
03                        LCALL SETUL
04 06671 100410           SETUL=ASCRA*1B11+100010
05                        RANDOM          ;WITH THE RESULT IN AC2.
06                        LCALL FRANG
07 06672 100450           FRANG=ASCRA*1B11+100010
08 06673 176620           SUBZR 3,3       ;SIMULATE THE AND VIA
09 06674 131000           MOV 1,2         ;LOOKING FOR ADDER CARRY.
10 06675 113400           AND 0,2
11 06676 101113 AND3L:    MOVL# 0,0,SNC
12 06677 000403           JMP .+3
13 06700 125112           MOVL# 1,1,SZC
14 06701 101141           MOVOL 0,0,SKP
15 06702 101121           MOVZL 0,0,SKP
16 06703 125141           MOVOL 1,1,SKP
17 06704 125120           MOVZL 1,1
18 06705 175224           MOVZR 3,3,SZR
19 06706 000770           JMP AND3L
20 06707 106415           SUB# 0,1,SNR    ;CHECK IF AC0-1 ARE
21 06710 132414           SUB# 1,2,SZR    ;THE SAME AND IF THEY
22                        ERROR           ;ARGEE WITH INST.
23 06711 004402           JSR .+2
24 06712 000402           JMP .+2
25                        LCALL ERRET
26 06713 100470           ERRET=ASCRA*1B11+100010
27                        LOOP
28                        LCALL LLOUP
29 06714 100430           LLOUP=ASCRA*1B11+100010
```

```
10129 MNMRT
01                ;TEST PROCESSOR VIA EXCLUSIVE OR ROUTINES.
02
03                ;X1:
04                        SETUP           ;C(AC1) IS SAVED IN C(AC2).
05                        LCALL SETUL
06 06715 100410           SETUL=ASCRA+1B11+100010
07                        RANDOM          ;AC0 IS EXCLUSIVE ORED
08                        LCALL FRANG
09 06716 100450           FRANG=ASCRA+1B11+100010
10 06717 131000           MOV 1,2         ;WITH AC1 TWICE, THE SECOND
11                        CALL            ;EXCLUSIVE OR SHOULD
12                        XORA            ;RESTORE AC1 TO ITS
13 06720 004467           JSR XOR,0
14                        CALL            ;ORIGIONAL CONTENTS.
15                        XORA
16 06721 004466           JSR XOR,0
17 06722 132414           SUB# 1,2,SZR
18                        ERROR
19 06723 004402           JSR .+2
20 06724 000402           JMP .+2
21                        LCALL ERRET
22 06725 100470           ERRET=ASCRA+1B11+100010
23                        LOOP
24                        LCALL LLOUP
25 06726 100430           LLOUP=ASCRA+1B11+100010
26
27                ;X2:
28                        SETUP           ;THE FIRST EXCLUSIVE OR
29                        LCALL SETUL
30 06727 100410           SETUL=ASCRA+1B11+100010
31                        RANDOM          ;ROUTINE EXCHANGES THE
32                        LCALL FRANG
33 06730 100450           FRANG=ASCRA+1B11+100010
34                        CALL            ;CONTENTS
35                        XOR2            ;OF AC0 AND AC1, IT ALSO
36 06731 004474           JSR XOR,2
37                        CALL            ;FORMS THE EXCLUSIVE OR
38                        XOR1            ;IN AC2. THE SECOND EXCLUSIVE
39 06732 004463           JSR XOR,1
40 06733 112414           SUB# 0,2,SZR    ;OR FORMS THE RESULT
41                        ERROR           ;OF AC0-AC1 IN AC0.
42 06734 004402           JSR .+2
43 06735 000402           JMP .+2
44                        LCALL ERRET
45 06736 100470           ERRET=ASCRA+1B11+100010
46                        LOOP
47                        LCALL LLOUP
48 06737 100430           LLOUP=ASCRA+1B11+100010
```

```
10130 MNMRT
01                ;X3:
02                        SETUP           ;SAVE C(AC1) NEGATED IN
03                        LCALL SETUL
04 06740 100410           SETUL=ASCRA+1B11+100010
05                        RANDOM          ;C(AC2). EXCLUSIVE OR C(AC0) TO
06                        LCALL FRANG
07 06741 100450           FRANG=ASCRA+1B11+100010
08 06742 130400           NEG 1,2         ;C(AC1). EXCLUSIVE OR THE
09                        CALL            ;RESULT BACK TO AC0.
10                        XORA            ;CHECK VIA ADDITION TO
11 06743 004444           JSR XOR,0
12                        CALL            ;COMPLIMENT OF ORIGIONAL
13                        XOR1            ;NUMBER.
14 06744 004451           JSR XOR,1
15 06745 113014           AUD# 0,2,SZR
16                        ERROR
17 06746 004402           JSR .+2
18 06747 000402           JMP .+2
19                        LCALL ERRET
20 06750 100470           ERRET=ASCRA+1B11+100010
21                        LOOP
22                        LCALL LLOUP
23 06751 100430           LLOUP=ASCRA+1B11+100010
24
25                ;X4:
26                        SETUP           ;EXCLUSIVE OR C(AC0) TO
27                        LCALL SETUL
28 06752 100410           SETUL=ASCRA+1B11+100010
29                        RANDOM          ;ALL ZEROS IN C(AC1).
30                        LCALL FRANG
31 06753 100450           FRANG=ASCRA+1B11+100010
32 06754 126400           SUB 1,1
33                        CALL
34                        XORA
35 06755 004432           JSR XOR,0
36 06756 106414           SUB# 0,1,SZR
37                        ERROR
38 06757 004402           JSR .+2
39 06760 000402           JMP .+2
40                        LCALL ERRET
41 06761 100470           ERRET=ASCRA+1B11+100010
42                        LOOP
43                        LCALL LLOUP
44 06762 100430           LLOUP=ASCRA+1B11+100010
```

```
10131 MNMRT
01                      ;X5:
02                              SETUP           ;EXCLUSIVE OR C(AC1) TO
03                              LCALL SETUL
04 06765 100410                 SETUL=ASCRA*1B11+100010
05                              RANDOM          ;ALL ZEROS IN C(AC0).
06                              LCALL FRANG
07 06764 100450                 FRANG=ASCRA*1B11+100010
08 06765 102400                 SUB 0,0
09                              CALL
10                              XOR1
11 06766 004427                 JSR XOR,1
12 06767 106414                 SUB# 0,1,SZR
13                              ERROR
14 06770 004402                 JSR .+2
15 06771 004402                 JMP .+2
16                              LCALL ERRET
17 06772 100470                 ERRET=ASCRA*1B11+100010
18                              LOOP
19                              LCALL LLOOP
20 06773 100430                 LLOOP=ASCRA*1B11+100010
21
22                      ;X6:
23                              SETUP           ;EXCLUSIVE OR C(AC1) TO
24                              LCALL SETUL
25 06774 100410                 SETUL=ASCRA*1B11+100010
26                              RANDOM          ;(-1) IN C(AC0).
27                              LCALL FRANG
28 06775 100450                 FRANG=ASCRA*1B11+100010
29 06776 102000                 ADC 0,0         ;THE COMPLIMENT OF
30                              CALL            ;THIS RESULT SHOULD
31                              XOR1            ;EQUAL C(AC1).
32 06777 004416                 JSR XOR,1
33 07000 110000                 COM 0,2
34 07001 146414                 SUB# 2,1,SZR
35                              ERROR
36 07002 004402                 JSR .+2
37 07003 004402                 JMP .+2
38                              LCALL ERRET
39 07004 100470                 ERRET=ASCRA*1B11+100010
40                              LOOP
41                              LCALL LLOOP
42 07005 100430                 LLOOP=ASCRA*1B11+100010
43 07006 004436                 JMP X7
```

```
10132 MNMRT
01 07007 054433 XOR,0:  STA 3,XOR,4
02 07010 135000         MOV 1,3         ;EXCLUSIVE OR C(AC0),C(AC1).
03 07011 117520         ANDZL 0,3       ;RESULT IS IN C(AC1).
04 07012 107000         ADD 0,1
05 07013 166400         SUB 3,1
06 07014 002426         JMP @XOR,4
07
08 07015 054425 XOR,1:  STA 3,XOR,4
09 07016 135000         MOV 1,3         ;EXCLUSIVE OR C(AC0),C(AC1).
10 07017 117400         AND 0,3         ;RESULT IS IN C(AC0).
11 07020 174000         COM 3,3
12 07021 163400         AND 3,0
13 07022 123000         ADD 1,0
14 07023 163400         AND 3,0
15 07024 002416         JMP @XOR,4
16
17 07025 054415 XOR,2:  STA 3,XOR,4
18 07026 034415         LDA 3,XM20      ;EXCLUSIVE OR C(AC0),C(AC1).
19 07027 054412         STA 3,XORTEM    ;RESULT IN C(AC2).
20 07030 115000         MOV 0,3         ;THE CONTENTS OF AC0 AND
21 07031 137200         ADDR 1,3        ;AC1 ARE EXCHANGED.
22 07032 151200         MOVR 2,2
23 07033 101220         MOVZR 0,0
24 07034 125200         MOVR 1,1
25 07035 103200         ADDR 0,0
26 07036 010403         ISZ XORTEM
27 07037 000771         JMP XOR,2+3
28 07040 002402         JMP @XOR,4
29 07041 000000 XORTEM: 0
30 07042 000000 XOR,4: 0
31 07043 177760 XM20: -20
```

```
10133 MNMRT
01                X7:       SETUP             ;C(AC1) IS SET EQUAL TO
02                          LCALL SETUL
03 07044 100410             SETUL=ASCRA*1B11+100010
04                          RANDOM            ;C(AC0), C(AC0) AND C(AC1)
05                          LCALL FRANG
06 07045 100450             FRANG=ASCRA*1B11+100010
07 07046 105000             MOV 0,1           ;ARE EXCLUSIVE ORED WITH
08                          CALL              ;THE RESULT GOING TO AC2.
09                          XOR2
10 07047 004756             JSR XOR,2
11 07050 151004             MOV 2,2,SZR
12                          ERROR
13 07051 004402             JSR .+2
14 07052 000402             JMP .+2
15                          LCALL ERRET
16 07053 100470             ERRET=ASCRA*1B11+100010
17                          LOOP
18                          LCALL LLOOP
19 07054 100430             LLOOP=ASCRA*1B11+100010
20
21                ;X8:
22                          SETUP             ;C(AC0) IS SET TO THE
23                          LCALL SETUL
24 07055 100410             SETUL=ASCRA*1B11+100010
25                          RANDOM            ;COMPLIMENT OF C(AC1). THE
26                          LCALL FRANG
27 07056 100450             FRANG=ASCRA*1B11+100010
28 07057 120000             COM 1,0           ;RESULT OF A EXCLUSIVE OR
29                          CALL              ;SHOULD BE ALL BITS
30                          XOR2              ;SET (-1).
31 07060 004745             JSR XOR,2
32 07061 150014             COM# 2,2,SZR
33                          ERROR
34 07062 004402             JSR .+2
35 07063 000402             JMP .+2
36                          LCALL ERRET
37 07064 100470             ERRET=ASCRA*1B11+100010
38                          LOOP
39                          LCALL LLOOP
40 07065 100430             LLOOP=ASCRA*1B11+100010
```

```
10134 MNMRT
01                ;X9:
02                          SETUP             ;EXCLUSIVE OR ALL ONES
03                          LCALL SETUL
04 07066 100410             SETUL=ASCRA*1B11+100010
05 07067 102300             ADCS 0,0          ;TO ALL ONES. THE
06 07070 126000             ADC 1,1           ;RESULT SHOULD BE
07                          CALL              ;ALL ZEROS.
08                          XORA
09 07071 004716             JSR XOR,0
10 07072 127704             ANDS 1,1,SZR
11                          ERROR
12 07073 004402             JSR .+2
13 07074 000402             JMP .+2
14                          LCALL ERRET
15 07075 100470             ERRET=ASCRA*1B11+100010
16                          LOOP
17                          LCALL _LOOP
18 07076 100430             LLOOP=ASCRA*1B11+100010
19
20                ;X10:
21                          SETUP             ;EXCLUSIVE OR ALL ZEROS
22                          LCALL SETUL
23 07077 100410             SETUL=ASCRA*1B11+100010
24 07100 102700             SUBS 0,0          ;TO ALL ZEROS. THE
25 07101 126400             SUB 1,1           ;RESULT SHOULD BE
26                          CALL              ;ALL ZEROS IN C(AC1).
27                          XORA
28 07102 004705             JSR XOR,0
29 07103 107314             ANDS# 0,1,SZR
30                          ERROR
31 07104 004402             JSR .+2
32 07105 000402             JMP .+2
33                          LCALL ERRET
34 07106 100470             ERRET=ASCRA*1B11+100010
35                          LOOP
36                          LCALL LLOOP
37 07107 100430             LLOOP=ASCRA*1B11+100010
```

```
10135 MNMRT
01                ;X11:
02                        SETUP           ;FORM EXCLUSIVE OF FUNCTION
03                        LCALL SETUL
04 07110 100410           SETUL=ASCRA*1B11+100010
05                        RANDOM          ;IN C(AC2).
06                        LCALL FRANG
07 07111 100450           FRANG=ASCRA*1B11+100010
08                        CALL            ;CALL ANOTHER EXCLUSIVE
09                        XOR2            ;OR FUNCTION SEVEN TIMES.
10 07112 004713           JSR XOR,2
11                        CALL            ;THE RESULT SHOULD BE THE
12                        XORA            ;SAME AS THE FIRST XORA
13 07113 004674           JSR XOR,0
14                        CALL
15                        XORA
16 07114 004673           JSR XOR,0
17                        CALL
18                        XORA
19 07115 004672           JSR XOR,0
20                        CALL
21                        XORA
22 07116 004671           JSR XOR,0
23                        CALL
24                        XORA
25 07117 004670           JSR XOR,0
26                        CALL
27                        XORA
28 07120 004667           JSR XOR,0
29                        CALL
30                        XORA
31 07121 004666           JSR XOR,0
32 07122 132414           SUB# 1,2,SZR
33                        ERROR
34 07123 004402           JSR .+2
35 07124 000402           JMP .+2
36                        LCALL ERRET
37 07125 100470           ERRET=ASCRA*1B11+100010
38                        LOOP
39                        LCALL LLOOP
40 07126 100430           LLOOP=ASCRA*1B11+100010
41
42 07127 000402           JMP A1
43 07130 000657 XOR,L: JMP XOR,0
```

