DataGeneral

DIAGNOSTIC LISTING

LISTING

096-000422-00

PROGRAM

MICRO NOVA SC MEMORY TEST

TAPE

095-000422-00

ABSTRACT

THE SC MEMORY TEST CONSISTS OF A SERIES OF SC MEMORY TESTS AND A SIMPLE 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 MNSCM MACRO REV 04.00              13:00:33 12/03/76
01
02
03
04
05
06
07              ;*****************************************************************
08              ;
09              ; NAME: MNSCMT.SR                          PART NUMBER: 094-000833
10              ;
11              ;
12              ; DESCRIPTION: MICRO NOVA SC MEMORY 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              ;*****************************************************************
```

```
10002 MNSCM
01              ;       MICRO-NOVA SC-MEMORY TEST
02              ;1.     ABSTRACT
03              ;               THE SC-MEMORY TEST CONSISTS OF A SERIES
04              ;               OF SC-MEMORY TESTS AND A SIMPLE
05              ;               SUPERVISOR PROGRAM. (THE DIAGNOSTIC LINKER)
06              ;
07              ;               THE DIAGNOSTIC LINKER IS A PROGRAM
08              ;               DESIGNED TO "LINK" THE VARIETY OF
09              ;               SC MEMORY TESTS.
10              ;
11              ;
12              ;2.     MACHINE REQUIREMENTS
13              ;2.1    MICRO-NOVA PROCESSOR WITH 4 TO 32K OF
14              ;       READ/WRITE MEMORY
15              ;               (ALLOWS FOR EXPANSION IN 1K INCREMENTS
16              ;               BUT MEMORY MUST BE CONTIGUOUS)
17              ;
18              ;2.6    PREREQUISITES
19              ;2.6.2  SOFTWARE PREREQUISITES
20              ;               THE MICRO-NOVA LOGIC TEST SHOULD HAVE
21              ;               BEEN RUN BEFORE ATTEMPING THIS TEST.
22              ;3.       SWITCH SETTINGS
23              ;3.1      AUTO-SIZE AND GO START AT 200
24              ;3.2      MANUAL SELECT/DELETE TESTS START AT 206
```

```
01              ;3.4      KEY ENTERED OPTIONS
02              ;
03              ;       KEY T           SETS SWREG2 BIT 0 = 1
04              ;                       WILL ALLOW THE OPERATOR TO
05              ;                       SET LOWER LIMIT OF TEST(PHSLO)
06              ;                       AND UPPER LIMIT OF TEST(PHSHI)
07              ;                       VALUE ENTERED SHOULD BE
08              ;                       A MODULO 1K DECIMAL NUMBER .
09              ;
10              ;               ENTERED LIMITS MUST:
11              ;               1. FALL WITHIN THE MEMORY LIMITS
12              ;                  SIZED BY THE PROGRAM.
13              ;               2. NOT INCLUDE AREA WHICH THE PROGRAM
14              ;                  OCCUPIES.
15              ;               3. FOR PHSHI BE GREATER THAN OR EQUAL
16              ;                  TO PHSLO OR THE LIMITS WILL NOT
17              ;                  BE ACCEPTED.
18              ;
19              ;                       EXAMPLE:
20              ;                       TO TEST AREA BETWEEN 16K
21              ;                       AND 32K ENTER:
22              ;                       (PHSLO,PHSHI) 16,31
23              ;
24              ;                       TERMINATE INPUT LIMITS WITH A CARRIAGE
25              ;                       RETURN.
26              ;
27              ;       KEY 0   SW0=1   ENTER THE KEY PARAMETERS
28              ;                       UNTIL A CR IS INPUTTED,
29              ;                       EACH KEY WILL COMPLEMENT
30              ;                       THE PREVIOUS SELECTION
31              ;                       OF THE SAME KEY.
32              ;       KEY 1   SW1=1   CONTINUE FROM ERROR,NO LOOPING
33              ;       KEY 2   SW2=1   INHIBIT ALL TTO TYPEOUTS
34              ;       KEY 4   SW4=1   INHIBIT PASS # PRINTOUT
35              ;       KEY 6   SW6=1   HALT AFTER ERROR TYPEOUT
36              ;                       TYPE ANY KEY TO CONTINUE
37              ;       KEY Q           WILL CAUSE MARCH/GALPAT/GALWREC TO USE
38              ;                        ALL 1'S DATA PATTERN ON EACH PASS.
39              ;                       SWRG2 BIT 2 = 1
40              ;       KEY P           PRINT PASSED FOR EACH TEST COMPLETED
41              ;                       SUCESSFULLY, SWRG2 BIT 1 = 1
42              ;       KEY (C)S        PRINTS STATUS OF EACH TEST WHICH
43              ;                       HAS COMPLETED A PASS, SWRG2 BIT 15 = 1
44              ;       KEY (C)D        WILL SET SWREG TO DEFAULT MODE
45              ;                       ALL BITS = 0
46              ;                       AND WILL RESTART THE PROGRAM.
47              ;       KEY M           LIST CURRENT OPERATING MODES
48              ;       KEY (C)R        WILL RESTART THE PROGRAM
49              ;                       WITHOUT MODIFING THE SWREG
50              ;
51              ;       WHERE (C) IS A CONTROL KEY
```



```
01              ;4.     OPERATING PROCEDURES
02              ;4.1      LOAD THE PROGRAM VIA THE BINARY LOADER
03              ;4.2      SET SWITCHES TO:
04              ;               200 FOR AUTO SIZE AND GO
05              ;               206 FOR MANUAL SELECT/DELETE
06              ;
07              ;
08              ;4.3      PRESS START
09              ;4.4      PROCESSOR WILL TYPE:
10              ;               MICRO-NOVA SC-MEMORY TEST
11              ;               MEM SIZE #1K'S
12              ;               PROGRAM RUN LIST
13              ;               PROG#   DESCRIPTION
14              ;
15              ;4.5      IF START WAS 200 THE LIST OF
16              ;               PROGRAMS TO BE RUN SEQUENTIALLY WILL
17              ;               THEN BE LISTED AND THE TEST PROGRAM
18              ;               WILL AUTO START.
19              ;               NOTE: GALPAT AND GALWREC WILL
20              ;               NOT BE SELECTED ON AN AUTO START.
21              ;
22              ;4.6      IF START WAS 206 LINKER WILL
23              ;               PAUSE AT THE END OF EACH TEST
24              ;               DESCRIPTION AND WAIT FOR KEYBOARD
25              ;               INPUT.  TYPING IN A SPACE WILL
26              ;               ENABLE THAT TEST TO BE RUN.
27              ;               TYPING IN ANY OTHER CHARACTER WILL
28              ;               DELETE THAT TEST FROM BEING RUN
29              ;4.8    THE PROCESSOR WILL THEN WAIT FOR THE OPERATOR TO SET
30              ;               ANY BIT SWITCH OPTIONS, TEST WILL START
31              ;               AFTER PRESSING ANY CHARACTER ON THE KEYBOARD.
32
```

```
100005 MNSCM
01
02              ;5.     ERROR DESCRIPTION
03              ;               MOST ERRORS DETECTED BY EITHER
04              ;               THE INDIVIDUAL TESTS OR
05              ;               BY THE DIAGNOSTIC LINKER WILL
06              ;               RESULT IN AN ERROR TYPEOUT. SOME
07              ;               SMALL NUMBER OF HIGHLY IMPROBABLE
08              ;               ERRORS MAY RESULT IN A PROGRAM HALT
09              ;               IF THEY ARE OF A NATURE THAT THE LINKER
10              ;               CAN'T RECOVER FROM AND LOGICALLY PROCEED.
11              ;
12              ;
```

```
100006 MNSCM
01              ;5.2    ERROR FORMAT
02              ;       EACH TEST WILL OUTPUT AN UNIQUE ERROR
03              ;       TYPEOUT INCLUDING TEST NAME,DATA ASSOCIATED
04              ;       WITH ERROR,ERROR LOCATIONS,SCRATCH LIMITS
05              ;       USED FOR THIS PASS OF THE TEST,AND THE
06              ;       MEMORY LIMITS SELECTED TO BE EXERCISED
07              ;       BY EITHER THE OPERATOR OR THE PROGRAM.
08              ;
09              ;       DEFINITION OF ERROR PRINTOUT TERMS:
10              ;
11              ;       C(X)    =  CONTENTS OF LOCATION X
12              ;       LOC(X)  =  ADDRESS OF LOCATION X (LOGICAL OR PHYSICAL)
13              ;       C/(X)   =  COMPLEMENT OF THE CONTENTS OF LOCATION X
14              ;       SCRLO/HI=  SCRATCH LIMITS OF THIS PASS OF THE TEST
15              ;                  (LOGICAL OR PHYSICAL)
16              ;       TSTLO/HI=  SCRATCH LIMITS EXPRESSED IN DECIMAL 1K'S
17              ;       PHSLO/HI= ENTERED,OR PROGRAM SELECTED,MEMORY LIMITS
18              ;                 TO BE EXERCISED IN DECIMAL 1K'S.
19              ;
20              ;       EXAMPLES:
21              ;
22              ;       ERROR TYPE OUT
23              ;       ----------------------------------
24              ;
25              ;       GALPAT
26              ;       C(Y)    C(X)    LOC(X)  LOC(Y)
27              ;       000000  000100  016010  017345
28              ;       SCRLO/HI        016000  017345
29              ;       TSTLO/HI        7       7
30              ;       PHSLO/HI        3       31      TYPE ANY KEY
```

```
10007 MNSCM
01              ;
02              ;
03              ;
04              ;               IF SW2=1 THE
05              ;               TEST WILL HALT WAITING FOR
06              ;               THE OPERATOR TO PRESS A KEY
07              ;               ON THE CONSOLE
```

```
10008 MNSCM
01
02              ; 6.0   PROGRAM INITIALIZE
03              ;               THE DIAGNOSTIC LINKER INITIALIZES ITSELF
04              ;               AND INDIVIDUAL TESTS IN THE FOLLOWING
05              ;               SEQUENCE:
06              ;                  1. SYSTEM IS RESET.
07              ;                  2. ANY OTHER NECESSARY CONSTANTS
08              ;                       ARE INITIALIZED
09              ;                  3. MEMORY IS SIZED IN 1K INCREMENTS
10              ;                       FROM 0 TO 32K
11              ;
12              ;                  4. LINKER THEN TYPES THE PROGRAM
13              ;                       NAME AND REVISION LEVEL,SYSTEM SIZE,
14              ;                       THE PROGRAM RUN LIST (AND WILL ALLOW THE
15              ;                       OPERATOR  TO SELECT OR DELETE SPECIFIC T
16              ;                       IF START WAS 200
17              ; 6.1 OPTION SELECTION
18              ;
19              ;               IF THE PROGRAM WAS NOT AUTOSTARTED
20              ;               (NOT LOC 200) THE LINKER WILL PRINT
21              ;               "OPTIONS?" AND WAIT FOR A CARRIAGE RETURN
22              ;               TO START EXECUTING THE TESTS.
23              ;               THIS ALLOWS THE OPERATOR TO SET UP
24              ;               THE KEY ENTRY OPTIONS INCLUDING
25              ;               KEY "T" WHICH ALLOWS SETTING OF MEMORY
26              ;               TEST AREA LIMITS.
```

```
10009 MNSCM
01              ;6.3    PROGRAM EXECUTION
02              ;               ONCE THE LINKER HAS COMPLETED ALL
03              ;               INITIALIZATION THE FOLLOWING SERIES
04              ;               OF OPERATIONS IS LOOPED THROUGH
05              ;
06              ;
07              ;                  1. LINKER SEARCHES THRU LIST OF TESTS
08              ;                       UNTIL IT FINDS ONE WHICH IS
09              ;                       NOT DELETED.
10              ;
11              ;                  2. LINKER THEN SETS UP SEGMENT SIZE
12              ;                     BAISED ON THE VALUE IN THE
13              ;                       PARAMETER TABLE FOR EACH TEST.
14              ;
15              ;                  3.   THE LINKER THEN SETS SCRLO AS THE
16              ;                       BEGINNING OF THE SEGMENT TO BE TESTED
17              ;                       AND SCRHI AS THE END OF THE SEGMENT.
18              ;
19              ;                  4.   THE LINKER RE-ENTERS THE TEST WITH
20              ;                       EACH SEGMENT UNTIL THE AREA SELECTED
21              ;                       HAS BEEN EXERCISED. AFTER COMPLETION
22              ;                       THE LINKER SEARCHES FOR ANOTHER TEST
23              ;                       IN THE SERIES.
24              ;
25              ;
26              ;               5.      AFTER SEVERAL PASSES OF EACH TEST
27              ;                       SELECTED THE LINKER WILL PRINT
28              ;                       "PASS XX" IF SWREG BIT 4 HAS NOT
29              ;                       BEEN SET.
30              ;
31              ;               6.      IF PROGRAM WAS LOADED FROM DTOS WITH
32              ;                       EITHER CAT OR KITTEN IT WILL START
33              ;                       CAT/KITTEN AFTER FIRST "PASS".
34              ;
```

```
10010 MNSCM
01              ;7.0 TEST DESCRIPTIONS
02              ;
03              ;
04              ;       7.2 DATA EQUALS ADDRESS TEST
05              ;
06              ;               THIS TEST WRITES THE ADDRESS OF EACH
07              ;               LOCATION INTO EACH LOCATION AS DATA
08              ;
09              ;               IT THEN READS BACK ALL LOCATIONS AND CHECKS
10              ;               THE VALUE READ AGAINST THE ADDRESS.
11              ;
12              ;       7.3 ISZ INSTRUCTION TEST
13              ;
14              ;               7.3.1 FORWARD ISZ TEST
15              ;
16              ;               THIS TEST FILLS ALL SCRATCH WITH A MINUS
17              ;               ONE PATTERN, THEN PERFORMS A ISZ INSTR.
18              ;               FOLLOWED BY A READ OF THE LOC. TO
19              ;               VERIFY IT CONTAINS A ZERO. THIS IS
20              ;               DONE AT EACH LOCATION FROM SCRLO TO
21              ;               SCRHI.
22              ;
23              ;               7.3.2 REVERSE ISZ  TEST
24              ;
25              ;               THIS TEST IS IDENTICAL TO THE ABOVE TEST
26              ;               EXCEPT THAT THE MEMORY ADDRESSES ARE SCANED
27              ;               FROM SCRHI TO SCRLO.
28              ;
29              ;
30              ;
31              ;       7.4 MARCH
32              ;
33              ;               THIS TEST FUNCTIONALLY CHECKS EACH BIT IN THE
34              ;               MEMORY AND THE ADDRESSING .
35              ;
36              ;               THIS TEST USES EITHER RANDOM DATA (KEY OPTION "Q
37              ;               OR ALL ONES DATA (OPTION "Q"= 1).
38              ;
39              ;               A TEST PATTERN IS WRITTEN INTO THE BACKGROUND
40              ;               STARTING AT SCRLO AND ENDING AT SCRHI.
41              ;               ADDRESSING IS THEN SCANNED ACROSS THIS RANGE
42              ;               AND AT EACH ADDRESS THE TEST WORD IS
43              ;               READ AND A COMPLEMENTED TEST WORD IS
44              ;               WRITTEN BACK INTO THE SAME LOCATION.
45              ;
46              ;               THE DATA IS THEN COMPLEMENTED AND THE ABOVE
47              ;               SEQUENCE REPEATED.
48              ;
49              ;               THE PROCESS IS THEN REPEATED
50              ;               STARTING AT SCRHI AND PROCEEDING TO SCRLO .
```

7.5 MASEST (SEGMENT MAX. = 2)

CHECKS FOR DESTRUCTION OF STORED DATA THAT MAY RESULT FROM MULTIPLE SELECTION OF ADDRESSES INTERNAL TO THE MEMORY CAUSED BY FAULTY DECODERS OR LOGICAL SWITCHING HAZARDS.

ALTERNATE ALL ONE'S, ALL ZERO'S ARE WRITTEN IN ASCENDING LOCATIONS, EACH LOCATION IS THEN READ AND VERIFIED WHILE GOING THRU THE ADDRESS SEQUENCE OF ADDRESS, COMPLEMENT OF ADDRESS, ADDRESS PLUS ONE, COMPLEMENT OF ADDRESS PLUS ONE, ETC. . MEMORY IS THEN READ SEQUENTIALLY AND THE ALTERNATE ONE'S ZERO'S PATTERN VERIFIED.

7.6 SLDIAG (SEGMENT MAX = 1)

THIS TEST IS NOT SELECTED IF THE PROGRAM IS STARTED AT LOC 200.

THIS TEST FILLS THE BACKGROUND WITH ALL 0'S PATTERN AND THEN WRITES A DIAGONAL PATTERN USING THE COMPLEMENT OF THE BACKGROUND.

THE ARRAY IS THEN VERIFIED READING DOWN EACH COLUMN INSTEAD OF ACROSS EACH ROW.

IF NO ERRORS ARE FOUND THE PROCCES IS REPEATED WITH THE DIAGONAL SHIFTED ONE POSITION UNTIL ALL POSITIONS ARE USED.

NEXT THE BACKGROUND IS COMPLEMENTED AND THE TEST REPEATED.

7.7 GALPAT (SEGMENT MAX = 1)

THIS TEST DOESN'T RUN ON AN AUTOSTART.

THIS TEST CHECKS ADDRESSING, INTERACTION BETWEEN BITS, AND PATTERN AND SEQUENCE DEPENDENCY FOR TRANSIENT PERFORMANCE.

THIS TEST EIMER USES RANDOM DATA OR ALL ONES (SEE KEY OPTION "Q")

A BACKGROUND PATTERN IS WRITTEN THRU-OUT MEMORY. THEN STARTING AT THE FIRST LOCATION, A TEST WORD IS WRITTEN(COMPLEMENT OF BACKGROUND).
MEMORY IS THEN READ IN ALL LOCATIONS IN THE FOLLOWING SEQUENCE: BACKGROUND, TEST WORD NEXT BACKGROUND, TEST WORD, NEXT BACKGROUND, ETC.
AFTER COMPLETION OF A PASS FROM SCRLO TO SCRHI , THE TEST WORD IS MOVED TO THE NEXT SEQUENTIAL LOCATION AND THE PROCESS OF READING REPEATED.
THIS CONTINUES UNTIL THE TEST WORD HAS BEEN LOCATED IN EVERY MEMORY LOCATION FROM SCRLO TO SCRHI.

AT COMPLETION,THE ABOVE SEQUENCE IS REPEATED USEING A COMPLEMENTED PATTERN.

FOR SAKE OF TYPEOUTS:
LOC(X) = TEST WORD LOCATION
LOC(Y) = BACKGROUND LOCATION

7.8 GALWREC (SEGMENT MAX = 1)

THIS TEST DOESN'T RUN ON AN AUTOSTART.

THIS TEST CHECKS ALL POSSIBLE WRITES FOLLOWED BY READS AT DIFFERENT LOCATIONS

THIS TEST EITHER USES RANDOM DATA OR ALL ONES (SEE KEY OPTION "Q")

A BACKGROUND PATTERN(B) IS WRITTEN THRU-OUT MEMORY. EVERY PAIR OF ADDRESSES ARE THEN CHECKED IN THE FOLLOWING MANNER, STARTING WITH THE FIRST LOCATION,LOC(X):
WRITE T(INVERTED B) IN LOC(Y)=(X+1),READ B IN LOC(X),WRITE B IN LOC(Y),
READ B IN LOC(X),WRITE T IN LOC(Y)=(Y+1)
READ B IN LOC(X),ETC.

AFTER ALL LOC. HAVE BEEN CHECKED IN RELATION TO LOCATION ONE(X),THE SEQUENCE IS REPEATED WITH RESPECT TO LOC(X)=(X+1), ETC.

AT COMPLETION , THE ABOVE SEQUENCE IS REPEATED USEING A COMPLEMENTED PATTERN.

