DataGeneral

# DIAGNOSTIC LISTING

LISTING

Ø96-ØØØ44Ø-ØØ

PROGRAM

MOS MEMORY DIAGNOSTIC

TAPE

Ø95-ØØØ44Ø-ØØ

ABSTRACT

THIS DIAGNOSTIC WILL CHECK OUT MEMORIES OF NOVA/ECLIPSE COMPUTERS DESIGNED AROUND 1 AND 4K MOS RAMS.

```
          ;*************************************************************
          ;
          ; NAME: MOSMEM.SR                        PART NUMBER: 094-000838
          ;
          ;
          ; DESCRIPTION: MOS MEMORY DIAGNOSTIC
          ;
          ;
          ; REVISION HISTORY:
          ;
          ;       REV.             DATE
          ;
          ;       00               12/03/76
          ;
          ;
          ; COPYRIGHT (C) DATA GENERAL CORPORATION, 1976
          ; ALL RIGHTS RESERVED.
          ;*************************************************************
```

```
                                .TITL   MOSMEM
          000000                .DO     0

                        1.      ABSTRACT


                                THIS DIAGNOSTIC WILL CHECKOUT MEMORIES OF NOVA/ECLIPSE
                                COMPUTERS DESIGNED ARROUND 1 AND 4K MOS RAMS.


                        2.      MACHINE REQUIREMENTS

                        2.1     ANY NOVA LINE OR ECLIPSE PROCESSOR
                        2.2     2K OF READ/WRITE MEMORY
                        2.3     TELETYPE OR CRT


                        3.      OPERATING PROCEDURE

                                THE DIAGNOSTIC CAN BE LOADED AND RUN IN A STAND-ALONE
                                MODE AS WELL AS UNDER DTOS/EDTOS/PDTOS.

                        3.1     STAND ALONE
                                ON LOADING THE PROGRAM VIA BINARY LOADER IT WILL AUTO
                                START AT LOCATION 200. DURING THE FIRST PASS OPERATOR
                                WILL BE ASKED TO SET THE "SWREG" & "USREG" OPTIONS. THE
                                OPTIONS CAN BE SELECTED BY TYPING A NUMBER FOLLOWED BY A
                                "RETURN" (THE RIGHT MOST 16 BITS OF THE NUMBER TYPED IN
                                WILL BE ACCEPTED AS THE ANSWER). A "RUBOUT" WILL CAUSE
                                THE QUESTION TO BE REPEATED. TYPING OF A "RETURN" ONLY,
                                WILL DEPOSIT A 0 IN THE OPEN OPTION LOCATION.
                                IF BIT 0 OF "USREG" IS SET THEN THE OPERATOR HAS TO
                                DEFINE THE MEMORY BOUNDRIES TO BE TESTED IN WHICH CASE
                                QUESTIONS REGARDING THE LOWEST MEMORY LOC. TO BE TESTED
                                "LOMEM", HIGHEST MEMORY LOC. TO BE TESTED "HIMEM", AND
                                MAXIMUM MEMORY AVAILABLE "MXMEM" WILL BE ASKED. ANSWERS
                                TO THESE QUESTIONS WILL BE OF THE SAME FORM AS FOR
                                "SWREG" AND "USREG".
                                IF BIT 0 OF "USREG" IS NOT SET THEN THE PROGRAM WILL
                                SELF SIZE THE MEMORY AND THE ENTIRE MEMORY WILL BE
                                TESTED.
                                IN EITHER OF THE CASES "X - Y IS MEM TO TEST" WILL BE
                                TYPED OUT, WHERE X = LOWEST LOCATION TO BE TESTED AND Y
                                HIGHEST LOCATION TO BE TESTED. CONTENETS OF MEMORY BETWE
                                X AND Y EXCEPT THE PROGRAM ITSELF AND 128 WORDS OF MINI-
                                MONITOR FOR DTOS WILL BE DESTROYED.


                        3.2     UNDER DTOS/EDTOS/PDTOS

                        3.2.1   RUN ALL
                                UNDER THIS MODE THE PROGRAM RUNS WITH THE "USREG" SET
                                TO A DEFAULT MODE OF 0.

                        3.2.2   PROGRAM LOAD
                                ONCE THE PROGRAM HAS BEEN LOADED THE SAME PROCEDURE AS
                                STAND ALONE MODE HAS TO BE FOLLOWED.
```

### 4. ERROR DESCRIPTION

THE PROGRAM IS DESIGNED TO ENCOUNTER TWO TYPES OF ERRORS

### 4,1 FATAL ERROR

THIS WILL BE AN ERROR CAUSED BY SOME THINGN OTHER THAN THE UNIT UNDER TEST, EXAMPLE OF THIS TYPE OF ERROR IS PROGRAM FLOW BEEING OUT SEQUENCE,
A FATAL ERROR OVER RIDES THE SWITCH SETTINGS, THE ERROR MESSAGE IS PRINTED ON BOTH THE TTY & LPT (IF AVAILABLE), & THE PROGRAM HALTS, THE PROGRAM SHOULD NOT BE CONTINUED AFTER A FATAL ERROR,

### 4,2 SOFT ERROR

THIS ERROR WILL BE REPORTED ON SEEING SOME PROBLEM WITH THE UNIT UNDER TEST, AFTER REPORTING THE ERROR THE PROGRAM WILL ACT ACCORDING TO THE SETTINGS OF THE "SWREG", THE PROGRAM CAN BE CONTINUED AFTER A HALT ON SOFT ERROR,

### 4,2,1 SOFT ERROR FORMAT

ON SEEING A BIT FAILING FOR THE FIRST TIME AN ERROR WITH THE FOLLOWING FORMAT WILL BE REPORTED:

```
BDATA - GDATA - LOC - TSTNM - PC

WHERE   BDATA   IS THE BAD OR FOUND DATA
        GDATA   IS THE GOOD OR EXPECTED DATA
        LOC     IS THE ADDRESS OF THE FAILING LOCATION
        TSTNM   IS THE TEST NUMBER
        PC      IS THE PC WHERE THE ERROR WAS DETECTED
```

### 4,3 ERROR HISTORY

ON COMPLETION OF A PASS ERROR HISTORY OF THE FAILING BIT BE REPORTED UNLESS "SWREG" IS SET TO SUPPRESS THE END OF PASS TYPE OUT, IN WHICH CASE THE INFORMATION WILL BE COLLECTED UNTILL IT IS OBTAINED BY HITTING ANY TTY KEY, ERROR HISTORY WILL BE FOR THE TIME SINCE IT WAS REPORTED LAST,

### 4,3,1 ERROR HISTORY FORMAT

THE ERROR HISTORY WILL BE REPORTED IN THE FOLLOWING FORM

```
BITNM - QUAD - BANK - ERTMS

WHERE   BITNM   IS THE FAILING BIT NUMBER (0-15)
        QUAD    IS THE 1K QUADRANT OF THE MEMORY BANK
                EACH BANK IS DEVIDED INTO 4 QUADRANTS,
                LOCATION 0 WILL BE REPORTED IN Q0 OF B0
        BANK    IS THE BANK NUMBER
                THE MEMORY IS DEVIDED INTO 4K BANKS
                BANK 0 IS 0-7777, BANK 1 IS 10000-17777
                SO ON,
        ERTMS   IS THE NUMBER OF TIMES THE BIT FAILED
                REPORTED IN DECIMAL,
                HIGHEST NUMBER REPORTED WILL BE 32767,
```

### 5. SWITCH SETTINGS

LOCATION "SWREG" IS USED TO SELECT THE PROGRAM OPTIONS (NOT SYSTEM CONFIGURATION), WHILE RUNNING UNDER DTOS/ PDTOS/EDTOS THIS LOCATION WILL BE LOADED BY THE MONITOR HOWEVER UNDER STAND ALONE AND PROGRAM LOAD MODES THIS LOCATION WILL BE SET ACCORDING TO THE ANSWER SUPPLIED BY THE OPERATOR,

### 5,1 SWITCH OPTIONS

DIFFERENT BITS AND THEIR INTERPRETATION AT LOCATION "SWREG" IS AS FOLLOWS:

| BIT | OCTAL VALUE | BINARY VALUE | INERPRETATION |
|---|---|---|---|
| 1 | | 0 | LOOP ON ERROR |
| | 40000 | 1 | SKIP LOOPING ON ERROR |
| 2 | | 0 | PRINT TO CONSOLE |
| | 20000 | 1 | ABORT PRINT OUT TO CONSOLE |
| 3 | | 0 | PRINT DETAILED ERROR ON THE SELECTED DEVICE/DEVICES |
| | 10000 | 1 | ABORT ERROR PRINT OUTS |
| 4 | | 0 | ALLOW END OF PASS PRINT OUT |
| | 04000 | 1 | SUPPRESS END OF PASS PRINT OUT |
| 5 | | 0 | DO NOT PRINT ON THE LINE PRINTER |
| | 02000 | 1 | PRINT ON THE LINE PRINTER |
| 6 | | 0 | DO NOT HALT ON ERROR |
| | 01000 | 1 | HALT ON ERROR |

### 5.2 USER'S OPTIONS

LOCATION "USREG" IS USED TO DEFINE ONE OF THE FOLLOWING OPTIONS:

