```
CARD # LOC  CODE  CARD
     2              ;
     3              ;     MICROCOMPUTER ASSOCIATES, INC.
     4              ;             DEMON DEBUG MONITOR
     5              ;             VERSION 1.0  AUGUST 31, 1975
     6              ;             COPYRIGHT 1975 MOS TECHNOLOGY
     7              ;             ALL RIGHTS RESERVED.  UNAUTHORIZED USE
     8              ;             OF ALL OR PART STRICTLY PROHIBITED.
     9              ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    10              ;
    11              ; PROMPTING CHARACTER IS A PERIOD (.)
    12              ; ----------------------------------
    13              ;
    14              ;
    15              ; DISPLAY COMMANDS
    16              ; ----------------
    17              ;
    18              ; .R          DISPLAY REGISTERS (PC,F,A,X,Y,SP)
    19              ; .M  ADDR    DISPLAY MEMORY ( 8 BYTES BEGINNING AT ADDR )
    20              ;
    21              ;
    22              ; ALTER COMMAND (:)
    23              ; -----------------
    24              ; .:  DATA      ALTERS PREVIOUSLY DISPLAYED ITEM OR NEXT ITEM
    25              ;
    26              ;
    27              ; PAPER TAPE I/O COMMANDS
    28              ; -----------------------
    29              ;
    30              ; .LH                     LOAD HEX TAPE
    31              ; .WB ADDR1 ADDR2         WRITE BNPF TAPE (FROM LOW ADDR1 TO HIGH ADDR2)
    32              ; .WH ADDR1 ADDR2         WRITE HEX TAPE (FROM LOW ADDR1 TO HIGH ADDR2)
    33              ;
    34              ; CONTROL COMMANDS
    35              ; ----------------
    36              ;
    37              ; .G                      GO, CONTINUE EXECUTION FROM CURRENT PC ADDRESS
    38              ;
    39              ; .H                      TOGGLES HIGH-SPEED-READER OPTION
    40              ;                           (IF ITS ON, TURNS IT OFF; IF OFF, TURNS ON
    41              ;
    42              ; BRK AND NMI ENTRY POINTS TO DEMON
    43              ; ---------------------------------
    44              ;
    45              ;     DEMON IS NORMALLY ENTERED WHEN A 'BRK' INSTRUCTION IS
    46              ;         ENCOUNTERED DURING PROGRAM EXECUTION.  AT THAT
    47              ;         TIME CPU REGISTERS ARE OUTPUT:  * PC F A X Y SP
    48              ;         AND CONTROL IS GIVEN TO THE KEYBOARD.
    49              ;     USER MAY ENTER DEMON BY PROGRAMMED BRK OR INDUCED NMI.  NMI
    50              ;         ENTRIES CAUSE A '#' TO PRECEDE THE '*' IN THE CPU REGISTER
    51              ;         PRINTOUT FORMAT
    52              ;
    53              ; NON-BRK IRQ (EXTERNAL DEVICE) INTERRUPT HANDLING
```

```
CARD # LOC  CODE      CARD
   54               ;  ------------------------------------------------
   55               ;
   56               ;       A NON-BRK IRQ INTERRUPT CAUSES AN INDIRECT JUMP TO THE ADDRESS
   57               ;           LOCATED AT 'UINT' (HEX FFF8).  THIS LOCATION CAN BE SET
   58               ;           USING THE ALTER CMD, OR LOADED AUTOMATICALLY IN PAPER TAPE
   59               ;           FORM WITH THE LH CMD IF THE USER ASSIGNS HIS IRQ INTERRUPT
   60               ;           VECTOR TO $FFF8 IN THE SOURCE ASSEMBLY PROGRAM.
   61               ;       IF NOT RESET BY THE USER, UINT IS SET TO CAUSE EXTERNAL
   62               ;           DEVICE INTERRUPTS TO ENTER DEMON AS NMI'S.  I.E.,
   63               ;           IF A NMI OCCURS WITHOUT AN INDUCED NMI SIGNAL, IT IS
   64               ;           AN EXTERNAL DEVICE INTERRUPT.
   65               ;
   66               ;  SETTING AND RESETTING PROGRAM BREAKPOINTS
   67               ;  ----------------------------------------
   68               ;
   69               ;       BREAKPOINTS ARE SET AND RESET USING THE MEMORY DISPLAY
   70               ;           AND ALTER COMMANDS.  BRK HAS A '00' OPERATION CODE.
   71               ;       TO SET A BREAKPOINT SIMPLY DISPLAY THE MEMORY LOCATION
   72               ;           (FIRST INSTRUCTION BYTE) AT WHICH THE BREAKPOINT IS
   73               ;           TO BE PLACED THEN ALTER THE LOCATION TO '00'.  THERE IS
   74               ;           NO LIMIT TO THE NUMBER OF BREAKPOINTS THAT CAN BE
   75               ;           ACTIVE AT ONE TIME.
   76               ;       TO RESET A BREAKPOINT, RESTORE THE ALTERED MEMORY LOCATION
   77               ;           TO ITS ORIGINAL VALUE.
   78               ;       WHEN AND IF A BREAKPOINT IS ENCOUNTERED DURING EXECUTION,
   79               ;           THE BREAKPOINT DATA PRECEDED BY AN '*' IS DISPLAYED.
   80               ;           THE PROGRAM COUNTER VALUE DISPLAYED IS THE BRK
   81               ;           INSTRUCTION LOCATION + 1.
