GT40

ROM BOOTSTRAP

MD-11-DDGTD-C

EP-DDGTD-C-DL-B

FICHE 1 OF 1

DEC 1976

digital

MADE IN USA

```
1                       .REM    *
```

```
.TITLE  GT40 ROM BOOTSTRAP TEST MAINDEC-11-DDGTD-C
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DDGTD-C-D
PRODUCT NAME: GT40 ROM VERIFY
DATE: DECEMBER 1976
MAINTAINER: DIAGNOSTIC GROUP

1. ABSTRACT

THIS VERSION OF THE PROGRAM SUPPORTS NON-SWITCH REGISTER CPU'S.
FOR THESE CPU'S, THE SWITCH REGISTER CAN BE CHANGED BY CHANGING
THE CONTENTS OF SWREG (170).

THE DDGTD-C DIAGNOSTIC PROGRAM IS WRITTEN TO BE USED AS AN AID
TO HARDWARE DEBUGGING AND MAINTENANCE OF THE GT40 ROM
BOOTSTRAP LOADERVERSION 1 OR 2.

THE AVAILABLE TESTS ARE
PRG0 - LOGIC TESTS
PRG1 - ROM DATA DUMP TO THE CONSOLE TELETYPE

2. REQUIREMENTS

2.1 EQUIPMENT

GT40 DISPLAY PROCESSOR WITH ROM BOOTSTRAPVERSION 1 OR 2.

2.2 STORAGE

THIS PROGRAM USES MEMORY LOCATIONS 0-7776 + 16000-16776(8).

3. LOADING PROCEDURE

PROCEDURE FOR A NORMAL BINARY TAPE SHOULD BE FOLLOWED.

4. STARTING PROCEDURE

4.1 STARTING ADDRESSES

0200 PROGRAM 0, ROM LOGIC TEST.
0204 PROGRAM 1, ROM DATA DUMP ON CONSOLE TTY.

4.2 SWITCH SETTINGS

CONSOLE SW 11=0	NORMAL RUN (64. INTERATIONS/TEST)
CONSOLE SW 11=1	SUPPRES SUBPROGRAM INTERATIONS
CONSOLE SW 08=0	TEST AS VERSION 2 ROM (512. WORDS)
CONSOLE SW 08=1	TEST AS VERSION 1 ROM (256. WORDS)

5. PROGRAM DESCRIPTIONS

5.1 PRGO - LOGIC TESTS

THE LOGIC TESTS CONSIST OF 4 ROUTINES TO TEST THE GT40 ROM BOOTSTRAP LOGIC

5.1.1 ROUTINE DESCRIPTIONS

ROUTINE	TESTS
T1	ADDRESSABILITY OF GT40 ROM BOOTSTRAP
T2	DATA RELIABILITY
T3	THAT GT40 ROM BOOTSTRAP TIMES OUT WHEN REFERENCED BY A DATIP BUS CYCLE
T4	THAT DATA READ FROM THE ROM IS CORRECT

5.2 PRG1 - ROM DATA DUMP

THIS PROGRAM TYPES OUT THE 512./256. WORDS OF ROM DATA ON THE CONSOLE TELETYPE AND HALTS.

6. ERRORS

THE PROGRAM WILL ONLY HALT ON ERROR. THE PROGRAM DOES NOT CONTAIN FACILITIES FOR REPORTING ERROR CONDITIONS.
TO PLACE THE PROGRAM INTO A SCOPE LOOP, REPLACE THE ERROR HALT WITH A NOP.

7. EXECUTION TIME

PRGO TAKES APPROX. 5 SECONDS PER PASS.
PROG1 N/A
PROG2 N/A

```
                                        .LIST ME,BIN,SEQ,LD
                                        .ENABL  ABS,AMA

                                ;LOAD ADDRESS=0200
                                ;DEPRESS START
                                ;STACk POINTER IS AT 500

                                        .LIST
                    000034              .=34
000034  002024                          SCOPEC
000036  000000                          0
                                ;EQUATE STATEMENTS
                    104400              SCOPE=TRAP
                    177564              TPCSR=177564
                    177566              TPDBR=177566
                    177776              PSW=177776
                    177570              DSWR=177570             ;ADDRESS OF SWITCH REGISTER
                    000004              ERRVEC=4
                    000500              STKPTR=500              ;INITIAL STACK SETTING
                    000170              .=170
000170  000000              SWREG:      .WORD 0
000172  177570              SWR:        DSWR
                    000200              .=200
000200  000137  001024                  JMP     PRMTRS
000204  000137  001500                  JMP     PRG1
```

```
                    001000                          .=1000
001000  166000      ROMADD: 166000                  ;ROM ADDRESS
001002  001000      WORDS:  512.    ;256.           ;ROM LENGTH
001004  006000      IMAGE:  START   ;STARTA         ;ROM IMAGE
001006  172002      DSR:    172002                  ;DISPLAY STATUS REGISTER
001010  000010      FILLER: 10                      ;# OF FILLER CHAR
001012  000010      FILCNT: 10
001014  000000      ICNT:   0
001016  000000      DUMP:   0
001020  000000      CHARA:  0
001022  000000      TERM:   0
001024  012706  000500      PRMTRS: MOV     #STKPTR,%6      ;SET STACK PTR
001030  004737  002256              JSR     PC,SWITCH       ;CHECK ROM VERSION

                    ;PROGRAM 0 LOGIC TESTS

001034  005037  001014      PRG0:   CLR     ICNT            ;CLEAR PASS COUNT
001040  012706  000500      PRG0R:  MOV     #STKPTR,%6
001044  012737  001040  002100          MOV     #PRG0R,RETURN   ;SET RETURN ADDRESS FOR SCOPE

                    ;TEST1 TEST ABILITY TO REFERENCE ROM WITHOUT TIMING OUT

001052  013700  001000      T1:     MOV     ROMADD,%0       ;GET ROM ADDRESS
001056  013701  001002              MOV     WORDS,%1        ;GET ADDRESS COUNTER
001062  012737  001122  000004          MOV     #ERROR1,4       ;SET UP TIME OUT VECTOR
001070  011003              T1A:    MOV     (0),%3          ;REFERENCE
001072  005720                      TST     (0)+            ;ROM
001074  064037  001016              ADD     -(0),DUMP       ;
001100  021010                      CMP     (0),(0)         ;
001102  132020                      BITB    (0)+,(0)+       ;
001104  164037  001016              SUB     -(0),DUMP
001110  062700  000002              ADD     #2,%0           ;INCREMENT POINTER
001114  005301                      DEC     %1              ;DECREMENT ADDRESS COUNTER
001116  001364                      BNE     T1A             ;BRANCH IF NOT FINISHED
001120  000403                      BR      T1B             ;GO TO SCOPE LOOP
001122  022626      ERROR1: CMP     (6)+,(6)+       ;REPOSITION STACK
001124  000000              HALT                    ;ERROR, TIME-OUT  ON ROM ADDRESS
001126  000760              BR      T1A             ;LOOP ON ERROR
001130  104400      T1B:    SCOPE
```

```
                                        ;TEST2  TEST THAT ROM DATA CAN BE READ RELIABLY.

001132  013700  001000        T2:     MOV     ROMADD,%0       ;GET ROM ADDRESS
001136  013701  001002                MOV     WORDS,%1        ;GET ADDRESS COUNTER
001142  012737  000006  000004        MOV     #6,4            ;INITIALIZE TIME OUT VECTOR
001150  005037  001016        T2A:    CLR     DUMP            ;INITIALIZE DUMP
001154  011003                        MOV     (0),%3          ;GET DATA
001156  062037  001016                ADD     (0)+,DUMP       ;ADD DATA TO DUMP
001162  163703  001016                SUB     DUMP,%3         ;SUBTRACT DATA FROM DATA
001166  001402                        BEQ     T2B             ;BRANCH IF EQUAL
001170  000000        ERROR2: HALT                            ;DATA ERROR
001172  000766                        BR      T2A             ;LOOP ON ERROR
001174  044037  001016        T2B:    BIC     -(0),DUMP       ;CLEAR DUMP BITS
001200  001402                        BEQ     T2C             ;BRANCH IF EQUAL TO C
001202  000000                        HALT                    ;DATA ERROR
001204  000773                        BR      T2B             ;LOOP ON ERROR
001206  021010        T2C:    CMP     (0),(0)                 ;COMPARE DATA
001210  001402                        BEQ     T2D             ;BRANCH IF EQUAL
001212  000000                        HALT                    ;DATA ERROR
001214  000774                        BR      T2C             ;LOOP ON ERROR
001216  122040        T2D:    CMPB    (0)+,-(0)               ;COMPARE DATA (BYTE OPERATION)
001220  001402                        BEQ     T2E             ;BRANCH IF EQJAL
001222  000000                        HALT                    ;DATA ERROR
001224  000774                        BR      T2D             ;LOOP ON ERROR
001226  005720        T2E:    TST     (0)+                    ;INCREMENT ADDRESS POINTER
001230  005301                        DEC     %1              ;DECREMENT ADDRESS COUNTER
001232  001346                        BNE     T2A             ;RETURN IF NOT DONE
001234  104400                        SCOPE
```

```
                                        ;TEST3 TEST THAT ROM TIMES OUT IF REFERENCED BY OTHER
                                        ;THAN DATI BUS CYCLE

001236  012706  000500          T3:     MOV     #STKPTR,%6      ;SET STACK PTR
001242  013700  001000                  MOV     ROMADD,%0       ;GET ROM ADDRESS
001246  013701  001002                  MOV     WORDS,%1        ;GET ADDRESS COUNTER
001252  012737  001266  000004  T3AA:   MOV     #T3B,4          ;SET UP TIME OUT VECTOR
001260  010010          T3A:    MOV     %0,(0)          ;ATTEMPT TO ALTER DATA
001262  000000                  HALT                    ;HERE IF DID NOT TIME OUT
001264  000775                  BR      T3A             ;LOOP ON ERROR
001266  012737  001304  000004  T3B:    MOV     #T3D,4          ;SET UP TIME OUT VECTOR
001274  022626                  CMP     (6)+,(6)+       ;REPOSITION STACK
001276  005210          T3C:    INC     (0)             ;ATTEMPT TO ALTER DATA
001300  000000                  HALT                    ;HERE IF DID NOT TIME OUT
001302  000775                  BR      T3C             ;LOOP ON ERROR
001304  012737  001324  000004  T3D:    MOV     #T3F,4          ;SET UP TIME OUT VECTOR
001312  022626                  CMP     (6)+,(6)+       ;REPOSITION STACK
001314  005077  177460          T3E:    CLR     @ROMADD         ;ATTEMPT TO ALTER DATA
001320  000000                  HALT                    ;HERE IF DID NOT TIME OUT
001322  000774                  BR      T3E             ;LOOP ON ERROR
001324  005720          T3F:    TST     (0)+            ;INCREMENT ADDRESS POINTER
001326  022626                  CMP     (6)+,(6)+       ;REPOSITION STACK
001330  005301                  DEC     %1              ;DECREMENT ADDRESS COUNTER
001332  001347                  BNE     T3AA            ;RETURN IF NOT DONE
001334  012737  000006  000004  MOV     #6,@#4          ;RESTORE TIME OUT TRAP
001342  104400                  SCOPE                   ;SCOPE LOOP
```

```
                                        ;COMPARE THE ROM DATA TO THE IMAGE DATA
                                        ;       R0=WORD NUMBER
                                        ;       R1=GOOD ADDRESS
                                        ;       R2=GOOD DATA
                                        ;       R3=BAD ADDRESS
                                        ;       R4=BAD  DATA

001344  012700  000000          T4:     MOV     #0,%0           ;SET UP INITIAL WORD COUNT
001350  013701  001004                  MOV     IMAGE,%1        ;SET UP STARTING ADDRESS OF ROM IMAGE
001354  013703  001000                  MOV     ROMADD,%3       ;SET UP STARTING ROM ADDRESS
001360  011102                  T4A:    MOV     (%1),%2         ;READ EXPECTED VALUE
001362  011304                          MOV     (%3),%4         ;READ ROM VALUE
001364  020204                          CMP     %2,%4           ;COMPARE EXPECTED TO THE VALUE READ
001366  001402                          BEQ     T4B             ;BRANCH IF CORRECT
001370  000000                          HALT                    ;ERROR, ROM VALUE FAILED TO EQUAL EXPECTED
001372  000772                          BR      T4A

001374  022123                  T4B:    CMP     (%1)+,(%3)+     ;INCREMENT ADDRESSES POINTERS
001376  005200                          INC     %0              ;INCREMENT WORD COUNT
001400  023700  001002                  CMP     WORDS,%0        ;COMPARE IF END WORD
001404  001365                          BNE     T4A             ;BRANCH IF NOT LAST WORD
001406  104400                  T4E:    SCOPE

001410  005237  001014          END:    INC     ICNT            ;INCREMENT PASS COUNT
001414  012777  000001  177364          MOV     #1,@DSR         ;RING THE GT40 BELL
001422  012737  000207  177566  DONE0:  MOV     #207,@#TPDBR    ;RING THE TELETYPE BELL
001430  105737  177564                  TSTB    @#TPCSR
001434  100375                          BPL     .-4
001436  012737  000207  177566          MOV     #207,@#TPDBR
001444  105737  177564          1$:     TSTB    @#TPCSR
001450  100375                          BPL     1$
001452  013700  000042                  MOV     @#42,%0         ;RETURN TO DECTAPE MONITOR?
001456  001406                          BEQ     DONE1
001460  000005                          RESET
001462  000005                          RESET
001464  004710                          JSR     7,(0)           ;RETURN!
001466  000240                          NOP
001470  000240                          NOP
001472  000240                          NOP
001474  000137  001034          DONE1:  JMP     PRG0
```