```
10136 MNMRT
01                A1:     SETUP           ;SAVE C(AC2) IN C(AC1).
02                        LCALL SETUL
03 07131 100410           SETUL=ASCRA*1B11+100010
04                        RANDOM          ;ADD AND ADD C(AC0) TO
05                        LCALL FRANG
06 07132 100450           FRANG=ASCRA*1B11+100010
07 07133 145000           MOV 2,1         ;C(AC2). THE VALUE IN
08 07134 112400           SUB 0,2         ;AC2 SHOULD NOT BE
09 07135 113000           ADD 0,2         ;CHANGED.
10 07136 146414           SUB# 2,1,SZR
11                        ERROR
12 07137 004402           JSR .+2
13 07140 000402           JMP .+2
14                        LCALL ERRET
15 07141 100470           ERRET=ASCRA*1B11+100010
16                        LOOP
17                        LCALL LLOOP
18 07142 100430           LLOOP=ASCRA*1B11+100010
19
20                ;A2:
21                        SETUP
22                        LCALL SETUL
23 07143 100410           SETUL=ASCRA*1B11+100010
24                        RANDOM
25                        LCALL FRANG
26 07144 100450           FRANG=ASCRA*1B11+100010
27 07145 102000           ADC 0,0
28 07146 123000           ADD 1,0
29 07147 111400           INC 0,2
30 07150 146414           SUB# 2,1,SZR
31                        ERROR
32 07151 004402           JSR .+2
33 07152 000402           JMP .+2
34                        LCALL ERRET
35 07153 100470           ERRET=ASCRA*1B11+100010
36                        LOOP
37                        LCALL LLOOP
38 07154 100430           LLOOP=ASCRA*1B11+100010
```

```
10137 MNMRT
01                      ;A3:
02                              SETUP           ;INCREMENT THE VALUE IN
03                              LCALL SETUL
04 07155 100410                 SETUL=ASCRA*1B11+100010
05                              RANDOM          ;AC1 AND ADD THAT VALUE
06                              LCALL FRANG
07 07156 100450                 FRANG=ASCRA*1B11+100010
08 07157 131400                 INCC 1,2        ;TO (-1). THE RESULT
09 07160 102360                 ADCCS 0,0       ;SHOULD BE THE ORIGIONAL
10 07161 143060                 ADDC 2,0        ;NUMBER.
11 07162 106414                 SUB# 0,1,SZR
12                              ERROR
13 07163 004402                 JSR .+2
14 07164 000402                 JMP .+2
15                              LCALL ERRET
16 07165 100470                 ERRET=ASCRA*1B11+100010
17                              LOOP
18                              LCALL LLOOP
19 07166 100430                 LLOOP=ASCRA*1B11+100010
20                      ;A4:
21                              SETUP           ;SAVE THE C(AC1) IN C(AC0).
22                              LCALL SETUL
23 07167 100410                 SETUL=ASCRA*1B11+100010
24                              RANDOM          ;A "ADDR" INSTRUCTION SHOULD
25                              LCALL FRANG
26 07170 100450                 FRANG=ASCRA*1B11+100010
27 07171 121020                 MOVZ 1,0        ;NOT CHANGE THE VALUE OF
28 07172 127200                 ADDR 1,1        ;THE AC.
29 07173 127200                 ADDR 1,1
30 07174 127200                 ADDR 1,1
31 07175 127200                 ADDR 1,1
32 07176 127200                 ADDR 1,1
33 07177 127200                 ADDR 1,1
34 07200 127200                 ADDR 1,1
35 07201 127200                 ADDR 1,1
36 07202 127200                 ADDR 1,1
37 07203 127200                 ADDR 1,1
38 07204 127200                 ADDR 1,1
39 07205 127200                 ADDR 1,1
40 07206 127200                 ADDR 1,1
41 07207 127200                 ADDR 1,1
42 07210 106454                 SUBO# 0,1,SZR
43                              ERROR
44 07211 004402                 JSR .+2
45 07212 000402                 JMP .+2
46                              LCALL ERRET
47 07213 100470                 ERRET=ASCRA*1B11+100010
48                              LOOP
49                              LCALL LLOOP
50 07214 100430                 LLOOP=ASCRA*1B11+100010
```

```
10138 MNMRT
01                      ;A5:
02                              SETUP           ;THE RANDOM NUMBER IN
03                              LCALL SETUL
04 07215 100410                 SETUL=ASCRA*1B11+100010
05                              RANDOM          ;C(AC0) IS INCREMENTED VIA "INC"
06                              LCALL FRANG
07 07216 100450                 FRANG=ASCRA*1B11+100010
08 07217 115400                 INC 0,3         ;AND DECREMENTED VIA "ADD".
09 07220 120000                 ADC 1,1         ;THE FINAL RESULT IN C(AC2)
10 07221 137000                 ADD 1,3         ;SHOULD BE EQUAL TO THE
11 07222 175400                 INC 3,3         ;ORIGIONAL NUMBER IN C(AC0).
12 07223 137000                 ADD 1,3
13 07224 171400                 INC 3,2
14 07225 133000                 ADD 1,2
15 07226 151400                 INC 2,2
16 07227 133000                 ADD 1,2
17 07230 142414                 SUB# 2,0,SZR
18                              ERROR
19 07231 004402                 JSR .+2
20 07232 000402                 JMP .+2
21                              LCALL ERRET
22 07233 100470                 ERRET=ASCRA*1B11+100010
23                              LOOP
24                              LCALL LLOOP
25 07234 100430                 LLOOP=ASCRA*1B11+100010
```

```
10139 MNMRT
01                      ;A6:
02                              SETUP           ;THE SUM OF AC0-1 IS
03                              LCALL SETUL
04 07235 100410                 SETUL=ASCRA*1B11+100010
05                              RANDOM          ;CHECKED WITH THE SIMULATED
06                              LCALL FRANG
07 07236 100450                 FRANG=ASCRA*1B11+100010
08 07237 135000                 MOV 1,3         ;SUM.
09 07240 117000                 ADD 0,3
10 07241 054424                 STA 3,ADDTEM    ;SUM VIA ADD INSTRUCTION.
11 07242 131000 A6L:    MOV 1,2
12                              CALL
13 07243 004665                 JSR XOR,L               ;SIMULATE THE ADD VIA
14 07244 143524                 ANDZL 2,0,SZR   ;EXCLUSIVE OR, C(AC2)=
15 07245 000775                 JMP A6L         ;RIPPLE CARRY,C(AC1)=RESULT.
16 07246 020417                 LDA 0,ADDTEM
17 07247 122414                 SUB# 1,0,SZR
18                              ERROR
19 07250 004402                 JSR .+2
20 07251 000402                 JMP .+2
21                              LCALL ERRET
22 07252 100470                 ERRET=ASCRA*1B11+100010
23                              LOOP
24                              LCALL LLOOP
25 07253 100430                 LLOOP=ASCRA*1B11+100010
26
27                      ;A7:
28                              SETUP           ;ADDITION OF NEGATED
29                              LCALL SETUL
30 07254 100410                 SETUL=ASCRA*1B11+100010
31                              RANDOM          ;NUMBERS SHOULD PRODUCE
32                              LCALL FRANG
33 07255 100450                 FRANG=ASCRA*1B11+100010
34 07256 110440                 NEGO 0,2        ;ZERO AND A CARRY.
35 07257 143204                 ADDR 2,0,SZR
36                              ERROR
37 07260 004402                 JSR .+2
38 07261 000402                 JMP .+2
39                              LCALL ERRET
40 07262 100470                 ERRET=ASCRA*1B11+100010
41                              LOOP
42                              LCALL LLOOP
43 07263 100430                 LLOOP=ASCRA*1B11+100010
44
45 07264 101001                 MOV 0,0,SKP
46 07265 177777 ADDTEM: -1
```

```
10140 MNMRT
01                      ;A8:
02                              SETUP           ;ADD TEST.
03                              LCALL SETUL
04 07266 100410                 SETUL=ASCRA*1B11+100010
05                              RANDOM
06                              LCALL FRANG
07 07267 100450                 FRANG=ASCRA*1B11+100010
08 07270 152520                 SUBZL 2,2       ;=(+1)
09 07271 151300                 MOVS 2,2        ;=(400)
10 07272 150400                 NEG 2,2         ;=(177400)
11 07273 143700                 ANDS 2,0        ;SAVE HIGH ORDER 8 BITS.
12 07274 105320                 MOVZS 0,1       ;SAME 8 BITS TO C(AC1)L.
13 07275 103100                 ADDL 0,0        ;MOVE C(AC0) LEFT VIA
14 07276 103100                 ADDL 0,0        ;ADD SHIFT.
15 07277 103100                 ADDL 0,0
16 07300 103100                 ADDL 0,0
17 07301 100414                 SUB# 0,1,SZR
18                              ERROR
19 07302 004402                 JSR .+2
20 07303 000402                 JMP .+2
21                              LCALL ERRET
22 07304 100470                 ERRET=ASCRA*1B11+100010
23                              LOOP
24                              LCALL LLOOP
25 07305 100430                 LLOOP=ASCRA*1B11+100010
26
```

```
10141 MNMRT
01 07306 101001          MOV 0,0,SKP
02 07307 17/400 M400:    -400
03              ;AR1:
04                       SETUP           ;THE ORIGIONAL CONTENTS OF
05                       LCALL SETUL
06 07310 100410          SETUL=ASCRA*1B11+100010
07                       RANDOM          ;AC1, BITS 0-7 ARE SQUARED
08                       LCALL FRANG
09 07311 100450          FRANG=ASCRA*1B11+100010
10 07312 030775          LDA 2,M400      ;VIA MULTIPLY. THE SQUARE
11 07313 133700          ANDS 1,2        ;ROOT OF THE PRODUCT SHOULD
12 07314 145000          MOV 2,1         ;EQUAL THE ORIGIONAL.
13                       CALL
14                       .MPYU           ;SEE SYSTEM REFFERANCE
15 07315 004442          JSR MPYU
16 07316 121000          MOV 1,0         ;MANUAL FOR FURTHER INFORMATION
17                       CALL            ;ON MULTIPLY/SQ ROOT
18                       SQRT            ;PROGRAMS.
19 07317 004467          JSR SQRT.
20 07320 112414          SUB# 0,2,SZR
21                       ERROR
22 07321 004402          JSR .+2
23 07322 000402          JMP .+2
24                       LCALL ERRET
25 07323 100470          ERRET=ASCRA*1B11+100010
26                       LOOP
27                       LCALL LLOOP
28 07324 100430          LLOOP=ASCRA*1B11+100010
```

```
10142 MNMRT
01              ;AR2:
02                       SETUP           ;TAKE THE SQUARE ROOT
03                       LCALL SETUL
04 07325 100410          SETUL=ASCRA*1B11+100010
05                       RANDOM          ;OF A NUMBER. THE SQUARE
06                       LCALL FRANG
07 07326 100450          FRANG=ASCRA*1B11+100010
08                       CALL            ;ROOT OF THE RESULT SQUARED
09                       SQRT            ;SHOULD BE THE SAME AS THE
10 07327 004457          JSR SQRT.
11 07330 105060          MOVC 0,1        ;ORIGIONAL ROOT.
12 07331 131060          MOVC 1,2
13                       CALL
14                       .MPYU
15 07332 004425          JSR MPYU
16 07333 121000          MOV 1,0
17                       CALL            ;SQUARED NOW TAKE ROOT.
18                       SQ
19 07334 004462          JSR SQ.
20 07335 112714          SUBS# 0,2,SZR
21                       ERROR
22 07336 004402          JSR .+2
23 07337 000402          JMP .+2
24                       LCALL ERRET
25 07340 100470          ERRET=ASCRA*1B11+100010
26                       LOOP
27                       LCALL LLOOP
28 07341 100430          LLOOP=ASCRA*1B11+100010
```

```
!0143 MNMRT
01                      !AR3:
02                              SETUP           !FIND SQUARE ROOT VIA
03                              LCALL SETUL
04 07342 100410                 SETUL=ASCHA*1B11+100010
05                              RANDOM          !DIFFIRENT SUBROUTINES.
06                              LCALL FRANG
07 07343 100450                 FRANG=ASCHA*1B11+100010
08 07344 141000                 MOV 2,0
09                              CALL
10                              SQRT
11 07345 004441                 JSR SQRT.
12 07346 105000                 MOV 0,1         !SAVE FIRST RESULT IN AC1
13 07347 141000                 MOV 2,0
14                              CALL
15                              SQ              !RESULT IN AC0.
16 07350 004446                 JSR SQ.
17 07351 106714                 SUBS# 0,1,SZR
18                              ERROR
19 07352 004402                 JSR .+2
20 07353 000402                 JMP .+2
21                              LCALL ERRET
22 07354 100470                 ERRET=ASCHA*1B11+100010
23                              LOOP
24                              LCALL LLOOP
25 07355 100430                 LLOOP=ASCHA*1B11+100010
26 07356 000455                 JMP AR4
```

```
!0144 MNMRT
01 07357 102460 MPYU:   SUBC 0,0        !C(AC1)*C(AC2)
02 07360 054411 MPYA:   STA 3,.CB03     !RESULT IN AC0,AC1.
03 07361 034411         LDA 3,.CB20     !SEE SYSTEM REFFERANCE
04 07362 125203 .CB99:  MOVR 1,1,SNC    !MANUAL FOR FURTHER
05 07363 101201         MOVR 0,0,SKP    !INFORMATION.
06 07364 143220         ADDZR 2,0
07 07365 175404         INC 3,3,SZR
08 07366 000774         JMP .CB99
09 07367 125200         MOVLR 1,1
10 07370 002401         JMP @.CB03
11 07371 000000 .CB03:  0
12 07372 177760 .CB20:  -20
13
14 07373 102400 DIVI:   SUB 0,0         !C(AC0),C(AC1)/C(AC2).
15 07374 054775 DIVU:   STA 3,.CB03     !AC0=REMAINDER
16 07375 034775         LDA 3,.CB20     !AC1=QUOTIENT
17 07376 125120         MOVZL 1,1       !SEE SYSTEM REFFERANCE
18 07377 101100 .CC98:  MOVL 0,0        !MANUAL.
19 07400 142412         SUB# 2,0,SZC
20 07401 142400         SUB 2,0
21 07402 125100         MOVL 1,1
22 07403 175404         INC 3,3,SZR
23 07404 000773         JMP .CC98
24 07405 002764         JMP @.CB03
25
```