   82               ;
   83               ;  ---------------------------------------------------------------------
   84               ;
   85               MDBK   =%00010110        ; X,X,X,PCR,DATA-AVAIL,GOT-DATA,SERIAL-OUT,IN
   86               DAVAIL =$08
   87               GOTDAT =$04
   88               ICBASE =$6ECC
   89               MPA    =ICBASE+0
   90               MCA    =ICBASE+1
   91               MPB    =ICBASE+2
   92               MCB    =ICBASE+3
   93               MCLKIT =ICBASE+4
   94               MCLKRD =ICBASE+4
   95               MCLKIF =ICBASE+5
   96               UINT   =$FFF8
   97               NCMDS  =7
   98               MPC    =$7000
   99               MP1    =$71CC
  100               MP2    =$7200
  101               MP3    =$7300
  102               ;
  103               ;  ZERO PAGE MONITOR RESERVE AREA
  104               ;
  105               CRDLY  =227              ;DELAY FOR CR IN BIT-TIMES
```

```
CARD # LOC  CODE      CARD
  106               WRAP   =228          ;ADDRESS WRAP-AROUND FLAG
  107               DIFF   =229
  108               HSPTR  =231
  109               HSRCP  =232
  110               PREVC  =233
  111               MAJCRT =234
  112               MINCRT =235
  113               ACMD   =236
  114               TMP0   =238
  115               TMP2   =240
  116               TMP4   =242
  117               TMP6   =244
  118               PCL    =246
  119               PCH    =247
  120               FLGS   =248
  121               ACC    =249
  122               XR     =250
  123               YR     =251
  124               SP     =252
  125               SAVX   =253
  126               TMPC   =254
  127               TMPC2  =255
  128               RCNT   =TMPC
  129               LCNT   =TMPC2
  130               ;
  131               ;  64 BYTE RAM MONITER RESERVE AREA
  132               ;
  133               RAM64  =$FFCC
  134  CCCC                 *=RAM64
```

```
CARD # LOC  CODE      CARD
  136                 ;
  137                 ;
  138                 ;  DEMON PAGE 0 (RELATIVE)
  139  FF00                 *=MP0
  140                 ;
  141  7000  85 F9    NMINT  STA ACC          ; SAVE A
  142  7002  A9 23           LDA #'#          ; SET A=# TO INDICATE NMINT ENTRY
  143  7004  D0 55           BNE B3           ; JMP B3
  144                 ;
  145  7006  A9 16    RESET  LDA #MDBK        ; INIT DIR REG, POR TO 1 RELOCATES
  146                 ;
  147  7008  8D C3 6E        STA MDB
  148                 ;
  149  700B  A2 08           LDX #8           ; X=0
  150  700D  BD F7 73 R1     LDA INTVEC-1,X   ; INITIALIZE INT VECTORS
  151  7010  9D F7 FF        STA LINT-1,X
  152  7013  CA              DEX
  153  7014  D0 F7           BNE R1
  154                 ;
  155  7016  86 EA           STX MAJCRT       ; INIT MAJOR T COUNT TO ZERO
  156  7018  86 E7           STX HSPTR        ; CLEAR HSPTR FLAGS
  157  701A  86 E8           STX HSROP
  158  701C  CA              DEX              ; X=FF
  159  701D  9A              TXS              ; SP=FF
  160                 ;
  161                 ;                      ; COMPUTE BIT-TIME CONSTANT, X=FF
  162                 ;
  163  701E  A0 01           LDY #1           ; SET TO MEASURE 2 BITS
  164  7020  84 E3           STY CRDLY        ;INIT CR DELAY TIME PARAMETER
  165  7022  AD 02 6E  R0    LDA MPB          ; WAIT FOR START
  166  7025  4A              LSR A
  167  7026  90 FA           BCC R0
  168                 ;
  169  7028  8E C4 6E  R2    STX MCLKIT       ; START CLOCK INITIALLY WITH FF
  170  702B  AD 05 6E  R3    LDA MCLKIF
  171  702E  10 04           BPL R4
  172  7030  E6 EA           INC MAJCRT       ; COUNT MAJOR T
  173  7032  D0 F4           BNE R2           ; GO RESTART CLOCK WITH X = FF
  174                 ;
  175  7034  98      R4      TYA
  176  7035  4D 02 6E        EOR MPB
  177  7038  29 01           AND #1
  178  703A  F0 EF           BEQ R3           ; WAIT FOR Y BIT 0 AND SERIAL-IN NOT EQU
  179  703C  88              DEY
  180  703D  10 EC           BPL R3           ; LOOP UNTIL START OF BIT 2
  181                 ;
  182  703F  AD C4 6E        LDA MCLKRD
  183  7042  49 FF           ECR #$FF         ; COMPLEMENT RESIDUE
  184  7044  4A      R5      LSR A            ; HALF IT
  185  7045  46 EA           LSR MAJCRT       ; HALF MAJOR
  186  7047  90 02           BCC R6
  187  7049  09 80           ORA #$80         ; PROPAGATE HO TO LO
```

```
CARD #  LOC    CODE        CARD
  188   7C4B   C8          R6      INY
  189   7C4C   FC F6               BEQ R5
  19C   7C4E   85 EB               STA MINCRT
  191                      ;
  192   7C50   58                 CLI                 ; ENABLE INTS
  193   7C51   CC                 BRK                 ; ENTER DEMON BY BRK
  194                      ;
  195   7C52   85 F9       INTRQ   STA ACC             ; SAVE ACC
  196   7C54   68                 PLA                 ; FLAGS TO A
  197   7C55   48                 PHA                 ; RESTORE STACK STATUS
  198   7C56   29 1C              AND #$10            ; TEST BRK FLAG
  199   7C58   FC 27              BEQ BX              ; USER INTERRUPT
  2CC                      ;
  2C1   7C5A   CA                 ASL A               ; SET A=SPACE (10 X 2 = 20)
  202   7C5B   85 FE       B3      STA TMPC            ; SAVE INT TYPE FLAG
  2C3   7C5D   D8                 CLD                 ; CLEAR DECIMAL MODE
  2C4   7C5E   4A                 LSR A               ; # IS ODD, SPACE IS EVEN
  2C5                             ;                   ; SET CY FOR PC BRK CORRECTION
  2C6                      ;
  2C7   7C5F   8E FA              STX XR              ; SAVE X
  2C8   7C61   84 F8              STY YR              ; Y
  2C9   7C63   68                 PLA
  21C   7C64   85 F8              STA FLGS            ; FLAGS
  211   7C66   68                 PLA
  212   7C67   69 FF              ADC #$FF            ; CY SET TO PC-1 FOR BRK
  213   7C69   85 F6              STA PCL
  214   7C6B   68                 PLA
  215   7C6C   69 FF              ADC #$FF
  216   7C6E   85 F7              STA PCH
  217   7C7C   BA                 TSX
  218   7C71   86 FC              STX SP              ; SAVE ORIG SP
  219                      ;
  22C   7C73   2C 8A 72    B5      JSR CRLF
  221   7C76   A6 FE              LDX TMPC
  222                      ;
  223   7C78   A9 2A              LDA #'*
  224   7C7A   2C CC 72           JSR WRTWO
  225   7C7D   A9 52              LDA #'R             ; SET FOR R DISPLAY TO PERMIT
  226   7C7F   DC 16              BNE SC              ;   IMMEDIATE ALTER FOLLOWING BREAKPOINT.
  227                      ;
  228   7C81   A5 F9       BX      LDA ACC
  229   7C83   6C FE FF           JMP (UINT)          ; CONTROL TO USER IRQ SERVICE ROUTINE
  23C                      ;
  231   7C86   A9 CC       START   LDA #0              ;NEXT COMMAND FROM USER
  232   7C88   85 E7               STA HSPTR           ;CLEAR H. S. PAPER TAPE FLAG
  233   7C8A   85 E4               STA WRAP            ;CLEAR ADDRESS WRAP-AROUND FLAG
  234   7C8C   2C 8A 72           JSR CRLF
  235   7C8F   A9 2E              LDA #'.             ; TYPE PROMPTING '.'