```
                                        ;THIS PROGRAM TYPES OUT ROM DATA
                                        
001500  012706  000500          PRG1:   MOV     #STKPTR,SP      ;INITIALIZE STACK
001504  012737  000006  000004          MOV     #6,@#4          ;SET UP BUSS ERROR
001512  004737  002256                  JSR     PC,SWITCH
001516  004537  001662                  JSR     5,TYPEM
001522  002245                          M8
001524  004537  001662                  JSR     5,TYPEM         ;TYPE MESSAGE
001530  002222                          M7                      ;'ROM DATA'
001532  013701  001002                  MOV     WORDS,R1        ;GET # OF WORDS
001536  013700  001000          PRG1A:  MOV     ROMADD,R0       ;GET STARTING ADDRESS
001542  012702  000010                  MOV     #10,R2          ;GET ADDRESS INDICATOR
001546  105737  177564                  TSTB    TPCSR           ;WAIT FOR
001552  100375                          BPL     .-4             ;TELEPRINTER FLAG
001554  010037  002114          PRG1B:  MOV     R0,D2BTYP       ;GET ADDRESS
001560  004737  002116                  JSR     7,O2A           ;AND TYPE IT
001564  004537  001662                  JSR     5,TYPEM         ;TYPE
001570  002251                          M9                      ;CR/LF
001572  012037  002114          PRG1C:  MOV     (R0)+,D2BTYP    ;TYPE
001576  004737  002116                  JSR     7,O2A           ;DATA
001602  105737  177564                  TSTB    TPCSR           ;WAIT FOR
001606  100375                          BPL     .-4             ;TELEPRINTER FLAG
001610  012737  000040  177566          MOV     #' ,TPDBR       ;TYPE SPACE
001616  005301                          DEC     R1              ;ALL DATA TYPED
001620  001410                          BEQ     PRG1D           ;GO TO FINISH
001622  005302                          DEC     R2
001624  001362                          BNE     PRG1C           ;RETURN TO PRG1B
001626  012702  000010                  MOV     #10,R2          ;GET ADDRESS INDICATOR
001632  004537  001662                  JSR     5,TYPEM         ;TYPE
001636  002245                          M8                      ;CR/LF
001640  000745                          BR      PRG1B           ;RETURN TO PRG1B
001642  004537  001662          PRG1D:  JSR     5,TYPEM
001646  002245                          M8
001650  004537  001662                  JSR     5,TYPEM
001654  002245                          M8
001656  000000                          HALT
001660  000707                          BR      PRG1
                                        
                                        ;ROUTINE TO LOOP ON A SINGLE ADDRESS
```

```
                                        ;ROUTINE TO TYPE A MESSAGE

001662  010026                  TYPEM:  MOV     %0,(6)+         ;SAVE REGISTER 0
001664  012500                          MOV     (5)+,%0         ;PLACE MESSAGE ADDRESS IN R0
001666  112037  001022                  MOVB    (0)+,TERM       ;GET TERMINATOR CHARACTER
001672  112037  001020          TYPEMA: MOVB    (0)+,CHARA      ;GET NEXT CHARACTER
001676  123737  001020  001022          CMPB    CHARA,TERM      ;WAS NEXT CHARACTER THE TERM
001704  001005                          BNE     TYPEMB          ;CHARACTER
001706  014600                          MOV     -(6),%0         ;RESTORE R0
001710  105737  177564                  TSTB    TPCSR
001714  100375                          BPL     .-4
001716  000205                          RTS     5               ;AND EXIT
001720  123727  001020  000045  TYPEMB: CMPB    CHARA,#'%       ;WAS CHARACTER %
001726  001027                          BNE     TYPEMC
001730  105737  177564                  TSTB    TPCSR           ;TEST TELEPRINTER FLAG
001734  100375                          BPL     .-4             ;AND WAIT FOR DONE
001736  012737  000215  177566          MOV     #215,TPDBR      ;LOAD TELEPRINTER WITH CAR. RET
001744  013737  001010  001012          MOV     FILLER,FILCNT           ;LOAD FILLER COUNT
001752  000403                          BR      1$
001754  012737  000006  177566  2$:     MOV     #6,TPDBR        ;PRINT FILLER CHAR
001762  105737  177564          1$:     TSTB    TPCSR           ;TEST TELEPRINTER FLAG
001766  100375                          BPL     .-4             ;AND WAIT FOR DONE
001770  005337  001012                  DEC     FILCNT          ;FINISHED FILLERS ?
001774  001367                          BNE     2$              ;BR IF NOT
001776  012737  000212  177566          MOV     #212,TPDBR      ;LOAD TELEPRINTER WITH LINE FEED
002004  000732                          BR      TYPEMA          ;GET NEXT CHARACTER
002006  105737  177564          TYPEMC: TSTB    TPCSR           ;TEST TELEPRINTER FLAG
002012  100375                          BPL     .-4             ;AND WAIT FOR DONE
002014  013737  001020  177566          MOV     CHARA,TPDBR     ;LOAD TELEPRINTER BUFFER
002022  000723                          BR      TYPEMA          ;AND GET NEXT CHARACTER

                                        ;SCOPE ROUTINE. THIS ROUTINE IS ENTERED AT THE END OF EACH SUBTEST.

002024  032777  040000  176140  SCOPEC: BIT     #40000,@SWR     ;TEST SR FOR SCOPE
002032  001023                          BNE     SCOPEB          ;YES SCOPE
002034  032777  004000  176130          BIT     #4000,@SWR      ;TEST FOR ITERATION
002042  001007                          BNE     SCOPEG          ;INHIBIT ITERATION
002044  023737  002076  002074          CMP     SCOPEF,ICOUNT   ;ITERATION COMPLETE
002052  001403                          BEQ     SCOPEG          ;ITERATION COMPLETE GO TO SCOPEG
002054  005237  002076                  INC     SCOPEF          ;INCREMENT ITERATION COUNT
002060  000410                          BR      SCOPEB          ;GO TO SCOPEB
002062  005037  002076  SCOPEG: CLR     SCOPEF          ;CLEAR ITERATION COUNT
002066  011637  002100          MOV     @%6,RETURN      ;GET ADDRESS OF NEXT TEST
002072  000002                  RTI                     ;EXIT
002074  000100          ICOUNT: 100
002076  000000          SCOPEF: 0                       ;CONTAINS SUBTEST ITERATION COUNT
002100  000000          RETURN: .WORD   0               ;CONTAINS RETURN PC FOR SCOPE
002102  005726          SCOPEB: TST     (6)+            ;POP PC
002104  012637  177776          MOV     (6)+,PSW        ;RESTORE CONDITION CODES
002110  000177  177764          JMP     @RETURN
```

```
                                        ;THIS ROUTINE CONVERTS AN OCTAL NUMBER TO ASCII AND TYPES IT ON THE TTY.

002114  000000                  D2BTYP: 0
002116  013746  177564          02A:    MOV     TPCSR,-(6)      ;SAVE TPCSR
002122  010246                          MOV     %2,-(6)         ;SAVE R2
002124  010146                          MOV     %1,-(6)         ;SAVE R1
002126  010046                          MOV     %0,-(6)         ;SAVE R0
002130  013700  002114                  MOV     D2BTYP,%0       ;GET DATA TO BE TYPED
002134  012701  000006                  MOV     #6,%1           ;GET COUNTER
002140  005002                          CLR     %2              ;CLEAR WORKING REGISTER
002142  006100                          ROL     %0              ;MOV FIRST BIT (MSB) INTO
002144  006102                          ROL     %2              ;R2
002146  062702  000260          02AA:   ADD     #260,%2         ;FORM ASCII CODE
002152  105737  177564                  TSTB    TPCSR           ;TEST TELEPRINTER
002156  100375                          BPL     .-4             ;FLAG AND WAIT UNTIL DONE
002160  010237  177566                  MOV     %2,TPDBR        ;LOAD TELEPRINTER BUFFER
002164  005002                          CLR     %2              ;CLEAR WORKING REGISTER
002166  006100                          ROL     %0              ;ROTATE THE
002170  006102                          ROL     %2              ;NEXT
002172  006100                          ROL     %0              ;OCTAL CHARACTER
002174  006102                          ROL     %2              ;INTO
002176  006100                          ROL     %0              ;REGISTER
002200  006102                          ROL     %2              ;TWO
002202  005301                          DEC     %1              ;DECREMENT COUNTER
002204  001360                          BNE     02AA            ;GO TO 02AA IF NOT 0
002206  012600                          MOV     (6)+,%0         ;FINISHED.  RESTORE REGISTERS
002210  012601                          MOV     (6)+,%1         ;
002212  012602                          MOV     (6)+,%2         ;
002214  012637  177564                  MOV     (6)+,TPCSR      ;AND TPCSR
002220  000207                          RTS     7               ;AND EXIT

                                        ;ASCII MESSAGES
002222  022500  052107  032055  M7:     .ASCII  '@%GT-40 ROM DATA%%@'
002230  020060  047522  020115
002236  040504  040524  022445
002244     100
002245     100  022445     100  M8:     .ASCII  '@%%@'
002251     100  020040     100  M9:     .ASCII  '@  @'
         002256                          .EVEN

002256  013746  000004          SWITCH: MOV     @#ERRVEC,-(SP)  ;SAVE VECTORS CONTENTS
002262  012737  002310  000004          MOV     #1$,@#ERRVEC    ;SET UP FOR TRAP
002270  012737  177570  000172          MOV     #DSWR,@#SWR     ;SET UP TO TEST FOR SWITCH REGISTER
002276  022777  177777  175666          CMP     #-1,@SWR        ;TEST FOR SWITCH REGISTER
002304  001005                          BNE     3$              ;SWITCH REGISTER IS PRESENT
002306  000401                          BR      2$              ;NO SWITCH REGISTER
002310  022626                  1$:     CMP     (SP)+,(SP)+     ;POP 2 WORDS OFF STACK
002312  012737  000170  000172  2$:     MOV     #SWREG,@#SWR    ;SET UP FOR SOFTWARE SWITCH REGISTER
002320  012637  000004          3$:     MOV     (SP)+,@#ERRVEC  ;RESTORE VECTORS CONTENTS
002324  032777  000400  175640          BIT     #400,@SWR       ;TEST BIT 8
002332  001007                          BNE     4$              ;BR IF VERSION 1
002334  012737  001000  001002          MOV     #512.,WORDS     ;SET UP VERSION 2 LENGTH
002342  012737  006000  001004          MOV     #START,IMAGE    ;SET UP VERSION 2 STARTING ADD.
002350  000406                          BR      5$
```

```
   455  002352  012737  000400  001002  4$:     MOV     #256.,WORDS     ;SET UP VERSION 1 LENGTH
   456  002360  012737  016000  001004          MOV     #STARTA,IMAGE   ;SET UP VERSION 1 STARTING ADD.
   457  002366  000207                  5$:     RTS     PC
   458
   459                                          .SBTTL  ROM VERSION 2 VALUES
```

```
                        ;*********************************************************
                        ;       EXCEPT FOR THE NEW ORGIN ADDRESS AND SEVERAL "!160000"
                        ;       FOR ADDRESS FUDGING THIS IS AN EXACT COPY OF THE CONTENTS
                        ;       OF THE GT-40 BOOTSTRAP VERSION #2
                        ;*********************************************************


                                .TITLE  SCROLLING ROM BOOTSTRAP FOR THE GT40




                        ;                       BOOTGT.T16      OCT 10, 1973



                        ;               COPYRIGHT 1973, DIGITAL EQUIPMENT CORPORATION
                        ;               146 MAIN STREET
                        ;               MAYNARD, MASSACHUSETTS
                        ;                                       01754


                        ;               WRITTEN BY JACK BURNESS.



                        ;               THIS PROGRAM IS THE SECOND VERSION THE THE ROM BOOTSTRAP FOR
                        ;       THE GT40 DISPLAY TERMINAL. IT INCLUDES SCROLLING AND AN END OF
                        ;       MEMORY SEARCH FOR THE LOADER.


                                .ENABL  ABS,AMA                 ;ASSEMBLER DIRECTIVES FOR ABSOLUTE BINARY OUTPUT
                                                                ;  NOTE: USE "MACDLX" TO ASSEMBLE THIS PROGRAM.




                                .SBTTL  DEFINITION SECTION
                                .PAGE
```

```
                        ;
                        ;                       REGISTER DEFINITIONS
                        ;                       -------- -----------
                        ;
                        ;
                        ;               BASIC DEFINTIONS
                        ;               ----- ----------
                        ;
000000                          R0=%0                           ;DEFINE STANDARD VALUES.
000001                          R1=%1
000002                          R2=%2
000003                          R3=%3
000004                          R4=%4
000005                          R5=%5
000006                          SP=%6
000007                          PC=%7
                        ;
                        ;               GT40 DEFINTIONS
                        ;               ---- ----------
                        ;
000000                          CHAR=R0                         ;CONTAINS THE INPUT CHARACTER.
000001                          POINTR=R1                       ;POINTS TO NEXT INSEPTION BYTE IN DISPLAY BUFFER
000002                          TABCNT=R2                       ;CHARACTER COUNTER FOR THE "TAB" FEATURE.
000003                          SCAN=R3                         ;GENERALLY CONTAINS A POINTER WHICH
                                                                ;IS USED WHEN SCANNING MEMORY FOR SOMETHING.
000004                          HOLD=R4                         ;TYPICALLY A TEMPORARY WHICH IS USED TO RETAIN
                                                                ;A VALUE FOR A SHORT TIME.
000005                          COUNTR=R5                       ;TYPICALLY USED AS A COUNTER.


