



.REM \*

IDENTIFICATION

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

COPYRIGHT (C) 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DEC'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

1.     ABSTRACT

THIS PROGRAM IS A QUICK GO-NOGO TEST OF THE GT40 SYSTEM. THE PURPOSE OF THIS TEST IS TO QUICKLY IDENTIFY ANY PROBLEM IN THE SYSTEM. THE PROGRAM WILL START THE DISPLAY AND THEN INITIATE THE COMMUNICATION LINE. TWO BACKGROUND TASKS ARE EXECUTED. THE FIRST IS A GT-40 ROM VERIFY TEST. THE SECOND TASK IS A WORSE CASE NOISE TEST THRU MEMORY.

2.     REQUIREMENTS

## 2.1    EQUIPMENT

GT40 SYSTEM (11/05, DISPLAY PROCESSOR AND VRI4 SCOPE)  
MODEM TEST CONNECTOR WHICH CONNECTS DATA OUT TO DATA IN.

## 2.2    STORAGE

THIS PROGRAM USED MEMORY LOCATIONS 0-7776 AND 16000-16776 (2K OF MEMORY).

3.     LOADING PROCEDURE

## 3.1    METHOD

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

4.     STARTING PROCEDURE

## 4.1    CONTROL SWITCH SETTINGS

CONSOLE SW DB = 0           TEST AS VERSION 2 ROM (512. WORDS)  
CONSOLE SW DB = 1           TEST AS VERSION 1 ROM (256. WORDS)

## 4.2    STARTING ADDRESS OR ADDRESSES

200     IS THE ONLY STARTING ADDRESS OF THIS TEST

5. OPERATING PROCEDURE

ONCE STARTED THE TEST WILL RUN IN THEIR NORMAL MANNER WITHOUT OPERATOR INTERVENTION OR SWITCH SELECTION. THE OPERATOR MUST VERIFY THE DATA RETURNING FROM THE COMMUNICATION LINE BY COMPARING 'COM OUTPUT' TO 'COM INPUT' ON THE DISPLAY SCREEN. BY TYPING ON THE CONSOLE KEYBOARD, THE CHARACTER AND OCTAL VALUE WILL BE DISPLAYED.

6. ERRORS

THE PROGRAM WILL ONLY HALT ON AN ERROR. THE PROGRAM DOES NOT CONTAIN FACILITES FOR REPORTING MESSAGES OR ERROR CONDITIONS.

7. RESTRICTIONS

A COMMUNICATION TEST PLUG MUST BE INSTALLED ON THE DL-11.

8. MISCELANEOUS8.1 EXECUTION TIME

THE TEST WILL TAKE APPROXIMATELY 10 SECONDS FOR COMPLETION AND WILL RING THE 'GT-40' BELL.

8.2 DEVICE ADDRESS PROGRAM LOCATIONS

LOCATION 1000 CONTAINS THE GT40 DEVICE ADDRESS  
LOCATION 1002 CONTAINS THE GT40 INTERRUPT VECTOR.  
LOCATION 1004 CONTAINS THE GT40 INTERRUPT LEVEL.  
LOCATION 1006 CONTAINS THE DL-11 DEVICE ADDRESS.  
LOCATION 1010 CONTAINS THE DL-11 INTERRUPT VECTOR.  
LOCATION 1012 CONTAINS THE DL-11 INTERRUPT LEVEL.  
LOCATION 1014 CONTAINS THE GT-40 ROM BOOTSTRAP ADDRESS.  
LOCATION 1016 CONTAINS THE GT-40 ROM WORD LENGTH,

9. PROGRAM DESCRIPTION

9.1 DISPLAY FILE <FORGROUND TASK>

THE DISPLAY FILE IS A COMPACT VISUAL TEST OF ALL GT40 DISPLAY INSTRUCTIONS. A BOX OUTLINING THE SCREEN WITH DIFFERENT LINE TYPE VALUE IS DISPLAYED. THREE PAIRS OF ASCII STRINGS ARE ALSO DISPLAYED TO TEST THE CHARACTER LOGIC. THE FIRST LINE OF A STRING IS DISPLAYED IN 'NORMAL' FONT THE SECOND LINE OF A STRING IS DISPLAYED IN 'ITALICS'. ALSO INCLUDED IN THIS VISUAL TEST ARE THE 8 DIFFERENT INTENSITY LEVELS. THE DISPLAY PATTERN IS ENHANCED BY THE USE OF BLINKING OCTAGONS AND MOVING SINE WAVES. THE DISPLAY PATTERN ALSO SERVES AS FOR VISUAL INSPECTION OF THE COMMUNICATION LINE DATA. ALL LINES AND CHARACTERS ARE ENABLED FOR LIGHT-PEN INTERACTION EXCEPT FOR THE LARGEST OCTAGON. UPON LIGHT-PEN HIT, THE TEXT 'LIGHT-PEN HIT' WILL BE DISPLAYED NEAR CENTER SCREEN.

9.2 COMMUNICATION DATA <FORGROUND TASK>

THE DATA PRESENTED TO THE COMMUNICATION LINE APPEARS ON THE DISPLAY SCREEN AS FOUND AT 'COM OUTPUT'. (DECGRAPHIC-11 DISPLAY TERMINAL GT40 VR14) THE DATA ECHOED BACK BY THE TEST CONNECTOR IS DISPLAYED ON THE SCREEN AS FOUND AT 'COM INPUT'. A VISUAL TEST OF THE DATA MUST BE PERFORMED.

9.3 ROM VERIFY TEST <BACKGROUND TASK>

THIS TEST VERIFIES THE DATA CONTAINED IN THE GT-40 ROM BOOTSTRAP.

9.4 WORSE CASE NOISE TEST <BACKGROUND TASK>

THIS IS A BACKGROUND TEST OF ALL AVAILABLE MEMORY. A SMALL PROGRAM IS LOADED INTO ALL EXISTING MEMORY AND THEN EXECUTED THRU THE REMAINDER OF MEMORY.

9.5 KEYBOARD DATA <FORGROUND TASK>

UPON DEPRESSING A KEYBOARD KEY, THE OCTAL VALUE WILL BE DISPLAYED AND ECHO ONTO THE SCREEN.

.TITLE GT40 QUICK VERIFY MAINDEC-11-DDGTE-B  
.ENABL ABS,AMA

.LIST SEQ,BIN  
.NLIST MD,MC,CND

180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225

000000  
000000  
000002  
000024  
000026  
000200  
000137 001022  
001000  
001000 172000  
001002 000320  
001004 000200  
001006 175610  
001010 000300  
001012 000240  
001014 166000  
001016 001000  
001020 006000  
012706 000500  
012777 000340 000162  
004737 001302  
032777 000400 000146  
001007  
012737 001000 001016  
012737 006000 001020  
000406  
012737 000400 001016 1\$:  
012737 016000 001020 2\$:  
005077 000042  
005077 000036  
004737 001636  
004737 001220  
005077 000070  
001126 002544

. = 0 ;0-776 IS LOADED WITH .+2, HALT  
HALT  
HALT  
. = 24  
PWRFL ;POWER FAIL VECTOR  
340  
. = 200  
JMP STARTB  
. = 1000  
GTADD: 172000 ;GT-40 ADDRESS  
GTVCT: 320 ;GT-40 VECTOR  
GTBRL: 200 ;GT-40 BR LEVEL  
DLADD: 175610 ;DL-11 ADDRESS  
DLVCT: 300 ;DL-11 VECTOR  
DLBRL: 240  
ROMADD: 166000 ;ROM STARTING ADDRESS  
WORDS: 512.  
IMAGE: 6000  
STARTB: MOV #500, SP ;LOAD THE STACK POINTER  
MOV #340, @PSW ;RAISE PSW  
JSR PC, INITGT ;INIT DEVICE ADDRESSES  
BIT #400, @SWR ;TEST ROM SWITCH  
BNE 1\$ ;BR IF SET  
MOV #512, WORDS ;ASSUME VER. 2 ROM  
MOV #START, IMAGE ;LOAD IMAGE ADDRESS  
BR 2\$ ;START TEST  
1\$: MOV #256, WORDS ;SELECT VER. 1 ROM  
MOV #STARTA, IMAGE ;LOAD IMAGE ADDRESS  
2\$: CLR @LODBR ;CLEAR OUTPUT  
JSR PC, DOCORE ;SET UP CORE SIZE  
JSR PC, PRIME ;INIT THE DEVICES  
CLR @PSW  
JMP OVER ;EXECUTE BACKGROUND TASK

```

226
227
228 001132 172000      GTPC: 172000      ;DISPLAY PC
229 001134 172002      G1SR: 172002      ;DISPLAY STATUS REG.
230 001136 172004      GTXPOS: 172004    ;DISPLAY X REGISTER
231 001140 172006      GTYPOS: 172006    ;DISSPLAY Y REGISTER
232
233 001142 175610      DLICSR: 175610    ;DL-11 STATUS
234 001144 175612      DLIDBR: 175612    ;DL-11 BUFFER
235 001146 175614      DLOCSR: 175614    ;DL-11 STATUS
236 001150 175616      DLODBR: 175616    ;DL-11 BUFFER
237
238 001152 000320      GTDONE: 320       ;DISPLAY DONE VECTOR
239 001154 000322      GTDNE1: 322
240
241 001156 000324      GTLPH: 324        ;DISPLAY LIGHT-PEN VECTOR
242 001160 000326      GTLPH1: 326
243
244 001162 000330      GTSOTM: 330       ;DISPLAY SHIFT-OUT/ TIME-OUT VECTOR
245 001164 000332      GTSOT1: 332
246
247 001166 000300      DLIVT: 300
248 001170 000302      DLIVT1: 302
249 001172 000304      DLOVT: 304
250 001174 000306      DLOVT1: 306
251
252 001176 177560      TKS: 177560
253 001200 177562      TKB: 177562
254 001202 177564      TPS: 177564
255 001204 177566      TPB: 177566
256
257 001206 000060      KRBT: 60
258 001210 000062      KRBT1: 62
259
260 001212 000200      KRBBRL: 200
261
262 001214 177570      SWR: 177570
263 001216 177776      PSW: 177776
264
265 001220 012777 002724 177704 PRIME: MOV #FILE00, @GTPC ; START THE DISPLAY
266 001226 012777 000100 177712      MOV #100, @DLOCSR ; ENABLE DL OUTPUT
267 001234 012777 000100 177700      MOV #100, @DLICSR ; ENABLE DL INPUT
268 001242 012777 000100 177726      MOV #100, @TKS ; ENABLE KEYBOARD
269 001250 113777 005334 177672      MOVB BUFF1, @DLODBR ; OUTPUT A CHAR
270 001256 012737 000001 002344      MOV #1, PPNT ; PRESET PRINT POINTER
271 001264 005037 002346      CLR RPNT ; CLEAR READ BUFFER
272 001270 005037 002540      CLR KPNT
273 001274 000207      RTS PC ;EXIT
274
275 001276 017476      SIZE: 17476
276 001300 000000      GTDLY0: 0

