GT40

ROM BOOTSTRAP

MD-11-DDGTD-B

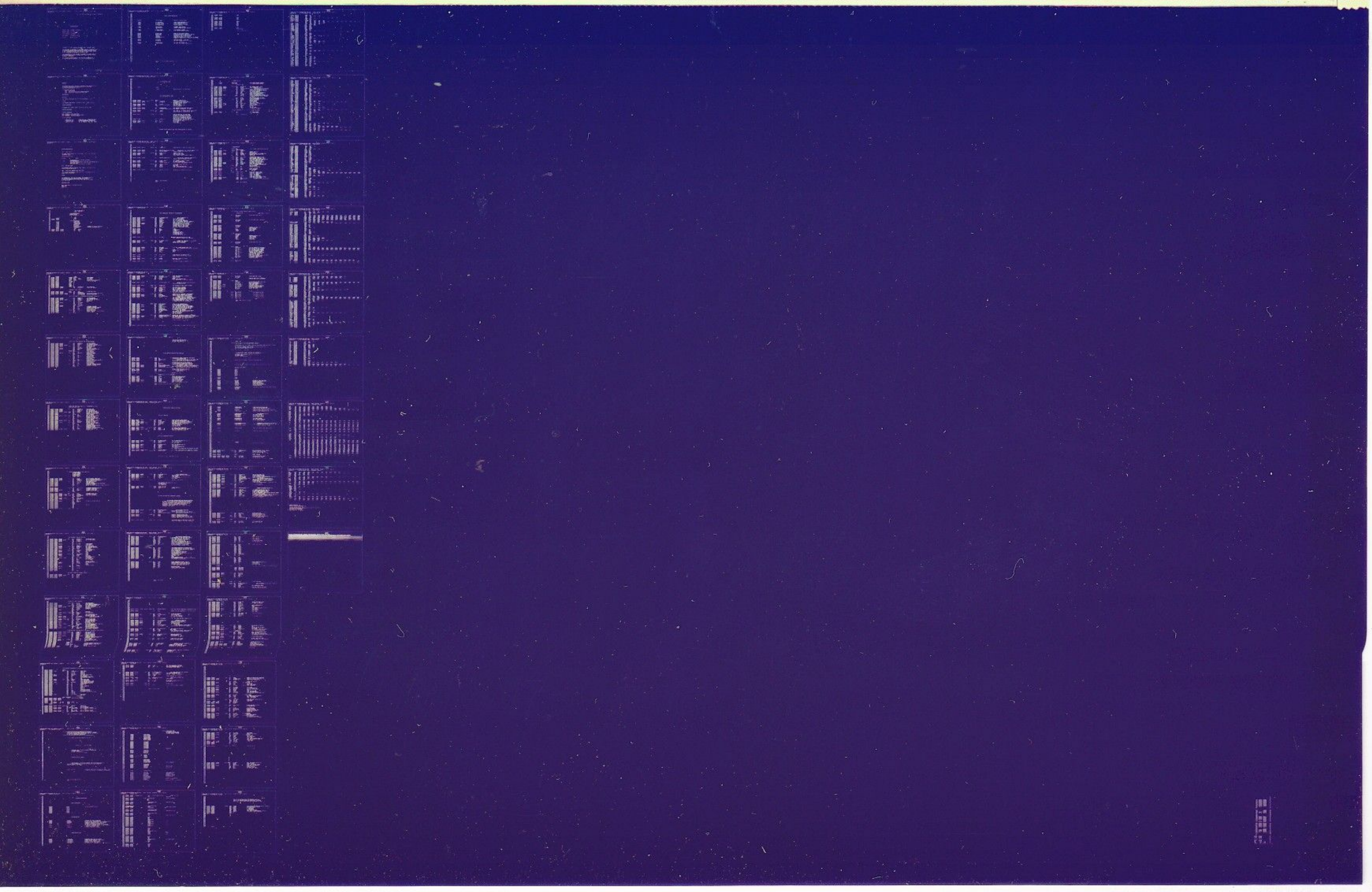
EP DDGTD B DL A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA



```
1                       .REM    ←                   .TITLE  GT40 ROM BOOTSTRAP TEST MAINDEC-11-DDGTD-B
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DDGTD-B-D
PRODUCT NAME: GT40 ROM VERIFY
DATE CREATED: NOVEMBER 1, 1973
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: RAYMOND SHOOP

1. ABSTRACT

THE DDGTD-B 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
- PRG2 - SINGLE ROM ADDRESS READ DATA LOOP

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.
0210 PROGRAM 2, SINGLE ROM READ.

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 PRG0 - 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.

5.3 PRG2 - SINGLE ROM ADDRESS READ DATA LOOP

THIS PROGRAM CONTINUOUSLY READS DATA FROM THE ADDRESS IN THE CONSOLE SWITCH REGISTER.

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

PRG0 TAKES APPROX. 5 SECONDS PER PASS.
PRG1 N/A
PRG2 N/A

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

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

                                        .LIST
            000034                      .=34
000034  002054                          $COPEC
000036  000000                          0
                                ;EQUATE STATEMENTS
            104400                      SCOPE=TRAP
            177564                      TPCSR=177564
            177566                      TPDBR=177566
            177776                      PSW=177776
            177570                      SR=177570
            177570                      DISPLAY=177570          ;ADDRESS OF DISPLAY REGISTER
            000500                      STKPTR=500              ;INITIAL STACK SETTING
            000200                      .=200
000200  000137  001024                  JMP     PRMTRS
000204  000137  001506                  JMP     PRG1
000210  000137  001670                  JMP     PRG2
```

```
                    001000                          .=1000
001000  166000                    ROMADD: 166000                  ;ROM ADDRESS
001002  001000                    WORDS:  512.    ;256.           ;ROM LENGTH
001004  006000                    IMAGE:  START   ;START@         ;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  002306                    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  002130            MOV     #PRG0R,RETURN   ;SET RETURN ADDRESS FOR SCOPE
001052  013737  001014  177570            MOV     ICNT,@#DISPLAY  ;DISPLAY PASS COUNT

                                ;TEST1 TEST ABILITY TO REFERENCE ROM WITHOUT TIMING OUT

001060  013700  001000            T1:     MOV     ROMADD,%0       ;GET ROM ADDRESS
001064  013701  001002                    MOV     WORDS,%1        ;GET ADDRESS COUNTER
001070  012737  001130  000004            MOV     #ERROR1,4       ;SET UP TIME OUT VECTOR
001076  011003                    T1A:    MOV     (0),%3          ;REFERENCE
001100  005720                            TST     (0)+            ;ROM
001102  064037  001016                    ADD     -(0),DUMP       ;
001106  021010                            CMP     (0),(0)         ;
001110  132020                            BITB    (0)+,(0)+       ;
001112  164037  001016                    SUB     -(0),DUMP
001116  062700  000002                    ADD     #2,%0           ;INCREMENT POINTER
001122  005301                            DEC     %1              ;DECREMENT ADDRESS COUNTER
001124  001364                            BNE     T1A             ;BRANCH IF NOT FINISHED
001126  000403                            BR      T1B             ;GO TO SCOPE LOOP
001130  022626                    ERROR1: CMP     (6)+,(6)+       ;REPOSITION STACK
001132  000000                            HALT                    ;ERROR, TIME-OUT  ON ROM ADDRESS
001134  000760                            BR      T1A             ;LOOP ON ERROR
001136  104400                    T1B:    SCOPE
```