  236   7C91   2C C6 72           JSR WROC
  237   7C94   2C E9 72           JSR RDOC            ; READ CMD, CHAR RETURNED IN A
  238                      ;
  239   7C97   A2 C6       SC      LDX #NCMDS-1        ; LOOK-UP CMD
```

```
CARD #  LOC   CODE        CARD
  240   7C99  DD 06 71   S1      CMP CMDS,X
  241   7C9C  D0 19             BNE S2
  242                     ;
  243   7C9E  A5 FD             LDA SAVX            ; SAVE PREVIOUS CMD
  244   7CA0  85 E9             STA PREVC
  245   7CA2  86 FD             STX SAVX            ; SAVE CURRENT CMD INDEX
  246   7CA4  A9 71             LDA #MP1/256        ; JMP INDIRECT TO CMD CODE
  247   7CA6  85 ED             STA ACMD+1          ;   ALL CMD CODE BEGINS ON MP1
  248   7CA8  BD CC 71          LDA ADRS,X
  249   7CAB  85 EC             STA ACMD
  250   7CAD  E0 C3             CPX #3              ; IF :, R OR M (0, 1, OR 2)  SPACE 2
  251   7CAF  B0 03             BCS IJMP
  252   7CB1  20 74 73          JSR SPAC2
  253                     ;
  254   7CB4  6C EC CC  IJMP    JMP (ACMD)
  255                     ;
  256   7CB7  CA        S2      DEX
  257   7CB8  10 DF             BPL S1              ; LOOP FOR ALL CMDS
  258                     ;
  259   7CBA  A9 3F     ERRCPR  LDA #'?             ; OPERATOR ERR, TYPE '?', RESTART
  260   7CBC  20 C6 72          JSR WRCC
  261   7CBF  90 C5             BCC START           ; JMP START (WRCC RETURNS CY=0)
  262                     ;
  263   7CC1  38        DCMP    SEC                 ; TMP2-TMP0 DOUBLE SUBTRACT
  264   7CC2  A5 F0             LDA TMP2
  265   7CC4  E5 EE             SBC TMP0
  266   7CC6  85 E5             STA DIFF
  267   7CC8  A5 F1             LDA TMP2+1
  268   7CCA  E5 EF             SBC TMP0+1
  269   7CCC  A8                TAY                 ;RETURN HIGH ORDER PART IN Y
  270   7CCD  C5 E5             CMP DIFF            ; CY LO FOR EQL TEST
  271   7CCF  60                RTS
  272                     ;
  273   7CD0  A5 EE     PUTP    LDA TMP0            ; MOVE TMP0 TO PCH,PCL
  274   7CD2  85 F6             STA PCL
  275   7CD4  A5 EF             LDA TMP0+1
  276   7CD6  85 F7             STA PCH
  277   7CD8  60                RTS
  278                     ;
  279   7CD9  A9 00     ZTMP    LDA #0              ; CLEAR REGS
  280   7CDB  95 EE             STA TMP0,X
  281   7CDD  95 EF             STA TMP0+1,X
  282   7CDF  60                RTS
  283                     ;
  284                     ;  READ AND STORE BYTE.  NO STORE IF SPACE OR RCNT=0.
  285                     ;
  286   7CE0  20 B3 73  BYTE    JSR RDCE            ; CHAR IN A, CY=0 IF SP
  287   7CE3  90 10             BCC BY3             ; SPACE
  288                     ;
  289   7CE5  A2 00             LDX #0              ; STORE BYTE
  290   7CE7  81 EE             STA (TMP0,X)
  291                     ;
```

```
CARD #  LOC    CODE        CARD
  292   7CE9   C1 EE             CMP (TMPC,X)      ; TEST FOR VALID WRITE (RAM)
  293   7CEB   F0 05             BEQ BY2
  294   7CED   68                PLA               ; ERR, CLEAR JSR ADR IN STACK
  295   7CEE   68                PLA
  296   7CEF   4C BA 70          JMP ERRCPR
  297                    ;
  298   7CF2   20 7C 72  BY2     JSR DADD          ; INCR CKSUM
  299   7CF5   20 97 73  BY3     JSR INCTMP        ; GO INCR TMPC ADR
  300   7CF8   C6 FE             DEC RCNT
  301   7CFA   60                RTS
  302                    ;
  303   7CFB   A9 F8     SETR    LDA #FLGS         ; SET TO ACCESS REGS
  304   7CFD   85 EE             STA TMPC
  305   7CFF   A9 CC             LDA #0
  306   71C1   85 EF             STA TMPC+1
  307   71C3   A9 05             LDA #5
  308   71C5   60                RTS
  309                    ;
  310   71C6   3A       CMDS    .BYTE ':'
  311   71C7   52                .BYTE 'R'
  312   71C8   4D                .BYTE 'M'
  313   71C9   47                .BYTE 'G'
  314   71CA   48                .BYTE 'H'
  315   71CB   4C                .BYTE 'L'
  316   71CC   57                .BYTE 'W'         ; W MUST BE LAST CMD IN CHAIN
  317   71CD   3A       ADRS    .BYTE ALTER-MP1
  318   71CE   14                .BYTE DSPLYR-MP1
  319   71CF   1C                .BYTE DSPLYM-MP1
  320   711C   5C                .BYTE GO-MP1
  321   7111   6F                .BYTE HSP-MP1
  322   7112   74                .BYTE LH-MP1
  323   7113   C2                .BYTE WC-MP1
```

```
CARD # LOC   CODE       CARD
  325                   ;
  326                   ;
  327                   ;  NOTE -- ALL CMD CODE MUST BEGIN ON MP1
  328                   ;
  329                   ;  DISPLAY REG CMD - A,F,X,Y, AND SP
  330                   ;
  331  7114  20 A6 72   DSPLYR JSR WRPC          ; WRITE PC
  332  7117  20 FB 7C          JSR SETR
  333  711A  D0 C7             BNE MC           ; USE DSPLYM
  334                   ;
  335  711C  20 A4 73   DSPLYM JSR RDDA          ; READ MEM ADR INTO TMPC
  336  711F  90 16             BCC ERRS1        ; ERR IF NO ADDR
  337  7121  A9 08             LDA #8
  338  7123  85 FE      MC     STA TMPC
  339  7125  A0 00             LDY #0
  340  7127  20 77 73   M1     JSR SPACE        ; TYPE 8 BYTES OF MEM
  341  712A  B1 EE             LDA (TMPC),Y     ; (TMPC) PRESERVED FOR POSS ALTER
  342  712C  20 B1 72          JSR WRDB
  343  712F  C8                INY              ; INCR INDEX
  344  7130  C6 FE             DEC TMPC
  345  7132  D0 F3             BNE M1
  346  7134  4C EE 7C   BEGS1  JMP START
  347                   ;
  348  7137  4C BA 7C   ERRS1  JMP ERRDSP
  349                   ;
  350                   ;  ALTER LAST DISPLAYED ITEM (ADR IN TMPC)
  351                   ;
  352  713A  C6 E9      ALTER  DEC PREVC         ; R INDEX = 1
  353  713C  D0 0D             BNE A3
  354                   ;
  355  713E  20 A4 73          JSR RDDA         ; CY=0 IF SP
  356  7141  90 03             BCC A2           ; SPACE