```

# H01

GT40 QUICK VERIFY MAINDEC-11-DDGTE-B  
DDGTEB.P11

MACY11 27(732) 09-SEP-76 15:20 PAGE 7

277	001302	012700	001132		INITGT:	MOV	#GTPC,RO		;LOAD STARTING ADDRESS
278	001306	013701	001000			MOV	GTADD,RI		;SAVE VALUE
279	001312	004737	001364			JSR	PC,LOADRO		;LOAD GT ADDR
280	001316	013701	001006			MOV	DLADD,RI		;LOAD STARTING ADDRESS <DL-11>
281	001322	004737	001364			JSR	PC,LOADRO		;LOAD DL-11 ADDRESSES
282	001326	013701	001002			MOV	GTVCT,RI		;LOAD VECTOR VALUE
283	001332	004737	001364			JSR	PC,LOADRO		;LOAD GT-40 VECTORS
284	001336	010110				MOV	R1,(RO)		
285	001340	062720	000010			ADD	#10,(RO)+		;LOAD GT TIME-OUT
286	001344	010110				MOV	R1,(RO)		
287	001346	062720	000012			ADD	#12,(RO)+		
288	001352	013701	001010			MOV	DLVCT,RI		;LOAD VECTOR VALUE
289	001356	004737	001364			JSR	PC,LOADRO		;LOAD DL-11 VECTORS
290	001362	000413				BR	INGT		;BR
291	001364	010120			LOADRO:	MOV	R1,(RO)+		;LOAD DONE
292	001366	010110				MOV	R1,(RO)		
293	001370	062720	000002			ADD	#2,(RO)+		
294	001374	010110				MOV	R1,(RO)		
295	001376	062720	000004			ADD	#4,(RO)+		;LOAD DONE
296	001402	010110				MOV	R1,(RO)		
297	001404	062720	000006			ADD	#6,(RO)+		;LOAD PSW
298	001410	000207				RTS	PC		;EXIT
299									
300	001412	012777	002044	177532	INGT:	MOV	#GTSTOP,@GTDONE		;LOAD DONE VECTOR
301	001420	013777	001004	177526		MOV	GTBRL,@GTDNE1		
302	001426	012777	002134	177522		MOV	#GTLPEN,@GTLPH		;LOAD LIGHT-PEN VECTOR
303	001434	013777	001004	177516		MOV	GTBRL,@GTLPH1		
304	001442	012777	002152	177512		MOV	#GTSHIF,@GTSOTM		;LOAD SHIFT-OUT VECTOR
305	001450	013777	001004	177506		MOV	GTBRL,@GTSOT1		
306	001456	012737	000040	001300		MOV	#40,GTDLYD		
307	001464	012737	005664	005634		MOV	#FILEDC,FILEDA		
308	001472	012737	174104	003006		MOV	#STATSB!INCR+4,GRPINC		
309	001500	012700	005430			MOV	#BUFF2,RO		
310	001504	005020			INTD:	CLR	(RO)+		
311	001506	022700	005500			CMP	#BUFF2+50,RO		
312	001512	001374				BNE	INTD		
313	001514	012700	005524			MOV	#BUFF3,RO		;SET UP KRB BUFFER
314	001520	005020			INTE:	CLR	(RO)+		
315	001522	022700	005574			CMP	#BUFF3+50,RO		
316	001526	001374				BNE	INTE		
317	001530	105037	005611			CLRB	OCTA		
318	001534	105037	005612			CLRB	OCTA+1		
319	001540	105037	005613			CLRB	OCTA+2		
320	001544	012777	002242	177414		MOV	#DLIN,@DLIVT		
321	001552	013777	001012	177410		MOV	DLBRL,@DLIVT1		
322	001560	012777	002160	177404		MOV	#DLOUT,@DLOVT		
323	001566	013777	001012	177400		MOV	DLBRL,@DLOVT1		
324	001574	012777	002354	177404		MOV	#KBIN,@KRBVT		;LOAD KRB VECTOR
325	001602	013777	001212	177400		MOV	KRBRL,@KRBVT1		;LOAD PSW
326	001610	000207				RTS	PC		



```

327
328 001612 012737 001622 000024 PWRFL: MOV #PWRUP,2#24 ;LOAD VECTOR
329 001620 000000 HALT
330
331 001622 000005 PWRUP: RESET
332 001624 012737 001612 000024 MOV #PWRFL,2#24
333 001632 000137 001022 JMP STARTB ;RESTART AT BEGINING
334
335 ;SUBROUTINE TO DETERMINE THE SIZE OF CORE
336 ; AND SET UP LOCATION SIZE WITH THE VALUE
337
338 001636 012737 001670 000004 DDCORE: MOV #25,2#4 ;SET UP FOR NEM
339 001644 012701 017776 MOV #17776,R1 ;SET UP ADDRESS
340 001650 005000 CLR RO
341 001652 062701 020000 1$: ADD #20000,R1 ;MOVE TO THE NEXT BANK
342 001656 005200 INC RO ;INC BANK COUNTER
343 001660 005711 TST (1) ;TIMEOUT ?
344 001662 022701 177776 CMP #177776,R1 ;END ?
345 001666 001371 BNE 1$
346 001670 005300 2$: DEC RO ;DECREMENT BANK COUNT
347 001672 012737 000006 000004 MOV #6,2#4 ;RESET BUSS ERROR
348 001700 022626 CMP (SP)+,(SP)+ ;POP THE STACK X2
349 001702 162701 020000 SUB #20000,R1 ;RESTORE R1
350 001706 010137 001276 MOV R1,SIZE
351 001712 162737 007776 001276 SUB #7776,SIZE ;BACK PAST LOADER
    
```

```

352 ;ROUTINE TO LOAD EXCESS CORE WITH WORSE CASE MEMORY TEST
353
354 001720 013700 001276  CORTST: MOV     SIZE,R0      ;GET LAST FREE CORE ADDRESS
355 001724 012701 017000  MOV     #BUFFER,R1    ;GET END OF PROGRAM
356 001730 020001          CMP     R0,R1        ;TEST FOR EQUAL
357 001732 103410          BLO    XMRTS         ;BRANCH IF NO ROOM
358 001734 012702 001770  XMLOP1: MOV     #MENTST,R2 ;MOVE CODE BETWEEN
359 001740 012221          XMLOP2: MOV     (R2)+,(R1)+ ;MENTST AND MEMEND TILL
360 001742 022702 002040  CMP     #MEMEND,R2   ;CORE IS FULL
361 001746 001374          BNE    XMLOP2
362 001750 020100          CMP     R1,R0        ;TEST FOR MORE ROOM
363 001752 101770          BLOS  XMLOP1
364 001754 012721 000207  XMRTS:  MOV     #207,(R1)+ ;SETUP RTS PC
365 001760 005021          CLR    (R1)+
366 001762 005021          CLR    (R1)+
367 001764 000207          RTS     PC
368
369 001766 151456          ROTVAL: 151456
370          .DSABL  AMA
371
372 ;THIS IS THE BACKGROUND TASK WHICH WILL BE LOADED THRU
373 ; THE REMAINDER OF MEMORY
374
375 001770 000277          MENTST: SCC          ;SET CARRY BIT
376 001772 012727 123456  MOV     #123456,(PC)+ ;MENTDAT CONTAINS
377 001776 123456          MENDAT: 123456
378 002000 106067 177773  RORB   MENDAT+1      ;ROTATE LEFT BYTE OF MENDAT
379 002004 103401          BCS   .+4
380 002006 000000          HALT          ;C BIT WAS NOT SET
381 002010 102001          BVC   .+4
382 002012 000000          HALT          ;V BIT WAS SET
383 002014 022767 151456 177754  CMP     #151456,MENDAT ;CHECK HERE FOR CORRECT ROTATE
384 002022 001401          BEQ   .+4
385 002024 000000          HALT          ;ROTATE FAILED
386 002026 026737 177744 001766  CMP     MENDAT,#ROTVAL ;CHECK AGAIN REFERENCE LOW MEMORY
387 002034 001401          BEQ   .+4
388 002036 000000          HALT          ;REF. TO LOW MEMORY FAILED
389 002040 000000          MEMEND: 0
390 002042 000000          0
391          .ENABL  AMA
392
393

```

```

394
395
396
397 002044 005777 177064
398 002050 100403
399 002052 000000
400 002054 000137 001022
401
402 002060 005337 001300
403 002064 001014
404 002066 012737 000040 001300
405 002074 005237 003006
406 002100 022737 174110 003006
407 002106 001003
408 002110 012737 174100 003006
409 002116 012737 005664 005634
410 002124 012777 000001 177000
411 002132 000002
412
413 002134 012737 005636 005634
414 002142 012777 000001 176762
415 002150 000002
416
417 002152 000000
418 002154 000137 001022

; INTERRUPT SERVICE FOR THE GT STOP INTERRUPT
GTSTOP: TST @GTSR ; TEST STOP
        BMI IS
        HALT ; ERROR STOP INTERRUPT BUT NO STOP FLAG
        JMP STARTB ; RESTART TEST

IS:     DEC GTDLYD ; DECREMENT DELAY
        BNE GTST1 ; BRANCH IF NOT
        MOV #40, GTDLYD ; RESET DELAY
        INC GRPINC ; UPDATE GRAPH INCREMENT
        CMP #STATSB!INCR+10, GRPINC ; TEST FOR INCREMENT
        BNE GTST1 ; BRANCH IF NOT
        MOV #STATSB!INCR, GRPINC ; RESET GRAPH INCREMENT
GTST1:  MOV #FILEOC, FILEOA
        MOV #1, @GTPC ; RESUME THE DISPLAY
        RTI ; EXIT

GTLPEN: MOV #FILEOB, FILEOA
        MOV #1, @GTPC ; RESUME THE DISPLAY
        RTI

GTSHIF: HALT
        JMP STARTB ; GT-40 SHIFT-OUT/TIME-OUT ERROR

```

```

419
420
421 ; INTERRUPT SERVICE FOR THE DL PRINTER
422 002160 105777 176762 DLOUT: TSTB @DLOCSR ; TEST FOR DONE
423 002164 100403 BMI .+10
424 002166 000000 HALT ; ERROR, PRINTER INTERRUPT BUT NO PRINTER FLAG
425 002170 000137 001022 JMP STARTB ; RESTART TEST
426
427 002174 010446 MOV R4, -(SP)
428
429 002176 013704 002344 DLOUTA: MOV PPNT, R4 ; LOAD R4 WITH BYTE POINTER
430 002202 116437 005334 002350 MOVB BUFF1(R4), PUNCHR ; LOAD A CHARACTER TO BE OUTPUTTED
431 002210 005237 002344 INC PPNT ; UPDATE CHARACTER POINTER
432 002214 022737 000050 002344 CMP #40, PPNT ; TEST FOR END
433 002222 001002 BNE DLOUTB
434 002224 005037 002344 CLR PPNT ; CLEAR PUNCH POINTER
435
436 002230 113777 002350 176712 DLOUTB: MOVB PUNCHR, @DLOOBR ; OUTPUT A CHARACTER
437 002236 012604 MOV (SP)+, R4 ; RESTORE R4
438 002240 000002 RTI ; EXIT
439
440 ; INTERRUPT SERVICE FOR THE DL READER
441
442 002242 105777 176674 DLIN: TSTB @DLICSR ; TEST FOR DONE
443 002246 100403 BMI .+10
444 002250 000000 HALT ; NOT DL INPUT FLAG
445 002252 000137 001022 JMP STARTB ; RESTART TEST
446
447 002256 010446 MOV R4, -(SP) ; SAVE R4
448 002260 013704 002346 MOV RPNT, R4
449 002264 117737 176654 002352 MOVB @DLIOBR, REDCHR ; READ A CHARACTER
450 002272 042737 177600 002352 BIC #177600, REDCHR ; MASK CHARACTER
451 002300 113764 002352 005430 MOVB REDCHR, BUFF2(R4) ; PUT CHARACTER INTO THE BUFFER
452 002306 005237 002346 INC RPNT ; UPDATE READ POINTER
453 002312 022737 000050 002346 CMP #40, RPNT ; TEST FOR END
454 002320 001002 BNE DLINB
455 002322 005037 002346 CLR RPNT
456 002326 013704 002346 DLINB: MOV RPNT, R4
457 002332 112764 000177 005430 MOVB #177, BUFF2(R4) ; ADD CURSOR
458 002340 012604 MOV (SP)+, R4 ; RESTORE R4
459 002342 000002 RTI ; EXIT
460
461 002344 000000 PPNT: 0
462 002346 000000 RPNT: 0
463 002350 000240 PUNCHR: 240
464 002352 000240 REDCHR: 240

```

```

465
466
467
468
469
470 002354 105777 176616
471 002360 100403
472 002362 000000
473 002364 000137 001022
474
475 002370 010346
476 002372 010446
477 002374 013704 002540
478 002400 117737 176574 002542
479 002406 042737 177600 002542
480 002414 113764 002542 005524
481 002422 005237 002540
482 002426 022737 000050 002540
483 002434 001002
484 002436 005037 002540
485 002442 013704 002540
486 002446 112764 000177 005524
487
488
489
490 002454 013703 002542
491 002460 004737 002524
492 002464 110437 005613
493 002470 004737 002516
494 002474 110437 005612
495 002500 004737 002516
496 002504 110437 005611
497 002510 012604
498 002512 012603
499 002514 000002
500
501 002516 006003
502 002520 006003
503 002522 006003
504 002524 010304
505 002526 042704 177770
506 002532 052704 000060
507 002536 000207
508
509 002540 000000
510 002542 000240

; INTERRUPT SERVICE FOR THE KEYBOARD
KBIN: TSTB @TKS ; TEST FOR DONE
      BMI .+10
      HALT ; NOT KRB INPUT FLAG
      JMP STARTB ; RESTART

      MOV R3, -(SP) ; SAVE R3
      MOV R4, -(SP) ; SAVE R4
      MOV KPNT, R4
      MOVB @TKB, KBCHR ; READ CHARACTER
      BIC #177600, KBCHR ; MASK
      MOVB KBCHR, BUFF3(4) ; SAVE THE CHAR
      INC KPNT ; UPDATE POINTER
      CMP #40., KPNT ; TEST FOR END
      BNE 1S
      CLR KPNT ; CLEAR POINTER
1S: MOV KPNT, R4
     MOVB #177, BUFF3(R4) ; ADD CURSOR

; UPDATE OCTAL READOUT
      MOV KBCHR, R3 ; GET CHR
      JSR PC, 10S ; LOAD BITS
      MOVB R4, OCTA+2 ; SAVE BITS
      JSR PC, 11S ; MOVE BITS
      MOVB R4, OCTA+1 ; SAVE BITS
      JSR PC, 11S ; MOVE BITS
      MOVB R4, OCTA ; SAVE BITS
      MOV (SP)+, R4 ; RESTORE R4
      MOV (SP)+, R3 ; RESTORE R3
      RTI ; EXIT

11S: ROR R3
     ROR R3
     ROR R3
10S: MOV R3, R4 ; LOAD R4
     BIC #177770, R4 ; MASK BITS
     ADD #60, R4 ; MAKE A NUMBER
     RTS PC

KPNT: 0
KBCHR: 240