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

001140  013700  001000        T2:     MOV     ROMADD,%0       ;GET ROM ADDRESS
001144  013701  001002                MOV     WORDS,%1        ;GET ADDRESS COUNTER
001150  012737  000006  000004        MOV     #6,4            ;INITIALIZE TIME OUT VECTOR
001156  005037  001016        T2A:    CLR     DUMP            ;INITIALIZE DUMP
001162  011003                        MOV     (0),%3          ;GET DATA
001164  062037  001016                ADD     (0)+,DUMP       ;ADD DATA TO DUMP
001170  163703  001016                SUB     DUMP,%3         ;SUBTRACT DATA FROM DATA
001174  001402                        BEQ     T2B             ;BRANCH IF EQUAL
001176  000000        ERROR2: HALT                            ;DATA ERROR
001200  000766                        BR      T2A             ;LOOP ON ERROR
001202  044037  001016        T2B:    BIC     -(0),DUMP       ;CLEAR DUMP BITS
001206  001402                        BEQ     T2C             ;BRANCH IF EQUAL TO 0
001210  000000                        HALT                    ;DATA ERROR
001212  000773                        BR      T2B             ;LOOP ON ERROR
001214  021010        T2C:    CMP     (0),(0)         ;COMPARE DATA
001216  001402                        BEQ     T2D             ;BRANCH IF EQUAL
001220  000000                        HALT                    ;DATA ERROR
001222  000774                        BR      T2C             ;LOOP ON ERROR
001224  122040        T2D:    CMPB    (0)+,-(0)       ;COMPARE DATA (BYTE OPERATION)
001226  001402                        BEQ     T2E             ;BRANCH IF EQUAL
001230  000000                        HALT                    ;DATA ERROR
001232  000774                        BR      T2D             ;LOOP ON ERROR
001234  005720        T2E:    TST     (0)+            ;INCREMENT ADDRESS POINTER
001236  005301                        DEC     %1              ;DECREMENT ADDRESS COUNTER
001240  001346                        BNE     T2A             ;RETURN IF NOT DONE
001242  104400                        SCOPE
```

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

001244  012706  000500        T3:     MOV     #STKPTR,%6      ;SET STACK PTR
001250  013700  001000                MOV     ROMADD,%0       ;GET ROM ADDRESS
001254  013701  001002                MOV     WORDS,%1        ;GET ADDRESS COUNTER
001260  012737  001274  000004 T3AA:   MOV     #T3B,4          ;SET UP TIME OUT VECTOR
001266  010010                T3A:    MOV     %0,(0)          ;ATTEMPT TO ALTER DATA
001270  000000                        HALT                    ;HERE IF DID NOT TIME OUT
001272  000775                        BR      T3A             ;LOOP ON ERROR
001274  012737  001312  000004 T3B:    MOV     #T3D,4          ;SET UP TIME OUT VECTOR
001302  022626                        CMP     (6)+,(6)+       ;REPOSITION STACK
001304  005210                T3C:    INC     (0)             ;ATTEMPT TO ALTER DATA
001306  000000                        HALT                    ;HERE IF DID NOT TIME OUT
001310  000775                        BR      T3C             ;LOOP ON ERROR
001312  012737  001332  000004 T3D:    MOV     #T3F,4          ;SET UP TIME OUT VECTOR
001320  022626                        CMP     (6)+,(6)+       ;REPOSITION STACK
001322  005077  177452        T3E:    CLR     @ROMADD         ;ATTEMPT TO ALTER DATA
001326  000000                        HALT                    ;HERE IF DID NOT TIME OUT
001330  000774                        BR      T3E             ;LOOP ON ERROR
001332  005720                T3F:    TST     (0)+            ;INCREMENT ADDRESS POINTER
001334  022626                        CMP     (6)+,(6)+       ;REPOSITION STACK
001336  005301                        DEC     %1              ;DECREMENT ADDRESS COUNTER
001340  001347                        BNE     T3AA            ;RETURN IF NOT DONE
001342  012737  000006  000004        MOV     #6,@#4          ;RESTORE TIME OUT TRAP
001350  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
001352  012700  000000          T4:     MOV     #0,%0           ;SET UP INITIAL WORD COUNT
001356  013701  001004                  MOV     IMAGE,%1        ;SET UP STARTING ADDRESS OF ROM IMAGE
001362  013703  001000                  MOV     ROMADD,%3       ;SET UP STARTING ROM ADDRESS
001366  011102                  T4A:    MOV     (%1),%2         ;READ EXPECTED VALUE
001370  011304                  T4A:    MOV     (%3),%4         ;READ ROM VALUE
001372  020204                          CMP     %2,%4           ;COMPARE EXPECTED TO THE VALUE READ
001374  001402                          BEQ     T4B             ;BRANCH IF CORRECT
001376  000000                          HALT                    ;ERROR, ROM VALUE FAILED TO EQUAL EXPECTED
001400  000772                          BR      T4A

001402  022123                  T4B:    CMP     (%1)+,(%3)+     ;INCREMENT ADDRESSES POINTERS
001404  005200                          INC     %0              ;INCREMENT WORD COUNT
001406  023700  001002                  CMP     WORDS,%0        ;COMPARE IF END WORD
001412  001365                          BNE     T4A             ;BRANCH IF NOT LAST WORD
001414  104400                  T4E:    SCOPE

001416  005237  001014          END:    INC     ICNT            ;INCREMENT PASS COUNT
001422  012777  000001  177356          MOV     #1,@DSR         ;RING THE GT40 BELL
001430  012737  000207  177566  DONE0:  MOV     #207,@#TPDBR    ;RING THE TELETYPE BELL
001436  105737  177564                  TSTB    @#TPCSR
001442  100375                          BPL     .-4
001444  012737  000207  177566          MOV     #207,@#TPDBR
001452  105737  177564          1$:     TSTB    @#TPCSR
001456  100375                          BPL     1$
001460  013700  000042                  MOV     @#42,%0         ;RETURN TO DECTAPE MONITOR?
001464  001406                          BEQ     DONE1
001466  000005                          RESET
001470  000005                          RESET
001472  004710                          JSR     7,(0)           ;RETURN!
001474  000240                          NOP
001476  000240                          NOP
001500  000240                          NOP
001502  000137  001034          DONE1:  JMP     PRG0
```

```
                                        ;THIS PROGRAM TYPES OUT ROM DATA
                                        
001506  012706  000500          PRG1:   MOV     #STKPTR,%6      ;INITIALIZE STACK
001512  012737  000006  000004          MOV     #6,@#4          ;SET UP BUSS ERROR
001520  004737  002306                  JSR     PC,SWITCH
001524  004537  001712                  JSR     5,TYPEM
001530  002275                          M8
001532  004537  001712                  JSR     5,TYPEM         ;TYPE MESSAGE
001536  002252                          M7                      ;'ROM DATA'
001540  013701  001002                  MOV     WORDS,%1        ;GET # OF WORDS
001544  013700  001000          PRG1A:  MOV     ROMADD,%0       ;GET STARTING ADDRESS
001550  012702  000010                  MOV     #10,%2          ;GET ADDRESS INDICATOR
001554  105737  177564                  TSTB    TPCSR           ;WAIT FOR
001560  100375                          BPL     .-4             ;TELEPRINTER FLAG
001562  010037  002144          PRG1B:  MOV     %0,D2BTYP       ;GET ADDRESS
001566  004737  002146                  JSR     7,02A           ;AND TYPE IT
001572  004537  001712                  JSR     5,TYPEM         ;TYPE
001576  002301                          M9                      ;CR/LF
001600  012037  002144          PRG1C:  MOV     (0)+,D2BTYP     ;TYPE
001604  004737  002146                  JSR     7,02A           ;DATA
001610  105737  177564                  TSTB    TPCSR           ;WAIT FOR
001614  100375                          BPL     .-4             ;TELEPRINTER FLAG
001616  012737  000040  177566          MOV     #' ,TPDBR       ;TYPE SPACE
001624  005301                          DEC     %1              ;ALL DATA TYPED
001626  001410                          BEQ     PRG1D           ;GO TO FINISH
001630  005302                          DEC     %2
001632  001362                          BNE     PRG1C           ;RETURN TO PRG1B
001634  012702  000010                  MOV     #10,%2          ;GET ADDRESS INDICATOR
001640  004537  001712                  JSR     5,TYPEM         ;TYPE
001644  002275                          M8                      ;CR/LF
001646  000745                          BR      PRG1B           ;RETURN TO PRG1B
001650  004537  001712          PRG1D:  JSR     5,TYPEM
001654  002275                          M8
001656  004537  001712                  JSR     5,TYPEM
001662  002275                          M8
001664  000000                          HALT
001666  000707                          BR      PRG1
                                        
                                        ;ROUTINE TO LOOP ON A SINGLE ADDRESS
                                        
001670  012706  000500          PRG2:   MOV     #STKPTR,%6
001674  012737  000006  000004          MOV     #6,@#4
001702  013700  177570                  MOV     SR,%0
001706  011001                          MOV     (0),%1
001710  000767                          BR      PRG2
```

```
                                        ;ROUTINE TO TYPE A MESSAGE
                                        
001712  010026                  TYPEM:  MOV     %0,(6)+         ;SAVE REGISTER 0
001714  012500                          MOV     (5)+,%0         ;PLACE MESSAGE ADDRESS IN R0
001716  112037  001022                  MOVB    (0)+,TERM       ;GET TERMINATOR CHARACTER
001722  112037  001020          TYPEMA: MOVB    (0)+,CHARA      ;GET NEXT CHARACTER
001726  123737  001020  001022        CMPB    CHARA,TERM      ;WAS NEXT CHARACTER THE TERM
001734  001005                        BNE     TYPEMB          ;CHARACTER
001736  014600                        MOV     -(6),%0         ;RESTORE R0
001740  105737  177564                TSTB    TPCSR
001744  100375                        BPL     .-4
001746  000205                        RTS     5               ;AND EXIT
001750  123727  001020  000045  TYPEMB: CMPB    CHARA,#'%       ;WAS CHARACTER %
001756  001027                        BNE     TYPEMC
001760  105737  177564                TSTB    TPCSR           ;TEST TELEPRINTER FLAG
001764  100375                        BPL     .-4             ;AND WAIT FOR DONE
001766  012737  000215  177566        MOV     #215,TPDBR      ;LOAD TELEPRINTER WITH CAR. RET
001774  013737  001010  001012        MOV     FILLER,FILCNT           ;LOAD FILLER COUNT
002002  000403                        BR      1$
002004  012737  000006  177566  2$:   MOV     #6,TPDBR        ;PRINT FILLER CHAR
002012  105737  177564          1$:   TSTB    TPCSR           ;TEST TELEPRINTER FLAG
002016  100375                        BPL     .-4             ;AND WAIT FOR DONE
002020  005337  001012                DEC     FILCNT          ;FINISHED FILLERS ?
002024  001367                        BNE     2$              ;BR IF NOT
002026  012737  000212  177566        MOV     #212,TPDBR      ;LOAD TELEPRINTER WITH LINE FEED
002034  000732                        BR      TYPEMA          ;GET NEXT CHARACTER
002036  105737  177564          TYPEMC: TSTB    TPCSR           ;TEST TELEPRINTER FLAG
002042  100375                        BPL     .-4             ;AND WAIT FOR DONE
002044  013737  001020  177566        MOV     CHARA,TPDBR     ;LOAD TELEPRINTER BUFFER
002052  000723                        BR      TYPEMA          ;AND GET NEXT CHARACTER

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

002054  032737  040000  177570  SCOPEC: BIT     #40000,SR       ;TEST SR FOR SCOPE
002062  001023                        BNE     SCOPEB          ;YES SCOPE
002064  032737  004000  177570        BIT     #4000,SR        ;TEST FOR ITERATION
002072  001007                        BNE     SCOPEG          ;INHIBIT ITERATION
002074  023737  002126  002124        CMP     SCOPEF,ICOUNT   ;ITERATION COMPLETE
002102  001403                        BEQ     SCOPEG          ;ITERATION COMPLETE GO TO SCOPEG
002104  005237  002126                INC     SCOPEF          ;INCREMENT ITERATION COUNT
002110  000410                        BR      SCOPEB          ;GO TO SCOPEB
002112  005037  002126  SCOPEG: CLR     SCOPEF          ;CLEAR ITERATION COUNT
002116  011637  002130        MOV     @%6,RETURN      ;GET ADDRESS OF NEXT TEST
002122  000002                RTI                     ;EXIT
002124  000100          ICOUNT: 100
002126  000000          SCOPEF: 0                       ;CONTAINS SUBTEST ITERATION COUNT
002130  000000          RETURN: .WORD   0               ;CONTAINS RETURN PC FOR SCOPE
002132  005726          SCOPEB: TST     (6)+            ;POP PC
002134  012637  177776        MOV     (6)+,PSW        ;RESTORE CONDITION CODES
002140  000177  177764        JMP     @RETURN
```

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

002144  000000                  D2BTYP: 0
002146  013746  177564          O2A:    MOV     TPCSR,-(6)      ;SAVE TPCSR
002152  010246                          MOV     %2,-(6)         ;SAVE R2
002154  010146                          MOV     %1,-(6)         ;SAVE R1
002156  010046                          MOV     %0,-(6)         ;SAVE R0
002160  013700  002144                  MOV     D2BTYP,%0       ;GET DATA TO BE TYPED
002164  012701  000006                  MOV     #6,%1           ;GET COUNTER
002170  005002                          CLR     %2              ;CLEAR WORKING REGISTER
002172  006100                          ROL     %0              ;MOV FIRST BIT (MSB) INTO
002174  006102                          ROL     %2              ;R2
002176  062702  000260          O2AA:   ADD     #260,%2         ;FORM ASCII CODE
002202  105737  177564                  TSTB    TPCSR           ;TEST TELEPRINTER
002206  100375                          BPL     .-4             ;FLAG AND WAIT UNTIL DONE
002210  010237  177566                  MOV     %2,TPDBR        ;LOAD TELEPRINTER BUFFER
002214  005002                          CLR     %2              ;CLEAR WORKING REGISTER
002216  006100                          ROL     %0              ;ROTATE THE
002220  006102                          ROL     %2              ;NEXT
002222  006100                          ROL     %0              ;OCTAL CHARACTER
002224  006102                          ROL     %2              ;INTO
002226  006100                          ROL     %0              ;REGISTER
002230  006102                          ROL     %2              ;TWO
002232  005301                          DEC     %1              ;DECREMENT COUNTER
002234  001360                          BNE     O2AA            ;GO TO O2AA IF NOT 0
002236  012600                          MOV     (6)+,%0         ;FINISHED.  RESTORE REGISTERS
002240  012601                          MOV     (6)+,%1         ;
002242  012602                          MOV     (6)+,%2         ;
002244  012637  177564                  MOV     (6)+,TPCSR      ;AND TPCSR
002250  000207                          RTS     7               ;AND EXIT