```
 0013 MNSCM
 01
 02        000000          .LOC 0
 03 000000 005225          DIRT
 04        000045          .LOC 45
 05 000045 000060          EGGS
 06        000060          .LOC 60
 07 000060 000000 EGGS:    0       ;AUTO-RUN SWITCH
 08 000061 000000          0       ;DEVICE CODE
 09 000062 000000          0       ;CAT SWITCH
 10 000063 000000          0       ;NUMBER OF PASSES THIS RUN
 11 000064 000000          0       ;RETURN ADR OF DTOS/EDTUS/PDTOS
 12 000065 000000 SWREG:   0       ;SOFTWARE SWITCH REGISTER FOR OPTIONS
 13
                           .TITL   LINKR
 15                ;DIAGNOSTIC PROGRAM LINKER
 16                ;LINKS TO TYPEOUT ROUTINES
 17                ;PAGE 0 LINKS
 18 000066 000046 TPLOC: 46                ;TRAP PC STORAGE
 19 000067 000047 TPADR: 47                ;TRAP THRU THIS LOCATION
 20 00070  001633 IPDEC:  PDE?
 21 000071 001577 IMESS:  MES?
 22 00072  001626 IZOCT:  ZOC?
 23 00073  001627 IPOCT:  POC?
 24 000074 001710 ICRLF:  CLF?
 25        000073 PDECI= IPDEC     ;DECIMAL PRINT
 26        000071 LMESS= IMESS     ;TEXT PRINTER
 27        000073 LPOCT= IPOCT     ;PRINT OCTAL
 28        000072 PZOCT= IZOCT     ;ZERO SUPPRESS OCTAL
 29        000074 PCRLF= ICRLF
 30        000074 ICLF?=PCRLF
 31        000070 IPDE?=PDECI
 32        000072 IZOC?=PZOCT
 33 000075 002516 ERRTX: ERTXT     ;TEXT TYPEOUT CALL
 34 000076 001363 RETRN: LRETP     ;RETURN FROM TEST
 35 000077 002270 RANG:  RANGN     ;RANDOM # GENERATOR
 36 000100 002337 ERROI:  ERROH    ;INIT ERROR TYPEOUTS
 37 000101 002431 ERRUX:  ERROR    ;ERROR HANDLER NO PRINT OF AC3
 38 000102 002426 ERRUY:  ERRO3    ;ERROR HANDLER,PRINTS AC3 ALSO
 39 000103 004066 LWRBK: WRBKG             ;FILL BACKGROUND SUBR.
 40 000104 000141 TIBL: LZMAX              ;START OF TEST TABLES
 41 000105 004776 LPGMK:  PRGGK    ;# OF PGMS TO INIT
 42                ;TEST PARAMETER LOCATIONS
 43 000106 000004 K4:     4
 44 000107 000037 K37:    37
 45 000110 076000 K76K:   76000
 46 000111 000000 DTS1K:  0        ;BLOCK WHICH DTOS RESIDES IN
 47 000112 000000 PENDA:  0        ;LAST LOC OF TEST
 48 000113 000000 CURPR:  0
 49 000114 000000 PSTRT: 0         ;FIRST LOC OF TST
 50 000115 000000 PRFLG: 0         ;=0, NO PRINT OF AC3
 51 000116 000000 PR.S3: 0         ;AC3 SAVED FOR THE ERROR PRINT SUBR.
 52 000117 000000 SEGMT:  0        ;TEST SEGMENT SIZE
 53 000120 000000 SWRG2:  0        ;SECOND SOFTWARE SWITCH REGISTER
 54 000121 000000 ITRER:  0        ;ITERATION ERROR SWITCH
 55 000122 000000 RTIM:   0        ;REAL TIME
 56 000123 000000 STATS: 0
 57 000124 000000 ER50:   0        ;FLAGS WHEN 50 ERRORS ARE REACHED
 58 000125 000000 LASTI:  0        ;TTI CHARACTER SAVED
 59                                ;BIT 0=1 IF NOT YET SERVICED
 60 000126 000000 KATSW:  0        ;-1 IF CAT/KITTEN HAS BEEN STARTED
```

```
 0014 MNSCM
 01 000127 000000 ADFLG: 0         ;=-1,PRINT ADDRESS LIMITS
 02 000130 000000 ERTOT: 0         ;ERROR ACCUMULATOR
 03                ;SCRATCH AREA SIZE PARAMETER LOCS FOR TEST USAGE
 04 000131 000000 PHSLX: 0         ;LOWEST PHYSICAL MEMORY TESTABLE (OCTAL 1K'S)
 05 000132 000000 PHSHX: 0         ;HIGHEST PHYSICAL MEMORY TESTABLE   "
 06 000133 000000 PHSLO:  0        ;ADJUSTED LO PHYSICAL LIMITS              "
 07 000134 000000 PHSHI:  0        ;ADJUSTED HI PHYSICAL LIMIT               "
 08 000135 000000 TSTLO:  0        ;PHYSICAL LO LIMIT FOR THIS TEST PASS     "
 09 000136 000000 TSTHI:  0        ;PHYSICAL HI LIMIT FOR THIS TEST PASS   "
 10 000137 000000 SCRLO: 0         ;CURRENT LOWEST LOGICAL SCRATCH ADDRESS
 11 000140 000000 SCRHI: 0         ;CURRENT HIGHEST LOGICAL SCRATCH ADDRESS
 12        000141 LZMAX=SCRHI+1
 13        000141 LPG0=.
```

```
10015 MNSCM
01
02                ;NOVA3/MOM INSTRUCTION SET
03        061401 .DIAC PSH, = 61401
04        061601 .DIAC POP, = 61601
05        061001 .DIAC MTSP = 61001
06        060001 .DIAC MTFP = 60001
07        061201 .DIAC MFSP = 61201
08        000002 .DUSR MAP = 2
09        000004 .DUSR PRTY = 4
10        061077 .DUSR I.RST = DOA 0,77
11        071077 .DUSR RTCEN = DOA 2,77
12        065077 .DUSR RTCDS = DOA 1,77
13
```

```
10016 MNSCM
01
02                .LOC 200
03 000200 000202 STRT1:  JMP @BGNADR
04 000201 000000 ITRET:  0       ;
05 000202 000210 BGNADR: LINKR           ;SIZE AND GO
06 000203 000000 PASS: 0
07 000204 000004 PASSIN: 4
08 000205 000004 PASSVL: 4
09 000206 000207 STRT2:  JMP .+1
10 000207 000211         LINKM           ;SIZE AND WAIT FOR SELECTIONS
11                ;LINKER MAIN LINE DISPATCH ROUTINE
12 000210 102401 LINKR:  SUB 0,0,SKP     ;AUTOSTART ENTRY
13 000211 102000 LINKM:  ADC 0,0         ;MANUAL ENTRY START
14 000212 152400         SUB 2,2
15                LINKP:
16 000213 044230         STA 0,LAUTO     ;SET ENTRY TYPE SW
17 000214 004220         JSR LDSPR       ;START DISPATCH
18 000215 000232         LTBL1   ;THROUGH INIT SEQ
19 000216 004220 LRUNS:  JSR LDSPR
20 000217 000242         LTBL2   ;THROUGH RUN TABLE
21                ;DISPATCH ROUTINE
22                ;ENTER SUBROUTINES IN SEQ VIA TABLE SPEC BY (R3)
23                ;END OF EACH TABLE IS LRUNS WHICH WILL START US
24                ;BACK AT THE BEGINNING OF THE RUN TABLE
25 000220 021400 LDSPR:  LDA 0,0,3       ;ADRS OF DISPATCH TABLE
26 000221 040227         STA 0,LIDIS
27 000222 036227 LDS.1:  LDA 3,@LIDIS
28 000223 005400         JSR 0,3
29 000224 005231         JSR @KEY.I      ;CHECK FOR TTI REQUESTS
30 000225 010227         ISZ LIDIS
31 000226 000222         JMP LDS.1
32 000227 000000 LIDIS: 0
33 000230 000000 LAUTO: 0
34 000231 001502 KEY.I:  CKKEY
```

```
10017 MNSCM
01                      .MACRO NEXTT
02              ;DEFINITION TO LINKR PARAMETERS FOLLOWS
03              LMEML=.
04                      .LOC LPG0
05                      ^1.00
06              LPG0=.
07                      .LOC LMEML
08                      0                       ;TST ERR CTR THIS PASS
09                      0                       ;TEST PASS COUNTER
10                      0                       ;TEST ERROR COUNTER
11              ^1.00:  JMP ^1.02                       ;EXECUTE ENTRY ADDRESS
12                      0                               ;WAIT SWITCH
13                      ^2                              ;MAXIMUM SEGMENT SIZE
14                      .TXTE 1^31
15              %
16
17              .MACRO LCALL
18                      **A1-PDEC1*1B11+100010
19              %
```

```
10018 MNSCM
01                      ;LTBL1-INIT SYSTEM DISPATCH TABLE
02                      ;END OF TABLE IS LRUNS
03  00232 000500 LTBL1:  LSYSR    ;RESET SYS
04  00233 000653         LSIZE    ;SIZE MEMORY
05  00234 001230         TINIT    ;INIT TESTS TO BE RUN
06  00235 003667         LPRSL    ;LISTS TESTS TO BE RUN
07  00236 000252         BTSW0    ;ALLOW OPERATOR TO INPUT SWITCH OPTIONS
08  00237 000216         LRUNS
09  00240 000000         0
10  00241 000000         0
11                      ;
12                      ;
13                      ;LTBL2-RUN SYS DISPATCH TABLE
14  00242 000715 LTBL2:  LRANP    ;SEQUENTIAL PROG SELECT
15  00243 001302         SETML    ;SET LO/HI MEMORY LIMITS IF BIT 0=1
16  00244 001126         EXCTS    ;EXECUTE TEST
17  00245 001367         LSTAT    ;ADJUST RUN STATISTICA
18  00246 000216         LRUNS
19  00247 000000         0
20  00250 000000         0
21                      ;
22                      ;
23                      ;BTSW0-ALLOW OPERATOR TO INPUT THE SWITCH OPTION BEFORE STARTING
24                      ;
25                      ;
26  00251 000230         LAUTO
27  00252 054274 BTSW0:  STA 3,BTSV3      ;SAVE AC3 FOR RETURN
28  00253 022251         LDA 0,@BTSW0-1   ;GET LAUTO SW
29  00254 100004         COM 0,0,SZR      ;SKP IF NOT AUTOSTART
30  00255 002274         JMP @BTSV3       ;RETURN
31  00256 006071         JSR @LMESS       ;PRINT OPERATOR MESSAGE
32  00257 002642         TXT,4            ;TEXT ADDRESS
33  00260 060177         INTEN
34  00261 020125 BT.CK:  LDA 0,LASTI      ;CK IF REQUESTED YET
35  00262 101133         MOVZL# 0,0,SNC   ;SKP=REQUEST
36  00263 000261         JMP .-2          ;NOT YET
37  00264 024275         LDA 1,BT.CR
38  00265 100415         SUB# 0,1,SNR     ;WAS IT A CR?
39  00266 000271         JMP .+3          ;NOT YET
40  00267 000231         JSR @KEY.I       ;SERVICE REQUEST
41  00270 000261         JMP BT.CK
42  00271 102400         SUB 0,0
43  00272 040125         STA 0,LASTI      ;CLR REQ. BIT
44  00273 002274         JMP @BTSV3       ;RETURN TO LINKR
45
46  00274 000000 BTSV3: 0                ;SAVE RETURN ADDRESS LOC.
47  00275 100015 BT.CR:  100015
```

```
10019 MNSCM
01
02          000500              .LOC 500
03                      ;LSYSR=RESET SYSTEM
04 000500 061077 LSYSR:  I.RST              ;IORST
05 000501 102400         SUB 0,0
06 000502 040124         STA 0,ER50         ;ZERO 50 ERROR FLAG
07 000503 040120         STA 0,SWRG2        ;DEFAULT MODE SWREG2
08 000504 040203         STA 0,PASS         ;START WITH PASS 0
09 000505 040125         STA 0,LASTI        ;CLR START FLG
10 000506 040130         STA 0,ERTOT        ;CLR ACCUM ERRS
11 000507 040113         STA 0,CURPR        ;SET CURRENT PGM TO 0
12 000510 040126         STA 0,KATSW        ;RESET CAT/KITTEN SWITCH
13 000511 020205         LDA 0,PASSVL       ;PASS CNT VALUE
14 000512 024062         LDA 1,EGGS+2       ;CAT SW
15 000513 125005         MOV 1,1,SNR        ;SKIP IS USING CAT
16 000514 000402         JMP .+2
17 000515 103120         ADDZL 0,0          ;*4 FOR CAT PASSES
18 000516 040205         STA 0,PASSVL       ;UPDATE TO 4 TIMES OLD VALUE
19 000517 020441         LDA 0,LSYEE        ;GET END OF PGM ADDRESS
20 000520 024110         LDA 1,K76K
21 000521 123400         AND 1,0
22 000522 101220         MOVZR 0,0
23 000523 101220         MOVZR 0,0
24 000524 101300         MOVS 0,0
25 000525 101400         INC 0,0
26 000526 040133         STA 0,PHSLO        ;SAVE IT IN PHSLO
27 000527 040131         STA 0,PHSLX        ;PLACE IN PHSLX
28 000530 020414         LDA 0,LC.K4
29 000531 061001         MTSP 0
30 000532 060001         MTFP 0             ;INIT STACK TO LOC 400
31 000533 020415         LDA 0,STKIN
32 000534 040003         STA 0,3            ;LOC 3 FOR STACK ERRORS
33 000535 020411         LDA 0,LC.K8
34 000536 040001         STA 0,1            ;FOR INTRAS
35 000537 020410         LDA 0,LC.K9
36 000540 040002         STA 0,2            ;FOR RTC INTR
37 000541 020404         LDA 0,LC.K5
38 000542 042067         STA 0,@TPADR       ;SETUP FOR TRAPS
```

```
10020 MNSCM
01 000543 001400         JMP 0,3
02 000544 000400 LC.K4:  400
03 000545 001062 LC.K5:  CDISP
04 000546 000576 LC.K8:  LINTR
05 000547 000570 LC.K9:  RTCIN
```

```
10021 MNSCM
01                      ;STACK FAULT HANDLER
02 00550 000551 STKIN:  .+1
03 00551 000071         JSR @LMESS      ;STACK FAULT MESS
04 00552 000561         STKTX
05 00553 063077         HALT            ;HALT IF STACK ERROR
06 00554 024001         JMP @.+1
07 00555 000211         LINKM
08 00556 001550 LITI: TT.TI
09 00557 000074 LS.74: 74
10 00560 004777 LSYEE: LSYSE    ;END OF PGM ADDRESS
11 00561 005215 STKTX:  .TXTE [<15><12>STACK FAULT]
12
13
14                      ;MICRO-NOVA RTC INTERRUPT HANDLER
15 00570 021122 RTCIN:  ISZ RTTIM
16 00571 101000         MOV 0,0
17 00572 021122         ISZ RTTIM       ;INC REAL TIME BY TWO
18 00573 101000         MOV 0,0
19 00574 060177         INTEN
20 00575 002000         JMP @0
```

```
10022 MNSCM
01                      ;LINTR - LINKER INTERRUPT HANDLER
02 00576 075401 LINTR:  PSH. 3          ;SAVE AC3
03 00577 071401         PSH. 2          ;AC2
04 00600 065401         PSH. 1          ;AC1
05 00601 061401         PSH. 0          ;AC0
06 00602 020000         LDA 0,0
07 00603 101102         MOVL 0,0,SZC    ;CARRY
08 00604 063077         HALT            ;BIT 0 OF 0=1?
09 00605 061401         PSH. 0          ;PC*2+CRY IN BIT 15
10 00606 063610         SKPDN TTI       ;TTI INTR?
11 00607 000403         JMP .+3
12 00610 000746         JSR @L"TI
13 00611 000411         JMP LINTD
14 00612 061477         INTA 0
15 00613 100415         COM# 0,0,SNR    ;=1
16 00614 000417         JMP PFAIL
17 00615 030402         LDA 2,.+2
18 00616 143001         ADD 2,0,SKP
19 00617 060200         NIOC 0
20 00620 034401         STA 0,.+1
21 00621 060200         NIOC 0          ;CHANGED TO CURRENT DEV#
```

```
10023 MNSCM
01
02                ;LINTD - LINKER INTERRUPT DISMISS
03
04                LINTD:
05 00022 061601           POP. 0  ;GET CRY+2*PC
06 00023 101220           MOVZR 0,0       ;RESTORE CRY
07 00024 040000           STA 0,0
08 00025 061601           POP. 0
09 00026 065601           POP. 1
10 00027 071601           POP. 2
11 00030 075601           POP. 3
12 00031 060177           INTEN
13 00032 002000           JMP @0  ;GETS TO LOGICAL 0
14
15
16
17
18 00033 063077 PFAIL:    HALT
19 00034 000071           JSR @LMESS
20 00035 000037           PFTXT
21 00036 000206           JMP STRT2
22 00037 005215 PFTXT:    .TXTE |<15><12>POWER FAIL RESTART<15><12>|
```