  357  7143  20 DC 7C          JSR PUTP         ; ALTER PC
  358  7146  20 FB 70   A2     JSR SETR         ; ALTER R'S
  359  7149  D0 05             BNE A4           ; JMP A4 (SETR RETURNS ACC = 5)
  360  714B  20 9A 72   A3     JSR WRDA         ; ALTER M, TYPE ADR
  361  714E  A9 08             LDA #8           ; SET CNT=8
  362                   ;
  363  7150  85 FE      A4     STA RCNT
  364  7152  20 77 73   A5     JSR SPACE        ; PRESERVES Y
  365  7155  20 E0 7C          JSR BYTE
  366  7158  D0 F8             BNE A5
  367  715A  F0 D8      A9     BEQ BEGS1
  368                   ;
  369  715C  A6 FC      GO     LDX SP
  370  715E  9A                TXS              ; CHIC OR NEW SP VALUE TO SP
  371  715F  A5 F7             LDA PCH
  372  7161  48                PHA
  373  7162  A5 F6             LDA PCL
  374  7164  48                PHA
  375  7165  A5 F8             LDA FLGS
  376  7167  48                PHA
```

```
CARD #  LOC   CODE        CARD
   377  7168  A5 F9            LDA ACC
   378  716A  A6 FA            LDX XR
   379  716C  A4 FB            LDY YR
   380  716E  4C               RTI
   381                  ;
   382  716F  E6 E8      HSP   INC HSRCP       ; TOGGLE BIT C
   383  7171  4C EE 7C         JMP START
   384                  ;
   385  7174  20 E9 72  LH    JSR RDCC          ; READ SECOND CMD CHAR
   386  7177  20 8A 72        JSR CRLF
   387  717A  A6 E8            LDX HSROP       ; ENABLE PTR OPTION IF SET
   388  717C  86 E7            STX HSPTR
   389  717E  20 E9 72  LH1   JSR RDOC
   390  7181  C9 3B            CMP #';         ; FIND NEXT RCD MARK (;)
   391  7183  D0 F9            BNE LH1
   392                  ;
   393  7185  A2 C4            LDX #4
   394  7187  20 D9 7C         JSR ZTMP        ; CLEAR CKSUM REGS TMP4
   395  718A  20 B3 73         JSR RDOB
   396  718D  D0 C6            BNE LH2
   397                  ;
   398  718F  A2 00            LDX #0          ; CLEAR HS RCR FLAG
   399  7191  86 E7            STX HSPTR
   400  7193  F0 9F            BEQ BEGS1       ; FINISHED
   401                  ;
   402  7195  85 FE      LH2   STA RCNT        ; RCNT
   403  7197  20 7C 72         JSR DADD        ; RCD LNGH TO CKSUM
   404  719A  20 B3 73         JSR RDOB        ; SA HO TO TMPC+1
   405  719D  85 EF            STA TMPC+1
   406  719F  20 7C 72         JSR DADD        ; ADD TO CKSUM
   407  71A2  20 B3 73         JSR RDOB        ; SA LO TO TMPC
   408  71A5  85 EE            STA TMPC
   409  71A7  20 7C 72         JSR DADD        ; ADD TO CKSUM
   410                  ;
   411  71AA  20 EC 7C  LH3   JSR BYTE        ; BYTE SUB/R DECRS RCNT ON EXIT
   412  71AD  D0 FB            BNE LH3
   413  71AF  20 A4 73         JSR RDCA        ; CKSUM FROM HEX RCD TO TMP0
   414  71B2  A5 F2            LDA TMP4        ; TMP4 TO TMP2 FOR COMP
   415  71B4  85 F0            STA TMP2
   416  71B6  A5 F3            LDA TMP4+1
   417  71B8  85 F1            STA TMP2+1
   418  71BA  20 C1 7C         JSR COMP
   419  71BD  F0 BF            BEQ LH1
   420  71BF  4C BA 70  ERRP1 JMP ERRCPR
   421                  ;
   422  71C2  20 E9 72  WC    JSR RDOC        ; RD 2ND CMD CHAR
   423  71C5  85 FE            STA TMPC
   424  71C7  20 77 73         JSR SPACE
   425  71CA  20 A4 73         JSR RDCA
   426  71CD  20 87 73         JSR T2T2        ; SA TO TMP2
   427  71D0  20 77 73         JSR SPACE       ; SPACE BEFORE NEXT ADDRESS
   428  71D3  20 A4 73         JSR RDCA
```

```
CARD # LOC   CODE       CARD
  429 71D6  20 E7 72        JSR T2T2          ; SA TO TMP0, EA TO TMP2
  430 71D9  20 E9 72        JSR RDOC          ; DELAY FOR FINAL CR
  431 71DC  A5 FE           LDA TMPC
  432                  ;
  433 71DE  C9 48           CMP #'H
  434 71E0  D0 59           BNE WB
  435                  ;
  436 71E2  A6 E4      WHC  LDX WRAP          ;IF ADDR HAS WRAPPED AROUND
  437 71E4  D0 52           BNE BCCST         ;THEN TERMINATE WRITE OPERATION
  438                  ;
  439 71E6  20 8A 72        JSR CRLF
  440 71E9  A2 18           LDX #24
  441 71EB  86 FE           STX RCNT          ; RCNT=24
  442 71ED  A2 C4           LDX #4            ; CLEAR CKSUM
  443 71EF  20 D9 7C        JSR ZTMP
  444                  ;
  445 71F2  A9 3B           LDA #';
  446 71F4  20 C6 72        JSR WROC          ; WR RCD MARK
  447                  ;
  448 71F7  20 C1 7C        JSR DCMP          ; EA-SA (TMPC+2-TMP0) DIFF IN LOC DIFF,+1
  449 71FA  98              TYA               ; MS BYTE OF DIFF
  450 71FB  D0 CA           BNE WH1
  451 71FD  A5 E5           LDA DIFF
  452 71FF  C9 17           CMP #23
  453 7201  B0 C4           BCS WH1           ; DIFF GT 24
  454 7203  85 FE           STA RCNT          ; INCR LAST RCNT
  455 7205  E6 FE           INC RCNT
  456 7207  A5 FE      WH1  LDA RCNT
  457 7209  20 7C 72        JSR CADD          ; ADD TO CKSUM
  458 720C  20 B1 72        JSR WRCB          ; RCD CNT IN A
  459 720F  A5 EF           LDA TMPC+1        ; SA HC
  460 7211  20 7C 72        JSR CADD
  461 7214  20 B1 72        JSR WRCB
  462 7217  A5 EE           LDA TMPC          ; SA LC
  463 7219  20 7C 72        JSR CADD
  464 721C  20 B1 72        JSR WRCB
  465                  ;
  466 721F  A0 00      WH2  LDY #0
  467 7221  B1 EE           LDA (TMPC),Y
  468                  ;
  469 7223  20 7C 72        JSR CADD          ; INC CKSUM, PRESERVES A
  470 7226  20 B1 72        JSR WRDB
  471 7229  20 97 73        JSR INCTMP        ; INC SA
  472 722C  C6 FE           DEC RCNT
  473 722E  D0 EF           BNE WH2           ; LOOP FOR UP TO 24 BYTES
  474                  ;
  475 7230  20 9E 72        JSR WRCA4         ; WRITE CKSUM
  476                  ;
  477 7233  20 C1 7C        JSR DCMP
  478 7236  B0 AA           BCS WHC           ; LOOP WHILE EA GT CR = SA
  479 7238  4C 86 7C   BCCST JMP START
  480                  ;
```

MP1 DEMON PAGE 1

```
CARD # LOC   CODE        CARD