                                        ;ASCII MESSAGES
002252  022500  052107  032055  M7:     .ASCII  '@%GT-40 ROM DATA%%@'
002260  020060  047522  020115
002266  040504  040524  022445
002274      100
002275      100  022445      100  M8:     .ASCII  '@%%@'
002301      100  020040      100  M9:     .ASCII  '@  @'
            002306                      .EVEN

002306  032737  000400  177570  SWITCH: BIT     #400,SR         ;TEST BIT 8
002314  001007                          BNE     1$              ;BR IF VERSION 1
002316  012737  001000  001002          MOV     #512.,WORDS     ;SET UP VERSION 2 LENGTH
002324  012737  006000  001004          MOV     #START,IMAGE    ;SET UP VERSION 2 STARTING ADD.
002332  000406                          BR      2$
002334  012737  000400  001002  1$:     MOV     #256.,WORDS     ;SET UP VERSION 1 LENGTH
002342  012737  016000  001004          MOV     #STARTA,IMAGE   ;SET UP VERSION 1 STARTING ADD.
002350  000207                  2$:     RTS     PC

                                        .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 INSERTION 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.
007012                  JMPADD=BLIMIT+10.       ;WHERE THE POINTER IS TO FIRST CHAR ON SCREEN
000040                  NUMLIN=32.              ;NUMBER OF LINES ON 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  VTO5 SIMULATOR
.PAGE
```

```
                                ;               VT05 (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]
005202  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.
```

```
                                                ;MAKE THE CHARACTER A LINEFEED.
006262  012700  000012        FFLOOP: MOV     #12,CHAR
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        GET8:   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     GET8TB-2!160000(INDEX)          ;AND DISPATCH ACCORDING TO THE INDEX.
                                ;
006700  000404                GET8TB: 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.
                                ;
```

```
 989
 990  006712  004737  166630      GET81:  JSR     PC,GETSIX!160000        ;GET ANOTHER CHARACTER NOW.
 991  006716  010004                      MOV     L.BYT,HOLD              ;AND PRESERVE IT FOR NEXT TIME THROUGH.
 992  006720  006300                      ASL     L.BYT                   ;NOW THROW AWAY LEFT MOST BITS OF
 993  006722  006300                      ASL     L.BYT                   ;THE 8 BIT CHARACTER. NOW MERGE IN
 994  006724  106300                      ASLB    L.BYT                   ;THE LEFT TWO BITS OF THE
 995  006726  106116                      ROLB    (SP)                    ;NEW SIX BIT CHARACTER WITH THE SIX
 996  006730  106300                      ASLB    L.BYT                   ;BITS FROM THE CHARACTER ON THE
 997  006732  106116                      ROLB    (SP)                    ;STACK. 1ST CHARACTER IS NOW ASSEMBLED,
 998  006734  012600                      MOV     (SP)+,L.BYT             ;SO WE'LL RETURN IT TO THE USER.
 999  006736  000207                      RTS     PC                      ;AND THEN WE SHALL RETURN TO HIM.
1000
1001
1002  006740  006300      GET82:  ASL     L.BYT                   ;THE SECOND CHARACTER IS CREATED FROM
1003  006742  006300              ASL     L.BYT                   ;THE 4 RIGHT BITS OF THE PREVIOUS CHARACTER
1004  006744  106300              ASLB    L.BYT                   ;AND THE FOUR MIDDLE BITS OF THE PRESENT
1005  006746  106104              ROLB    HOLD                    ;8 BIT CHARACTER.
1006  006750  106300              ASLB    L.BYT                   ;WE WILL CREATE THE NEW 8 BIT
1007  006752  106104              ROLB    HOLD                    ;IN THIS REGISTER, SINCE IT
1008  006754  106300              ASLB    L.BYT                   ;MORE CONVIENT. WE WILL MOVE OVER THE
1009  006756  106104              ROLB    HOLD                    ;ANSWER AT THE END.
1010  006760  106300              ASLB    L.BYT                   ;ONE MORE TO GO
1011  006762  106104              ROLB    HOLD                    ;DONE.
1012  006764  010400              MOV     HOLD,L.BYT              ;BRING OVER THE VALUE.
1013  006766  012604              MOV     (SP)+,HOLD              ;AND REMEMBER THE LAST CHARACTER WE RECEIVED.
1014  006770  000207              RTS     PC                      ;AND RETURN TO THE CALLER.
1015
1016
1017  006772  006100      GET83:  ROL     L.BYT                   ;FINAL CHARACTER IS EASY. JUST A
1018  006774  106100              ROLB    L.BYT                   ;SIMPLE MERGER OF LEFT TWO BITS OF
1019  006776  006004              ROR     HOLD                    ;PREVIOUS VALUE WITH RIGHT SIX BITS
1020  007000  106000              RORB    L.BYT                   ;OF LAST (4TH) CHARACTER RECEIVED.
1021  007002  006004              ROR     HOLD
1022  007004  106000              RORB    L.BYT                   ;AND WE ARE DONE.
1023  007006  005726              TST     (SP)+                   ;FINALLY THROW AWAY STACK.
1024  007010  000207              RTS     PC                      ;AND RETURN TO THE CALLER.
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037                      .SBTTL  THE LOADER
1038                      .PAGE
```

```
1039
1040
1041
1042                                    ;               THE LOADER
1043                                    ;               --- ------
1044
1045
1046
1047
1048  007012  012737  173000  007010  LOADER: MOV     #DISTOP,JMPADD-2        ;STOP THE GT40 BY INSERTING A "DISTOP" IN THE LI
1049
1050  007020  005003                          CLR     INDEX                   ;RESET THE 8 BIT ASSEMBLER TO THE FIRST CHAR
1051
1052
1053  007022  005005                  L.LD2:  CLR     L.CKSM                  ;CLEAR THE CHECKSUM
1054  007024  004737  167114                  JSR     PC,L.PTR!160000         ;GET A BYTE NOW.
1055  007030  105300                          DECB    L.BYT                   ;IS IT A ONE (HEADER)?
1056  007032  001373                          BNE     L.LD2                   ;NO. WAIT FOR THE ONE.
1057
1058  007034  004737  167114                  JSR     PC,L.PTR!160000         ;YES. SKIP OVER THE NEXT CHARACTER NOW.
1059
1060  007040  004737  167126                  JSR     PC,L.GWRD!160000                ;ASSEMBLE A WORD NOW.
1061  007044  010002                          MOV     L.BYT,L.BC              ;MOVE OVER TO THE COUNTER.
1062  007046  162702  000004                  SUB     #4,L.BC                 ;REDUCE TO ACTUAL DATA COUNT.
1063  007052  022702  000002                  CMP     #2,L.BC                 ;ANY DATA AT ALL?
1064  007056  001433                          BEQ     L.JMP                   ;NO. MUST BE END
1065  007060  004737  167126                  JSR     PC,L.GWRD!160000                ;YES. ASSEMBLE A DATA WORD NOW.
1066  007064  010001                          MOV     L.BYT,L.ADR             ;AND THIS MUST BE THE FIRST ADDRESS.
1067
1068
1069  007066  004737  167114          L.LD3:  JSR     PC,L.PTR!160000         ;GET A BYTE OF DATA NOW.
1070  007072  002006                          BGE     L.LD4                   ;ALL DONE?
1071  007074  105705                          TSTB    L.CKSM                  ;YEP. COUNTER IS MINUS. CHECK CHECKSUM.
1072  007076  001751                          BEQ     L.LD2                   ;CHECKSUM GOOD. GET NEXT COMMAND.
1073
1074
1075  007100  004337  166652          L.BAD:  JSR     SCAN,OUTLIT!160000      ;BAD LOAD INFORM HOST
1076  007104     175     102                  .BYTE   ALTMOD,'B               ;SEND ALTMODE B
1077  007106  000646                          BR      PRESTRT                 ;AND RESTART THE DISPLAY.
1078
1079
1080  007110  110021                  L.LD4:  MOVB    L.BYT,(L.ADR)+          ;INSERT BYTE INTO MEMORY.
1081  007112  000765                          BR      L.LD3                   ;AND GET THE NEXT BYTE.
1082
1083
1084
1085  007114  004737  166664          L.PTR:  JSR     PC,GET8!160000                  ;ASSEMBLE AN 8 BIT CHARACTER NOW.
1086  007120  060005                          ADD     L.BYT,L.CKSM            ;UPDATE THE CHECKSUM NOW.
1087  007122  005302                          DEC     L.BC                    ;DECREMENT THE CHARACTER COUNTER.
1088  007124  000207                          RTS     PC                      ;AND RETURN TO THE CALLER NOW.
1089
1090
1091
1092  007126  004737  167114          L.GWRD: JSR     PC,L.PTR!160000         ;ASSEMBLE A WORD. FIRST GET A CHARACTER
1093  007132  010046                          MOV     L.BYT,-(SP)             ;AND SAVE IT.
1094  007134  004737  167114                  JSR     PC,L.PTR!160000         ;AND THEN GET ANOTHER ONE.
```

```
1095  007140  000300                    SWAB    L.BYT                   ;AND THEN REASSEMBLE THE MESS.
1096  007142  052600                    BIS     (SP)+,L.BYT             ;WITH THE FEARSOME POWER OF THE 11.
1097  007144  000207                    RTS     PC                      ;AND RETURN TO THE CALLER.
1098
1099
1100
1101
1102  007146  004737  167126  L.JMP:    JSR     PC,L.GWRD!160000                ;ALL DONE WITH THE LOAD. ASSEMBLE
1103  007152  010046                    MOV     L.BYT,-(SP)             ;THE STARTING ADDRESS NOW.
1104  007154  004737  167114            JSR     PC,L.PTR!160000         ;AND DON'T FORGET TO CHECKSUM IT.
1105  007160  105705                    TSTB    L.CKSM
1106  007162  001346                    BNE     L.BAD                   ;A BAD CHECKSUM. ALL IS EVIL.
1107
1108  007164  004337  166652            JSR     SCAN,OUTLIT!160000              ;GOOD CHKSUM,INFORM HOST
1109  007170     175     107            .BYTE   ALTMOD,'G               ;WITH ALTMOD G
1110
1111  007172  032716  000001            BIT     #1,(SP)                 ;DO WE WANT TO START EXECUTION?
1112  007176  001401                    BEQ     L.JMP1                  ;YES. AWAY WE GO.
1113
1114  007200  000000  L.HALT:  HALT                                    ;IF NOT, HALT.
1115
1116  007202  000136  L.JMP1:  JMP     @(SP)+                          ;IF GO, THEN GO ALREADY. WHEEEE!
1117
1118
1119
1120                            .SBTTL  THE SELF TEST
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133                    .PAGE
```