| BIT(S) | OCTAL VALUE | BINARY VALUE | INTERPRETATION |
|---|---|---|---|
| 0 | | 0 | CHECK THE ENTIRE MEMORY |
| | 100000 | 1 | MEMORY BOUNDRIES WILL BE DEFINED BY THE OPERATOR, |
| 1 | | 0 | MEMORY BOARD IS DESIGNED ARROUND 1K RAMS OR DG'S 4K RAMS |
| | 040000 | 1 | MEMORY BOARD IS DESIGNED ARROUND OTHER 4K RAMS, |
| 2 | | 0 | ALLOW THE RELOCATION OF THE PROG TO THE HIGHEST AVAILABLE MEMORY, |
| | 020000 | 1 | SUPPRESS RELOCATION |
| 3 | | 0 | ONLY ROW/COLUMN GALLOPING TEST IS TO BE PERFORMED, |
| | 010000 | 1 | ENABLE LONG GALLOPING TEST |
| 11 | | 0 | DO NOT LOOP ON A TEST |
| | 000020 | 1 | LOOP ON THE TEST SPECIFIED BY BITS 12-15 |
| 12-15 | | | TEST NUMBER,<br>ONLY TEST DEFINED BY BITS 12-15 THROUGH THE LAST TEST WILL BE EXECUTED, |

```
            6.      TTY COMMANDS

                    ANY KEY WILL CAUSE THE ERROR HISTORY TO BE TYPED OUT.
                            THE PROGRAM WILL CONTINUE RUNNING.
                    ^D      THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE
                            PROGRAM AND THE SWITCH OPTIONS WILL BE ASKED.
                    ^C      THIS COMMAND GIVEN AT ANY TIME WILL CAUSE THE
                            PROGRAM TO GET RELOCATED BACK TO ITS ORIGINAL
                            CORE AND HALT.
                            A CONTINUE AFTER ^C WILL START THE PROGRAM WITH
                            THE PREVIOUS SETTINGS OF THE OPTINS.
                            THE PROGRAM CAN BE STARTED AT 200 TO RESET THE
                            OPTIONS.


            7.      PROGRAM DESCRIPTION

                    THE DIAGNOSTIC IS DESIGNED TO CHECK OUT 1 AND 4K MOS
                    RAMS AND THE CIRCUITRY ARROUND THEM. 4K RAMS MANUFACTUR-
                    ED BY DATA GENERAL ARE TREATED AS 4 1K RAMS.
                    THE TEST STARTS WITH THE CHECKING OF SIMPLE HARDWARE AND
                    AS IT PROGRASSES MORE AND MORE COMPLEX TESTING IS DONE.
                    IF THE RELOCATION IS ALLOWED THEN THE PROGRAM CAN NOT BE
                    STOPPED IN THE MIDDLE AND RESTARTED AT 200 UNLESS A TTY
                    COMMAND IS USED TO STOP THE PROGRAM.
                    A TEST CAN NOT BE STARTED IN THE MIDDLE OF THE PROGRAM.
                    INORDER TO GO TO A SPECIFIC TEST PROPER VALUE SHOULD BE
                    STORED AT "USREG".

            7.1     END OF PASS
                    ON COMPLETION OF A PASS WORDS "# PASS" WILL BE
                    PRINTED. (# IS THE PASS NUMBER OF THE DIAGNOSTIC IN
                    DECIMAL).
                    .ENDC
```

```
      000001          .NOCON  1
      000000          .NOLOC  0
      000000          .LOC    0
00000 001772          DIRT

      000002          .LOC    2

                      ;*OCTAL ACCEPT ROUTINE
                      ;*--------------------
                      ;*
                      ;*THIS ROUTINE IS USED TO ACCEPT OCTAL INPUTS FROM
                      ;*THE OPERATOR.
                      ;*ALL OF THE ACCUMULATORS ARE DESTROYYED. AC3 RETURNS
                      ;*WITH A 0 AFTER A "CR".
                      ;*

00002 001744 IHSTRY:: HSTRY+1
00003 000502 IRESTR:: RESTR
00004 000000 RTINO:   0

00005 054004 TINO:    STA     3,RTINO          ;SAVE THE RETURN ADDRESS
00006 006371          JSR     @IMESS
00007 136640          .TXTE   / = /
      000240
00011 152400          SUB     2,2              ;AC2 WILL RECEIVE THE INPUT
00012 063610 OPRIN:   SKPDN   TTI
00013 000012          JMP     .-1              ;WAIT FOR THE INPUT
00014 060610          DIAC    0,TTI            ;READ
00015 034166          LDA     3,N177           ;AC3 = 177
00016 163400          AND     3,0              ;STRIP THE PARITY BIT
00017 006167          JSR     @IPRINT          ;ECHO THE RECEIVED CHARACTER
00020 034165          LDA     3,N15            ;AC3 = 15
00021 116405          SUB     0,3,SNR          ;SKIP IF NOT A RETURN
00022 002004          JMP     @RTINO
00023 034004          LDA     3,RTINO          ;AC3 = CALLING LOCATION +1
00024 024047          LDA     1,N60            ;AC1 = 60
00025 106000          ADC     0,1
00026 020115          LDA     0,M77767         ;AC0 = 177767
00027 106052          ADCOM   0,1,SZC          ;IF AN OCTAL NUMBER WAS TYPED
                                               ;THEN AC1 SHOULD BE BETWEEN
                                               ;177770 AND 177777
00030 001772          JMP     -6,3             ;IN CASE OF ILLEGAL CHARAC-
                                               ;TER RETURN TO ASK THE QUEST-
                                               ;ION AGAIN
00031 151120          MOVZL   2,2
00032 151120          MOVZL   2,2
00033 151120          MOVZL   2,2
00034 132000          ADC     1,2              ;LOAD AC2
00035 000012          JMP     OPRIN            ;GETT THE NEXT CHARACTER

00036 051437 TOCAT:   STA     2,CC,3           ;STORE AC2 IN "HIGHM"
                                               ;CC = HIGHM-(CALCAT+1)
00037 051442          STA     2,CA,3           ;DEFINE THE NEW MAX. MEM. FOR
                                               ;PROGRAM RELOCATION USE
                                               ;CA = MXMEM-(CALCAT+1)
00040 025435          LDA     1,CB,3           ;LOAD AC1 WITH "TEMPA"
                                               ;CB = TEMPA-(CALCAT+1)
00041 045441          STA     1,CD,3           ;STORE AT "ENDPRG"
                                               ;CD = ENDPRG-(CALCAT+1)
00042 024045          LDA     1,IEGGS          ;AC1 = 375
```

```
 0007 MOSME
01 00043 133000          ADD     1,2
02 00044 001004          JMP     4,2                ;START THE CAT
```

```
10008 MUSME
01        000045         .LOC    45
02 00045 000375 IEGGS:   EGGS
03 00046 000000 BTCNT:   0
04 00047 000060 N60:     60
05
06        000050         .LOC    50
07 00050 000010          .BLK    8.                 ;RESERVED FOR DEBUGGER'S USE
08
09
10                       ;*MEMORY SIZING ROUTINE
11                       ;*
12
13 00060 054261 SIZE:    STA     3,LOMEM            ;"LOMEM" = 0
14 00061 030002          LDA     2,IHSTRY           ;AC2=END OF THE PROGRAM
15 00062 151400          INC     2,2
16 00063 025001          LDA     1,1,2              ;SAVE
17 00064 134000          COM     1,3
18 00065 055001          STA     3,1,2              ;WRITE
19 00066 021001          LDA     0,1,2              ;READ THE LOCATION WRITTEN INTO
20 00067 045001          STA     1,1,2              ;RESTORE
21 00070 050262          STA     2,HIMEM
22 00071 151513          INCL#   2,2,SNC            ;SKIP AFTER REACHING 32K BOUNDRY
23 00072 116414          SUB#    0,3,SZR            ;SKIP IF SIZING IS NOT OVER
24 00073 000253          JMP     MAXSV
25 00074 000062          JMP     SIZE+2
```

```
                                ;*ERROR SUMMARY ROUTINE
                                ;*----------------------
                                ;*
                                ;*THIS ROUTINE IS USED TO PRINT ERROR SUMMARY EITHER
                                ;*AFTER COMPLETION OF A PASS OR WHEN REQUESTED BY THE
                                ;*OPERATOR THROUGH A TTY KEY.
                                ;*THIS CODE CAN ONLY BE EXECUTED WHEN THE PROGRAM
                                ;*IS RESIDING IN THE LOWER CORE.
                                ;*


00075 152400 ERSUM:  SUB     2,2
00076 050113         STA     2,BANK
00077 102400 ERSM0:  SUB     0,0
00100 050112         STA     2,QUAD            ;"QUAD" = QUADRANT NUMBER
00101 040111         STA     0,BITNM
00102 020374         LDA     0,N20
00103 040046         STA     0,BTCNT
00104 032004 ERSM1:  LDA     2,@RTINO          ;AC2=ERROR COUNT FOR ONE BIT
00105 151005         MOV     2,2,SNR           ;SKIP IF THIS BIT EVER FAILED
00106 000117         JMP     ERTMS+3
00107 050114         STA     2,ERTMS
00110 006174         JSR     @IPDEC            ;PRINT THE HISTORY OF FAILURE
                                               ;AND RETURN ,+5
00111 000000 BITNM:  0                         ;THIS LOCATION WILL HAVE THE
                                               ;FAILING BIT NUMBER
00112 000000 QUAD:   0                         ;QUADRANT OF THE FAILIN BANK
                                               ;(A QUADRANT=1K)
00113 000000 BANK:   0                         ;THIS LOCATION WILL HAVE THE
                                               ;FAILING BANK NUMBER (A BANK=4K)
00114 000000 ERTMS:  0                         ;NUMBER OF TIMES THIS BIT FAILED
00115 177767 M77767: ANDCS   3,3,SBN           ;BIT 0 OF THIS LOC. SHOULD BE
                                               ;1. M77767 = 177767
                                               ;NEXT LOCATION WILL NEVER BE
                                               ;SKIPPED
00116 052004         STA     2,@RTINO          ;CLEAR ERROR COUNTER
00117 010004         ISZ     RTINO             ;CHECK THE NEXT BIT
00120 010111         ISZ     BITNM
00121 014046         DSZ     BTCNT
00122 000104         JMP     ERSM1
00123 024112         LDA     1,QUAD
00124 125237         MOVZR#  1,1,SBN           ;SKIP IF QUADRANT 3 HAS BEEN
                                               ;CHECKED
00125 131401         INC     1,2,SKP
00126 010113         ISZ     BANK
00127 034113         LDA     3,BANK
00130 177120         ADDZL   3,3
00131 157300         ADDS    2,3
00132 177120         ADDZL   3,3               ;AC3 = LOWEST ADDRESS IN THE
                                               ;NEXT MEMORY QUAD.
00133 024262         LDA     1,HIMEM           ;AC1=TOP OF THE MEMORY
00134 136052         ADCO#   1,3,SZC           ;SKIP IF ERROR STACK FOR MEM.
                                               ;UNDER TEST HAS BEEN CHECKED
00135 000077         JMP     ERSM0

00136 006174 EOP:    JSR     @IPDEC            ;PRINT THE PASS NUMBER
                                               ;AND RETURN WITH AC2 = 0
00137 000000 INPAS:  0
00140 126520         SUBZL   1,1               ;AC1 = 1
00141 006371         JSR     @IMESS            ;PRINT END OF PASS
```

```
                                               ;RETURN WITH AC0 & AC2 = 0
00142 050240         .TXTE   / PASS  /
      051501
      120123
      000240
00146 034045 TERM:   LDA     3,IEGGS           ;UNDER DTOS ?
00147 021400         LDA     0,0,3
00150 063710         SKPDZ   TTI
00151 070610         DIAC    2,TTI             ;READ THE TTI KEY
00152 101004         MOV     0,0,SZR           ;SKIP IF NOT
00153 000352         JMP     RTRN-2
00154 034166         LDA     3,N177
00155 173420         ANDZ    3,2               ;GET RID OF THE PARITY BIT AND
                                               ;SET THE CARRY BIT TO 0
00156 127120         ADDZL   1,1               ;AC1 = 4
00157 146654         SUBOR#  2,1,SZR           ;SKIP IF IT IS AC OR AD
00160 000340         JMP     CUNT
00161 146415         SUB#    2,1,SNR           ;SKIP IF IT IS AC
00162 000200         JMP     START
00163 063077         HALT
00164 000260         JMP     SIZED

00165 000015 N15:    15
00166 000177 N177:   177
00167 001712 IPRINT: PRINT
```

```
      000170          .LOC    170


                      ;*STARTING OF THE PROGRAM
                      ;*

00170 102441 OFF:     SUBO    0,0,SKP         ;SET TO 0
00171 102000 ON:      ADC     0,0             ;SET TO -1
00172 040377          STA     0,CATSW
00173 000200          JMP     START

00174 001636 IPDEC:   PDEC
00175 001642 IPOCT:   POCT

      000200          .LOC    200

00200 060277 START:   INTDS
00201 152400          SUB     2,2
00202 155520          INCZL   2,3             ;AC3 = 2
00203 020375          LDA     0,EGGS          ;UNDER DTOS ?
00204 101004          MOV     0,0,SZR         ;SKIP IF NOT
00205 000224          JMP     UNDTS
00206 040574          STA     0,SWREG
00207 006371          JSR     @IMESS          ;ASK FOR THE OPTIONS
00210 005215          .TXTE   /<15><12>SWREG/
      153523
      142722
      000107
00214 004005          JSR     TINO
00215 050137          STA     2,INPAS         ;TEMPORARILY SAVE THE "SWREG"
00216 006371          JSR     @IMESS          ;ASK FOR THE OTHER OPTIONS
00217 005215          .TXTE   /<15><12>USREG/
      051525
      142722
      000107
00223 004005          JSR     TINO
00224 050557 UNDTS:   STA     2,USREG
00225 151113          MOVL#   2,2,SNC         ;AC2 = USREG
                                              ;SKIP IF THE MEMORY BOUNDRIES
                                              ;ARE TO BE DEFINED
00226 000060          JMP     SIZE
00227 006371 OTHIN:   JSR     @IMESS          ;ASK FOR THE LOWEST MEMORY TO BE
                                              ;TESTED
00230 005215          .TXTE   /<15><12>LOMEM/
      147714
      142515
      000115
00234 004005          JSR     TINO
00235 050261          STA     2,LOMEM
00236 006371          JSR     @IMESS          ;ASK FOR THE HIGHEST MEMORY TO B
                                              ;TESTED
00237 005215          .TXTE   /<15><12>HIMEM/
      144510
      142515
      000115
00243 004005          JSR     TINO
00244 050262          STA     2,HIMEM
00245 006371          JSR     @IMESS          ;ASK FOR THE MAXIMUM AVAILABLE