  481                   ;
  482  723B  E6 FD      WB      INC SAVX          ; SAVX TO = NOMOS FOR ASCII SUB/R
  483  723D  A5 E4      WB1     LDA WRAP          ;IF ADDR HAS WRAPPED AROUND
  484  723F  DC F7              BNE BCCST         ;THEN TERMINATE WRITE OPERATION
  485                   ;
  486  7241  A9 04              LDA #4
  487  7243  85 EC              STA ACMD
  488  7245  2C 8A 72           JSR CRLF
  489  7248  2C 9A 72           JSR WROA          ; OUTPUT HEX ADR
  490                   ;
  491  724B  2C 77 73   WBNPF   JSR SPACE
  492  724E  A2 09              LDX #9
  493  7250  86 FE              STX TMPC          ; LOOP CNT =9
  494  7252  A1 E5              LDA (TMPC-9,X)
  495  7254  85 FF              STA TMPC2         ; BYTE TO TMPC2
  496  7256  A9 42              LDA #'B           ; WRITE B
  497  7258  DC C8              BNE WBF2
  498                   ;
  499  725A  A9 5C      WBF1    LDA #'P
  500  725C  C6 FF              ASL TMPC2
  501  725E  BC C2              BCS WBF2
  502  7260  A9 4E              LDA #'N
  503                   ;
  504  7262  2C C6 72   WBF2    JSR WRCC          ; WRITE N OR P
  505  7265  C6 FE              DEC TMPC
  506  7267  DC F1              BNE WBF1          ; LOOP
  507  7269  A9 46              LDA #'F
  508  726B  2C C6 72           JSR WRCC          ; WRITE F
  509                   ;
  510  726E  2C 97 73           JSR INCTMP
  511                   ;
  512  7271  C6 EC              DEC ACMD          ; TEST FOR MULTIPLE OF FOUR
  513  7273  DC D6              BNE WBNPF
  514                   ;
  515  7275  2C C1 7C           JSR COMP
  516  7278  BC C3              BCS WB1           ; LOOP WHILE EA GT OR = SA
  517  727A  9C BC              BCC BCCST
  518                   ;
  519  727C  48         DADD    PHA               ; SAVE A
  520  727D  18                 CLC
  521  727E  65 F2              ADC TMP4
  522  7280  85 F2              STA TMP4
  523  7282  A5 F3              LDA TMP4+1
  524  7284  69 00              ADC #0
  525  7286  85 F3              STA TMP4+1
  526  7288  68                 PLA               ; RESTORE A
  527  7289  60                 RTS
  528                   ;
  529  728A  A2 0D      CRLF    LDX #$0D
  530  728C  A9 0A              LDA #$0A
  531  728E  2C CC 72           JSR WRTWO
  532  7291  AE E3              LDX CRDLY         ;BIT-TIME COUNT FOR DELAY
```

```
CARD # LOC   CODE      CARD
  533  7293  20 1D 73  CR1    JSR DLY2          ;DELAY OF ONE BIT-TIME
  534  7296  CA               DEX
  535  7297  DO FA            BNE CR1
  536  7299  60               RTS
  537                    ;
  538                    ;  WRITE ADR FROM TMPC STORES
  539                    ;
  540  729A  A2 01      WRCA   LDX #1
  541  729C  DO CA             BNE WRCA1
  542  729E  A2 05      WRCA4  LDX #5
  543  72A0  DO C6             BNE WRCA1
  544  72A2  A2 07      WRCA6  LDX #7
  545  72A4  DO C2             BNE WRCA1
  546  72A6  A2 09      WRPC   LDX #9
  547  72A8  B5 ED      WRCA1  LDA TMPC-1,X
  548  72AA  48                PHA
  549  72AB  B5 EE             LDA TMPC,X
  550  72AD  20 B1 72          JSR WRCB
  551  72B0  68                PLA
  552                    ;
  553                    ;  WRITE BYTE - A = BYTE
  554                    ;  UNPACK BYTE DATA INTO TWO ASCII CHARS. A=BYTE; X,A=CHARS
  555                    ;
  556  72B1  48        WRCB   PHA
  557  72B2  4A               LSR A
  558  72B3  4A               LSR A
  559  72B4  4A               LSR A
  560  72B5  4A               LSR A
  561  72B6  20 58 73         JSR ASCII         ; CONVERT TO ASCII
  562  72B9  AA               TAX
  563  72BA  68               PLA
  564  72BB  29 0F            AND #$0F
  565  72BD  20 58 73         JSR ASCII
  566                    ;
  567                    ;  WRITE 2 CHARS - X,A = CHARS
  568                    ;
  569  72C0  48        WRTWO  PHA
  570  72C1  8A               TXA
  571  72C2  20 C6 72         JSR WRT
  572  72C5  68               PLA
  573                    ;
  574                    ;  WRITE SERIAL OUTPUT
  575                    ;  A = CHAR TO BE OUTPUT
  576                    ;
  577  72C6  20 1D 73  WRT    JSR DLY2
  578  72C9  A2 09            LDX #9
  579                    WROC   =WRT
  580  72CB  49 FF            EOR #$FF          ; COMPLEMENT A
  581  72CD  38               SEC
  582                    ;
  583  72CE  20 DA 72  WRT1   JSR OUT
  584  72D1  20 1D 73         JSR DLY2
```

```
CARD # LOC  CODE      CARD
  585  72D4  4A            LSR A
  586  72D5  CA            DEX
  587  72D6  DC F6 -        BNE WRT1
  588  72D8  FC 3F          BEQ RDT5
  589                                        ; ALSE BNE?
  590                  ;
  591  72DA  48        CLT  PHA              ; SAVE A
  592  72DB  AD C2 6E       LDA MPB          ; OUTPUT BIT FROM CY
  593  72DE  29 FD          AND #%11111101
  594  72EC  9C C2          BCC CLT1
  595  72E2  09 C2          ORA #%CCCCCC1C
  596  72E4  8D C2 6F  CLT1 STA MPB
  597  72E7  68             PLA              ; RESTORE A
  598  72E8  6C             RTS
  599                  ;
  600                  ;  OUTPUT  RETURNS CHAR IN A
  601                  ;
  602  72E9  A5 E7     RDT  LDA HSPTR        ; TEST HS PTR OPTION
  603  72EB  4A             LSR A
  604  72EC  BC 4F          BCS RDHSR
  605                  RDOC =RDT
  606  72EE  A2 08          LDX #8
  607                  ;
  608  72FC  AD C2 6E  RDT1 LDA MPB
  609  72F3  4A             LSR A            ; WAIT FOR START BIT
  610  72F4  9C FA          BCC RDT1
  611                  ;
  612  72F6  2C 2C 73       JSR DLY1
  613  72F9  2C DA 72       JSR CLT          ; ECHO START BIT
  614                  ;
  615  72FC  2C 1C 73  RDT2 JSR DLY2
  616  72FF  AD C2 6E       LDA MPB          ; CY = NEXT BIT
  617  73C2  4A             LSR A
  618  73C3  2C DA 72       JSR CLT          ; ECHO
  619                  ;