```
10145 MNMRT
01 07406 054423 SQRT.:  STA 3,SQ.S3
02 07407 126520         SUBZL 1,1       ;FIND SQ ROOT OF C(AC0).
03 07410 135120         MOVZL 1,3       ;SEE THE SYSTEM REFFERANCE
04 07411 122422         SUBZ 1,0,SZC    ;MANUAL.
05 07412 167001         ADD 3,1,SKP     ;RESULT IN AC0.
06 07413 121221         MOVZR 1,0,SKP   ;AC1 DESTROYED.
07 07414 000775         JMP .-3
08 07415 002414         JMP @SQ.S3
09
10 07416 054413 SQ.:    STA 3,SQ.S3
11 07417 176400         SUB 3,3         ;FIND SQ ROOT OF C(AC0).
12 07420 054412         STA 3,SQTEM     ;SAME RESULT AS PREVIOUS
13 07421 162023         ADCZ 3,0,SNC    ;TEST BUT CODE IS
14 07422 000405         JMP SQ1         ;DIFFIRENT.
15 07423 010407         ISZ SQTEM
16 07424 010406         ISZ SQTEM
17 07425 034405         LDA 3,SQTEM     ;RESULT WILL BE IN AC0.
18 07426 000773         JMP .-5
19 07427 161200 SQ1:    MOVR 3,0
20 07430 002401         JMP @SQ.S3
21
22 07431 000000 SQ.S3: 0
23 07432 000000 SQTEM:  0
```

```
10146 MNMRT
01              AR4:    SETUP           ;THE CONTENTS OF AC2 IS
02                      LCALL SETUL
03 07433 100410         SETUL=ASCRA*1B11+100010
04              AR4L:   RANDOM          ;DIVIDED INTO AC0-1.
05                      LCALL FRANG
06 07434 100450         FRANG=ASCRA*1B11+100010
07 07435 040471         STA 0,DAC0
08 07436 044471         STA 1,DAC1
09 07437 050471         STA 2,DAC2
10 07440 142432         SUBZ# 2,0,SZC   ;THIS RESULT MULTIPLIED
11 07441 000773         JMP AR4L                ;BY AC2 SHOULD PRODUCE
12                      CALL            ;THE ORIGIONAL NUMBERS.
13                      .DIVU
14 07442 004732         JSR DIVU
15                      CALL
16                      .MPYA
17 07443 004715         JSR MPYA
18 07444 034464         LDA 3,DAC2
19 07445 156414         SUB# 2,3,SZR
20                      ERROR           ;AC2 CHANGED?
21 07446 004402         JSR .+2
22 07447 000402         JMP .+2
23                      LCALL ERRET
24 07450 100470         ERRET=ASCRA*1B11+100010
25 07451 034456         LDA 3,DAC1
26 07452 136714         SUBS# 1,3,SZR
27                      ERROR           ;AC1 CHANGED.
28 07453 004402         JSR .+2
29 07454 000402         JMP .+2
30                      LCALL ERRET
31 07455 100470         ERRET=ASCRA*1B11+100010
32 07456 034450         LDA 3,DAC0
33 07457 116714         SUBS# 0,3,SZR
34                      ERROR           ;AC0 CHANGED.
35 07460 004402         JSR .+2
36 07461 000402         JMP .+2
37                      LCALL ERRET
38 07462 100470         ERRET=ASCRA*1B11+100010
39                      LOOP
40                      LCALL LLOOP
41 07463 100430         LLOOP=ASCRA*1B11+100010
```

```
10147 MNMRT
01                      ;AR5:
02                              SETUP           ;USE INC TO FORM
03                              LCALL SETUL
04 07464 100410                 SETUL=ASCRA*1B11+100010
05 07465 152440                 SUBO 2,2        ;THE NUMBER 177400
06 07466 151504                 INCL 2,2,SZR    ;IN AC1 AND 400 IN AC2.
07 07467 151504                 INCL 2,2,SZR
08 07470 151504                 INCL 2,2,SZR
09 07471 151504                 INCL 2,2,SZR
10 07472 151507                 INCL 2,2,SBN    ;THE "SBN/SZR" SHOULD
11 07473 151507                 INCL 2,2,SBN    ;NOT CAUSE A SKIP.
12 07474 151507                 INCL 2,2,SBN
13 07475 145707                 INCS 2,1,SBN
14 07476 151407                 INC 2,2,SBN
15 07477 151400                 INC 2,2
16 07500 133014                 ADD# 1,2,SZR
17                              ERROR
18 07501 004402                 JSR .+2
19 07502 000402                 JMP .+2
20                              LCALL ERRET
21 07503 100470                 ERRET=ASCRA*1B11+100010
22                              LOOP
23                              LCALL LLOOP
24 07504 100430                 LLOOP=ASCRA*1B11+100010
```

```
10148 MNMRT
01                      ;AR6:
02                              SETUP           ;USE THE INCR INSTRUCTION
03                              LCALL SETUL
04 07505 100410                 SETUL=ASCRA*1B11+100010
05 07506 126420                 SUBZ 1,1        ;TO FORM THE NUMBER
06 07507 125606                 INCR 1,1,SEZ    ;177400, THIS NUMBER
07 07510 125606                 INCR 1,1,SEZ    ;COMPLIMENTED AND SWAPPED
08 07511 125606                 INCR 1,1,SEZ    ;SHOULD BE THE SAME
09 07512 125606                 INCR 1,1,SEZ    ;NUMBER.
10 07513 125606                 INCR 1,1,SEZ    ;THE "SEZ" SHOULD NEVER
11 07514 125606                 INCR 1,1,SEZ    ;CAUSE A SKIP.
12 07515 125606                 INCR 1,1,SEZ
13 07516 125606                 INCR 1,1,SEZ
14 07517 130304                 COMS 1,2,SZR
15 07520 132714                 SUBS# 1,2,SZR
16                              ERROR
17 07521 004402                 JSR .+2
18 07522 000402                 JMP .+2
19                              LCALL ERRET
20 07523 100470                 ERRET=ASCRA*1B11+100010
21                              LOOP
22                              LCALL LLOOP
23 07524 100430                 LLOOP=ASCRA*1B11+100010
24 07525 000405                 JMP .+5
25 07526 000000 OAC0: 0
26 07527 000000 OAC1: 0
27 07530 000000 OAC2: 0
28 07531 000627 MPYAL:  JMP MPYA
```

```
0149 MNMRT
01                      ;AR7:
02                              SETUP           ;100000 NEGATED IS
03                              LCALL SETUL
04 07532 100410                 SETUL=ASCRA*1B11+100010
05 07533 102625                 SUBZR 0,0,SNR   ;STILL 100000.
06                              ERROR
07 07534 004402                 JSR .+2
08 07535 000402                 JMP .+2
09                              LCALL ERRET
10 07536 100470                 ERRET=ASCRA*1B11+100010
11 07537 104406                 NEG 0,1,SEZ     ;C(CARRY)=0
12                              ERROR
13 07540 004402                 JSR .+2
14 07541 000402                 JMP .+2
15                              LCALL ERRET
16 07542 100470                 ERRET=ASCRA*1B11+100010
17 07543 124402                 NEG 1,1,SZC
18                              ERROR
19 07544 004402                 JSR .+2
20 07545 000402                 JMP .+2
21                              LCALL ERRET
22 07546 100470                 ERRET=ASCRA*1B11+100010
23 07547 124463                 NEGC 1,1,SNC    ;C(CARRY)=1
24                              ERROR
25 07550 004402                 JSR .+2
26 07551 000402                 JMP .+2
27                              LCALL ERRET
28 07552 100470                 ERRET=ASCRA*1B11+100010
29 07553 124407                 NEG 1,1,SBN
30                              ERROR
31 07554 004402                 JSR .+2
32 07555 000402                 JMP .+2
33                              LCALL ERRET
34 07556 100470                 ERRET=ASCRA*1B11+100010
35 07557 124465                 NEGC 1,1,SNK    ;C(1)=100000
36                              ERROR
37 07560 004402                 JSR .+2
38 07561 000402                 JMP .+2
39                              LCALL ERRET
40 07562 100470                 ERRET=ASCRA*1B11+100010
41 07563 124465                 NEGC 1,1,SNK
42                              ERROR
43 07564 004402                 JSR .+2
44 07565 000402                 JMP .+2
45                              LCALL ERRET
46 07566 100470                 ERRET=ASCRA*1B11+100010
47 07567 124407                 NEG 1,1,SBN
48                              ERROR
49 07570 004402                 JSR .+2
50 07571 000402                 JMP .+2
51                              LCALL ERRET
52 07572 100470                 ERRET=ASCRA*1B11+100010
53 07573 124407                 NEG 1,1,SBN
54                              ERROR
55 07574 004402                 JSR .+2
56 07575 000402                 JMP .+2
57                              LCALL ERRET
58 07576 100470                 ERRET=ASCRA*1B11+100010
59 07577 106414                 SUB# 0,1,SZK
60                              ERROR
```

```
0150 MNMRT
01 07600 004402                 JSR .+2
02 07601 000402                 JMP .+2
03                              LCALL ERRET
04 07602 100470                 ERRET=ASCRA*1B11+100010
05                              LOOP
06                              LCALL LLOOP
07 07603 100430                 LLOOP=ASCRA*1B11+100010
08
```

```
10151 MNMRT
01                      ;AR8:
02                              SETUP           ;NEGATION OF ZERO
03                              LCALL SETUL
04 07604 100410                 SETUL=ASCRA+1B11+100010
05 07605 102440                 SUBO 0,0        ;SHOULD PRODUCE ZERO
06 07606 100702                 NEGCS 0,0,SZC   ;AND A CARRY.
07                              ERROR
08 07607 004402                 JSR .+2
09 07610 000402                 JMP .+2
10                              LCALL ERRET
11 07611 100470                 ERRET=ASCRA+1B11+100010
12 07612 100702                 NEGCS 0,0,SZC
13                              ERROR
14 07613 004402                 JSR .+2
15 07614 000402                 JMP .+2
16                              LCALL ERRET
17 07615 100470                 ERRET=ASCRA+1B11+100010
18 07616 100706                 NEGS 0,0,SEZ
19                              ERROR
20 07617 004402                 JSR .+2
21 07620 000402                 JMP .+2
22                              LCALL ERRET
23 07621 100470                 ERRET=ASCRA+1B11+100010
24 07622 100706                 NEGS 0,0,SEZ
25                              ERROR
26 07623 004402                 JSR .+2
27 07624 000402                 JMP .+2
28                              LCALL ERRET
29 07625 100470                 ERRET=ASCRA+1B11+100010
30 07626 100544                 NEGOL 0,0,SZR
31                              ERROR
32 07627 004402                 JSR .+2
33 07630 000402                 JMP .+2
34                              LCALL ERRET
35 07631 100470                 ERRET=ASCRA+1B11+100010
36 07632 100403                 NEG 0,0,SNC
37                              ERROR
38 07633 004402                 JSR .+2
39 07634 000402                 JMP .+2
40                              LCALL ERRET
41 07635 100470                 ERRET=ASCRA+1B11+100010
42 07636 100644                 NEGOR 0,0,SZR
43                              ERROR
44 07637 004402                 JSR .+2
45 07640 000402                 JMP .+2
46                              LCALL ERRET
47 07641 100470                 ERRET=ASCRA+1B11+100010
48 07642 104704                 NEGS 0,1,SZR
49                              ERROR
50 07643 004402                 JSR .+2
51 07644 000402                 JMP .+2
52                              LCALL ERRET
53 07645 100470                 ERRET=ASCRA+1B11+100010
54 07646 130704                 NEGS 1,2,SZR
55                              ERROR
56 07647 004402                 JSR .+2
57 07650 000402                 JMP .+2
58                              LCALL ERRET
59 07651 100470                 ERRET=ASCRA+1B11+100010
60                              LOOP
```

```
 0152 MNMRT
01                              LCALL LLOOP
02 07652 100430                 LLOOP=ASCRA+1B11+100010
```

```
;0153 MNMRT
01                ;AR9:
02                        SETUP           ;C(AC1)*0+C(AC0) SHOULD
03                        LCALL SETUL
04 07653 100410           SETUL=ASCRA*1B11+100010
05                        RANDOM          ;PLACE AC0 IN AC1, SEE
06                        LCALL FRANG
07 07654 100450           FRANG=ASCRA*1B11+100010
08 07655 040651           STA 0,DAC0
09 07656 152400           SUB 2,2         ;SYSTEM REFFERANCE MANUAL
10                        CALL            ;FOR FURTHER INFORMATION.
11 07657 004652           JSR     MPYAL
12 07660 034646           LDA 3,DAC0
13 07661 166414           SUB# 3,1,SZR
14                        ERROR
15 07662 004402           JSR .+2
16 07663 000402           JMP .+2
17                        LCALL ERRET
18 07664 100470           ERRET=ASCRA*1B11+100010
19                        LOOP
20                        LCALL LLOOP
21 07665 100430           LLOOP=ASCRA*1B11+100010
22
23                ;AR10:
24                        SETUP           ;TEST "COM/INC" SWAPPED,
25                        LCALL SETUL
26 07666 100410           SETUL=ASCRA*1B11+100010
27 07667 102300           ADCS 0,0
28 07670 105705           INCS 0,1,SNR
29 07671 130304           COMS 1,2,SZR
30 07672 145705           INCS 2,1,SNR
31 07673 124346           COMOS 1,1,SEZ
32 07674 131707           INCS 1,2,SBN
33 07675 150304           COMS 2,2,SZR
34 07676 151704           INCS 2,2,SZR
35                        ERROR
36 07677 004402           JSR .+2
37 07700 000402           JMP .+2
38                        LCALL ERRET
39 07701 100470           ERRET=ASCRA*1B11+100010
40                        LOOP
41                        LCALL LLOOP
42 07702 100430           LLOOP=ASCRA*1B11+100010
```