                                              ;MEMORY
00246 005215          .TXTE   /<15><12>MXMEM/
```

```
      154115
      142515
      000115
00252 004005          JSR     TINO
00253 050536 MAXSV:   STA     2,MXMEM         ;SAVE THE HIGHEST AVAILABLE
                                              ;MEMORY ADDRESS
00254 030137          LDA     2,INPAS         ;READ THE SWREG OPTIONS
00255 034375          LDA     3,EGGS
00256 175005          MOV     3,3,SNR         ;SKIP IF DTOS HAS SET THE
                                              ;"SWREG"
00257 050523          STA     2,SWREG
00260 006175 SIZED:   JSR     @IPOCT          ;PRINT AND RETURN .+3
00261 000000 LOMEM:   0                       ;LOWEST LOCATION TO BE TESTED
00262 007777 HIMEM:   7777                    ;HIGHEST LOCATION TO BE TESTED
00263 177400 C377:    AND     3,3             ;BIT 0 OF THIS LOCATION SHOULD
                                              ;BE 1. (C377 = 177400)
00264 006371          JSR     @IMESS
00265 144640          .TXTE   / IS MEM TO TEST/
      120123
      142515
      120115
      147724
      152240
      051705
      000324
00275 040137          STA     0,INPAS         ;MAKE THE PASS COUNTER = 0
00276 034002          LDA     3,IHSTRY        ;AC3 = ADDRESS OF THE POINTER
                                              ;TO THE ERROR HISTORY LOC.
00277 030261          LDA     2,LOMEM
00300 050505          STA     2,LOWM          ;MAKE A COPY OF LOW MEM.
00301 030262          LDA     2,HIMEM
00302 050504          STA     2,HIGHM
00303 024166          LDA     1,N177          ;AC1 = 177
00304 151300          MOVS    2,2
00305 133620          ANDZR   1,2
00306 150240          COMOR   2,2             ;AC2 = TOTAL NUMBER OF 1K QUAD.
00307 024374 CLEAN:   LDA     1,N20           ;AC1 = 20
00310 041400          STA     0,0,3           ;CLEAR THE ERROR HISTORY LOC.
00311 175400          INC     3,3
00312 106004          ADC     0,1,SZR         ;DECREMENT AC1
00313 000310          JMP     .-3
00314 151404          INC     2,2,SZR
00315 000307          JMP     CLEAN
00316 054466          STA     3,TEMPA         ;SAVE END OF THE ERROR HISTORY
                                              ;COLLECTING LOCATIONS
00317 030472          LDA     2,MXMEM         ;AC2 = MAXIMUM MEMORY
00320 024375          LDA     1,EGGS
00321 125004          MOV     1,1,SZR         ;SKIP IF NOT UNDER DTOS
00322 024166          LDA     1,N177          ;AC1 = 177
00323 020377          LDA     0,CATSW
00324 101004          MOV     0,0,SZR         ;SKIP IF CAT IS OFF
00325 024372          LDA     1,N1777         ;AC1 = 1777
00326 132400          SUB     1,2
00327 124001          COM     1,1,SKP
00330 151400          INC     2,2
00331 021000          LDA     0,0,2           ;SAVE THE CAT PROG./MINI MONITOR
00332 041400          STA     0,0,3           ;ALONG WITH THE MAIN PROG.
00333 175400          INC     3,3
00334 125404          INC     1,1,SZR         ;SKIP AFTER SAVING ALL
00335 000330          JMP     .-5
```

```
00336 054452          STA     3,ENDPRG        ;"ENDPRG" HAS THE FIRST LOCATION
                                              ;THAT CAN BE TESTED
00337 000347          JMP     CONT1

00340 030451 CONT:    LDA     2,MXMEM
00341 034372          LDA     3,N1777
00342 172000          ADC     3,2
00343 020377          LDA     0,CATSW
00344 034137          LDA     3,INPAS
00345 163155          ADDOL#  3,0,SNR         ;SKIP IF CAT IS OFF OR IT IS
                                              ;NOT PASS 1
00346 004036 CALCAT:  JSR     TOCAT
00347 102400 CONT1:   SUB     0,0
00350 040442          STA     0,RELTG
00351 002003          JMP     @IRESTR


00352 015403          DSZ     3,3
00353 000340          JMP     CONT            ;NOT DONE YET
00354 021403 RTRN:    LDA     0,3,3           ;AC0 = PASS COUNT
00355 031404          LDA     2,4,3           ;GET THE RETURN ADDRESS
00356 024434          LDA     1,RELTG-.,1     ;AC1 = -1 IF THE PROGRAM IS
                                              ;RESIDING AT THE TOP OF THE
                                              ;MEMORY AND 0 OTHERWISE
00357 125414          INC#    1,1,SZR
00360 024703          LDA     1,C377-.,1      ;AC1 = -377
00361 041376 MMRSTR:  STA     0,-2,2          ;SAVE PASS COUNT FOR DTOS
00362 125405          INC     1,1,SNR         ;SKIP UNTILL TOP 376 WORDS ARE
                                              ;RESTORED
00363 001000          JMP     0,2             ;BACK TO DTOS
00364 014424          DSZ     ENDPRG
00365 036423          LDA     3,@ENDPRG       ;RESTORE MINI-MONITOR
00366 056423          STA     3,@MXMEM
00367 014422          DSZ     MXMEM
00370 000771          JMP     MMRSTR-.,1

00371 001611 IMESS:   MESS
00372 001777 N1777:   1777
00373 007777 N7777:   7777
00374 000020 N20:     20
```

```

00375 000000 EGGS:   0
00376 000000         0
00377 000000 CATSW:  0
      000400 PASS=   .
00400 000001                 .BLK    1
      000401 RETURN= .
00401 000001                 .BLK    1
      000402 SWREG=  .
00402 000001                 .BLK    1
      000403 USREG=  .
00403 000001                 .BLK    1
      000404 TEMPA=  .
00404 000001                 .BLK    1
      000405 LOWM=   .
00405 000001                 .BLK    1
      000406 HIGHM=  .
00406 000001                 .BLK    1
      000407 BKPAT=  .
00407 000001                 .BLK    1
      000410 ENDPRG= .
00410 000001                 .BLK    1
      000411 MXMEM=  .
00411 000001                 .BLK    1
      000412 RELTG=  .
00412 000001                 .BLK    1
      000437 TSTAD=  437
      000440            .LOC     440

00440 000017 N17:       17

                        ;*RELOCATE ROUTINE
                        ;*----------------
                        ;*
                        ;*THIS ROUTINE IS USED TO RELOCATE THE PROGRAM FROM
                        ;*LOW TO HIGH, AND HIGH TO LOW CORE. IF THE PROGRAM
                        ;*IS RELOCATED FROM HIGH TO LOW CORE THEN ERROR
                        ;*SUMMARY AND END OF PASS IS REPORTED.

00441 020742 RELCT:     LDA      0,USREG-.,1      ;SEE IF RELOCATION IS ALLOWED
00442 103100            ADDL     0,0
00443 176000            ADC      3,3
00444 101102            MOVL     0,0,SZC          ;SKIP IF SO
00445 000420            JMP      ENDPAS-.,1
00446 034742            LDA      3,ENDPRG-.,1
00447 126000            ADC      1,1              ;AC1 = -1
00450 137000            ADD      1,3              ;AC3 = NUMBER OF LOCATIONS
                                                  ;OCCUPIED BY THE PROGRAM
00451 171000            MOV      3,2
00452 020737            LDA      0,MXMEM-.,1      ;AC0 = HIGHEST AVAILABLE
                                                  ;MEMORY
00453 010737            ISZ      RELTG-.,1        ;CHECK IF THE PROGRAM IS GOING
                                                  ;TO BE RELOCATED TO THE HIGHER
                                                  ;CORE OR TO THE LOWER.
00454 115001            MOV      0,3,SKP
00455 111001            MOV      0,2,SKP
00456 044734            STA      1,RELTG-.,1      ;"RELTG" IS -1 IF THE PROGRAM
                                                  ;IS GOING TO BE EXECUTED FROM
                                                  ;THE HIGHER CORE, 0 OTHERWISE.
00457 021000            LDA      0,0,2            ;READ
00460 041400            STA      0,0,3            ;WRITE
00461 133043            ADDO     1,2,SNC          ;DECREMENT
00462 137042            ADDO     1,3,SZC          ;SKIP IF THE END IS NOT REACHED
00463 000402            JMP      .+2-.,1
00464 000773            JMP      .-5-.,1

00465 030002 ENDPAS:    LDA      2,IHSTRY
00466 050004            STA      2,RTINO          ;PREPARE FOR ERROR SUMMARY PRINT
                                                  ;OUT
00467 063610            SKPDN    TTI
00470 010137            ISZ      INPAS            ;INCREMENT THE ABSOLUTE LOC.
                                                  ;"INPAS", IF THE PROGRAM HAS
                                                  ;BEEN RELOCATED TO THE HIGH
                                                  ;CORE THEN THIS INC. WILL BE
                                                  ;IGNORED.
00471 020711            LDA      0,SWREG-.,1
00472 024540            LDA      1,N4000-.,1      ;AC1 = 4000
00473 123405            AND      1,0,SNR          ;SKIP IF END OF PASS TYPE OUTS
                                                  ;ARE SUPPRESSED (BIT 6)
00474 175414            INC#     3,3,SZR          ;A PASS IS COMPLETE AFTER RES-
                                                  ;TORING THE PROGRAM BACK TO
                                                  ;THE LOWER CORE
00475 063710            SKPDZ    TTI              ;END OF PASS WILL ALSO BE
                                                  ;TYPED IF A KEY HAS BEEN HIT
00476 000075            JMP      ERSUM            ;GO TO THE ABSOLUTE ADDRESS
                                                  ;OF ERROR SUMMARY ROUTINE
00477 031403            LDA      2,X,3            ;AC2 = RESTR-0
```

```
                                                  ;X = IRESTR-0
00500 173000            ADD      3,2              ;AC2 = RELOCATED ADDRESS OF "RES
00501 001000            JMP      0,2

00502 030671 RESTR:     LDA      2,N7777-.,1      ;AC2 = 7777
00503 172452            SUBO#    3,2,SZC
00504 155000            MOV      2,3
00505 030701            LDA      2,HIGHM-.,1      ;AC2 = HIGHEST MEM. TO TEST
00506 172452            SUBO#    3,2,SZC          ;SKIP IF AC2 IS MORE THAN
                                                  ;AC3
00507 155000            MOV      2,3
00510 024700            LDA      1,ENDPRG-.,1     ;AC1 = 1ST LOC. THAT CAN BE
                                                  ;TESTED
00511 020701            LDA      0,RELTG-.,1      ;AC0 WILL DETERMINE IF THE
                                                  ;PROGRAM IS IN A RELOCATED
                                                  ;STATE OR NOT
00512 101405            INC      0,0,SNR          ;SKIP IF NOT RELOCATED
00513 171001            MOV      3,2,SKP
00514 121001            MOV      1,0,SKP
00515 024670            LDA      1,LOWM-.,1       ;AC1 = LOWEST MEM. TO BE
                                                  ;TESTED
00516 132052            ADCO#    1,2,SZC          ;SKIP IF AC2 IS MORE THAN
                                                  ;AC1
00517 000722            JMP      RELCT-.,1
00520 024665            LDA      1,LOWM-.,1
00521 106452            SUBO#    0,1,SZC          ;SKIP IF AC1 IS EQUAL OR MORE
                                                  ;THAN AC0
00522 105000            MOV      0,1
00523 044575            STA      1,BTMEM-.,1
00524 050575            STA      2,TPMEM-.,1      ;TOP OF THE MEM. THAT WILL
                                                  ;BE TESTED WHILE THE PROG.
                                                  ;IS IN THIS STATE
00525 030713            LDA      2,N17-.,1        ;AC2 = 17
00526 024655            LDA      1,USREG-.,1      ;READ THE USER'S OPTIONS
00527 000420            JMP      SEND-.,1

00530 024500 NEXT:      LDA      1,DTSTNM-.,1     ;AC1 = TSTNM-NXTST
00531 137000            ADD      1,3              ;AC3 IS POINTING TO THE TEST
                                                  ;NUMBER
00532 025400            LDA      1,0,3            ;AC1 = TEST NUMBER THAT WAS JUST
                                                  ;EXECUTED
00533 132415            SUB#     1,2,SNR          ;CHECK FOR PROPER TEST SEQUENCE
00534 000410            JMP      NXT1-.,1
00535 005412            JSR      @8,3             ;GO TO THE MESSAGE TYPE-OUT
                                                  ;ROUTINE
00536 005215            .TXTE    /<15><12>SEQ ERR/
      142523
      120321
      151305
      000322
00543 063077            HALT
00544 020630 NXT1:      LDA      0,N20-.,1        ;AC0 = 20
00545 034636            LDA      3,USREG-.,1
00546 163404            AND      3,0,SZR          ;SKIP IF LOOPING ON TEST IS NOT
                                                  ;REQUIRED
00547 133405 SEND:      AND      1,2,SNR
00550 151400            INC      2,2              ;TEST 0 IS SAME AS TEST 1
00551 102401            SUB      0,0,SKP
00552 010635 RIT1S:     ISZ      BKPAT-.,1        ;LOOK AT THE PREVIOUS BACK
                                                  ;GROUND WORD AND SKIP IF IT
```

```
0017 MOSME
                                                    ;WAS =1.
00553 145001          MOV     2,1,SKP               ;AC1 = TEST TO BE EXECUTED
00554 000504          JMP     NXTST-.,1
00555 063710 LOOP:    SKPDZ   TTI                   ;CHECK FOR THE OPR. INPUT
00556 000663          JMP     RELCT-.,1
00557 030542          LDA     2,TPMEM-.,1
00560 034540          LDA     3,BTMEM-.,1
00561 041400          STA     0,0,3                 ;WRITE THE MEMORY UNDER TEST
                                                    ;TO A BACK GROUND OF 0
00562 175400          INC     3,3
00563 156052          ADCO#   2,3,SZC               ;SKIP AFTER REACHIING THE END