```
10024 MNSCM
01                ;LSIZE- SIZE MEMORY
02                ;SET UP  PMSMX
03                ;
04 00053 054413 LSIZE:    STA     3,LS.S3
05 00054 004414           JSR     MSZ32   ;SIZE 0 TO 32K
06 00055 024110           LDA     1,K76K  ;5 BITS PHYS PAGE 37
07 00056 107400           AND     0,1
08 00057 125300           MOVS    1,1     ;AC1=LAST PHYS PAGE(32K)
09 00060 125220           MOVZR   1,1
10 00061 125220           MOVZR   1,1
11 00062 044134           STA     1,PMSMI
12 00063 044111           STA     1,DTS1K ;SMD BE LAST BLOCK IN 32K
13 00064 044132           STA     1,PMSMX ;IN CASE NOT 32K OR NO MAP
14 00065 004402           JMP     LSIZR
15 00066 000000 LS.S3: 0
16 00067 002777 LSIZR:    JMP @LS.S3      ;RETURN
17
```

```
10025 MNSCM
01
02                ;MSZ32=MEMORY SIZER 32K
03                ;SIZES IN 1K INCR EXIT AC0=HIGHEST AVAIL ADRS.
04                ;
05 00670 054424 MSZ32:  STA     3,XMS32 ;SAVE
06 00671 126400         SUB     1,1     ;0FOR FIRST 1K
07 00672 030420         LDA     2,K1K   ;1777 FOR END OF 1K
08 00673 133000         ADD     1,2     ;=CURRENT 1K FIELD
09 00674 025000         LDA     1,0,2   ;GET CELL
10 00675 120000         COM     1,0     ;CHNG BITS
11 00676 041000         STA     0,0,2
12 00677 021000         LDA     0,0,2   ;=COM MEM EXISTS
13 00700 122405         SUB     1,0,SNR ;AND WE'LL SKIP
14 00701 000407         JMP     M32SZ   ;LST WAS NONEXIST
15 00702 050411         STA     2,M32TEM
16 00703 045000         STA     1,0,2   ;RESTORE CELL
17 00704 024407         LDA     1,M32TE ;AC1=LAST 1K TOP ADDRESS
18 00705 125400         INC     1,1
19 00706 125133         MOVZL#  1,1,SNC ;SKP IF LAST CELL =32K
20 00707 000763         JMP     MSZ32+2 ;NOT DONE SIZING
21 00710 024403 M32SZ:  LDA     0,M32TE ;AC0=HIGHEST AVAIL.
22 00711 002403         JMP     @XMS32
23 00712 001777 K1K: 1777
24 00713 000000 M32TE: 0
25 00714 000000 XMS32: 0
```

```
10026 MNSCM
01
02                ;LRANP=SELECT A PROGRAM TO RUN
03                ;SCAN  SWITCHES FOR NOT DELETED
04                ;ENTER ANY TEST NOT DELETED
05                ; IF NOT SELECT ANOTHER TEST
06 00715 056447 LRANP:  STA 3,@LR,I3
07 00716 020113         LDA 0,CURPR
08 00717 030104         LDA 2,TTBL      ;START OF TEST LINKS
09 00720 113000         ADD 0,2         ;POINT TO NEXT TEST TO TRY
10 00721 035000 LPRL1:  LDA 3,0,2       ;GET TEST LINK
11 00722 175005         MOV 3,3,SNR     ;SKP IS TEST EXISTS
12 00723 000410         JMP LPRL2       ;START THRU LIST AGAIN
13 00724 025401         LDA 1,1,3       ;GET INTR SW
14 00725 124015         COM# 1,1,SNR    ;WAITING INT OR DISABLED?
15 00726 000402         JMP LPS1E       ;YES TRY NEXT TEST
16 00727 002433         JMP @ILRG       ;SETUP FOR THIS TEST
17 00730 101400 LPS1E:  INC 0,0
18 00731 151400         INC 2,2
19 00732 000767         JMP LPRL1       ;TRY NEXT TEST
20 00733 014204 LPRL2:  DSZ PASSIN
21 00734 000424         JMP NXTPAS
22 00735 020205         LDA 0,PASSVL
23 00736 040204         STA 0,PASSIN
24 00737 020130         LDA 0,ERTOT
25 00740 101004         MOV 0,0,SZR     ;PRINT PASS XX IF NO ERRS
26 00741 000411         JMP LPRL3       ;ERROR DON'T PRINT
27 00742 020065         LDA 0,SWREG
28 00743 024420         LDA 1,LR,B4     ;=1B4
29 00744 123404         AND 1,0,SZR     ;SKP = PRINT PASS XX
30 00745 000405         JMP LPRL3       ;DON'T PRINT PASS XX
31 00746 006071         JSR @LMESS
32 00747 000765         PASST
33 00750 020203         LDA 1,PASS
34 00751 006070         JSR @PDECI
35 00752 010203 LPRL3:  ISZ PASS
36 00753 101000         MOV 0,0
37 00754 034045         LDA 3,45
38 00755 021404         LDA 0,4,3
39 00756 101004         MOV 0,0,SZR
40 00757 000412         JMP DTSRN       ;DTUS RUN
41 00760 102400 NXTPAS: SUB 0,0
42 00761 000736         JMP LRANP+2     ;START THRU LIST AGAIN
43 00762 001263 ILRG:   LPRGO
44 00763 000000 LR.B4:  1B4
45 00764 003000 LR.I3:  LR.S3
46
47 00765 005215 PASST: .TXTE |<15><12>PASS |
48       040520
49       051523
50       000240
```

```
↓0027 MNSCM
01                      ;DTOS RUN- CHECK FOR CAT/KITTEN AND IF AUTO-RUN
02 00771 020126 DTSRN:  LDA 0,KATSW     ;CAT/KITTEN ALREADY STARTED?
03 00772 101004         MOV 0,0,SZR     ;SKIP IS NOT STARTED
04 00773 000416         JMP AUTOK
05 00774 020130         LDA 0,ERTOT
06 00775 101004         MOV 0,0,SZR     ;DON'T START CAT IF ANY ERRORS
07 00776 000413         JMP AUTOK
08 00777 021402         LDA 0,2,3       ;CHECK IF WITH CAT/KITTEN
09 01000 101005         MOV 0,0,SNR     ;SKP IS CAT/KITTEN LOADED
10 01001 000410         JMP AUTOK
11 01002 040126         STA 0,KATSW     ;SET SWITCH
12 01003 025404         LDA 1,4,3       ;GET DTOS STARTING ADDR.
13 01004 030423         LDA 2,K1375     ;FORM CAT/KITTEN STARTING ADDR.
14 01005 146400         SUB 2,1
15 01006 135000         MOV 1,3
16 01007 060277         INTDS
17 01010 000400         JSR 0,3         ;START CAT/KITTEN
18 01011 034045 AUTOK:  LDA 3,45
19 01012 021400         LDA 0,0,3       ;AUTO RUN?
20 01013 101005         MOV 0,0,SNR     ;SKIP IS AUTO-RUN
21 01014 000744         JMP NXTPAS
22 01015 020130         LDA 0,ERTOT
23 01016 101004         MOV 0,0,SZR
24 01017 000403         JMP DTOSR       ;RETURN TO DTOS IF ERRORS
25 01020 015403         DSZ 3,3
26 01021 000737         JMP NXTPAS
27 01022 061077 DTOSR:  I,RST
28 01023 021403         LDA 0,3,3
29 01024 035404         LDA 3,4,3       ;GET DTOS RETURN ADDR
30 01025 041776         STA 0,-2,3
31 01026 001400         JMP 0,3         ;RETURN TO DTOS
32 01027 001375 K1375:  1375
33 01030 000576 DTS,I:  LINTR
34
```

```
↓0028 MNSCM
01                      ;ILLEGAL SUPERVISOR CALL TYPEOUT
02 01031 000071 ICALL:  JSR @LMESS
03 01032 001044         ICALT           ;TEXT
04 01033 024467         LDA 1,CD.LA
05 01034 000073         JSR @LPOCT
06 01035 020461         LDA 0,CD.S0
07 01036 024461         LDA 1,CD.S1
08 01037 030461         LDA 2,CD.S2
09 01040 000100         JSR @ERROI      ;TYPE PRM AC'S ETC
10 01041 000401         JMP .+1
11 01042 036417         LDA 3,@IOV.R
12 01043 001400         JMP 0,3         ;RETURN
13 01044 005215 ICALT:  .TXTE (<15><12>
14 01045 144411         ILLEGAL SUPER CALL AT (
15                      )
16 01061 012105 IOV.R:          EXC.3
```

```
10029 MNSCM
01                ;CALL DISPATCH ROUTINE
02 01062 040434 CDISP:  STA     0,CD.S0
03 01063 044434         STA     1,CD.S1
04 01064 050434         STA     2,CD.S2
05 01065 054434         STA     3,CD.S3
06 01066 006435         JSR     @CD.KY          ;CK FOR SWPAK REQUESTS
07 01067 036066         LDA     3,@TPLOC        ;GET PC
08 01070 021400 CDGLC:  LDA     0,0,3           ;GET CALL
09 01071 054431         STA     3,CD.LA         ;SV LGICL ADRS
10                ;NOW DETERMINE IF IT IS A VALID CALL
11 01072 030432         LDA     2,CALLS         ;START OF CALLS
12 01073 142433         SUBZ#   2,0,SNC         ;MUST BE=>
13 01074 000735         JMP     ICALL                   ;ILLEGAL CALL?
14 01075 024430         LDA     1,CALLE         ;LAST VALID CALL
15 01076 122032         ADCZ#   1,0,SZC         ;MUST BE =<
16 01077 000732         JMP     ICALL           ;INVALID CALL?
17 01100 142640         SUBOR   2,0     ;CREATE CALL ADRS
18 01101 115220         MOVZR   0,3     ;MOV 4 R
19 01102 175220         MOVZR   3,3
20 01103 175220         MOVZR   3,3     ;(AC3)=CALL#
21 01104 020412         LDA     0,CD.S0
22 01105 024412         LDA     1,CD.S1
23 01106 030412         LDA     2,CD.S2
```

```
10030 MNSCM
01 01107 007470 CD.EX:  JSR     @PDECI,3        ;CALL JSR
02 01110 000402         JMP     .+2
03 01111 010411         ISZ     CD.LA   ;TO GET PAST MESSAGE ADDR.
04 01112 010410         ISZ     CD.LA   ;NORMAL +1 RETURN
05 01113 034406         LDA     3,CD.S3
06 01114 060177         INTEN
07 01115 002405         JMP     @CD.LA
08                ;ABOVE JMP RETURNS TO USER
09 01116 000000 CD.S0:  0
10 01117 000000 CD.S1:  0
11 01120 000000 CD.S2:  0
12 01121 000000 CD.S3:  0
13 01122 000000 CD.LA:  0
14 01123 007502 CD.KY:  CKKEY
15              CALLS:  LCALL PDECI
16              CALLE:  LCALL LWRBK
```

```
 0031 MNSCM
01                ;EXCTS-EXECUTE TEST SELECTED
02                ;CONTROL MEMORY TEST LIMITS
03                ;
04 01126 054457 EXCTS:  STA 3,EXC.3
05 01127 020457         LDA 0,K33       ;MAX USEABLE SEGMENT
06 01130 014117         DSZ SEGMT       ;SEGMT SIZE-1
07 01131 000401         JMP .+1
08 01132 024117         LDA 1,SEGMT     ;MAX SEGMENT FOR THIS TEST
09 01133 122433         SUBZ# 1,0,SNC   ;SKP IF SEGMT <= 33
10 01134 105000         MOV 0,1         ;USE 33
11 01135 044453         STA 1,EX.SG     ;SAVE IT
12 01136 020133         LDA 0,PHSLO
13 01137 040135 EXC.L:  STA 0,TSTLO     ;SETUP LO LIMIT OF TEST
14 01140 107000         ADD 0,1         ;ADD SEGMT TO LO LIMIT
15 01141 030134         LDA 2,PHSHI     ;END OF TESTABLE AREA
16 01142 132433         SUBZ# 1,2,SNC   ;SKP IS WITHIN HI LIMIT
17 01143 000403         JMP .+3
18 01144 044136         STA 1,TSTHI     ;SETUP HI LIMIT OF TEST
19 01145 000402         JMP .+2
20 01146 050136         STA 2,TSTHI     ;USE PHSHI IF TSTLO+SEGMT>PHSHI
21                ;CHECK THAT DTOS-1K IS NOT USED IN TESTING
22 01147 030060         LDA 2,EGGS      ;CHECK IF AUTO LOAD
23 01150 151004         MOV 2,2,SZR     ;SKIP IS NOT AUTO
24 01151 000404         JMP PRDTS
25 01152 030062         LDA 2,EGGS+2    ;CAT/KITTEN LOADED?
26 01153 151005         MOV 2,2,SNR     ;SKP IS YES
27 01154 000420         JMP EXC.J       ;NO NEED TO PROTECT DTOS
28 01155 030111 PRDTS:  LDA 2,DTS1K     ;GET DTOS-1K WORD
29 01156 112433         SUBZ# 0,2,SNC   ;TSTLO <= DTS1K?
30 01157 000415         JMP EXC.J       ;LIMITS OK, CONTINUE
31 01160 024136         LDA 1,TSTHI
32 01161 132032         ADCZ# 1,2,SZC   ;TSTHI => DTOS-1K
33 01162 000412         JMP EXC.J       ;LIMITS OK
34 01163 112414         SUB# 0,2,SZR    ;TSTLO = DTOS-1K?
35 01164 000405         JMP EXC.C       ;
36 01165 132415         SUB# 1,2,SNR    ;TSTHI = TSTLO = DTOS-1K ?
37 01166 000410         JMP EXC.R       ;TSTLO=TSTHI=DTS1K,DON'T USE IT
38 01167 010135         ISZ TSTLO       ;INC TSTLO TO DTOS.1K+1
39 01170 000404         JMP EXC.J       ;LIMITS NOW OK
40 01171 120000 EXC.C:  ADC 1,1
41 01172 133000         ADD 1,2         ;DTS1K-1
42 01173 050136         STA 2,TSTHI     ;FIX TSTHI TO DTS1K-1
43                ;SET UP SCRLO/HI LIMITS AND MAP IF IT IS TO BE USED
44 01174 004415 EXC.J:  JSR LDMAP       ;LOAD MAP AND ADJUST SCRLO/HI
45
46                ;LIMITS ALL SET UP NOW EXECUTE THIS PASS OF TEST
47 01175 006412         JSR @ILSTR      ;START TESTING
48                ;NOW SEE IF ALL TESTING AREA USED
49 01176 020136 EXC.R:  LDA 0,TSTHI
50 01177 024134         LDA 1,PHSHI
51 01200 106033         ADCZ# 0,1,SNC   ;DONE ALL?
52 01201 002404         JMP @EXC.3      ;YEP RETURN
53 01202 101400         INC 0,0 ;NOT YET
54 01203 024405         LDA 1,EX.SG     ;GET SEGMT SIZE
55 01204 000733         JMP EXC.L       ;CONTINUE TESTING
56 01205 000000 EXC.3:  0
57 01206 000033 K33:    33
58 01207 001352 ILSTR:  LSTRP
59 01210 000000 EX.SG:  0
60
```

```
 0032 MNSCM
01                ;LDMAP-LOAD MAP ENTRY TABLES AND ADJUST SCRLO/HI
02                ;FOR THIS TEST PASS
03                ;
04 01211 054416 LDMAP:  STA 3,LD.S3
05 01212 000401         JMP LD.NM       ;DON'T USE MAP
```

```
10033 MNSCM
01                ;MAP DOESN'T EXIST SO SIMPLY SET SCRLO/HI LIMITS
02                ;TO TSTLO/HI
03                ;
04 01213 020135 LD.NM:  LDA 0,TSTLO
05 01214 103300         MOVS 0,0
06 01215 103120         ADDZL 0,0
07 01216 040137         STA 0,SCRLO       ;TSTLO * 1024 = SCRLO
08 01217 020136         LDA 0,TSTHI
09 01220 101400         INC 0,0
10 01221 103300         MOVS 0,0
11 01222 103120         ADDZL 0,0
12 01223 126000         ADC 1,1
13 01224 123000         ADD 1,0           ;-1 AC0
14 01225 040140         STA 0,SCRHI       ;(TSTHI*1024)-1=SCRHI
15 01226 002401         JMP @LD.S3        ;RETURN LIMITS SET
16
17 01227 000000 LD.S3:  0
```

```
10034 MNSCM
01                ;TINIT=TEST INITIALIZE
02                ;SEQUENCE THROUGH THE INITIALIZE ADDRESSES
03                ;FOR EACH TEST LOADED ALONG WITH LINKER
04 01230 054430 TINIT:  STA     3,XTINI
05 01231 102400         SUB     0,0       ;START WITH PGM # 0
06 01232 040427         STA     0,NPROG   ;PROG TO INIT
07 01233 026105         LDA     1,@LPGMK
08 01234 034425         LDA     3,NPROG   ;NEXT PROG TO INIT
09 01235 166415         SUB#    3,1,SNR   ;SKP IS NOT DONE ALL
10 01236 002422         JMP     @XTINI    ;EXIT ALL PROGS INITED
11 01237 026104         LDA     0,TTBL
12 01240 117000         ADD     0,3
13 01241 031400         LDA     2,0,3     ;GET INIT ADRS
14 01242 102400         SUB     0,0
15 01243 041376         STA     0,-2,2    ;PASS CNTR
16 01244 041377         STA     0,-1,2    ;ERR CNTR
17 01245 034414         LDA     3,NPROG   ;CHECK
18 01246 024106         LDA     1,K4
19 01247 166032         ADCZ#   3,1,SZC   ;IF PROGRM IS => AC1
20 01250 000404         JMP     TINIP     ;NOT YET
21 01251 026411         LDA     1,@TINTL          ;CHECK AUTO FLG
22 01252 124014         COM#    1,1,SZR   ;SKP IS NOT AUTO START
23 01253 102000         ADC     0,0       ;-1 IF AUTOSTART
24 01254 041001 TINIP:  STA     0,1,2     ;WAIT SW
25 01255 011404         ISZ     NPROG     ;STEP TO NXT PROG
26 01256 101000         MOV     0,0
27 01257 000754         JMP     TINIT+3   ;AND DO AGN
28 01260 000000 XTINI:  0
29 01261 000000 NPROG:  0
30 01262 000230 TINTL:  LAUTO             ;POINT TO AUTO FLG
```