```
;0154 MNMRT
01                ;AR11:
02                        SETUP           ;COMPLIMENT AND INCREMENT
03                        LCALL SETUL
04 07703 100410           SETUL=ASCRA*1B11+100010
05                        RANDOM          ;SHOULD BE THE SAME AS
06                        LCALL FRANG
07 07704 100450           FRANG=ASCRA*1B11+100010
08 07705 110400           NEG 0,2         ;NEGATE
09 07706 104000           COM 0,1
10 07707 125400           INC 1,1
11 07710 132414           SUB# 1,2,SZR
12                        ERROR
13 07711 004402           JSR .+2
14 07712 000402           JMP .+2
15                        LCALL ERRET
16 07713 100470           ERRET=ASCRA*1B11+100010
17                        LOOP
18                        LCALL LLOOP
19 07714 100430           LLOOP=ASCRA*1B11+100010
20
21                ;AR12:
22                        SETUP           ;TEST COM LEFT AND RIGHT,
23                        LCALL SETUL
24 07715 100410           SETUL=ASCRA*1B11+100010
25                        RANDOM
26                        LCALL FRANG
27 07716 100450           FRANG=ASCRA*1B11+100010
28 07717 110100           COML 0,2
29 07720 144260           COMCR 2,1
30 07721 106714           SUBS# 0,1,SZR
31                        ERROR
32 07722 004402           JSR .+2
33 07723 000402           JMP .+2
34                        LCALL ERRET
35 07724 100470           ERRET=ASCRA*1B11+100010
36                        LOOP
37                        LCALL LLOOP
38 07725 100430           LLOOP=ASCRA*1B11+100010
```

```
10155 MNMRT
01                ;AR13:
02                        SETUP           ;FORM THE PARITY OF
03                        LCALL SETUL
04 07726 100410           SETUL=ASCRA*1B11+100010
05 07727 105020           MOVZ 0,1        ;C(AC0) IN DIFFIRENT
06 07730 176000           ADC 3,3         ;ROUTINES. CHECK THAT
07 07731 117000           ADD 0,3         ;THE RESULTS ARE EQUAL.
08 07732 163704           ANDS 3,0,SZR
09 07733 000775           JMP .-3
10 07734 102660           SUBCR 0,0       ;SAVE PARITY IN BIT 0.
11 07735 176620           SUBZR 3,3
12 07736 125102           MOVL 1,1,SZC
13 07737 101400           INC 0,0         ;BIT 15 WILL CONTAIN
14 07740 175224           MOVZR 3,3,SZR   ;THE PARITY.
15 07741 000775           JMP .-3
16 07742 115200           MOVM 0,3        ;CHECK TO SEE IF BITS
17 07743 103012           ADD# 0,0,SZC    ;ARE LIKE.
18                        ERROR
19 07744 004402           JSR .+2
20 07745 000402           JMP .+2
21                        LCALL ERRET
22 07746 100470           ERRET=ASCRA*1B11+100010
23                        LOOP
24                        LCALL LLOOP
25 07747 100430           LLOOP=ASCRA*1B11+100010
26
27                ;AR15:
28                        SETUP           ;MISC TEST OF SUB LEFT
29                        LCALL SETUL
30 07750 100410           SETUL=ASCRA*1B11+100010
31                        RANDOM          ;AND RIGHT.
32                        LCALL FRANG
33 07751 100450           FRANG=ASCRA*1B11+100010
34 07752 131000           MOV 1,2
35 07753 106500           SUBL 0,1
36 07754 127600           ANDR 1,1
37 07755 107000           ADD 0,1
38 07756 106500           SUBL 0,1
39 07757 127600           ANDR 1,1
40 07760 107000           ADD 0,1
41 07761 132414           SUB# 1,2,SZR
42                        ERROR
43 07762 004402           JSR .+2
44 07763 000402           JMP .+2
45                        LCALL ERRET
46 07764 100470           ERRET=ASCRA*1B11+100010
47                        LOOP
48                        LCALL LLOOP
49 07765 100430           LLOOP=ASCRA*1B11+100010
```

```
10156 MNMRT
01                ;AR16:
02                        SETUP           ;MISC TEST OF AND RIGHT.
03                        LCALL SETUL
04 07766 100410           SETUL=ASCRA*1B11+100010
05                        RANDOM
06                        LCALL FRANG
07 07767 100450           FRANG=ASCRA*1B11+100010
08 07770 131040           MOVO 1,2
09 07771 113600           ANDR 0,2
10 07772 107400           AND 0,1
11 07773 151100           MOVL 2,2
12 07774 146414           SUB# 2,1,SZR
13                        ERROR
14 07775 004402           JSR .+2
15 07776 000402           JMP .+2
16                        LCALL ERRET
17 07777 100470           ERRET=ASCRA*1B11+100010
18                        LOOP
19                        LCALL LLOOP
20 10000 100430           LLOOP=ASCRA*1B11+100010
```

```
10157 MNMRT
01                      ;AR17:
02                              SETUP           ;MISC ADD SWAPPED TEST.
03                              LCALL SETUL
04 10001 100410                 SETUL=ASCRA*1B11+100010
05                              RANDOM
06                              LCALL FRANG
07 10002 100450                 FRANG=ASCRA*1B11+100010
08 10003 131000                 MOV 1,2
09 10004 113000                 ADD 0,2
10 10005 107300                 ADDS 0,1
11 10006 125300                 MOVS 1,1
12 10007 132414                 SUB# 1,2,SZR
13                              ERROR
14 10010 004402                 JSR .+2
15 10011 000402                 JMP .+2
16                              LCALL ERRET
17 10012 100470                 ERRET=ASCRA*1B11+100010
18                              LOOP
19                              LCALL LLOOP
20 10013 100430                 LLOOP=ASCRA*1B11+100010
21
22                      ;AR18:
23                              SETUP           ;CHECK ADC LEFT.
24                              LCALL SETUL
25 10014 100410                 SETUL=ASCRA*1B11+100010
26                              RANDOM
27                              LCALL FRANG
28 10015 100450                 FRANG=ASCRA*1B11+100010
29 10016 131000                 MOV 1,2
30 10017 112120                 ADCZL 0,2
31 10020 100000                 COM 0,0
32 10021 107120                 ADDZL 0,1
33 10022 132414                 SUB# 1,2,SZR
34                              ERROR
35 10023 004402                 JSR .+2
36 10024 000402                 JMP .+2
37                              LCALL ERRET
38 10025 100470                 ERRET=ASCRA*1B11+100010
39                              LOOP
40                              LCALL LLOOP
41 10026 100430                 LLOOP=ASCRA*1B11+100010
```

```
10158 MNMRT
01                      ;AR19:
02                              SETUP           ;TEST ADC RIGHT.
03                              LCALL SETUL
04 10027 100410                 SETUL=ASCRA*1B11+100010
05                              RANDOM
06                              LCALL FRANG
07 10030 100450                 FRANG=ASCRA*1B11+100010
08 10031 131000                 MOV 1,2
09 10032 112220                 ADCZR 0,2
10 10033 100000                 COM 0,0
11 10034 107220                 ADDZR 0,1
12 10035 132414                 SUB# 1,2,SZR
13                              ERROR
14 10036 004402                 JSR .+2
15 10037 000402                 JMP .+2
16                              LCALL ERRET
17 10040 100470                 ERRET=ASCRA*1B11+100010
18                              LOOP
19                              LCALL LLOOP
20 10041 100430                 LLOOP=ASCRA*1B11+100010
21
22                      ;AR20:
23                              SETUP           ;TEST SUB RIGHT.
24                              LCALL SETUL
25 10042 100410                 SETUL=ASCRA*1B11+100010
26                              RANDOM
27                              LCALL FRANG
28 10043 100450                 FRANG=ASCRA*1B11+100010
29 10044 131000                 MOV 1,2
30 10045 100620                 SUBZR 0,1
31 10046 100420                 NEGZ 0,0
32 10047 113200                 ADDR 0,2
33 10050 132414                 SUB# 1,2,SZR
34                              ERROR
35 10051 004402                 JSR .+2
36 10052 000402                 JMP .+2
37                              LCALL ERRET
38 10053 100470                 ERRET=ASCRA*1B11+100010
39                              LOOP
40                              LCALL LLOOP
41 10054 100430                 LLOOP=ASCRA*1B11+100010
```

```
10159 MNMRT
01                AMJSR:  SETUP           ;TEST THAT INDEX WITH
02                        LCALL SETUL
03 10055 100410           SETUL=ASCRA*1B11+100010
04 10056 004401           JSR .+1         ;SIGN BIT SET DOES
05 10057 171122           MOVZL 3,2,SZC   ;NOT LOAD INTO PC ON JSR.
06                        ERROR
07 10060 004402           JSR .+2
08 10061 000402           JMP .+2
09                        LCALL ERRET
10 10062 100470           ERRET=ASCRA*1B11+100010
11 10063 151240           MOVOR 2,2
12 10064 005007           JSR 7,2         ;GO TO NEXT LOCATION
13 10065 004401           JSR .+1
14 10066 165000           MOV 3,1         ;JSR SHOULD NEVER
15 10067 125112           MOVL# 1,1,SZC   ;STORE THE SIGN BIT.
16                        ERROR
17 10070 004402           JSR .+2
18 10071 000402           JMP .+2
19                        LCALL ERRET
20 10072 100470           ERRET=ASCRA*1B11+100010
21                        LOOP
22                        LCALL LLOOP
23 10073 100430           LLOOP=ASCRA*1B11+100010
24
25                AREND:  LCALL RETU2
26 10074 100510           RETU2=ASCRA*1B11+100010
27 10075 000000 ATEND: 0
```

```
10160 MNMRT
01                ;       .TITL MUDVT
02                ;MULTIPLY DIVIDE TEST COMPATABLE WITH
03                ;THE DIAGNOSTIC LINKER
04                        .MACRO STORE
05                        JSR @IXIOR
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 MUDI0
20          010076 LMEML=.
21          000152        .LOC LPG0
22 00152 010101           MUDI0
23          000153 LPG0=.
24          010076        .LOC LMEML
25 10076 000000           0       ;TEST PASS CTR
26 10077 000000           0       ;TEST ERROR CTR
27 10100 000000           0       ;INTERRUPT TIMEOUT SWITCH
28
```

```
10161 MNMRT
01 10101 010120 MUDI0:  MUDI1
02 10102 010123         MDA00
03 10103 000000         0
04 10104 000000         0
05 10105 177777         -1
06 10106 176000         176000
07 10107 010770         MDTEX
08 10110 010770         MDTEX
09 10111 052515         .TXTE (MUL/DIV TEST(
10       127714
11       144504
12       120126
13       142724
14       152123
15       000000
16
17 10120 102400 MUDI1:  SUB 0,0
18 10121 040762         STA 0,MUDI0+2
19 10122 001400         JMP 0,3
```

```
10162 MNMRT
01              MLA00:  SETUP   #0*0=0
02                      LCALL SETUL
03 10123 100410         SETUL=ASCHA*1B11+100010
04                      STORE
05 10124 006530         JSR @IXTOR
06 10125 000000         0                       #AC0
07 10126 000000         0                       #AC1
08 10127 000000         0                       #AC2
09 10130 006525         JSR @HMUL
10                      MULCK
11 10131 006525         JSR @IMCK
12 10132 000000         0
13 10133 000000         0
14 10134 000000         0
15                      LOOP
16                      LCALL LLOOP
17 10135 100430         LLOOP=ASCHA*1B11+100010
18
19              #A01:
20                      SETUP
21                      LCALL SETUL
22 10136 100410         SETUL=ASCHA*1B11+100010
23                      STORE
24 10137 006515         JSR @IXTOR
25 10140 000000         0
26 10141 000000         0
27 10142 177777         -1
28 10143 006512         JSR @HMUL
29                      MULCK
30 10144 006512         JSR @IMCK
31 10145 000000         0
32 10146 000000         0
33 10147 177777         -1
34                      LOOP
35                      LCALL LLOOP
36 10150 100430         LLOOP=ASCHA*1B11+100010
37
```

```
10163 MNMRT
01                  ;A02:
02                          SETUP
03                          LCALL SETUL
04 10151 100410             SETUL=ASCHA*1B11+100010
05                          STORE
06 10152 006502             JSR @IXTOR
07 10153 000000             0
08 10154 177777             -1
09 10155 000000             0
10 10156 006477             JSR @MMUL
11                          MULCK
12 10157 006477             JSR @IMCK
13 10160 000000             0
14 10161 000000             0
15 10162 000000             0
16                          LOOP
17                          LCALL LLOOP
18 10163 100430             LLOOP=ASCHA*1B11+100010
19
20                  ;A03:
21                          SETUP
22                          LCALL SETUL
23 10164 100410             SETUL=ASCHA*1B11+100010
24                          STORE
25 10165 006467             JSR @IXTOR
26 10166 177777             -1
27 10167 000000             0
28 10170 000000             0
29 10171 006464             JSR @MMUL
30                          MULCK
31 10172 006464             JSR @IMCK
32 10173 000000             0
33 10174 177777             -1
34 10175 000000             0
35                          LOOP
36                          LCALL LLOOP
37 10176 100430             LLOOP=ASCHA*1B11+100010
```

```
10164 MNMRT
01                  ;A04:
02                          SETUP
03                          LCALL SETUL
04 10177 100410             SETUL=ASCHA*1B11+100010
05                          STORE
06 10200 006454             JSR @IXTOR
07 10201 000000             0
08 10202 000001             1
09 10203 177777             -1
10 10204 006451             JSR @MMUL
11                          MULCK
12 10205 006451             JSR @IMCK
13 10206 000000             0
14 10207 177777             -1
15 10210 177777             -1
16                          LOOP
17                          LCALL LLOOP
18 10211 100430             LLOOP=ASCHA*1B11+100010
19
20                  ;A05:
21                          SETUP
22                          LCALL SETUL
23 10212 100410             SETUL=ASCHA*1B11+100010
24                          STORE
25 10213 006441             JSR @IXTOR
26 10214 000000             0
27 10215 177777             -1
28 10216 000001             1
29 10217 006436             JSR @MMUL
30                          MULCK
31 10220 006436             JSR @IMCK
32 10221 000000             0
33 10222 177777             -1
34 10223 000001             1
35                          LOOP
36                          LCALL LLOOP
37 10224 100430             LLOOP=ASCHA*1B11+100010
38
```