```
1134                                                    ;THIS IS GT40 QUICK TEST
1135                                                            ;GIVES QUICK VISUAL TEST
1136                                                            ;OF CONDITION OF MACHINE
1137                                                            ;WITHOUT READING IN DIAG.
1138            100000          CHAR=100000
1139            104000          SHORTV=104000
1140            110000          LONGV=110000
1141            114000          POINT=114000
1142            120000          GRAPHX=120000
1143            124000          GRAPHY=124000
1144            130000          RELATV=130000
1145
1146            002000          INT0=2000
1147            002200          INT1=2200
1148            002400          INT2=2400
1149            002600          INT3=2600
1150            003000          INT4=3000
1151            003200          INT5=3200
1152            003400          INT6=3400
1153            003600          INT7=3600                       ;BRIGHTEST
1154
1155            000100          LPOFF=100
1156            000140          LPON=140
1157            000020          BLKOFF=20
1158            000030          BLKON=30
1159
1160            000004          LINE0=4
1161            000005          LINE1=5
1162            000006          LINE2=6
1163            000007          LINE3=7
1164
1165            160000          DJMP=160000
1166            164000          DNOP=164000
1167            170000          STATSA=170000
1168            173400          DSTOP=173400                    ;STOP INTERRUPT
1169
1170            000300          LPLITE=300
1171            000200          LPDARK=200
1172            000040          ITAL0=40                        ;ITALICS OFF
1173            000060          ITAL1=60                        ;"       ON
1174            000004          SYNON=4                         ;SYNC ON
1175
1176
1177            174000          STATSB=174000
1178
1179            000100          INCR=100                        ;LOAD GRAPH INCR
1180            040000          INTX=40000                      ;INTENSIFY BIT
1181            001777          MAXX=1777                       ;BIGGEST X VECTOR
1182            001377          MAXY=1377                       ;BIGGEST Y VECTOR
1183            020000          MINUSX=20000                    ;THE MINUS BIT
1184            020000          MINUSY=MINUSX
1185            017600          MAXSX=17600                     ;BIGGEST X IN SHORTVEC
1186            000077          MAXSY=77                        ;"       Y IN   "
1187            000100          MINSUY=100                      ;MINUS BIT FOR Y IN SHORTVEC
1188
1189
```

```
1190  007204  012737  167214  172000          MOV     #FILE0!160000,GT40PC            ;START THE GT40
1191  007212  000001                          WAIT                                    ;AND WAIT
1192
1193  007214  114020                  FILE0:  POINT!BLKOFF                            ;POINT--INVISIBLE
1194  007216  000000                          0
1195  007220  001377                          MAXY
1196
1197  007222  112004                          LONGV!INT0!LINE0                        ;DRAW.TOP LINE
1198  007224  041777                          INTX!MAXX
1199  007226  000000                          0
1200
1201  007230  112405                          LONGV!INT2!LINE1
1202  007232  040000                          INTX                                    ;DRAW LINE TO RIGHT
1203  007234  021377                          MINUSX!MAXY
1204
1205  007236  113006                          LONGV!INT4!LINE2
1206  007240  061777                          INTX!MINUSX!MAXX                        ;DRAW BOTTOM LINE
1207  007242  000000                          0
1208
1209  007244  113407                          LONGV!INT6!LINE3
1210  007246  040000                          INTX
1211  007250  001377                          MAXY                                    ;DRAW LINE TO LEFT
1212
1213  007252  114000                          POINT
1214  007254  000400                          400
1215  007256  000500                          500
1216  007260  106200                          SHORTV!INT1
1217  007262  057677                          57677                                   ;+X+Y
1218  007264  106600                          SHORTV!INT3
1219  007266  077677                          77677                                   ;+X-Y
1220  007270  107200                          SHORTV!INT5
1221  007272  077777                          77777                                   ;-X-Y
1222  007274  107600                          SHORTV!INT7
1223  007276  057777                          57777                                   ;-X+Y
1224
1225  007300  114000                  POINT
1226  007302  001400                  1400
1227  007304  000500                  500
1228  007306  133030                  RELATV!INT4!BLKON
1229  007310  057677                  57677                                           ;+X+Y
1230  007312  077677                  77677                                           ;+X-Y
1231  007314  077777                  77777                                           ;-X-Y
1232  007316  057777                  57777                                           ;-X+Y
1233
1234  007320  114000                  POINT
1235  007322  000400                  400
1236  007324  000100                  100
1237  007326  174120                  STATSB!INCR+20                          ;TRY GRAPH MODES
1238  007330  114000                  POINT
1239  007332  001000                  1000
1240  007334  000200                  200
1241
1242  007336  120000                  GRAPHX
1243  007340  001010                  1010
1244  007342  001020                  1020
1245  007344  001030                  1030
```

```
1246  007346  001040                        1040
1247  007350  001050                        1050
1248
1249  007352  114000                        POINT
1250  007354  001000                        1000
1251  007356  001200                        1200
1252
1253  007360  124000                        GRAPHY
1254  007362  001020                        1020
1255  007364  001030                        1030
1256  007366  001040                        1040
1257  007370  001050                        1050
1258  007372  001060                        1060
1259
1260  007374  160000                        DJMP
1261  007376  167214                        FILEO!160000
1262
1263                                        .SBTTL  PAPER TAPE BOOT
```

```
1264                                    ;
1265                                    ; PAPER TAPE BOOT
1266                                    ;
1267            177550                  HSR=177550                      ;HIGH SPEED READER ADDRESS
1268            177560                  LSR=177560                      ;LOW SPEED READER ADDRESS
1269                                    ;       .=ORIGIN+1400
1270                                    ;
1271  007400   012701  160000           PTBOOT: MOV     #160000,R1      ;SET MEMORY CHECK LIMITS
1272  007404   012702  000004                   MOV     #4,R2           ;TRAP ADDRESS IS LOC. 4
1273  007410   012703  167500                   MOV     #DEV+4!160000,R3        ;POINTER TO DEVICE ADDRESSES
1274  007414   010712                           MOV     PC,@R2          ;PRESET TRAP ADDRESS IN LOC. 4
1275  007416   012706  000024                   MOV     #24,SP          ;STACK SET UP AT SPECIAL ADDRESS
1276  007422   014304                   DEV1:   MOV     -(R3),R4        ;GET DEVICE ADDRESS
1277  007424   005714                           TST     @R4             ;CHECK AVAILABILITY OF DEVICE
1278  007426   100775                           BMI     DEV1            ;CHECK DEVICE FOR ERRORS
1279  007430   010712                           MOV     PC,@R2          ;RESET TRAP ADDRESS AT LOC. 4
1280  007432   012706  000024                   MOV     #24,SP          ;SPECIAL ADDRESS USED AS MASK LATER
1281  007436   010441                           MOV     R4,-(R1)        ;DO MEM CHK:READER STATUS ADDRESS
1282                                                                    ;IS MOVED
1283  007440   040601                           BIC     SP,R1           ;SET R1=X7752,MASK IN SP=24
1284  007442   010111                           MOV     R1,@R1          ;STORE OWN ADDRESS IN POINTER
1285  007444   011102                   LOOP:   MOV     @R1,R2          ;GET BYTE POINTER
1286  007446   005214                           INC     @R4             ;ENABLE READER
1287  007450   105714                           TSTB    @R4             ;TEST DONE BIT
1288  007452   100376                           BPL     .-2             ;WAIT UNTIL READY
1289  007454   116412  000002                   MOVB    2(R4),@R2       ;THEN PICK IT UP AND STORE IT
1290  007460   005211                           INC     @R1             ;BUMP POINTER
1291  007462   120227  000375                   CMPB    R2,#375         ;STORED JUMP OFFSET?
1292  007466   001366                           BNE     LOOP            ;NOT YET
1293  007470   105222                           INCB    (R2)+           ;YES,ALL DONE
1294  007472   000142                           JMP     -(R2)           ;GO EXECUTE AS BRANCH
1295                                    ;
1296                                    ; DEVICE ADDRESSES FOLLOW - DO NOT CHANGE THE ORDER
1297                                    ;
1298  007474   177560                   DEV:    LSR                     ;LOW SPEED READER
1299  007476   177550                           HSR                     ;HIGH SPEED READER
1300
1301                                            .SBTTL  CASSETTE BOOT
```

```
1302                                    ;
1303                                    ; CASSETTE BOOT
1304                                    ;
1305            177500                  TACS=177500                     ;TA-11 CONTROL AND STATUS REGISTER
1306                                            .=ORIGIN+1500
1307  007500   012700  177580           TABOOT: MOV     #TACS,R0
1308  007504   005010                           CLR     (R0)            ;SELECT UNIT #0
1309  007506   010701                   RES:    MOV     PC,R1           ;USE FOR PIC
1310  007510   062701  000052                   ADD     #TABLE-.,R1     ;R1 HOLDS ADDR. OF COMMAND TABLE
1311  007514   012702  000375                   MOV     #375,R2         ;MEMORY PTR. AND DATA FLAG
1312  007520   112103                           MOVB    (R1)+,R3        ;TEST BITS
1313                                    ;
1314  007522   112110                   LOOP1:  MOVB    (R1)+,(R0)      ;COMMAND FROM TABLE TO TACS
1315  007524   100413                           BMI     DONE            ;WHEN COMMAND CODE NEG., QUIT
1316  007526   130310                   LOOP2:  BITB    R3,(R0)         ;TEST READY AND T-REQ BITS IN TACS
1317  007530   001776                           BEQ     LOOP2           ;LOOP 'TIL SOMETHING COMES UP
1318  007532   105202                           INCB    R2              ;ADVANCE MEMORY POINTER
1319  007534   100772                           BMI     LOOP1           ;IF MINUS, TRY NEXT COMMAND
1320  007536   116012  000002                   MOVB    2(R0),(R2)      ;READ DATA INTO MEMORY
1321  007542   120337  000000                   CMPB    R3,@#0          ;FIRST BYTE READ SHOULD BE '240'
1322  007546   001767                           BEQ     LOOP2           ;IF O.K., GO READ ANOTHER BYTE
1323  007550   000000                   STOP:   HALT                    ;HALT ON ERROR
1324  007552   000755                           BR      RES             ;RESTART ON CONTINUE
1325                                    ;
1326  007554   005710                   DONE:   TST     (R0)            ;CHECK FOR ERROR
1327  007556   100774                           BMI     STOP            ;HALT ON ERROR
1328  007560   005007                           CLR     PC              ;= 'JMP @#0'
1329                                    ;
1330  007562   017640                   TABLE:  .WORD   17640           ;.BYTE 240: READY+T-REQ.
1331                                            .WORD                   ;.BYTE 37:  ILBS+READY+GO
1332  007564   002415                           .WORD   2415            ;.BYTE  15: SFB+GO
1333                                            .WORD                   ;.BYTE   5: READ+GO
1334  007566   112024                           .WORD   112024          ;.BYTE  24: READ+ILBS
1335                                            .WORD                   ;.BYTE 224: READ+ILBS+E.O.TABLE
1336  007570   000000  000000                   .WORD   0,0             ;THESE ARE FILLER WORDS
1337  007574   167500                           .WORD   TABOOT!160000           ;POWER UP VECTOR AND PRIORITY
1338  007576   000340                           .WORD   340             ;
1339
1340
1341                                            .SBTTL  MR11-DB BOOT
```