```
10035 MNSLM
01                ;SET PROGRAM START/END ADDRESSES
02 01263 040113 LPRGO:  STA 0,CURPR      ;CURRENT PROG #
03 01264 030104         LDA 2,TTBL       ;LAST LINKR ZLOC+1
04 01265 113000         ADD 0,2          ;A2=PNTR TO PARAM ADRS
05 01266 035000         LDA 3,0,2
06 01267 054114         STA 3,PSTRT      ;START ADDRES OF PRG
07 01270 031001         LDA 2,1,2        ;GET START OF NXT PGM
08 01271 151005         MOV 2,2,SNR      ;=0 IS LAST TEST
09 01272 030407         LDA 2,LW.K9      ;AND WE USE NMAX
10 01273 050112         STA 2,PENDA
11 01274 014112         DSZ PENDA        ;REAL END OF THIS PGM
12 01275 031402         LDA 2,2,3        ;GET SEGMENT SIZE FOR TEST
13 01276 050117         STA 2,SEGMT      ;SAVE IT
14 01277 002401         JMP @LR.S3
15 01300 000000 LR.S3: 0
16 01301 000560 LW.K9:  LSYEE
17                ;SETML-SET MEMORY LIMITS IF SW0=1
18 01302 054433 SETML:  STA 3,SETS3      ;SAVE RETURN ADDZRESS
19 01303 020120         LDA 0,SWRG2      ;GET MEM LIMIT FLG
20 01304 126620         SUBZR 1,1
21 01305 123415         AND# 1,0,SNR     ;IS BIT 0 SET?
22 01306 000425         JMP SETMX+1      ;NOPE, EXIT
23 01307 101120         MOVZL 0,0
24 01310 101220         MOVZR 0,0        ;CLR BIT 0
25 01311 040120         STA 0,SWRG2      ;CLR MEM LMT FLG
26 01312 060277         INTDS
27 01313 006071 GTPHS:  JSR @LMESS
28 01314 002557         TXT.0            ;TEXT FOR PHSLO,PHSHI
29 01315 004421         JSR TTIN         ;INPUT FROM OPERATOR
30 01316 030131         LDA 2,PHSLX      ;GET LOW PHYSICAL ADDR.
31 01317 112032         ADCZ# 0,2,SZC    ;PHSLO=>PHSLX?
32 01320 000773         JMP GTPHS        ;NO,TRY AGAIN
33 01321 040133         STA 0,PHSLO      ;PLACE INPUT IN PHSLO
34 01322 004414         JSR TTIN         ;INPUT FROM OPERATOR
35 01323 030132         LDA 2,PHSHX      ;GET HIGH PHYSICAL ADDR.
36 01324 112433         SUBZ# 0,2,SNC    ;PHSLO<=PHSHX?
37 01325 000766         JMP GTPHS        ;NO,TRY AGAIN
38 01326 040134         STA 0,PHSHI      ;NEW PHSHI LIMITS
39 01327 024133         LDA 1,PHSLO      ;GET PHSLO LIMIT
40 01330 122423         SUBZ 1,0,SNC     ;IS PHSHI>PHSLO ?
41 01331 000762         JMP GTPHS        ;NOPE, TRY AGAIN
42 01332 006074 SETMX:  JSR @PCRLF       ;CRLF
43 01333 060177         INTEN
44 01334 002401         JMP @SETS3       ;RETURN
45 01335 000000 SETS3: 0
46                ;INPUT HANDLER FOR MEMORY LIMITS
47 01336 054411 TTIN:   STA 3,TTIS3
48 01337 006411         JSR @TT.DE
49 01340 000403         JMP PNTQ         ;ERROR IN INPUT
50 01341 121000         MOV 1,0
51 01342 002405         JMP @TTIS3       ;RETURN
52
53 01343 020403 PNTQ:   LDA 0,QST
54 01344 000405         JSR @LCHAR
55 01345 000772         JMP TTIN+1       ;TRY AGAIN
56 01346 000077 QST:    077              ; "?"
57 01347 000000 TTIS3:  0
58 01350 001735 TT.DE:  TID?
59 01351 001671 LCHAR:  CHC?T
```

```
10036 MNSCM
01                ;LSTRP-START PROGRAM
02                ;ENTER TEST SELECTED AT ITS EXECUTION ENTRY POINT
03 01352 054413 LSTRP:  STA 3,L.SS3
04 01353 102400         SUB 0,0
05 01354 040121         STA 0,ITRER
06 01355 020114         LDA 0,PSTRT
07 01356 040410         STA 0,LS.I1
08 01357 111000         MOV 0,2
09 01360 126400         SUB 1,1
10 01361 045375         STA 1,-3,2       ;CLR ERR-CTR FOR THIS PASS
11 01362 002404         JMP @LS.I1
12                ;LRETP-RETURN FROM TEST PROG CALL
13 01363 060177 LRETP:  INTEN            ;ALLOW INTR NOW
14 01364 002401         JMP @L.SS3
15 01365 000000 L.SS3: 0
16 01366 000000 LS.I1: 0
```

```
1#037 MNSCM
01                ;END OF TEST PASS, SEE IF ANY EXTRANEOUS ERRORS
02                ;NEED TO BE REPORTED
03 01367 054466 LSTAT:  STA 3,LST.3
04 01370 030114 PRSTAT: LDA 2,PSTRT       ;PTR, TO XX,00
05 01371 011376         ISZ -2,2          ;+1 RUN CTR THIS TEST
06 01372 101000         MOV 0,0
07 01373 02#124         LDA 0,SWRG2       ;CHECK SWREG2
08 01374 103123         ADDZL 0,0,SNC     ;BIT 1 = 1?
09 01375 000414         JMP PRS01         ;NOPE
10 01376 025375         LDA 1,-3,2        ;GET ERROR CNTR FOR THIS TEST AND PASS
11 01377 125004         MOV 1,1,SZR       ;SKP = PRINT PASSED,NO ERRORS
12 01400 000411         JMP PRS01         ;DON'T PRINT PASSED
13 01401 006074         JSR @PCRLF
14 01402 02#114         LDA 0,PSTRT
15 01403 101400         INC 0,0
16 01404 101400         INC 0,0
17 01405 101400         INC 0,0
18 01406 006075         JSR @ERRTX        ;PRINT TEST TITLE
19 01407 006071         JSR @LMESS
20 01410 001475         PASTX             ;ADDRESS OF PASS MSG
21 01411 02#120 PRS01:  LDA 0,SWRG2       ;=BIT 15  IS TYPE ALL STATS
22 01412 126520         SUBZL 1,1
23 01413 107405         AND 0,1,SNR       ;BIT 15 ON?
24 01414 000435         JMP LSXIT         ;NOBODY'S INTERESTED
25 01415 101220         MOVZR 0,0
26 01416 101120         MOVZL 0,0
27 01417 040120         STA 0,SWRG2       ;DON'T PRINT TWICE
28 01420 102400         SUB 0,0
29 01421 040433         STA 0,PR.SV       ;START WITH 0
```

```
1#038 MNSCM
01                ;PRINT PASSES AND ERROR COUNTS BY INDIVIDUAL TEST
02 01422 006071         JSR @LMESS
03 01423 001456         STMDR
04 01424 006074         JSR @PCRLF
05 01425 024427 PRSTL:  LDA 1,PR.SV
06 01426 010426         ISZ PR.SV         ;STEP IT TO NEXT
07 01427 034104         LDA 3,TTBL        ;POINTS TO TEST 0
08 01430 137000         ADD 1,3           ;NOW TO  CURRENT TEST
09 01431 031400         LDA 2,0,3         ;GET "XX,00" ADRS
10 01432 151005         MOV 2,2,SNR       ;=0 IS TYPED ALL
11 01433 000416         JMP LSXIT                 ;RETURN FROM WHENCE
12 01434 021376         LDA 0,-2,2        ;GET PASS CTR
13 01435 101005         MOV 0,0,SNR       ;SKP IF TEST EXECUTED
14 01436 000767         JMP PRSTL         ;TEST NEVER ENTERED
15 01437 006072         JSR @PZOCT        ;TYPE TEST#
16 01440 025376         LDA 1,-2,2        ;GET # PASSES
17 01441 006070         JSR @PDECI        ;PRINT IT
18 01442 025377         LDA 1,-1,2        ;GET ACCUMULATED ERRS
19 01443 006070         JSR @PDECI        ;PRINT # ERRS THIS TEST
20 01444 025375         LDA 1,-3,2        ;GET ERRS FOR THIS PASS
21 01445 125004         MOV 1,1,SZR       ;=0 DON'T PRINT
22 01446 006070         JSR @PDECI        ;PRINT ERRS FOR THIS PASS
23 01447 006074         JSR @PCRLF
24 01450 000755         JMP PRSTL
25 01451 010113 LSXIT:  ISZ CURPR         ;INC CUR PGM POINTR
26 01452 101000         MOV 0,0           ;DELETE SKIP
27 01453 002402         JMP @LST.3                ;RETURN
28 01454 000000 PR.SV:  0
29 01455 000000 LST.3:  0                 ;SAVE AC3
30 01456 005215 STMDR:  .TXTE |<15><12>TST#<11>PASSES<11>ACERRS<11>PSERRS |
31 01475 050240 PASTX:  .TXTE | PASSED<15><12>|
```

```
0039 MNSCM
01                      ;CHECK FOR TTI INPUTTED CHARACTERS
02 01502 054440 CKKEY:  STA 3,CK.S3
03 01503 020125         LDA 0,LASTI
04 01504 101123         MOVZL 0,0,SNC   ;TTI REQUEST?
05 01505 002435         JMP @CK.S3      ;NOPE!
06 01506 101220         MOVZR 0,0
07 01507 040125         STA 0,LASTI     ;CLR FLAG
08 01510 006433         JSR @KEYIN      ;YEP GO TO KEY IN SERVICE
09 01511 151000         MOV 3,0
10 01512 032120         LDA 2,SWRG2     ;CURRENT CONTENTS OF SWRG2
11 01513 024431         LDA 1,CK.K1     ;=121
12 01514 122425         SUBZ 1,0,SNR    ;CHECK FOR A "Q"
13 01515 000417         JMP WAS.Q       ;IT WAS A "Q"
14 01516 101405         INC 0,0,SNR     ;CHECK IT FOR A "P"
15 01517 000411         JMP WAS.P       ;FOUND A "P"
16 01520 024106         LDA 1,K4
17 01521 122425         SUBZ 1,0,SNR    ;CHECK IF IT WAS A "T"
18 01522 000410         JMP WAS.T       ;IT WAS
19 01523 024424         LDA 1,CK.23     ;=23, CONTROL S
20 01524 136414         SUB# 1,3,SZR    ;SKP IS CONTROL S
21 01525 000414         JMP CK.3A
22 01526 126520         SUBZL 1,1       ;IT WAS
23 01527 000406         JMP XOR.T
24 01530 024415 WAS.P:  LDA 1,CK.K2     ;=181
25 01531 000404         JMP XOR.T
26 01532 126620 WAS.T:  SUBZR 1,1
27 01533 000402         JMP XOR.T
28 01534 024412 WAS.Q:  LDA 1,CK.K3     ;=182
29 01535 147414 XOR.T:  AND# 2,1,SZR
30 01536 132401         SUB 1,2,SKP     ;XOR AC1 WITH AC2
31 01537 133000         ADD 1,2
32 01540 050120         STA 2,SWRG2     ;PUT NEW BIT IN SWRG2
33 01541 002401 CK.3A:  JMP @CK.S3
34 01542 000000 CK.S3:  0
35 01543 002106 KEYIN:  INP?            ;ADDRESS OF TTYINTR PACKAGE
36 01544 000121 CK.K1:  121
37 01545 040000 CK.K2:  181
38 01546 020000 CK.K3:  182
39 01547 000023 CK.23:  23
40
41                      ;TTY INTERRUPT HANDLER
42 01550 060610 TT.TI:  DIAC 0,TTI
43 01551 024416         LDA 1,TT177
44 01552 123400         AND 1,0
45 01553 024125         LDA 1,LASTI     ;CHECK IF LAST INTA SERVICED
46 01554 125123         MOVZL 1,1,SNC   ;SKP = NOT SERVICED
47 01555 000407         JMP TT.DI       ;WAS SERVICED,DISMISS INTR
48 01556 024412         LDA 1,TT.04     ;CHECK FOR (C)D
49 01557 106415         SUB# 0,1,SNR    ;SKP IS NOT (C)D
50 01560 000412         JMP TT.P?       ;IT WAS GO TO TTI SERVICE
51 01561 024410         LDA 1,TT.22     ;CHECK FOR (C)R
52 01562 106415         SUB# 0,1,SNR    ;SKP = NOT (C)R
53 01563 000407         JMP TT.P?
54 01564 103240 TT.DI:  ADDOR 0,0
55 01565 040125         STA 0,LASTI     ;PLACE IN LASTI WITH BIT 0 SET=1
56 01566 001400         JMP 0,3
57 01567 000177 TT177:  177
58 01570 000004 TT.04:  4
59 01571 000022 TT.22:  22
60
```

```
0040 MNSCM
01 01572 040125 TT.P?:  STA 0,LASTI
02 01573 054403         STA 3,TT.S3
03 01574 006747         JSR @KEYIN      ;SERVICE TTI REQUEST
04 01575 002401         JMP @TT.S3
05 01576 000000 TT.S3:  0
```

```
0041 MNSCM
01                      ;FILENAME=TTYIO
02
03                      ;TELETYPE NON 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                      ;"PDE?" 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 01577 054524 MES?:   STA     3,MES?R         ;PRINT A TEXT MESSAGE
31 01600 044465         STA     1,PAC?1
32 01601 050466         STA     2,PAC?2
33 01602 010521         ISZ     MES?R
34 01603 031400         LDA     2,0,3           ;C(2) POINTS TO MESSAGE
35 01604 024464         LDA     1,PJ777         ;A 8 BIT MASK
36 01605 021000         LDA     0,0,2           ;C(2)=DATA WORD
37 01606 125112         MOVL#   1,1,SZC
38 01607 123701         ANDS    1,0,SKP
39 01610 123401         AND     1,0,SKP         ;C(0)=DATA CHARACTOR RIGHT
40 01611 151400         INC     2,2             ;INC TO NEXT WORD
41 01612 124000         COM     1,1             ;FLIP MASK
42 01613 024456         JSR     CHC?T           ;PRINT
43 01614 000771         JMP     MES?+0          ;ANOTHER
44 01615 000402         JMP     .+2
45 01616 004453 PLS?T:  JSR     CHC?T
46 01617 024447 PEX?T:  LDA     1,PAC?1
47 01620 030447         LDA     2,PAC?2
48 01621 063511         SKPBZ   TTO
49 01622 000777         JMP     .-1
50 01623 101000         MOV     0,0
51 01624 060211         NIOC    TTO
52 01625 002476         JMP     @MES?R          ;LAST
53 01626 102401 ZOC?:   SUB     0,0,SKP
54 01627 020563 POC?:   LDA     0,PC622
55 01630 050437         STA     2,PAC?2
56 01631 030565         LDA     2,OCT?AB                ;PRINT C(1) IN OCTAL
57 01632 000404         JMP     .+4
58 01633 050434 PDE?:   STA     2,PAC?2
59 01634 030572         LDA     2,DEC?TB                ;PRINT C(1) IN DECIMAL
60 01635 102400         SUB     0,0
```

```
0042 MNSCM
01 01636 054465         STA     3,MES?R         ;BOTH ENTRYS PRINT NUMBER
02 01637 044427         STA     1,PAC?1
03 01640 040425         STA     0,ZSU?P         ;THEN TAB TO NEXT POSITION
04 01641 050401         STA     2,.+1
05 01642 000000 DEC?OC: 0                       ;A"LDA 2,TABLE" INSTRUCTION
06 01643 410777         ISZ     .-1
07 01644 020571         LDA     0,PTB?
08 01645 151005         MOV     2,2,SNR         ;IF TABLE ENTRY=0
09 01646 000750         JMP     PLS?T           ;EXIT WITH A SPACE/TAB
10 01647 034416         LDA     3,ZSU?P         ;ZEROS SUPPRESS STUF
11 01650 102400         SUB     0,0
12 01651 146452 DEC?T:  SUBO#   2,1,SZC
13 01652 000405         JMP     DEC?P
14 01653 146400         SUB     2,1             ;FORM THE DIGIT
15 01654 034536         LDA     3,PC622
16 01655 101400         INC     0,0
17 01656 000773         JMP     DEC?T
18 01657 151235 DEC?P:  MOVZR#  2,2,SNR
19 01660 034532         LDA     3,PC620
20 01661 054404         STA     3,ZSU?P         ;C(0)=DIGIT
21 01662 163004         AUD     3,0,SZR         ;MAKE ASCII
22 01663 004406         JSR     CHC?T           ;PRINT
23 01664 000756         JMP     DEC?OC          ;GET NEXT DIGIT
24
25 01665 000000 ZSU?P:  0
26 01666 000000 PAC?1:  0
27 01667 000000 PAC?2:  0
28 01670 000377 PJ777:  377
29 01671 054431 CHC?T:  STA     3,CHR?T         ;PRINT C(0) RIGHT
30 01672 101315         MOVS#   0,0,SNR         ;RETURN +2 IF NULL
31 01673 001401         JMP     1,3
32 01674 034425         LDA     3,PC171         ;AC3 = 11
33 01675 116415         SUB#    0,3,SNR         ;SKIP IF A TAB IS NOT TO
34                                              ;BE SIMULATED
35 01676 000403         JMP     CHA?3
36 01677 004540         JSR     TYP?            ;PRINT IT
37 01700 002422         JMP     @CHR?T          ;EXIT
38 01701 004535 CHA?3:  JSR     TPS?            ;PRINT A SPACE
39 01702 020416         LDA     0,CHR?Z
40 01703 034414         LDA     3,PC7?          ;AC3 = 7
41 01704 163404         AND     3,0,SZR
42 01705 000774         JMP     CHA?3           ;SIMULATE A TABE WITH 1
43                                              ;TO 7 SPACES
44 01706 040412         STA     0,CHR?Z
45 01707 002413         JMP     @CHR?T
46
47 01710 054413 CLF?:   STA     3,MES?R         ;SAVE RETURN
48 01711 044755         STA     1,PAC?1
49 01712 050755         STA     2,PAC?2
50 01713 020554         LDA     0,PC175
51 01714 004755         JSR     CHC?T           ;PRINT CARRIAGE AND LF
52 01715 020515         LDA     0,PC172
53 01716 000760         JMP     PLS?T
54
55 01717 000007 PC7?:   7
56 01720 000000 CHR?Z:  0
57 01721 000011 PC171:  11
58 01722 000000 CHR?T:  0
59 01723 000000 MES?R:  0
60 01724 020506 TIC?:   LDA     0,PC172
```