                        ;               LOADER DEFINITIONS
                        ;               ------ ----------


000000                          L.BYT=CHAR                      ;CHARACTER INPUT FOR THE LOADER.
000001                          L.ADR=POINTR                    ;CURRENT MEMORY ADDRESS TO BE LOADED.
000002                          L.BC=TABCNT                     ;NUMBER OF DATA ITEMS TO LOAD.
000005                          L.CKSM=COUNTR                   ;CHECKSUM ON THE INPUT DATA.
000003                          INDEX=SCAN                      ;INDICATES HOW TO ASSEMBLE THE 8 BIT CHARACTER.


                                .PAGE
```

```
                        ;
                        ;                       MAJOR SYSTEM DEFINITIONS
                        ;                       ----- ------ -----------

166000                  ORIGIN=166000           ;ORIGIN OF THE BOOTSTRAP.

175610                  DL11IS=175610           ;INPUT STATUS REGISTER OF DL11
175612                  DL11IB=DL11IS+2         ;INPUT CHARACTER FROM DL11
175614                  DL11OS=DL11IB+2         ;OUTPUT STATUS OF THE DL11
175616                  DL11OB=DL11OS+2         ;OUTPUT CHARACTER TO THE DL11

177560                  KBDIS=177560            ;KEYBOARD INPUT STATUS
177562                  KBDIB=KBDIS+2           ;CURRENT CHARACTER FROM KEYBOARD.

172000                  GT40PC=172000           ;GT40 PROGRAM COUNTER.
172002                  GT40SR=GT40PC+2         ;GT40 STATUS REGISTER ADDRESS.


001000                  BSTART=1000             ;START OF THE DISPLAY BUFFER
007000                  BLIMIT=7000             ;APPROXIMATE END OF THE DISPLAY BUFFER.
007776                  TMPEND=7776             ;LOCATION OF INITIALIZATION STACK.
000004                  CORSTR=4                ;LOCATION OF PDP-11 TRAP VECTOR.
007712                  JMPADD=BLIMIT+10.       ;WHERE THE POINTER IS TO FIRST CHAR ON SCREEN
000040                  NUMLIN=32.              ;NUMBER OF LINES OF TEXT TO SHOW ON THE SCREEN

005015                  CRLF=5015               ;CARRIAGE RETURN - LINE FEED
000175                  ALTMOD=175              ;THE "KEY" CHARACTER (I.E. ALTMODE).

160000                  DISJMP=160000           ;THE GT40 JMP INSTRUCTION
173000                  DISTOP=173000           ;THE GT40 STOP DISPLAY INSTRUCTION.


                        .SBTTL  INITIALIZATION AND RESTART CODE
                        .PAGE
```

```
                                ;               GT40 BOOTSTRAP CODE
                                ;               ---- --------- ----

006000                                  .=6000
                                ;       .=ORIGIN                        ;DEFINE ORIGIN OF THE BOOTSTRAP.

                                ;               COLD INITIALIZATION CODE
                                ;               ---- -------------- ----

006000  000005                  START:  RESET                           ;RESET ALL HARDWARE NOW.
006002  012737  000007  175610          MOV     #7,DL11IS               ;INITIALIZE DL-11 INPUT NOW.
006010  012706  007776                  MOV     #TMPEND,SP              ;ESTABLISH A GOOD TEMPORARY STACK
                                                                        ;POINTER FOR CORE SEARCH.
006014  005237  175614                  INC     DL11OS                  ;SET BREAK BIT
006020  004337  166652                  JSR     SCAN,OUTLIT!160000      ;FOR 2 CHARACTER TIMES
006024  000000                          .WORD 0                         ;SEND TWO ZERO'S

006026  012703  000004                  MOV     #CORSTR,SCAN            ;GET ADDRESS OF BAD CORE TRAP VECTOR.
006032  012723  166042                  MOV     #NOTHERE!160000,(SCAN)+ ;AND INSERT A POINTER TO US THERE.

006036  005023          ENDCOR: CLR     (SCAN)+                         ;NOW CLEAR ALL OF MEMORY BEYOND THE POINTER,
006040  000776                  BR      ENDCOR                          ;UNTIL WE RUN OUT OF MEMORY AND TRAP.

006042  005743          NOTHER: TST     -(SCAN)                         ;WHEN WE TRAP OUT, WE COME HERE.
                                                                        ;WE BACK UP POINTER TO GOOD CORE.
                                                                        ;NOTE THAT IF WE TRAP OUT AGAIN, IT
                                                                        ;IS STILL OK, BECAUSE WE WILL LOOP
                                                                        ;UNTIL WE GET A GOOD CORE ADDRESS.
006044  010306                  MOV     SCAN,SP                         ;WHEN WE GET ONE, THAT IS LAST LOCATION
                                                                        ;IN THE MACHINE, AND HENCE OUR SP.
006046  105737  175614  1$:     TSTB    DL11OS                          ;SEE IF BREAK IS DONE
006052  100375                  BPL     1$                              ;NO GO BACK
006054  005037  175614          CLR     DL11OS                          ;CLEAR BREAK BIT

                                ;               RESTART INITIALIZATION CODE WHEN COMMUNICATIONS IS WORKING.
                                ;               ------- -------------- ---- ---- ------------- -- --------
```

```
006060  052706  007776          RESTRT: BIS     #TMPEND,SP              ;FORCE THE SP TO LIMIT OF EXISTING CORE.

006064  012703  006700                  MOV     #BLIMIT-NUMLIN-NUMLIN,SCAN      ;NOW WE WILL FILL THE KEY AREAS OF THE
006070  012702  000040                  MOV     #NUMLIN,TABCNT          ;DISPLAY BUFFER WITH INITIAL CR-LF'S.

006074  012723  005015          SETLP1: MOV     #CRLF,(SCAN)+           ;INSERT A CRLF NOW.
006100  005302                  DEC     TABCNT                  ;AND LOOP UNTIL DONE.
006102  003374                  BGT     SETLP1                  ;THUS DISPLAY CORE IS ALMOST CORRECT.

006104  012703  166432                  MOV     #SETUP!160000,SCAN              ;NOW WE WILL INITALIZE CORE FOR THE
                                                                ;DISPLAY. PICK UP POINTER TO LIST.

006110  012302          SETLP2: MOV     (SCAN)+,TABCNT          ;GET NUMBER OF ITEMS TO INSERT.
006112  001405                  BEQ     SETDUN                  ;IF ZERO, WE ARE DONE.
006114  012301                  MOV     (SCAN)+,POINTR          ;PICK UP FIRST CORE ADDRESS POINTER.

006116  012321          SETLP3: MOV     (SCAN)+,(POINTR)+       ;MOVE OVER A DATA ITEM NOW.
006120  005302                  DEC     TABCNT                  ;ALL DONE?
006122  003375                  BGT     SETLP3                  ;NOPE. MOVE OVER THE NEXT.
006124  000771                  BR      SETLP2                  ;YES. GET NEXT MAJOR LIST TO INSERT.

006126  012701  006776  SETDUN: MOV     #BLIMIT-2,POINTR        ;ESTABLISH THE BUFFER POINTER NOW.
```

```
                        .SBTTL  VT05 SIMULATOR
                        .PAGE
```

```
                        ;              VTOS (SCROLLING) PORTION OF THE BOOTSTRAP
                        ;              ---- ---------  ------- -- --- ---------

006132  004737  166564  NXTCHR: JSR     PC,GETCHR!160000        ;GET A CHARACTER NOW.
006136  020027  000177          CMP     CHAR,#177               ;IS IT OUT OF RANGE?
006142  002373                  BGE     NXTCHR                  ;YEP. GET ANOTHER ONE.
006144  020027  000040          CMP     CHAR,#40                ;IS IT A PRINTING CHARACTER?
006150  002020                  BGE     NORMAL                  ;YES. IT'S A NORMAL PRINTING CHARACTER.
006152  010003                  MOV     CHAR,SCAN               ;MOVE IT OVER SO WE CAN PLAY WITH IT.
006154  162703  000007          SUB     #7,SCAN                 ;BIAS SO THAT BELL [7] IS ZERO.
006160  020327  000007          CMP     SCAN,#7                 ;IF CHARACTER IS LESS THEN BELL OR
006164  103362                  BHIS    NXTCHR                  ;GREATER THEN CR, THEN IGNORE.
006166  006303                  ASL     SCAN                    ;IF GOOD, MAKE IT WORD INDEX.
006170  060307                  ADD     SCAN,PC                 ;AND GO TO THE CORRECT ROUTINE.

006172  000426                  BR      BELL                    ;7=BELL
006174  000406                  BR      NORMAL                  ;10=BACKSPACE
006176  000411                  BR      TAB                     ;11=TAB
006200  000437                  BR      LF                      ;12=LINE FEED [LF]
006202  000420                  BR      VT                      ;13=VERTICAL TAB [VT]
006204  000424                  BR      FF                      ;14=FORM FEED [FF]
                                                                ;15=CARRIAGE RETURN [CR]


006206  012702  177777  CR:     MOV     #-1,TABCNT              ;RESET TAB POSITION ON A CR, AND
                                                                ;FALL THROUGH TO INSERT THE CHARACTER.

006212  004737  166350  NORMAL: JSR     PC,INSERT!160000        ;INSERT THE CHARACTER IN THE BUFFER.
006216  005202                  INC     TABCNT                  ;UPDATE TAB POSITION NOW.
006220  000744                  BR      NXTCHR                  ;AND GET NEXT CHARACTER.


006222  012700  000040  TAB:    MOV     #40,CHAR                ;ON A TAB, INSERT BLANKS UNTIL THE
006226  004737  166350          JSR     PC,INSERT!160000        ;NEXT CHARACTER POSITION IS A MULTIPLE
006232  005202                  INC     TABCNT                  ;OF 8.
006234  032702  000007          BIT     #7,TABCNT               ;ARE WE DONE YET?
006240  001370                  BNE     TAB                     ;NOPE.
006242  000733                  BR      NXTCHR                  ;YES.


006244  111705          VT:     MOVB    (PC),COUNTR             ;THIS PUTS THE LOW BYTE OF THE
                                                                ;BRANCH CODE IN COUNTR-SAVE A WORD
006246  000405                  BR      FFLOOP                  ;

006250  005037  172002  BELL:   CLR     GT40SR                  ;RING BELL -WRITE IN GT40SR
006254  000726                  BR      NXTCHR                  ;AND LOOP BACK

006256  012705  000040  FF:     MOV     #NUMLIN,COUNTR          ;FORM FEED IS DONE BY INSERTING LF'S.
```

```
                                                                                
                                                                                
006262  012700  000012          FFLOOP: MOV     #12,CHAR                ;MAKE THE CHARACTER A LINEFEED.
006266  004737  166304                  JSR     PC,LFSUB!160000         ;DO A LINEFEED.
006272  005305                          DEC     COUNTR                  ;DONE?
006274  003372                          BGT     FFLOOP                  ;NOPE. KEEP SENDING THEM.
006276  000715                          BR      NXTCHR                  ;YES. NOW RETURN. DO NOT FALL THROUGH.
                                                                                
                                                                                
006300  012746  166132          LF:     MOV     #NXTCHR!160000,-(SP)            ;RETURN TO NXTCHR AFTER PROCESSING
                                                                        ;THE LF BY FAKING A JSR.
                                                                                
006304  013703  007012          LFSUB:  MOV     JMPADD,SCAN             ;GET POINTER TO FIRST CHAR ON SCREEN
                                                                                
006310  122300                  LFLOOP: CMPB    (SCAN)+,CHAR            ;AND LOOK FOR A LINEFEED.
006312  001406                          BEQ     LFOUND                  ;GOT IT. SEARCH HAS ENDED.
006314  020327  007000                  CMP     SCAN,#BLIMIT            ;ARE WE AT END OF BUFFER?
006320  103773                          BLO     LFLOOP                  ;NOPE. KEEP ON LOOKING.
006322  012703  001000                  MOV     #BSTART,SCAN            ;IF AT TOP, RESET TO BOTTOM OF BUFFER
006326  000770                          BR      LFLOOP                  ;AND KEEP ON LOOKING.
                                                                                
006330  005203                  LFOUND: INC     SCAN                    ;WE'VE GOT THE LINE FEED. STOP SHOWING
006332  042703  000001                  BIC     #1,SCAN                 ;FIRST LINE BY CHANGING THE "DISJMP"
006336  010337  007012                  MOV     SCAN,JMPADD             ;INSTRUCTION TO FIRST CHAR BEYOND LF.
006342  004737  166350                  JSR     PC,INSERT!160000                ;INSERT THE LF IN THE BUFFER.
006346  005000                          CLR     CHAR                    ;AND THEN INSERT ONE NULL CHARACTER BECAUSE
                                                                        ;THE "DISJMP" ADDRESS MUST BE EVEN, AND
                                                                        ;THIS GUARANTEES WE WILL NOT LOSE A
                                                                        ;A GOOD DATA CHARACTER. WE FALL THROUGH
                                                                        ;TO INSERT THE NULL IN THE BUFFER.
                                                                                
                                                                                
006350  110021                  INSERT: MOVB    CHAR,(POINTR)+          ;STICK IN THE CHARACTER NOW.
006352  032701  000001                  BIT     #1,POINTR               ;IS NEXT POSITION EVEN OR ODD?
006356  001021                          BNE     INSRTX                  ;ODD. NO PROBLEMS. SPACE IS ALLOCATED.
006360  020127  007000                  CMP     POINTR,#BLIMIT          ;EVEN. ARE WE AT THE END OF THE BUFFER?
006364  103410                          BLO     INSRTL                  ;NO. JUST MAKE ROOM FOR ANOTHER WORD.
006366  010103                          MOV     POINTR,SCAN             ;AT THE END. MOVE THE STUFF TO THE
006370  012701  001000                  MOV     #BSTART,POINTR          ;BEGINNING OF THE BUFFER.
006374  004737  166406                  JSR     PC,INSRTL!160000                ;CALL THE ROUTINE TO SAVE SPACE.
006400  005023                          CLR     (SCAN)+                 ;AND CLEAR UP THE INSTRUCTIONS AT THE
006402  005013                          CLR     (SCAN)                  ;END OF THE BUFFER.
006404  000207                          RTS     PC                      ;AND THEN RETURN.
                                                                                