```
1342                                    ;MR11-DB BULK STORAGE PROGRAM LOADER LISTING
1343
1344                                    ;       .=ORIGIN+1600                   ;KEEP TRACK OF ORIGIN
1345
1346  007600  010702                    RF11:   MOV PC,R2               ;FIXED HEAD DISK (256 KW)
1347  007602  000451                            BR OTHER
1348  007604  177462                            177462
1349  007606  000005                            5
1350
1351  007610  010702                    RK11:   MOV PC,R2               ;MOVING HEAD DISK (CARTRIDGE)
1352  007612  000445                            BR OTHER
1353  007614  177406                            177406
1354  007616  000005                            5
1355
1356
1357  007620  010702                    TC11:   MOV PC,R2
1358  007622  000417                            BR TAPES
1359  007624  177344                            177344              ;ADDRESS OF WORD COUNT
1360  007626  000005                            5                   ;LAST COMMAND
1361  007630  004003                            4003                ;FIRST COMMAND
1362  007632  100000                            100000              ;DONE MASK
1363  007634  024000                            24000               ;ERROR MASK
1364
1365
1366  007636  010702                    TM11:   MOV PC,R2
1367  007640  000410                            BR TAPES
1368  007642  172524                            172524              ;ADDRESS OF BYTE COUNT
1369  007644  060003                            60003               ;LAST COMMAND
1370  007646  060011                            60011               ;FIRST COMMAND
1371  007650  000200                            200                 ;DONE MASK
1372  007652  100000                            100000              ;ERROR MASK
1373
1374
1375  007654  010702                    RP11:   MOV PC,R2               ;MOVING HEAD DISK (PACK)
1376  007656  000423                            BR OTHER
1377  007660  176715                            176716
1378
1379
1380  007662  000005                    TAPES:  RESET
1381  007664  010200                            MOV R2,R0               ;GET THE ADDRESS OF THE BRANCH
1382  007666  005720                            TST (0)+                ;R0 TO POINT AT LAST COMMAND
1383  007670  012001                            MOV (0)+,R1             ;GET THE WORD COUNT ADDRESS
1384  007672  005311                            DEC (1)                 ;SET UP FOR ADVANCE 1 RECORD
1385  007674  005720                            TST (0)+                ;MOVE R0 TO FIRST COMMAND
1386  007676  012041                            MOV (0)+,-(1)           ;COMMAND WORD TO COMMAND REG.
1387  007700  031011                            BIT (0),(1)             ;LOOK FOR DONE INDICATORS
1388  007702  001776                            BEQ .-2                 ;NONE SET, TRY AGAIN
1389  007704  005720                            TST (0)+                ;DONE FIRST COMMAND, CHECK FOR ERROR
1390  007706  031041                            BIT (0),-(1)            ;LOOK FOR SET ERROR BITS
1391  007710  001406                            BEQ OTHER               ;NO ERRORS - TRY THE READ
1392  007712  000112                    AGAIN:  JMP (2)                 ;RERUN FOR ERRORS
1393
1394
1395  007714  167600                    RFVEC:  RF11!160000                     ;RF11 POWER UP VECTOR
1396  007716  000340                            340
1397
```

```
1398  007720  010702                  RC11:   MOV PC,R2               ;FIXED HEAD DISK (64KW)
1399  007722  000401                          BR OTHER
1400  007724  177450                          177450                  ;ADRS OF WORD COUNT (COMMAND+2)
1401                                                                  ;COMMAND WORD (5) IS THE RESET
1402
1403  007726  000005                  OTHER:  RESET
1404  007730  010200                          MOV R2,R0               ;R0 TO POINT AT WORD COUNT ADRS
1405  007732  005720                          TST (0)+                ;POINT TO ADDRESS
1406  007734  012001                          MOV (0)+,R1             ;WORD COUNT ADDRESS TO R1
1407  007736  012711  177000                 MOV #-1000,(1)          ;LOAD WORD COUNT
1408  007742  011041                          MOV (0),-(1)            ;COMMAND TO COMMAND REGISTER
1409  007744  032711  100200                 BIT #100200,(1)         ;CHECK FOR ERROR OR DONE
1410  007750  001775                          BEQ .-4                 ;IF NEITHER, KEEP LOOKING
1411  007752  100757                          BMI AGAIN               ;ERROR, TRY AGAIN
1412  007754  005007                          CLR PC
1413
1414  007756  000000                          0                       ;FILLER
1415  007760  167610                  RKVEC:  RK11!160000                     ;RK POWER UP VECTOR
1416  007762  000340                          340
1417  007764  167720                  RCVEC:  RC11!160000                     ;RC POWER UP VECTOR
1418  007766  000340                          340
1419  007770  167654                  RPVEC:  RP11!160000                     ;RP POWER UP VECTOR
1420  007772  000340                          340
1421  007774  167620                  TCVEC:  TC11!160000                     ;TC11 POWER UP VECTOR
1422  007776  000340                          340
1423
1424                                          .SBTTL  ROM VERSION 1 VALUES
1425                                          .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.S09  5/2/72      (SPECIAL)


                ;       VT-40 BOOTSTRAP LOADER, VERSION S09, 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
```

```
1482
1483            036000                  COREND=36000                    ;FIRST LOCATION OF NON-CORE.
1484            166000                  ROMORG=166000                   ;WHERE THE ROM PROGRAM SHOULD GO.
1485
1486            000000                  STARTX=0                        ;WHERE TO START DISPLAYING THE X POSITIONS.
1487            001360                  STARTY=1360                     ;WHERE TO START DISPLAYING THE Y.
1488
1489
1490            022000                  VT40PC=172000-150000                    ;VT40 PROGRAM COUNTER.
1491            027560                  KBDIS=27560                     ;TTY INPUT STATUS.
1492            025614                  P100S=25614                     ;PDP-10 OUTPUT STATUS.
1493            025610                  P10IS=25610                     ;PDP-10 INPUT STATUS.
1494
1495            027562                  KBDIB=KBDIS+2                   ;TTY INPUT BUFFER.
1496            025612                  P10IB=P10IS+2                   ;PDP-10 INPUT CHARACTER.
1497            025616                  P100B=P100S+2                   ;PDP-10 OUTPUT BUFFER.
1498
1499
1500            045776                  P100C=COREND-2+10000                    ;CHARACTER TO BE SENT TO THE PDP-10
1501            045772                  P10IC=P100C-4                   ;INPUT CHARACTER FROM 10 PLUS ONE SAVE CHARACTER
1502            015770                  STKSRT=P10IC-2-30000                    ;FIRST LOCATION OF STACK.
1503
1504
1505            160000                  JMPDIS=160000                   ;THE VT-40 DISPLAY JUMP INSTRUCTION.
1506
1507
1508            000024                  PWRFAL=24                       ;POWER FAIL RESTART LOCATION.
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520            016000                  .=16000
1521                            ;       .=ROMORG                        ;SET THE ORIGIN NOW!!!!
1522
1523
1524
1525
1526
1527
1528
1529    016000  012705  000026  STARTA: MOV     #PWRFAL+2,SCR1          ;PICK UP POINTER TO P.F. STATUS.
1530    016004  005015                  CLR     @SCR1                   ;CLEAR IT OUT TO BE SURE.
1531    016006  010745                  MOV     PC,-(SCR1)              ;SET UP THE RESTART LOCATION.
1532
1533    016010  000005                  RESET                           ;RESET THE BUS.
1534
1535    016012  012767  000007  007570  MOV     #7,P10IS                ;INITIALIZE PDP-10 INPUT
1536    016020  012767  000001  011532  MOV     #1,KBDIS                ;INITIALIZE TTY INPUT.
1537    016026  012767  000201  007560  MOV     #201,P100S              ;INITIALIZE PDP-10 OUTPUT.
```

```
1538
1539
1540
1541  016034  012706  015770        RSTRT:  MOV     #STKSRT,SP              ;SET UP THE STACK NOW!
1542  016040  005001                        CLR     LADR                    ;CLEAR ADDRESS POINTER.
1543  016042  012702  160000                MOV     #JMPDIS,INP2            ;PLACE A DISPLAY JUMP INSTRUCTION IN A REGISTER.
1544  016046  010221                        MOV     INP2,(LADR)+            ;MOVE IT TO LOCATION 0.
1545  016050  012711  166756                MOV     #DISPRG+150000,(LADR)           ;MOVE ADDRESS POINTER INTO 2.
1546  016054  012701  000030                MOV     #PWRFAL+4,LADR          ;SET UP WHERE WE WILL STORE CHARACTERS.
1547  016060  005000                        CLR     RET1                    ;PREPARE TO INSERT A ZERO CHARACTER.
1548  016062  004767  000022                JSR     PC,DOCHAR               ;INSERT IT NOW.
1549  016066  005067  003706                CLR     VT40PC                  ;CLEAR THE DISPLAY PROGRAM COUNTER AND START.
1550
1551  016072  004767  000210        MAJOR:  JSR     PC,GTCHR                ;GT A CHARACTER NOW.
1552  016076  000240                        NOP
1553  016100  000240                        NOP
1554  016102  000240                        NOP
1555  016104  012746  166072                MOV     #MAJOR+150000,-(SP)             ;INSERT IN DISPLAY BUFFER NOW.
1556
1557  016110  010105        DOCHAR: MOV     LADR,SCR1               ;GT CURRENT BUUFER POSITION NOW.
1558  016112  022525                        CMP     (SCR1)+,(SCR1)+         ;BYPASS CURRENT DISPLAY JUMP.
1559  016114  005025                        CLR     (SCR1)+                 ;CLEAR FUTURE ADDRESS FOR JUMP.
1560  016116  010225                        MOV     INP2,(SCR1)+            ;STICK IN TEMPORARY JUMP WHILE WE REPLACE CURREN
1561  016120  005015                        CLR     (SCR1)                  ;A DISPLAY JUMP TO ZERO.
1562  016122  005011                        CLR     (LADR)                  ;NOW REPLACE CURRENT DISPLAY JUMP BY THE CHARACT
1563  016124  050021                        BIS     RET1,(LADR)+            ;IT'S DONE THIS WAY TO WASTE 2 CYCLES.
1564  016126  010211                        MOV     INP2,(LADR)             ;TO AVOID TIMING PROBLEMS WITH THE VT40.
1565  016130  000207                        RTS     PC                      ;AND FINALLY RETURN.
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582  016132  004767  000124          GT8:    JSR     PC,GTSIX                ;GT SIX BITS NOW.
1583  016136  010046                          MOV     RET1,-(SP)              ;SAVE THE CHARACTER NOW.
1584  016140  000401                          BR      GTP84                   ;BYPASS THE 8'ER
1585  016142  005002                  GT84:   CLR     INP2                    ;RESET THE MAGIC REGISTER NOW.
1586  016144  005722                  GTP84:  TST     (INP2)+                 ;INCREMENT WHERE TO GO.
1587  016146  066207  166250                  ADD     GT8TB+150000(INP2),PC           ;UPDATE PC NOW.
1588
1589          016152                  GT8P=.
1590
1591  016152  004767  000104          GT81:   JSR     PC,GTSIX                ;GT A CHARACTER NOW.
1592  016156  010004                          MOV     RET1,WORK2              ;SAVE FOR A SECOND.
1593  016160  006300                          ASL     RET1
```