```
0043 MNSCM
01 01725 004512          JSR     TYP?
02 01726 010775 TIX?:    ISZ     MES?R
03 01727 024737 TIR?:    LDA     1,PAC?1
04 01730 034735          LDA     3,ZSU?P
05 01731 175102          MOVL    3,3,SZC
06 01732 124400          NEG     1,1
07 01733 000665          JMP     PEX?T+1
08
09 01734 102121 TIO?:    ADCZL   0,0,SKP         ;OCTAL ENTRY
10 01735 102440 TID?:    SUBO    0,0             ;DECIMAL ENTRY
11 01736 054765          STA     3,MES?R
12 01737 050730          STA     2,PAC?2         ;AC2 IS SAVED
13 01740 030472          LDA     2,PC1??
14 01741 113000          ADD     0,2
15 01742 102440          SUBO    0,0
16 01743 044722          STA     0,ZSU?P         ;MINUS SIGN AND LEADING SPACES
17                                               ;FLAG
18 01744 034721 TIS?:    LDA     3,ZSU?P
19 01745 175004          MOV     3,3,SZR
20 01746 000760          JMP     TIX?
21 01747 054717 TIW?:    STA     3,PAC?1
22 01750 063610          SKPDN   TTI
23 01751 000777          JMP     .-1
24 01752 101000          MOV     0,0
25 01753 060610          DIAC    0,TTI
26 01754 004715          JSR     CHC?T
27 01755 034436          LDA     3,PC470
28 01756 116414          SUB#    0,3,SZR
29 01757 101015          MOV#    0,0,SNR
30 01760 000764          JMP     TIS?            ;SPACE, TAB, OR NULL
31 01761 024434          LDA     1,TIN?2
32 01762 106015          ADC#    0,1,SNR         ;COMMA
33 01763 000743          JMP     TIX?            ;OR
34 01764 106424          SUBZ    0,1,SZR         ;MINUS
35 01765 000405          JMP     TIM?            ;IF NEITHER THEN GO TO TIM?
36 01766 034677          LDA     3,ZSU?P
37 01767 177200          ADDR    3,3             ;COMPLEMENT SIGN
38 01770 054675          STA     3,ZSU?P
39 01771 000757          JMP     TIW?+1
40 01772 136415 TIM?:    SUB#    1,3,SNR         ;IS IT A CARRIAGE RETURN?
41 01773 000731          JMP     TIC?            ;IF CR THEN GO TO TIC?
42 01774 024420 TIN?:    LDA     1,TIN?1
43 01775 107022          ADDZ    0,1,SZC         ;SKIP IF NOT A DIGIT
44 01776 146513          SUBL#   2,1,SNC         ;SKIP IF DIGIT
45 01777 000730          JMP     TIR?
46 02000 010665          ISZ     ZSU?P           ;OUT OF LEADING SPACES
47 02001 020665          LDA     0,PAC?1
48 02002 101120          MOVZL   0,0
49 02003 113120          MOVZL   0,3
50 02004 173120          MOVZL   3,3
51 02005 137000          ADD     1,3             ;8 OLD PAC?1'S + NEW DIGIT
52 02006 145220          MOVZR   2,1
53 02007 125232          MOVZR#  1,1,SZC         ;SKIP IF OCTAL MODE
54 02010 117000          ADD     0,3             ;ADD 2 OLD PAC?1'S
55 02011 000736          JMP     TIW?
56 02012 000060 PC6?0:   60
57 02013 000040 PC4?0:   40
58 02014 177720 TIN?1:   -60
59 02015 000055 TIN?2:   55
60 02016 030555 OCT?AB:  LDA     2,.+1+.-DEC?OC
```

```
0044 MNSCM
01 02017 100000          100000
02 02020 010000          10000
03 02021 001000          1000
04 02022 000100 C10?0:   100
05 02023 000010          10
06 02024 000001          1
07 02025 000000          0
08 02026 030565 DEC?TB:  LDA     2,.+1+.-DEC?OC
09       000012 .RDX 10
10 02027 023420          10000
11 02030 001750          1000
12 02031 000144          100
13 02032 000012 PC1??:   10
14 02033 000001          1
15 02034 000000          0
16       000010 .RDX     8
17
18 02035 000011 PIR?:    11
19
20 02036 020755 TPS?:    LDA     0,PC4?0         ;PREPARE TO PRINT A SPACE
21 02037 054431 TYP?:    STA     3,TYP?R ;TYPE THE RIGHT BYTE OF AC0
22 02040 034426          LDA     3,INT?          ;IF IT IS HERE DUE TO SWITCH
                                                 ;SETTING ROUTINE THEN THE TYPE
                                                 ;OUTS TO THE TTY WILL BE ENABLED
23
24
25 02041 175004          MOV     3,3,SZR
26 02042 034065          LDA     3,SWREG         ;READ THE SWITCHES
27 02043 177100          ADDL    3,3             ;SHIFT AC3 BY 2 PLACES
28 02044 175112          MOVL#   3,3,SZC         ;SKIP IF TYPEOUTS ARE NOT
                                                 ;SUPPRESSED
29
30 02045 000405          JMP     PLP?T
31 02046 063511 TPT?Y:   SKPBZ   TTO
32 02047 000777          JMP     .-1
33 02050 101000          MOV     0,0
34 02051 061111          DOAS    0,TTO
35 02052 034413 PLP?T:   LDA     3,P1777
36 02053 163400          AND     3,0             ;STRIP THE PARITY BIT
37 02054 034411 TPR?T:   LDA     3,P1777         ;AC3 = 177
38 02055 116043          ADCO    0,3,SNC         ;SKIP IF IT WAS A RUBOUT
39 02056 034735          LDA     3,PC470         ;AC3 = 40
40 02057 162432          SUBZ#   3,0,SZC         ;SKIP FOR NON PRINTING
                                                 ;CHARACTERS
41
42 02060 010640          ISZ     CHR?Z
43 02061 034406          LDA     3,PC175         ;AC3 = 15
44 02062 110445          SUBO    0,3,SNR         ;SKIP IF IT WAS NOT A "CR"
45 02063 054635          STA     3,CHR?Z         ;CLEAR HORZ POS
46 02064 002004          JMP     @TYP?R
47
48 02065 000177 P1777:   177
49 02066 177777 INT?:    -1
50 02067 000015 PC175:   15
51 02070 000000 TYP?R:   0
```

```
10045 MNSCM
01                ;FILENAME= SWRPACK
02
03                ;THIS PACKAGE IS USED TO CHANGE THE SETTINGS OF LOCATION
04                ;"SWREG" OF PAGE 0. THE PROGRAM CONTROL SHOULD ENTER "INP?"
05                ;WITH AC3 HAVING THE RETURN ADDRESS, THE CHARACTER INPUTED
06                ;BY THE OPERATOR IS ECHOED AFTER A "CR". IF THE COMMAND IS
07                ;NOT A LEGAL ONE THEN THE CONTROL IS RETURNED WITHOUT DOING
08                ;ANY THING, OTHERWISE ONE OF THE FOLLOWING COMMANDS IS
09                ;EXECUTED:
10                ;KEYS 1-9 AND A-F ARE USED TO COMPLEMENT THE CURRENT VALUE
11                ;OF BITS 1-15 OF "SWREG". IF ONE OF THESE KEYS IS HIT THE
12                ;CORRESPONDING BIT OF "SWREG" IS COMPLEMENTED AND THE CONTROL
13                ;IS RETURNED TO THE STATE PROGRAM HAD BEFORE HITTING THE KEY
14                ;TYPING OF A "0" WILL LOCK THE PROGRAM IN A SWITCH MODIFICATION
15                ;MODE IN WHICH CASE MORE THAN ONE BITS CAN BE CHANGED BEFORE
16                ;THE CONTROL IS ALLOWED TO RETURN TO THE MAIN PROGRAM. HITTING
17                ;THE "CR" KEY WILL UNLOCK THE PROGRAM FROM THIS MODE.
18                ;"AD" THIS COMMAND GIVEN AT ANY TIME WILL RESET THE "SWREG"
19                ;TO DEFAULT MODE (ALL ZEROS) AND RESTART THE PROGRAM AT ADD.
20                ;STORED IN LOCATION "INS?"
21                ;"AR" THIS COMMANND GIVEN AT ANY TIME  WILL RESTART THE PROG.
22                ;AT ADDRESS STORED IN LOCATION "INS?"
23                ;"M" THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE  CURRENT
24                ;OPERATING MODES.
25                ;BEFORE THE CONTROL IS RETURNED TO THE MAIN PROGRAM BIT 0 WILL
26                ;BE SET IF ANY OF THE OTHER BITS OF "SWREG" IS SET, OTHERWISE
27                ;IT WILL BE CLEARED
28
29                ;THIS PACKAGE EXITS WITH C(AC3) = CHARACTER TYPED IN (PARITY
30                ;STRIPPED).
31
32 02071 000136 IN1?36: 136
33 02072 000104 IN1?04: 104
34 02073 000122 IN1?22: 122
35 02074 000033 INC?33: 33
36 02075 000000 INL?K:  0
37 02076 000000 INR?T:  0
38 02077 000000 INS?0:  0
39 02100 000000 INS?1:  0
40 02101 000000 INS?2:  0
41 02102 000000 INS?3:  0
42 02103 000000 INS?C:  0
43 02104 000206 INS?:   STRT?
44 02105 000000 SVP?TB: 0
45
46 02106 054770 INP?:   STA     3,INR?T         ;SAVE THE RETURN ADDRESS
47 02107 040770         STA     0,INS?0         ;SAVE AC0
48 02110 044770         STA     1,INS?1         ;AC1
49 02111 050770         STA     2,INS?2         ;AND AC2
50 02112 175200         MOVR    3,3             ;
51 02113 054770         STA     3,INS?C         ;SAVE CARRY
52 02114 176400         SUB     3,3
53 02115 054751         STA     3,INT?
54 02116 006074         JSR     @ICLF?
55 02117 040756         STA     0,INL?K         ;"INL?K" IS NOT -1
56 02120 020125 IN@?:   LDA     0,LASTI         ;READ THE INPUT
57 02121 024746         LDA     1,PC175         ;AC1 = 15
58 02122 100415         SUB#    0,1,SNR         ;SKIP IF THE CHARACTER TYPED
59                                              ;WAS NOT "CR"
60 02123 000527         JMP     INR?
```

```
 0046 MNSCM
01 02124 040756         STA     0,INS?3         ;SAVE CHARACTER
02 02125 024675         LDA     1,C10?0         ;AC1 = 100
03 02126 034744         LDA     3,PC12?         ;AC3 = 12
04 02127 116414         SUB#    0,3,SZR         ;SKIP IF IT IS A LINE FEED
05 02130 034744         LDA     3,INC?33                ;AC3 = 33
06 02131 162452         SUBO#   3,0,SZC         ;SKIP IF AC0 IS EQUAL OR MORE
07                                              ;THAN AC3
08 02132 000502         JMP     IN5?
09 02133 004704 IN1?:   JSR     TYP?            ;ECHO THE CHARACTER
10 02134 034656         LDA     3,PC6?0         ;AC3 = 60
11 02135 152620         SUBZR   2,2             ;AC2 = 100000
12 02136 116405         SUB     0,3,SNR         ;SKIP IF THE DIGIT TYPED WAS
13                                              ;NOT 0
14 02137 000447         JMP     IN3?
15
16 02140 151221 IN2?:   MOVZR   2,2,SKP         ;SHIFT AC2 TO RIGHT
17 02141 126520         SUBZL   1,1             ;AC1 = 1
18 02142 175405         INC     3,3,SNR
19 02143 000445         JMP     IN3?+2
20 02144 147415         AND#    2,1,SNR         ;STAY IN LOOP UNTIL ALL BITS
21                                              ;OF SWREG ARE CHECKED
22 02145 000773         JMP     IN2?
23 02146 106400         SUB     0,1             ;WHEN THE CONTROL COMES HERE
24                                              ;FOR THE FIRST TIME AC1 = 100
25 02147 135000         MOV     1,3
26 02150 151225         MOVZR   2,2,SNR
27 02151 000444         JMP     IN4?
28 02152 024715         LDA     1,PC175         ;AC1 = 15
29 02153 167004         ADD     3,1,SZR         ;SKIP IF THE COMMAND WAS "M"
30 02154 000765         JMP     IN2?+1
31
32 02155 034600 INM?:   LDA     3,PTB?
33 02156 054727         STA     3,SVP?TB                ;SAVE PTB?
34 02157 034634         LDA     3,PC40          ;PREPARE TO PRINT A SPACE
35                                              ;AFTER EACH NUMBER
36 02160 054655         STA     3,PTB?
37 02161 006074         JSR     @ICLF?          ;TYPE A "CR" AND "LF"
38 02162 006070         JSR     @IPDE?          ;PRINT THE CONTENTS OF AC1
39 02163 034647         LDA     3,PC12?         ;AC3 = 12
40 02164 125400         INC     1,1
41 02165 166452         SUBO#   3,1,SZC         ;SKIP IF AC1 IS GREATER OR EQUAL
42                                              ;TO AC3
43 02166 004650         JSR     TPS?            ;TYPE A SPACE
44 02167 101220         MOVZR   0,0             ;AC0 = 20
45 02170 122414         SUB#    1,0,SZR         ;SKIP AFTER TYPING # 15
46 02171 000771         JMP     INM?+5
47 02172 006074         JSR     @ICLF?
48 02173 030065         LDA     2,SWREG
49 02174 151140         MOVOL   2,2             ;AC2 HAD SWITCH SETTINGS
50 02175 120560         SUBCL   1,1             ;BRING THE CARRY BIT IN AC1
51 02176 006072         JSR     @IZOC?          ;TYPE THE CONTENTS OF AC1
52 02177 004637         JSR     TPS?            ;TYPE A SPACE
53 02200 151124         MOVZL   2,2,SZR         ;SKIP AFTER TYPING ALL THE 15
54                                              ;BITS
55 02201 000774         JMP     .-4
56 02202 006074         JSR     @ICLF?
57 02203 034702         LDA     3,SVP?TB
58 02204 054631         STA     3,PTB?          ;RESTORE PTB?
59 02205 000410         JMP     IN4?
60
```

```
 0047 MNSCM
01 02200 176000 IN3?:   ADC     3,3             ;AC3 = -1
02 02201 054666         STA     3,INL?K         ;LOCK IN SWITCH INPUT MODE
03 02202 024065         LDA     1,SWREG         ;READ THE CURRRENT VALUE OF
04                                              ;"SWREG"
05 02203 133414         AND#    1,2,SZR         ;TAKE XOR OF AC1 AND AC2
06 02204 146401         SUB     2,1,SKP
07 02205 147000         ADD     2,1
08 02206 044065         STA     1,SWREG         ;SAVE THE NEW VALUE OF "SWREG"
09 02207 014660 IN4?:   ISZ     INL?K           ;SKIP IF THE PROGRAM IS LOCKED
10                                              ;IN SWITCH INPUT MODE
11 02210 000434         JMP     INR?
12 02211 014656         DSZ     INL?K           ;NEVER SKIP
13 02212 020125         LDA     0,LASTI
14 02213 101122         MOVZL   0,0,SZC
15 02214 000406         JMP     .+6
16 02215 063610         SKPDN   TTI     ;WAIT FOR OPERATOR INPUT
17 02216 000774         JMP     .-4
18 02217 101000         MOV     0,0
19 02220 060610         DIAC    0,TTI   ;GET CHARACTER
20 02221 000402         JMP     .+2
21 02222 101220         MOVZR   0,0
22 02223 040125         STA     0,LASTI ;PUT IN LASTI
23 02224 000667         JMP     IN0?+1
24
25 02225 002037 ITYP?:  TYP?
26 02226 107000 IN5?:   ADD     0,1             ;AC1 = 100+ ASCII VALUE OF
27                                              ;CONTROL CHARACTER
28 02227 020634         LDA     0,IN1?36                ;AC0 = 136
29 02230 006775         JSR     @ITYP?          ;TYPE A
30 02231 121000         MOV     1,0
31 02232 006773         JSR     @ITYP?
32 02233 034632         LDA     3,IN1?22                ;AC3 = 122
33 02234 116405         SUB     0,3,SNR         ;SKIP IF IT IS NOT AR
34 02235 000405         JMP     IN6?
35 02236 034626         LDA     3,IN1?04                ;AC3 = 104
36 02237 116404         SUB     0,3,SZR         ;SKIP IF IT WAS A AD
37 02240 000747         JMP     IN4?
38 02241 054065         STA     3,SWREG         ;SET SWITCHHES TO DEFAULT
39                                              ;MODE
40 02242 034634 IN6?:   LDA     3,INS?          ;AC3 = ADDRESS OF THE LOCATION
41                                              ;WHERE THE PROGRAM WILL START
42 02243 054625         STA     3,INR?T
43 02244 006774 INR?:   JSR     @ICLF?
44 02245 014613         ISZ     INT?
45 02246 030065         LDA     2,SWREG
46 02247 176220         ADCZR   3,3             ;AC3 = 77777
47 02250 173404         AND     3,2,SZR         ;SKIP IF THE SWITCHES ARE SET
48                                              ;TO ALL ZERO'S
49 02251 172000         ADC     3,2
50 02252 050065         STA     2,SWREG
51 02253 020622         LDA     0,INS?C         ;RESTORE CARRY
52 02254 101100         MOVL    0,0             ;
53 02255 020614         LDA     0,INS?0
54 02256 024614         LDA     1,INS?1
55 02257 030614         LDA     2,INS?2         ;RESTORE THE ACCUMULATORS
56 02260 034614         LDA     3,INS?3         ;RETURN WITH C(AC3)=CHAR INPUT
57 02261 002607         JMP     @INR?T          ;RETURN
```