00564 000775          JMP     .-3-.,1
00565 004446          JSR     TSTAB-.,1             ;GET THE ADDRESS OF THE TEST
                                                    ;TABLE
00566 030441          LDA     2,DPNTR-.,1           ;AC2 = DIFFERENCE BETWEEN THE
                                                    ;STARTING OF THE TEST TABLE AND
                                                    ;"PNTR"
00567 173000          ADD     3,2                   ;AC2 IS POINTING TO "PNTR"
00570 040617          STA     0,BKPAT-.,1           ;SAVE THE BACK GROUND WORD
00571 041313          STA     0,Y,2                 ;SAVE THE BACK GROUND WORD AT
                                                    ;"BKGND", Y = BKGND-PNTR
00572 045176          STA     1,Z,2                 ;SAVE THE TEST NUMBER AT "TSTNM"
                                                    ;Z = TSTNM-PNTR
00573 102521          SUBZL   0,0,SKP               ;AC0 = 1
00574 175400          INC     3,3
00575 106404          SUB     0,1,SZR               ;SKIP IF AC3 IS POINTING TO
                                                    ;THE "ADDRESS TABLE" +1 OF THE
                                                    ;TEST TO BE EXECUTED
00576 000776          JMP     .-2-.,1
00577 020603          LDA     0,SWREG-.,1           ;READ THE SWITCHES
00600 041166          STA     0,XX,2                ;STORE THE SWITCHES AT LOC.
                                                    ;"CPSWR", XX = CPSWR-PNTR
00601 020602          LDA     0,USREG-.,1           ;READ THE MEMORY TYPE
00602 041250          STA     0,WW,2                ;STORE USER'S OPTIONS AT
                                                    ;"CPUSR", WW=CPUSR-PNTR
00603 025400          LDA     1,0,3
00604 137000          ADD     1,3
00605 024424          LDA     1,N37-.,1             ;AC1=37
00606 054631          STA     3,TSTAD-.,1           ;SAVE THE STARTING ADD. OF THE
                                                    ;TEST
00607 103112          ADDL#   0,0,SZC               ;SKIP FOR DG MEMORY CHIPS
00610 125140          MOVOL   1,1                   ;AC1=77
00611 045201          STA     1,W,2                 ;W = ENDCLM - PNTR
                                                    ;"ENDCLM" HAS THE END OF
                                                    ;COLUMN ADDRESS
00612 120000          COM     1,0
00613 135401          INC     1,3,SKP
00614 167000          ADD     3,1
00615 101404          INC     0,0,SZR
00616 000776          JMP     .-2-.,1
00617 045204          STA     1,ZZ,2                ;ZZ = ENDBNK -PNTR
                                                    ;"ENDBNK" HAS THE ADDRESS OF THE
                                                    ;LAST LOCATION IN THE BANK
00620 034501          LDA     3,TPMEM-.,1           ;AC3 = TOP OF THE MEMORY UNDER
                                                    ;TEST
00621 055351          STA     3,YB,2                ;SAVE AT "TPM1"
                                                    ;YB = TPM1-PNTR
00622 055202          STA     3,YY,2                ;SAVE AC3 AT "STRA3"
                                                    ;YY = STRA3-PNTR
00623 030475          LDA     2,BTMEM-.,1
```

```
0018 MUSME
00624 021000          LDA     0,0,2                 ;AC0 = MEMORY BACK GROUND
00625 105000          MOV     0,1
00626 002611          JMP     @TSTAD-.,1            ;START THE TEST

00627 000545 DPNTR:   PNTR-(TSTAB+1)
00630 000716 DTSTNM:  TSTNM-(NXTST+1)
00631 000037 N37:     37
00632 004000 N4000:   4000

00633 005400 TSTAB:   JSR     0,3
00634 000013          TST1-.
00635 000045          TST2-.
00636 000064          TST3-.
00637 000100          TST4-.
00640 000130          TST5-.
00641 000153          TST6-.
00642 000231          TST7-.
00643 000410          TST10-.
00644 000410          TST11-.
00645 000507          TST12-.
00646 177573          RELCT-.
```

```
                     ;****************************************************************
                     ;TST#1  CHECK THE EXTERNAL ADDRESS CIRCUITRY
                     ;****************************************************************


   00647 165000 TST1:    MOV      3,1
   00650 176521          SUBZL    3,3,SKP              ;AC3=1
   00651 175120          MOVZL    3,3
   00652 156452          SUBO#    2,3,SZC              ;SKIP IF AC3 IS GREATER OR
                                                       ;EQUAL TO AC2
   00653 000776          JMP      .-2-.,1
   00654 054565          STA      3,TEMP-.,1
   00655 054536 XADRP:   STA      3,TEMP2-.,1
   00656 152520          SUBZL    2,2                  ;AC2 = 1
   00657 136053          ADCO#    1,3,SNC              ;SKIP IF AC3 IS NOT GREATER
                                                       ;THAN AC1
   00660 004650 NXTST:   JSR      NEXT-.,1
   00661 030560          LDA      2,TEMP-.,1
   00662 015400          DSZ      0,3                  ;LOCATION POINTED BY AC3 = -1
   00663 021000 XADR:    LDA      0,0,2                ;READ THE LOCATION POINTED
                                                       ;BY AC2
   00664 126400          SUB      1,1
   00665 172415          SUB#     3,2,SNR              ;SKIP IF AC3 IS NOT EQUAL
                                                       ;TO AC2
   00666 124000          COM      1,1
   00667 106414          SUB#     0,1,SZR
   00670 004567          JSR      JP1ER-.,1
   00671 034522          LDA      3,TEMP2-.,1
   00672 024427          LDA      1,TPMEM-.,1
   00673 151120          MUVZL    2,2
   00674 132052          ADCU#    1,2,SZC              ;SKIP IF AC2 IS GREATER THAN AC1
   00675 000766          JMP      XADR-.,1
   00676 102400          SUB      0,0
   00677 041400          STA      0,0,3                ;AC0=0
   00700 175120          MUVZL    3,3
   00701 000754          JMP      XADRP-.,1
```

```
                     ;****************************************************************
                     ;TST#2  CHECK THE EXTERNAL DATA CIRCUITRY
                     ;****************************************************************


   00702 126521 TST2:    SUBZL    1,1,SKP              ;AC1=1
   00703 126120          ADCZL    1,1                  ;AC1 = 177776
   00704 045000 XDTR:    STA      1,0,2                ;WRITE
   00705 021000          LDA      0,0,2                ;READ
   00706 106414          SUB#     0,1,SZR
   00707 004550          JSR      JP1ER-.,1
   00710 121100          MOVL     1,0                  ;MAKE BIT 0 OF AC1 THE CARRY
                                                       ;BIT
   00711 125100          MOVL     1,1
   00712 125235          MOVZR#   1,1,SNR
   00713 000770          JMP      XDTR-1-.,1           ;JUMP IF THE 1 HAS BEEN FLOATED
                                                       ;THROUG ALL THE 16 DATA BITS
   00714 124234          COMZR#   1,1SZR               ;SKIP IF THE 0 HAS BEEN FLOATED
                                                       ;THROUGH ALL THE 16 DATA BITS
   00715 000767          JMP      XDTR-.,1
   00716 130120          COMZL    1,2                  ;AC2 = 2
   00717 000741          JMP      NXTST-.,1

   00720 000000 BTMEM:   0
   00721 000000 TPMEM:   0
```

```
0021 MUSME
                        ;*************************************************************
                        ;TST#3  CHECK THE MEMORY FOR DATA HOLDING CAPABILTIES
                        ;*************************************************************


00722 021000 TST3:   LDA     0,0,2            ;READ
00723 106414         SUB#    0,1,SZR          ;CHECK
00724 004533         JSR     JP1ER-.,1
00725 151400         INC     2,2
00726 172052         ADCO#   3,2,SZC          ;SKIP AFTER READING ALL THE
                                              ;LOCATION
00727 000773         JMP     .-5-.,1
00730 030403         LDA     2,N3-.,1         ;AC2 = 3
00731 102000 JPRT1:  ADC     0,0              ;PREPARE TO WRITE A BACK GROUND
                                              ;OF ALL 1'S
00732 000620         JMP     RIT1S-.,1

00733 000003 N3:     3
00734 000004 N4:     4
00735 000005 N5:     5
00736 000006 N6:     6
```

```
0022 MUSME
                        ;*************************************************************
                        ;TST#4  CHECK THE VOLATILITY OF MEMORY CELLS
                        ;*************************************************************


00737 050454 TST4:   STA     2,TEMP2-.,1      ;"TEMP2" HAS THE ADDRESS OF THE
                                              ;LOC. FOR WHICH READ=COMPLEMENT-
                                              ;WRITE CYCLE WILL BE PERFORMED
      000737 VLT1=   TST4

00740 102400         SUB     0,0
00741 041000         STA     0,0,2            ;AC0=0
00742 015000         DSZ     0,2              ;LOCATION POINTED BY AC2=-1
00743 011000         ISZ     0,2              ;LOCATION POINTED BY AC2=0
00744 001777 C1777:  1777
00745 101404         INC     0,0,SZR
00746 000774         JMP     VLT1+3-.,1       ;STAY IN THE LOOP
00747 045000         STA     1,0,2
00750 030750         LDA     2,BTMEM-.,1      ;"BTMEM" IS THE LOWEST LOCATION
                                              ;UNDER TEST
00751 021000 VLT2:   LDA     0,0,2            ;READ
00752 106414         SUB#    0,1,SZR          ;CHECK
00753 004504         JSR     JP1ER-.,1
00754 151400         INC     2,2
00755 172052         ADCO#   3,2,SZC          ;SKIP AFTER READING ALL THE
                                              ;LOCATIONS
00756 000773         JMP     VLT2-.,1
00757 030434         LDA     2,TEMP2-.,1
00760 020764         LDA     0,C1777-.,1      ;AC0=1777
00761 113000         ADD     0,2
00762 063710         SKPDZ   TTI              ;SKIP IF NO TTI KEY HAS BEEN HIT
00763 000403         JMP     .+3
00764 172052         ADCO#   3,2,SZC          ;SKIP IF AC2 IS BEYOND LIMITS
00765 000752         JMP     VLT1-.,1
00766 030746         LDA     2,N4-.,1         ;AC2 = 4
00767 000742         JMP     JPRT1-.,1
```

```
                       ;*****************************************************************
                       ;TST#5  MARCHING ONE'S AND ZERO'S
                       ;*****************************************************************


00770 102521 TST5:     SUBZL     0,0,SKP              ;AC0=1
00771 113001 MRCH1:    ADD       0,2,SKP              ;AC2 = ADDRESS OF THE NEXT CELL
00772 040421           STA       0,TEMP2-.,1
00773 021000           LDA       0,0,2
00774 122414           SUB#      1,0,SZR              ;SKIP IF THE WORD READ IS OK
00775 004462           JSR       JP1ER-.,1
00776 125004           MOV       1,1,SZR              ;SKIP IF AC1=0
00777 011000           ISZ       0,2                  ;MAKE THE LOC. POINTED BY AC2
                                                      ;A ZERO AND SKIP
01000 015000           DSZ       0,2                  ;MAKE THE LOCATION POINTED BY AC
                                                      ;-1, NEVER SKIP
01001 020412           LDA       0,TEMP2-.,1
01002 172414           SUB#      3,2,SZR              ;SKIP AFTER CHECKING ALL THE
                                                      ;CELLS
01003 000766           JMP       MRCH1-.,1
01004 124000           COM       1,1
01005 034713           LDA       3,BTMEM-.,1          ;AC3 = LOWEST MEM. UNDER TEST
01006 054575           STA       3,STRA3-.,1
01007 100542           NEGOL     0,0,SZC              ;IF AC0 WAS 1 THEN MAKE IT -1
                                                      ;SKIP IF IT WAS ALREADY -1
01010 000762           JMP       MRCH1+1-.,1
01011 030724           LDA       2,N5-.,1             ;AC2 = 5
01012 000717           JMP       JPRT1-.,1

01013 000000 TEMP2:    0
      001013 DGRW=     TEMP2
```

```
                       ;*****************************************************************
                       ;TST#6  SHIFTING DIAGONAL TEST
                       ;*****************************************************************


                                                      ;THIS TEST IS DESIGNED TO CHECK-
                                                      ;OUT THE MEMORY BY WRITING A
                                                      ;DIAGONAL OF 1'S IN A FIELD OF
                                                      ;0'S AND A DIAGONAL OF 0'S IN A
                                                      ;FIELD OF 1'S IN EACH BANK AND
                                                      ;THEN READING FOR CORRECT DATA.
                                                      ;THE DIAGONAL IS SHIFTED THROUGH
                                                      ;EACH ROW OF THE BANK
01014 040425 TST6:     STA       0,TEMP-.,1           ;"TEMP" HAS THE BACKGROUND DATA
01015 020565 DGNOL:    LDA       0,ENDCLM-.,1
01016 100000           COM       0,0
01017 040570           STA       0,TEMP1-.,1          ;"TEMP1" HAS THE NUMBER OF TIMES
                                                      ;THE DIAGONAL WILL BE SHIFTED
01020 050564           STA       2,DGBOT-.,1          ;"DGBOT" HAS THE LOWEST ADDRESS