006406  022121                  INSRTL: CMP     (POINTR)+,(POINTR)+     ;BYPASS THE "DISJMP" BY ADDING 4 TO POINTR.
006410  012711  166474                  MOV     #HEADER!160000,(POINTR) ;NOW INSERT THE DISJMP INSTRUCTION TO OUR HEADER
006414  012741  160000                  MOV     #DISJMP,-(POINTR)       ;AND IT'S ADDRESS (PUT THEM IN BACKWARDS).
006420  005041                          CLR     -(POINTR)               ;MAKE AVAILABLE A NEW CHARACTER SPOT.
                                                                                
006422  000207                  INSRTX: RTS     PC                      ;FINALLY RETURN TO THE CALLER.
                                                                                
                                                                                
                                                                                
                                                                                
                                                                                
006424  012737  001000  172000  GTBUSE: MOV     #BSTART,GT40PC          ;ON A BUS ERROR, WE MERELY RESTART THE GT40 AT
                                                                                
```

```
                                                        ;THE RTI FOR THIS ROUTINE
                                                        ;IS THE FIRST WORD OF THE TABLE
                                                        ;BELOW-IT SAVES A WORD!
                        ;                       INITIALIZATION TABLE FOR THE SCROLLER
                        ;                       -------------- ----- --- --- --------

006432  000002          SETUP:  .WORD   2               ;INITIALIZE 2 WORDS.--ALSO RTI FROM ABOVE
006434  000330                  .WORD   330             ;STARTING AT LOCATION 330
006436  166424                  .WORD   GTBUSE!160000           ;FIRST WORD IS POINTER TO BUS ERROR ROUT
006440  000200                  .WORD   200             ;SECOND WORD IS NEW STATUS WORD ON INTERUPT.

006442  000007                  .WORD   7               ;INITIALIZE THE END OF THE BUFFER TO
006444  006776                  .WORD   BLIMIT-2        ;A CLEAR SPACE TO INSERT THE CHARACTER.
006446  000000                  .WORD   0               ;THIS IS THE "RUNNING" START. THIS IS
006450  160000  166474          .WORD   DISJMP,HEADER!160000            ;FOLLOWED BY A DISJMP TO OUR HEADER BLOC
006454  160000  001000          .WORD   DISJMP,BSTART   ;AND THEN A DISJMP TO THE START OF THE BUFFER
006460  160000  006700          .WORD   DISJMP,BLIMIT-NUMLIN-NUMLIN     ;AND A DISJMP TO THE FIRST CHAR ON SCREE

006464  000001                  .WORD   1               ;FINALLY START THE GT40 GOING AT
006466  172000                  .WORD   GT40PC          ;THE POSITION INSTRUCTION IN THE
006470  166474                  .WORD   HEADER!160000           ;HEADER BLOCK.

006472  000000                  .WORD   0               ;END OF INIT CODE

                        ;
                        ;               HEADER BLOCK FOR THE SCROLLER

006474  103334          HEADER: .WORD   103334          ;ENABL CHAR MODE,BLINKING
006476  000177                  .WORD   177             ;A BLINKING BOX-RUB OUT!
006500  116124                  .WORD   116124          ;GO TO POINT MODE
006502  171340                  .WORD   171340          ;LOAD STATUS REGISTER
006504  000000  001352          .WORD   0,1352          ;POINT TO UPPER LEFT
006510  103324                  .WORD   103324          ;BACK TO CHAR MODE
006512  160000  007010          .WORD   DISJMP,JMPADD-2 ;AND TO THE CHANGING JMP INST.

                        .SBTTL  COMMUNICATIONS AND MISC. SUPPORT ROUTINES
                        .PAGE
```

```
                                ;                       COMMUNICATIONS HANDLING ROUTINES
                                ;                       -------------- -------- --------

                                ;               THE DL-11 HANDLER
                                ;               --- ----- -------

006516  105737  175610        GETDL:  TSTB    DL11IS          ;CHECK THE HOST INPUT STATUS.
006522  100011                        BPL     GETDL1          ;HOST DID NOT SEND ANYTHING, YET.
006524  113700  175612                MOVB    DL11IB,CHAR     ;HOST SENT US A CHARACTER. PROCESS IT.
006530  012737  000007  175610        MOV     #7,DL11IS       ;REENABLE THE HOST TELECOMMUNICATIONS.
006536  042700  177600                BIC     #-200,CHAR      ;MAKE CHARACTER JUST SEVEN BITS.
006542  001765                        BEQ     GETDL           ;IF NULL, IGNORE IT.
006544  000207                        RTS     PC              ;ELSE RETURN NOW.

006546  105737  177560        GETDL1: TSTB    KBDIS           ;DID USER TYPE A CHARACTER?
006552  100361                        BPL     GETDL           ;NO. GO BACK AND CHECK HOST MACHINE.
006554  113737  177562  175616        MOVB    KBDIB,DL110B    ;MOVE THE CHARACTER TO THE HOST.
006562  000755                        BR      GETDL           ;AND CHECK AGAIN FOR INPUT.


                                ;               THE "GET CHARACTER" ROUTINE
                                ;               --- --- ---------  -------

006564  004737  166516        GETCHR: JSR     PC,GETDL!160000 ;GET A CHARACTER FROM THE HOST NOW.
006570  020027  000175                CMP     CHAR,#ALTMOD    ;IS IT AN "ALTMODE"
006574  001025                        BNE     GETEXT          ;NO. EXIT NOW.

006576  004737  166516                JSR     PC,GETDL!160000 ;YES. GET ANOTHER ONE NOW.
006602  020027  000114                CMP     CHAR,#'L        ;IS IT AN "L"
006606  001501                        BEQ     LOADER          ;YES. START LOADING NOW.
006610  020027  000122                CMP     CHAR,#'R        ;IS IT AN "R"
006614  001015                        BNE     GETEXT          ;NO. IGNORE THE ALTMODE AND JUST RETURN THE CHAR

006616  012737  173000  007010        MOV     #DISTOP,JMPADD-2 ;YES. RESET. STOP DISPLAY BY INSERTING A "DISTOP
006624  000137  166060        PRESTR: JMP     RESTRT!160000           ;INSTRUCTION IN THE BUFFER, AND RESTART.


                                ;               THE "GET A SIX BIT CHARACTER" ROUTINE
```

```
                        ;               --- --- - --- --- ---------  -------
                        ;
                        ;
                        ;
006630  004737  166564  GETSIX: JSR     PC,GETCHR!160000                ;GET A CHARACTER NOW.
006634  020027  000040          CMP     CHAR,#40                ;IS IT A LEGAL PRINTING CHARACTER?
006640  002517                  BLT     L.BAD                   ;NOPE. ABORT
006642  020027  000137          CMP     CHAR,#137               ;IT'S BIG ENOUGH. IS IT TOO BIG?
006646  003114                  BGT     L.BAD                   ;YEP. ABORT.
                        ;
006650  000207  GETEXT: RTS     PC                              ;RETURN TO THE CALLER.
                        ;
                        ;
                        ;               THIS OUTPUTS TWO CHARACTERS VIA A
                        ;               JSR SCAN,OUTLIT
                        ;               'TWO CHARACTERS'
                        ;
006652  112337  175616  OUTLIT: MOVB    (SCAN)+,DL110B
006656  112337  175616          MOVB    (SCAN)+,DL110B          ;DOUBLE BUFFERED
006662  000203                  RTS     SCAN                    ;RETURN
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;               THE "GET AN EIGHT BIT CHARACTER" ROUTINE
                        ;               --- --- -- ----- --- ---------  -------
                        ;
                        ;
                        ;
                        ;                       THIS ROUTINE DIFFERS FROM THE PREVIOUS ROUTINES
                        ;               IN THAT IT WILL TAKE SIX BIT CHARACTERS AND ASSEMBLE
                        ;               THEM FOR THE LOADER TO USE. NOTE THAT FROM THIS POINT
                        ;               ON WE WILL SWITCH TO THE LOADER DEFINITIONS OF THE
                        ;               REGISTERS. THUS THE CHARACTER IS RETURNED IN
                        ;               REGISTER "L.BYT" RATHER THAN CHAR (THOUGH THEY ARE
                        ;               PHYSICALLY THE SAME).
                        ;
                        ;
006664  004737  166630  GETB:   JSR     PC,GETSIX!160000                ;GET A SIXBIT CHARACTER.
006670  010046                  MOV     L.BYT,-(SP)             ;SAVE IT ON THE STACK.
006672  005723                  TST     (INDEX)+                ;UPDATE INDEX TO NEXT ITEM (ALL ARE #2)
006674  000163  166676          JMP     GETBTB-2!160000(INDEX)          ;AND DISPATCH ACCORDING TO THE INDEX.
                        ;
006700  000404  GETBTB: BR      GET81                   ;INDEX=2: ASSEMBLE FIRST CHAR
006702  000416          BR      GET82                   ;INDEX=4: ASSEMBLE SECOND CHAR
006704  000432          BR      GET83                   ;INDEX=6: ASSEMBLE THIRD AND LAST CHAR
                                                        ;INDEX=8: RESET INDEX TO 0 [2] AND RETRY.
                        ;
                        ;
006706  012703  000002  GET84:  MOV     #2,INDEX                ;THE FOURTH INDEX IS THE SAME AS THE FIRST
                                                        ;INDEX. JUST RESET IT AND FALL THROUGH.
                        ;
```

```
  991
  992  006712  004737  166630        GET81:  JSR     PC,GETSIX!160000        ;GET ANOTHER CHARACTER NOW.
  993  006716  010004                        MOV     L.BYT,HOLD              ;AND PRESERVE IT FOR NEXT TIME THROUGH.
  994  006720  006300                        ASL     L.BYT                   ;NOW THROW AWAY LEFT MOST BITS OF
  995  006722  006300                        ASL     L.BYT                   ;THE 8 BIT CHARACTER. NOW MERGE IN
  996  006724  106300                        ASLB    L.BYT                   ;THE LEFT TWO BITS OF THE
  997  006726  106116                        ROLB    (SP)                    ;NEW SIX BIT CHARACTER WITH THE SIX
  998  006730  106300                        ASLB    L.BYT                   ;BITS FROM THE CHARACTER ON THE
  999  006732  106116                        ROLB    (SP)                    ;STACK. 1ST CHARACTER IS NOW ASSEMBLED,
 1000  006734  012600                        MOV     (SP)+,L.BYT             ;SO WE'LL RETURN IT TO THE USER.
 1001  006736  000207                        RTS     PC                      ;AND THEN WE SHALL RETURN TO HIM.
 1002
 1003
 1004  006740  006300        GET82:  ASL     L.BYT                   ;THE SECOND CHARACTER IS CREATED FROM
 1005  006742  006300                ASL     L.BYT                   ;THE 4 RIGHT BITS OF THE PREVIOUS CHARACTER
 1006  006744  106300                ASLB    L.BYT                   ;AND THE FOUR MIDDLE BITS OF THE PRESENT
 1007  006746  106104                ROLB    HOLD                    ;8 BIT CHARACTER.
 1008  006750  106300                ASLB    L.BYT                   ;WE WILL CREATE THE NEW 8 BIT
 1009  006752  106104                ROLB    HOLD                    ;IN THIS REGISTER, SINCE IT
 1010  006754  106300                ASLB    L.BYT                   ;MORE CONVIENT. WE WILL MOVE OVER THE
 1011  006756  106104                ROLB    HOLD                    ;ANSWER AT THE END.
 1012  006760  106300                ASLB    L.BYT                   ;ONE MORE TO GO
 1013  006762  106104                ROLB    HOLD                    ;DONE.
 1014  006764  010400                MOV     HOLD,L.BYT              ;BRING OVER THE VALUE.
 1015  006766  012604                MOV     (SP)+,HOLD              ;AND REMEMBER THE LAST CHARACTER WE RECEIVED.
 1016  006770  000207                RTS     PC                      ;AND RETURN TO THE CALLER.
 1017
 1018
 1019  006772  006100        GET83:  ROL     L.BYT                   ;FINAL CHARACTER IS EASY. JUST A
 1020  006774  106100                ROLB    L.BYT                   ;SIMPLE MERGER OF LEFT TWO BITS OF
 1021  006776  006004                ROR     HOLD                    ;PREVIOUS VALUE WITH RIGHT SIX BITS
 1022  007000  106000                RORB    L.BYT                   ;OF LAST (4TH) CHARACTER RECEIVED.
 1023  007002  006004                ROR     HOLD
 1024  007004  106000                RORB    L.BYT                   ;AND WE ARE DONE.
 1025  007006  005726                TST     (SP)+                   ;FINALLY THROW AWAY STACK.
 1026  007010  000207                RTS     PC                      ;AND RETURN TO THE CALLER.
 1027
 1028
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039                                .SBTTL  THE LOADER
 1040                                .PAGE
```

```
                                        ;               THE LOADER
                                        ;               --- ------

007012  012737  173000  007010  LOADER: MOV     #DISTOP,JMPADD-2        ;STOP THE GT40 BY INSERTING A "DISTOP" IN THE LI

007020  005003                  L.LD2:  CLR     INDEX                   ;RESET THE 8 BIT ASSEMBLER TO THE FIRST CHAR

007022  005005                          CLR     L.CKSM                  ;CLEAR THE CHECKSUM
007024  004737  167114                  JSR     PC,L.PTR!160000         ;GET A BYTE NOW.
007030  105300                          DECB    L.BYT                   ;IS IT A ONE (HEADER)?
007032  001373                          BNE     L.LD2                   ;NO. WAIT FOR THE ONE.