```
10048 MNSCM
01
02                      ;RANGN=RANDOM # GENERATOR
03                      ;SPIN #'S OUT IN A HURRY FORGET THE MATH
04 02270 044427 RANGN:  STA 1,RN.S1
05 02271 054427         STA 2,RN.S2
06 02272 030430         LDA 2,RN.K1     ;7 FOR MASKING AT 8
07 02273 020430         LDA 0,RN.C2     ;CYCLIC CONSTANT
08 02274 024425         LDA 1,RN.C1
09 02275 133404         AND 1,2,SZR     ;ROTAT C2 EVERY 8
10 02276 000404         JMP RAN.1
11 02277 101122         MOVZL 0,0,SZC
12 02300 101400         INC 0,0
13 02301 040422         STA 0,RN.C2
14 02302 024423 RAN.1:  LDA 1,RTABL
15 02303 133000         AUD 1,2         ;TO GET NXT SUM VAR
16 02304 025000         LDA 1,0,2
17 02305 123000         AUD 1,0
18 02306 041000         STA 0,0,2       ;NEW SUM IN VAR
19 02307 024415         LDA 1,RANNM     ;LAST RAN#
20 02310 123300         AUDS 1,0
21 02311 040413         STA 0,RANNM
22 02312 030406 RAN.2:  LDA 2,RN.S2
23 02313 024404         LDA 1,RN.S1
24 02314 010405         ISZ RN.C1
25 02315 001400         JMP 0,3
26 02316 001400         JMP 0,3
27 02317 000000 RN.S1:  0
28 02320 000000 RN.S2:  0
29 02321 000000 RN.C1:  0
30 02322 000007 RN.K1:  7
31 02323 123456 RN.C2:  123456
32 02324 000000 RANNM:  0
33 02325 002326 RTABL:  RTABL+1
34 02326 027247         027247
35 02327 145651         145651
36 02330 162724         162724
37 02331 071352         071352
38 02332 034565         034565
39 02333 116272         116272
40 02334 047135         047135
41 02335 113523         113523
```

```
⅛⅛49 MNSCM
01                ;ERROM - ERROR HANDLER - PRINT ALL ERR INFO
02                ;FIRST PRINT PRG# AND (AC'S)
03 ⅛2336 ⅛01122 ER.SS:  CO.LA
04 ⅛2337 ⅛40463 ERROM:  STA 0,ER.S0
05 ⅛2340 ⅛44463         STA 1,ER.S1
06 ⅛2341 ⅛50463         STA 2,ER.S2
07 ⅛2342 ⅛54463         STA 3,ER.S3
08 ⅛2343 ⅛20⅛65         LDA 0,SWREG
09 ⅛2344 1⅛1120         MOVZL 0,0
10 ⅛2345 1⅛3122         ADDZL 0,0,SZC    ;SW2=1 IS NO TYPE
11 ⅛2346 ⅛⅛⅛444         JMP EMEXI        ;EXIT NO PRINTING
12 ⅛2347 ⅛36767         LDA 3,@ER.SS
13 ⅛2350 ⅛214⅛1         LDA 0,1,3        ;GET TEXT ADDRESS
14 ⅛2351 ⅛4⅛4⅛2         STA 0,.+2        ;PUT IN LOC FOLLOWING CALL TO PRINT
15 ⅛2352 ⅛⅛6⅛71         JSR @LMESS       ;PRINT ERROR MESSAGE
16 ⅛2353 ⅛⅛⅛⅛⅛⅛         ⅛                ;LOC OF TEXT
17 ⅛2354 ⅛24446         LDA 1,ER.S0      ;PRINT AC'S
18 ⅛2355 ⅛⅛6⅛73         JSR @LPUCT       ;AT ERROR CALL
19 ⅛2356 ⅛24445         LDA 1,ER.S1
20 ⅛2357 ⅛⅛6⅛73         JSR @LPUCT
21 ⅛2360 ⅛24444         LDA 1,ER.S2
22 ⅛2361 ⅛⅛6⅛73         JSR @LPUCT
23 ⅛2362 ⅛24115         LDA 1,PRFLG      ;GET PRINT FLAG FOR AC3
24 ⅛2363 125⅛⅛5         MOV 1,1,SNR
25 ⅛2364 ⅛⅛⅛4⅛3         JMP .+3          ;DON'T PRINT AC3
26 ⅛2365 ⅛24116         LDA 1,PR.S3      ;GET SAVED AC3
27 ⅛2366 ⅛⅛6⅛73         JSR @LPUCT       ;PRINT IT
```

```
⅛⅛50 MNSCM
01
02
03                ;PRINT MEM ALLOCATION ASSIGNMENTS
04
05 ⅛2367 ⅛⅛6⅛71 EMMPP:  JSR @LMESS
06 ⅛2370 ⅛⅛2612          TXT.2
07 ⅛2371 ⅛24137          LDA 1,SCRLO
08 ⅛2372 ⅛⅛6⅛73          JSR @LPUCT       ;PRINT SCRATCH LIMITS
09 ⅛2373 ⅛24144          LDA 1,SCRHI
10 ⅛2374 ⅛⅛6⅛73          JSR @LPUCT
11 ⅛2375 ⅛⅛6⅛71          JSR @LMESS
12 ⅛2376 ⅛⅛2672          TXT.8
13 ⅛2377 ⅛24135          LDA 1,TSTLO
14 ⅛2400 ⅛⅛6⅛7⅛          JSR @PDECI       ;PRINT TSTLO
15 ⅛2401 ⅛24136          LDA 1,TSTHI
16 ⅛2402 ⅛⅛6⅛7⅛          JSR @PDECI       ;PRINT TSTHI
17 ⅛2403 ⅛⅛6⅛71          JSR @LMESS
18 ⅛2404 ⅛⅛2664          TXT.7
19 ⅛2405 ⅛24133          LDA 1,PHSLO
20 ⅛2406 ⅛⅛6⅛7⅛          JSR @PDECI       ;PRINT PHSLO
21 ⅛2407 ⅛24134          LDA 1,PHSHI
22 ⅛241⅛ ⅛⅛6⅛7⅛          JSR @PDECI       ;PRINT PHSHI
23 ⅛2411 ⅛⅛4⅛⅛1          JMP EMEXI        ;DONE NOW EXIT
24
```

```
10051 MNSCM
01 02412 03u413 EREXI:  LDA 2,ER.S3      ;GET RETRN ADDRESS
02 02413 151400         INC 2,2          ;INC TO RETURN ADDRESS
03 02414 05u411         STA 2,ER.S3
04 02415 004516         JSR KEY2W
05 02416 02u404 ERFXT:  LDA 0,ER.S0
06 02417 u2404          LDA 1,ER.S1
07 02420 03u404         LDA 2,ER.S2
08 02421 002404         JMP @ER.S3
09
10 02422 000000 ER.S0:  0
11 02423 000000 ER.S1:  0
12 02424 000000 ER.S2:  0
13 02425 000000 ER.S3:  0
14
```

```
10052 MNSCM
01
02                      ;ERROR-ERROR HANDLER WITHOUT PRINT OF AC3
03                      ;ERRO3- "    "          WITH    "       "
04
05 02426 054456 ERRO3:  STA 3,ERR.3      ;SAVE RETURN ADDR.
06 02427 170000         ADC 3,3
07 02430 000403         JMP .+3
08 02431 054453 ERROR:  STA 3,ERR.3
09 02432 176400         SUB 3,3
10 02433 054115         STA 3,PRFLG      ;SET PRINT OF AC3 FLG
11 02434 050447         STA 2,ERR.2
12 02435 044445         STA 1,ERR.1
13 02436 040443         STA 0,ERR.0      ;SAVE THE AC'S
14 02437 036445         LDA 3,@ERR.3     ;GET TEXT ADDRESS
15 02440 054402         STA 3,ERRMS      ;PLACE IN ERROR CALL
16 02441 006100 ER.PR:  JSR @ERROI       ;PRINT ERROR MESS
17 02442 000000 ERRMS:  0                ;TXT ADDRESS
18 02443 030114 ERNPR:  LDA 2,PSTRT      ;POINTER TO XX,00
19 02444 011377         ISZ -1,2         ;INC TEST'S ERROR COUNTER
20 02445 000401         JMP .+1
21 02446 011375         ISZ -3,2         ;#OF ERRS THIS PASS
22 02447 000401         JMP .+1
23 02450 010130         ISZ ERTOT        ;INC ERROR TOTAL
24 02451 000401         JMP .+1
25 02452 020130         LDA 0,ERTOT
26 02453 024432         LDA 1,ERR.5
27 02454 106404         SUB 0,1,SZR      ;50 ERRORS?
28 02455 000410         JMP .+8.
29 02456 006071         JSR @LMESS
30 02457 002620         TXT,3            ;50 ERRORS TEXT
31 02460 060277         INTDS
32 02461 063610         SKPDN TTI
33 02462 000777         JMP .-1          ;WAIT FOR KEY ENTRY
34 02463 060210         NIOC TTI
35 02464 060177         INTEN
36 02465 034065         LDA 3,SWREG
37 02466 177122         ADDZL 3,3,SZC    ;LOOP?
38 02467 000404         JMP ERRRT        ;NO,RETURN
39 02470 120000         ADC 1,1          ;YES,RESTART TEST
40 02471 133000         ADD 1,2
41 02472 050406         STA 2,ERR.L
42 02473 020406 ERRRT:  LDA 0,ERR.0
43 02474 024406         LDA 1,ERR.1
44 02475 030406         LDA 2,ERR.2
45 02476 034406         LDA 3,ERR.3      ;RESTORE THE AC'S
46 02477 001401         JMP 1,3          ;RETURN TO CDISP TO RESTART
47 02500 001122 ERR.L:  CD.LA
48 02501 000000 ERR.0:  0
49 02502 000000 ERR.1:  0
50 02503 000000 ERR.2:  0
51 02504 000000 ERR.3:  0        ;RETURN ADDRESS
52 02505 000062 ERR.5:  50.
```

```
↓0053 MNSCM
01
02 02506 120240 KYMES:  .TXTE ↓  TYPE ANY KEY↓
03               ;TEXT CALL ADRS OF TEXT IS IN AC0
04               ;CALL MUST ONLY BE MADE WHILE IN FIRST LEVEL TEST
05 02510 040704 EMTXT:  STA 0,ER,S0
06 02517 040411         STA 0,ER,TP
07 02520 044003         STA 1,ER,S1
08 02521 050703         STA 2,ER,S2
09 02522 054703         STA 3,ER,S3
10 02523 024065         LDA 0,SWREG
11 02524 101120         MOVZL 0,0
12 02525 103112         ADDL# 0,0,SZC    ;OK TO PRINT ERROR TXT?
13 02526 000403         JMP ER,TP+1
14 02527 006071         JSR @LMESS
15 02530 000000 ER,TP:  0                ;TEXT ADRS STORED HERE
16 02531 034674         LDA 3,ER,S3
17 02532 000664         JMP EREXT        ;EXIT
18
19
20               ;THIS SUBROUTINE WAITS FOR TTI INPUT IF SW6=1
21 02533 054421 KEY2W:  STA 3,KEY,3
22 02534 050421         STA 2,KEY,2
23 02535 034065         LDA 3,SWREG
24 02536 030420         LDA 2,KEY,6
25 02537 157405         AND 2,3,SNR      ;BIT 6 IS HALT ON ERROR
26 02540 000412         JMP KEY,3-2
27 02541 006071         JSR @LMESS
28 02542 002506         KYMES
29 02543 060177         INTEN
30 02544 176400         SUB 3,3
31 02545 054125         STA 3,LASTI      ;SET FLG
32 02546 034125         LDA 3,LASTI      ;CHECK FLG
33 02547 175103         MOVL 3,3,SNC     ;SKP = TTI INTR
34 02550 000776         JMP .-2
35 02551 006231         JSR @KEY,I       ;SERVICE TTI REQUEST
36 02552 030403         LDA 2,KEY,2
37 02553 002401         JMP @KEY,3       ;RETURN
38 02554 000000 KEY,3:  0
39 02555 000000 KEY,2:  0
40 02556 001700 KEY,6:  176
```

```
↓0054 MNSCM
01 02557 000215 TXT,0:  .TXTE ↓<15><12>ENTER MEMORY LIMITS (PHSLO,PHSHI) IN DECI
02 02612 000215 TXT,2:  .TXTE (<15><12>
03 02613 141523 SCRLO/HI (
04 02620 000215 TXT,3:  .TXTE ↓<15><12>TOTAL ERRORS = 50<11>TYPE ANY KEY<15><12>
05 02642 000215 TXT,4:  .TXTE ↓<15><12>ENTER ANY OPTIONS,CR TO CONTINUE↓
06 02664 000215 TXT,7:  .TXTE ↓<15><12>PHSLO/HI ↓
07 02672 000215 TXT,8:  .TXTE ↓<15><12>TSTLO/HI ↓
08 02700 000215 MM,T1:  .TXTE ↓<15><12>DATA = ADDR TEST<15><12>
09 02712 141501 ACTUAL0EXPECTED0LOCATION<15><12>↓
10 02730 000215 M1,T1:  .TXTE ↓<15><12>ISZ TEST-STORE0VERIFY MINUS ONE
11 02750 100705 <15><12>ACTUAL0EXPECTED0LOCATION<15><12>↓
12 02767 000215 M1,T2:  .TXTE ↓<15><12>FORW,ISZ TEST-WORD NOT =1 BEFORE ISZ<15><
13 03013 141501 ACTUAL0EXPECTED0LOCATION<15><12>↓
14 03031 000215 M1,T3:  .TXTE ↓<15><12>FORW,ISZ TEST-READ0VERIFY<15><12>
15 03047 040412 ACTUAL0EXPECTED0LOCATION<15><12>↓
16 03065 000215 M1,T4:  .TXTE ↓<15><12>REV,ISZ TEST-WORD NOT =1 BEFORE ISZ<15><1
17 03110 040412 ACTUAL0EXPECTED0LOCATION<15><12>↓
18 03126 000215 M1,T5:  .TXTE ↓<15><12>REV,ISZ TST- READ0VERIFY<15><12>
19 03144 141501 ACTUAL0EXPECTED0LOCATION<15><12>↓
20 03162 000215 M1,T6:  .TXTE ↓<15><12>FORW, ISZ TEST=ISZ DIDN'T SKIP
21 03202 000215 <15><12>ACTUAL0EXPECTED0LOCATION<15><12>↓
22 03221 000215 M1,T7:  .TXTE ↓<15><12>FORW, ISZ TEST=WORD NOT =1 AFTER DSZ<15><
23 03245 141501 ACTUAL0EXPECTED0LOCATION<15><12>↓
24 03263 000215 M1,T8:  .TXTE ↓<15><12>EREV, ISZ TST-ISZ DIDN'T SKIP<15><12>
25 03303 040412 ACTUAL0EXPECTED0LOCATION<15><12>↓
26 03321 000215 M2,T1:  .TXTE ↓<15><12>MARCH (FORWARD)<15><12>
27 03332 040412 ACTUAL0EXPECTED0LOCATION0BACKGND<15><12>↓
28 03354 000215 M2,T2:  .TXTE ↓<15><12>MARCH (REVERSE)<15><12>
29 03365 040412 ACTUAL0EXPECTED0LOCATION0BACKGND<15><12>↓
30 03407 000215 M3,T1:  .TXTE ↓<15><12>MASEST (COMPLEMENT TEST)<15><12>
31 03425 024303 C(X)    C/(Y)   LOC(X)   LOC(Y) <15><12>↓
32 03447 000215 M3,T2:  .TXTE ↓<15><12>MASEST (BACKGND)<15><12>
33 03461 141501 ACTUAL0EXPECTED0LOCATION<15><12>↓
34 03477 000215 M5,T1:  .TXTE ↓<15><12>GALPAT<15><12>
35 03504 024303 C(Y)    C(X)   LOC(Y)  LOC(X)<15><12>↓
36 03524 000215 M4,T1:  .TXTE ↓<15><12>SDIAG TEST<15><12>
37 03533 141501 ACTUAL0EXPECTED0LOCATION0BACKGROUND<15><12>↓
38 03556 000215 M6,T1:  .TXTE ↓<15><12>GALWREC (X AFTER WRITE INV, BACKGND)
39 03601 000215 <15><12>EXPECTED0ACTUAL0LOC(X)0 LOC(Y)<15><12>↓
40 03623 000215 M6,T2:  .TXTE ↓<15><12>GALWREC (X AFTER WRITE BACKGND)<15><12>
41 03644 142412 EXPECTED0ACTUAL0 LOC(X) 0LOC(Y)<15><12>↓
```

```
↓0055 MNSCM
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 03666 000230           LAUTO
07 03667 054466 LPRSL:    STA 3,LPSV3
08 03670 122400           SUB 0,0
09 03671 044465           STA 0,LPRGN
10 03672 020471           LDA 0,LPRT1       ;FIRST TXT ADRS
11 03673 006075           JSR @ERRTX
12 03674 020463           LDA 1,@LPHIK      ;GET HIGHR PHYS.
13 03675 125400           INC 1,1           ;=MOD 1K
14 03676 006070           JSR @PDECI        ;PRINT SUPR 0'S
15 03677 026767           LDA 1,@LPRSL=1
16 03700 020513           LDA 0,LPRT2
17 03701 125004           MOV 1,1,SZR       ;SKP IF AUTO STRT
18 03702 020524           LDA 0,LPRT3       ;NOT AUTO USE OTHR HDR
19 03703 006075           JSR @ERRTX
20 03704 020545           LDA 0,LPRT4
21 03705 006075           JSR @ERRTX        ;PRT PRG HDR
22 03706 000074           JSR @PCRLF        ;CARRET LFEED TST 0
```

```
↓0056 MNSCM
01                ;PRINT INDIVIDUAL TEST DESCRIPTIONS
02                ;GIVE OPR CHANCE TO DELETE IF LAUTO=-1
03 03707 020447 LPRLP:    LDA 0,LPRGN       ;CUR PROG #
04 03710 034104           LDA 3,TTBL
05 03711 117000           ADD 0,3
06 03712 031400           LDA 2,0,3
07 03713 151005           MOV 2,2,SNR       ;SKP IF NOT LAST
08 03714 002441           JMP @LPSV3
09 03715 125000           MOV 0,1
10 03716 021001           LDA 0,1,2         ;GET PRG WAIT SW
11 03717 101004           MOV 0,0,SZR       ;SKP IF PRG NOT WAIT
12 03720 000432           JMP LPR1E         ;TEST MUST BE DELETED
13 03721 050441           STA 2,LPIDX
14 03722 000072           JSR @PZOCT        ;TYPE PRG #
15 03723 020437           LDA 0,LPIDX
16 03724 101400           INC 0,0
17 03725 101400           INC 0,0
18 03726 101400           INC 0,0
19 03727 000075           JSR @ERRTX                ;PRINT DESCR OF PGM
20 03730 026736           LDA 1,@LPRSL=1
21 03731 125005           MOV 1,1,SNR       ;SKP IS LET OPR SELECT
22 03732 000417           JMP LPR1F=1       ;CR/LF AND DO NXT PRG
```