                                                      ;IN THE BANK UNDER TEST

01021 050565           STA       2,DGBGN-.,1
01022 102120           ADCZL     0,0                  ;AC0 = 177776
01023 040770           STA       0,DGRW-.,1           ;STORE A 177776 AT "DGRW"
01024 030560           LDA       2,DGBOT-.,1
01025 034561           LDA       3,DGBGN-.,1          ;"DGBGN" HAS THE ADDRESS OF
                                                      ;THE LOCATON WHERE THE DIAGONAL
                                                      ;WILL START
01026 024413 DGRIT:    LDA       1,TEMP-.,1
01027 156415           SUB#      2,3,SNR              ;SKIP IF IT WAS NOT A DIAGONAL
                                                      ;LOCATION


01030 124000           COM       1,1
01031 020762           LDA       0,DGRW-.,1           ;"DGRW" WILL BE -2 IF THE DIAGO-
                                                      ;NAL IS BEING WRITTEN AND -1 IF
                                                      ;IT IS ONLY BEING READ
01032 101404           INC       0,0,SZR
01033 045000           STA       1,0,2                ;WRITE
01034 021000           LDA       0,0,2                ;READ
01035 054546           STA       3,STRA3-.,1
01036 122414           SUB#      1,0,SZR              ;CHECK
01037 004420           JSR       JP1ER-.,1
01040 151401           INC       2,2,SKP
01041 000000 TEMP:     0
01042 020543           LDA       0,ENDBNK-.,1
01043 024656           LDA       1,TPMEM-.,1
01044 143414           AND#      2,0,SZR
01045 132053           ADCO#     1,2,SNC
01046 000412           JMP       DGCHK-.,1            ;JUMP IF THE END OF THE BANK
                                                      ;OR TOP OF THE MEMORY HAS
                                                      ;BEEN REACHED
01047 020533           LDA       0,ENDCLM-.,1
01050 172014           ADC#      3,2,SZR              ;SKIP IF THE LOCATION READ/
                                                      ;WRITTEN WAS IN THE DIAGONAL
01051 000755           JMP       DGRIT-.,1
01052 143414           AND#      2,0,SZR              ;SKIP IF IT WAS END OF COLUMN
01053 155401           INC       2,3,SKP
01054 175401           INC       3,3,SKP
01055 117000           ADD       0,3                  ;AC3 HAS THE ADDRESS OF THE
```

```
                                             ;NEXT LOCATION IN THE COLUMN
01056 000750          JMP     DGRIT-.,1

01057 000571 JP1ER:   JMP     JP2ER-.,1

01060 010733 DGCHK:   ISZ     DGRW-.,1
01061 000743          JMP     DGRIT-2-.,1    ;OTHERWISE READ THE MEMORY
01062 010524          ISZ     DGBGN-.,1
01063 063710          SKPDZ   TTI
01064 000405          JMP     END6-1-.,1
01065 010522          ISZ     TEMP1-.,1
01066 000734          JMP     DGRIT-4-.,1    ;REPEAT WITH THE SHIFTED DIAG.
01067 132452          SUBO#   1,2,SZC        ;SKIP IF THE END OF MEMORY
                                             ;HAS BEEN REACHED
01070 000725          JMP     DGNUL-.,1
01071 030645          LDA     2,N6-.,1       ;AC2 = 6
01072 000637 END6:    JMP     JPRT1-.,1
```

```
                      ;************************************************************
                      ;TST#7  CHECK THE WORST CASE ADDRESS AND DATA TRANSITIONS
                      ;************************************************************


01073 050557 TST7:    STA     2,BTM1-.,1      ;"BTM1" HAS THE ADDRESS OF THE
                                              ;LOWEST LOC. TO BE TESTED
01074 050510 RITGLP:  STA     2,DGBOT-.,1     ;"DGBOT" HAS THE LOWEST LOCATION
                                              ;OF THE MEMORY BANK  UNDER TEST
01075 020510          LDA     0,ENDBNK-.,1    ;AC0=END OF A BANK OF MEM
01076 104000          COM     0,1
01077 133400          AND     1,2             ;AC2=LOWEST LOCATION IN
                                              ;A MEMORY BANK
01100 105620          INCZR   0,1
01101 044505          STA     1,DGBGN-.,1     ;"DGBGN" HAS HALF THE NUMBER OF
                                              ;CELLS IN ONE MEMORY BANK
01102 147000          ADD     2,1
01103 143000          ADD     2,0
01104 040735          STA     0,TEMP-.,1      ;"TEMP" HAS TOP OF THE BANK
01105 162053          ADCO#   3,0,SNC         ;AC3=TOP OF THE MEMORY UNDER
01106 161000          MOV     3,0             ;TEST
01107 040500          STA     0,TEMP1-.,1
01110 034542          LDA     3,BTM1-.,1
01111 136452          SUBO#   1,3,SZC         ;SKIP IF THE LOWEST LOCATION TO
                                              ;BE TESTED IN THE BANK IS
                                              ;HIGHER THAN THE MIDDLE OF BANK
01112 122052          ADCO#   1,0,SZC         ;SKIP IF THE HIGHEST LOCATION TO
                                              ;BE TESTED IS HIGHER THAN THE
                                              ;MIDDLE OF THE BANK
01113 000453          JMP     NXTBNK          ;OTHERWISE TRY THE NEXT BANK
01114 135000          MOV     1,3
01115 024577          LDA     1,BKGND-.,1
01116 124000          COM     1,1             ;UPPER HALF OF THE MEMORY WILL
                                              ;BE WRITTEN TO COMPLEMENTED
                                              ;BACKGROUND
01117 045400          STA     1,0,3
01120 175400          INC     3,3
01121 116052          ADCO#   0,3,SZC         ;SKIP AFTER REACHING THE END
01122 000775          JMP     .-3
01123 134520          NEGZL   1,3             ;AC3 = 2 IF UPPER HALF OF MEM.
                                              ;IS WRITTEN TO -1 & 1 IF IT IS 0
01124 054456          STA     3,ENDCLM-.,1    ;SAVE IT
01125 125404          INC     1,1,SZR         ;SKIP IF THE UPPER HALF OF THE
                                              ;MEMORY HAS BEEN WRITTEN TO -1
01126 000475          JMP     RDRC2-.,1
01127 004445 RDRC1:   JSR     RDRC3-.,1       ;SUBROUTINE "RDRC4" WILL RETURN
                                              ;HERE
01130 004460          JSR     RDRC4-.,1       ;STAY IN THE LOOP UNTIL EITHER
                                              ;OF THE TWO CONDITIONS IN "RDRC3
                                              ;OR "RDRC4" ARE MET
01131 004443          JSR     RDRC3-.,1       ;SUBROUTINE "RDRC5"  WILL RETURN
                                              ;HERE
01132 012707          ISZ     @TEMP-.,1       ;INCREMENT AND SKIP
                                              ;CONTROLL WILL COME HERE AFTER
                                              ;THE CONDITION IN RDRC4 IS MET
01133 060077          NIO     CPU             ;NO-OP
01134 004462          JSR     RDRC5-.,1       ;STAY IN THE LOOP UNTIL CONDIT-
                                              ;ION IN "RDRC3" IS MET
01135 015000          DSZ     0,2             ;SUBROUTINE "RDRC4" WILL RETURN
                                              ;HERE
```

```
01136 004452         JSR     RDRC4-.,1          ;STAY IN LOOP UNTIL CONDITION
                                                ;IN "RDRC4" IS MET
01137 015000         DSZ     0,2                ;CONTROL WILL COME HERE IF
                                                ;CONDITION IN "RDRC3" IS MET
                                                ;WHEN CALLED AT RDRC1+1
01140 034701         LDA     3,TEMP-.,1         ;CONTROL WILL COME HERE AFTER
                                                ;THE CONDITION IN RDRC4 IS MET
01141 126401         SUB     1,1,SKP
01142 015000         DSZ     0,2                ;LOCATION POINTED BY AC2=-1
01143 011400         ISZ     0,3                ;INCREMENT AND SKIP
01144 060077         NIO     CPU
01145 151400         INC     2,2
01146 136000         ADC     1,3
01147 014437         DSZ     DGBGN-.,1          ;SKIP AFTER REACHING THE MIDDLE
01150 000772         JMP     .-6-.,1
01151 155000 RDRCH:  MOV     2,3                ;AC3=MIDDLE OF THE BANK
01152 030432         LDA     2,DGBOT-.,1        ;AC2=LOWEST LOCATION OF THE
                                                ;MEMORY BANK
01153 021000         LDA     0,0,2              ;READ
01154 024540         LDA     1,BKGND-.,1
01155 172452         SUBO#   3,2,SZC
01156 124000         COM     1,1
01157 054424         STA     3,STRA3-.,1
01160 106414         SUB#    0,1,SZR            ;CHECK
01161 004467         JSR     JP2ER-.,1
01162 151400         INC     2,2
01163 020424         LDA     0,TEMP1-.,1
01164 112052         ADCO#   0,2,SZC            ;SKIP AFTER CHECKING ALL THE
                                                ;CELLS
01165 000766         JMP     RDRCH+2-.,1
01166 111400 NXTBNK: INC     0,2
01167 034563         LDA     3,TPM1-.,1
01170 172452         SUBO#   3,2,SZC            ;SKIP AFTER REACHING TOP OF THE
                                                ;MEMORY  UNDER TEST
01171 000703         JMP     RITGLP-.,1
01172 030407         LDA     2,N7-.,1           ;AC2 = 7
01173 000677 END7:   JMP     END6-.,1

01174 024456 RDRC3:  LDA     1,BTM1-.,1         ;CONDITION TO BE MET IS:
                                                ;AC2 SHOULD BE GREATER THAN OR
                                                ;EQUAL TO THE LOWEST LOCATION
                                                ;UNDER TEST
01175 132453         SUBO#   1,2,SNC
01176 001405         JMP     5,3                ;IF THE CONDITION IS MET THEN
                                                ;RETURN TO CALLING LOCATION +5.
01177 025000         LDA     1,0,2              ;READ
01200 001400         JMP     0,3                ;AND RETURN

01201 000007 N7:     7
01202 000000 ENDCLM: 0
01203 000000 STRA3:  0
01204 000000 DGBOT:  0
01205 000000 ENDBNK: 0
01206 000000 DGBGN:  0
01207 000000 TEMP1:  0

01210 024631 RDRC4:  LDA     1,TEMP-.,1         ;CONDITION TO BE MET IS:
                                                ;TEMP SHOULD BE LESS THAN OR
                                                ;EQUAL TO THE HIGHEST LOCATION
                                                ;UNDER TEST
```

```
01211 122453         SUBO#   1,0,SNC
01212 005401 LOAD5:  JSR     1,3
01213 026626         LDA     1,@TEMP-.,1        ;READ
01214 024766         LDA     1,ENDCLM-.,1
01215 137000         ADD     1,3                ;ADD 1 OR 2
01216 151400 RDRC5:  INC     2,2
01217 014622         DSZ     TEMP-.,1           ;DECREMENT
01220 014766         DSZ     DGBGN-.,1          ;SKIP AFTER REACHING THE MIDDLE
01221 001774         JMP     -4,3               ;OTHERWISE STAY IN THE LOOP
01222 000727         JMP     RDRCH-.,1

01223 004751 RDRC2:  JSR     RDRC3-.,1          ;SUBROUTINE "RDRC4" WILL RETURN
                                                ;HERE
01224 060077         NIO     CPU                ;NO-OP
01225 004763         JSR     RDRC4-.,1          ;STAY IN THE LOOP UNTIL EITHER
                                                ;OF THE TWO CONDITIONS IN "RDRC3
                                                ;OR "RDRC4" ARE MET
01226 004746         JSR     RDRC3-.,1          ;SUBROUTINE "RDRC5"  WILL RETURN
                                                ;HERE
01227 016612         DSZ     @TEMP-.,1          ;DECREMENT
                                                ;CONTROLL WILL COME HERE AFTER
                                                ;THE CONDITION IN RDRC4 IS MET
01230 004764         JSR     RDRC5-2-.,1        ;STAY IN THE LOOP UNTIL CONDIT-
                                                ;ION IN "RDRC3" IS MET
01231 011000         ISZ     0,2                ;SUBROUTINE "RDRC4" WILL RETURN
                                                ;HERE
01232 060077         NIO     CPU
01233 004755         JSR     RDRC4-.,1          ;STAY IN LOOP UNTIL CONDITION
                                                ;IN "RDRC4" IS MET
01234 011000         ISZ     0,2                ;CONTROL WILL COME HERE IF
                                                ;CONDITION IN "RDRC3" IS MET
                                                ;WHEN CALLED AT RDRC1+1
01235 060077         NIO     CPU
01236 034603         LDA     3,TEMP-.,1         ;CONTROL WILL COME HERE AFTER
                                                ;THE CONDITION IN RDRC4 IS MET
01237 126401         SUB     1,1,SKP
01240 011000         ISZ     0,2                ;LOCATION POINTED BY AC2=0
01241 060077         NIO     CPU                ;NO-OP
01242 015400         DSZ     0,3                ;DECREMENT
01243 151400         INC     2,2
01244 136000         ADC     1,3
01245 014741         DSZ     DGBGN-.,1          ;SKIP AFTER REACHING THE MIDDLE
01246 000772         JMP     .-6-.,1
01247 000702         JMP     RDRCH-.,1

01250 000503 JP2ER:  JMP     JP3ER-.,1
01251 000000 CPUSR:  0                          ;COPY OF USER'S REG.
01252 000000 BTM1:   0
```

```
                            ;*****************************************************************
                            ;TST#10 ROW/COLUMN AND LONG GALLOPING TEST
                            ;*****************************************************************


01253 102001 TST10:  ADC      0,0,SKP          ;ENTER TO PERFORM GALLOPING TEST
                                               ;ON SEQUENTIAL MEMORY CELLS
                                               ;WITH A BOUNDRY OF ONE BANK