```

```

511 ;PART 1 OF THE BACKGROUND TASK
512
513 002544 012737 001000 002722 OVER: MOV #1000,PCOUNT ;SET UP EXECUTION COUNT
514
515 : COMPARE THE ROM DATA TO THE IMAGE DATA
516 :
517 : RD=WORD NUMBER
518 : R1=GOOD DATA
519 : R2=GOOD DATA
520 : R3=BAD ADDRESS
521 : R4=BAD DATA
522 002552 012700 000000 BACK: MOV #0,%0 ;SETUP INITIAL WORD NUMBER
523 002556 013701 001020 MOV IMAGE,%1 ;SET UP BUFFER
524 002562 013703 001014 MOV ROMADD,%3 ;SET UP ROM ADDRESS
525 002566 011102 BACK1: MOV (%1),%2 ;READ A IMAGE WORD
526 002570 011304 MOV (%3),%4 ;READ A ROM WORD
527 002572 020204 CMP %2,%4 ;TEST FOR EQUAL
528 002574 001402 BEQ BACK2 ;BRANCH IF OK
529 002576 000000 HALT ;ERROR ROM VALUE FAILED TO EQUAL THE
530 002600 000772 BR BACK1 ; THE EXPECTED
531
532 002602 022123 BACK2: CMP (%1)+,(%3)+ ;BUMP BOTH REGISTERS
533 002604 005200 INC %0 ;UPDATE WORD COUNTER
534 002606 023700 001016 CMP WORDS,%0 ;TEST FOR LAST WORD
535 002612 001365 BNE BACK1 ;BRANCK IF NOT LAST
536
537 ;PART 2 OF THE BACKGROUND TASK
538 ; EXECUTE WORSE CASE NOISE TEST THRU MEMORY
539
540 002614 004737 017000 JSR PC,BUFFER ;EXECUTE NOISE TEST
541 002620 005337 002722 DEC PCOUNT ;DONE PASS ?
542 002624 001352 BNE BACK ;NO
543 002626 012777 000001 176300 MOV #1,%GTSR ;YES RING THE BELL
544 002634 012777 000207 176342 MOV #207,%TPB ;RING THE BELL
545 002642 105777 176334 15: TSTB %TPS
546 002646 100375 BPL 15
547 002650 012777 000207 176326 25: MOV #207,%TPB
548 002656 105777 176320 TSTB %TPS
549 002662 100375 BPL 25
550 002664 005737 000042 TST #42 ;TEST LOC. 42
551 002670 001725 BEQ OVER ;BR IF =0
552 002672 000005 RESET
553 002674 000005 RESET
554 002676 000005 RESET
555 002700 013700 000042 MOV #42,R0 ;READ VALUE
556 002704 004710 LOGICAL: JSR PC,(R0)
557 002706 000240 NOP
558 002710 000240 NOP
559 002712 000240 NOP
560 002714 000240 NOP
561 002716 000137 001022 JMP STARTB
562
563
564 002722 000000 PCOUNT: 0

```

```

565
566
567
568 002724 114140 FILEDD: POINT!LPON
569 002726 000000 0
570 002730 001377 MAXY
571 002732 174300 STATSB!LPLITE
572
573 ;LINE THE EDGES OF THE SCREEN
574
575 002734 113004 LONGV!INT4!LINE0 ;TOP LINE
576 002736 041777 INTX!MAXX
577 002740 000000 0
578 002742 110005 LONGV!LINE1 ;RIGHT LINE
579 002744 040000 INTX
580 002746 021377 MINUSX!MAXY
581 002750 110006 LONGV!LINE2 ;BOTTOM LINE
582 002752 061777 INTX!MINUSX!MAXX
583 002754 000000 0
584 002756 110007 LONGV!LINE3 ;LEFT LINE
585 002760 040000 INTX
586 002762 001377 MAXY
587
588 ;SETUP THE X SINEWAVE
589
590 002764 114004 POINT!LINE0
591 002766 000400 400
592 002770 000200 200
593 002772 110000 LONGV
594 002774 041200 INTX+1200 ;DRAW X AXIS
595 002776 000000 0
596 003000 114000 POINT
597 003002 000440 440
598 003004 000200 200
599 003006 174104 GRPINC: STATSB!INCR+4 ;GRAPHPLOT THE X SINEWAVE
600
601 003010 124000 GRAPHY
602
603 ;SETUP THE Y SINEWAVE
604
605 003470 114000 POINT
606 003472 000200 200
607 003474 000040 40
608 003476 110000 LONGV ;DRAW Y AXIS
609 003500 040000 INTX
610 003502 001200 1200
611 003504 114000 POINT
612 003506 000200 200
613 003510 000100 100
614 003512 120000 GRAPHX ;GRAPHPLOT THE Y SINEWAVE
615
616

```

;SETUP TO DISPLAY THE OCTAGONS

617			
618			
619			
620	004172	114070	POINT
621	004174	001434	1434
622	004176	000724	724
623	004200	130030	RELATV:BLKON
624	004222	114000	POINT
625	004224	001430	1430
626	004226	000710	710
627	004230	130020	RELATV:BLKOFF
628	004252	114070	POINT
629	004254	001420	1420
630	004256	000660	660
631	004260	104030	SHORTV:BLKON
632	004302	114000	POINT
633	004304	001400	1400
634	004306	000600	600
635	004310	104020	SHORTV:BLKOFF
636	004332	114030	POINT:BLKON
637	004334	001360	1360
638	004336	000520	520
639	004402	114120	POINT:BLKOFF:LPJFF
640	004404	001340	1340
641	004406	000440	440
642			
643	004452	114140	POINT:LPON
644	004454	000100	100
645	004456	001277	MAXY-100
646	004460	164000	DNOP
647	004462	170040	STATSA:ITALO
648	004464	100000	CHAR
649	004562	164000	DNOP
650	004564	170060	STATSA:ITALI
651	004566	114000	POINT
652	004570	000100	100
653	004572	001247	MAXY-130
654	004574	100000	CHAR
655	004672	170040	STATSA:ITALO
656	004674	114000	POINT
657	004676	000220	220
658	004700	001177	MAXY-200
659	004702	100000	CHAR
660	004744	170060	STATSA:ITALI
661	004746	114000	POINT
662	004750	000220	220
663	004752	001147	MAXY-230
664	004754	100000	CHAR
665	005016	170040	STATSA:ITALO
666	005020	114000	POINT
667	005022	000220	220
668	005024	001077	MAXY-300
669	005026	100000	CHAR
670	005072	170060	STATSA:ITALI
671	005074	114000	POINT
672	005076	000220	220



673 005100 001047  
674 005102 100000  
675 005146 170040  
676  
677  
678 005150 114000  
679 005152 000340  
680 005154 001000  
681 005156 113604  
682 005160 040400  
683 005162 000000  
684 005164 114000  
685 005166 000340  
686 005170 000740  
687 005172 113400  
688 005174 040400  
689 005176 000000  
690 005200 114000  
691 005202 000340  
692 005204 000700  
693 005206 113200  
694 005210 040400  
695 005212 000000  
696 005214 114000  
697 005216 000340  
698 005220 000640  
699 005222 113000  
700 005224 040400  
701 005226 000000  
702 005230 114000  
703 005232 000340  
704 005234 000600  
705 005236 112600  
706 005240 040400  
707 005242 000000  
708 005244 114000  
709 005246 000340  
710 005250 000540  
711 005252 112400  
712 005254 040400  
713 005256 000000  
714 005260 114000  
715 005262 000340  
716 005264 000500  
717 005266 112200  
718 005270 040400  
719 005272 000000  
720 005274 114000  
721 005276 000340  
722 005300 000440  
723 005302 112000  
724 005304 040400  
725 005306 000000  
726  
727  
728

MAXY-330  
CHAR  
STATSA:ITALO

;SETUP INTENSITY LEVEL TEST

POINT  
340  
1000  
LONGV!INT7!LINED  
INTX+400  
0  
POINT  
340  
740  
LONGV!INT6  
INTX+400  
0  
POINT  
340  
700  
LONGV!INT5  
INTX+400  
0  
POINT  
340  
640  
LONGV!INT4  
INTX+400  
0  
POINT  
340  
600  
LONGV!INT3  
INTX+400  
0  
POINT  
340  
540  
LONGV!INT2  
INTX+400  
0  
POINT  
340  
500  
LONGV!INT1  
INTX+400  
0  
POINT  
340  
440  
LONGV!INT0  
INTX+400  
0

;SETUP THE MESSAGA BUFFERS

729	005310	117000				POINT:INT4
730	005312	000400				400
731	005314	000020				20
732	005316	100000				CHAR
733	005320	047503	020115	052517		.ASCII /COM OUTPUT /
734	005326	050124	052125	020040		
735	005334	042504	043503	040522	BUFF1:	.ASCII /DECGRAPHIC-11 DISPLAY TERMINAL GT40 VR14/
736	005342	044120	041511	030455		
737	005350	020061	044504	050123		
738	005356	040514	020131	042524		
739	005364	046522	047111	046101		
740	005372	043440	032124	020060		
741	005400	051126	032061			
742	005404	114000				POINT
743	005406	000400				400
744	005410	000320				320
745	005412	100000				CHAR
746	005414	047503	027115	044440		.ASCII /COM. INPUT /
747	005422	050116	052125	020040		
748	005430	000	000	000	BUFF2:	.BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
749	005433	000	000	000		
750	005436	000	000	000		
751	005441	000	000	000		
752	005444	000	000	000		
753	005447	000	000	000		
754	005452	000	000	000		
755	005454	000	000	000		.BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
756	005457	000	000	000		
757	005462	000	000	000		
758	005465	000	000	000		
759	005470	000	000	000		
760	005473	000	000	000		
761	005476	000	000	000		
762	005500	114000				POINT
763	005502	000400				400
764	005504	000350				350
765	005506	100000				CHAR
766	005510	051113	027102	044440		.ASCII /KRB. INPUT /
767	005516	050116	052125	020040		
768	005524	000	000	000	BUFF3:	.BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
769	005527	000	000	000		
770	005532	000	000	000		
771	005535	000	000	000		
772	005540	000	000	000		
773	005543	000	000	000		
774	005546	000	000	000		
775	005550	000	000	000		.BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
776	005553	000	000	000		
777	005556	000	000	000		
778	005561	000	000	000		
779	005564	000	000	000		
780	005567	000	000	000		
781	005572	000	000	000		
782						
783	005574	114000				POINT
784	005576	000400				400

785	005600	000400				400
786	005602	100000				CHAR
787	005604	041517	020124	000		.ASCIZ /OCT /
788	005611	000	000	000	OCTA:	.BYTE 0,0,0
789	005614	164000				DNOP
790	005616	164000				DNOP
791	005620	164000				DNOP
792	005622	164000				DNOP
793	005624	164000				DNOP
794	005626	164000				DNOP
795	005630	164000				DNOP
796	005632	160000				DJMP
797	005634	005664			FILEDA:	FILEOC
798	005636	114000			FILEDB:	POINT
799	005640	001000				1000
800	005642	000440				440
801	005644	100000				CHAR
802	005646	044514	044107	026524		.ASCIZ /LIGHT-PEN HIT/
803	005654	042520	020116	044510		
804	005662	000124				
805						.EVEN
806	005664	173400			FILEOC:	DSTOP
807	005666	160000				DJMP
808	005670	002724				FILEOO
809						

810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862

```
*****  
: 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 BURNES.
```

```
;  
: 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

863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915

REGISTER DEFINITIONS  
-----

BASIC DEFINITIONS  
-----

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 DEFINITIONS  
-----

000000	CHAR=R0	;CONTAINS THE INPUT CHARACTER. ;POINTS TO NEXT INSERTION BYTE IN DISPLAY BUFFER ;CHARACTER COUNTER FOR THE "TAB" FEATURE. ;GENERALLY CONTAINS A POINTER WHICH ;IS USED WHEN SCANNING MEMORY FOR SOMETHING. ;TYPICALLY A TEMPORARY WHICH IS USED TO RETAIN ;A VALUE FOR A SHORT TIME. ;TYPICALLY USED AS A COUNTER.
000001	POINTR=R1	
000002	TABCNT=R2	
000003	SCAN=R3	
000004	HOLD=R4	
000005	COUNTR=R5	

LOADER DEFINITIONS  
-----

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

916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967

;  
;  
;

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	KBOIS=177560	;KEYBOARD INPUT STATUS
177562	KBOIB=KBOIS+2	;CURRENT CHARACTER FROM KEYBOARD.
172000	GT40PC=172000	;GT40 PROGRAM COUNTER.
172002	GT4OSR=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	ALTM00=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

```

968
969
970
971      ;          GT40 BOOTSTRAP CODE
972      ;          -----
973
974
975
976
977      006000      .=6000
978
979      ;          .=ORIGIN          ;DEFINE ORIGIN OF THE BOOTSTRAP.
980
981
982
983
984
985
986      ;          COLD INITIALIZATION CODE
987      ;          -----
988
989
990
991      006000      000005      START:  RESET          ;RESET ALL HARDWARE NOW.
992      006002      012737      000007      175610      MOV          #7,DL11IS      ;INITIALIZE DL-11 INPUT NOW.
993      006010      012706      007776      MOV          #TAPEND,SP      ;ESTABLISH A GOOD TEMPORARY STACK
994      ;          POINTER FOR CORE SEARCH.
995      006014      005237      175614      INC          DL110S      ;SET BREAK BIT
996      006020      004337      166652      JSR          SCAN,OUTLIT!160000      ;FOR 2 CHARACTER TIMES
997      006024      000000      .WORD 0      ;SEND TWO ZERO'S
998
999      006026      012703      000004      MOV          #CORSTR,SCAN      ;GET ADDRESS OF BAD CORE TRAP VECTOR.
1000      006032      012723      166042      MOV          #NOTHERE!160000,(SCAN)+      ;AND INSERT A POINTER TO US THERE.
1001
1002      006036      005023      ENDCOR:  CLR          (SCAN)+      ;NOW CLEAR ALL OF MEMORY BEYOND THE POINTER,
1003      006040      000776      BR          ENDCOR      ;UNTIL WE RUN OUT OF MEMORY AND TRAP.
1004
1005
1006      006042      005743      NOTHER:  TST          -(SCAN)      ;WHEN WE TRAP OUT, WE COME HERE.
1007      ;          WE BACK UP POINTER TO GOOD CORE.
1008      ;          NOTE THAT IF WE TRAP OUT AGAIN, IT
1009      ;          IS STILL OK, BECAUSE WE WILL LOOP
1010      ;          UNTIL WE GET A GOOD CORE ADDRESS.
1011      006044      010306      MOV          SCAN,SP      ;WHEN WE GET ONE, THAT IS LAST LOCATION
1012      ;          IN THE MACHINE, AND HENCE OUR SP.
1013      006046      105737      175614      IS:      TSTB         DL110S      ;SEE IF BREAK IS DONE
1014      006052      100375      BPL          IS      ;NO GO BACK
1015      006054      005037      175614      CLR          DL110S      ;CLEAR BREAK BIT
1016
1017
1018
1019
1020
1021
1022      ;          RESTART INITIALIZATION CODE WHEN COMMUNICATIONS IS WORKING.
1023      ;          -----

```