```
!0057 MNSCM
01                ;WAIT FOR OPERATOR INPUT TO SELECT TEST
02                ;SPACE IS SELECT ANY OTHER IS DELETE
03 03735 060277           INTDS
04 03734 063610           SKPDN TTI
05 03735 000777           JMP .-1
06 03736 064610           DIAC 1,TTI        ;GET CHAR
07 03737 030421           LDA 2,LPR77
08 03740 147400           AND 2,1
09 03741 030420           LDA 2,LPR40
10 03742 146415           SUB# 2,1,SNR      ;SKP IF DELETED
11 03743 000406           JMP LPR1E-1       ;SELECTED CR/LF DO NEXT
12 03744 030416           LDA 2,LPIDX
13 03745 102000           ADC 0,0
14 03746 041001           STA 0,1,2         ;SET WAIT SW IN PRG
15 03747 020511           LDA 0,LPDTX
16 03750 006075           JSR @ERRTX        ;PRINT DELETED
17 03751 006074           JSR @PCRLF        ;CR/LF
18 03752 010404 LPR1E:    ISZ LPRGN         ;+1 PROG #
19 03753 060177           INTEN
20 03754 000733           JMP LPRLP         ;DO NEXT PROG
21 03755 000000 LPSV3:    0
22 03756 000000 LPRGN:    0
23 03757 000132 LPHIK:    PHSHX
24 03760 000077 LPR77:    77
25 03761 000040 LPR40:    40
26 03762 000000 LPIDX:    0
27 03763 003764 LPRT1:    .+1
28 03764 005215           .TXTE [<15><12>MDM SC-MEMORY TEST  REV 00
29 04002 005215 <15><12>TOTAL # 1K'S = ]
30 04013 004014 LPRT2:    .+1
31 04014 005215           .TXTE (<15><12>PROGRAM RUN LIST(
32 04026 004027 LPRT3:    .+1
33 04027 005215           .TXTE (<15><12> SPACE SELECTS-ALL OTHERS  DELETE(
34 04051 004052 LPRT4:    .+1
35 04052 005215           .TXTE (<15><12>PRG#      NAME(
36 04060 004061 LPDTX:    .+1
37 04061 042240           .TXTE    ( DELETED(
38
```

```
!0058 MNSCM
01                ;WRBKG=WRITE CONTENTS OF AC0 INTO BACKGROUND
02 04066 054417 WRBKG:    STA 3,WRBSV       ;SAVE RETURN ADDRESS
03 04067 030137           LDA 2,SCRLO
04 04070 034140           LDA 3,SCRHI       ;GET  LOW/HI MEMORY LIMITS
05 04071 041000           STA 0,0,2         ;WRITE INTO BACKGROUND
06 04072 151400           INC 2,2           ;INC ADDRESS
07 04073 156432           SUBZ# 2,3,SZC     ;SKP = > SCRHI
08 04074 000775           JMP WRBKG+3       ;CONTINUE
09 04075 002401           JMP  @WRBSV       ;RETURN TO CALLER
10
11 04076 000000 WRBSV:    0                 ;SAVE RETURN LOC.
```

```
10059 MNSCM
01                      ;MEMORY TEST #0   DATA EQUALS ADDRESS
                                .TITL DEQADR
03                              NEXIT M0,37,DATA=ADDR
04                      ;DEFINITION TO LINKR PARAMETERS FOLLOWS
05              004077  LMEML=.
06              000141          .LOC LPG0
07  000141     004102           M0.00
08              000142  LPG0=.
09              004077          .LOC LMEML
10  004077     000000           0                       ;TST ERR CTR THIS PASS
11  004100     000000           0                       ;TEST PASS COUNTER
12  004101     000000           0                       ;TEST ERROR COUNTER
13  004102     000410  M0.00:   JMP M0.02                       ;EXECUTE ENTRY ADDRESS
14  004103     000000           0                               ;WAIT SWITCH
15  004104     000037           37                              ;MAXIMUM SEGMENT SIZE
16  004105     004504           .TXTE [DATA=ADDR]
17              004724
18              040675
19              042164
20              000322
21                      ;EXECUTE ENTRY
22  004112     030137  M0.02:   LDA 2,SCRLO
23  004113     034140           LDA 3,SCRHI     ;SET LIMIT OF TEST
24  004114     051000  M0.03:   STA 2,0,2       ;PUT ADDRESS IN LOC.
25  004115     151400           INC 2,2
26  004116     156432           SUBZ# 2,3,SZC   ;WRITTEN ALL?
27  004117     000775           JMP M0.03       ;NOPE
28  004120     030137  M0.04:   LDA 2,SCRLO
29  004121     021000           LDA 0,0,2       ;GET DATA
30  004122     145000           MOV 2,1         ;SET AC1=ADDRESS
31  004123     106435           SUBZ# 0,1,SNR   ;CHECK VALUE READ
32  004124     000403           JMP M0.4C
33                              LCALL ERROX     ;ERROR CALL
34  004126     002700           M0.T1           ;TXT ADDRESS
35  004127     151400  M0.4C:   INC 2,2
36  004130     156432           SUBZ# 2,3,SZC   ;DONE?
37  004131     000770           JMP M0.04+1     ;NOT YET
38                              LCALL RETRN     ;RETURN TO LINKR
```

```
10060 MNSCM
01
02                      ;MEMORY TEST #1  ISZ INSTRUCTION TEST
                                .TITL ISZT
04                              NEXTT M1,37,ISZ=TEST
05                      ;DEFINITION TO LINKR PARAMETERS FOLLOWS
06              004133  LMEML=.
07              000142          .LOC LPG0
08  000142     004136           M1.00
09              000143  LPG0=.
10              004133          .LOC LMEML
11  004133     000000           0                       ;TST ERR CTR THIS PASS
12  004134     000000           0                       ;TEST PASS COUNTER
13  004135     000000           0                       ;TEST ERROR COUNTER
14  004136     000410  M1.00:   JMP M1.02                       ;EXECUTE ENTRY ADDRESS
15  004137     000000           0                               ;WAIT SWITCH
16  004140     000037           37                              ;MAXIMUM SEGMENT SIZE
17  004141     051711           .TXTE [ISZ=TEST]
18              026532
19              142724
20              152123
21              000000
22
23                      ;EXECUTE ENTRY
24  004146     030137  M1.02:   LDA 2,SCRLO
25  004147     034140           LDA 3,SCRHI     ;GET TEST LIMITS
26  004150     126000           ADC 1,1
27  004151     045000           STA 1,0,2       ;PUT -1 IN SCRATCH AREA
28  004152     021000           LDA 0,0,2       ;VERIFY IT GOT THERE
29  004153     106435           SUBZ# 0,1,SNR   ;DID IT?
30  004154     000403           JMP M1.2C
31                              LCALL ERROX
32  004156     002730           M1.T1
33  004157     151400  M1.2C:   INC 2,2
34  004160     156432           SUBZ# 2,3,SZC   ;DONE ALL
35  004161     000770           JMP M1.02+3     ;NOT YET!
36
37                      ;FORWARD ISZ TEST
38  004162     030137  M1.03:   LDA 2,SCRLO
39  004163     021000           LDA 0,0,2
40  004164     100015           COM# 0,0,SNR    ;LOC =1 BEFORE ISZ?
41  004165     000404           JMP M1.3C       ;NO ERROR,SKIP
42  004166     126000           ADC 1,1
43                              LCALL ERROX     ;ERROR CALL
44  004170     002767           M1.T2           ;TXT ADDR,
45  004171     011000  M1.3C:   ISZ 0,2         ;ISZ LOCATION X
46  004172     000402           JMP .+2         ;ISZ DIDN'T SKIP,ERROR
47  004173     000406           JMP M1.3D       ;NO,ERROR,SKIP
48  004174     021000           LDA 0,0,2       ;GET LOCATION
49  004175     126400           SUB 1,1         ;EXPECTED
50                              LCALL ERROX     ;ERROR CALL
51  004177     003162           M1.T6           ;TXT ADDR
52  004200     045000           STA 1,0,2       ;RESORE LOCATION
53  004201     021000  M1.3D:   LDA 0,0,2
54  004202     101005           MOV 0,0,SNR     ;LOC 0 AFTER ISZ?
55  004203     000405           JMP M1.3E       ;NO ERROR ,SKIP
56  004204     126400           SUB 1,1
57                              LCALL ERROX     ;ERROR CALL
58  004206     003031           M1.T3           ;TXT ADDRESS
59  004207     045000           STA 1,0,2       ;RESTORE LOCATION
60  004210     015000  M1.3E:   DSZ 0,2         ;-1 LOC AGAIN
```

```
 0001 MNSCM
01 04211 101000          MOV 0,0
02 04212 021000          LDA 0,0,2       ;GET LOC AFTER ISZ
03 04213 100015          COM# 0,0,SNR    ;WAS IT -1?
04 04214 000405          JMP M1.3F               ;NO ERROR,SKIP
05 04215 126000          ADC 1,1
06                       LCALL ERROX
07 04217 003221          M1.T7
08 04220 045000          STA 1,0,2       ;RESTORE LOCATION AFTER ERROR
09 04221 151400 M1.3F:   INC 2,2         ;NEXT LOC
10 04222 156432          SUBZ# 2,3,SZC   ;DONE ALL?
11 04223 000740          JMP M1.03+1     ;NOT YET
12 04224 000401          JMP M1.04
13
```

```
 0002 MNSCM
01                       ;REVERSE ISZ TEST
02 04225 030140 M1.04:   LDA 2,SCRMI     ;STARTING ADDR
03 04226 034137          LDA 3,SCRLO     ;FINISHING ADDR
04 04227 126000          ADC 1,1
05 04230 021000          LDA 0,0,2       ;GET LOC BEFORE ISZING
06 04231 100015          COM# 0,0,SNR    ;IS IT = -1?
07 04232 000404          JMP M1.4C       ;NO ERROR,SKIP
08                       LCALL ERROX
09 04234 003205          M1.T4
10 04235 045000          STA 1,0,2       ;RESTORE LOCATION
11 04236 011000 M1.4C:   ISZ 0,2         ;ISZ THIS LOCATION
12 04237 000402          JMP .+2         ;ISZ DIDN'T SKIP=ERROR
13 04240 000406          JMP M1.4D       ;NO ERROR ,SKIP
14 04241 021000          LDA 0,0,2
15 04242 126400          SUB 1,1
16                       LCALL ERROX
17 04244 003203          M1.T8
18 04245 045000          STA 1,0,2       ;RESTORE LOCATION
19 04246 021000 M1.4D:   LDA 0,0,2
20 04247 101005          MOV 0,0,SNR     ;LOC 0 AFTER ISZ?
21 04250 000404          JMP M1.4E       ;NO ERROR,SKIP
22 04251 126400          SUB 1,1
23                       LCALL ERROX
24 04253 003126          M1.T5
25 04254 126000 M1.4E:   ADC 1,1
26 04255 133000          ADD 1,2         ;DECREMENT AC2 TO NEXT LOC
27 04256 172432          SUBZ# 3,2,SZC   ;DONE ALL?
28 04257 000751          JMP M1.04+3     ;NO
29                       LCALL RETRN     ;DONE, RETURN  TO LINKR
```

```
00063 MNSCM
01
02                ;MEMORY TEST #2.......MARCH.....
                  .TITL MARCH
04                NEXTT M2,37,MARCH
05                ;DEFINITION TO LINKR PARAMETERS FOLLOWS
06        004261 LMEML=.
07        000143          .LOC LPG0
08 00143 004264           M2.00
09        000144 LPG0=.
10        004261          .LOC LMEML
11 04261 000000           0                       ;TST ERR CTR THIS PASS
12 04262 000000           0                       ;TEST PASS COUNTER
13 04263 000000           0                       ;TEST ERROR COUNTER
14 04264 000407 M2.00:    JMP M2.02                       ;EXECUTE ENTRY ADDRESS
15 04265 000000           0                               ;WAIT SWITCH
16 04266 000037           37                              ;MAXIMUM SEGMENT SIZE
17 04267 040515           .TXTE ;MARCH1
18        141722
19        000110
20
21                ;EXECUTE ENTRY POINT
22 04272 020000           182
23 04273 020120 M2.02:    LDA 0,SWRG2     ;CHECK FOR FLG SET
24 04274 126000           ADC 1,1
25 04275 044467           STA 1,M2.DR     ;SET PASS SWITCH
26 04276 024774           LDA 1,M2.02-1
27 04277 123405           AND 1,0,SNR     ;SKP IS USE ALL 1'S PATTERN
28 04300 000403           JMP .+3         ;USE RANDOM DATA
29 04301 100000           ADC 0,0         ;ALL ONES
30 04302 000402           JMP M2.2A       ;
31                        LCALL ARANG     ;GET RANDOM #
32 04304 040457 M2.2A:    STA 0,M2,SV     ;SAVE IT
33                        LCALL LWRBK     ;FILL BACKGROUND SUBROUTINE
34 04306 034140           LDA 3,SCRHI
35
36 04307 030137 M2.03:    LDA 2,SCRLO     ;GET STARTING ADDRESS
37 04310 024453           LDA 1,M2,SV     ;GET SAVED DATA VALUE
38 04311 021000           LDA 0,0,2       ;READ DATA WORD
39 04312 106415           SUB# 0,1,SNR    ;SAME AS WRITTEN?
40 04313 000406           JMP M2.3C       ;NO ERROR ,SKIP
41 04314 034447           LDA 3,M2,SV
42 04315 054116           STA 3,PR,S3     ;SET UP FOR AC3 PRINT
43                        LCALL ERROY     ;ERROR PRINT WITH AC3
44 04317 003321           M2.T1
45 04320 034140           LDA 3,SCRHI
46 04321 120000 M2.3C:    COM 1,0         ;OK, NOW COMPLEMENT VALUE
47 04322 041000           STA 0,0,2       ;WRITE BACK COMPLEMENTED VALUE
48 04323 151400           INC 2,2         ;INC ADDRESS
49 04324 156432           SUBZ# 2,3,SZC   ;CHECK IF DONE ALL
50 04325 000764           JMP M2.03+2
51
52                ;MARCH AGAIN WITH COMPLEMENTED DATA PATTERN
53 04326 020435           LDA 0,M2.SV
54 04327 100000           COM 0,0
55 04330 040433           STA 0,M2.SV
56 04331 014433           ISZ M2.DR       ;SKP IF FIRST PASS
57 04332 000402           JMP M2.04       ;NOW DO REVERSE TEST
58 04333 000754           JMP M2.03
```

```
00064 MNSCM
01
02                ;REVERSE MARCH
03 04334 030140 M2.04:    LDA 2,SCRHI     ;GET STARTING ADDR..HI VALUE
04 04335 034137           LDA 3,SCRLO     ;GET END ADDRESS
05 04336 024425           LDA 1,M2.SV     ;GET SAVED DATA VALUE
06 04337 021000 M2.4A:    LDA 0,0,2       ;READ DATA WORD
07 04340 122415           SUB# 1,0,SNR    ;CHECK AGAINST SAVED DATA
08 04341 000405           JMP M2.4C       ;SKIP,NO ERROR
09 04342 044116           STA 1,PR,S3     ;SETUP AC3 FOR PRINT
10                        LCALL ERROY
11 04344 003334           M2.T2
12 04345 034137           LDA 3,SCRLO
13 04346 120000 M2.4C:    COM 1,0         ;COMPLEMENT
14 04347 041000           STA 0,0,2       ;WRITE BACK COMP. VALUE
15 04350 102520           SUBZL 0,0       ;+1
16 04351 112400           SUB 0,2         ;DECREMENT ADDRESS
17 04352 172432           SUBZ# 3,2,SZC   ;CHECK IF LESS THAN SCRLO
18 04353 000764           JMP M2.4A       ;CONTINUE
19                ;COMPLEMENT DATA AND REPEAT
20 04354 020407 M2.05:    LDA 0,M2.SV
21 04355 100000           COM 0,0
22 04356 040405           STA 0,M2.SV
23 04357 014405           DSZ M2.DR       ;DONE  BOTH PATTERNS?
24 04360 000402           JMP M2.EX       ;EXIT BOTH DONE
25 04361 000753           JMP M2.04
26                M2.EX:  LCALL RETRN     ;RETURN TO LINKR
27 04363 000000 M2.SV: 0                  ;SAVED DATA VALUE
28 04364 000000 M2.DR:  0                 ;PASS FLAG
```

```
10065 MNSCM
01                      ;MEMORY TEST #4
                                .TITL MSEST
03                              NEXTT M3,2,MASEST
04                      ;DEFINITION TO LINKR PARAMETERS FOLLOWS
05              004365 LMEML=.
06              000144          .LOC LPG0
07 000144 004370                M3.00
08              000145 LPG0=.
09              004365          .LOC LMEML
10 004365 000000                0                       ;TST ERR CTR THIS PASS
11 004366 000000                0                       ;TEST PASS COUNTER
12 004367 000000                0                       ;TEST ERROR COUNTER
13 004370 000407 M3.00:         JMP M3.02                       ;EXECUTE ENTRY ADDRESS
14 004371 000000                0                               ;WAIT SWITCH
15 004372 000002                2                               ;MAXIMUM SEGMENT SIZE
16 004373 004515                .TXTE ;MASEST;
17              142523
18              152123
19              000000
20
21
22                      ;EXECUTION ENTRY POINT
23                      ; FILL BACKGROUND WITH ALTERNATE WORDS OF ALL 1'S/0'S
24 004377 034137 M3.02:         LDA 2,SCRLO     ;GET STARTING ADDR
25 004400 034140                LDA 3,SCRHI     ;GET ENDING ADDR
26 004401 102400                SUB 0,0         ;GENERATE 0'S
27 004402 041000 M3.2A:         STA 0,0,2       ;STORE IN BACKGROUND
28 004403 151400                INC 2,2         ;STEP TO NEXT WORD
29 004404 156433                SUBZ# 2,3,SNC   ;CHEC IF DONE
30 004405 000424                JMP M3.05       ;VERIFY BACKGROUNND
31 004406 100000                COM 0,0         ;COMPLEMENT DATA
32 004407 000773                JMP M3.2A       ;CONTINUE WITH COMPLEMENTED VALUE
33                      ;
34                      ;FORM LIMITED COMPLEMENT OF ADDR IN AC2 INTO AC3
35 004410 034140 CMPAD:         LDA 3,SCRHI     ;
36 004411 024137                LDA 1,SCRLO     ;
37 004412 156400                SUB 2,3         ;AC3=SCRHI-AC2
38 004413 137000                ADD 1,3         ;AC3=SCRLO+SCRHI-AC2
```