  620  73C6  C8             PHP              ; SAVE BIT
  621  73C7  98             TYA              ; Y CONTAINS CHAR BEING FORMED
  622  73C8  4A             LSR A
  623  73C9  28             PLP              ; RECALL BIT
  624  73CA  9C C2          BCC RDT4
  625  73CC  C9 80          ORA #$8C         ; ADD IN NEXT BIT
  626  73CE  A8        RDT4 TAY
  627  73CF  CA             DEX
  628  731C  DC EA          BNE RDT2         ; LOOP FOR 8 BITS
  629  7312  49 FF          EOR #$FF         ; COMPLEMENT DATA
  630  7314  29 7F          AND #$7F         ; CLEAR PARITY
  631                  ;
  632  7316  2C 1C 73       JSR DLY2
  633  7319  18        RDT5 CLC
  634  731A  2C DA 72       JSR CLT          ;AND DELAY 2 HALF-BIT-TIMES
  635                  ;
  636  731C  2C 2C 73  DLY2 JSR DLY1
```

```
CARD # LOC    CODE          CARD
       7320   48            DLY1    PHA               ; SAVE FLAGS AND A
       7321   08                    PHP
       7322   8A                    TXA               ; SAVE X
       7323   48                    PHA
       7324   A6 EA                 LDX MAJCRT
       7326   A5 EB                 LDA MINCRT
                            ;
       7328   8D C4 6E      DL2     STA MCLKIT
                            ;
       732B   AD C5 6E      DL3     LDA MCLKIF
       732E   10 FB                 BPL DL3
       7330   CA                    DEX
       7331   08                    PHP
       7332   AD C4 6E              LDA MCLKRD        ; RESET TIMER INT FLAG
       7335   28                    PLP
       7336   10 F3                 BPL DL3
                            ;
       7338   68                    PLA               ; RESTORE REGS
       7339   AA                    TAX
       733A   28                    PLP
       733B   68                    PLA
       733C   60            DLX     RTS
                            ;
       733D   AD C2 6E      RDHSR   LDA MPB           ; LOOP ON DATA AVAIL
       7340   29 08                 AND #DAVAIL
       7342   F0 F9                 BEQ RDHSR
                            ;
       7344   AE 00 6E              LDX MPA           ; READ DATA
       7347   AD C2 6E              LDA MPB           ; SEND OUT-DATA PULSE
       734A   09 04                 ORA #OUTDAT
       734C   8D C2 6E              STA MPB
       734F   29 FB                 AND #%11111011
       7351   8D C2 6E              STA MPB
       7354   8A                    TXA
       7355   29 7F                 AND #$7F
       7357   60                    RTS
                            ;
       7358   18            ASCII   CLC
       7359   69 06                 ADC #6
       735B   69 F0                 ADC #$F0
       735D   90 02                 BCC ASC1
       735F   69 06                 ADC #$06
                            ;
       7361   69 3A         ASC1    ADC #$3A
       7363   48                    PHA               ; TEST FOR LETTER B IN ADR DURING WBNPF
       7364   C9 42                 CMP #'B
       7366   D0 0A                 BNE ASCX
       7368   A5 FD                 LDA SAVX
       736A   C9 07                 CMP #NCMDS
       736C   D0 04                 BNE ASCX          ; NOT WB CMD
       736E   68                    PLA
       736F   A9 20                 LDA #'            ; FOR WB, BLANK B'S IN ADR
```

```
CARD # LOC   CODE      CARD
  689  7371  48            PHA
  690  7372  EE        ASCX  PLA
  691  7373  60            RTS
  692                 ;
  693  7374  20 77 73  SPAC2 JSR SPACE
  694  7377  48        SPACE PHA                 ; SAVE A,X,Y
  695  7378  8A            TXA
  696  7379  48            PHA
  697  737A  98            TYA
  698  737B  48            PHA
  699  737C  A9 2C         LDA #'
  700  737E  20 C6 72      JSR WRT               ; TYPE SP
  701  7381  68            PLA                   ; RESTORE A,X,Y
  702  7382  A8            TAY
  703  7383  68            PLA
  704  7384  AA            TAX
  705  7385  68            PLA
  706  7386  60            RTS
  707                 ;
  708  7387  A2 C2   T2T2  LDX #2
  709  7389  B5 ED   T2T21 LDA TMPC-1,X
  710  738B  48            PHA
  711  738C  B5 EF         LDA TMP2-1,X
  712  738E  95 ED         STA TMPC-1,X
  713  7390  68            PLA
  714  7391  95 EF         STA TMP2-1,X
  715  7393  CA            DEX
  716  7394  D0 F3         BNE T2T21
  717  7396  60            RTS
  718                 ;
  719                 ;INCREMENT (TMPC,TMPC+1) BY 1
  720  7397  E6 EE   INCTMP INC TMPO             ;LOW BYTE
  721  7399  F0 C1         BEQ INCT1
  722  739B  60            RTS
  723                 ;
  724  739C  E6 EF   INCT1 INC TMPC+1            ;HIGH BYTE
  725  739E  F0 C1         BEQ SETWRP
  726  73A0  60            RTS
  727                 ;
  728  73A1  E6 E4   SETWRP INC WRAP             ;POINTER HAS WRAPPED AROUND - SET FLAG
  729  73A3  60            RTS
  730                 ;
  731                 ; READ HEX ADR, RETURN HO IN TMPC, LO IN TMPO+1 AND CY=1
  732                 ;     IF SP CY=0
  733                 ;
  734  73A4  20 B3 73  RDOA  JSR RDOB             ; READ 2 CHAR BYTE
  735  73A7  90 C2           BCC RDOA2           ; SPACE
  736                 ;
  737  73A9  85 EF           STA TMPO+1
  738  73AB  20 B3 73  RDOA2 JSR RDOB
  739  73AE  90 C2           BCC RDEXIT          ; SP
  740  73B0  85 EE           STA TMPC
```

```
CARD # LOC  CODE      CARD
  741  73B2  6C       RDEXIT RTS
  742                 ;
  743                 ;  READ HEX BYTE AND RETURN IN A, AND CY=1
  744                 ;     IF SP CY=0
  745                 ;     Y REG IS PRESERVED
  746                 ;
  747  73B3  98       RDDE   TYA              ; SAVE Y
  748  73B4  48              PHA
  749  73B5  A9 00           LDA #0           ; SET DATA = 0
  750  73B7  85 EC           STA ACMD
  751  73B9  20 E9 72        JSR RDDC
  752  73BC  C9 0D           CMP #$0D         ; CR?