                                               ;OR OF ONE ROW/COLUMN
01254 102400 TST11:  SUB      0,0              ;ENTER TO PERFORM GALLOPING TEST
                                               ;ON CELLS 32 OR 64 LOCATIONS
                                               ;APART
01255 040731         STA      0,DGBGN-.,1
01256 034724 GALP1:  LDA      3,ENDCLM-.,1     ;AC3=END OF COLUM/ROW ADD.
01257 054725         STA      3,DGBOT-.,1      ;"DGBOT"=NUMBER OF ROWS ON THE
                                               ;MEMORY CHIP
01260 024771         LDA      1,CPUSR-.,1      ;READ USER'S OPTIONS
01261 127100         ADDL     1,1
01262 101004         MOV      0,0,SZR          ;SKIP IF GALLOPING ON SEQUENTIAL
                                               ;CELLS IS NOT BEING PERFORMED
01263 127102         ADDL     1,1,SZC          ;SKIP IF ONLY ROW/COLUMN GALLOP-
                                               ;ING TEST HAS BEEN SELECTED
01264 034721         LDA      3,ENDBNK-.,1     ;AC3=ADDRESS OF END OF BANK
01265 164000         COM      3,1
01266 147400         AND      2,1
01267 137000         ADD      1,3              ;AC3=ADDRESS OF THE LAST
                                               ;LOCATION FOR GALLOP PURPOSES
01270 024462         LDA      1,TPM1-.,1       ;AC1=TOP OF THE MEMORY UNDER
                                               ;TEST
01271 136053         ADCU#    1,3,SNC          ;SKIP IF AC3 IS LESS OR EQUAL TO
                                               ;AC1
01272 135000         MOV      1,3
01273 050714         STA      2,TEMP1-.,1      ;"TEMP1" IS THE LOWEST LOCATION
                                               ;FOR WHICH GALLOPING WILL BE
                                               ;PERFORMED
01274 050470 GALP2:  STA      2,TEMP3-.,1      ;"TEMP3" IS THE LOCATION UNDER
                                               ;GALLOP TEST
01275 024417         LDA      1,BKGND-.,1
01276 124000         COM      1,1
01277 045000         STA      1,0,2            ;WRITE COMPLEMENTED DATA AT THE
                                               ;LOC. FOR WHICH GALLOPING WILL
01300 030707         LDA      2,TEMP1-.,1
01301 020463 GALP3:  LDA      0,TEMP3-.,1
01302 142405         SUB      2,0,SNR
01303 000410         JMP      .+8.-.,1
01304 021000         LDA      0,0,2            ;READ A LOCATON
01305 020457         LDA      1,@TEMP3-.,1     ;READ THE GALLOP LOCATION
01306 054675         STA      3,STRA3-.,1
01307 125414         INC#     1,1,SZR          ;ONE OF THE TWO LOCATIONS SHOULD
                                               ;HAVE -1
01310 101415         INC#     0,0,SNR
01311 107414         AND#     0,1,SZR          ;MAKE SURE THAT THEY ARE
                                               ;COMPLEMENT OF EACH OTHER
01312 004560         JSR      ERR2-.,1
01313 151401         INC      2,2,SKP
01314 000000 BKGND:  0
01315 020671         LDA      0,DGBGN-.,1      ;"DGBGN" IS -1 FOR SEQUENTIAL
                                               ;GALLOPING AND 0 OTHERWISE
01316 024664         LDA      1,ENDCLM-.,1
01317 101005         MOV      0,0,SNR          ;SKIP IF IT IS NOT ROW/COLUMN
```

```
                                               ;GALLOPING
01320 133000         ADD      1,2
01321 172052         ADCO#    3,2,SZC          ;SKIP AFTER REACHING THE END
01322 000757         JMP      GALP3-.,1

01323 030771 GALP4:  LDA      2,BKGND-.,1
01324 052440         STA      2,@TEMP3-.,1     ;RESTORE THE LOCATION
01325 063710         SKPDZ    TTI              ;CHECK FOR A TTY KEY
01326 000421         JMP      ENDGLP-.,1
01327 010435         ISZ      TEMP3-.,1
01330 030434         LDA      2,TEMP3-.,1
01331 101005         MOV      0,0,SNR
01332 133000         ADD      1,2
01333 172052         ADCO#    3,2,SZC          ;SKIP AFTER REACHING THE END
01334 000740         JMP      GALP2-.,1
01335 101004         MOV      0,0,SZR          ;CHECK FOR THE TYPE OF GALLOPING
                                               ;TEST
01336 000406         JMP      .+6-.,1
01337 030650         LDA      2,TEMP1-.,1
01340 151400         INC      2,2
01341 014643         DSZ      DGBOT-.,1
01342 000731         JMP      GALP2-1-.,1      ;REPEAT ROW/COLUMN GALLOPING
01343 171400         INC      3,2              ;AC2 = STARTING ADDRESS OF THE
                                               ;NEXT BANK/QUADRANT
01344 034406         LDA      3,TPM1-.,1
01345 172052         ADCO#    3,2,SZC          ;SKIP AFTER REACHING THE END
01346 000710         JMP      GALP1-.,1        ;IF GALLOPING IS NOT OVER THEN
                                               ;REPEAT THE TEST (AC2=NEXT
                                               ;LOWEST LOCATON FOR GALLOPING)
01347 030522 ENDGLP: LDA      2,T11-.,1        ;AC2 = 11
01350 113000         ADD      0,2              ;AC0 = 0 FOR TEST 11 AND -1
                                               ;FOR TEST 10
01351 000622         JMP      END7-.,1         ;OTHERWISE WRITE A BACKGROUND
                                               ;OF ALL 1'S
01352 000000 TPM1:   0
01353 000554 JP3ER:  JMP      ERR1-.,1
```

```
                     ;*************************************************************
                     ;TST#12 READ MEMORY THRU INDIRECT ADDRESSING
                     ;*************************************************************


01354 126001 TST12:  ADC     1,1,SKP       ;AC1 = -1
01355 030675 CNTRC:  LDA     2,BTM1        ;AGAIN START FROM THE LOWEST
                                           ;MEM. LOCATION
01356 044406         STA     1,DIREC       ;CLOCKWISE/ANTI-CLOCKWISE
01357 020524         LDA     0,N40         ;AC0 = 40
01360 112452         SUBO#   0,2,SZC       ;DON'T TEST AUTO-INC. AND AUTO-D
                                           ;LOCATIONS
01361 111000         MOV     0,2
01362 050670         STA     2,BTM1
01363 004627 NXTCL:  JSR     LOAD5         ;LOAD AC3 WITH THE ADDRESS
                                           ;OF "LOAD5+1", & RETURN .+2
01364 000000 TEMP3:  0                     ;NEVER RETURN HERE
      001364 DIREC=  TEMP3
01365 051774         STA     2,DA,3        ;STORE THE ADDRESS OF THE LOC.
                                           ;UNDER TEST AT TEMP1.
                                           ;DA = TEMP1-(LOAD5+1)
                                           ;AC2 = ADDRESS OF CELL UNDER TES
01366 020423         LDA     0,M4          ;AC0 = -4
01367 040517         STA     0,COUNT
01370 125112         MOVL#   1,1,SZC       ;SKIP IF MOVING CLOCKWISE
01371 116000         ADC     0,3
01372 020610         LDA     0,ENDCLM      ;AC0 = CELLS IN ONE COLUMN
01373 151400         INC     2,2
01374 051770         STA     2,DB,3        ;SET UP THE FIRST LOCATION
                                           ;DB = STRA3-(LOAD5+1)
01375 137000         ADD     1,3           ;INC./DEC.
01376 113000         ADD     0,2
01377 051770         STA     2,DB,3        ;SET UP THE NEXT LOCATION
01400 137000         ADD     1,3           ;INCREMENT/DECREMENT
01401 112000 PNTR:   ADC     0,2
01402 150400         NEG     2,2
01403 150000         COM     2,2           ;DECREMENT AC2
01404 051770         STA     2,DB,3        ;SET UP THE ADDRESS OF THE NEXT
                                           ;LOCATION
01405 137000         ADD     1,3           ;INCREMENT/DECREMENT
01406 112400         SUB     0,2
01407 051770         STA     2,DB,3        ;STORE THE ADDRESS OF THE NEXT
                                           ;LOCATION
01410 004602         JSR     LOAD5         ;LOAD AC3 WITH THE ADDRESS OF
                                           ;LOAD5 + 1, RETURN .+2
01411 177774 M4:     -4                    ;NEVER RETURN HERE
01412 126000         ADC     1,1
01413 031770 SETUP:  LDA     2,DB,3        ;AC2 = ADDRESS OF LOC. WHICH WIL
                                           ;BE LOADED
01414 175400         INC     3,3
01415 020735         LDA     0,TPM1        ;MAKE SURE THAT ADDRESS IN AC2
01416 142443         SUBO    2,0,SNC       ;IS WITHIN THE MEM. BOUNDRIES
01417 020633         LDA     0,BTM1
01420 112452         SUBO#   0,2,SZC       ;SKIP IF O.K.
01421 000405         JMP     UPDAT
01422 050463         STA     2,LAST        ;"LAST" IS LAST LOCATION WITHIN
                                           ;THE MEM. BOUNDRIES
01423 153240         ADDOR   2,2           ;SET BIT 0 OF AC2
01424 125405         INC     1,1,SNR       ;SKIP IF BEGINING OF INDIRECT LO
                                           ;HAS ALREADY BEEN SETUP
```

```
01425 050457         STA     2,INBGN
01426 031770 UPDAT:  LDA     2,DB,3        ;AC2 = NEXT ADDRESS IN THE LOOP
01427 141100         MOVL    2,0
01430 010456         ISZ     COUNT         ;COUNT WAS SET UP TO -4
01431 111240         MOVOR   0,2           ;FOR FIRST 4 CELLS SET BIT 0
01432 125414         INC#    1,1,SZR       ;SKIP IF LOC. "LAST" HAS NOT BEE
                                           ;INITIALIZED
01433 052452         STA     2,@LAST       ;STORE AT LAST GOOD LOCATION
01434 151112         MOVL#   2,2,SZC       ;SKIP IF INDIRECT LOOP IS OVER
01435 000756         JMP     SETUP
01436 034445         LDA     3,N40         ;AC3 = 40
01437 054447         STA     3,COUNT       ;PREPARE TO LOOP 40 TIMES
01440 024654         LDA     1,BKGND       ;AC1 = MEM. BACK GROUND PAT.
01441 124000         COM     1,1
01442 045000         STA     1,0,2         ;WRITE THE LOCATION UNDER TEST
01443 022441         LDA     0,@INBGN      ;READ THE LOC. UNDER TEST
01444 106414         SUB#    0,1,SZR       ;SKIP IF O.K.
01445 004462         JSR     ERR1
01446 014440         DSZ     COUNT         ;SKIP AFTER LOOPING 10. TIMES
01447 000774         JMP     .-4
01450 124000         COM     1,1           ;BACK TO ORIGINAL BACK GROUND
                                           ;WORD
01451 020433         LDA     0,INBGN
01452 115000         MOV     0,3           ;AC3 = ADDRESS WHERE THE LOOP
                                           ;BEGAN
01453 021400         LDA     0,0,3         ;GET THE NEXT LOC. OF THE LOOP
01454 045400         STA     1,0,3         ;RESTORE THE MEM. LOC.
01455 172414         SUB#    3,2,SZR       ;SKIP AFTER RESTORING THE LOC.
                                           ;UNDER TEST
01456 000774         JMP     .-4
01457 151400         INC     2,2
01460 024704         LDA     1,DIREC       ;AC1 = CIRCULAR DIRECTION
01461 034671         LDA     3,TPM1        ;AC3 = TOP OF THE MEM.
01462 156453         SUBO#   2,3,SNC       ;SKIP IF ALL THE LOC. HAVE
                                           ;BEEN TESTED
01463 000700         JMP     NXTCL         ;REPEAT WITH THE NEXT MEM. CELL
01464 063610         SKPDN   TTI           ;SKIP IF A TTI KEY IS HIT
01465 124142         COMOL   1,1,SZC       ;MOVE CLOCKWISE AND ANTI-CLOCKWI
01466 102521         SUBZL   0,0,SKP       ;AC0 = 1
01467 000666         JMP     CNTRC
01470 000657         JMP     ENDGLP

01471 000011 T11:    11
```

```
                               ;*ERROR ROUTINE
                               ;*-------------
                               ;*
                               ;*THIS SUBROUTINE IS CALLED UPON SEEING AN ERROR. THE
                               ;*SUBROUTINE IS DESIGNED TO REPORT TEST NUMBER, LOC.
                               ;*+1 WHERE THE ERROR WAS DETECTED, CONTENTS OF AC2
                               ;*AS THE FAILING LOCATION, AC1 AS THE EXPECTED DATA
                               ;*AND AC0 AS THE DATA FOUND.
                               ;*ERROR MESSAGE ISN'T REPORTED IF ERROR PRINT-OUTS HAVE
                               ;*BEEN SUPRESSED OR IF THE FAILING BIT HAS ALREADY BEEN
                               ;*REPORTED UNCE

01472 054506 ERR2:    STA      3,ERLOC-.,1       ;LOCATION WHERE THE ERROR WAS
                                                 ;DETECTED
01473 044411          STA      1,STRA1-.,1       ;SAVE AC1
01474 024620          LDA      1,BKGND-.,1       ;"BKGND" HAS THE BACKGROUND
                                                 ;PATTERN WRITTEN  IN THE MEMORY
01475 004433          JSR      ERR1+1-.,1
01476 124000          COM      1,1               ;AC1=PATTERN AND
01477 020405          LDA      0,STRA1-.,1
01500 030664          LDA      2,TEMP3-.,1       ;AC2=ADDRESS OF THE GALLOP
                                                 ;LOCATION
01501 034477          LDA      3,ERLOC-.,1
01502 000427          JMP      ERR1+2-.,1

01503 000040 N40:     40
01504 000000 STRA1:   0
01505 000000 STRA2:   0
01506 000000 RTMES:   0
```

```
                               ;*ERROR ROUTINE CONTINUED

01507 136400 LOAD4:   SUB      1,3               ;AC3 = EGGS
01510 025400          LDA      1,0,3             ;AC1 = CONTENTS OF EGGS
01511 020467          LDA      0,ERLOC-.,1       ;AC0 = ERROR PC
01512 125004          MOV      1,1,SZR           ;SKIP IF NOT UNDER DTOS
01513 001760          JMP      AA,3              ;GO TO "RTRN+1"
01514 020600          LDA      0,BKGND-.,1
01515 024462          LDA      1,TSTNM-.,1       ;AC1 = TEST THAT FAILED
01516 153103          ADDL     2,2,SNC           ;SKIP IF NOT LOOPING ON ERROR