007034  004737  167114                  JSR     PC,L.PTR!160000         ;YES. SKIP OVER THE NEXT CHARACTER NOW.

007040  004737  167126                  JSR     PC,L.GWRD!160000                ;ASSEMBLE A WORD NOW.
007044  010002                          MOV     L.BYT,L.BC              ;MOVE OVER TO THE COUNTER.
007046  162702  000004                  SUB     #4,L.BC                 ;REDUCE TO ACTUAL DATA COUNT.
007052  022702  000002                  CMP     #2,L.BC                 ;ANY DATA AT ALL?
007056  001433                          BEQ     L.JMP                   ;NO. MUST BE END
007060  004737  167126                  JSR     PC,L.GWRD!160000                ;YES. ASSEMBLE A DATA WORD NOW.
007064  010001                          MOV     L.BYT,L.ADR             ;AND THIS MUST BE THE FIRST ADDRESS.

007066  004737  167114  L.LD3:  JSR     PC,L.PTR!160000         ;GET A BYTE OF DATA NOW.
007072  002006                  BGE     L.LD4                   ;ALL DONE?
007074  105705                  TSTB    L.CKSM                  ;YEP. COUNTER IS MINUS. CHECK CHECKSUM.
007076  001751                  BEQ     L.LD2                   ;CHECKSUM GOOD. GET NEXT COMMAND.

007100  004337  166652  L.BAD:  JSR     SCAN,OUTLIT!160000      ;BAD LOAD INFORM HOST
007104     175     102          .BYTE   ALTMOD,'B               ;SEND ALTMODE B
007106  000646                  BR      PRESTRT                 ;AND RESTART THE DISPLAY.

007110  110021  L.LD4:  MOVB    L.BYT,(L.ADR)+          ;INSERT BYTE INTO MEMORY.
007112  000765                  BR      L.LD3                   ;AND GET THE NEXT BYTE.

007114  004737  166664  L.PTR:  JSR     PC,GET8!160000                  ;ASSEMBLE AN 8 BIT CHARACTER NOW.
007120  060005                  ADD     L.BYT,L.CKSM            ;UPDATE THE CHECKSUM NOW.
007122  005302                  DEC     L.BC                    ;DECREMENT THE CHARACTER COUNTER.
007124  000207                  RTS     PC                      ;AND RETURN TO THE CALLER NOW.

007126  004737  167114  L.GWRD: JSR     PC,L.PTR!160000         ;ASSEMBLE A WORD. FIRST GET A CHARACTER
007132  010046                  MOV     L.BYT,-(SP)             ;AND SAVE IT.
007134  004737  167114          JSR     PC,L.PTR!160000         ;AND THEN GET ANOTHER ONE.
```

```
1097  007140  000300                  SWAB    L.BYT                   ;AND THEN REASSEMBLE THE MESS.
1098  007142  052600                  BIS     (SP)+,L.BYT             ;WITH THE FEARSOME POWER OF THE 11.
1099  007144  000207                  RTS     PC                      ;AND RETURN TO THE CALLER.
1100
1101
1102
1103
1104  007146  004737  167126  L.JMP:  JSR     PC,L.GWRD!160000                ;ALL DONE WITH THE LOAD. ASSEMBLE
1105  007152  010046                  MOV     L.BYT,-(SP)             ;THE STARTING ADDRESS NOW.
1106  007154  004737  167114          JSR     PC,L.PTR!160000         ;AND DON'T FORGET TO CHECKSUM IT.
1107  007160  105705                  TSTB    L.CKSM
1108  007162  001346                  BNE     L.BAD                   ;A BAD CHECKSUM. ALL IS EVIL.
1109
1110  007164  004337  166652          JSR     SCAN,OUTLIT!160000              ;GOOD CHKSUM,INFORM HOST
1111  007170     175     107          .BYTE   ALTMOD,'G               ;WITH ALTMOD G
1112
1113  007172  032716  000001          BIT     #1,(SP)                 ;DO WE WANT TO START EXECUTION?
1114  007176  001401                  BEQ     L.JMP1                  ;YES. AWAY WE GO.
1115
1116  0072__  000000  L.HALT: HALT                                    ;IF NOT, HALT.
1117
1118  007202  000136  L.JMP1: JMP     @(SP)+                          ;IF GO, THEN GO ALREADY. WHEEEE!
1119
1120
1121
1122                          .SBTTL  THE SELF TEST
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135                          .PAGE
```

```
1136                                                    ;THIS IS GT40 QLICK TEST
1137                                                            ;GIVES QUICK VISUAL TEST
1138                                                            ;OF CONDITION OF MACHINE
1139                                                            ;WITHOUT READING IN DIAG.
1140            100000          CHAR=100000
1141            104000          SHORTV=104000
1142            110000          LONGV=110000
1143            114000          POINT=114000
1144            120000          GRAPHX=120000
1145            124000          GRAPHY=124000
1146            130000          RELATV=130000
1147
1148            002000          INT0=2000
1149            002200          INT1=2200
1150            002400          INT2=2400
1151            002600          INT3=2600
1152            003000          INT4=3000
1153            003200          INT5=3200
1154            003400          INT6=3400
1155            003600          INT7=3600                       ;BRIGHTEST
1156
1157            000100          LPOFF=100
1158            000140          LPON=140
1159            000020          BLKOFF=20
1160            000030          BLKON=30
1161
1162            000004          LINE0=4
1163            000005          LINE1=5
1164            000006          LINE2=6
1165            000007          LINE3=7
1166
1167            160000          DJMP=160000
1168            164000          DNOP=164000
1169            170000          STATSA=170000
1170            173400          DSTOP=173400                    ;STOP INTERRUPT
1171
1172            000300          LPLITE=300
1173            000200          LPDARK=200
1174            000040          ITAL0=40                        ;ITALICS OFF
1175            000060          ITAL1=60                        ;"       ON
1176            000004          SYNON=4                         ;SYNC ON
1177
1178
1179            174000          STATSB=174000
1180
1181            000100          INCR=100                        ;LOAD GRAPH INCR
1182            040000          INTX=40000                      ;INTENSIFY BIT
1183            001777          MAXX=1777                       ;BIGGEST X VECTOR
1184            001377          MAXY=1377                       ;BIGGEST Y VECTOR
1185            020000          MINUSX=20000                    ;THE MINUS BIT
1186            020000          MINUSY=MINUSX
1187            017600          MAXSX=17600                     ;BIGGEST X IN SHORTVEC
1188            000077          MAXSY=77                        ;"      Y IN  "
1189            000100          MINSUY=100                      ;MINUS BIT FOR Y IN SHORTVEC
1190
1191
```

```
1182  007204  012737  167214  172000          MOV     @FILEO!160000,GT40PC            ;START THE GT40
1183  007212  000001                          WAIT                                    ;AND WAIT
1184
1185  007214  114022                  FILEO:  POINT!BLKOFF                            ;POINT--INVISIBLE
1186  007216  000000                          0
1187  007220  001377                          MAXY
1188
1189  007222  112004                          LONGV!INT0!LINE0                        ;DRAW TOP LINE
1200  007224  041777                          INTX!MAXX
1201  007226  000000                          0
1202
1203  007230  112405                          LONGV!INT2!LINE1
1204  007232  040000                          INTX                                    ;DRAW LINE TO RIGHT
1205  007234  021377                          MINUSX!MAXY
1206
1207  007236  113006                          LONGV!INT4!LINE2
1208  007240  061777                          INTX!MINUSX!MAXX                        ;DRAW BOTTOM LINE
1209  007242  000000                          0
1210
1211  007244  113407                          LONGV!INT6!LINE3
1212  007246  040000                          INTX
1213  007250  001377                          MAXY                                    ;DRAW LINE TO LEFT
1214
1215  007252  114000                          POINT
1216  007254  000400                          400
1217  007256  000500                          500
1218  007260  106200                          SHORTV!INT1
1219  007262  057677                          57677                                   ;+X+Y
1220  007264  106600                          SHORTV!INT3
1221  007266  077677                          77677                                   ;+X-Y
1222  007270  107200                          SHORTV!INT5
1223  007272  077777                          77777                                   ;-X-Y
1224  007274  107600                          SHORTV!INT7
1225  007276  057777                          57777                                   ;-X+Y
1226
1227  007300  114000                          POINT
1228  007302  001400                          1400
1229  007304  000500                          500
1230  007306  133000                          RELATV!INT4!BLKON
1231  007310  057677                          57677                                   ;+X+Y
1232  007312  077677                          77677                                   ;+X-Y
1233  007314  077777                          77777                                   ;-X-Y
1234  007316  057777                          57777                                   ;-X+Y
1235
1236  007320  114000                          POINT
1237  007322  000400                          400
1238  007324  000100                          100
1239  007326  174120                          STATSB!INCR+20                          ;TRY GRAPH MODES
1240  007330  114000                          POINT
1241  007332  001000                          1000
1242  007334  000200                          200
1243
1244  007336  120000                          GRAPHX
1245  007340  001010                          1010
1246  007342  001020                          1020
1247  007344  001030                          1030
```

```
1248  007346  001040                  1040
1249  007350  001050                  1050
1250
1251  007352  114000                  POINT
1252  007354  001000                  1000
1253  007356  001200                  1200
1254
1255  007360  124000                  GRAPHY
1256  007362  001020                  1020
1257  007364  001030                  1030
1258  007366  001040                  1040
1259  007370  001050                  1050
1260  007372  001060                  1060
1261
1262  007374  160000                  DJMP
1263  007376  167214                  FILED!160000
1264
1265                                  .SBTTL  PAPER TAPE BOOT
```

```
                                ;
                                ; PAPER TAPE BOOT
                                ;
            177550              HSR=177550                      ;HIGH SPEED READER ADDRESS
            177560              LSR=177560                      ;LOW SPEED READER ADDRESS
                                        .=ORIGIN+1400
                                ;
007400  012701  160000          PTBOOT: MOV     #160000,R1      ;SET MEMORY CHECK LIMITS
007404  012702  000004                  MOV     #4,R2           ;TRAP ADDRESS IS LOC. 4
007410  012703  167500                  MOV     #DEV+4!160000,R3        ;POINTER TO DEVICE ADDRESSES
007414  010712                          MOV     PC,@R2          ;PRESET TRAP ADDRESS IN LOC. 4
007416  012706  000024                  MOV     #24,SP          ;STACK SET UP AT SPECIAL ADDRESS
007422  014304                  DEV1:   MOV     -(R3),R4        ;GET DEVICE ADDRESS
007424  005714                          TST     @R4             ;CHECK AVAILABILITY OF DEVICE
007426  100775                          BMI     DEV1            ;CHECK DEVICE FOR ERRORS
007430  010712                          MOV     PC,@R2          ;RESET TRAP ADDRESS AT LOC. 4
007432  012706  000024                  MOV     #24,SP          ;SPECIAL ADDRESS USED AS MASK LATER
007436  010441                          MOV     R4,-(R1)        ;DO MEM CHK:READER STATUS ADDRESS
                                                                ;IS MOVED
007440  040601                          BIC     SP,R1           ;SET R1=X7752,MASK IN SP=24
007442  010111                          MOV     R1,@R1          ;STORE OWN ADDRESS IN POINTER
007444  011102                  LOOP:   MOV     @R1,R2          ;GET BYTE POINTER
007446  005214                          INC     @R4             ;ENABLE READER
007450  105714                          TSTB    @R4             ;TEST DONE BIT
007452  100376                          BPL     .-2             ;WAIT UNTIL READY
007454  116412  000002                  MOVB    2(R4),@R2       ;THEN PICK IT UP AND STORE IT
007460  005211                          INC     @R1             ;BUMP POINTER
007462  120227  000375                  CMPB    R2,#375         ;STORED JUMP OFFSET?
007466  001366                          BNE     LOOP            ;NOT YET
007470  105222                          INCB    (R2)+           ;YES,ALL DONE
007472  000142                          JMP     -(R2)           ;GO EXECUTE AS BRANCH
                                ;
                                ; DEVICE ADDRESSES FOLLOW - DO NOT CHANGE THE ORDER
                                ;
007474  177560                  DEV:    LSR                     ;LOW SPEED READER
007476  177550                          HSR                     ;HIGH SPEED READER