```
10165 MNMRT
01                ;A06:
02                        SETUP
03                        LCALL SETUL
04 10225 100410           SETUL=ASCHA*1B11+100010
05                        STORE
06 10226 006426           JSR @IXTOR
07 10227 177777           -1
08 10230 177777           -1
09 10231 000000           0
10 10232 006423           JSR @MMUL
11                        MULCK
12 10233 006423           JSR @IMCK
13 10234 000000           0
14 10235 177777           -1
15 10236 000000           0
16                        LOOP
17                        LCALL LLOOP
18 10237 100430           LLOOP=ASCHA*1B11+100010
19
20                ;A07:
21                        SETUP
22                        LCALL SETUL
23 10240 100410           SETUL=ASCHA*1B11+100010
24                        STORE
25 10241 006413           JSR @IXTOR
26 10242 000001           1
27 10243 000002           2
28 10244 000002           2
29 10245 006410           JSR @MMUL
30                        MULCK
31 10246 006410           JSR @IMCK
32 10247 000000           0
33 10250 000005           5
34 10251 000002           2
35                        LOOP
36                        LCALL LLOOP
37 10252 100430           LLOOP=ASCHA*1B11+100010
38 10253 000406           JMP MDA08
39 10254 010462 IXTOR:    SXTOR
40 10255 010705 MMUL:     XMMUL
41 10256 010426 IMCK:     MCK
42 10257 010427 IDCK:     DCK
43 10260 010717 MDIV:     XMDIV
```

```
10166 MNMRT
01                MDA08:  SETUP   ;EXPECT A DIVIDE ERROR
02                        LCALL SETUL
03 10261 100410           SETUL=ASCHA*1B11+100010
04                        STORE           ;AND NO CHANGE TO ACS.
05 10262 006772           JSR @IXTOR
06 10263 000000           0
07 10264 000000           0
08 10265 000000           0
09 10266 006772           JSR @MDIV
10                        DIVCK
11 10267 006770           JSR @IDCK
12 10270 000000           0
13 10271 000000           0
14 10272 000000           0
15                        LOOP
16                        LCALL LLOOP
17 10273 100430           LLOOP=ASCHA*1B11+100010
18
19                ;A09:
20                        SETUP   ;EXPECT DIVIDE ERROR.
21                        LCALL SETUL
22 10274 100410           SETUL=ASCHA*1B11+100010
23                        STORE           ;NO CHANGE TO ACS.
24 10275 006757           JSR @IXTOR
25 10276 177777           -1
26 10277 177777           -1
27 10300 177777           -1
28 10301 006757           JSR @MDIV
29                        DIVCK
30 10302 006755           JSR @IDCK
31 10303 177777           -1
32 10304 177777           -1
33 10305 177777           -1
34                        LOOP
35                        LCALL LLOOP
36 10306 100430           LLOOP=ASCHA*1B11+100010
37
```

```
10167 MNMRT
01                  ;A10:
02                          SETUP   ;DIVIDE ERROR SHOULD
03                          LCALL SETUL
04 10307 100410             SETUL=ASCRA*1B11+100010
05                          STORE           ;SET THE CARRY
06 10310 006744             JSR @IXTOR
07 10311 177777             -1
08 10312 177777             -1
09 10313 177777             -1
10 10314 101020             MOVZ 0,0
11 10315 006743             JSR @MDIV
12 10316 101003             MOV 0,0,SNC
13                          DIVER
14 10317 006705             JSR @EDIV
15                          LOOP
16                          LCALL LLOOP
17 10320 100430             LLOOP=ASCRA*1B11+100010
18
19                  ;A11:
20                          SETUP
21                          LCALL SETUL
22 10321 100410             SETUL=ASCRA*1B11+100010
23                          STORE
24 10322 006732             JSR @IXTOR
25 10323 000000             0
26 10324 000000             0
27 10325 000001             1
28 10326 006732             JSR @MDIV
29                          DIVCK
30 10327 006730             JSR @IDCK
31 10330 000000             0
32 10331 000000             0
33 10332 000001             1
34                          LOOP
35                          LCALL LLOOP
36 10333 100430             LLOOP=ASCRA*1B11+100010
```

```
10168 MNMRT
01                  ;A12:
02                          SETUP   ;NO DIVIDE ERROR SHOULD
03                          LCALL SETUL
04 10334 100410             SETUL=ASCRA*1B11+100010
05                          STORE           ;OCCURE AND CARRY SHOULD
06 10335 006717             JSR @IXTOR
07 10336 000000             0               ;BE A ZERO.
08 10337 000000             0
09 10340 000001             1
10 10341 101040             MOVO 0,0
11 10342 006716             JSR @MDIV
12 10343 101002             MOV 0,0,SZC
13                          DIVER
14 10344 006460             JSR @EDIV
15                          LOOP
16                          LCALL LLOOP
17 10345 100430             LLOOP=ASCRA*1B11+100010
18
19                  ;A13:
20                          SETUP   ;4/2=2
21                          LCALL SETUL
22 10346 100410             SETUL=ASCRA*1B11+100010
23                          STORE
24 10347 006705             JSR @IXTOR
25 10350 000000             0
26 10351 000004             4
27 10352 000002             2
28 10353 006705             JSR @MDIV
29                          DIVCK
30 10354 006703             JSR @IDCK
31 10355 000000             0
32 10356 000002             2
33 10357 000002             2
34                          LOOP
35                          LCALL LLOOP
36 10360 100430             LLOOP=ASCRA*1B11+100010
37
```

```
10169 MNMRT
01                ;A14:
02                        SETUP   ;CHECK REMAINDER
03                        LCALL SETUL
04 10361 100410           SETUL=ASCRA*1B11+100010
05                        STORE
06 10362 006672           JSR @IXTOR
07 10363 000000           0
08 10364 077777           77777
09 10365 100000           100000
10 10366 006672           JSR @MDIV
11                        DIVCK
12 10367 006670           JSR @IDCK
13 10370 077777           77777
14 10371 000000           0
15 10372 100000           100000
16                        LOOP
17                        LCALL LLOUP
18 10373 100430           LLOUP=ASCRA*1B11+100010
19
20                ;A15:
21                        SETUP
22                        LCALL SETUL
23 10374 100410           SETUL=ASCRA*1B11+100010
24                        STORE
25 10375 006657           JSR @IXTOR
26 10376 000000           0
27 10377 177777           -1
28 10400 177777           -1
29 10401 006657           JSR @MDIV
30                        DIVCK
31 10402 006655           JSR @IDCK
32 10403 000000           0
33 10404 000001           1
34 10405 177777           -1
35                        LOOP
36                        LCALL LLOUP
37 10406 100430           LLOUP=ASCRA*1B11+100010
```

```
10170 MNMRT
01                ;A16:
02                        SETUP
03                        LCALL SETUL
04 10407 100410           SETUL=ASCRA*1B11+100010
05                        STORE
06 10410 006644           JSR @IXTOR
07 10411 000001           1
08 10412 000000           0
09 10413 000002           2
10 10414 006644           JSR @MDIV
11                        DIVCK
12 10415 006642           JSR @IDCK
13 10416 000000           0
14 10417 100000           100000
15 10420 000002           2
16                        LOOP
17                        LCALL LLOUP
18 10421 100430           LLOUP=ASCRA*1B11+100010
19
20                ;A17:
21 10422 002401           JMP .+1
22 10423 010500           MTST
23 10424 010736 EDIV:     DERR
24 10425 010731 EMUL:     MERR
```

```
10171 MNMRT
01
02 10426 101021 MCK:    MOVZ 0,0,SKP    ;CHECK MUL RESULT
03 10427 101040 DCK:    MOVO 0,0        ;CHECK DIV RESULT
04 10430 054444         STA 3,XCKRET
05 10431 035402         LDA 3,2,3
06 10432 156414         SUB# 2,3,SZR
07 10433 000411         JMP CK1         ;AC2 WRONG
08 10434 034440         LDA 3,XCKRET
09 10435 031400         LDA 2,0,3
10 10436 035401         LDA 3,1,3
11 10437 112415         SUB# 0,2,SNR    ;CHECK AC0
12 10440 136414         SUB# 1,3,SZR    ;CHECK AC1
13 10441 000403         JMP CK1         ;ERROR
14 10442 034432         LDA 3,XCKRET
15 10443 001403         JMP 3,3
16 10444 034430 CK1:    LDA 3,XCKRET
17 10445 021400         LDA 0,0,3
18 10446 025401         LDA 1,1,3
19 10447 031402         LDA 2,2,3
20 10450 040550         STA 0,OKAC
21 10451 044550         STA 1,OKMQ
22 10452 050550         STA 2,OKMD
23
24 10453 101002 CK2:    MOV 0,0,SZC     ;IF C(CARRY)=0 ITS MUL ERR
25 10454 000403         JMP .+3         ;OTHERWISE ITS A DIVIDE ER
26                      MULER
27 10455 006750         JSR @EMUL
28 10456 101001         MOV 0,0,SKP
29                      DIVER
30 10457 006745         JSR @EDIV
31 10460 034414         LDA 3,XCKRET
32 10461 001403         JMP 3,3
33
```

```
10172 MNMRT
01 10462 021400 SXTOR:  LDA 0,0,3       ;PICK UP ARGUEMENTS
02 10463 025401         LDA 1,1,3       ;AND STORE IN ORIG
03 10464 031402         LDA 2,2,3       ;NUMBERS TABLE.
04 10465 042404         STA 0,@XOAC
05 10466 046404         STA 1,@XOMQ
06 10467 052404         STA 2,@XOMD
07 10470 001403         JMP 3,3
08 10471 010675 XOAC:   OAC
09 10472 010676 XOMQ:   OMQ
10 10473 010677 XOMD:   OMD
11 10474 000000 XCKRET: 0
12 10475 010705 IHMUL: XHMUL
13 10476 010702 IHMD: HMD
14 10477 010717 IHDIV: XHDIV
15
```

```
10173 MNMRT
01                MTST:   SETUP   ;CHECK MULTIPLY WITH
02                        LCALL SETUL
03 10500 100410           SETUL=ASCRA*1B11+100010
04 10501 004522           JSR RAN
05 10502 006773           JSR @IHMUL
06 10503 022766           LDA 0,@XOAC
07 10504 026766           LDA 1,@XOMQ
08 10505 032766           LDA 2,@XOMD
09 10506 004537           JSR XMUL          ;PROGRAMED MULTIPLY
10 10507 036767           LDA 3,@IHMD
11 10510 156414           SUB# 2,3,SZR
12 10511 000405           JMP .+5
13 10512 030566           LDA 2,HAC
14 10513 034566           LDA 3,HMQ
15 10514 112415           SUB# 0,2,SNR
16 10515 136414           SUB# 1,3,SZR
17                        MULER             ;MULTIPLY FAILED
18 10516 006707           JSR @EMUL
19                        LOOP
20                        LCALL LLOOP
21 10517 100430           LLOOP=ASCRA*1B11+100010
22
23                DTST:   SETUP   ;CHECK DIVIDE WITH
24                        LCALL SETUL
25 10520 100410           SETUL=ASCRA*1B11+100010
26 10521 004502           JSR RAN ;RANDOM NUMBERS,
27 10522 006755           JSR @IHDIV        ;HARDWARE DIVIDE
28 10523 020552           LDA 0,DAC
29 10524 024552           LDA 1,DMQ
30 10525 030552           LDA 2,DMD
31 10526 004533           JSR XDIV          ;PROGRAMED DIVIDE
32 10527 034553           LDA 3,HMD
33 10530 156414           SUB# 2,3,SZR
34 10531 000405           JMP .+5
35 10532 030546           LDA 2,HAC
36 10533 034546           LDA 3,HMQ
37 10534 112415           SUB# 0,2,SNR
38 10535 136414           SUB# 1,3,SZR
39                        DIVER             ;DIVIDE FAILED
40 10536 006666           JSR @EDIV
41                        LOOP
42                        LCALL LLOOP
43 10537 100430           LLOOP=ASCRA*1B11+100010
```

```
10174 MNMRT
01 10540 020454 MLTST:  LDA 0,M100        ;MULTIPLY DIVIDE TEST
02 10541 040425           STA 0,FOB
03                        SETUP
04                        LCALL SETUL
05 10542 100410           SETUL=ASCRA*1B11+100010
06 10543 004460           JSR RAN
07 10544 004553           JSR XMDIV         ;HARD DIVIDE
08 10545 040450           STA 0,D0
09 10546 044450           STA 1,D1
10 10547 050450           STA 2,D2
11 10550 004535           JSR XMMUL         ;HARD MULTIPLY
12 10551 034526           LDA 3,DMD
13 10552 156414           SUB# 2,3,SZR
14 10553 000414           JMP MDT2
15 10554 030521           LDA 2,DAC
16 10555 034521           LDA 3,DMQ
17 10556 112415           SUB# 0,2,SNR
18 10557 136414           SUB# 1,3,SZR
19 10560 000407           JMP MDT2
20                MDT1:   LOOP
21                        LCALL LLOOP
22 10561 100430           LLOOP=ASCRA*1B11+100010
23 10562 010404           ISZ FOB
24 10563 000757           JMP MDTST+2
25 10564 002401           JMP @.+1
26 10565 010770           MDTEX
27 10566 000000 FOB: 0
28 10567 020506 MDT2:   LDA 0,DAC         ;EITHER MUL OR DIV
29 10570 024506           LDA 1,DMQ         ;FAILED , TYR TO FIND
30 10571 030506           LDA 2,DMD         ;WHICH ONE.
31 10572 004467           JSR XDIV
32 10573 034422           LDA 3,D0
33 10574 116414           SUB# 0,3,SZR
34 10575 000407           JMP MDT3
35 10576 034420           LDA 3,D1
36 10577 136414           SUB# 1,3,SZR
37 10600 000404           JMP MDT3
38 10601 034416           LDA 3,D2
39 10602 156415           SUB# 2,3,SNR
40 10603 000403           JMP .+3
41                MDT3:   DIVER             ;ITS A DIVIDE ERROR
42 10604 006620           JSR @EDIV
43 10605 000754           JMP MDT1
44 10606 044467           STA 0,DAC
45 10607 044467           STA 1,DMQ
46 10610 050467           STA 2,DMD
47 10611 004434           JSR XMUL
48                        MULER             ;ITS A MULTIPLY ERR
49 10612 006613           JSR @EMUL
50 10613 000746           JMP MDT1
51 10614 177700 M100:   -100
52 10615 000000 D0: 0
53 10616 000000 D1: 0
54 10617 000000 D2: 0
```