  753  73BE  D0 06           BNE RDDB1
  754  73C0  68              PLA              ;YES - GO TO START
  755  73C1  68              PLA              ;CLEANING STACK UP FIRST
  756  73C2  68              PLA
  757  73C3  4C 86 70        JMP START
  758                 ;
  759  73C6  C9 20    RDDB1  CMP #' '         ; SPACE
  760  73C8  D0 0A           BNE RDDB2
  761  73CA  20 E9 72        JSR RDDC         ; READ NEXT CHAR
  762  73CD  C9 20           CMP #' '
  763  73CF  D0 0F           BNE RDDB3
  764  73D1  18              CLC              ; CY=0
  765  73D2  90 12           BCC RDDB4
  766                 ;
  767  73D4  20 EB 73 RDDB2  JSR HEXIT        ; TO HEX
  768  73D7  0A              ASL A
  769  73D8  0A              ASL A
  770  73D9  0A              ASL A
  771  73DA  0A              ASL A
  772  73DB  85 EC           STA ACMD
  773  73DD  20 E9 72        JSR RDDC         ; 2ND CHAR ASSUMED HEX
  774  73E0  20 EB 73 RDDB3  JSR HEXIT
  775  73E3  05 EC           ORA ACMD
  776  73E5  38              SEC              ; CY=1
  777  73E6  AA       RDDB4  TAX
  778  73E7  68              PLA              ; RESTORE Y
  779  73E8  A8              TAY
  780  73E9  8A              TXA              ;SET Z & N FLAGS FOR RETURN
  781  73EA  6C              RTS
  782                 ;
  783  73EB  C9 3A    HEXIT  CMP #$3A
  784  73ED  08              PHP              ; SAVE FLAGS
  785  73EE  29 0F           AND #$0F
  786  73F0  28              PLP
  787  73F1  90 02           BCC HEXO9        ; 0-9
  788  73F3  69 08           ADC #8           ; ALPHA ADD 8+CY=9
  789  73F5  6C       HEXO9  RTS
  790                 ;
  791  73F6                  *=MP3+$F8
  792                 ;
```

```
CARD #  LOC    CODE      CARD
   793  73F8  CC 7C    INTVEC .WORD NMINT       ; DEFAULT USER IRQ TO NMINT
   794  73FA  CC 7C           .WORD NMINT
   795  73FC  C6 7C           .WORD RESET
   796  73FE  52 7C           .WORD INTRQ
   797                ;

END OF MOS/TECHNOLOGY 6501 ASSEMBLY VERSION 3
NUMBER OF ERRORS =    0,  NUMBER OF WARNINGS =    0
```

SYMBOL TABLE

SYMBOL	VALUE	LINE DEFINED	CROSS-REFERENCES
ACC	C0F9	121	141 195 228 377
ACMD	C0EC	113	247 249 254 487 512 750 772 775
ADRS	71CD	317	248
ALTER	713A	352	317
ASCII	7358	674	561 565
ASCX	7372	69C	683 686
ASC1	7361	68C	677
A2	714E	358	356
A3	714B	36C	353
A4	715C	363	359
A5	7152	364	366
A9	715A	367	
BCCST	7238	479	437 484 517
BEQS1	7134	346	367 400
BX	7081	228	199
BYTE	70EC	286	265 411
BY2	70F2	298	293
BY3	70F5	299	287
B3	7C5B	2C2	143
B5	7C73	22C	
CMDS	71C6	31C	24C
CRDLY	C0F3	1C5	164 532
CRLF	728A	529	22C 234 386 439 488
CR1	7293	533	535
DADD	727C	519	298 4C3 406 409 457 460 463 469
DAVAIL	C00E	8E	661
DCMP	70C1	263	418 448 477 515
DIFF	C0F5	1C7	26E 27C 451
DLX	733C	658	
DLY1	732C	637	612 636
DLY2	731C	636	533 577 584 615 632
DL2	7328	644	
DL3	732B	64E	647 652
DSPLYM	711C	335	319
DSPLYR	7114	331	318
ERRCPR	70BA	259	29E 348 42C
ERRP1	71EF	42C	
ERRS1	7137	348	336
FLGS	C0F8	12C	21C 3C3 375
GC	715C	369	32C
GOTDAT	CCC4	87	666
HEXIT	73EB	783	767 774
HEXC9	73F5	789	787
HSP	716F	382	321
HSPTR	C0E7	1C8	15E 232 388 399 6C2
HSRCP	C0E8	109	157 382 387
IJMP	7CB4	254	251
INCTMP	7397	72C	299 471 51C
INCT1	739C	724	721
INTRC	7C52	195	79E
INTVEC	73F8	793	150

SYMBOL	VALUE	LINE DEFINED	CROSS REFERENCES								
ICBASE	6ECC	88	89	90	91	92	93	94	95		
LCNT	C0FF	125									
LF	7174	385	322								
LH1	717E	389	391	419							
LH2	7195	402	396								
LH3	71AA	411	412								
MAJORT	CCEA	111	155	172	185	641					
MCLKIF	6EC5	95	170	646							
MCLKRC	6EC4	94	182	650							
MCLKIT	6EC4	93	169	644							
MDA	6EC1	90									
MDB	6EC3	92	147								
MDBK	CC16	85	145								
MINORT	00EF	112	190	642							
MPA	6ECC	89	664								
MPB	6EC2	91	165	176	592	596	608	616	660	665	667	669
MPC	7CC0	98	139								
MP1	71CC	99	246	317	318	319	320	321	322	323	
MP2	72CC	100									
MP3	73CC	101	791								
MC	7123	338	333								
M1	7127	340	345								
NCMDS	CCC7	97	239	685							
NMINT	7CCC	141	793	794							
OUT	72DA	591	583	613	618	634					
OUT1	72E4	596	594								
PCH	C0F7	119	216	276	371						
PCL	C0F6	118	213	274	373						
PREVC	C0E5	110	244	352							
PUTP	7CDC	273	357								
RAM64	FFC0	133	134								
RCNT	C0FE	128	300	363	402	441	454	455	456	472	
RDEXIT	73B2	741	739								
RDFSR	733C	660	604	662							
RDOA	73A4	734	335	355	413	425	428				
RDOA2	73AB	738	735								
RDOB	73B3	747	286	395	4C4	4C7	734	738			
RDOB1	73C6	759	753								
RDOB2	73D4	767	760								
RDOB3	73EC	774	763								
RDOB4	73E6	777	765								
RDOC	72E9	605	237	385	389	422	430	751	761	773	
RDT	72F9	602	605								
RDT1	72FC	608	610								
RDT2	72FC	615	628								
RDT4	73CE	626	624								
RDT5	7319	633	588								
RESET	7CC6	145	795								
RC	7C22	165	167								
R1	7CCD	150	153								
R2	7C28	169	173								
R3	7C2B	170	178	180							
R4	7C34	175	171								
R5	7C44	184	189								

SYMBOL	VALUE	LINE DEFINED	CROSS-REFERENCES								
F6	704E	188	186								
SAVX	CCFC	125	243	245	482	684					
SETR	7CFB	303	332	358							
SETWRP	73A1	728	725								
SP	CCFC	124	218	369							
SPACE	7377	694	340	364	424	427	491	693			
SPAC2	7374	693	252								
START	7C86	231	261	346	383	479	757				
SC	7C97	239	226								
S1	7C99	240	257								
S2	7CB7	256	241								
TMPC	CCFE	126	128	202	221	338	344	423	431	493	505
TMPC2	CCFF	127	129	495	500						
TMPC	CCEE	114	265	268	273	275	280	281	290	292	304
			341	405	408	459	462	467	494	547	549
			712	720	724	737	740				
TMP2	CCFC	115	264	267	415	417	711	714			
TMP4	CCF2	116	414	416	521	522	523	525			
TMP6	CCF4	117									
T2T2	7387	708	426	429							
T2T21	7389	709	716								
UINT	FFF8	96	151	229							
WB	723B	482	434								
WBF1	725A	499	506								
WBF2	7262	504	497	501							
WBNFF	724B	491	513								
WB1	723D	483	516								
WFC	71E2	436	478								
WF1	72C7	456	450	453							
WF2	721F	466	473								
WC	71C2	422	323								
WRAP	CCE4	106	233	436	483	728					
WRCA	729A	540	360	489							
WRCA1	72A8	547	541	543	545						
WRCA4	729E	542	475								
WRCA6	72A2	544									
WRCB	72B1	556	342	458	461	464	470	550			
WRCC	72C6	579	236	260	446	504	508				
WRPC	72A6	546	331								
WRT	72C6	577	571	579	700						