01517 001560          JMP      AB,3              ;GO TO "LOOP", AB = LOOP-EGGS
01520 030512          LDA      2,DSTRA3-.,1      ;AC2 =STRA3-EGGS
01521 157000          ADD      2,3
01522 020452          LDA      0,SAV0-.,1
01523 024452          LDA      1,SAV1-.,1
01524 030507          LDA      2,ERSV2-.,1       ;RESTORE AC2
01525 035400          LDA      3,0,3             ;AND AC3
01526 002507          JMP      @RTERR-.,1        ;AND RETURN

01527 054451 ERR1:    STA      3,ERLOC-.,1
01530 050503          STA      2,ERSV2-.,1       ;SAVE AC2
01531 050445          STA      2,SAV2-.,1        ;SAVE THE ACCUMULATORS
01532 044443          STA      1,SAV1-.,1
01533 040441          STA      0,SAV0-.,1
01534 054501          STA      3,RTERR-.,1       ;SAVE THE RETURN ADDRESS
01535 135000          MOV      1,3
01536 117520          ANDZL    0,3
01537 107000          ADD      0,1               ;TAKE EXCLUSIVE OR OF THE
01540 166400          SUB      3,1               ;DATA FOUND AND THE EXPECTED
                                                 ;DATA
01541 034440          LDA      3,M377-.,1        ;AC3=177400
01542 173700          ANDS     3,2
01543 151220          MOVZR    2,2               ;PLACE THE BANK NUMBER AND
01544 150240          COMOR    2,2               ;QUADRANT NUMBER IN BITS 10-15
01545 020467          LDA      0,N16.-.,1        ;AC0=20 (16.)
01546 004575          JSR      HSTRY-.,1         ;LOAD AC3 WITH THE STARTING
                                                 ;ADDRESS OF ERROR DIRECTORY AND
                                                 ;RETURN TO .+2
01547 117000          ADD      0,3               ;ADJUST AC3 TO THE RIGHT BANK
01550 151404          INC      2,2,SZR
01551 000776          JMP      .-2-.,1
01552 125123 ERRA:    MOVZL    1,1,SNC
01553 000407          JMP      ERRB-.,1
01554 011400          ISZ      0,3               ;INCREMENT THE ERROR COUNTER
                                                 ;FOR THE FAILING BIT
01555 031400          LDA      2,0,3
01556 151112          MOVL#    2,2,SZC
01557 015400          DSZ      0,3               ;FREZE THE ERROR COUNTER AT
                                                 ;77777
01560 151225          MOVZR    2,2,SNR           ;SKIP IF THE BIT FAILURE HAS
                                                 ;ALREADY BEEN REPORTED
01561 102000          ADC      0,0               ;AC0 WILL BE -1 WHEN THE ERROR
                                                 ;IS TO BE REPORTED
01562 175400 ERRB:    INC      3,3
01563 125004          MOV      1,1,SZR           ;SKIP IF DATA FOR ALL THE
                                                 ;FAILING BITS HAVE BEEN
                                                 ;COLLECTED
01564 000766          JMP      ERRA-.,1
01565 034402          LDA      3,CFSWR-.,1       ;SEE IF ERROR PRINT-OUTS ARE
                                                 ;REQUIRED
```

```
 0035 MUSME
   01566 177101              ADDL     3,3,SKP
   01567 000000 CPSWR:  0                                   ;THIS LOCATION WILL HAVE A
                                                            ;COPY OF "SWREG"
   01570 177103              ADDL     3,3,SNC               ;SKIP IF NOT
   01571 101404              INC      0,0,SZR               ;SKIP IF THIS BIT HAS FAILED FOR
                                                            ;THE FIRST TIME
   01572 000410              JMP      M377+1-.,1
   01573 004447              JSR      POCT-.,1              ;PRINT THE FOLLOWING LOCATIONS
                                                            ;IN OCTAL AND RETURN TO "M377"
   01574 000000 SAV0:   0                                   ;DATA FOUND
   01575 000000 SAV1:   0                                   ;EXPECTED DATA
   01576 000000 SAV2:   0                                   ;FAILING LOCATION
   01577 000000 TSTNM:  0                                   ;TEST NUMBER
   01600 000000 ERLOC:  0                                   ;LOCATION WHERE ERROR WAS
                                                            ;DETECTED

   01601 177400 M377:   AND      3,3                        ;BIT 0 OF THIS LOCATION SHOULD
                                                            ;BE 1 (M377 = 177400)
   01602 030765         LDA      2,CPSWR-.,1                ;CHECK IF HALT ON ERROR IS
                                                            ;REQUIRED
   01603 034425         LDA      3,N1000-.,1                ;AC3 = 1000
   01604 157404         AND      2,3,SZR                    ;SKIP IF NOT
   01605 063077         HALT
   01606 046770         STA      1,@SAV2-.,1                ;RESTORE THE FAILING LOC.
   01607 024422         LDA      1,DEGGS-.,1                ;AC1 = LOAD3+1-(EGGS)
   01610 004677 LOAD3:  JSR      LOAD4-.,1
```

```
10036 MUSME
                        ;*MESSAGE PRINT-OUT ROUTINE
                        ;*-------------------------
                        ;*
                        ;*THIS ROUTINE PRINTS THE ASCII MESSAGE RESIDING AT CALL
                        ;*LOCATION +1.  THE MESSAGE SHOULD TERMINATE IN A ZERO
                        ;*BYTE. AC0 AND AC3 ARE DESTROYED, AC0 RETURNS WITH 0.


   01611 050674 MESS:   STA      2,STRA2-.,1        ;SAVE AC2
   01612 054674         STA      3,RTMES-.,1        ;SAVE THE RETURN ADDRESS
   01613 030766         LDA      2,M377-.,1         ;AC2=177400
   01614 150000         COM      2,2
   01615 034671         LDA      3,RTMES-.,1
   01616 021400         LDA      0,0,3              ;READ THE CHARACTER TO BE TYPED
   01617 143705         ANDS     2,0,SNR            ;UNLESS IT IS 0
   01620 000406         JMP      .+6-.,1
   01621 151212         MOVR#    2,2,SZC
   01622 101301         MOVS     0,0,SKP
   01623 010663         ISZ      RTMES-.,1          ;IF THE LEFT BYTE WAS READ THEN
                                                    ;THE NEXT CHARACTER WILL BE THE
                                                    ;RIGHT BYTE OF THE NEXT WORD
   01624 004466         JSR      PRINT-.,1
   01625 000767         JMP      MESS+3-.,1
   01626 030657         LDA      2,STRA2-.,1        ;RESTORE AC2
   01627 001401         JMP      1,3                ;RETURN

   01630 001000 N1000:  1000
   01631 001214 DEGGS:  LOAD3+1-EGGS
   01632 000606 DSTRA3: STRA3-EGGS
   01633 000000 ERSV2:  0
   01634 000020 N16.:   16.
   01635 000000 RTERR:  0
```

```
                      ;*OCTAL AND DECIMAL TYPE-OUTS
                      ;*----------------------------
                      ;*
                      ;*THIS ROUTINE IS USED TO PRINT STRING OF NUMBERS
                      ;*SEPARATED BY " - ". THESE NUMBERS SHOULD BE STORED
                      ;*AFTER CALLING LOC.
                      ;*THE TYPE OUT IS TERMINATED ON SEEING A NUMBER,
                      ;*OTHER THAN THE ONE AT CALLING LOCATION + 1, WITH
                      ;*BIT 0 = 1. CONTROL RETURNS TO THE LOCATION WITH
                      ;*BIT 0 = 1.
                      ;*"POCT" ENTRY ALLOWS THE TYPE OUT OF NUMBERS IN
                      ;*OCTAL WITHOUT SUPPRESSING THE LEADING ZEROS.
                      ;*"PDEC" ENTRY ALLOWS THE TYPE OUT OF NUMBERS IN
                      ;*DECIMAL, LEADING ZEROS SUPPRESSED.
                      ;*IN BOTH OF THE CASES ONLY FIRST NUMBER TO BE TYPED
                      ;*CAN HAVE BIT 0 AS ONE.
                      ;*A RETURN AND LINE FEED IS TYPED BEFORE THE FIRST
                      ;*NUMBER.
                      ;*ALL OF THE ACCUMULATORS ARE DESTROYED.
                      ;*AC2 RETURNS WITH 0.


01636 030475 PDEC:    LDA      2,D10000-.,1    ;AC2 = 10000.
01637 020413          LDA      0,N12-.,1       ;AC0=12
01640 126400          SUB      1,1
01641 000404          JMP      .+4-.,1
01642 024473 POCT:    LDA      1,C60-.,1       ;AC1=60
01643 152620          SUBZR    2,2             ;AC2=100000
01644 020470          LDA      0,N10-.,1       ;AC0 = 10
01645 040471          STA      0,DIVDN-.,1
01646 044473          STA      1,ZSUPP-.,1
01647 054470          STA      3,RTOCT-.,1     ;"RTOCT" WILL CARRY THE ADDRESS
                                               ;OF THE NUMBER TO BE PRINTED
01650 004741          JSR      MESS-.,1
01651 106615 PTXT:    .TXTE    /<15><15><12>/
      000012
01653 000410          JMP      POCT2-.,1

      001652 N12=     PTXT+1

01654 010463 POCT1:   ISZ      RTOCT-.,1
01655 036462          LDA      3,@RTOCT-.,1    ;FETCH THE NEXT WORD
01656 175526          INCZL    3,3,SEZ         ;RETURN WITH BOTH AC3 AND CARRY
                                               ;NON ZERO
01657 002460          JMP      @RTOCT-.,1      ;RETURN
01660 004732          JSR      MESS+1-.,1      ;PRINT " - " AND LOAD AC2
01661 026640 OCTXT:   .TXTE    / - /
      000240
01663 034456 POCT2:   LDA      3,ZSUPP-.,1
01664 026453          LDA      1,@RTOCT-.,1    ;FETCH THE NUMBER TO BE PRINTED
01665 000405          JMP      .+5-.,1
01666 146400          SUB      2,1
01667 101401          INC      0,0,SKP         ;AC0=DIGIT TO BE PRINTED
01670 151235          MOVZR#   2,2,SNR         ;SKIP IF THIS IS THE LAST DIGIT
01671 034444 POCT3:   LDA      3,C60-.,1       ;AC3=60
01672 146453          SUBU#    2,1,SNC
01673 000773          JMP      .-5-.,1         ;FORM THE DIGIT
01674 163005          ADD      3,0,SNR
01675 020606          LDA      0,N40-.,1
01676 054444          STA      3,ZSUP1-.,1
```

```
01677 004413          JSR      PRINT-.,1       ;PRINT A DIGIT OR A SPACE
01700 141000          MOV      2,0
01701 151225          MOVZR    2,2,SNR
01702 000752          JMP      POCT1-.,1       ;IF ALL OF THE DIGITS HAVE BEEN
                                               ;PRINTED THEN JUMP
01703 152400          SUB      2,2
01704 034432          LDA      3,DIVDN-.,1     ;AC3 = 10 FOR OCTAL TYPE OUT
                                               ;AND 12 FOR DECIMAL
01705 151400          INC      2,2
01706 162404          SUB      3,0,SZR
01707 000776          JMP      .-2-.,1
01710 034432          LDA      3,ZSUP1-.,1     ;RESTORE AC3
01711 000757          JMP      POCT3-1-.,1
```

```
10039 MOSME
                      ;*PRINT ROUTINE
                      ;*-------------
                      ;*
                      ;*THIS ROUTINE IS USED TO OUTPUT A CHARACTER TO THE TTY
                      ;*AND LPT DEPENDING ON THE SETTING OF THE SWITCHES.
                      ;*

01712 054426 PRINT:   STA     3,RTPRN-.,1       ;SAVE THE RETURN ADDRESS
01713 034654          LDA     3,CPSWR-.,1
01714 177100          ADDL    3,3
01715 175102          MOVL    3,3,SZC           ;READ SWITCH 2 FOR PRINT-OUTS
                                                ;TO TTY
01716 000405          JMP     PRLPT-.,1
01717 061111          DOAS    0,TTO             ;OUT-PUT THE CHARACTER
01720 063511          SKPBZ   TTO
01721 000777          JMP     .-1
01722 060211          NIOC    TTO

01723 177100 PRLPT:   ADDL    3,3
01724 175103          MOVL    3,3,SNC           ;READ SWITCH 5 FOR PRINT-OUTS
                                                ;TO LPT
01725 002413          JMP     @RTPRN-.,1
01726 061117          DOAS    0,LPT
01727 063517          SKPBZ   LPT
01730 000777          JMP     .-1
01731 060217          NIOC    LPT
01732 002406          JMP     @RTPRN-.,1        ;RETURN

01733 023420 D10000:  10000.
01734 000010 N10:     10
01735 000060 C60:     60
01736 000000 DIVDN:   0
01737 000000 RTOCT:   0
01740 000000 RTPRN:   0
01741 000000 ZSUPP:   0
01742 000000 ZSUP1:   0
01743 005401 HSTRY:   JSR     1,3               ;16 WORD/K OF MEMORY WILL BE
                                                ;ADDED FROM HERE ON FOR ERROR
                                                ;COLLECTION


01744 047503          .TXT    /COPYRIGHT (C) DGC, 1976
      054520
      044522
      044107
      020124
      041450
      020051
      043504
      026103
      030440
      033471
01757 040466 ALL RIGHTS RESERVED./
      046114
      051040
      043511
      052110
      020123
      042522
      042523
      053122
```

```
0040 MOSME
      042105
      000056

01772 147515 DIRT:    .TXTE   !MOSMEM   0!
      046523
      046705
      120240
      030240
      000000
02000 000000          0
02001 000200          START
02002 000002          2
02003 000300          300
02004 000000          0
02005 000000          0
02006 000000          0
02007 000000          0

      177760 AA=      RTRN+1-EGGS
      000160 AB=      LOOP-EGGS
      000012 BB=      MESS-TSTNM
      000042 CA=      MXMEM-(CALCAT+1)
      000035 CB=      TEMPA-(CALCAT+1)
      000037 CC=      HIGHM-(CALCAT+1)
      000041 CD=      ENDPRG-(CALCAT+1)
      177774 DA=      TEMP1-(LOAD5+1)
      177770 DB=      STRA3-(LOAD5+1)
      177601 W=       ENDCLM-PNTR
      000003 X=       IRESTR-0
      177713 Y=       BKGND-PNTR
      000176 Z=       TSTNM-PNTR
      177650 WW=      CPUSR-PNTR
      000166 XX=      CPSWR-PNTR
      177751 YB=      TPM1-PNTR
      177602 YY=      STRA3-PNTR
      177604 ZZ=      ENDBNK-PNTR

      001504 INBGN=   STRA1
      001505 LAST=    STRA2
      001506 COUNT=   RTMES

                      .END

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