```
↓0175 MNMRT
01 10620 000000 OKAC: 0
02 10621 000000 OKMQ: 0
03 10622 000000 OKMD: 0
04
05 10623 054421 RAN:    STA 3,RANRET     ;GET RANDOM OPERATORS
06              RAN1:   LCALL ARANG
07 10624 100270         ARANG-ASCRA*1811+100010
08 10625 110700         NEGS 0,2         ;FORM MQ*MD
09 10626 105120         MOVZL 0,1
10 10627 127100         ADDL 1,1
11 10630 107300         ADDS 0,1
12 10631 112415         SUB# 0,2,SNR
13 10632 000772         JMP RAN1         ;REJECT IF AC=MD
14 10633 142432         SUBZ# 2,0,SZC
15 10634 115001         MOV 0,3,SKP
16 10635 000403         JMP .+3
17 10636 141000         MOV 2,0          ;MAKE AC LESS THAN
18 10637 171000         MOV 3,2          ;MD IN ALL CASES.
19 10640 040435 RAN2:   STA 0,OAC        ;STORE IN ORIGIONAL
20 10641 044435         STA 1,OMQ        ;NUMBER BLOCK..
21 10642 050435         STA 2,OMD
22 10643 002401         JMP @RANRET
23 10644 000000 RANRET: 0
24 10645 054436 XMUL:   STA 3,MSAV       ;PROGRAMED MULTIPLY
25 10646 034436         LDA 3,M20
26 10647 125203         MOVR 1,1,SNC
27 10650 101201         MOVR 0,0,SKP
28 10651 143220         ADDZR 2,0
29 10652 175404         INC 3,3,SZR
30 10653 000774         JMP XMUL+2
31 10654 125260         MOVCR 1,1
32 10655 040743 XMUL1:  STA 0,OKAC       ;STORE RESULTS
33 10656 044743         STA 1,OKMQ
34 10657 050743         STA 2,OKMD
35 10660 002423         JMP @MSAV
36
```

```
↓0176 MNMRT
01 10661 054422 XDIV:   STA 3,MSAV       ;PROGRAMED DIVIDE
02 10662 142432         SUBZ# 2,0,SZC
03 10663 001400         JMP 0,3          ;OV EXIT
04 10664 034420         LDA 3,M20
05 10665 125120         MOVZL 1,1
06 10666 101100 XDIV1:  MOVL 0,0
07 10667 142412         SUB# 2,0,SZC
08 10670 142400         SUB 2,0
09 10671 125100         MOVL 1,1
10 10672 175404         INC 3,3,SZR
11 10673 000773         JMP XDIV1
12 10674 000761         JMP XMUL1        ;STORE RESULTS.
13 10675 000000 OAC: 0
14 10676 000000 OMQ: 0
15 10677 000000 OMD: 0
16 10700 000000 HAC: 0
17 10701 000000 HMQ: 0
18 10702 000000 HMD: 0
19 10703 000000 MSAV: 0
20 10704 177760 M20: -20
```

```
↓0177 MNMRT
01 10705 054423 XMMUL:  STA 3,XMRET     ;HARDWARE MULTIPLY
02 10706 073301          DOCP 2,01
03 10707 004402          JSR .+2
04 10710 010710          .
05 10711 040767          STA 0,MAC
06 10712 020776          LDA 0,.-2
07 10713 116434          SUBZ# 0,3,SZR
08 10714 000415          JMP MERR
09 10715 020763          LDA 0,MAC
10 10716 000406          JMP XMCOM
11
12 10717 054411 XMDIV:  STA 3,XMRET     ;HARDWARE DIVIDE
13 10720 176440          SUBO 3,3
14 10721 073101          DOCS 2,01
15 10722 175004          MOV 3,3,SZR
16 10723 000413          JMP DERR
17 10724 040754 XMCOM:  STA 0,MAC       ;STORE HARDWARE RESULTS.
18 10725 044754          STA 1,MMQ
19 10726 050754          STA 2,MMD
20 10727 002401          JMP @XMRET
21 10730 000000 XMRET:  0
22
23 10731 054412 MERR:   STA 3,MERET     ;MULTIPLY ERROR
24 10732 101001          MOV 0,0,SKP
25 10733 010772          MULMED         ;HEADING FOR MUL.
26 10734 020777          LDA 0,.-1
27 10735 000407          JMP GBR
28
29 10736 054405 DERR:   STA 3,MERET     ;DIVIDE ERROR
30 10737 101001          MOV 0,0,SKP
31 10740 011004          DIVMED
32 10741 020777          LDA 0,.-1
33 10742 000402          JMP GBR
34 10743 000000 MERET:  0
```

```
↓0178 MNMRT
01 10744 040425 GBR:    STA 0,XMDR      ;PRINT GOOD-BAD RESULTS
02                      LCALL ERROI
03 10745 100350         ERROI=ASCRA*1B11+100010
04 10746 000403         JMP .+3
05 10747 101001         MOV 0,0,SKP
06 10750 020421         LDA 0,XMDR
07                      LCALL ERRTX
08 10751 100170         ERRTX=ASCRA*1B11+100010
09 10752 020723         LDA 0,OAC
10 10753 024723         LDA 1,OMQ
11 10754 030723         LDA 2,OMD
12                      LCALL ERROC
13 10755 100370         ERROC=ASCRA*1B11+100010
14 10756 000401         JMP .+1
15 10757 020721         LDA 0,MAC
16 10760 024721         LDA 1,MMQ
17 10761 030721         LDA 2,MMD
18                      LCALL ERROC
19 10762 100370         ERROC=ASCRA*1B11+100010
20 10763 020635         LDA 0,OKAC
21 10764 024635         LDA 1,OKMQ
22 10765 030635         LDA 2,OKMD
23                      LCALL ERROC
24 10766 100370         ERROC=ASCRA*1B11+100010
25 10767 000401         JMP .+1
26              MDTEX:  LCALL RETRN
27 10770 100210         RETRN=ASCRA*1B11+100010
28 10771 000000 XMDR:   0
29 10772 005215 MULMED: .TXTE 1<15><12>
30 10773 052515 MUL<15><12>AC0  AC1     AC2 1
31 11004 005215 DIVMED: .TXTE 1<15><12>
32 11005 144504 DIV<15><12>AC0  AC1     AC2 1
```

```
10179 MNMRT
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              011016 LMEML=.
11              000153          .LOC LPG0
12 000153 011021                RT.00
13              000154 LPG0=.
14              011016          .LOC LMEML
15 11016 000000                 0       ;TEST PASS CTR
16 11017 000000                 0       ;TEST ERROR CTR
17 11020 000000                 0       ;INTERRUPT TIMEOUT SWITCH
18 11021 011041 RT.00:  RT.01
19 11022 011055         RT.02
20 11023 000000         0
21 11024 000000         0
22 11025 177777         -1
23 11026 176000         176000
24 11027 011100         RT.03
25 11030 011100         RT.03
26 11031 142722         .TXTE (REAL TIME CLOCK(
27              146101
28              152240
29              046711
30              120305
31              140303
32              141717
33              000113
```

```
10180 MNMRT
01                      ;SET UP INTERRUPT FOR REAL TIME CLOCK
02              071077 .DUSR RTCEN=071077
03              065077 .DUSR RTCDS=065077
04 11041 060277 RT.01:  INTDS
05 11042 065077         RTCDS   ;TURN IT OFF
06 11043 102400         SUB 0,0
07 11044 040757         STA 0,RT.00+2   ;ENABLE TEST
08 11045 020407         LDA 0,RT.02-1   ;SET UP EXECUTE ENTRY ADDR.
09 11046 040754         STA 0,RT.00+1
10 11047 020403         LDA 0,RT.K3
11 11050 040002         STA 0,2         ;INTR ADDRESS
12 11051 001400         JMP 0,3         ;RETURN
13 11052 011133 RT.K3: RT.ID
14 11053 000005 RTFIV: 5
```

```
10181 MNMRT
01                      ;START CLOCK TEST IS NOT DELETED
02 11054 011055                 RT.02
03 11055 020424 RT.02:          LDA 0,RT.K4
04 11056 040744                 STA 0,RT.00+1
05 11057 102420                 SUBZ 0,0
06 11060 101500                 INCL 0,0
07 11061 024443                 LDA 1,RT.K5
08 11062 044443                 STA 1,RTSEC     ;TO COUNT 1 SECOND
09 11063 024437                 LDA 1,RT.K6
10 11064 044442                 STA 1,RTMIN     ;60 SEC'S =1MIN.
11 11065 024766                 LDA 1,RTFIV
12 11066 131120                 MOVZL 1,2
13 11067 044440                 STA 1,RTCTR     ;TO COUNT DOWN 1ST
14 11070 050440                 STA 2,RTCTR+1   ;2ND AFTER 10 MORE
15 11071 133000                 ADD 1,2
16 11072 126400                 SUB 1,1
17 11073 044133                 STA 1,RTTIM
18 11074 126000                 ADC 1,1
19 11075 044137                 STA 1,TIMSW     ;SET RT=0 INHIBIT TIME
20 11076 050433                 STA 2,RTCTR+2   ;3RD AT HALF HOUR
21 11077 071077                 RTCEN           ;START THE RTC
22                      RT.03:  LCALL RETRN
23 11100 100210                 RETRN-ASCRA*1B11+100010
24 11101 011102 RT.K4: RT.04
25 11102 102400 RT.04:  SUB 0,0
26 11103 040713                 STA 0,RT.00-3   ;SO NO 65K TYPEOUT
27 11104 020137                 LDA 0,TIMSW
28 11105 101004                 MOV 0,0,SZR     ;PRINT TIME
29 11106 000772                 JMP RT.03       ;NOT YET
30 11107 102000                 ADC 0,0
31 11110 040137                 STA 0,TIMSW     ;RESET INH. SW
32 11111 020470                 LDA 0,RTTEX
33                              LCALL ERRTX
34 11112 100170                 ERRTX-ASCRA*1B11+100010
35 11113 024133                 LDA 1,RTTIM
36                              LCALL PDECI     ;ELAPSED TIME IN MINUTES
37 11114 100150                 PDECI-ASCRA*1B11+100010
38 11115 024140                 LDA 1,ERTOT     ;# ERR TYPEOUTS
39 11116 125005                 MOV 1,1,SNR
40 11117 000761                 JMP RT.03
41                              LCALL PDECI     ;PRINT ERROR TOTAL
42 1112Ø 100150                 PDECI-ASCRA*1B11+100010
43 11121 000757                 JMP RT.03
```

```
10182 MNMRT
01 11122 000074 RT.K6:  60.
02 11123 000034 RT.28:  28.     ;ERROR CORRECTION COUNT
03 11124 001037 RT.K5:  543.    ;RTC FREQ = 542.53 HERTZ
04 11125 000000 RTSEC:  0
05 11126 000000 RTMIN:  0
06 11127 000000 RTCTR:  0
07 11130 000000          0
08 11131 000000          0
09 11132 000036          30.
10 11133 061401 RT.ID:  PSH 0
11 11134 061401         PSH 1
12 11135 061401         PSH 2
13 11136 061401         PSH 3
14 11137 020000         LDA 0,0         ;GET PC
15 11140 101100         MOVL 0,0
16 11141 061401         PSH 0
17 11142 014763         DSZ RTSEC       ;1 SECOND
18 11143 000425         JMP RTSTR       ;NO
19 11144 020760         LDA 0,RT.K5
20 11145 040760         STA 0,RTSEC
21 11146 014760         DSZ RTMIN       ;1 MINUTE
22 11147 000421         JMP RTSTR       ;NO
23 11150 010133         ISZ RTTIM       ;BUMP ELAPSED MINUTES
24 11151 024751         LDA 1,RT.K6
25 11152 030751         LDA 2,RT.28     ;CORRECT SECOND COUNT
26 11153 142400         SUB 2,0         ;BY 28 COUNTS EVERY MIN.
27 11154 040751         STA 0,RTSEC     ;CORRECTED SEC COUNT
28 11155 044751         STA 1,RTMIN     ;RES. MIN. CTR
29 11156 014751         DSZ RTCTR       ;TIME TO PRINT
30 11157 000411         JMP RTSTR       ;NOT YET
31 11160 126400         SUB 1,1
32 11161 044137         STA 1,TIMSW     ;CLR PR. INH. SW
33 11162 020746         LDA 0,RTCTR+1
34 11163 024746         LDA 1,RTCTR+2
35 11164 030746         LDA 2,RTCTR+3
36 11165 040742         STA 0,RTCTR
37 11166 044742         STA 1,RTCTR+1
38 11167 050742         STA 2,RTCTR+2
39 1117Ø 061601 RTSTR:  POP 0
40 11171 101220         MOVZR 0,0       ;PC AND CRY
41 11172 040000         STA 0,0
42 11173 075601         POP 3
43 11174 071601         POP 2
44 11175 065601         POP 1
45 1117Ø 061601         POP 0
46 11177 060177         INTEN
47 1120Ø 002000         JMP @0          ;RETURN
48 11201 011202 RTTEX:  .+1
49 11202 005215         .TXTE (<15><12>M/1=(
```

```
10183 MNMRT
01                ;       .TITL TITES
02                        NEXTT TT.00
03        011206 LMEML=.
04        000154          .LOC LPG0
05 000154 011211          TT.00
06        000155 LPG0=.
07        011206          .LOC LMEML
08 11206  000000          0       ;TEST PASS CTR
09 11207  000000          0       ;TEST ERROR CTR
10 11210  000000          0       ;INTERRUPT TIMEOUT SWITCH
11 11211  011226 TT.00:   TT.01
12 11212  011251          TT.02
13 11213  000000          0       ;WAIT FOR INTERRUPT SW
14 11214  177700          -64.
15 11215  177716          -50.
16 11216  176000          176000
17 11217  011255          TT.XI
18 11220  011255          TT.XI
19        011221          .IXTE (
20 11221  152324 TTY TEST(
21        120131
22        142724
23        152123
24        000000
25                ;INITIALIZE TTY TEST
26 11226  102400 TT.01:   SUB 0,0
27 11227  040535          STA 0,TT.CK     ;CLR CHAR COUNT
28 11230  040763          STA 0,TT.00+2   ;CLR WAIT INTR
29 11231  020425          LDA 0,TT.03     ;INTR ADRS
30 11232  054454          STA 3,TT.S3
31 11233  020411          LDA 0,TT.K1
32 11234  024411          LDA 1,TT.K2
33 11235  030411          LDA 2,TT.K3
34 11236  006071          JSR @EINTS      ;ENTER KYBORD INT SERV
35 11237  020410          LDA 0,TT.K4
36 11240  030410          LDA 2,TT.K5
37 11241  006071          JSR @EINTS      ;ENTER TTO INT SERV
38 11242  034444          LDA 3,TT.S3
39 11243  001400          JMP 0,3
40 11244  000010 TT.K1:   TTI
41 11245  000003 TT.K2:   3
42 11246  011257 TT.K3:   TT.TI
43 11247  000011 TT.K4:   TTO
44 11250  011324 TT.K5:   TT.TO
45                ;START TTY PRINTING
46 11251  102620 TT.02:   SUBZR 0,0
47 11252  040741          STA 0,TT.00+2   ;SET WAITING INT
48 11253  020505          LDA 0,TT.CR
49 11254  061111          DOAS 0,TTO      ;OUT CARG RET
50                TT.XI:  LCALL RETRN
51 11255  100210          RETRN=ASCHA*1611+100010
```