```

1024
1025
1026
1027 006060 052706 007776      RESTR: BIS      #TMPEND,SP      ;FORCE THE SP TO LIMIT OF EXISTING CORE.
1028
1029
1030 006064 012703 006700      MOV      #BLIMIT-NUMLIN-NUMLIN,SCAN ;NOW WE WILL FILL THE KEY AREAS OF THE
1031 006070 012702 000040      MOV      #NUMLIN,TABCNT ;DISPLAY BUFFER WITH INITIAL CR-LF'S.
1032
1033 006074 012723 005015      SETLP1: MOV     #CRLF,(SCAN)+ ;INSERT A CRLF NOW.
1034 006100 005302                DEC     TABCNT ;AND LOOP UNTIL DONE.
1035 006102 003374                BGT    SETLP1 ;THUS DISPLAY CORE IS ALMOST CORRECT.
1036
1037
1038 006104 012703 166432      MOV      #SETUP!160000,SCAN ;NOW WE WILL INITIALIZE CORE FOR THE
1039                                ;DISPLAY. PICK UP POINTER TO LIST.
1040
1041 006110 012302      SETLP2: MOV     (SCAN)+,TABCNT ;GET NUMBER OF ITEMS TO INSERT.
1042 006112 001405      BEQ     SETDUN ;IF ZERO, WE ARE DONE.
1043 006114 012301      MOV     (SCAN)+,POINTR ;PICK UP FIRST CORE ADDRESS POINTER.
1044
1045 006116 012321      SETLP3: MOV     (SCAN)+,(POINTR)+ ;MOVE OVER A DATA ITEM NOW.
1046 006120 005302      DEC     TABCNT ;ALL DONE?
1047 006122 003375      BGT    SETLP3 ;NOPE. MOVE OVER THE NEXT.
1048 006124 000771      BR     SETLP2 ;YES. GET NEXT MAJOR LIST TO INSERT.
1049
1050
1051 006126 012701 006776      SETDUN: MOV     #BLIMIT-2,POINTR ;ESTABLISH THE BUFFER POINTER NOW.
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063

```



1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119

VTOS (SCROLLING) PORTION OF THE BOOTSTRAP

```

NXTCHR: JSR PC,GETCHR!160000
          CMP CHAR,#177
          BGE NXTCHR
          CMP CHAR,#40
          BGE NORMAL
          MOV CHAR,SCAN
          SUB #7,SCAN
          CMP SCAN,#7
          BHS NXTCHR
          ASL SCAN
          ADD SCAN,PC

          BR BELL
          BR NORMAL
          BR TAB
          BR LF
          BR VT
          BR FF

CR:      MOV #7,TABCNT

NORMAL:  JSR PC,INSERT!160000
          INC TABCNT
          BR NXTCHR

TAB:     MOV #40,CHAR
          JSR PC,INSERT!160000
          INC TABCNT
          BIT #7,TABCNT
          BNE TAB
          BR NXTCHR

VT:      MOVB (PC),COUNTR
          BR FFLOOP

BELL:    CLR GT4OSR
          BR NXTCHR

FF:      MOV #NUMLIN,COUNTR
    
```

```

;GET A CHARACTER NOW.
;IS IT OUT OF RANGE?
;YEP. GET ANOTHER ONE.
;IS IT A PRINTING CHARACTER?
;YES. IT'S A NORMAL PRINTING CHARACTER.
;MOVE IT OVER SO WE CAN PLAY WITH IT.
;BIAS SO THAT BELL [7] IS ZERO.
;IF CHARACTER IS LESS THEN BELL OR
;GREATER THEN CR, THEN IGNORE.
;IF GOOD, MAKE IT WORD INDEX.
;AND GO TO THE CORRECT ROUTINE.

;7=BELL
;10=BACKSPACE
;11=TAB
;12=LINE FEED [LF]
;13=VERTICAL TAB [VT]
;14=FORM FEED [FF]
;15=CARRIAGE RETURN [CR]

;RESET TAB POSITION ON A CR, AND
;FALL THROUGH TO INSERT THE CHARACTER.

;INSERT THE CHARACTER IN THE BUFFER.
;UPDATE TAB POSITION NOW.
;AND GET NEXT CHARACTER.

;ON A TAB, INSERT BLANKS UNTIL THE
;NEXT CHARACTER POSITION IS A MULTIPLE
;OF 8.
;ARE WE DONE YET?
;NOPE.
;YES.

;THIS PUTS THE LOW BYTE OF THE
;BRANCH CODE IN COUNTR-SAVE A WORD

;RING BELL -WRITE IN GT4OSR
;AND LOOP BACK

;FORM FEED IS DONE BY INSERTING LF'S.
    
```

```

1120
1121 006262 012700 000012      FFLOOP: MOV      #12, CHAR      ;MAKE THE CHARACTER A LINEFEED.
1122 006266 004737 166304      JSR      PC, LFSUB!160000      ;DO A LINEFEED.
1123 006272 005305              DEC      COUNTR                ;DONE?
1124 006274 003372              BGT      FFLOOP                ;NOPE. KEEP SENDING THEM.
1125 006276 000715              BR       NXTCHR                 ;YES. NOW RETURN. DO NOT FALL THROUGH.
1126
1127
1128 006300 012746 166132      LF:      MOV      #NXTCHR!160000, -(SP) ;RETURN TO NXTCHR AFTER PROCESSING
1129                                         ;THE LF BY FAKING A JSR.
1130
1131 006304 013703 007012      LFSUB:  MOV      JMPADD, SCAN    ;GET POINTER TO FIRST CHAR ON SCREEN
1132
1133 006310 122300              LFLOOP: CMPB     (SCAN)+, CHAR    ;AND LOOK FOR A LINEFEED.
1134 006312 001406              BEQ      LFOUND                 ;GOT IT. SEARCH HAS ENDED.
1135 006314 020327 007000      CMP      SCAN, #BLIMIT          ;ARE WE AT END OF BUFFER?
1136 006320 103773              BLO      LFLOOP                 ;NOPE. KEEP ON LOOKING.
1137 006322 012703 001000      MOV      #BSTART, SCAN          ;IF AT TOP, RESET TO BOTTOM OF BUFFER
1138 006326 000770              BR       LFLOOP                 ;AND KEEP ON LOOKING.
1139
1140 006330 005203              LFOUND: INC      SCAN            ;WE'VE GOT THE LINE FEED. STOP SHOWING
1141 006332 042703 000001      BIC      #1, SCAN              ;FIRST LINE BY CHANGING THE "DISJMP"
1142 006336 010337 007012      MOV      SCAN, JMPADD           ;INSTRUCTION TO FIRST CHAR BEYOND LF.
1143 006342 004737 166350      JSR      PC, INSERT!160000      ;INSERT THE LF IN THE BUFFER.
1144 006346 005000              CLR      CHAR                  ;AND THEN INSERT ONE NULL CHARACTER BECAUSE
1145                                         ;THE "DISJMP" ADDRESS MUST BE EVEN, AND
1146                                         ;THIS GUARANTEES WE WILL NOT LOSE A
1147                                         ;A GOOD DATA CHARACTER. WE FALL THROUGH
1148                                         ;TO INSERT THE NULL IN THE BUFFER.
1149
1150
1151 006350 110021              INSERT: MOVVB    CHAR, (POINTR)+  ;STICK IN THE CHARACTER NOW.
1152 006352 032701 000001      BIT      #1, POINTR            ;IS NEXT POSITION EVEN OR ODD?
1153 006356 001021              BNE      INSRTX                 ;ODD. NO PROBLEMS. SPACE IS ALLOCATED.
1154 006360 020127 007000      CMP      POINTR, #BLIMIT        ;EVEN. ARE WE AT THE END OF THE BUFFER?
1155 006364 103410              BLO      INSRTL                 ;NO. JUST MAKE ROOM FOR ANOTHER WORD.
1156 006366 010103              MOV      POINTR, SCAN          ;AT THE END. MOVE THE STUFF TO THE
1157 006370 012701 001000      MOV      #BSTART, POINTR        ;BEGINNING OF THE BUFFER.
1158 006374 004737 166406      JSR      PC, INSRTL!160000      ;CALL THE ROUTINE TO SAVE SPACE.
1159 006400 005023              CLR      (SCAN)+               ;AND CLEAR UP THE INSTRUCTIONS AT THE
1160 006402 005013              CLR      (SCAN)                ;END OF THE BUFFER.
1161 006404 000207              RTS      FC                     ;AND THEN RETURN.
1162
1163 006406 022121              INSRTL: CMP      (POINTR)+, (POINTR)+ ;BYPASS THE "DISJMP" BY ADDING 4 TO POINTR.
1164 006410 012711 166474      MOV      #HEADER!160000, (POINTR) ;NOW INSERT THE DISJMP INSTRUCTION TO OUR HEADER
1165 006414 012741 160000      MOV      #DISJMP, -(POINTR)      ;AND IT'S ADDRESS (PUT THEM IN BACKWARDS).
1166 006420 005041              CLR      -(POINTR)              ;MAKE AVAILABLE A NEW CHARACTER SPOT.
1167
1168 006422 000207              INSRTX: RTS      PC             ;FINALLY RETURN TO THE CALLER.
1169
1170
1171
1172
1173
1174 006424 012737 001000 172000  GTBUSE: MOV      #BSTART, GT40PC ;ON A BUS ERROR, WE MERELY RESTART THE GT40 AT
1175

```

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

1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223

INITIALIZATION TABLE FOR THE SCOLLER  
-----

;					
SETUP:	.WORD	2		; INITIALIZE 2 WORDS.--ALSO RTI FROM ABOVE	
	.WORD	330		; STARTING AT LOCATION 330	
	.WORD	GTBUSE!160000		; FIRST WORD IS POINTER TO BUS ERROR ROUT	
	.WORD	200		; SECOND WORD IS NEW STATUS WORD ON INTERRUPT.	
	.WORD	7		; INITIALIZE THE END OF THE BUFFER TO	
	.WORD	BLIMIT-2		; A CLEAR SPACE TO INSERT THE CHARACTER.	
	.WORD	0		; THIS IS THE "RUNNING" START, THIS IS	
	.WORD	DISJMP,HEADER!160000	166474	; FOLLOWED BY A DISJMP TO OUR HEADER BLOC	
	.WORD	DISJMP,BSTART	001000	; AND THEN A DISJMP TO THE START OF THE BUFFER	
	.WORD	DISJMP,BLIMIT-NUMLIN-NUMLIN	006700	; AND A DISJMP TO THE FIRST CHAR ON SCREE	
	.WORD	1		; FINALLY START THE GT40 GOING AT	
	.WORD	GT40PC		; THE POSITION INSTRUCTION IN THE	
	.WORD	HEADER!160000		; HEADER BLOCK.	
	.WORD	0		; END OF INIT CODE	

HEADER BLOCK FOR THE SCOLLER

HEADER:	.WORD	103334		; ENAP, CHAR MODE,BLINKING
	.WORD	177		; A BLINKING BOX-RUB OUT:
	.WORD	116124		; C. TO POINT MODE
	.WORD	171340		; LOAD STATUS REGISTER
	.WORD	0,1352	001352	; POINT TO UPPER LEFT
	.WORD	103324		; BACK TO CHAR MODE
	.WORD	DISJMP,JMPADD-2	007010	; AND TO THE CHANGING JMP INST.

.SBTTL COMMUNICATIONS AND MISC. SUPPORT ROUTINES

1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279

COMMUNICATIONS HANDLING ROUTINES  
 -----

THE DL-11 HANDLER  
 -----

1240	006516	105737	175610	GETDL:	TSTB	DL11IS	:CHECK THE HOST INPUT STATUS.
1241	006522	100011			BPL	GETDL	:HOST DID NOT SEND ANYTHING, YET.
1242	006524	113700	175612		MOVB	DL11IB,CHAR	:HOST SENT US A CHARACTER. PROCESS IT.
1243	006530	012737	000007	175610	MOV	#7,DL11IS	:REENABLE THE HOST TELECOMMUNICATIONS.
1244	006536	042700	177600		BIC	#-200,CHAR	:MAKE CHARACTER JUST SEVEN BITS.
1245	006542	001765			BEO	GETDL	:IF NULL, IGNORE IT.
1246	006544	000207			RTS	PC	:ELSE RETURN NOW.
1248	006546	105737	177560	GETDL1:	TSTB	KBOIS	:DID USER TYPE A CHARACTER?
1249	006552	100361			BPL	GETDL	:NO. GO BACK AND CHECK HOST MACHINE.
1250	006554	113737	177562	175616	MOVB	KBOIB,DL110B	:MOVE THE CHARACTER TO THE HOST.
1251	006562	000755			BR	GETDL	:AND CHECK AGAIN FOR INPUT.

THE "GET CHARACTER" ROUTINE  
 -----

1262	006564	004737	166516	GETCHR:	JSR	PC,GETDL!160000	:GET A CHARACTER FROM THE HOST NOW.
1263	006570	020027	000175		CMP	CHAR,#ALTMOD	:IS IT AN "ALTMODE"
1264	006574	001025			BNE	GETEXT	:NO. EXIT NOW.
1266	006576	004737	166516		JSR	PC,GETDL!160000	:YES. GET ANOTHER ONE NOW.
1267	006602	020027	000114		CMP	CHAR,#'L	:IS IT AN "L"
1268	006606	001501			BEO	LOADER	:YES. START LOADING NOW.
1269	006610	020027	000122		CMP	CHAR,#'R	:IS IT AN "R"
1270	006614	001015			BNE	GETEXT	:NO. IGNORE THE ALTMODE AND JUST RETURN THE CHAR
1272	006616	012737	173000	007010	PRESTR:	MOV	#DISTOP,JMPADD-2
1273	006624	000137	166060		JMP	RESTR!160000	:YES. RESET. STOP DISPLAY BY INSERTING A "DISTOP ;INSTRUCTION IN THE BUFFER, AND RESTART.

THE "GET A SIX BIT CHARACTER" ROUTINE