```
1594  016162  006300                    ASL     RET1                    ;SHIFT TO LEFT OF BYTE
1595  016164  106300                    ASLB    RET1
1596  016166  106116                    ROLB    @SP                     ;PACK THEM IN.
1597  016170  106300                    ASLB    RET1
1598  016172  106116                    ROLB    @SP                     ;A GOOD 8 BIT THING.
1599  016174  012600                    MOV     (SP)+,RET1              ;POP AND RETURN NOW.
1600  016176  000207                    RTS     PC
1601
1602  016200  006300            GT82:   ASL     RET1                    ;WORST CASE. SHIFT 4
1603  016202  006300                    ASL     RET1
1604  016204  106300                    ASLB    RET1
1605  016206  106104                    ROLB    WORK2
1606  016210  106300                    ASLB    RET1
1607  016212  106104                    ROLB    WORK2
1608  016214  106300                    ASLB    RET1
1609  016216  106104                    ROLB    WORK2
1610  016220  106300                    ASLB    RET1
1611  016222  106104                    ROLB    WORK2
1612  016224  010400                    MOV     WORK2,RET1
1613  016226  012604                    MOV     (SP)+,WORK2
1614  016230  000207                    RTS     PC
1615
1616  016232  006100            GT83:   ROL     RET1
1617  016234  006100                    ROL     RET1
1618  016236  006004                    ROR     WORK2
1619  016240  106000                    RORB    RET1
1620  016242  006004                    ROR     WORK2
1621  016244  106000                    RORB    RET1                    ;FINAL CHARACTER ASSEMBLED.
1622  016246  005726                    TST     (SP)+                   ;FUDGE STACK.
1623  016250  000207                    RTS     PC                      ;AND RETURN NOW.
1624
1625          016250            GT8TB   =       .-2                     ;PUSH ZERO CONDITION BACK INTO NEVER-NEVER LAND.
1626
1627  016252  000000                    .WORD   GT81-GT8P
1628  016254  000026                    .WORD   GT82-GT8P
1629  016256  000060                    .WORD   GT83-GT8P
1630  016260  177770                    .WORD   GT84-GT8P
1631
1632
1633  016262  004767  000020    GTSIX:  JSR     PC,GTCHR
1634  016266  020027  000040            CMP     RET1,#40
1635  016272  002546                    BLT     LBAD
1636  016274  020027  000137            CMP     RET1,#137
1637  016300  003143                    BGT     LBAD
1638  016302  000207                    RTS     PC
1639
1640
1641
1642  016304  005726            GTCHP:  TST     (SP)+                   ;UPDATE THE STACK.
1643
1644  016306  012700  015772    GTCHR:  MOV     #P10IC-30000,RET1               ;SET UP POINTER TO THE INPUT CHARACTER.
1645  016312  004767  000064    GTCHL:  JSR     PC,CHECK
1646  016316  005710                    TST     @RET1                   ;ANY CHARACTERS THERE?
1647  016320  001774                    BEQ     GTCHL
1648  016322  011046                    MOV     @RET1,-(SP)             ;PUSH THE CHAR ON THE STACK.
1649  016324  005020                    CLR     (RET1)+                 ;CLEAR THE CHAR GOT FLAG NOW.
```

```
1650  016326  042716  177600          BIC     #-200,(SP)              ;CLEAR AWAY PARITY NOW.
1651  016332  001764                  BEQ     GTCHP                   ;IF ZERO, GT ANOTHER
1652  016334  022716  000177          CMP     #177,(SP)
1653  016340  001761                  BEQ     GTCHP                   ;ALSO IGNORE RUBOUTS.
1654  016342  022710  000175          CMP     #175,@RET1              ;WAS IT A "175"
1655  016346  001007                  BNE     GTNP                    ;NOPE.
1656  016350  011610                  MOV     (SP),@RET1              ;YEP. RESET IN CASE OF ABORT.
1657  016352  021027  000122          CMP     @RET1,#122              ;IS IT AN R
1658  016356  001626                  BEQ     RSTRT                   ;YEP. RESTART
1659  016360  021027  000114          CMP     @RET1,#114              ;IS IT AN L
1660  016364  001455                  BEQ     LOAD                    ;YEP. LOAD.
1661
1662  016366  011610          GTNP:   MOV     (SP),@RET1              ;NOW DO THE FDUGING.
1663  016370  012600                  MOV     (SP)+,RET1
1664  016372  020027  000175          CMP     RET1,#175
1665  016376  001743                  BEQ     GTCHR                   ;IF ALTMODE, LOOP
1666  016400  000207                  RTS     PC
1667
1668
1669
1670
1671
1672
1673
1674
1675  016402  005767  027370  CHECK:  TST     P100C                   ;DO WE WANT TO OUTPUT?
1676  016406  001410                  BEQ     CHECK1                  ;NO.
1677  016410  105767  007200          TSTB    P100S                   ;WE DO. IS THE 10 READY?
1678  016414  100005                  BPL     CHECK1                  ;NOT QUITE.
1679  016416  016767  027354  007172  MOV     P100C,P100B             ;IT'S READY. SEND THE CHARACTER.
1680  016424  005067  027346          CLR     P100C                   ;AND THE SAVED CHARACTER.
1681
1682  016430  105767  011124  CHECK1: TSTB    KBDIS                   ;HEY, IS THE KEYBOARD READY?
1683  016434  100014                  BPL     CHECK3                  ;NOPE. NO LUCK.
1684  016436  116746  011120          MOVB    KBDIB,-(SP)             ;YEP. SAVE THE CHARACTER NOW.
1685  016442  012767  000001  011110  MOV     #1,KBDIS                ;AND REENABLE THE COMMUNICATIONS DEVICE.
1686
1687  016450  004767  177726  CHECK2: JSR     PC,CHECK                ;IS THE OUTPUT READY?
1688  016454  005767  027316          TST     P100C
1689  016460  001373                  BNE     CHECK2                  ;IF NOT, WAIT TILL DONE.
1690  016462  012667  007130          MOV     (SP)+,P100B             ;AND THEN SEND OUT THE CHARACTER.
1691
1692
1693  016466  105767  007116  CHECK3: TSTB    P10IS                   ;IS THE 10 TALKING TO ME.
1694  016472  100011                  BPL     CHECK4                  ;NOPE. EXIT.
1695  016474  116767  007112  027270  MOVB    P10IB,P10IC             ;GT THE CHARACTER NOW.
1696  016502  052767  177400  027262  BIS     #-400,P10IC             ;MAKE SURE IT'S NONE ZERO.
1697  016510  012767  000007  007072  MOV     #7,P10IS                ;REINITIALIZE COMMUNICATION LINE.
1698
1699  016516  000207          CHECK4: RTS     PC                      ;AND RETURN.
1700
1701
1702
1703
1704
1705
```

```
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716                                    ;       THE     L O A D E R
1717
1718  016520  005002                    LOAD:   CLR     INP2                    ;RESET TO FIRST 8 BIT CHARACTER.
1719  016522  012712  172000                    MOV     #172000,(INP2)          ;AND ALSO CLEVERLY STOP THE VT40.
1720  016526  012706  015770                    MOV     #STKSRT,SP              ;RESET STACK POINTER NOW.
1721
1722  016532  005003                    LLD2:   CLR     LCKSM                   ;CLEAR THE CHECKSUM
1723  016534  004767  000070                    JSR     PC,LPTR         ;GT A BYTE NOW.
1724  016540  105300                            DECB    LBYT                    ;IS IT ONE?
1725  016542  001373                            BNE     LLD2                    ;NOPE. WAIT AWHILE
1726  016544  004767  000060                    JSR     PC,LPTR         ;YEP. GT NEXT CHARACTER.
1727
1728  016550  004767  000072                    JSR     PC,LGWRD                ;GT A WORD.
1729  016554  010005                            MOV     LBYT,LBC                ;GT THE COUNTER NOW.
1730  016556  162705  000004                    SUB     #4,LBC                  ;CHOP OFF EXTRA STUFF.
1731  016562  022705  000002                    CMP     #2,LBC                  ;NULL?
1732  016566  001437                            BEQ     LJMP                    ;YEP. MUST BE END.
1733  016570  004767  000052                    JSR     PC,LGWRD                ;NOPE. GT THE ADDRESS.
1734  016574  010001                            MOV     LBYT,LADR               ;AND REMEMBER FOR OLD TIMES SAKE.
1735
1736  016576  004767  000026          LLD3:   JSR     PC,LPTR         ;GT A BYTE (DATA)
1737  016602  002010                            BGE     LLD4                    ;ALL DONE WITH THE COUNTER?
1738  016604  105703                            TSTB    LCKSM                   ;YEP. GOOD CHECK SUM?
1739  016606  001751                            BEQ     LLD2                    ;NOPE. LOAD ERROR.
1740
1741  016610  012700                    LBAD:   MOV     (PC)+,RET1              ;SEND OUT SOME CHARACTERS NOW.
1742                                    ;       .BYTE   175,102                 ;"CTRL BAD"
1743  016612     102     175                    .BYTE   102,175                 ;"BAD CTRL"
1744  016614  004767  000110                    JSR     PC,SENDIT
1745  016620  000167  177210                    JMP     RSTRT
1746
1747  016624  110021                    LLD4:   MOVB    LBYT,(LADR)+            ;PLACE THE BYTE IN CORE.
1748  016626  000763                            BR      LLD3                    ;GT ANOTHER ONE.
1749
1750  016630  004767  177276          LPTR:   JSR     PC,GT8                  ;GT 8 BITS NOW.
1751  016634  060003                            ADD     LBYT,LCKSM              ;UPDATE CHECKSUM
1752  016636  042700  177400                    BIC     #177400,LBYT            ;CLEAN UP THE BYTE NOW.
1753  016642  005305                            DEC     LBC                     ;UPDATE THE COUNTER.
1754  016644  000207                            RTS     PC                      ;RETURN NOW.
1755
1756  016646  004767  177756          LGWRD:  JSR     PC,LPTR         ;GT A CHARACTER.
1757  016652  010046                            MOV     LBYT,-(SP)              ;SAVE FOR A SECOND.
1758  016654  004767  177750                    JSR     PC,LPTR         ;GT ANOTHER CHARACTER.
1759  016660  000300                            SWAB    LBYT                    ;NOW ASSEMBLE THE WORD.
1760  016662  052600                            BIS     (SP)+,LBYT              ;AND RETURN WITH A 16 BITER.
1761  016664  000207                            RTS     PC
```