```
10184 MNMRT
01                ;TTY INTR SERVICE
02 11256  011257 TT.03:   .+1
03 11257  054427 TT.TI:   STA 3,TT.S3
04 11260  020141          LDA 0,EACTV
05 11261  101202          MOVR 0,0,SZC    ;SKP=NOT WAITING KEY IN
06 11262  000431          JMP TT.CW       ;CLEAR WAITING
07 11263  060610 TT.RD:   DIAC 0,TTI
08 11264  024142          LDA 1,LASTI     ;CK IF LAST TTI INTA SERVICED
09 11265  125103          MOVL 1,1,SNC    ;SKP IS NOT SERVICED
10 11266  000414          JMP TT.DI-2
11 11267  024422          LDA 1,T177
12 11270  123400          AND 1,0         ;MASK OFF PARITY BIT
13 11271  024417          LDA 1,TT.17
14 11272  106415          SUB#0,1,SNR     ;SKP IF NOT CONTROL O
15 11273  000423          JMP EODT        ;GO DIRECT TO ODT
16 11274  024411          LDA 1,TTI04     ;CONTROL "O"
17 11275  122415          SUB# 1,0,SNR    ;SKIP IS NOT CONTROL O
18 11276  000422          JMP TT.P?       ;KEY INPUT SERVICE
19 11277  024410          LDA 1,TT.22     ;CK FOR CONTROL R
20 11300  122415          SUB# 1,0,SNR    ;SKP IS NOT CONTROL R
21 11301  000417          JMP TT.P?       ;KEY INPUT SERVICE
22 11302  103240          ADDOR 0,0
23 11303  040142          STA 0,LASTI     ;SAVE CHAR FOR LATER
24 11304  002402 TT.DI:   JMP @TT.S3
25 11305  000004 TTI04:   4
26 11306  000000 TT.S3:   0
27 11307  000022 TT.22:   22
28 11310  000017 TT.17:   17
29 11311  000177 T177:    177
30 11312  003314 PODT:    ODT
31
32                ;CLEAR BIT 15 OF EACTV
33 11313  101120 TT.CW:   MOVZL 0,0
34 11314  040141          STA 0,EACTV
35 11315  000746          JMP TT.RD
36
37 11316  006774 EODT:    JSR @PODT
38 11317  002767          JMP @TT.S3      ;RETURN TO DISMIS INTR
39
40 11320  040142 TT.P?:   STA 0,LASTI     ;PLACE CHARACTER IN LASTI
41 11321  006402          JSR @IN.P?
42 11322  002764          JMP @TT.S3
43 11323  004146 IN.P?:   INP?
```

```
↓0185 MNMRT
01                      ;TTY OUTPUT INTR HANDLER
02 11324 020440 TT.TO:  LDA 0,TI.CK
03 11325 060211         NIOC TTO
04 11326 054760         STA 3,TI.S3
05 11327 034141         LDA 3,EACTV
06 11330 175102         MOVL 3,3,SZC    ;SKP IF NOT ETYPE
07 11331 002755         JMP @TT.S3      ;THROW DONE AWAY
08 11332 034061         LDA 3,TI.00+2   ;WAITING ITO INTR?
09 11333 024432         LDA 1,TT.B0     ;=180
10 11334 136404         SUB 1,3,SZR     ;SKP IS TIO TEST OUTPUT
11 11335 002751         JMP @TT.S3      ;THROW ONE AWAY
12 11336 101004         MOV 0,0,SZR     ;SKP FOR LINE FEED
13 11337 000406         JMP TT.04       ;INTO PRINTING CHAR
14 11340 020421         LDA 0,TT.LF
15 11341 061111         DOAS 0,ITO
16 11342 020421         LDA 0,TI.SP     ;GET SPACE
17 11343 040421         STA 0,TT.CK     ;TO NXT OUT
18 11344 000740         JMP TT.DI       ;DISMISS INTR
19                      ;OUTPUT SPACE TO Z
20 11345 024415 TT.04:  LDA 1,TT.EN
21 11346 122415         SUB# 1,0,SNR    ;SKP IS NOT Z YET
22 11347 000404         JMP TT.05
23 11350 061111         DOAS 0,TTO
24 11351 010413         ISZ TT.CK
25 11352 000732         JMP TT.DI
26                      ;ALL CHARACTERS PRINTED STOP INTERRUPTS
27 11353 102400 TT.05:  SUB 0,0
28 11354 040637         STA 0,TT.00+2
29 11355 060211         NIOC TTO
30 11356 040406         STA 0,TT.CK
31 11357 000725         JMP TT.DI
32 11360 000215 TT.CR:  215
33 11361 000212 TT.LF:  212
34 11362 000333 TT.EN:  333
35 11363 000240 TT.SP:  240
36 11364 000000 TT.CK:  0
37 11365 100000 TT.B0:  100
```

```
↓0186 MNMRT
01 11366 000000 EXISM:  0
02 11367 000000         0
03 11370 000000 AVALM:  0
04 11371 000000         0
05 11372 000000 DCHM0:  0       ;DCH A MAP
06 11373 000000 DCHM1:  0
07 11374 000000 LSYSE:  0
08 11375 047115 DIRT:   .TXTE [MNMRT    001
09 11403 000000         0
10 11404 000200         STRT1
11 11405 176771         176771
12 11406 000000         0
13 11407 000000         0
14 11410 000000         0
15 11411 000000         0
16 11412 000000         0
17 11413 047503         .TXT ;COPYRIGHT (C) DGC,1976
18 11426 044440  ALL RIGHTS RESERVED]
19 01000 000000         0
20 01001 000155         LPG0
21 01002 000000         0
22 01003 000000         0
23 01004 011374         LSYSE
24                      .END
```

C17	005625		104/05	104/14	105/19	106/19	107/33		
C1777	005630		107/36						
C20	005626		107/34	108/07	108/21				
C400	005624		104/15	105/15	106/04	107/32			
C77	005627		105/31	106/22	107/35				
CALL	000035	MC	116/06	129/11	129/14	129/34	129/37	130/09	130/12
			130/33	131/09	131/30	133/08	133/29	134/07	134/26
			135/08	135/11	135/14	135/17	135/20	135/23	135/26
			135/29	139/12	141/13	141/17	142/08	142/13	142/17
			143/09	143/14	146/12	146/15	153/10		
CALLE	002570		52/01	53/20	53/49				
CALLS	002567		52/01	53/17	53/47				
CATES	000001		3/01	3/12	10/01	179/01			
CB10K	005465		104/13	104/20					
CB17	005462		104/08	104/17					
CB37	005464		104/12	104/19					
CB400	005463		104/09	104/16					
CBBG2	005422		101/44	103/40					
CBLIM	002327		37/45	48/06					
CBLWR	002360		48/16	48/18	48/22	48/23	48/31		
CBPL2	005423		101/16	101/33	103/41				
CBRDS	000004		3/01	3/03	8/20	14/03	99/01		
CBTXT	005432		103/06	103/48					
CBUPR	002361		48/17	48/24	48/32				
CB.00	005213		99/12	99/18					
CB.01	005232		99/18	100/02					
CB.02	005235		99/19	100/07					
CB.03	005331		100/08	102/04					
CB.04	005354		102/13	102/30	103/17				
CB.05	005445		103/21	103/25	104/03				
CB.2A	005255		100/21	100/24	101/04				
CB.2C	005265		101/14						
CB.2L	005251		100/22	100/26					
CB.37	005411		100/16	103/31					
CB.BG	005414		101/26	103/34					
CB.EC	005363		99/24	99/25	103/05				
CB.EN	005415		101/42	102/19	103/35				
CB.ER	005361		102/21	103/02	107/39				
CB.ES	005421		100/13	103/16	103/22	103/39			
CB.F1	005573		107/07	107/20					
CB.F2	005576		107/10	107/14					
CB.F3	005603		107/08	107/15					
CB.F4	005611		107/18	107/21					
CB.F5	005615		107/23	107/25					
CB.FA	005567		103/46	107/03					
CB.L2	005302		101/27	101/32					
CB.L3	005323		101/45	101/50					
CB.LC	005413		101/23	101/38	102/08	103/10	103/26	103/33	
CB.PL	005412		101/14	103/27	103/32				
CB.RL	005277		101/24	103/26					
CB.SE	005420		102/11	103/02	103/11	103/38			
CB.TI	005416		102/10	102/30	102/31	103/36			
CB.TK	005410		100/03	100/07	102/04	102/26	103/09	103/23	103/30
CB.TS	005424		102/05	103/42					
CB.WK	005632		107/26	107/38					
CB.X1	005347		100/11	102/22	102/25	103/14	104/07	104/16	
CB.X2	005376		102/33	103/19					
CCTR	002444		50/20	50/23	50/26	50/33	50/38		

CDGLC	002522	53/06	53/08					
CDISP	002514	23/02	52/01	53/02				
CDODT	002571	53/06	53/51					
CD.EX	002547	52/01	52/13	53/30	64/45			
CD.IN	002563	52/13	53/10	53/43	90/01			
CD.LA	002564	52/01	52/04	52/05	53/09	53/32	53/33	53/36
		53/44	63/09	64/46				
CD.LP	002565	52/01	52/05	53/14	53/45	63/07	64/47	
CD.PP	002566	52/01	52/12	53/15	53/46			
CD.S0	002556	52/01	52/06	53/02	53/27	53/38	90/01	
CD.S1	002557	52/01	52/07	53/03	53/28	53/39	90/01	
CD.S2	002560	52/01	52/08	53/04	53/29	53/40	90/01	
CD.S3	002561	52/01	52/11	53/05	53/34	53/41	63/10	90/01
CHA23	003713	77/43	77/46	77/50				
CHC2T	003703	76/42	76/45	77/30	77/37	78/05	78/40	
CHECK	005550	90/01	106/10	106/13	106/15			
CHPAG	002140	45/16	45/25	45/38	47/25			
CHR2T	003735	77/37	77/45	77/53	78/13			
CHR2Z	003733	77/47	77/52	78/11	80/37	80/40		
CHSAV	002155	45/03	45/33	45/41	45/50	46/02	46/22	47/24
		47/30						
CHSK1	002161	45/19	45/29	46/06	47/16	47/26		
CHSTK	002103	27/23	45/03					
CH.K	003424	71/39	72/10					
CK1	010444	171/07	171/13	171/16				
CK2	010453	171/24						
CKKEY	001075	28/13	28/16					
CKODT	001060	26/29	28/03	53/51				
CK.EX	001121	28/22	28/29	28/32	28/36			
CK.S3	001132	28/10	28/15	28/41	28/43	28/45		
CLF?	003722	23/11	24/11	78/01				
CMAPB	002407	36/42	43/48	46/07	48/29	50/09	59/26	93/01
CM.S0	002445	50/09	50/34	50/39				
CM.S1	002446	50/10	50/35	50/40				
CM.S2	002447	50/11	50/36	50/41				
CMLFX	003551	72/07	74/12	75/01				
CR.LF	003421	70/24	72/07					
CURPR	000104	24/02	29/12	29/17	29/25	30/04	30/05	39/26
		39/29	40/03	40/32	65/31			
CXTES	000001	3/01	3/16	186/01				
D0	010615	174/08	174/32	174/52				
D1	010616	174/09	174/35	174/53				
D2	010617	174/10	174/38	174/54				
DCHMI	000146	24/43	24/44	66/09	94/18			
DCHLO	000145	24/42	66/09	90/07	94/12	110/01	179/01	186/01
DCHM0	011372	34/09	91/17	91/35	186/05			
DCHM1	011373	34/10	186/06					
DCK	010427	165/42	171/03					
DCUIS	000001	3/01	3/04	5/01	14/29	110/01		
DEC?0	003654	77/13	77/31	79/19	79/27			
DEC?P	003671	77/21	77/26					
DEC?T	004040	77/07	79/27					
DEO?T	003663	77/20	77/25					
DERR	010736	170/23	177/16	177/29				
DIGIT	003403	71/20	71/40					
DIRET	005744	103/41	109/02	109/13	109/14			
DIRT	011375	22/05	186/08					
DISDL	005664	108/24	108/57					