```

1280 ; -----
1281 ;
1282 ;
1283 ;
1284 006630 004737 166564 GETSIX: JSR PC,GETCHR!160000 ;GET A CHARACTER NOW.
1285 006634 020027 000040 CMP CHAR,#40 ;IS IT A LEGAL PRINTING CHARACTER?
1286 006640 002517 BLT L,BA0 ;NOPE. ABORT
1287 006642 020027 000137 CMP CHAR,#137 ;IT'S BIG ENOUGH. IS IT TOO BIG?
1288 006646 003114 BGT L,BA0 ;YEP. ABORT.
1289 ;
1290 006650 000207 GETEXT: RTS PC ;RETURN TO THE CALLER.
1291 ;
1292 ;
1293 ; THIS OUTPUTS TWO CHARACTERS VIA A
1294 ; JSR SCAN,OUTLIT
1295 ; 'TWO CHARACTERS'
1296 ;
1297 006652 112337 175616 OUTLIT: MOV (SCAN)+,DL1108
1298 006656 112337 175616 MOV (SCAN)+,DL1108 ;DOUBLE BUFFERED
1299 006662 000203 RTS SCAN ;RETURN
1300 ;
1301 ;
1302 ;
1303 ;
1304 ;
1305 ;
1306 ;
1307 ; THE "GET AN EIGHT BIT CHARACTER" ROUTINE
1308 ; -----
1309 ;
1310 ;
1311 ;
1312 ; THIS ROUTINE DIFFERS FROM THE PREVIOUS ROUTINES
1313 ; IN THAT IT WILL TAKE SIX BIT CHARACTERS AND ASSEMBLE
1314 ; THEM FOR THE LOADER TO USE. NOTE THAT FROM THIS POINT
1315 ; ON WE WILL SWITCH TO THE LOADER DEFINITIONS OF THE
1316 ; REGISTERS. THUS THE CHARACTER IS RETURNED IN
1317 ; REGISTER "L,BYT" RATHER THAN CHAR (THOUGH THEY ARE
1318 ; PHYSICALLY THE SAME).
1319 ;
1320 ;
1321 ;
1322 006664 004737 166630 GET8: JSR PC,GETSIX!160000 ;GET A SIXBIT CHARACTER.
1323 006670 010046 MOV L,BYT,-(SP) ;SAVE IT ON THE STACK.
1324 006672 005723 TST (INDEX)+ ;UPDATE INDEX TO NEXT ITEM (ALL ARE *2)
1325 006674 000163 166676 JMP GET8TB-2!160000(INDEX) ;AND DISPATCH ACCORDING TO THE INDEX.
1326 ;
1327 006700 000404 GET8TB: BR GET81 ;INDEX=2: ASSEMBLE FIRST CHAR
1328 006702 000416 BR GET82 ;INDEX=4: ASSEMBLE SECOND CHAR
1329 006704 000432 BR GET83 ;INDEX=6: ASSEMBLE THIRD AND LAST CHAR
1330 ;INDEX=8: RESET INDEX TO 0 (2) AND RETRY.
1331 ;
1332 ;
1333 006706 012703 000002 GET84: MOV #2,INDEX ;THE FOURTH INDEX IS THE SAME AS THE FIRST
1334 ;INDEX. JUST RESET IT AND FALL THROUGH.
1335 ;
    
```

```

1336
1337 006712 004737 166630 GETB1: JSR PC,GETSIX!160000
1338 006716 010004 MOV L,BYT,HOLD
1339 006720 006300 ASL L,BYT
1340 006722 006300 ASL L,BYT
1341 006724 106300 ASLB L,BYT
1342 006726 106116 ROLB (SP)
1343 006730 106300 ASLB L,BYT
1344 006732 106116 ROLB (SP)
1345 006734 012600 MOV (SP)+,L,BYT
1346 006736 000207 RTS PC
1347
1348
1349 006740 006300 GETB2: ASL L,BYT
1350 006742 006300 ASL L,BYT
1351 006744 106300 ASLB L,BYT
1352 006746 106104 ROLB HOLD
1353 006750 106300 ASLB L,BYT
1354 006752 106104 ROLB HOLD
1355 006754 106300 ASLB L,BYT
1356 006756 106104 ROLB HOLD
1357 006760 106300 ASLB L,BYT
1358 006762 106104 ROLB HOLD
1359 006764 010400 MOV HOLD,L,BYT
1360 006766 012604 MOV (SP)+,HOLD
1361 006770 000207 RTS PC
1362
1363
1364 006772 006100 GETB3: ROL L,BYT
1365 006774 106100 ROLB L,BYT
1366 006776 006004 ROR HOLD
1367 007000 106000 RORB L,BYT
1368 007002 006004 ROR HOLD
1369 007004 106000 RORB L,BYT
1370 007006 005726 TST (SP)+
1371 007010 000207 RTS PC
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
  
```

```

:GET AND ER CHARACTER NOW.
:AND PRESERVE IT FOR NEXT TIME THROUGH.
:NOW THROW AWAY LEFT MOST BITS OF
:THE 8 BIT CHARACTER. NOW MERGE IN
:THE LEFT TWO BITS OF THE
:NEW SIX BIT CHARACTER WITH THE SIX
:BITS FROM THE CHARACTER ON THE
:STACK, 1ST CHARACTER IS NOW ASSEMBLED,
:SO WE'LL RETURN IT TO THE USER.
:AND THEN WE SHALL RETURN TO HIM.
  
```

```

:THE SECOND CHARACTER IS CREATED FROM
:THE 4 RIGHT BITS OF THE PREVIOUS CHARACTER
:AND THE FOUR MIDDLE BITS OF THE PRESENT
:8 BIT CHARACTER.
:WE WILL CREATE THE NEW 8 BIT
:IN THIS REGISTER, SINCE IT
:MORE CONVIENT. WE WILL MOVE OVER THE
:ANSWER AT THE END.
:ONE MORE TO GO
:DONE.
:BRING OVER THE VALUE.
:AND REMEMBER THE LAST CHARACTER WE RECEIVED.
:AND RETURN TO THE CALLER.
  
```

```

:FINAL CHARACTER IS EASY. JUST A
:SIMPLE MERGER OF LEFT TWO BITS OF
:PREVIOUS VALUE WITH RIGHT SIX BITS
:OF LAST (4TH) CHARACTER RECEIVED.

:AND WE ARE DONE.
:FINALLY THROW AWAY STACK.
:AND RETURN TO THE CALLER.
  
```

.SBTTL THE LOADER

```

1385
1386
1387
1388
1389
1390
1391
1392
1393
1394 007012 012737 173000 007010 LOADER: MOV #DSTOP, JMPADD-2 ;STOP THE GT40 BY INSERTING A "DSTOP" IN THE LI
1395
1396 007020 005003 CLR INDEX ;RESET THE 8 BIT ASSEMBLER TO THE FIRST CHAR
1397
1398
1399 007022 005005 L.LD2: CLR L,CKSM ;CLEAR THE CHECKSUM
1400 007024 004737 167114 JSR PC,L.PTR!160000 ;GET A BYTE NOW.
1401 007030 105300 DECB L,BYT ;IS IT A ONE (HEADER)?
1402 007032 001373 BNE L.LD2 ;NO. WAIT FOR THE ONE.
1403
1404 007034 004737 167114 JSR PC,L.PTR!160000 ;YES. SKIP OVER THE NEXT CHARACTER NOW.
1405
1406 007040 004737 167126 JSR PC,L.GWRD!160000 ;ASSEMBLE A WORD NOW.
1407 007044 010002 MOV L,BYT,L.BC ;MOVE OVER TO THE COUNTER.
1408 007046 162702 000004 SUB #4,L.BC ;REDUCE TO ACTUAL DATA COUNT.
1409 007052 022702 000002 CMP #2,L.BC ;ANY DATA AT ALL?
1410 007056 001433 BEQ L.JMP ;NO. MUST BE END
1411 007060 004737 167126 JSR PC,L.GWRD!160000 ;YES. ASSEMBLE A DATA WORD NOW.
1412 007064 010001 MOV L,BYT,L.ADR ;AND THIS MUST BE THE FIRST ADDRESS.
1413
1414
1415 007066 004737 167114 L.LD3: JSR PC,L.PTR!160000 ;GET A BYTE OF DATA NOW.
1416 007072 002006 BGE L.LD4 ;ALL DONE?
1417 007074 105705 TSTB L,CKSM ;YEP. COUNTER IS MINUS. CHECK CHECKSUM.
1418 007076 001751 BEQ L.LD2 ;CHECKSUM GOOD. GET NEXT COMMAND.
1419
1420
1421 007100 004337 166652 L.BAD: JSR SCAN,OUTLIT!160000 ;BAD LOAD INFORM HOST
1422 007104 175 102 .BYTE ALTHOOD,'B ;SEND ALTHOOD B
1423 007106 000646 BR PRESTRY ;AND RESTART THE DISPLAY.
1424
1425
1426 007110 110021 L.LD4: MOVB L,BYT,(L.ADR)+ ;INSERT BYTE INTO MEMORY.
1427 007112 000765 BR L.LD3 ;AND GET THE NEXT BYTE.
1428
1429
1430
1431 007114 004737 166664 L.PTR: JSR PC,GETB!160000 ;ASSEMBLE AN 8 BIT CHARACTER NOW.
1432 007120 060005 ADD L,BYT,L,CKSM ;UPDATE THE CHECKSUM NOW.
1433 007122 005302 DEC L,BC ;DECREMENT THE CHARACTER COUNTER.
1434 007124 000207 RTS PC ;AND RETURN TO THE CALLER NOW.
1435
1436
1437
1438 007126 004737 167114 L.GWRD: JSR PC,L.PTR!160000 ;ASSEMBLE A WORD. FIRST GET A CHARACTER
1439 007132 010046 MOV L,BYT,-(SP) ;AND SAVE IT.
1440 007134 004737 167114 JSR PC,L.PTR!160000 ;AND THEN GET ANOTHER ONE.

```





;THIS IS GT40 QUICK TEST  
;GIVES QUICK VISUAL TEST  
;OF CONDITION OF MACHINE  
;WITHOUT READING IN DIAG.

1479		
1480		
1481		
1482		
1483	100000	CHAR=100000
1484	104000	SHORTV=104000
1485	110000	LONGV=110000
1486	114000	POINT=114000
1487	120000	GRAPHX=120000
1488	124000	GRAPHY=124000
1489	130000	RELATV=130000
1490		
1491	002000	INT0=2000
1492	002200	INT1=2200
1493	002400	INT2=2400
1494	002600	INT3=2600
1495	003000	INT4=3000
1496	003200	INT5=3200
1497	003400	INT6=3400
1498	003600	INT7=3600
1499		
1500	000100	LPOFF=100
1501	000140	LPON=140
1502	000020	BLKOFF=20
1503	000030	BLKON=30
1504		
1505	000004	LINE0=4
1506	000005	LINE1=5
1507	000006	LINE2=6
1508	000007	LINE3=7
1509		
1510	160000	DJMP=160000
1511	164000	DNOP=164000
1512	170000	STATSA=170000
1513	173400	DSTOP=173400
1514		
1515	000300	LPLITE=300
1516	000200	LPDARK=200
1517	000040	ITAL0=40
1518	000060	ITAL1=60
1519	000004	SYNON=4
1520		
1521		
1522	174000	STATSB=174000
1523		
1524	000100	INCR=100
1525	040000	INTX=40000
1526	001777	MAXX=1777
1527	001377	MAXY=1377
1528	020000	MINUSX=20000
1529	020000	MINUSY=MINUSX
1530	017600	MAXSX=17600
1531	000077	MAXSY=77
1532	000100	MINSUY=100
1533		
1534		

;BRIGHTEST

;STOP INTERRUPT

;ITALICS OFF  
ON  
;SYNC ON

;LOAD GRAPH INCR  
;INTENSIFY BIT  
;BIGGEST X VECTOR  
;BIGGEST Y VECTOR  
;THE MINUS BIT

;BIGGEST X IN SHORTVEC  
Y IN  
;MINUS BIT FOR Y IN SHORTVEC