WRTWO	72CC	569	224	531							
WRT1	72CE	583	587								
XR	CCFA	122	207	378							
YR	CCFB	123	208	379							
ZTMP	70D9	279	394	443							

```
CARD # LOC   CODE      CARD
     1                ;MEMORY ADDRESS TEST
     2                ;FOR EACH LOC IN TEST RANGE
     3                       ;CLEAR WHOLE RANGE
     4                ;      SET LOC TO $FF
     5                ;      VERIFY WHOLE RAGE $00 EXCEPT (LOC)
     6                ;      VERIFY (LOC) TO BE $FF
     7                ;BREAK TO MONITOR ON ERROR WITH LOC IN (C,1)
     8                ;PRINT "." ON COMPLETION OF PASS & REPEAT
     9                ;
    10  00C0                *=$00C0           ;PAGE C
    11                ;
    12                WRT   =$72C2
    13  0000          LOC   *=*+2             ;TEST CELL ADDR
    14  0002          LOW   *=*+2             ;LOWER LIMIT OF TEST
    15  0004          HIGH  *=*+2             ;UPPER LIMIT OF TEST+1
    16  0006          PTR   *=*+2             ;POINTER TO CELL UNDER TEST
    17                ;
    18  0008                *=$0010           ;START ADDR
    19                ;
    20  0010  A9 0D     MAD   LDA #$0D          ;TYPE CR
    21  0012  20 C2 72        JSR WRT
    22  0015  A9 0A           LDA #$0A          ;& LF
    23  0017  20 C2 72        JSR WRT
    24                ;
    25  001A  20 68 00        JSR RSTLOC        ;LOC=LOW

    26  001D  20 71 00        JSR RSTPTR        ;PTR=LOW

    27  0020  A2 00           LDX #0
    28                ;
    29                ;CLEAR MEMORY AREA UNDER TEST
    30  0022  A9 00     ML1   LDA #0
    31  0024  81 06           STA (PTR,X)       ;STORE ZERO
    32  0026  20 7A 00        JSR INCPTR        ;INCREMENT & TEST

    33  0029  D0 F7           BNE ML1           ;NEXT LOC
    34                ;
    35                ;PUT $FF IN SELEXTED CELL
    36  002B  A9 FF     TEST  LDA #$FF
    37  002D  81 00           STA (LOC,X)
    38                ;VERIFY ALL CELLS ZERO EXCEPT (LOC)
    39  002F  20 71 00        JSR RSTPTR        ;PTR=LOW

    40                ;
    41  0032  A1 06     VLOOP LDA (PTR,X)       ;GET CELL
    42  0034  F0 17           BEQ NEXTC         ;OK IF ZERO
    43  0036  A4 06           LDY PTR           ;NOT ZERO--IS THIS (LOC)?
    44  0038  C4 00           CPY LOC
    45  003A  F0 01           BEQ OK1
    46  003C  00              BRK               ;NOT (LOC)
    47                ;
    48  003D  A4 07     OK1   LDY PTR+1
```

```
CARD #  LOC   CODE        CARD
   49   003F  C4 01             CPY LOC+1
   50   0041  F0 C1             BEQ CK2
   51   0043  00                BRK                 ;NOT (LOC)
   52                    ;
   53   0044  C9 FF      CK2     CMP #$FF            ;IS (LOC)--IS DATA OK?
   54   0046  F0 01              BEQ OK3
   55   0048  00                BRK                 ;WRONG DATA
   56                    ;
   57   0049  A9 00      OK3     LDA #0              ;RESET (LOC)
   58   004B  81 00              STA (LOC,X)
   59                    ;
   60   004D  20 7A 00   NEXTC   JSR INCPTR          ;NEXT CELL

   61   0050  D0 E0              BNE VLOOP           ;IF NOT AT LIMIT
   62                    ;
   63   0052  A5 00              LDA LOC             ;PRINT STAR EVERY PAGE BOUNDARY
   64   0054  D0 07              BNE NOSTAR
   65   0056  A9 2A              LDA #'*
   66   0058  20 C2 72           JSR WRT
   67   005B  A2 00              LDX #0              ;FIX X AFTER MON CALL
   68                    ;
   69   005D  20 8B 00   NOSTAR  JSR INCLOC          ;NEXT LOC

   70   0060  D0 C9              BNE TEST
   71                    ;
   72   0062  20 68 00           JSR RSTLOC          ;PASS COMPLETE

   73   0065  4C 10 00           JMP MAD             ;NEXT PASS
   74                    ;
   75                    ;RESET LOC TO LOW
   76   0068  A5 02      RSTLOC  LDA LOW
   77   006A  85 00              STA LOC
   78   006C  A5 03              LDA LOW+1
   79   006E  85 01              STA LOC+1
   80   0070  60                 RTS
   81                    ;
   82                    ;RESET PTR TO LOW
   83   0071  A5 02      RSTPTR  LDA LOW
   84   0073  85 06              STA PTR
   85   0075  A5 03              LDA LOW+1
   86   0077  85 07              STA PTR+1
   87   0079  60                 RTS
   88                    ;
   89                    ;INCREMENT PTR & CHECK FOR LIMIT
   90   007A  E6 06      INCPTR  INC PTR             ;INCREMENT
   91   007C  D0 02              BNE INC1
   92                    ;
   93   007E  E6 07              INC PTR+1
   94                    ;
   95   0080  A5 04      INC1    LDA HIGH            ;CHECK
   96   0082  C5 06              CMP PTR
   97   0084  D0 04              BNE IPRET           ;NOT AT LIMIT
```

```
CARD # LOC   CODE      CARD
   98                ;
   99  0086  A5 05          LDA HIGH+1
  100  0088  C5 07          CMP PTR+1          ;Z=1 IF AT LIMIT
  101                ;
  102  008A  60       IPRET  RTS
  103                ;
  104                ;INCREMENT LOC & CHECK FOR LIMIT
  105  008B  E6 00    INCLOC INC LOC            ;INCR
  106  008D  D0 02          BNE INC2
  107                ;
  108  008F  E6 01          INC LOC+1
  109                ;
  110  0091  A5 04    INC2   LDA HIGH           ;CHECK
  111  0093  C5 00          CMP LOC
  112  0095  D0 04          BNE ILRET
  113  0097  A5 05          LDA HIGH+1
  114  0099  C5 01          CMP LOC+1          ;Z=1 IF AT LIMIT
  115                ;
  116  009B  60       ILRET  RTS
```

```
END OF MOS/TECHNOLOGY 6501 ASSEMBLY VERSION 3
NUMBER OF ERRORS =   0,  NUMBER OF WARNINGS =   0
```

SYMBOL TABLE

SYMBOL	VALUE	LINE DEFINED	CROSS-REFERENCES
HIGH	0004	15	95 99 110 113
ILRET	009B	116	112
INCLOC	008B	105	69
INCPTR	007A	90	32 60
INC1	008C	95	91
INC2	0091	110	106
IPRET	008A	102	97
LOC	0000	13	37 44 49 58 63 77 79 105 108 111 114
LOW	0002	14	76 78 83 85
MAC	001C	20	73
ML1	0022	30	33
NEXTC	004D	60	42
NOSTAR	005D	69	64
OK1	003D	48	45
OK2	0044	53	50
OK3	0049	57	54
PTR	0006	16	31 41 43 48 84 86 90 93 96 100
RSTLOC	0066	76	25 72
RSTPTR	0071	83	26 39
TEST	002B	36	70
VLOOP	0032	41	61
WRT	72C2	12	21 23 66