```
DISDD  00566M      108/11  108/20
DISTL  005642      108/06  108/16  108/59
DISTU  005640      103/40  103/41  103/44  108/04
DIVCK  000064  MC  100/10  106/10  106/29  107/29  108/29  109/11  109/30
                   170/11
DIVER  000066  MC  109/13  107/13  108/13  171/29  173/39  174/41
DIVME  011004      177/31  178/31
DIVI   007373      144/14
DIVID  002066       23/19   44/03
DIVU   007374      144/15  146/14
DIXOR  005730      108/20  108/58  109/02  109/12
DI.K1  002102       44/05   44/15
DI.L1  002072       44/07   44/12
DI.S3  002101       44/04   44/13   44/14
DLTBL  000051       22/37   28/26   28/28   66/09
DTRET  001243       30/31   30/46
DTST   01052M      173/23
EACTV  000141       24/37   32/09   70/21   75/18   80/08   80/13   80/28
                    80/29  179/01  184/04  184/34  185/05
ECHEC  005557      106/14  106/22
EDIST  005623      105/11  107/31
EDIV   010424      107/14  108/14  170/23  171/30  173/40  174/42
EDMAP  000102       23/29  110/01  179/01  186/01
EHALT  000031  MC   98/13   98/16
EINTP  004404       23/20   86/26
EINTS  000071       23/20  110/01  179/01  180/01  183/34  183/37  186/01
EI.S3  004414       86/26   86/33   86/35
EMSCR  004634       23/29   92/04
EMUL   010425      170/24  171/27  173/16  174/49
ENTPA  002572       54/05   58/16   60/27
ENT.2  002616       54/11   54/15   54/20   54/25
ENT.K  002617       54/17   54/26
EODT   011316      184/15  184/37
EPACS  003255       23/17   69/35
EPADR  003230       23/16   69/10
EPROG  005637      103/32  105/23  106/11  107/19  107/43  108/12
EP.0   003276       69/35   69/47   69/52
EP.1   003277       69/36   69/48   69/53
EP.2   003300       69/37   69/49   69/54
EP.3   003301       69/38   69/50   69/55
EP.RT  003302       69/30   69/56
EQUAL  003604       73/23   75/22   75/24   75/33
ER50.  003115       65/17   66/10
EREXI  003117       65/26   66/08   66/09   67/02   67/03   67/26   67/34
                    69/56
ERMPP  003105       66/02
ERPAC  000066       23/17  121/15
ERPAD  000065       23/16  121/23
ERR1   005633      106/17  107/39
ERRET  000077       23/26   98/15  123/14  123/33  124/11  124/17  124/23
                   125/31  126/31  127/24  127/42  128/26  129/22  129/45
                   130/20  130/41  131/17  131/39  133/16  133/37  134/15
                   134/34  135/37  136/15  136/35  137/16  137/47  138/22
                   139/22  139/40  140/22  141/25  142/25  143/22  146/24
                   146/31  146/38  147/21  148/20  149/10  149/16  149/22
                   149/28  149/34  149/40  149/46  149/52  149/58  150/04
                   151/11  151/17  151/23  151/29  151/35  151/41  151/47
                   151/53  151/59  153/18  153/39  154/16  154/35  155/22
```



```
                   155/46  156/17  157/17  157/36  158/17  158/38  159/10
                   159/20  160/01
ERRK2  002760       62/04   63/10
ERRUC  000073       23/22   52/14   90/01  103/13  110/01  114/18  121/15
                   160/01  178/13  178/19  178/24  179/01  186/01
ERRUE  003132       23/22   67/19
ERRUM  003036       23/21   65/04   65/05
ERRUI  000072       23/21   52/09   90/01  103/04  110/01  114/07  121/06
                   160/01  178/03  179/01  186/01
ERRUR  000025  MC   89/10   90/01   98/08   98/18  110/01  123/10  123/29
                   124/07  124/13  124/19  125/27  126/27  127/20  127/38
                   128/22  129/16  129/41  130/16  130/37  131/13  131/35
                   133/12  133/33  134/11  134/30  135/33  136/11  136/31
                   137/12  137/43  138/18  139/18  139/36  140/18  141/21
                   142/21  143/18  146/20  146/27  146/34  147/17  148/16
                   149/06  149/12  149/18  149/24  149/30  149/36  149/42
                   149/48  149/54  149/60  151/07  151/13  151/19  151/25
                   151/31  151/37  151/43  151/49  151/55  153/14  153/35
                   154/12  154/31  155/18  155/42  156/13  157/13  157/34
                   158/13  158/34  159/06  159/16
ERRMT  002747       23/26   62/03
ERRTX  000065       23/14   95/11   95/15   95/19   95/21   96/18  103/08
                   110/01  114/13  121/10  160/01  178/08  179/01  181/34
                   186/01
ERTIT  003074       65/12   65/34
ERTOT  000140       24/36   29/12   30/29   31/14   32/07   39/35   65/14
                    65/16  181/38
ERTXT  003156       23/14   67/42
ERXXT  003125       67/08   67/50   67/53
ER.C1  003114       66/09
ER.S0  003152       65/04   65/37   67/08   67/19   67/28   67/36   67/42
                    69/10   69/24
ER.S1  003153       65/05   65/39   67/09   67/20   67/30   67/37   67/44
                    69/11   69/18
ER.S2  003154       65/07   65/41   67/10   67/21   67/32   67/38   67/45
                    69/12   69/21   69/27
ER.S3  003155       65/08   67/07   67/11   67/22   67/39   67/46   69/13
ER.TP  003170       67/43   67/52
ESCMA  000055       23/08  100/23  110/01  111/17  118/11  160/01  179/01
                   186/01
EXISM  011366       34/12   36/41   37/15  186/01
EXSCR  002666       23/08   59/06
EX.I1  002707       59/20   59/23   59/26   60/17
EX.K1  002733       60/20   60/25
FILL   005506      105/20  105/33
FOB    010566      174/02  174/23  174/27
FPTST  000001        3/01    3/07    9/15   16/12  160/01  186/01
FRANG  000076       23/25  123/06  123/25  125/07  126/07  127/07  127/35
                   128/07  129/09  129/33  130/07  130/31  131/07  131/28
                   133/06  133/27  135/07  136/06  136/26  137/07  137/26
                   138/07  139/07  139/33  140/07  141/09  142/07  143/07
                   146/06  153/07  154/07  154/27  155/33  156/07  157/07
                   157/28  158/07  158/26  179/01  186/01
FRMOG  003541       74/32   74/36
FTYTX  005477       65/21   97/36
GBR    010744      177/27  177/33  178/01
GETPA  002363       49/06   56/04   60/26   63/07   66/10   91/34
GETP.  002405       49/11   49/15   49/24
```

0199 MNMRI

MM.X2	006131		111/41	113/43					
MNCON	004505		88/18	89/10					
MUDUA	005622		105/13	107/30					
MPYA	007360		144/02	146/17	148/26				
MPYAL	007531		148/26	153/11					
MPYU	007357		141/15	142/15	144/01				
MSI	006514		122/03	122/05	123/01				
MSAV	010703		175/24	175/35	176/01	176/19			
MSEL	002665		58/12	58/21					
MSKRG	004402		85/01	85/21	85/32	86/07	86/19	87/31	89/11
MSZ32	001424		35/04	36/10	36/34				
MS.LI	001464		36/28	36/42	37/01				
MTST	010500		170/22	173/01					
MTTES	000001		3/01	3/11	10/01	17/01	179/01		
MUDI0	010101		160/22	161/01	161/17	161/18			
MUDI1	010120		161/01	161/17					
MUDV1	000000		3/01	3/06	9/15	16/12	160/01		
MULLK	000062	MC	160/07	162/10	162/29	163/11	163/30	164/11	164/30
			165/11	165/30					
MULER	000070	MC	160/16	171/26	173/17	174/48			
MULHE	010772		177/25	178/29					
MVDSK	000001		3/01	3/10	10/01	17/01	179/01	186/01	
MVETA	001465		27/09	37/03	37/14				
MXDSK	000001		3/01	3/15	186/01				
MXTES	000001		3/01	3/17	186/01				
N100	003562		75/06	75/11					
N101	003420		72/06	72/23					
N115	003504		72/23	73/20					
N12	003476		71/27	73/18	75/04				
N121	003477		72/17	73/19					
N125	003417		72/05	72/23					
N136	003505		70/02	72/10					
N15	003422		71/33	72/08					
N17	001134		28/11	28/47					
N177	003423		71/11	72/09					
N40	003414		71/36	72/02	73/04				
N57	003475		71/18	72/23	73/17	74/30			
N60	003446		70/08	72/29					
N67	003413		71/15	72/01					
N7	003305		70/05	70/06					
N75	003304		70/03	72/14					
N77	003563		72/34	75/12					
NEXTT	000003	MC	26/37	99/09	110/01	110/04	117/02	160/01	160/19
			179/01	179/09	183/02	186/01			
NOMAP	003440		72/23						
NOVA	000001		2/03	3/01	11/01	37/01	44/16	52/01	62/09
			64/07	64/10	64/30	64/36	64/45	85/01	87/17
			87/20	90/01	97/28	98/16	179/01	186/01	186/17
NPRUG	001567		38/08	38/10	38/32	38/35			
NTOPN	003507		74/02	74/04					
NTPA	002735		60/21	60/27					
NVDSK	000001		3/01	3/09	10/01	17/01	179/01		
NWSTK	002157		45/11	46/04	47/13				
NXDSK	000001		3/01	3/14	186/01				
NXTLO	003504		72/12	73/11	74/01				
OAC	010675		172/08	173/28	174/15	174/28	174/44	175/19	176/13
			178/09						
OAC0	007526		146/07	146/32	148/25	153/08	153/12		

0200 MNMRI

OAC1	007527		146/08	146/25	148/26				
OAC2	007530		146/09	146/16	148/27				
OCT?A	004730		77/04	79/19					
ODT	003314		22/20	28/44	68/26	70/14	184/30		
OKAC	010620		171/20	175/01	175/32	178/20			
OKMU	010622		171/22	175/03	175/34	178/22			
OKMQ	010621		171/21	175/02	175/33	178/21			
OMD	010677		172/10	173/30	174/12	174/30	174/46	175/21	176/15
			178/11						
OMQ	010676		172/09	173/29	174/16	174/29	174/45	175/20	176/14
			178/10						
OPNLO	003517		72/25	74/13					
OPRIN	003335		70/12	71/01	71/48	72/33			
OP.EN	000220		22/27	70/22	74/01	74/14			
OTOTL	005417		102/16	103/37	107/38				
P1777	004121		80/26	80/33	80/42	82/11			
P3777	003702		76/35	77/36					
PAC?1	003700		76/31	76/46	77/10	77/34	78/02	78/18	78/36
			79/06						
PAC?2	003701		76/32	76/47	77/03	77/06	77/35	78/03	78/27
PATI	005635		105/16	106/05	107/41				
PC171	003734		77/40	78/12					
PC172	004044		78/06	78/15	78/28	79/32	82/19	83/08	
PC175	004123		80/38	80/44	82/13	82/44			
PL470	004025		78/41	79/16	80/02	80/35			
PC670	004024		77/02	77/23	77/27	79/15	82/26		
PC7?	003732		77/48	78/10					
PCRLF	000060		23/11	24/15	66/09	76/01	95/22	97/16	179/01
			186/01						
PC.40	004276		83/03	83/55					
PDECI	000062		23/13	24/18	29/12	179/01	181/37	181/42	186/01
PDE?	003645		23/13	24/12	77/06				
PENDA	000132		24/30	40/11	40/12	55/10			
PEX?T	003632		76/46	78/22					
PFAIL	000120		24/20	31/02	85/12	85/27	88/05		
PFAIS	000121		24/21	25/14	31/22	38/16			
PFTEX	004566		31/07	90/09					
PLP?T	004100		80/10	80/25					
PLS?T	003631		76/45	77/17	78/07				
PUCTX	003532		74/18	74/20	74/25	75/34			
PUC?	003641		24/16	47/04	77/02				
PUDT	011312		184/30	184/37					
PUNES	005034		105/06	105/27	105/29	107/21	107/40		
PKCD	003567		73/22	75/17					
PRIDG	003543		74/36						
PKOGK	001405		33/07	34/26	34/32	40/43			
PRSTA	001151		29/12						
PKSTL	001171		30/04	30/13	30/21	31/20			
PSTRT	000131		24/29	29/12	40/07	40/25	41/04	52/16	55/10
			65/09	90/01					
PTB?	004047		77/15	79/37	81/46				
PWRS0	001301		31/03	31/08	31/26				
PWRUP	001251		24/20	31/02	31/03				
P?OCT	000061		23/12	24/17	39/30	89/10	179/01	186/01	
P.MB1	004126		80/08	80/47					
P.MB5	004127		80/08	80/48					
P.MS0	004125		80/07	80/14	80/25	80/46			
P?TTO	004050		80/01	80/30	80/32				

XDIV	010661		173/31	174/31	176/01				
XDIV1	010666		176/06	176/11					
XMCOM	010724		177/10	177/17					
XMDIV	010717		165/43	172/14	174/07	177/12			
XMDK	010771		178/01	178/06	178/26				
XMMUL	010705		165/40	172/12	174/11	177/01			
XMRET	010736		177/01	177/12	177/20	177/21			
XJ15	003731		78/04	78/09					
XM20	007043		132/18	132/31					
XMS32	001461		36/10	36/36	36/39	37/01			
XMUL	010645		173/09	174/47	175/24	175/30			
XMUL1	010655		175/32	176/12					
XMVET	001500		37/03	37/13	37/14				
XNTPA	002620		54/05	54/24	54/27				
XUAC	010471		172/04	172/08	173/06				
XUMD	010473		172/06	172/10	173/08				
XUMQ	010472		172/05	172/09	173/07				
XUR1	000040	MC	116/11	129/38	130/13	131/10	131/31		
XUR2	000042	MC	116/14	129/35	133/09	133/30	135/09		
XURA	000036	MC	116/08	129/12	129/15	130/10	130/34	134/08	134/27
			135/12	135/15	135/18	135/21	135/24	135/27	135/30
XURTE	007041		132/19	132/26	132/29				
XUR.0	007007		129/13	129/16	130/11	130/35	132/01	134/09	134/28
			135/13	135/16	135/19	135/22	135/25	135/28	135/31
			135/43						
XUR.1	007015		129/39	130/14	131/11	131/32	132/08		
XUR.2	007025		129/36	132/17	132/27	133/10	133/31	135/10	
XUR.4	007042		132/01	132/06	132/08	132/15	132/17	132/28	132/30
XUR.L	007130		135/43	139/13					
XKMSE	002064		43/14	43/40	43/42	43/47			
XSYTB	001705		40/37	40/45					
XTINI	001566		38/06	38/12	38/24	38/34			
XUBL3	001522		37/24	37/37	37/38				
ZUC?	003640		23/12	24/13	77/01				
ZSU?P	003677		77/11	77/18	77/28	77/33	78/19	78/31	78/33
			78/50	78/52	79/05				
.BDEG	000122		22/33	24/22	30/22	30/28	30/39		
.CB03	007371		144/02	144/10	144/11	144/15	144/24		
.CB20	007372		144/03	144/12	144/16				
.CB99	007362		144/04	144/08					
.CC98	007377		144/18	144/23					
.DIVU	000044	MC	116/17	146/13					
.MPYA	000046	MC	116/20	146/16					
.MPYU	000056	MC	116/32	141/14	142/14				