                                        .SBTTL  CASSETTE BOOT
```

```
1304
1305                                    ;
1306                                    ; CASSETTE BOOT
1307                                    ;
1308            177500                  TACS=177500                     ;TA-11 CONTROL AND STATUS REGISTER
1309                                    ;       .=ORIGIN+1500
1310  007500  012700  177500            TABOOT: MOV     #TACS,R0
1311  007504  005010                            CLR     (R0)            ;SELECT UNIT #0
1312  007506  010701                    RES:    MOV     PC,R1           ;USE FOR PIC
1313  007510  062701  000052                    ADD     #TABLE-.,R1     ;R1 HOLDS ADDR. OF COMMAND TABLE
1314  007514  012702  000375                    MOV     #375,R2         ;MEMORY PTR. AND DATA FLAG
1315  007520  112103                            MOVB    (R1)+,R3        ;TEST BITS
1316                                    ;
1317  007522  112110                    LOOP1:  MOVB    (R1)+,(R0)      ;COMMAND FROM TABLE TO TACS
1318  007524  100413                            BMI     DONE            ;WHEN COMMAND CODE NEG., QUIT
1319  007526  130310                    LOOP2:  BITB    R3,(R0)         ;TEST READY AND T-REG BITS IN TACS
1320  007530  001776                            BEQ     LOOP2           ;LOOP 'TIL SOMETHING COMES UP
1321  007532  105202                            INCB    R2              ;ADVANCE MEMORY POINTER
1322  007534  100772                            BMI     LOOP1           ;IF MINUS, TRY NEXT COMMAND
1323  007536  116012  000002                    MOVB    2(R0),(R2)      ;READ DATA INTO MEMORY
1324  007542  120337  000000                    CMPB    R3,@#0          ;FIRST BYTE READ SHOULD BE '240'
1325  007546  001767                            BEQ     LOOP2           ;IF O.K., GO READ ANOTHER BYTE
1326  007550  000000                    STOP:   HALT                    ;HALT ON ERROR
1327  007552  000755                            BR      RES             ;RESTART ON CONTINUE
1328                                    ;
1329  007554  005710                    DONE:   TST     (R0)            ;CHECK FOR ERROR
1330  007556  100774                            BMI     STOP            ;HALT ON ERROR
1331  007560  005007                            CLR     PC              ;= 'JMP @#0'
1332                                    ;
1333  007562  017640                    TABLE:  .WORD   17640           ;.BYTE 240: READY+T-REQ.
1334                                                                    ;.BYTE 37:  ILBS+READY+GO
1335  007564  002415                            .WORD   2415            ;.BYTE  15: SFB+GO
1336                                                                    ;.BYTE   5: READ+GO
1337  007566  112024                            .WORD   112024          ;.BYTE  24: READ+ILBS
1338                                                                    ;.BYTE 224: READ+ILBS+E.O.TABLE
1339  007570  000000  000000                    .WORD   0,0             ;THESE ARE FILLER WORDS
1340  007574  167500                            .WORD   TABOOT!160000           ;POWER UP VECTOR AND PRIORITY
1341  007576  000340                            .WORD   340             ;
1342
1343                                    .SBTTL  MR11-DB BOOT
```

```
1344                                    ;MR11-DB BULK STORAGE PROGRAM LOADER LISTING
1345
1346                                    :       .=ORIGIN+1600                   ;KEEP TRACK OF ORIGIN
1347
1348  007600  010702                    RF11:   MOV PC,R2               ;FIXED HEAD DISK (256 KW)
1349  007602  000451                            BR OTHER
1350  007604  177462                            177462
1351  007606  000005                            5
1352
1353  007610  010702                    RK11:   MOV PC,R2               ;MOVING HEAD DISK (CARTRIDGE)
1354  007612  000445                            BR OTHER
1355  007614  177406                            177406
1356  007616  000005                            5
1357
1358
1359  007620  010702                    TC11:   MOV PC,R2
1360  007622  000417                            BR TAPES
1361  007624  177344                            177344                  ;ADDRESS OF WORD COUNT
1362  007626  000005                            5                       ;LAST COMMAND
1363  007630  004003                            4003                    ;FIRST COMMAND
1364  007632  100000                            100000                  ;DONE MASK
1365  007634  024000                            24000                   ;ERROR MASK
1366
1367
1368  007636  010702                    TM11:   MOV PC,R2
1369  007640  000410                            BR TAPES
1370  007642  172524                            172524                  ;ADDRESS OF BYTE COUNT
1371  007644  060003                            60003                   ;LAST COMMAND
1372  007646  060011                            60011                   ;FIRST COMMAND
1373  007650  000200                            200                     ;DONE MASK
1374  007652  100000                            100000                  ;ERROR MASK
1375
1376
1377  007654  010702                    RP11:   MOV PC,R2               ;MOVING HEAD DISK (PACK)
1378  007656  000423                            BR OTHER
1379  007660  176716                            176716
1380
1381
1382  007662  000005                    TAPES:  RESET
1383  007664  010200                            MOV R2,R0               ;GET THE ADDRESS OF THE BRANCH
1384  007666  005720                            TST (0)+                ;R0 TO POINT AT LAST COMMAND
1385  007670  012001                            MOV (0)+,R1             ;GET THE WORD COUNT ADDRESS
1386  007672  005311                            DEC (1)                 ;SET UP FOR ADVANCE 1 RECORD
1387  007674  005720                            TST (0)+                ;MOVE R0 TO FIRST COMMAND
1388  007676  012041                            MOV (0)+,-(1.           ;COMMAND WORD TO COMMAND REG.
1389  007700  031011                            BIT (0),(1)             ;LOOK FOR DONE INDICATORS
1390  007702  001776                            BEQ .-2                 ;NONE SET, TRY AGAIN
1391  007704  005720                            TST (0)+                ;DONE FIRST COMMAND, CHECK FOR ERROR
1392  007706  031041                            BIT (0),-(1)            ;LOOK FOR SET ERROR BITS
1393  007710  001406                            BEQ OTHER               ;NO ERRORS - TRY THE READ
1394  007712  000112                    AGAIN:  JMP (2)                 ;RERUN FOR ERRORS
1395
1396
1397  007714  167600                    RFVEC:  RF11!160000                     ;RF11 POWER UP VECTOR
1398  007716  000340                            340
1399
```

```
1400  007720  010702          RC11:   MOV PC,R2               ;FIXED HEAD DISK (64Kw
1401  007722  000401                  BR OTHER
1402  007724  177450                  177450                  ;ADRS OF WORD COUNT (COMMAND-2
1403                                                          ;COMMAND WORD (5) IS THE RESET
1404
1405  007726  000005          OTHER:  RESET
1406  007730  010200                  MOV R2,R0               ;R0 TO POINT AT WORD COUNT ADRS
1407  007732  005720                  TST (0)+                ;POINT TO ADDRESS
1408  007734  012001                  MOV (0)+,R1             ;WORD COUNT ADDRESS TO R1
1409  007736  012711  177000          MOV #-1000,(1)          ;LOAD WORD COUNT
1410  007742  011041                  MOV (0),-(1)            ;COMMAND TO COMMAND REGISTER
1411  007744  032711  100200          BIT #100200,(1)         ;CHECK FOR ERROR OR DONE
1412  007750  001775                  BEQ .-4                 ;IF NEITHER, KEEP LOOKING
1413  007752  100757                  BMI AGAIN               ;ERROR, TRY AGAIN
1414  007754  005007                  CLR PC
1415
1416  007756  000000                  0                       ;FILLER
1417  007760  167610          RKVEC:  RK11!160000                     ;RK POWER UP VECTOR
1418  007762  000340                  340
1419  007764  167720          RCVEC:  RC11!160000                     ;RC POWER UP VECTOR
1420  007766  000340                  340
1421  007770  167654          RPVEC:  RP11!160000                     ;RP POWER UP VECTOR
1422  007772  000340                  340
1423  007774  167620          TCVEC:  TC11!160000                     ;TC11 POWER UP VECTOR
1424  007776  000340                  340
1425
1426                                  .SBTTL  ROM VERSION 1 VALUES
1427                                  .PAGE
```

```
                                        .DSABL  AMA

                                :DATA PATTERN STORED IN THE GT40 BOOTSTRAP VERSION 1

                                ;       ************ THIS IS A IMAGE LISTING OF THE GT40 (VT40) BOOTSTRAP ********
                                ;
                                ;       THE DATA IS A MIRROR IMAGE OF THAT IN THE BOOTSTRAP ROMS
                                ;       ONLY THE ADDRESS FIELD IS CHANGED

                                ;BOOTVT.SO9  5/2/72      (SPECIAL)


                                ;       VT-40 BOOTSTRAP LOADER, VERSION SO9, RELEASE R01, 5/2/72

                                ;       COPYRIGHT 1972, DIGITAL EQUIPMENT CORPORATION.
                                ;       146 MAIN STREET
                                ;       MAYNARD, MASSACHUSSETTS
                                ;                               01754


                                ;       WRITTEN BY JACK BURNESS, SENIOR SYSTEMS ARCHITECT!



                                ;       THIS ROUTINE IS INTENDED TO BE LOADED IN THE ROM PORTION OF THE VT-40.


                                ;       REGISTER DEFINITIONS:

000000                          R0=%0
000001                          R1=%1
000002                          R2=%2
000003                          R3=%3
000004                          R4=%4
000005                          R5=%5
000006                          R6=%6
000007                          R7=%7

000006                          SP=R6
000007                          PC=R7

000000                          RET1=R0                 ;RETURN OF VALUE REGISTER.
000001                          INP1=R1                 ;ARGUMENT FOR CALLED FUNCTION
000002                          INP2=R2                 ;SECOND ARGUMENT.
000003                          WORK1=R3                ;FIRST WORK REGISTER.
000004                          WORK2=R4                ;SECOND WORKING REGISTER.
000005                          SCR1=R5                 ;SCRATCH REGISTER.

000003                          LCKSM=WORK1             ;OVERLAPPING DEFINITIONS FOR LOADER PORTION.
000000                          LBYT=RET1
000005                          LBC=SCR1
000001                          LADR=INP1
```

```
1484
1485            036000                  COREND=36000                    ;FIRST LOCATION OF NON-CORE.
1486            166000                  ROMORG=166000                   ;WHERE THE ROM PROGRAM SHOULD GO.
1487
1488            000000                  STARTX=0                        ;WHERE TO START DISPLAYING THE X POSITIONS.
1489            001360                  STARTY=1360                     ;WHERE TO START DISPLAYING THE Y.
1490
1491
1492            022000                  VT40PC=172000-150000                    ;VT40 PROGRAM COUNTER.
1493            027560                  KBDIS=27560                     ;TTY INPUT STATUS.
1494            025614                  P10OS=25614                     ;PDP-10 OUTPUT STATUS.
1495            025610                  P10IS=25610                     ;PDP-10 INPUT STATUS.
1496
1497            027562                  KBDIB=KBDIS+2                   ;TTY INPUT BUFFER.
1498            025612                  P10IB=P10IS+2                   ;PDP-10 INPUT CHARACTER.
1499            025616                  P10OB=P10OS+2                   ;PDP-10 OUTPUT BUFFER.
1500
1501
1502            045776                  P10OC=COREND-2+10000                    ;CHARACTER TO BE SENT TO THE PDP-10
1503            045772                  P10IC=P10OC-4                   ;INPUT CHARACTER FROM 10 PLUS ONE SAVE CHARACTER
1504            015770                  STKSRT=P10IC-2-30000                    ;FIRST LOCATION OF STACK.
1505
1506
1507            160000                  JMPDIS=160000                   ;THE VT-40 DISPLAY JUMP INSTRUCTION.
1508
1509
1510            000024                  PWRFAL=24                       ;POWER FAIL RESTART LOCATION.
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522            016000                          .=16000
1523                            ;               .=ROMORG                        ;SET THE ORIGIN NOW!!!!
1524
1525
1526
1527
1528
1529
1530
1531  016000  012705  000026        STARTA: MOV     #PWRFAL+2,SCR1          ;PICK UP POINTER TO P.F. STATUS.
1532  016004  005015                        CLR     @SCR1                   ;CLEAR IT OUT TO BE SURE.
1533  016006  010745                        MOV     PC,-(SCR1)              ;SET UP THE RESTART LOCATION.
1534
1535  016010  000005                        RESET                           ;RESET THE BUS.
1536
1537  016012  012767  000007  007570        MOV     #7,P10IS                ;INITIALIZE PDP-10 INPUT
1538  016020  012767  000001  011532        MOV     #1,KBDIS                ;INITIALIZE TTY INPUT.
1539  016026  012767  000201  007560        MOV     #201,P10OS              ;INITIALIZE PDP-10 OUTPUT.
```

```
1540
1541
1542
1543  016034  012706  015770      RSTPT:  MOV     #STKSRT,SP              ;SET UP THE STACK NOW!
1544  016040  005001                      CLR     LADR                    ;CLEAR ADDRESS POINTER.
1545  016042  012702  160000              MOV     #JMPDIS,INP2            ;PLACE A DISPLAY JUMP INSTRUCTION IN A REGISTER.
1546  016046  010221                      MOV     INP2,(LADR)+            ;MOVE IT TO LOCATION 0.
1547  016050  012711  166756              MOV     #DISPRG+150000,(LADR)           ;MOVE ADDRESS POINTER INTO 2.
1548  016054  012701  000030              MOV     #PWRFAL+4,LADR          ;SET UP WHERE WE WILL STORE CHARACTERS.
1549  016060  005000                      CLR     RET1                    ;PREPARE TO INSERT A ZERO CHARACTER.
1550  016062  004767  000022              JSR     PC,DOCHAR               ;INSERT IT NOW.
1551  016066  005067  003706              CLR     VT40PC                  ;CLEAR THE DISPLAY PROGRAM COUNTER AND START.
1552
1553  016072  004767  000210      MAJOR:  JSR     PC,GTCHR                ;GT A CHARACTER NOW.
1554  016076  000240                      NOP
1555  016100  000240                      NOP
1556  016102  000240                      NOP
1557  016104  012746  166072              MOV     #MAJOR+150000,-(SP)             ;INSERT IN DISPLAY BUFFER NOW.
1558
1559  016110  010105      DOCHAR: MOV     LADR,SCR1               ;GT CURRENT BUUFER POSITION NOW.
1560  016112  022525              CMP     (SCR1)+,(SCR1)+         ;BYPASS CURRENT DISPLAY JUMP.
1561  016114  005025              CLR     (SCR1)+                 ;CLEAR FUTURE ADDRESS FOR JUMP.
1562  016116  010225              MOV     INP2,(SCR1)+            ;STICK IN TEMPORARY JUMP WHILE WE REPLACE CURREN
1563  016120  005015              CLR     (SCR1)                  ;A DISPLAY JUMP TO ZERO.
1564  016122  005011              CLR     (LADR)                  ;NOW REPLACE CURRENT DISPLAY JUMP BY THE CHARACT
1565  016124  050021              BIS     RET1,(LADR)+            ;IT'S DONE THIS WAY TO WASTE 2 CYCLES.
1566  016126  010211              MOV     INP2,(LADR)             ;TO AVOID TIMING PROBLEMS WITH THE VT40.
1567  016130  000207              RTS     PC                      ;AND FINALLY RETURN.
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584  016132  004767  000124      GT8:    JSR     PC,GTSIX                ;GT SIX BITS NOW.
1585  016136  010046                      MOV     RET1,-(SP)              ;SAVE THE CHARACTER NOW.
1586  016140  000401                      BR      GTP84                   ;BYPASS THE 8'ER
1587  016142  005002      GT84:   CLR     INP2                    ;RESET THE MAGIC REGISTER NOW.
1588  016144  005722      GTP84:  TST     (INP2)+                 ;INCREMENT WHERE TO GO.
1589  016146  066207  166250              ADD     GT8TB+150000(INP2),PC           ;UPDATE PC NOW.
1590
1591          016152      GT8P=.
1592
1593  016152  004767  000104      GT81:   JSR     PC,GTSIX                ;GT A CHARACTER NOW.
1594  016156  010004                      MOV     RET1,WORK2              ;SAVE FOR A SECOND.
1595  016160  006300                      ASL     RET1
```