```
1762
1763  016666  004767  177754          LJMP:   JSR     PC,LGWRD                ;GT A WORD
1764  016672  010046                          MOV     LBYT,-(SP)              ;SAVE ON THE STACK.
1765  016674  004767  177730                  JSR     PC,LPTR         ;GT A CHARCTER.
1766  016700  105703                          TSTB    LCKSM                   ;IS IT ZERO?
1767  016702  001342                          BNE     LBAD                    ;YEP. WHAT CRAP.
1768  016704  032716  000001                  BIT     #1,(SP)                 ;IS IT ODD?
1769  016710  001406                          BEQ     LJMP1                   ;YEP. START PROGRAM GOING NOW.
1770  016712  012700                          MOV     (PC)+,RET1              ;TELL PDP-10 WE'VE LOADED OK.
1771                                  ;       .BYTE   175,107                 ;"CTRL GOOD"
1772  016714     107     175                  .BYTE   107,175                 ;"GOOD CTRL"
1773  016716  004767  000006                  JSR     PC,SENDIT
1774  016722  000000                          HALT
1775  016724  000776                          BR      .-2
1776
1777  016726  000136          LJMP1:  JMP     @(SP)+                  ;AND AWAY WE GO.
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795  016730  004767  177446          SENDIT: JSR     PC,CHECK                ;POLL THE OUTPUT DEVICE NOW.
1796  016734  005767  027036                  TST     P100C                   ;OUTPUT CLEAR?
1797  016740  001373                          BNE     SENDIT                  ;NOPE. LOOP AWHILE LONGER.
1798  016742  010067  006650                  MOV     RET1,P100B              ;SEND OUT THE CHARACTER.
1799  016746  105000                          CLRB    RET1                    ;CLEAR THE BYTE.
1800  016750  000300                          SWAB    RET1                    ;AND SWAP THEM NOW.
1801  016752  001366                          BNE     SENDIT                  ;IF NOT EQUAL, REPEAT.
1802  016754  000207                          RTS     PC
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
```

```
1818
1819
1820
1821                                    ;               THIS IS THE INITIALIZING VT40 PROGRAM WHICH WILL
1822                                    ;               JUMP TO THE PROGRAM AFTER THE POWER FAIL LOCATIONS
1823                                    ;               WHICH WILL JUMP TO ZERO WHICH WILL JUMP BACK TO HERE.
1824
1825
1826
1827
1828  016756  170256                    DISPRG: .WORD   170256          ;LOAD STATUS REGISTER FOR NORMAL OPERATION.
1829  016760  115124                            .WORD   115124          ;SET POINT MODE, "NORMAL".
1830  016762  000000                            .WORD   STARTX          ;X COORDINATE
1831  016764  001360                            .WORD   STARTY          ;Y COORDINATE
1832  016766  100000                            .WORD   100000          ;SET CHARACTER MODE.
1833  016770  160000                            .WORD   JMPDIS          ;THEN JUMP TO THE POWERFAIL LOCATION.
1834  016772  000030                            .WORD   PWRFAL+4        ;TO DISPLAY USERS CHARACTERS.
1835  016774  000000                            .WORD   0
1836  016776  000000                            .WORD   0
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850          000001                            .END
```

```
AGAIN   007712          1392#  1411
ALTMOD= 000175           595#   916   1076   1109
BELL    006250           736    768#
BLIMIT= 007000           588#   591    681    702    787    806    853    857
BLKOFF= 000020          1157#  1193
BLKON = 000030          1158#  1228
BSTART= 001000           587#   789    809    826    856
CHAR  = 100000           538#   558    725    727    729    756*   773*   785    796*   803    895*   897*   916
                         920    922    938    940   1138#
CHARA   001020           178#   360*   361    367    383
CHECK   016402          1645   1675#  1687   1795
CHECK1  016430          1676   1678   1682#
CHECK2  016450          1687#  1689
CHECK3  016466          1683   1693#
CHECK4  016516          1694   1699#
COREND= 036000          1483#  1500
CORSTR= 000004           590#   650
COUNTR=%000005           545#   561    764*   771*   775*
CR      006206           745#
CRLF  = 005015           594#   684
DEV     007474          1273   1298#
DEV1    007422          1276#  1278
DISJMP= 160000           597#   817    855    856    857    873
DISPLA= 177570           162#   188*
DISPRG  016756          1545   1828#
DISTOP= 173000           598#   925   1048
DJMP  = 160000          1165#  1260
DL11IB= 175612           576#   577    895
DL11IS= 175610           575#   576    643*   893    896*
DL110B= 175616           578#   903*   950*   951*
DL110S= 175614           577#   578    646*   664    666*
DNOP  = 164000          1166#
DOCHAR  016110          1548   1557#
DONE    007554          1315   1326#
DONE0   001430           293#
DONE1   001502           300    307#
DSR     001006           173#   292*
DSTOP = 173400          1168#
DUMP    001016           177#   197*   200*   216*   218*   219    223*
D2BTYP  002144           324*   328*   410#   415
END     001416           291#
ENDCOR  006036           653#   654
ERROR1  001130           194    205#
ERROR2  001176           221#
FF      006256           741    771#
FFLOOP  006262           766    773#   776
FILCNT  001012           175#   372*   377*
FILE0   007214          1190   1193#  1261
FILLER  001010           174#   372
GETCHR  006564           724    915#   937
GETDL   006516           893#   898    902    904    915    919
GETDL1  006546           894    901#
GETEXT  006650           917    923    943#
GETSIX  006630           937#   975    990
GETB    006664           975#  1085
GETBTB  006700           978    980#
```

```
GET81   006712            980    990#
GET82   006740            981   1002#
GET83   006772            982   1017#
GET84   006706            986#
GRAPHX= 120000           1142#  1242
GRAPHY= 124000           1143#  1253
GTBUSE  006424            826#   849
GTCHL   016312           1645#  1647
GTCHP   016304           1642#  1651   1653
GTCHR   016306           1551   1633   1644#  1665
GTNP    016366           1655   1662#
GTP84   016144           1584   1586#
GTSIX   016262           1582   1591   1633#
GT40PC= 172000            583#   584    826*   860   1190*
GT40SR= 172002            584#   768*
GTB     016132           1582#  1750
GTBP  = 016152           1589#  1627   1628   1629   1630
GTBTB = 016250           1587   1625#
GTB1    016152           1591#  1627
GTB2    016200           1602#  1628
GTB3    016232           1616#  1629
GTB4    016142           1585#  1630
HEADER  006474            816    855    861    867#
HOLD  =%000004            543#   991*  1005*  1007*  1009*  1011*  1012  1013*  1019*  1021*
HSR   = 177550           1267#  1299
ICNT    001014            176#   185*   188    291*
ICOUNT  002124            392    399#
IMAGE   001004            172#   276    451*   454*
INCR  = 000100           1179#  1237
INDEX =%000003            562#   977    978    986*  1050*
INP1  =%000001           1469#  1478
INP2  =%000002           1470#  1543*  1544   1560   1564   1585*  1586  1587  1718*  1719*
INSERT  006350            749    757    795    803#
INSRTL  006406            807    810    815#
INSRTX  006422            805    820#
INTX  = 040000           1180#  1198   1202   1206   1210
INTO  = 002000           1146#  1197
INT1  = 002200           1147#  1216
INT2  = 002400           1148#  1201
INT3  = 002600           1149#  1218
INT4  = 003000           1150#  1205   1228
INT5  = 003200           1151#  1220
INT6  = 003400           1152#  1209
INT7  = 003600           1153#  1222
ITALO = 000040           1172#
ITAL1 = 000060           1173#
JMPADD= 007012            591#   783    794*   873    925*  1048*
JMPDIS= 160000           1505#  1543   1833
KBDIB = 027562            581#   903   1495#  1684
KBDIS = 027560            580#   581    901   1491#  1495  1536*  1682  1685*
LADR  =%000001           1478#  1542*  1544*  1545*  1546*  1557  1562*  1563*  1564*  1734*  1747*
LBAD    016610           1635   1637   1741#  1767
LBC   =%000005           1477#  1729*  1730*  1731   1753*
LBYT  =%000000           1476#  1724*  1729   1734   1747   1751  1752*  1757  1759*  1760*  1764
LCKSM =%000003           1475#  1722*  1738   1751*  1766
LF      006300            739    780#
```