```

1535 007204 012737 167214 172000      MOV      #FILE0!160000,GT40PC      ;START THE GT40
1536 007212 000001                      WAIT                               ;AND WAIT
1537                                     FILE0: POINT!BLKOFF                ;POINT--INVISIBLE
1538 007214 114020                      0
1539 007216 000000                      MAXY
1540 007220 001377
1541                                     LONGV!INT0!LINE0                ;DRAW TOP LINE
1542 007222 112004                      INTX!MAXX
1543 007224 041777                      0
1544 007226 000000
1545                                     LONGV!INT2!LINE1
1546 007230 112405                      INTX                               ;DRAW LINE TO RIGHT
1547 007232 040000                      MINUSX!MAXY
1548 007234 021377
1549                                     LONGV!INT4!LINE2
1550 007236 113006                      INTX!MINUSX!MAXX                ;DRAW BOTTOM LINE
1551 007240 061777                      0
1552 007242 000000
1553                                     LONGV!INT6!LINE3
1554 007244 113407                      INTX
1555 007246 040000                      MAXY                               ;DRAW LINE TO LEFT
1556 007250 001377
1557                                     POINT
1558 007252 114000                      400
1559 007254 000400                      500
1560 007256 000500                      SHORTV!INT1
1561 007260 106200                      57677                            ;+X+Y
1562 007262 057677
1563 007264 106600                      SHORTV!INT3
1564 007266 077677                      77677                            ;+X-Y
1565 007270 107200                      SHORTV!INT5
1566 007272 077777                      77777                            ;-X-Y
1567 007274 107600                      SHORTV!INT7
1568 007276 057777                      57777                            ;-X+Y
1569
1570                                     POINT
1571 007302 001400                      1400
1572 007304 000500                      500
1573 007306 133030                      RELATV!INT4!BLKON
1574 007310 057677                      57677                            ;+X+Y
1575 007312 077677                      77677                            ;+X-Y
1576 007314 077777                      77777                            ;-X-Y
1577 007316 057777                      57777                            ;-X+Y
1578
1579                                     POINT
1580 007322 000400                      400
1581 007324 000100                      100
1582 007326 174120                      STATSB!INCR+20                    ;TRY GRAPH MODES
1583 007330 114000                      POINT
1584 007332 001000                      1000
1585 007334 000200                      200
1586
1587 007336 120000                      GRAPHX
1588 007340 001010                      1010
1589 007342 001020                      1020
1590 007344 001030                      1030

```

1591	007346	001040	1040
1592	007350	001050	1050
1593			
1594	007352	114000	POINT
1595	007354	001000	1000
1596	007356	001200	1200
1597			
1598	007360	124000	GRAPHY
1599	007362	001020	1020
1600	007364	001030	1030
1601	007366	001040	1040
1602	007370	001050	1050
1603	007372	001060	1060
1604			
1605	007374	160000	DJMP
1606	007376	167214	FILED!!160000
1607			
1608			.SBTTL PAPER TAPE BOOT

```

1609
1610      ; PAPER TAPE BOOT
1611
1612      177550      HSR=177550      ;HIGH SPEED READER ADDRESS
1613      177560      LSR=177560      ;LOW SPEED READER ADDRESS
1614
1615      ;=ORIGIN+1400
1616      PTBOOT: MOV      #160000,R1      ;SET MEMORY CHECK LIMITS
1617      MOV      #4,R2      ;TRAP ADDRESS IS LOC. 4
1618      MOV      @DEV+4!160000,R3      ; POINTER TO DEVICE ADDRESSES
1619      MOV      PC,@R2      ;PRESET TRAP ADDRESS IN LOC. 4
1620      MOV      #24,SP      ;STACK SET UP AT SPECIAL ADDRESS
1621      DEV1:  MOV      -(R3),R4      ;GET DEVICE ADDRESS
1622      TST      @R4      ;CHECK AVAILABILITY OF DEVICE
1623      BMI      DEV1      ;CHECK DEVICE FOR ERRORS
1624      MOV      PC,@R2      ;RESET TRAP ADDRESS AT LOC. 4
1625      MOV      #24,SP      ;SPECIAL ADDRESS USED AS MASK LATER
1626      MOV      R4,-(R1)      ;DO MEM CHK:READER STATUS ADDRESS
1627      ;IS MOVED
1628      BIC      SP,R1      ;SET R1=X7752,MASK IN SP=24
1629      MOV      R1,@R1      ;STORE OWN ADDRESS IN POINTER
1630      LOOP:  MOV      @R1,R2      ;GET BYTE POINTER
1631      INC      @R4      ;ENABLE READER
1632      TSTB   @R4      ;TEST DONE BIT
1633      BPL      -2      ;WAIT UNTIL READY
1634      MOVB   2(R4),@R2      ;THEN PICK IT UP AND STORE IT
1635      INC      @R1      ;BUMP POINTER
1636      CMPB   R2,#375      ;STORED JUMP OFFSET?
1637      BNE      LOOP      ;NOT YET
1638      INCB   (R2)+      ;YES,ALL DONE
1639      JMP      -(R2)      ;GO EXECUTE AS BRANCH
1640
1641      ; DEVICE ADDRESSES FOLLOW - DO NOT CHANGE THE ORDER
1642
1643      DEV:   LSR      ;LOW SPEED READER
1644      HSR      ;HIGH SPEED READER
1645
1646      .SBTTL  CASSETTE BOOT

```

```

1647
1648
1649
1650          177500
1651
1652 007500 012700 177500
1653 007504 005010
1654 007506 010701
1655 007510 062701 000052
1656 007514 012702 000375
1657 007520 112103
1658
1659 007522 112110
1660 007524 100413
1661 007526 130310
1662 007530 001776
1663 007532 105202
1664 007534 100772
1665 007536 116012 000002
1666 007542 120337 000000
1667 007546 001767
1668 007550 000000
1669 007552 000755
1670
1671 007554 005710
1672 007556 100774
1673 007560 005007
1674
1675 007562 017640
1676
1677 007564 002415
1678
1679 007566 112024
1680
1681 007570 000000 000000
1682 007574 167500
1683 007576 000340
1684
1685
1686

; CASSETTE BOOT
TACS=177500 ;TA-11 CONTROL AND STATUS REGISTER
TABOOT: .=ORIGIN+1500
        MOV #TACS,R0
        CLR (R0) ;SELECT UNIT #0
RES:    MOV PC,R1 ;USE FOR PIC
        ADD #TABLE-.,R1 ;R1 HOLDS ADDR. OF COMMAND TABLE
        MOV #375,R2 ;MEMORY PTR. AND DATA FLAG
        MOVB (R1)+,R3 ;TEST BITS

LOOP1:  MOVB (R1)+,(R0) ;COMMAND FROM TABLE TO TACS
        BMI DONE ;WHEN COMMAND CODE NEG. QUIT
LOOP2:  BITB R3,(R0) ;TEST READY AND T-REQ BITS IN TACS
        BEQ LOOP2 ;LOOP 'TIL SOMETHING COMES UP
        INCB R2 ;ADVANCE MEMORY POINTER
        BMI LOOP1 ;IF MINUS, TRY NEXT COMMAND
        MOVB 2(R0),(R2) ;READ DATA INTO MEMORY
        CMPB R3,#0 ;FIRST BYTE READ SHOULD BE '240'
        BEQ LOOP2 ;IF O.K., GO READ ANOTHER BYTE
STOP:   HALT ;HALT ON ERROR
        BR RES ;RESTART ON CONTINUE

DONE:   TST (R0) ;CHECK FOR ERROR
        BMI STOP ;HALT ON ERROR
        CLR PC ;='JMP #0'

TABLE:  .WORD 17640 ;.BYTE 240: READY+T-REQ.
        .WORD 2415 ;.BYTE 37: ILBS+READY+GO
        .WORD 112024 ;.BYTE 15: SFB+GO
        .WORD 0 ;.BYTE 5: READ+GO
        .WORD 0 ;.BYTE 24: READ+ILBS
        .WORD 0 ;.BYTE 224: READ+ILBS+E.O.TABLE
        .WORD TABOOT!160000 ;THESE ARE FILLER WORDS
        .WORD 340 ;POWER UP VECTOR AND PRIORITY

.SBTTL MR11-DB BOOT

```

```

1687 ;MR11-DB BULK STORAGE PROGRAM LOADER LISTING
1688 ;
1689 ; .=ORIGIN+1600 ;KEEP TRACK OF ORIGIN
1690
1691 007600 010702 RF11: MOV PC,R2 ;FIXED HEAD DISK (256 KW)
1692 007602 000451 BR OTHER
1693 007604 177462 177462
1694 007606 000005 5
1695
1696 007610 010702 RK11: MOV PC,R2 ;MOVING HEAD DISK (CARTRIDGE)
1697 007612 000445 BR OTHER
1698 007614 177406 177406
1699 007616 000005 5
1700
1701
1702 007620 010702 TC11: MOV PC,R2
1703 007622 000417 BR TAPES
1704 007624 177344 177344 ;ADDRESS OF WORD COUNT
1705 007626 000005 5 ;LAST COMMAND
1706 007630 004003 4003 ;FIRST COMMAND
1707 007632 100000 100000 ;DONE MASK
1708 007634 024000 24000 ;ERROR MASK
1709
1710
1711 007636 010702 TM11: MOV PC,R2
1712 007640 000410 BR TAPES
1713 007642 172524 172524 ;ADDRESS OF BYTE COUNT
1714 007644 060003 60003 ;LAST COMMAND
1715 007646 060011 60011 ;FIRST COMMAND
1716 007650 000200 200 ;DONE MASK
1717 007652 100000 100000 ;ERROR MASK
1718
1719
1720 007654 010702 RP11: MOV PC,R2 ;MOVING HEAD DISK (PACK)
1721 007656 000423 BR OTHER
1722 007660 176716 176716
1723
1724
1725 007662 000005 TAPES: RESET
1726 007664 010200 MOV R2,R0 ;GET THE ADDRESS OF THE BRANCH
1727 007666 005720 TST (0)+ ;RD TO POINT AT LAST COMMAND
1728 007670 012001 MOV (0)+,R1 ;GET THE WORD COUNT ADDRESS
1729 007672 005311 DEC (1) ;SET UP FOR ADVANCE 1 RECORD
1730 007674 005720 TST (0)+ ;MOVE RD TO FIRST COMMAND
1731 007676 012041 MOV (0)+,-(1) ;COMMAND WORD TO COMMAND REG.
1732 007700 031011 BIT (0),(1) ;LOOK FOR DONE INDICATORS
1733 007702 001776 BEQ ,-2 ;NONE SET, TRY AGAIN
1734 007704 005720 TST (0)+ ;DONE FIRST COMMAND, CHECK FOR ERROR
1735 007706 031041 BIT (0),-(1) ;LOOK FOR SET ERROR BITS
1736 007710 001406 BEQ OTHER ;NO ERRORS - TRY THE READ
1737 007712 000112 AGAIN: JMP (2) ;RERUN FOR ERRORS
1738
1739
1740 007714 167600 RFVEC: RF11!160000 ;RF11 POWER UP VECTOR
1741 007716 000340 340
1742

```

# M03

SCROLLING ROM BOOTSTRAP FOR THE GT40  
DOGTEB.P11 MR11-DB BOOT

MACY11 27(732) 09-SEP-76 15:20 PAGE 38

1743	007720	010702		RC11:	MOV PC,R2		;FIXED HEAD DISK (64KW)
1744	007722	000401			BR OTHER		
1745	007724	177450			177450		;ADRS OF WORD COUNT (COMMAND+2)
1746							;COMMAND WORD (5) IS THE RESET
1747				OTHER:	RESET		
1748	007726	000005			MOV R2,R0		;R0 TO POINT AT WORD COUNT ADRS
1749	007730	010200			TST (0)+		;POINT TO ADDRESS
1750	007732	005720			MOV (0)+,R1		;WORD COUNT ADDRESS TO R1
1751	007734	012001			MOV #-1000,(1)		;LOAD WORD COUNT
1752	007736	012711	177000		MOV (0),-(1)		;COMMAND TO COMMAND REGISTER
1753	007742	011041			BIT #100200,(1)		;CHECK FOR ERROR OR DONE
1754	007744	032711	100200		BEQ -4		;IF NEITHER, KEEP LOOKING
1755	007750	001775			BMI AGAIN		;ERROR, TRY AGAIN
1756	007752	100757			CLR PC		
1757	007754	005007					
1758							
1759	007756	000000			0		;FILLER
1760	007760	167610		RKVEC:	RK:1!160000		;RK POWER UP VECTOR
1761	007762	000340			340		
1762	007764	167720		RCVEC:	RC11!160000		;RC POWER UP VECTOR
1763	007766	000340			340		
1764	007770	167654		RPVEC:	RP11!160000		;RP POWER UP VECTOR
1765	007772	000340			340		
1766	007774	167620		TCVEC:	TC11!160000		;TC11 POWER UP VECTOR
1767	007776	000340			340		
1768							
1769							

.SBTTL ROM VERSION 1 VALUES

1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825

000000  
000001  
000002  
000003  
000004  
000005  
000006  
000007  
  
000006  
000007  
  
000000  
000001  
000002  
000003  
000004  
000005  
  
000003  
000000  
000005  
000001