```
10066 MNSCM
01                      ;CHECK ADDRESS CONTENTS/COMPLEMENTED ADDR CONTENTS
02 004414 021000 M3.03:         LDA 0,0,2       ;READ CONTENTS OF ADDR
03 004415 025400                LDA 1,0,3       ;READ CONTENTS OF COMPLEMENTED ADR
04 004416 124000                COM 1,1         ;COM THE LATTER
05 004417 106415                SUB# 0,1,SNR    ;CHECK ONE AGAINST THE OTHER
06 004420 000404                JMP M3.3C       ;SKIP,NO ERROR
07 004421 054116                STA 3,PC.53
08                              LCALL ERROY
09 004423 000347                M3.T1
10 004424 151400 M3.3C:         INC 2,2         ;INC ADDRESS
11 004425 020140                LDA 0,SCRHI     ;GET SCRHI
12 004426 142433                SUBZ# 2,0,SNC   ;DONE ALL WORDS?
13 004427 000421                JMP M3.EX       ;DONE
14 004430 000760                JMP CMPAD       ;DO THIS WORD
```

```
10067 MNSCM
01                      ;VERIFY THE BACKGROUND
02 004431 034140 M3.05:  LDA 3,SCRHI      ;GET SCRHI
03 004432 034137         LDA 2,SCRLO      ;GET STARTING ADDRESS
04 004433 126400         SUB 1,1          ;ZERO AC1
05 004434 021000 M3.5A:  LDA 0,0,2        ;READ BACKGROUND
06 004435 122415         SUB# 1,0,SNR     ;CHECK IT
07 004436 000403         JMP M3.5C        ;SKIP,NO ERROR
08                       LCALL ERROX
09 004440 003447         M3.T2
10 004441 151400 M3.5C:  INC 2,2          ;STEP TO NEXT WORD
11 004442 156433         SUBZ# 2,3,SNC    ;DONE ALL?
12 004443 000403         JMP M3.5D        ;DONE CHECKING
13 004444 124000         COM 1,1          ;COMPLEMENT EXPECTED VALUE
14 004445 000767         JMP M3.5A        ;DO NEXT WORD
15 004446 034137 M3.5D:  LDA 2,SCRLO
16 004447 100741         JMP CMPAD        ;FORM COMPLEMENT OF ADDRESS
17                M3.EX: LCALL RETRN      ;EXIT TO LINKR
```

```
10068 MNSCM
01                      ;MEMORY TEST 4
                        .TITL SDIAG
03                      NEXTT M4.1,SDIAG
04                      ;DEFINITION TO LINKR PARAMETERS FOLLOWS
05              004451 LMEML=.
06              000145          .LOC LPG0
07 000145 004454        M4.00
08              000146 LPG0=.
09              004451          .LOC LMEML
10 004451 000000                0               ;TST ERR CTR THIS PASS
11 004452 000000                0               ;TEST PASS COUNTER
12 004453 000000                0               ;TEST ERROR COUNTER
13 004454 000406 M4.00:  JMP M4.02              ;EXECUTE ENTRY ADDRESS
14 004455 000000                0               ;WAIT SWITCH
15 004456 000001                1               ;MAXIMUM SEGMENT SIZE
16 004457 042123                .TXTE |SDIAG|
17              004711
18              000107
19
20                      ;EXECUTE ENTRY
21 004462 102400 M4.02:  SUB 0,0
22 004463 040532         STA 0,BCKGN
23 004464 040532         STA 0,DIAG
24 004465 040525         STA 0,ROW
25
26                      ;FILL BACKGROUND
27 004466 030137 SD.03:  LDA 2,SCRLO
28 004467 034140         LDA 3,SCRHI
29 004470 020525         LDA 0,BCKGN
30 004471 041000 SD.LP:  STA 0,0,2              ;WRITE TO BACKGROUND
31 004472 151400         INC 2,2                ;STEP TO NEXT WORD
32 004473 156432         SUBZ# 2,3,SZC          ;SKP IS DONE ALL
33 004474 000775         JMP SD.LP
34
35                      ;GENERATE DIAGONAL PATTERN
36 004475 020521 SD.04:  LDA 0,DIAG
37 004476 040515         STA 0,CLMN
38 004477 030503         LDA 2,SD.32
39 004500 050501         STA 2,SDCNT            ;CNT = -32
40 004501 030137         LDA 2,SCRLO
41 004502 024510         LDA 1,ROW
42 004503 034107         LDA 3,K37
43 004504 167524         ANDZL 3,1,SZR
44 004505 127120         ADDZL 1,1
45 004506 127124         ADDZL 1,1,SZR          ;REPOSITION TO ROW BITS
46 004507 133000         ADD 1,2                ;PUT IN ADDRESS
47 004510 024503         LDA 1,CLMN
48 004511 167404         AND 3,1,SZR
49 004512 133000         ADD 1,2                ;ADD IN COLUMN
50 004513 024502         LDA 1,BCKGN
51 004514 124000         COM 1,1
52 004515 041000         STA 1,0,2              ;WRITE DIAGONAL WORD
53 004516 010474         ISZ ROW
54 004517 010474         ISZ CLMN
55 004520 010461         ISZ SDCNT              ;SKP IF DONE 32 WORDS
56 004521 000760         JMP SD.04+4            ;NOT YET!
```

```
10069 MNSCM
01                      ;READ AND VERIFY ARRAY CONTENTS
02 04522 102400 SD.05: SUB 0,0
03 04523 040467         STA 0,ROW
04 04524 040467         STA 0,CLMN
05 04525 030137         LDA 2,SCRLD
06 04526 024464         LDA 1,ROW
07 04527 034107         LDA 3,K37
08 04530 167524         ANDZL 3,1,SZR
09 04531 127120         ADDZL 1,1
10 04532 127124         ADDZL 1,1,SZR
11 04533 133000         ADD 1,2                 ;ADD IN THE ROW BITS
12 04534 024457         LDA 1,CLMN
13 04535 167404         AND 3,1,SZR
14 04536 133000         ADD 1,2                 ;ADD IN COLUMN TO ADDRESS
15 04537 021000         LDA 0,0,2               ;GET WORD IN ARRAY
16 04540 050454         STA 2,SD.AD             ;SAVE ADDRESS FOR LATER
17 04541 030455         LDA 2,DIAG
18 04542 140400         SUB 2,1                 ;CLMN = DIAG = ROW
19 04543 167400         AND 3,1
20 04544 030446         LDA 2,ROW
21 04545 132414         SUB# 1,2,SZR            ;SKP IS DIAGONAL WORD
22 04546 000412         JMP BCKCK               ;BACKGRD WORD CHECK
23 04547 034446         LDA 3,BCKGN
24 04550 164000         COM 3,1
25 04551 106415 SD.KX:  SUB# 0,1,SNR            ;SKP IS WORD ERROR
26 04552 000411         JMP SD.CR
27 04553 054116         STA 3,PK.53
28 04554 030440         LDA 2,SD.AD             ;ADDRESS OF WORD IN ERROR
29                      LCALL ERROY
30 04556 003524         M4.T1
31 04557 000404         JMP SD.CR               ;CONTINUE
32                      ;WORD WAS BACKGRND CHECK IT
33 04560 034435 BCKCK:  LDA 3,BCKGN
34 04561 165000         MOV 3,1
35 04562 000767         JMP SD.KX               ;CHECK IT
36 04563 024427 SD.CR:  LDA 1,ROW
37 04564 034107         LDA 3,K37
38 04565 106415         SUB# 3,1,SNR            ;SKP IS NOT DONE ALL
39 04566 000404         JMP SD.CL               ;DO NEXT COLUMN
40 04567 125400         INC 1,1
41 04570 044422         STA 1,ROW
42 04571 000734         JMP SD.05+3             ;NEXT ROW TO CHECK
43                      ;INC TO NEXT COLUMN
44 04572 102400 SD.CL:  SUB 0,0
45 04573 040417         STA 0,ROW
46 04574 020417         LDA 0,CLMN
47 04575 162415         SUB# 3,0,SNR            ;SKP IS NOT DONE ALL COLUMNS
48 04576 000405         JMP SD.06
49 04577 101400         INC 0,0
50 04600 000724         JMP SD.05+2             ;DO NEXT COLUMN
51 04601 000000 SUCNT:  0
52 04602 177740 SD.32:  -32.
```

```
10070 MNSCM
01                      ;SLIDE DIAGONAL BY ONE POSITION
02 04603 020413 SD.06:  LDA 0,DIAG
03 04604 116415         SUB# 0,3,SNR            ;SKP IS NOT ALL POSITIONS DONE
04 04605 000412         JMP SD.07               ;DONE ALL NOW COMPLEMENT PATTERN
05 04606 101400         INC 0,0
06 04607 040407         STA 0,DIAG
07 04610 102400         SUB 0,0
08 04611 000654         JMP M4.02+3             ;DO THIS DIAG.
09
10 04612 000000 ROW: 0
11 04613 000000 CLMN: 0
12 04614 000000 SD.AD: 0
13 04615 000000 BCKGN: 0
14 04616 000000 DIAG: 0
15
16                      ;COMPLEMENT THE BACKGROUND PATTERN
17 04617 020776 SD.07:  LDA 0,BCKGN
18 04620 100000         COM 0,0
19 04621 040774         STA 0,BCKGN
20 04622 101005         MOV 0,0,SNR             ;NO SKP IS DONE ALL TESTING THIS ARRAY
21 04623 000404         JMP M4.EX               ;EXIT
22 04624 102400         SUB 0,0
23 04625 002401         JMP .+1
24 04626 004464         M4.02+2                 ;START OVER WITH BACKGND COMPL.
25                      M4.EX:  LCALL RETRN
```

```
;0071 MASM
01                      ;MEMORY TEST #5
                                .TITL GLPAT
                                NEXIT M5,1,GALPAT
03                      ;DEFINITION TO LINKR PARAMETERS FOLLOWS
04          004630      LMEML=.
05          000146              .LOC LPG0
06 000146   004633              M5.00
07          000147      LPG0=.
08          004630              .LOC LMEML
09 004630   000000              0                   ;TST ERR CTR THIS PASS
10 004631   000000              0                   ;TEST PASS COUNTER
11 004632   000000              0                   ;TEST ERROR COUNTER
12 004633   000410      M5.00:  JMP M5.02                   ;EXECUTE ENTRY ADDRESS
13 004634   000000              0                           ;WAIT SWITCH
14 004635   000001              1                           ;MAXIMUM SEGMENT SIZE
15 004636   040507              .TXTE ;GALPAT;
16          050314
17          152101
18          000000
19
20                      ;EXECUTE ENTRY
21 004642   020000              1B2
22 004643   102000      M5.02:  ADC 0,0             ;AC0=-1
23 004644   040444              STA 0,M5.PS         ;PLACE IN CNTL WORD
24 004645   024120              LDA 1,SWRG2         ;CHECK DATA HOLD FLG
25 004646   030774              LDA 2,M5.02-1
26 004647   147404              AND 2,1,SZR         ;SKP IS USE RANDOM DATA
27 004650   000402              JMP M5.2A           ;
28                              LCALL ARANG         ;GET RANDOM #
29                      M5.2A:  LCALL LWRBK         ;FILL BACKGROUND
```

```
;0072 MASM
01                      ;WRITE PATTERN COMPLEMENTED
02 004653   034137      M5.03:  LDA 3,SCRLO         ;GET STARTING ADDRESS
03 004654   040435              STA 0,M5.SV         ;SAVE BACKGND DATA WORD
04 004655   100000              COM 0,0             ;INVERT PATTERN
05 004656   041400              STA 0,0,3           ;PUT INVERTED PATTERN IN X
06 004657   034137              LDA 2,SCRLO         ;GET VALUE FOR Y
07
08                      ;READ X+1,X,X+2,X,X+3,ETC.
09 004660   150415      M5.04:  SUB# 2,3,SNR        ;DON'T USE X
10 004661   000410              JMP M5.4C           ;NEXT WORD
11 004662   021000              LDA 0,0,2           ;GET CONTENTS OF THIS LOCATION
12 004663   025400              LDA 1,0,3           ;GET X CONTENTS
13 004664   107415              AND# 0,1,SNR        ;CHECK AGAINST INVERTED PATTERN
14 004665   000404              JMP M5.4C           ;NO ERROR, SKIP
15 004666   054116              STA 3,PR,S3
16                              LCALL ERROY
17 004670   003477              M5.T1
18 004671   024140      M5.4C:  LDA 1,SCRMI         ;GET SCRMI
19 004672   146033              ADC# 2,1,SNC        ;DONE ALL?
20 004673   000403              JMP M5.05
21 004674   151403              INC 2,2             ;INC TO NEXT Y
22 004675   000763              JMP M5.04           ;NOT YET
23
24                      ;INCREMENT X
25 004676   020413      M5.05:  LDA 0,M5.SV         ;GET BACKGND
26 004677   041400              STA 0,0,3           ;PUT PATTERN IN X
27 004700   175400              INC 3,3             ;STEP TO NEXT WORD
28 004701   166432              SUBZ# 3,1,SZC       ;X> SCRMI?
29 004702   000753              JMP M5.04-3         ;USE NEW X VALUE
30
31 004703   014405      M5.06:  ISZ M5.PS           ;CHECK PASS#
32 004704   000403              JMP M5.EX           ;EXIT IF NOT FIRST PASS
33 004705   100000              COM 0,0             ;INVERT DATA PATTERN
34 004706   000744              JMP M5.2A           ;GO FILL BACKGND AGAIN,TEST AGAIN
35
36                      ;EXIT POINT OF TEST
37                      M5.EX:  LCALL RETRN         ;RETURN TO LINKR
38
39 004710   000000      M5.PS:  0                   ;PASS CNTL #
40 004711   000000      M5.SV:  0                   ;SAVE BACKGND DATA
```

```
0073 MNSCM
01
02                ;MEMORY TEST# 6....GALLOPING WRITE RECOVERY
                          .TITL GLPWR
04                        NEXTT M6,1,GALWREC
05                ;DEFINITION TO LINKR PARAMETERS FOLLOWS
06        004712 LMEML=.
07        000147          .LOC LPG0
08 00147  004715          M6.00
09        000150 LPG0=.
10        004712          .LOC LMEML
11 04712  000000          0                       ;TST ERR CTR THIS PASS
12 04713  000000          0                       ;TEST PASS COUNTER
13 04714  000000          0                       ;TEST ERROR COUNTER
14 04715  000410 M6.00:   JMP M6.02                        ;EXECUTE ENTRY ADDRESS
15 04716  000000          0                               ;WAIT SWITCH
16 04717  000001          1                               ;MAXIMUM SEGMENT SIZE
17 04720  040507          .TXTE |GALWREC|
18        153714
19        142722
20        000303
21
22
23                ;EXECUTE ENTRY
24 04724  020000          1B2
25 04725  102000 M6.02:   AUC 0,0
26 04726  040447          STA 0,M6.PS             ;SET PASS CNTL TO -1
27 04727  030120          LDA 2,SWRG2             ;GET DATA FLAG
28 04730  024774          LDA 1,M6.02-1
29 04731  133404          AND 1,2,SZR             ;SKP IS USE RANDOM DATA
30 04732  000402          JMP M6.2A               ;USE ALL ONE'S PATTERN
31                        LCALL ARANG             ;USE RANDOM DATA
32                M6.2A:  LCALL LWRBK             ;FILL BACKGROUND
33
34                ;SET X TO SCRLO
35 04735  030137          LDA 2,SCRLO             ;X ADDRESS
36 04736  034137 M6.03:   LDA 3,SCRLO             ;GET SCRLO
37 04737  104000          COM 0,1                 ;PUT PATTERN COMPL. IN AC1
38
39                ;WRITE/READ FOR THIS VALUE OF X
40 04740  156415 M6.04:   SUB# 2,3,SNR            ;DON'T CHECK X
41 04741  000410          JMP M6.4C               ;NEXT WORD
42 04742  045400          STA 1,0,3               ;STORE INV. PATTERN IN Y
43 04743  025000          LDA 1,0,2               ;READ X
44 04744  106435          SUBZ# 0,1,SNR           ;CHECK READ AGAINST BACKGND
45 04745  000404          JMP M6.4C               ;NO ERROR,SKIP
46 04746  054116          STA 3,PR.S3
47                        LCALL ERROY
48 04750  003556          M6.T1
49 04751  041400 M6.4C:   STA 0,0,3               ;WRITE TO Y BACKGND
50 04752  025000          LDA 1,0,2               ;READ X
51 04753  106435          SUBZ# 0,1,SNR           ;CHECK AGAINST BACKGND
52 04754  000404          JMP M6.4D               ;NO ERROR, SKIP
53 04755  054116          STA 3,PR.S3
54                        LCALL ERROY
55 04757  003623          M6.T2
56 04760  024140 M6.4D:   LDA 1,SCRHI
57 04761  156033          ADCZ# 3,1,SNC           ;CHECK IF Y> SCRHI
58 04762  000403          JMP M6.05
59 04763  175400          INC 3,3                 ;INC TO NEXT Y
60 04764  000753          JMP M6.04-1             ;CONTINUE WITH NEXT Y
```

```
0074 MNSCM
01
02                ;INCREMENT X TO NEXT WORD
03 04765  151400 M6.05:   INC 2,2
04 04766  024140          LDA 1,SCRHI
05 04767  146432          SUBZ# 2,1,SZC           ;DONE ALL?
06 04770  000746          JMP M6.03               ;NOT YET
07
08                ;END OF PASS=CHECK PASS#
09                ;M6.06: ISZ M6.PS
10 04771  000403          JMP M6.EX               ;EXIT IF NOT FIRST PASS
11 04772  100000          COM 0,0
12 04773  000741          JMP M6.2A               ;START AGAIN WITH COM DATA
13
14                M6.EX:  LCALL RETRN
15 04775  000000 M6.PS:   0
16
```

```
                        .TITL MNSCMT
02
03 04776 000007 PROGK:  LPG0-LZMAX      ;#OF PGMS LOADED TO BE INIT
04
05 04777 000000 LSYSE: 0                ;END OF PGM
06
07 05000 047503         .TXT    /COPYRIGHT (C) DGC,1976
08 05013 046101 ALL RIGHTS RESERVED/
09
10
11 05025 04/115 DIRT:   .TXTE ;MNSCMT   001
12       141523
13       152115
14       120240
15       03@240
16       000060
17 05033 000000         0
18 05034 000200         STRT1
19 05035 176774         176774
20 05036 000000         0
21 05037 000000         0
22 05040 000000         0
23 05041 000000         0
24 05042 000000         0
25
26                      .END    STRT2

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```