```
LFLOOP  006310            785#    788     790
LFOUND  006330            786     792#
LFSUB   006304            774     783#
LGWRD   016646           1728    1733    1756#   1763
LINE0 = 000004           1160#   1197
LINE1 = 000005           1161#   1201
LINE2 = 000006           1162#   1205
LINE3 = 000007           1163#   1209
LJMP    016666           1732    1763#
LJMP1   016726           1769    1777#
LLD2    016532           1722#   1725    1739
LLD3    016576           1736#   1748
LLD4    016624           1737    1747#
LOAD    016520           1660    1718#
LOADER  007012            921    1048#
LONGV = 110000           1140#   1197    1201    1205    1209
LOOP    007444           1285#   1292
LOOP1   007522           1314#   1319
LOOP2   007526           1316#   1317    1322
LPDARK= 000200           1171#
LPLITE= 000300           1170#
LPOFF = 000100           1155#
LPON  = 000140           1156#
LPTR    016630           1723    1726    1736    1750#   1756    1758    1765
LSR   = 177560           1268#   1298
L.ADR =%000001            559#   1066*   1080*
L.BAD   007100            939     941    1075#   1106
L.BC  =%000002            560#   1061*   1062*   1063    1087*
L.BYT =%000000            558#    976     991     992*    993*    994*    996*    998*   1002*   1003*   1004*   1006*   1008*
                         1010*   1012*   1017*   1018*   1020*   1022*   1055*   1061    1066    1080    1086    1093    1095*
                         1096*   1103
L.CKSM=%000005            561#   1053*   1071    1086*   1105
L.GWRD  007126           1060    1065    1092#   1102
L.HALT  007200           1114#
L.JMP   007146           1064    1102#
L.JMP1  007202           1112    1116#
L.LD2   007022           1053#   1056    1072
L.LD3   007066           1069#   1081
L.LD4   007110           1070    1080#
L.PTR   007114           1054    1058    1069    1085#   1092    1094    1104
MAJOR   016072           1551#   1555
MAXSX = 017600           1185#
MAXSY = 000077           1186#
MAXX  = 001777           1181#   1198    1206
MAXY  = 001377           1182#   1195    1203    1211
MINSUY= 000100           1187#
MINUSX= 020000           1183#   1184    1203    1206
MINUSY= 020000           1184#
M7      002252            318     440#
M8      002275            316     339     342     344     444#
M9      002301            327     445#
NORMAL  006212            728     737     749#
NOTHER  006042            651     657#
NUMLIN= 000040            592#    681     682     771     857
NXTCHR  006132            724#    726     732     751     761     769     777     780
ORIGIN= 166000            573#
```

```
OTHER   007726          1347    1352    1376    1391    1399    1403#
OUTLIT  006652           647     950#   1075    1108
O2A     002146           325     329     411#
O2AA    002176           420#    432
PC     =%000007          181*    314*    455*    530#    724*    734*    749*    757*    764     774*    795*    810*    813*
                         820*    899*    915*    919*    937*    943*    975*    990*    999*    1014*   1024*   1054*   1058*
                        1060*   1065*   1069*   1085*   1088*   1092*   1094*   1097*   1102*   1104*   1274    1279    1309
                        1328*   1346    1351    1357    1366    1375    1398    1412*   1466#   1531    1548*   1551*   1565*
                        1582*   1587*   1591*   1600*   1614*   1623*   1633*   1638*   1645*   1666*   1687*   1699*   1723*
                        1726*   1728*   1733*   1736*   1741    1744*   1750*   1754*   1756*   1758*   1761*   1763*   1765*
                        1770    1773*   1795*   1802*
POINT = 114000          1141#   1193    1213    1225    1234    1238    1249
POINTR=%000001           539#    559     694*    696*    702*    803*    804     806     808     809*    815     816*    817*
                         818*
PRESTR  006624           926#   1077
PRGO    001034           185#    307
PRGOR   001040           186#    187
PRG1    001506           166     312#    346
PRG1A   001544           320#
PRG1B   001562           324#    340
PRG1C   001600           328#    336
PRG1D   001650           334     341#
PRG2    001670           167     350#    354
PRMTRS  001024           165     180#
PSW    = 177776          160#    403*
PTBOOT  007400          1271#
PWRFAL= 000024          1508#   1529    1546    1834
P10IB  = 025612         1496#   1695
P10IC  = 045772         1501#   1502    1644    1695*   1696*
P10IS  = 025610         1493#   1496    1535*   1693    1697*
P100B  = 025616         1497#   1679*   1690*   1798*
P100C  = 045776         1500#   1501    1675    1679    1680*   1688    1796
P100S  = 025614         1492#   1497    1537*   1677
RCVEC   007764          1417#
RC11    007720          1398#   1417
RELATV= 130000          1144#   1228
RES     007506          1309#   1324
RESTRT  006060           678#    926
RETURN  002130           187*    397*    401#    404
RET1   =%000000         1468#   1476    1547*   1563    1583    1592    1593*   1594*   1595*   1597*   1599*   1602*   1603*
                        1604*   1606*   1608*   1610*   1612*   1616*   1617*   1619*   1621*   1634    1636    1644*   1646
                        1648    1649*   1654    1656*   1657    1659    1662*   1663*   1664    1741*   1770*   1798    1799*
                        1800*
RFVEC   007714          1395#
RF11    007600          1346#   1395
RKVEC   007760          1415#
RK11    007610          1351#   1415
ROMADD  001000           170#    192     213     245     258*    277     320
ROMORG= 166000          1484#
RPVEC   007770          1419#
RP11    007654          1375#   1419
RSTRT   016034          1541#   1658    1745
R0     =%000000          523#    538    1307*   1308*   1314*   1316    1320    1326    1381*   1404*   1456#   1468
R1     =%000001          524#    539    1271*   1281*   1283*   1284*   1285    1290*   1309*   1310*   1312    1314    1383*
                        1406*   1457#   1469
R2     =%000002          525#    540    1272*   1274*   1279*   1285*   1289*   1291    1293*   1294    1311*   1318*   1320*
```

```
                    1346*  1351*  1357*  1366*  1375*  1381   1398*  1404   1458#  1470
R3     =%000003      526#   541   1273*  1276   1312*  1316   1321   1459#  1471
R4     =%000004      527#   543   1276*  1277   1281   1286*  1287   1289   1460#  1472
R5     =%000005      528#   545   1461#  1473
R6     =%000C06     1462#  1465
R7     =%000007     1463#  1466
SCAN   =%000003      541#   562    647*   650*   651*   653*   657    662    681*   684*   689*   692    694
                     696    729*   730*   731    733*   734    783*   785    787    789*   792*   793*   794
                     808*   811*   812*   950    951    952*  1075*  1108*
SCOPE  = 104400      157#   208    238    266    289
SCOPEB   002132      389    395    402#
SCCPEC   002054      154    388#
SCOPEF   002126      392    394*   396*   400#
SCOPEG   002112      391    393    396#
SCR1   =%000005     1473#  1477   1529*  1530*  1531*  1557*  1558   1559*  1560*  1561*
SENDIT   016730     1744   1773   1795#  1797   1801
SETDUN   006126      693    702#
SETLP1   006074      684#   686
SETLP2   006110      692#   699
SETLP3   006116      696#   698
SETUP    006432      689    847#
SHORTV=  104000     1139#  1216   1218   1220   1222
SP     =%000006      529#   644*   662*   678*   780*   976*   995*   997*   998    1013   1023   1093*  1096
                    1103*  1111   1116   1275*  1280*  1283   1465#  1541*  1555*  1583*  1596*  1598*  1599
                    1613   1622   1642   1648*  1650*  1652   1656   1662   1663   1684*  1690   1720*  1757*
                    1760   1764*  1768   1777
SR     = 177570      161#   352    388    390    448
START    006000      172    451    642#
STARTA   016000      454   1529#
STARTX=  000000     1486#  1830
STARTY=  001360     1487#  1831
STATSA=  170000     1167#
STATSB=  174000     1177#  1237
STKPTR=  000500      163#   180    186    244    312    350
STKSRT=  015770     1502#  1541   1720
STOP     007550     1323#  1327
SWITCH   002306      181    314    448#
SYNON  = 000004     1174#
TAB      006222      738    756#   760
TABCNT=%000002      540#   560    682*   685*   692*   697*   745*   750*   758*   759
TABLE    007562     1310   1330#
TABOOT   007500     1307#  1337
TACS   = 177500     1305#  1307
TAPES    007662     1358   1367   1380#
TCVEC    007774     1421#
TC11     007620     1357#  1421
TERM     001022      179#   359*   361
TMPEND=  007776      589#   644    678
TM11     007636     1366#
TPCSR  = 177564      158#   294    297    322    330    364    369    375    381    411    421    436*
TPDBR  = 177566      159#   293*   296*   332*   371*   374*   379*   383*   423*
TYPEM    001712      315    317    326    338    341    343    357#
TYPEMA   001722      360#   380    384
TYPEMB   001750      362    367#
TYPEMC   002036      368    381#
T1       001060      192#
```

DDGTDB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
T1A      001076            195#    203     207
T1B      001136            204     208#
T2       001140            213#
T2A      001156            216#    222     237
T2B      001202            220     223#    226
T2C      001214            224     227#    230
T2D      001224            228     231#    234
T2E      001234            232     235#
T3       001244            244#
T3A      001266            248#    250
T3AA     001260            247#    264
T3B      001274            247     251#
T3C      001304            253#    255
T3D      001312            251     256#
T3E      001322            258#    260
T3F      001332            256     261#
T4       001352            275#
T4A      001366            278#    283     288
T4B      001402            281     285#
T4E      001414            289#
VT       006244            740     764#
VT40PC=  022000           1490#   1549*
WORDS    001002            171#    193     214     246     287     319     450*    453*
WORK1  =%000003           1471#   1475
WORK2  =%000004           1472#   1592*   1605*   1607*   1609*   1611*   1612    1613*   1618*   1620*
.      = 017000            152#    153#    164#    169#    295     323     331     365     370     376     382     422     446#
                           628#   1288    1310    1388    1410    1520#   1589    1625    1775
```

```
ROL      418   419   425   426   427   428   429   430  1017  1616  1617
ROLB     995   997  1005  1007  1009  1011  1018  1596  1598  1605  1607  1609  1611
ROR     1019  1021  1618  1620
RORB    1020  1022  1619  1621
RTI      398
RTS      366   437   455   813   820   899   943   952   999  1014  1024  1098  1097  1565  1600
        1614  1623  1638  1666  1699  1754  1761  1802
SUB      200   219   730  1062  1730
SWAB    1095  1759  1800
TRAP     157
TST      196   235   261   402   657   977  1023  1277  1326  1382  1385  1389  1405  1586  1622
        1642  1646  1675  1688  1796
TSTB     294   297   322   330   364   369   375   381   421   664   893   901  1071  1105  1287
        1677  1682  1693  1738  1766
WAIT    1191
.ASCII   440   444   445
.BYTE   1076  1109  1743  1772
.DSABL  1426
.ENABL   146   499
.END    1850
.EVEN    446
.LIST      2   145   152
.NLIST     2   152
.PAGE    511   565   618   715   876  1038  1133  1425
.REM       2
.REPT    152
.SBTTL   457   510   617   714   875  1037  1120  1263  1301  1341  1424
.TITLE     1   465
.WORD    401   648   847   848   849   850   852   853   854   855   856   857   859   860   861
         863   867   868   869   870   871   872   873  1330  1332  1334  1336  1337  1338  1627
        1628  1629  1630  1828  1829  1830  1831  1832  1833  1834  1835  1836


ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*DDGTDB,DDGTDB.SEQ/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:DDGTDB.P11
RUN-TIME:  4  9  2 SECONDS
RUN-TIME RATIO: 55/18=3.0
CORE USED:  8K  (15 PAGES)
```

JO4

Spooler runtime 6 Seconds, 30 KCS, 156 disk reads, 3 disk writes, 47 pages

[redacted]

[redacted]