```

.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 BURNES, SENIOR SYSTEMS ARCHITECT!
;
;THIS ROUTINE IS INTENDED TO BE LOADED IN THE ROM PORTION OF THE VT-40.
;
;REGISTER DEFINITIONS:
;
;R0=%0
;R1=%1
;R2=%2
;R3=%3
;R4=%4
;R5=%5
;R6=%6
;R7=%7
;
;SP=R6
;PC=R7
;
;RET1=R0
;INP1=R1
;INP2=R2
;WORK1=R3
;WORK2=R4
;SCR1=R5
;
;RETURN OF VALUE REGISTER.
;ARGUMENT FOR CALLED FUNCTION
;SECOND ARGUMENT.
;FIRST WORK REGISTER.
;SECOND WORKING REGISTER.
;SCRATCH REGISTER.
;
;OVERLAPPING DEFINITIONS FOR LOADER PORTION.
;LCKSM=WORK1
;LBYT=RET1
;LBC=SCR1
;LADR=INP1

```



1826									
1827		036000				COREND=36000			;FIRST LOCATION OF NON-CORE.
1828		166000				ROMORG=166000			;WHERE THE ROM PROGRAM SHOULD GO.
1829									
1830		000000				STARTX=0			;WHERE TO START DISPLAYING THE X POSITIONS.
1831		001360				STARTY=1360			;WHERE TO START DISPLAYING THE Y.
1832									
1833									
1834		022000				VT40PC=172000-150000			;VT40 PROGRAM COUNTER.
1835		027560				KBOIS=27560			;TTY INPUT STATUS.
1836		025614				P100S=25614			;PDP-10 OUTPUT STATUS.
1837		025610				P10IS=25610			;PDP-10 INPUT STATUS.
1838									
1839		027562				KBOIB=KBOIS+2			;TTY INPUT BUFFER.
1840		025612				P10IB=P10IS+2			;PDP-10 INPUT CHARACTER.
1841		025616				P10OB=P10OS+2			;PDP-10 OUTPUT BUFFER.
1842									
1843									
1844		045776				P100C=COREND-2+10000			;CHARACTER TO BE SENT TO THE PDP-10
1845		045772				P10IC=P100C-4			;INPUT CHARACTER FROM IO PLUS ONE SAVE CHARACTER
1846		015770				STKSRT=P10IC-2-30000			;FIRST LOCATION OF STACK.
1847									
1848									
1849		160000				JMPDIS=160000			;THE VT-40 DISPLAY JUMP INSTRUCTION.
1850									
1851									
1852		000024				PWRFAL=24			;POWER FAIL RESTART LOCATION.
1853									
1854									
1855									
1856									
1857									
1858									
1859									
1860									
1861									
1862									
1863									
1864		016000				.=16000			
1865						.=ROMORG			;SET THE ORIGIN NOW!!!!
1866									
1867									
1868									
1869									
1870									
1871									
1872									
1873	016000	012705	000026		STARTA:	MOV	#PWRFAL+2, SCR1		;PICK UP POINTER TO P.F. STATUS.
1874	016004	005015				CLR	2SCR1		;CLEAR IT OUT TO BE SURE.
1875	016006	010745				MOV	PC, -(SCR1)		;SET UP THE RESTART LOCATION.
1876									
1877	016010	000005				RESET			;RESET THE BUS.
1878									
1879	016012	012767	000007	007570		MOV	#7, P10IS		;INITIALIZE PDP-10 INPUT
1880	016020	012767	000001	011532		MOV	#1, KBOIS		;INITIALIZE TTY INPUT.
1881	016026	012767	000201	007560		MOV	#201, P10OS		;INITIALIZE PDP-10 OUTPUT.

1892							
1893							
1894							
1895	016034	012706	015770	RSTRT:	MOV	#STKSRT, SP	:SET UP THE STACK NOW!
1896	016040	005001			CLR	LADR	:CLEAR ADDRESS POINTER.
1897	016042	012702	160000		MOV	#JMPDIS, INP2	:PLACE A DISPLAY JUMP INSTRUCTION IN A REGISTER.
1898	016046	010221			MOV	INP2, (LADR)+	:MOVE IT TO LOCATION 0.
1899	016050	012711	166756		MOV	#DISPRG+150000, (LADR)	:MOVE ADDRESS POINTER INTO 2.
1900	016054	012701	000030		MOV	#PWRFL+4, LADR	:SET UP WHERE WE WILL STORE CHARACTERS.
1891	016060	005000			CLR	RET1	:PREPARE TO INSERT A ZERO CHARACTER.
1892	016062	004767	000022		JSR	PC, DOCHAR	:INSERT IT NOW.
1893	016066	005067	003706		CLR	VT40PC	:CLEAR THE DISPLAY PROGRAM COUNTER AND START.
1894							
1895	016072	004767	000210	MAJOR:	JSR	PC, GTCHR	:GT A CHARACTER NOW.
1896	016076	000240			NOP		
1897	016100	000240			NOP		
1898	016102	000240			NOP		
1899	016104	012746	166072		MOV	#MAJOR+150000, -(SP)	:INSERT IN DISPLAY BUFFER NOW.
1900							
1901	016110	010105		DOCHAR:	MOV	LADR, SCR1	:GT CURRENT BUUFER POSITION NOW.
1902	016112	022525			CMP	(SCR1)+, (SCR1)+	:BYPASS CURRENT DISPLAY JUMP.
1903	016114	005025			CLR	(SCR1)+	:CLEAR FUTURE ADDRESS FOR JUMP.
1904	016116	010225			MOV	INP2, (SCR1)+	:STICK IN TEMPORARY JUMP WHILE WE REPLACE CURREN
1905	016120	005015			CLR	(SCR1)	:A DISPLAY JUMP TO ZERO.
1906	016122	005011			CLR	(LADR)	:NOW REPLACE CURRENT DISPLAY JUMP BY THE CHARACT
1907	016124	050021			BIS	RET1, (LADR)+	:IT'S DONE THIS WAY TO WASTE 2 CYCLES.
1908	016126	010211			MOV	INP2, (LADR)	:TO AVOID TIMING PROBLEMS WITH THE VT40.
1909	016130	000207			RTS	PC	:AND FINALLY RETURN.
1910							
1911							
1912							
1913							
1914							
1915							
1916							
1917							
1918							
1919							
1920							
1921							
1922							
1923							
1924							
1925							
1926	016132	004767	000124	GT8:	JSR	PC, GTSIX	:GT SIX BITS NOW.
1927	016136	010046			MOV	RET1, -(SP)	:SAVE THE CHARACTER NOW.
1928	016140	000401			BR	GTP84	:BYPASS THE B'ER
1929	016142	005002		GT84:	CLR	INP2	:RESET THE MAGIC REGISTER NOW.
1930	016144	005722		GTP84:	TST	(INP2)+	:INCREMENT WHERE TO GO.
1931	016146	066207	166250		ADD	GT8TB+150000(INP2), PC	:UPDATE PC NOW.
1932							
1933		016152		GT8P=.			
1934							
1935	016152	004767	000104	GT81:	JSR	PC, GTSIX	:GT A CHARACTER NOW.
1936	016156	010004			MOV	RET1, WORK2	:SAVE FOR A SECOND.
1937	016160	006300			ASL	RET1	

1938	016162	006300		ASL	RET1	;SHIFT TO LEFT OF BYTE
1939	016164	106300		ASLB	RET1	
1940	016166	106116		ROLB	2SP	;PACK THEM IN.
1941	016170	106300		ASLB	RET1	
1942	016172	106116		ROLB	2SP	;A GOOD 8 BIT THING.
1943	016174	012600		MOV	(SP)+,RET1	;POP AND RETURN NOW.
1944	016176	000207		RTS	PC	
1945						
1946	016200	006300	GT82:	ASL	RET1	;WORST CASE. SHIFT 4
1947	016202	006300		ASL	RET1	
1948	016204	106300		ASLB	RET1	
1949	016206	106104		ROLB	WORK2	
1950	016210	106300		ASLB	RET1	
1951	016212	106104		ROLB	WORK2	
1952	016214	106300		ASLB	RET1	
1953	016216	106104		ROLB	WORK2	
1954	016220	106300		ASLB	RET1	
1955	016222	106104		ROLB	WORK2	
1956	016224	010400		MOV	WORK2,RET1	
1957	016226	012604		MOV	(SP)+,WORK2	
1958	016230	000207		RTS	PC	
1959						
1960	016232	006100	GT83:	ROL	RET1	
1961	016234	006100		ROL	RET1	
1962	016236	006004		ROR	WORK2	
1963	016240	106000		RORB	RET1	
1964	016242	006004		ROR	WORK2	
1965	016244	106000		RORB	RET1	;FINAL CHARACTER ASSEMBLED.
1966	016246	005726		TST	(SP)+	;FLUDGE STACK.
1967	016250	000207		RTS	PC	;AND RETURN NOW.
1968						
1969		016250	GT8TB	=	.-2	;PUSH ZERO CONDITION BACK INTO NEVER-NEVER LAND.
1970						
1971	016252	000000		.WORD	GT81-GT8P	
1972	016254	000026		.WORD	GT82-GT8P	
1973	016256	000060		.WORD	GT83-GT8P	
1974	016260	177770		.WORD	GT84-GT8P	
1975						
1976						
1977	016262	004767	000020	GTSIX:	JSR	PC,GTCHR
1978	016266	020027	000040		CMF	RET1,#40
1979	016272	002546			BLT	LBAD
1980	016274	020027	000137		CMF	RET1,#137
1981	016300	003143			BGT	LBAD
1982	016302	000207			RTS	PC
1983						
1984						
1985						
1986	016304	005726		GTCHP:	TST	(SP)+
1987						;UPDATE THE STACK.
1988	016306	012700	015772	GTCHR:	MOV	#P10IC-30000,RET1
1989	016312	004767	000064	GTCHL:	JSR	PC,CHECK
1990	016316	005710			TST	2RET1
1991	016320	001774			BEG	GTCHL
1992	016322	011046			MOV	2RET1,-(SP)
1993	016324	005020			CLR	(RET1)+
						;SET UP POINTER TO THE INPUT CHARACTER.
						;ANY CHARACTERS THERE?
						;PUSH THE CHAR ON THE STACK.
						;CLEAR THE CHAR GOT FLAG NOW.

1994	016326	042716	177600		BIC	0-200, (SP)		; CLEAR AWAY PARITY NOW.
1995	016332	001764			BEQ	GTCHP		; IF ZERO, GT ANOTHER
1996	016334	022716	000177		CMP	0177, (SP)		
1997	016340	001761			BEQ	GTCHP		; ALSO IGNORE ABORTS.
1998	016342	022710	000175		CMP	0175, 0RET1		; WAS IT A "175"
1999	016346	001007			BNE	GTNP		; NOPE.
2000	016350	011610			MOV	(SP), 0RET1		; YEP. RESET IN CASE OF ABORT.
2001	016352	021027	000122		CMP	0RET1, 0122		; IS IT AN R
2002	016356	001626			BEQ	RSTR1		; YEP. RESTART
2003	016360	021027	000114		CMP	0RET1, 0114		; IS IT AN L
2004	016364	001455			BEQ	LOAD		; YEP. LOAD.
2005								
2006	016366	011610			GTNP:	MOV	(SP), 0RET1	; NOW DO THE FDUGING.
2007	016370	012600				MOV	(SP), 0RET1	
2008	016372	020027	000175			CMP	RET1, 0175	
2009	016376	001743				BEQ	GTCHR	; IF ALTMODE, LOOP
2010	016400	000207				RTS	PC	
2011								
2012								
2013								
2014								
2015								
2016								
2017								
2018								
2019	016402	005767	027370		CHECK:	TST	P100C	; DO WE WANT TO OUTPUT?
2020	016406	001410				BEQ	CHECK1	; NO.
2021	016410	105767	007200			TSTB	P100S	; WE DO. IS THE IO READY?
2022	016414	100005				BPL	CHECK1	; NOT QUITE.
2023	016416	016767	027354	007172		MOV	P100C, P100B	; IT'S READY. SEND THE CHARACTER.
2024	016424	105067	027346			CLR	P100C	; AND THE SAVED CHARACTER.
2025								
2026	016430	105767	011124		CHECK1:	TSTB	KBDIS	; HEY, IS THE KEYBOARD READY?
2027	016434	100014				BPL	CHECK3	; NOPE. NO LUCK.
2028	016436	116746	011120			MOVB	KBDIB, -(SP)	; YEP. SAVE THE CHARACTER NOW.
2029	016442	012767	000001	011110		MOV	01, KBDIS	; AND REENABLE THE COMMUNICATIONS DEVICE.
2030								
2031	016450	004767	177726		CHECK2:	JSR	PC, CHECK	; IS THE OUTPUT READY?
2032	016454	005767	027316			TST	P100C	
2033	016460	001373				BNE	CHECK2	; IF NOT, WAIT TILL DONE.
2034	016462	012667	007130			MOV	(SP)+, P100B	; AND THEN SEND OUT THE CHARACTER.
2035								
2036								
2037	016466	105767	007116		CHECK3:	TSTB	P101S	; IS THE IO TALKING TO ME.
2038	016472	100011				BPL	CHECK4	; NOPE. EXIT.
2039	016474	116767	007112	027270		MOVB	P101B, P101C	; GT THE CHARACTER NOW.
2040	016502	052767	177400	027262		BIS	0-400, P101C	; MAKE SURE IT'S NONE ZERO.
2041	016510	012767	000007	007072		MOV	07, P101S	; REINITIALIZE COMMUNICATION LINE.
2042								
2043	016516	000207			CHECK4:	RTS	PC	; AND RETURN.
2044								
2045								
2046								
2047								
2048								
2049								

2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105

016520	005002		
016522	012712	172000	
016526	012706	015770	
016532	005003		
016534	004767	000070	
016540	105300		
016542	001373		
016544	004767	000060	
016550	004767	000072	
016554	010005		
016556	162705	000004	
016562	022705	000002	
016566	001437		
016570	004767	000052	
016574	010001		
016576	004767	000026	
016602	002010		
016604	105703		
016606	001751		
015610	012700		
016612	102	175	
016614	004767	000110	
016620	000167	177210	
016624	110021		
016626	000763		
016630	004767	177276	
016634	060003		
016636	042700	177400	
016642	005305		
016644	000207		
016646	004767	177756	
016652	010046		
016654	004767	177750	
016660	000300		
016662	052600		
016664	000207		

THE LOADER

```

LOAD: CLR INP2 ; RESET TO FIRST 8 BIT CHARACTER
      MOV #172000,(INP2) ; AND ALSO CLEVERLY STOP THE VT40.
      MOV #STKSRT,SP ; RESET STACK POINTER NOW.