```
1596  016162  006300                GT82:   ASL     RET1                    ;SHIFT TO LEFT OF BYTE
1597  016164  106300                        ASLB    RET1
1598  016166  106116                        ROLB    @SP                     ;PACK THEM IN.
1599  016170  106300                        ASLB    RET1
1600  016172  106116                        ROLB    @SP                     ;A GOOD 8 BIT THING.
1601  016174  012600                        MOV     (SP)+,RET1              ;POP AND RETURN NOW.
1602  016176  000207                        RTS     PC
1603
1604  016200  006300                GT82:   ASL     RET1                    ;WORST CASE. SHIFT 4
1605  016202  006300                        ASL     RET1
1606  016204  106300                        ASLB    RET1
1607  016206  106104                        ROLB    WORK2
1608  016210  106300                        ASLB    RET1
1609  016212  106104                        ROLB    WORK2
1610  016214  106300                        ASLB    RET1
1611  016216  106104                        ROLB    WORK2
1612  016220  106300                        ASLB    RET1
1613  016222  106104                        ROLB    WORK2
1614  016224  010400                        MOV     WORK2,RET1
1615  016226  012604                        MOV     (SP)+,WORK2
1616  016230  000207                        RTS     PC
1617
1618  016232  006100                GT83:   ROL     RET1
1619  016234  006100                        ROL     RET1
1620  016236  006004                        ROR     WORK2
1621  016240  106000                        RORB    RET1
1622  016242  006004                        ROR     WORK2
1623  016244  106000                        RORB    RET1                    ;FINAL CHARACTER ASSEMBLED.
1624  016246  005726                        TST     (SP)+                   ;FUDGE STACK.
1625  016250  000207                        RTS     PC                      ;AND RETURN NOW.
1626
1627          016250                GT8TB   =       .-2                     ;PUSH ZERO CONDITION BACK INTO NEVER-NEVER LAND.
1628
1629  016252  000000                        .WORD   GT81-GT8P
1630  016254  000026                        .WORD   GT82-GT8P
1631  016256  000060                        .WORD   GT83-GT8P
1632  016260  177770                        .WORD   GT84-GT8P
1633
1634
1635  016262  004767  000020        GTSIX:  JSR     PC,GTCHR
1636  016266  020027  000040                CMP     RET1,#40
1637  016272  002546                        BLT     LBAD
1638  016274  020027  000137                CMP     RET1,#137
1639  016300  003143                        BGT     LBAD
1640  016302  000207                        RTS     PC
1641
1642
1643
1644  016304  005726                GTCHP:  TST     (SP)+                   ;UPDATE THE STACK.
1645
1646  016306  012700  015772        GTCHR:  MOV     #P10IC-30000,RET1               ;SET UP POINTER TO THE INPUT CHARACTER.
1647  016312  004767  000064        GTCHL:  JSR     PC,CHECK
1648  016316  005710                        TST     @RET1                   ;ANY CHARACTERS THERE?
1649  016320  001774                        BEQ     GTCHL
1650  016322  011046                        MOV     @RET1,-(SP)             ;PUSH THE CHAR ON THE STACK.
1651  016324  005020                        CLR     (RET1)+                 ;CLEAR THE CHAR GOT FLAG NOW.
```

```
1652  016326  042716  177600              BIC     #-200,(SP)              ;CLEAR AWAY PARITY NOW.
1653  016332  001764                      BEQ     GTCHP                   ;IF ZERO, GT ANOTHER
1654  016334  022716  000177              CMP     #177,(SP)
1655  016340  001761                      BEQ     GTCHP                   ;ALSO IGNORE RUBOUTS.
1656  016342  022710  000175              CMP     #175,@RET1              ;WAS IT A "175"
1657  016346  001007                      BNE     GTNP                    ;NOPE.
1658  016350  011610                      MOV     (SP),@RET1              ;YEP. RESET IN CASE OF ABORT.
1659  016352  021027  000122              CMP     @RET1,#122              ;IS IT AN R
1660  016356  001626                      BEQ     RSTRT                   ;YEP. RESTART
1661  016360  021027  000114              CMP     @RET1,#114              ;IS IT AN L
1662  016364  001455                      BEQ     LOAD                    ;YEP. LOAD.
1663
1664  016366  011610          GTNP:       MOV     (SP),@RET1              ;NOW DO THE FDUGING.
1665  016370  012600                      MOV     (SP)+,RET1
1666  016372  020027  000175              CMP     RET1,#175
1667  016376  001743                      BEQ     GTCHR                   ;IF ALTMODE, LOOP
1668  016400  000207                      RTS     PC
1669
1670
1671
1672
1673
1674
1675
1676
1677  016402  005767  027370      CHECK:  TST     P100C                   ;DO WE WANT TO OUTPUT?
1678  016406  001410                      BEQ     CHECK1                  ;NO.
1679  016410  105767  007200              TSTB    P100S                   ;WE DO. IS THE 10 READY?
1680  016414  100005                      BPL     CHECK1                  ;NOT QUITE.
1681  016416  016767  027354  007172      MOV     P100C,P100B             ;IT'S READY. SEND THE CHARACTER.
1682  016424  005067  027346              CLR     P100C                   ;AND THE SAVED CHARACTER.
1683
1684  016430  105767  011124      CHECK1: TSTB    KBDIS                   ;HEY, IS THE KEYBOARD READY?
1685  016434  100014                      BPL     CHECK3                  ;NOPE. NO LUCK.
1686  016436  116746  011120              MOVB    KBDIB,-(SP)             ;YEP. SAVE THE CHARACTER NOW.
1687  016442  012767  000001  011110      MOV     #1,KBDIS                ;AND REENABLE THE COMMUNICATIONS DEVICE.
1688
1689  016450  004767  177726      CHECK2: JSR     PC,CHECK                ;IS THE OUTPUT READY?
1690  016454  005767  027316              TST     P100C
1691  016460  001373                      BNE     CHECK2                  ;IF NOT, WAIT TILL DONE.
1692  016462  012667  007130              MOV     (SP)+,P100B             ;AND THEN SEND OUT THE CHARACTER.
1693
1694
1695  016466  105767  007116      CHECK3: TSTB    P10IS                   ;IS THE 10 TALKING TO ME.
1696  016472  100011                      BPL     CHECK4                  ;NOPE. EXIT.
1697  016474  116767  007112  027270      MOVB    P10I8,P10IC             ;GT THE CHARACTER NOW.
1698  016502  052767  177400  027262      BIS     #-400,P10IC             ;MAKE SURE IT'S NONE ZERO.
1699  016510  012767  000007  007072      MOV     #7,P10IS                ;REINITIALIZE COMMUNICATION LINE.
1700
1701  016516  000207      CHECK4: RTS     PC                      ;AND RETURN.
1702
1703
1704
1705
1706
1707
```

```
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718                                    ;       THE    L  O  A  D  E  R
1719
1720  016520  005002                    LOAD:   CLR     INP2                    ;RESET TO FIRST 8 BIT CHARACTER.
1721  016522  012712  172000                    MOV     #172000,(INP2)          ;AND ALSO CLEVERLY STOP THE VT40.
1722  016526  012706  015770                    MOV     #STKSRT,SP              ;RESET STACK POINTER NOW.
1723
1724  016532  005003                    LLD2:   CLR     LCKSM                   ;CLEAR THE CHECKSUM
1725  016534  004767  000070                    JSR     PC,LPTR         ;GT A BYTE NOW.
1726  016540  105300                            DECB    LBYT                    ;IS IT ONE?
1727  016542  001373                            BNE     LLD2                    ;NOPE. WAIT AWHILE
1728  016544  004767  000060                    JSR     PC,LPTR         ;YEP. GT NEXT CHARACTER.
1729
1730  016550  004767  000072                    JSR     PC,LGWRD                ;GT A WORD.
1731  016554  010005                            MOV     LBYT,LBC                ;GT THE COUNTER NOW.
1732  016556  162705  000004                    SUB     #4,LBC                  ;CHOP OFF EXTRA STUFF.
1733  016562  022705  000002                    CMP     #2,LBC                  ;NULL?
1734  016566  001437                            BEQ     LJMP                    ;YEP. MUST BE END.
1735  016570  004767  000052                    JSR     PC,LGWRD                ;NOPE. GT THE ADDRESS.
1736  016574  010001                            MOV     LBYT,LADR               ;AND REMEMBER FOR OLD TIMES SAKE.
1737
1738  016576  004767  000026            LLD3:   JSR     PC,LPTR         ;GT A BYTE (DATA)
1739  016602  002010                            BGE     LLD4                    ;ALL DONE WITH THE COUNTER?
1740  016604  105703                            TSTB    LCKSM                   ;YEP. GOOD CHECK SUM?
1741  016606  001751                            BEQ     LLD2                    ;NOPE. LOAD ERROR.
1742
1743  016610  012700                    LBAD:   MOV     (PC)+,RET1              ;SEND OUT SOME CHARACTERS NOW.
1744                                    :       .BYTE   175,102                 ;"CTRL BAD"
1745  016612     102     175                    .BYTE   102,175                 ;"BAD CTRL"
1746  016614  004767  000110                    JSR     PC,SENDIT
1747  016620  000167  177210                    JMP     RSTRT
1748
1749  016624  110021                    LLD4:   MOVB    LBYT,(LADR)+            ;PLACE THE BYTE IN CORE.
1750  016626  000763                            BR      LLD3                    ;GT ANOTHER ONE.
1751
1752  016630  004767  177276            LPTR:   JSR     PC,GT8                  ;GT 8 BITS NOW.
1753  016634  060003                            ADD     LBYT,LCKSM              ;UPDATE CHECKSUM
1754  016636  042700  177400                    BIC     #177400,LBYT            ;CLEAN UP THE BYTE NOW.
1755  016642  005305                            DEC     LBC                     ;UPDATE THE COUNTER.
1756  016644  000207                            RTS     PC                      ;RETURN NOW.
1757
1758  016646  004767  177756            LGWRD:  JSR     PC,LPTR         ;GT A CHARACTER.
1759  016652  010046                            MOV     LBYT,-(SP)              ;SAVE FOR A SECOND.
1760  016654  004767  177750                    JSR     PC,LPTR         ;GT ANOTHER CHARACTER.
1761  016660  000300                            SWAB    LBYT                    ;NOW ASSEMBLE THE WORD.
1762  016662  052600                            BIS     (SP)+,LBYT              ;AND RETURN WITH A 16 BITER.
1763  016664  000207                            RTS     PC
```

```
1764
1765  016666  004767  177754          LJMP:   JSR     PC,LGWRD                ;GT A WORD
1766  016672  010046                          MOV     LBYT,-(SP)              ;SAVE ON THE STACK.
1767  016674  004767  177730                  JSR     PC,LPTR         ;GT A CHARCTER.
1768  016700  105703                          TSTB    LCKSM                   ;IS IT ZERO?
1769  016702  001342                          BNE     LBAD                    ;YEP. WHAT CRAP.
1770  016704  032716  000001                  BIT     #1,(SP)                 ;IS IT ODD?
1771  016710  001406                          BEQ     LJMP1                   ;YEP. START PROGRAM GOING NOW.
1772  016712  012700                          MOV     (PC)+,RET1              ;TELL PDP-10 WE'VE LOADED OK.
1773                                  :       .BYTE   175,107                 ;"CTRL GOOD"
1774  016714     107     175                  .BYTE   107,175                 ;"GOOD CTRL"
1775  016716  004767  000006                  JSR     PC,SENDIT
1776  016722  000000                          HALT
1777  016724  000776                          BR      .-2
1778
1779  016726  000136          LJMP1:  JMP     @(SP)+                  ;AND AWAY WE GO.
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797  016730  004767  177446          SENDIT: JSR     PC,CHECK                ;POLL THE OUTPUT DEVICE NOW.
1798  016734  005767  027036                  TST     P100C                   ;OUTPUT CLEAR?
1799  016740  001373                          BNE     SENDIT                  ;NOPE. LOOP AWHILE LONGER.
1800  016742  010067  006650                  MOV     RET1,P100B              ;SEND OUT THE CHARACTER.
1801  016746  105000                          CLRB    RET1                    ;CLEAR THE BYTE.
1802  016750  000300                          SWAB    RET1                    ;AND SWAP THEM NOW.
1803  016752  001366                          BNE     SENDIT                  ;IF NOT EQUAL. REPEAT.
1804  016754  000207                          RTS     PC
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
```

```
                                        ;
                                        ;       THIS IS THE INITIALIZING VT40 PROGRAM WHICH WI__
                                        ;       JUMP TO THE PROGRAM AFTER THE POWER FAIL LOCATIONS
                                        ;       WHICH WILL JUMP TO ZERO WHICH WILL JUMP BACK TO HERE.