```
0041 MOSME

A      000013          2/20    2/27    2/29    2/30    2/31    2/53    3/09
                       3/12    3/18    3/21    3/33    4/53    5/10    5/24
                       5/26    5/27    5/31    6/01   19/01   19/05   19/37
                      20/05   20/26   21/05   21/22   22/05   22/37   23/05
                      23/32   24/05   25/18   26/05   28/49   29/05   29/06
                      30/37   31/05
AA     177760         34/07   40/19
AB     000160         34/11   40/20
B      000014          6/01   19/05   20/05   21/05   22/05   23/05   24/05
                      26/05   29/05   29/06   31/05
BANK   000113          3/43    3/46    3/47    3/49    3/51    9/13    9/29
                       9/45    9/46
BB     000012         16/44   40/21
BITNM  000111          3/43    3/45    9/16    9/25    9/38
BKGND  001314         26/32   27/20   29/38   29/56   30/06   32/13   33/16
                      34/08   40/30
BKPAT  000407         14/19   16/59   17/19
BTCNT  000046          8/03    9/18    9/39
BTM1   001252         26/06   26/23   27/39   28/48   31/07   31/14   31/53
BTMEM  000720         16/29   17/07   17/60   20/24   22/19   23/22
C1777  000744         22/15   22/29
C377   000263         12/15   13/26
C60    001735         37/27   37/55   39/31
CA     000042          6/53   40/22
CALCA  000346         13/12   40/22   40/23   40/24   40/25
CATSW  000377         11/09   12/50   13/08   14/04
CB     000035          6/56   40/23
CC     000037          6/51   40/24
CD     000041          6/58   40/25
CLEAN  000307         12/37   12/43
CNTRC  001355         31/07   32/40
CONT   000340         10/17   13/05   13/19
CONT1  000347         13/03   13/13
COUNT  001506         31/24   32/04   32/12   32/19   40/40
CPSWR  001567         34/59   35/02   35/19   39/09   40/33
CPUSR  001251         28/47   29/17   40/32
D1000  001733         37/23   39/29
DA     177774         31/19   40/26
DB     177770         31/29   31/33   31/38   31/42   31/48   32/02   40/27
DEGGS  001631         35/25   36/28
DGBGN  001206         24/22   24/26   25/08   26/15   27/14   27/54   28/08
                      28/42   29/13   29/57
DGBUT  001204         24/19   24/25   26/08   27/17   27/52   29/15   30/21
DGCHK  001060         24/50   25/06
DGNOL  001015         24/15   25/15
DGRIT  001026         24/29   24/56   25/02   25/07   25/12
DGRW   001013         23/31   24/24   24/35   25/06
DIREC  001364         31/09   31/18   32/32
DIRT   001772          6/04   40/04
DIVDN  001736         37/30   38/07   39/32
DPNTR  000627         17/15   18/05
DSTRA  001632         34/12   36/29
DTSTN  000630         16/37   18/06
EGGS   000375          8/02   11/20   12/08   12/47   14/02   36/28   36/29
                      40/19   40/20
END6   001072         25/10   25/17   27/37
END7   001173         27/37   30/33
ENDBN  001205         24/46   26/10   27/53   29/23   40/36
ENDCL  001202         24/15   24/53   26/42   27/50   28/04   29/14   29/59
```

```
0042 MOSME

                      31/27   40/28
ENDGL  001347         30/09   30/30   32/41
ENDPA  000465         15/17   15/40
ENDPR  000410         13/01   13/31   13/32   14/21   15/18   16/12   40/25
EOP    000136          9/56
ERLOC  001600         33/13   33/23   34/05   34/20   35/14
ERR1   001527         30/36   32/18   33/18   33/24   34/20
ERR2   001472         29/54   33/13
ERRA   001552         34/42   34/58
ERRB   001562         34/43   34/54
ERSM0  000077          9/14    9/54
ERSM1  000104          9/19    9/40
ERSUM  000075          9/12   15/56
ERSV2  001633         34/16   34/21   36/30
ERTMS  000114          3/43    3/53    9/21    9/22    9/31
GALP1  001256         29/14   30/27
GALP2  001274         29/36   30/15   30/22
GALP3  001301         29/43   30/04
GALP4  001323         30/06
HIGHM  000406         12/32   14/17   16/08   40/24
HIMEM  000262          8/21    9/51   11/57   12/14   12/31
HSTRY  001743          6/17   34/36   39/37
IEGGS  000045          6/60    8/02   10/06
IHSTR  000002          6/17    8/14   12/27   15/40
IMESS  000371          6/22    9/60   11/24   11/31   11/42   11/50   11/58
                      12/17   13/37
INBGN  001504         32/01   32/16   32/23   40/38
INPAS  000137          9/58   11/30   12/07   12/26   13/09   15/44
IPDEC  000174          9/23    9/56   11/12
IPOCT  000175         11/13   12/12
IPRIN  000167          6/31   10/25
IREST  000003          6/18   13/15   40/29
JP1ER  001057         19/27   20/11   21/08   22/23   23/11   24/43   25/04
JP2ER  001250         25/04   27/25   28/46
JP3ER  001353         28/46   30/36
JPRT1  000731         21/14   22/36   23/28   25/17
LAST   001505          4/58   31/56   32/08   40/39
LOAD3  001610         35/26   36/28
LOAD4  001507         34/03   35/26
LOAD5  001212         28/02   31/15   31/44   40/26   40/27
LOMEM  000261          8/13   11/49   12/13   12/29
LOOP   000555          4/18    4/53    4/54   17/04   40/20
LOWM   000405         12/30   14/15   16/20   16/25
M377   001601         34/31   35/07   35/17   36/11
M4     001411         31/23   31/46
M7776  000115          6/38    9/32
MAXSV  000253          8/24   12/05
MESS   001611         13/37   36/09   36/23   37/34   37/46   40/21
MMRST  000361         13/27   13/35
MRCH1  000771         23/07   23/20   23/26
MXMEM  000411         12/05   12/46   13/05   13/33   13/34   14/23   15/23
                      40/22
N10    001734         37/29   39/30
N1000  001630         35/21   36/27
N12    001652         37/24   37/39
N15    000165          6/32   10/23
N16.   001634         34/35   36/31
N17    000440         15/03   16/33
```

```
N177   000166        6/29   10/12   10/24   12/33   12/49
N1777  000372       12/52   13/06   13/38
N20    000374        9/17   12/37   13/40   16/52
N3     000733       21/13   21/18
N37    000631       17/38   18/07
N4     000734       21/19   22/35
N40    001503       31/10   32/11   33/26   37/59
N4000  000632       15/50   18/08
N5     000735       21/20   23/27
N6     000736       21/21   25/16
N60    000047        6/36    8/04
N7     001201       27/36   27/49
N7777  000373       13/39   16/05
NEXT   000530       16/37   19/17
NXT1   000544       16/43   16/52
NXTBN  001166       26/30   27/31
NXTCL  001363       31/15   32/36
NXTST  000660       17/03   18/06   19/17   20/22
OCTXT  001661       37/47
OFF    000170       11/07
ON     000171        2/24    3/10    3/15    3/18    3/21    3/33    4/18
                     4/19    4/22    4/27    4/28    4/29    4/30    4/53
                     4/54    5/31   11/08
OPRIN  000012        6/26    6/49
OTHIN  000227       11/42
PASS   000400        2/25    3/33    3/35    4/25    4/26    5/30    5/31
                     5/32   14/05
PDEC   001636       11/12   37/23
PNTR   001401       18/05   31/35   40/28   40/30   40/31   40/32   40/33
                    40/34   40/35   40/36
POCT   001642       11/13   35/08   37/27
POCT1  001654       37/41   38/04
POCT2  001663       37/37   37/49
POCT3  001671       37/55   38/13
PRINT  001712        3/10    4/20    4/21    4/22    4/24    4/25    4/26
                     4/27    4/28    5/32   10/25   36/22   38/01   39/08
PRLPT  001723       39/13   39/19
PTXT   001651       37/35   37/39
QUAD   000112        3/43    3/46    9/15    9/27    9/41
RDRC1  001127       26/46
RDRC2  001223       26/45   28/12
RDRC3  001174       26/46   26/51   27/39   28/12   28/18
RDRC4  001210       26/48   27/01   27/57   28/15   28/28
RDRC5  001216       26/57   28/06   28/23
RDRCH  001151       27/16   27/30   28/10   28/44
RELCT  000441       15/13   16/24   17/05   18/21
RELTG  000412       13/14   13/22   14/25   15/25   15/30   16/14
RESTR  000502        6/18   16/05
RETUR  000401       14/07
RIT1S  000552       16/59   21/16
RITGL  001074       26/08   27/35
RTERR  001635       34/18   34/25   36/32
RTINO  000004        6/19    6/21    6/34    6/35    9/19    9/36    9/37
                    15/41
RTMES  001506       33/29   36/10   36/13   36/19   40/40
RTOCT  001737       37/32   37/41   37/42   37/45   37/50   39/33
RTPRN  001740       39/08   39/22   39/27   39/34
RTRN   000354       10/11   13/20   40/19
```

```
SAV0   001574       34/14   34/24   35/10
SAV1   001575       34/15   34/23   35/11
SAV2   001576       34/22   35/12   35/24
SEND   000547       16/35   16/56
SETUP  001413       31/48   32/10
SIZE   000060        2/40    8/13    8/25   11/41
SIZED  000260       10/21   12/12
START  000200        2/25    5/10    5/12    5/21    5/26   10/19   11/10
                    11/17   40/11
STRA1  001504       33/15   33/20   33/27   40/38
STRA2  001505       33/28   36/09   36/24   40/39
STRA3  001203       23/23   24/41   27/23   27/51   29/48   36/29   40/27
                    40/35
SWREG  000402       11/23   12/11   14/09   15/49   17/38
T11    001471       30/30   32/43
TEMP   001041       19/12   19/18   24/14   24/29   24/45   26/19   26/53
                    27/06   27/57   28/03   28/07   28/20   28/34
TEMP1  001207       24/17   25/11   26/22   27/27   27/55   29/33   29/42
                    30/19   40/26
TEMP2  001013       19/13   19/28   22/06   22/28   23/08   23/17   23/38
                    23/31
TEMP3  001364       29/36   29/43   29/47   30/07   30/10   30/11   31/17
                    31/18   33/21
TEMPA  000404       12/44   14/13   40/23
TERM   000146       10/06
TESTM  000000  MC    6/02   19/01   19/05   19/37   20/05   20/26   21/05
                    21/22   22/05   22/37   23/05   23/32   24/05   25/18
                    26/05   28/49   29/05   30/37   31/05
TESTN  000024  MC    6/02   19/01   19/37   20/26   21/22   22/37   23/32
                    25/18   28/49   30/37
TINO   000005        6/21   11/29   11/36   11/48   11/56   12/04
TOCAT  000036        6/51   13/12
TPM1   001352       27/32   29/28   30/25   30/35   31/51   32/33   40/34
TPMEM  000721       16/30   17/06   17/54   19/29   20/25   24/47
TST1   000647       18/11   19/06
TST10  001253       18/18   29/06
TST11  001254       18/19   29/10
TST12  001354       18/20   31/06
TST2   000702       18/12   20/06
TST3   000722       18/13   21/06
TST4   000737       18/14   22/06   22/09
TST5   000770       18/15   23/06
TST6   001014       18/16   24/14
TST7   001073       18/17   26/06
TSTAB  000633       17/13   18/05   18/10
TSTAD  000437       14/27   17/39   18/03
TSTNM  001577        3/24    3/29   18/06   34/09   35/13   40/21   40/31
UNDTS  000224       11/22   11/37
UPDAT  001426       31/55   32/02
USREG  000403       11/37   14/11   15/13   16/34   16/53   17/33
VLT1   000737       22/09   22/17   22/34
VLT2   000751       22/21   22/27
W      177601       17/43   40/28
WW     177650       17/34   40/32
X      000003        2/43    2/45   15/60   40/29
XADR   000663       19/20   19/32
XADRP  000655       19/13   19/36
XDTR   000704       20/08   20/16   20/20
```

```
  ØØ45 MOSME

XX     ØØØ166        17/31   4Ø/33
Y      177713         2/42    2/43    2/45   17/20   4Ø/3Ø
YB     177751        17/56   4Ø/34
YY     1776Ø2        17/58   4Ø/35
Z      ØØØ176        17/22   4Ø/31
ZSUP1  ØØ1742        37/6Ø   38/12   39/36
ZSUPP  ØØ1741        37/31   37/49   39/35
ZZ     1776Ø4        17/51   4Ø/36
```