016756  170256          DISPRG: .WORD   170256          ;LOAD STATUS REGISTER FOR NORMAL OPERATION.
016760  115124                  .WORD   115124          ;SET POINT MODE, "NORMAL".
016762  000000                  .WORD   STARTX          ;X COORDINATE
016764  021360                  .WORD   STARTY          ;Y COORDINATE
016766  100000                  .WORD   100000          ;SET CHARACTER MODE.
016770  160000                  .WORD   JMPDIS          ;THEN JUMP TO THE POWERFAIL LOCATION.
016772  000030                  .WORD   PWRFAL+4        ;TO DISPLAY USERS CHARACTERS.
016774  000000                  .WORD   0
016776  000000                  .WORD   0

000001                          .END
```

```
AGAIN   007712          1394#  1413
ALTMOD= 000176           597#   918   1078   1111
BELL    006250           738    770#
BLIMIT= 007000           590#   593    683    704    789    808    855    859
BLKOFF= 000020          1159#  1195
BLKON = 000030          1160#  1230
BSTART= 001000           589#   791    811    828    858
CHAR  = 100000           542#
CHARA   001020           177#   353*   354    360    376
CHECK   016402          1647   1677#  1689   1797
CHECK1  016430          1678   1680   1684#
CHECK2  016450          1689#  1691
CHECK3  016466          1685   1695#
CHECK4  016516          1696   1701#
CORENO= 036000          1485#  1502
CORSTR= 000004           592#   652
CR      006206           747#
CRLF  = 005015           596#   686
DEV     007474          1275   1300#
DEV1    007422          1278#  1280
DISIMP= 160000           599#   819    857    858    859    875
DISPRG  016756          1547   1830#
DISTOP= 173000           600#   927   1050
DUMP  = 160000          1167#  1262
DL11IB= 175612           578#   579    897
DL11IS= 175610           577#   578    645*   895    898*
DL1108= 175616           580#   905*   952*   953*
DL110S= 175614           579#   580    648*   666    668*
DNOP  = 164000          1168#
DOCHAR  016110          1550   1559#
DONE    007554          1317   1328#
DONE0   001422           291#
DONE1   001474           298    305#
DSR     001306           172#   290*
DSTOP = 173400          1170#
DSWR  = 177570           158#   163    443
DUMP    001016           176#   195*   198*   214*   216*   217    221*
D2BTYP  002114           322*   326*   403#   408
END     001410           289#
ENDCOR  006036           655#   656
ERROR1  001122           192    203#
ERROR2  001170           219#
ERRVEC= 000004           159#   441    442*   449*
FF      006256           743    773#
FFLOOP  006262           768    775#   778
FILCNT  001012           174#   365*   370*
FILEO   007214          1192   1195#  1263
FILLER  001010           173#   365
GETCHR  006564           726    917#   939
GETDL   006516           895#   900    904    906    917    921
GETDL1  006546           896    903#
GETEXT  006650           919    925    945#
GETSIX  006630           939#   977    992
GETB    006664           977#  1087
GETBTB  006700           980    982#
GETB1   006712           982    992#
```

GETB2	006740	983	1004#						
GETB3	006772	984	1019#						
GETB4	006706	988#							
GRAPHX=	120000	1144#	1244						
GRAPHY=	124000	1145#	1255						
GTBUSE	006424	828#	851						
GTCHL	016312	1647#	1649						
GTCHP	016304	1644#	1653	1655					
GTCHR	016306	1553	1635	1646#	1667				
GTNP	016366	1657	1664#						
GTP84	016144	1586	1588#						
GTSIX	016262	1584	1593	1635#					
GT40PC=	172000	585#	586	828*	862	1192*			
GT40SR=	172002	586#	770*						
GTB	016132	1584#	1752						
GTBP =	016152	1591#	1629	1630	1631	1632			
GTBTB =	016250	1589	1627#						
GTB1	016152	1593#	1629						
GTB2	016200	1604#	1630						
GTB3	016232	1618#	1631						
GTB4	016142	1587#	1632						
HEADER	006474	818	857	863	869#				
HSP =	177550	1269#	1301						
ICNT	001014	175#	184*	289*					
ICOUNT	002074	385	392#						
IMAGE	001004	171#	274	453*	456*				
INCR =	000100	1181#	1239						
INSERT	006350	751	759	797	805#				
INSRTL	006436	809	812	817#					
INSRTX	006422	807	822#						
INTX =	[illegible]	1182#	1200	1204	1208	1212			
INT0 =	[illegible]	1148#	1199						
INT1 =	002200	1149#	1218						
INT2 =	002400	1150#	1203						
INT3 =	002600	1151#	1220						
INT4 =	[illegible]	1152#	1207	1230					
INT5 =	003200	1153#	1222						
INT6 =	003400	1154#	1211						
INT7 =	003600	1155#	1224						
INTRL0 =	000040	1174#							
INTRL1 =	000060	1175#							
JMPADD=	007012	593#	785	796*	875	927*	1050*		
JMPCLS=	160000	1507#	1545	1835					
KBDIB =	027562	583#	905	1497#	1686				
KBDIS =	027560	582#	583	903	1493#	1497	1538*	1684	1687*
LBAC	016610	1637	1639	1743#	1769				
LF	006300	741	782#						
LFLOOP	006310	787#	790	792					
LFOUND	006330	788	794#						
LFSJE	006304	776	785#						
LGWRC	016646	1730	1735	1758#	1765				
LINE0 =	000004	1162#	1199						
LINE1 =	000005	1163#	1203						
LINE2 =	000006	1164#	1207						
LINE3 =	000007	1165#	1211						
LJMP	016666	1734	1765#						

```
LJMP1    016726       1771    1779#
LLD2     016532       1724#   1727    1741
LLD3     016576       1739#   1750
LLD4     016624       1739    1749#
LOAD     016520       1662    1720#
LOADER   007012        923    1050#
LONGV  = 110000       1142#   1199    1203    1207    1211
LOOP     007444       1287#   1294
LOOP1    007522       1316#   1321
LOOP2    007526       1318#   1319    1324
LPDARK = 000200       1173#
LPLITE = 000300       1172#
LPOFF  = 000100       1157#
LPON   = 000140       1158#
LPTR     016634       1725    1728    1738    1752#   1758    1760    1767
LSR    = 177560       1270#   1300
L.BAD    007100        941     943    1077#   1108
L.GWRD   007126       1062    1067    1094#   1104
L.HALT   007200       1116#
L.JMP    007146       1066    1104#
L.JMP1   007202       1114    1118#
L.LD2    007022       1055#   1058    1074
L.LD3    007066       1071#   1083
L.LD4    007110       1072    1082#
L.PTR    007114       1056    1060    1071    1087#   1094    1096    1106
MAJOR    016072       1553#   1557
MAXSX  = 017600       1187#
MAXSY  = 000077       1188#
MAXX   = 001777       1183#   1200    1208
MAXY   = 001377       1184#   1197    1205    1213
MINSUY = 000100       1189#
MINUSX = 020000       1185#   1186    1205    1208
MINUSY = 020000       1186#
M7       002222        316     433#
M8       002245        314     337     340     342     437#
M9       002251        325     438#
NORMAL   006212        730     739     751#
NOTHER   006042        653     659#
NUMLIN = 000040        594#    683     684     773     859
NXTCHR   006132        726#    728     734     753     763     771     779     782
ORIGIN = 166000        575#
OTHER    007726       1349    1354    1378    1393    1401    1405#
OUTLIT   006652        649     952#   1077    1110
O2A      002116        323     327     404#
O2AA     002146        413#    425
POINT  = 114000       1143#   1195    1215    1227    1236    1240    1251
PRESTR   006624        928#   1079
PRGO     001034        184#    305
PRGOR    001040        185#    186
PRG1     001500        166     310#    344
PRG1A    001536        318#
PRG1B    001554        322#    338
PRG1C    001572        326#    334
PRG1D    001642        332     339#
PRMTRS   001024        165     179#
PSW    = 177776        157#    396#
```

```
PTBOOT  007400          1273#
PWRFAL= 000024          1510#  1531    1548    1836
P1CIB = 025612          1498#  1697
P1DIC = 045772          1503#  1504    1646    1697*   1698*
P1DIS = 025610          1495#  1498    1537*   1695    1699*
P1DOB = 025616          1499#  1681*   1692*   1800*
P1DOC = 045776          1502#  1503    1677    1681    1682*   1690    1798
P1DOS = 025614          1494#  1499    1539*   1679
RCVEC   007764          1419#
RC11    007720          1400#  1419
RELATV= 130000          1146#  1230
RES     007506          1311#  1326
RESTRT  006060           680#   928
RETURN  002100           186*   390*    394#    397
RFVEC   007714          1397#
RF11    007630          1348#  1397
RKVEC   007760          1417#
RK11    007610          1353#  1417
ROMADD  001000           169#   190     211     243     256*    275     318
ROMORG= 166000          1486#
RPVEC   007770          1421#
RP11    007654          1377#  1421
RSTRT   016034          1543#  1660    1747
SCOPE = 104400           154#   206     236     264     287
SCOPEB  002102           382    388     395#
SCOPEC  002024           151    381#
SCOPEF  002076           385    387*    389*    393#
SCOPEG  002062           384    386     389#
SENDIT  016730          1746    1775    1797#   1799    1803
SETDUN  006126           695    704#
SETLP1  006074           686#   688
SETLP2  006110           694#   701
SETLP3  006116           698#   700
SETUP   006432           691    849#
SHORTV= 104000          1141#  1218    1220    1222    1224
START   006000           171    453     644#
STARTA  016000           456   1531#
STARTX= 000000          1488#  1832
STARTY= 001360          1489#  1833
STATSA= 170000          1169#
STATSB= 174000          1179#  1239
STKPTR= 000500           160#   179     185     242     310
STKSRT= 015770          1504#  1543    1722
STOP    007550          1325#  1329
SWITCH  002256           180    312     441#
SWR     000172           163#   381     383     443#    444     448#    450
SWREG   000170           162#   448
SYNCN = 000004          1176#
TAB     006222           740    758#    762
TABLE   007562          1312   1332#
TABOOT  007500          1309#  1339
TACS  = 177500          1307#  1309
TAPES   007662          1360   1369    1382#
TCVEC   007774          1423#
TC11    007620          1359#  1423
TERM    001022           178#   352*    354
```

```
TMPEND= 002776            591#    646     680
TM11    001636           1368#
TPCSR = 177564            155#    292     295     320     329     357     362     368     374     404     414     429#
TPDBR = 177566            156#    291#    294#    330#    364#    367#    372#    376#    416#
TYPEM   001662            313     315     324     336     339     341     350#
TYPEMA  001672            353#    357     377
TYPEMB  001720            355     360#
TYPEMC  002006            361     374#
T1      001052            190#
T1A     001070            193#    201     205
T1B     001130            202     206#
T2      001132            211#
T2A     001150            214#    220     235
T2B     001158            218     221#    224
T2C     001206            222     225#    228
T2D     001216            226     229#    232
T2E     001226            230     233#
T3      001236            242#
T3A     001260            246#    248
T3AA    001250            245#    262
T3B     001266            245     249#
T3C     001276            251#    253
T3D     001304            249     254#
T3E     001314            256#    258
T3F     001324            254     259#
T4      001344            273#
T4A     001360            276#    281     286
T4B     001374            279     283#
T4E     001406            287#
VT      006244            742     766#
VT40PC= 022000           1492#   1551#
WORDS   001002            170#    191     212     244     285     317     452#    455#
.       = 017000            149#    150#    161#    164#    168#    293     321     329     358     363     369     375     415
                          439#    630#   1290    1312    1390    1412    1522#   1591    1627    1777
```

```
COMMEN      1#
ENDCOM      1#
ESCAPE      1#
GETPRI      1#
GETSWR      1#
MULT        1#
NEWTST      1#
POP         1#
PUSH        1#
REPORT      1#
SETPRI      1#
SETUP       1#
SKIP        1#
SLASH       1#
STARS       1#
SWRSU       1#
TYPBIN      1#
TYPDEC      1#
TYPNAM      1#
TYPNUM      1#
TYPOCS      1#
TYPOCT      1#
TYPTXT      1#
$$ESCA      1#
$$NEWT      1#
$$SKIP      1#
.EQUAT      1#
.HEADE      1#
.KTII       1#
.SETUP      1#
.SWRHI      1#
.$ACTI      1#
.$APTB      1#
.$APTH      1#
.$APTY      1#
.$ASTA      1#
.$CATC      1#
.$CMTA      1#
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#
.$ERRO      1#
.$ERRT      1#
.$MULT      1#
.$POWE      1#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2C      1#
.$SCOP      1#
.$SIZE      1#
```

```
SCROLLING ROM BOOTSTRAP FOR THE GT40    MACY11 27(1006)  05-NOV-76  12:17  PAGE 50
DDGTDC.P11    15-SEP-76 00:00           CROSS REFERENCE TABLE -- MACRO NAMES

.$SUPR      1#
.$TRAP      1#
.$TYPB      1#
.$TYPC      1#
.$TYPE      1#
.$TYPO      1#
.$40CA      1#
.1170       1#


. ABS.  017000     000


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DDGTDC.BIN,DDGTDC.SEQ SOL/CRF/NL:TOC=DDGTDC.SML,DDGTDC.P11
 RUN-TIME:  23 28 1 SECONDS
 RUN-TIME RATIO: 189/53=3.5
 CORE USED:  32K  (63 PAGES)
```

JO4

Spooler runtime 6 Seconds, 30 KCS, 145 disk reads, 3 disk writes, 47 pages

[redacted]