LLD2: CLR LCKSM ; CLEAR THE CHECKSUM
      JSR PC,LPTR ;GT A BYTE NOW.
      DECB LBYT ; IS IT ONE?
      BNE LLD2 ; NOPE. WAIT AWHILE
      JSR PC,LPTR ; YEP. GT NEXT CHARACTER.

      JSR PC,LGWRD ;GT A WORD.
      MOV LBYT,LBC ;GT THE COUNTER NOW.
      SUB #4,LBC ; CHOP OFF EXTRA STUFF.
      CMP #2,LBC ; NULL?
      BEQ LJMP ; YEP. MUST BE END.
      JSR PC,LGWRD ; NOPE. GT THE ADDRESS.
      MOV LBYT,LADR ; AND REMEMBER FOR OLD TIMES SAKE.

LLD3: JSR PC,LPTR ;GT A BYTE (DATA)
      BGE LLD4 ; ALL DONE WITH THE COUNTER?
      TSTB LCKSM ; YEP. GOOD CHECK SUM?
      BEQ LLD2 ; NOPE. LOAD ERROR.

LBAD: MOV (PC)+,RET1 ; SEND OUT SOME CHARACTERS NOW.
      .BYTE 175,102 ; "CTRL BAD"
      .BYTE 102,175 ; "BAD CTRL"
      JSR PC,SENDIT
      JMP RSTRT

LLD4: MOVB LBYT,(LADR)+ ; PLACE THE BYTE IN CORE.
      BR LLD3 ;GT ANOTHER ONE.

LPTR: JSR PC,GT8 ;GT 8 BITS NOW.
      ADD LBYT,LCKSM ; UPDATE CHECKSUM
      BIC #177400,LBYT ; CLEAN UP THE BYTE NOW.
      DEC LBC ; UPDATE THE COUNTER.
      RTS PC ; RETURN NOW.

LGWRD: JSR PC,LPTR ;GT A CHARACTER.
      MOV LBYT,-(SP) ; SAVE FOR A SECOND.
      JSR PC,LPTR ;GT ANOTHER CHARACTER.
      SWAB LBYT ; NOW ASSEMBLE THE WORD.
      BIS (SP)+,LBYT ; AND RETURN WITH A 16 BITER.
      RTS PC
    
```

```

2106
2107 016666 004767 177754
2108 016672 010046
2109 016674 004767 177730
2110 016700 105703
2111 016702 001342
2112 016704 032716 000001
2113 016710 001406
2114 016712 012700
2115
2116 016714 107 175
2117 016716 004767 000006
2118 016722 000000
2119 016724 000776
2120
2121 016726 000136
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139 016730 004767 177446
2140 016734 005767 027036
2141 016740 001373
2142 016742 010067 006650
2143 016746 105000
2144 016750 000300
2145 016752 001366
2146 016754 000207
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161

```

```

LJMP: JSR PC, LGWRD ;GT A WORD
      MOV LBYT, -(SP) ;SAVE ON THE STACK.
      JSR PC, LPTR ;GT A CHARACTER.
      TSTB LCKSM ;IS IT ZERO?
      BNE LBAD ;YEP. WHAT CRAP.
      BIT #1, (SP) ;IS IT 000?
      BEQ LJMP1 ;YEP. START PROGRAM GOING NOW.
      MOV (PC)+, RET1 ;TELL POP-10 WE'VE LOADED OK.
      .BYTE 175, 107 ;"CTRL GOOD"
      .BYTE 107, 175 ;"GOOD CTRL"
      JSR PC, SENDIT
      HALT
      BR -.2
LJMP1: JMP @ (SP)+ ;AND AWAY WE GO.

```

```

SENDIT: JSR PC, CHECK ;POLL THE OUTPUT DEVICE NOW.
        TST P100C ;OUTPUT CLEAR?
        BNE SENDIT ;NOPE. LOOP AWHILE LONGER.
        MOV RET1, P100B ;SEND OUT THE CHARACTER.
        CLRB RET1 ;CLEAR THE BYTE.
        SWAB RET1 ;AND SWAP THEM NOW.
        BNE SENDIT ;IF NOT EQUAL, REPEAT.
        RTS PC

```

2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172 016756 170256  
2173 016760 115124  
2174 016762 000000  
2175 016764 001360  
2176 016766 100000  
2177 016770 160000  
2178 016772 000030  
2179 016774 C 0000  
2180 016776 000000  
2181  
2182  
2183  
2184  
2185  
2186  
2187 017000 000000  
2188  
2189 000001

```

;
; THIS IS THE INITIALIZING VT40 PROGRAM WHICH WILL
; JUMP TO THE PROGRAM AFTER THE POWER FAIL LOCATIONS
; WHICH WILL JUMP TO ZERO WHICH WILL JUMP BACK TO HERE.
;
DISPRG: .WORD 170256 ;LOAD STATUS REGISTER FOR NORMAL OPERATION.
        .WORD 115124 ;SET POINT MODE, "NORMAL".
        .WORD STARTX ;X COORDINATE
        .WORD STARTY ;Y COORDINATE
        .WORD 100000 ;SET CHARACTER MODE.
        .WORD JMPDIS ;THEN JUMP TO THE POWERFAIL LOCATION.
        .WORD PWRFAIL+4 ;TO DISPLAY USERS CHARACTERS.
        .WORD 0
        .WORD 0

;STARTING FROM HERE TO THE TOP OF MEMORY
; A BACKGROUND WORSE CASE NOISE TASK WILL BE EXECUTED
BUFFER: 0
        .END

```





DOCORE	001636	222	338#					
DONE	007554	1660	1671#					
DSTOP =	173403	182#	806	1513#				
ENDCOR	006036	1002#	1003					
FF	006256	1089	1119#					
FFLOOP	006262	1114	1121#	1124				
FILED	007214	1535	1538#	1606				
FILEDA	005634	307*	409*	413*	797#			
FILEDB	005636	413	798#					
FILEDC	005664	307	409	797	806#			
FILEDD	002724	265	568#	808				
GETCHR	006564	1072	1262#	1284				
GETDL	006516	1240#	1245	1249	1251	1262	1266	
GETDL1	006546	1241	1248#					
GETEXT	006650	1264	1270	1290#				
GETSIX	006630	1284#	1322	1337				
GETB	006664	1322#	1431					
GETBTB	005700	1325	1327#					
GETB1	006712	1327	1337#					
GETB2	006740	1328	1349#					
GETB3	006772	1329	1364#					
GETB4	006706	1333#						
GRAPHX=	120000	182#	614	1487#	1587			
GRAPHY=	124000	182#	601	1488#	1598			
GRPINC	003006	308#	405*	406	408*	599#		
GTA00	001000	198#	278					
GTBRL	001004	200#	301	303	305			
GTBUSE	006424	1174#	1197					
GTCHL	016312	1989#	1991					
GTCHP	016304	1986#	1995	1997				
GTCHR	016306	1995	1977	1988#	2009			
GTDLY0	001300	276#	306*	402*	404*			
GTDNE1	001154	239#	301*					
GTDONE	001152	238#	300*					
GTLPEN	002134	302	413#					
GTLPH	001156	241#	302*					
GTLPH1	001160	241#	303*					
GTNP	016366	1959	2006#					
GTPC	001132	228#	265*	277	410*	414*		
GTPB4	016144	1928	1930#					
GTSHIF	002152	304	417#					
GTSIX	016262	1926	1935	1977#				
GTSOTH	001162	244#	304*					
GTSOT1	001164	245#	305*					
GTSR	001134	229#	397	543*				
GTSTOP	002044	300	397#					
GTST1	002116	403	407	409#				
GTVCT	001002	199#	282					
GTXPOS	001136	230#						
GTYP05	001140	231#						
GT40PC=	172000	933#	934	1174*	1208	1535*		
GT40SR=	172002	934#	1116*					
GTB	016132	1926#	2094					
GTBP =	016152	1933#	1971	1972	1973	1974		
GTBTB =	016250	1931	1969#					
GTB1	016152	1935#	1971					



LLD2	016532	2066#	2069	2083														
LLD3	016576	2080#	2092															
LLD4	016624	2081	2091#															
LOAD	016520	2004	2062#															
LOADER	007012	1268	1394#															
LOADRO	001364	279	281	283	289	291#												
LOGICA	002704	556#																
LONGV =	110000	182#	575	578	581	584	593	608	639	642	681	687	693	699				
		705	711	717	723	1485#	1542	1546	1550	1554								
LOOP	007444	1630#	1637															
LOOP1	007522	1659#	1664															
LOOP2	007526	1661#	1662	1667														
LPDARK=	000200	182#	1516#															
LPLITE=	000300	182#	571	1515#														
LPOFF =	000100	182#	639	1500#														
LPON =	000140	182#	568	643	1501#													
LPTR	016630	2067	2070	2080	2094#	2100	2102	2109										
LSR =	177560	1613#	1643															
L.AOR =x000001		910#	1412#	1426#														
L.BAD	007100	1286	1288	1421#	1452													
L.BC =x000002		911#	1407#	1408#	1409	1433#												
L.BYT =x000000		909#	1323	1338	1339#	1340#	1341#	1343#	1345#	1349#	1350#	1351#	1353#	1355#				
		1357#	1359#	1364#	1365#	1367#	1369#	1401#	1407	1412	1426	1432	1439	1441#				
		1442#	1449															
L.CKSM=x000005		912#	1399#	1417	1432#	1451												
L.GWRD	007126	1406	1411	1438#	1448													
L.HALT	007200	1460#																
L.JMP	007146	1410	1448#															
L.JMP1	007202	1458	1462#															
L.LD2	007022	1399#	1402	1418														
L.LD3	007066	1415#	1427															
L.LD4	007110	1416	1426#															
L.PTR	007114	1400	1404	1415	1431#	1438	1440	1450										
MAJOR	016072	1895#	1899															
MAXSX =	017600	182#	1530#															
MAXSY =	000077	182#	1531#															
MAXX =	001777	182#	576	582	1526#	1543	1551											
MAXY =	001377	182#	570	580	586	645	653	658	663	668	673	1527#	1540	1548				
		1556																
MEMDAT	001776	377#	378#	383	386													
MEMEND	002040	360	384#															
MEMTST	001770	358	375#															
MINSUY=	000100	182#	624	628	632	636	1532#											
MINUSX=	020000	182#	580	582	624	628	632	636	639	642	1528#	1529	1548	1551				
MINUSY=	020000	182#	1529#															
NORMAL	006212	1076	1085	1097#														
NOTHER	006042	1000	1006#															
NUMLIN=	000040	942#	1030	1031	1119	1205												
NXTCHR	006132	1072#	1074	1080	1099	1109	1117	1125	1128									
OCTA	005611	317#	318#	319#	492#	494#	496#	788#										
ORIGIN=	166000	923#																
OTHER	007726	1692	1697	1721	1736	1744	1748#											
OUTLIT	006652	996	1297#	1421	1454													
OVER	002544	225	513#	551														
PC =%000007		186#	212#	222#	223#	273#	279#	281#	283#	289#	298#	326#	367#	376#				
		491#	493#	495#	507#	540#	556#	881#	1072#	1082#	1097#	1105#	1112	1122#				

	1143*	1158*	1161*	1168*	1246*	1262*	1266*	1284*	1290*	1322*	1337*	1346*	1361*
	1371*	1400*	1404*	1406*	1411*	1415*	1431*	1434*	1438*	1440*	1443*	1448*	1450*
	1619	1624	1654	1673*	1691	1696	1702	1711	1720	1743	1757*	1810*	1875
	1892*	1895*	1909*	1926*	1931*	1935*	1944*	1958*	1967*	1977*	1982*	1989*	2010*
	2031*	2043*	2067*	2070*	2072*	2077*	2080*	2085	2088*	2094*	2098*	2100*	2102*
	2105*	2107*	2109*	2114	2117*	2139*	2146*						
PCOUNT 002722	513*	541*	564*										
POINT = 114000	182*	568	590	596	605	611	620	624	628	632	636	639	643
	651	656	661	666	671	678	684	690	696	702	708	714	720
	729	742	762	783	798	1486*	1538	1558	1570	1579	1583	1594	
POINTR=%000001	890*	910	1043*	1045*	1051*	1151*	1152	1154	1156	1157*	1163	1164*	1165*
	1166*												
PPNT 002344	270*	429	431*	432	434*	461*							
PRESTR 006624	1273*	1423											
PRIME 001220	223	265*											
PSW 001216	211*	224*	263*										
PTBOOT 007400	1616*												
PUNCHR 002350	430*	436	463*										
PWRFAL= 000024	1852*	1873	1890	2178									
PWRFL 001612	191	328*	332										
PWRUP 001622	328	331*											
P101B = 025612	1840*	2039											
P101C = 045772	1845*	1846	1988	2039*	2040*								
P101S = 025610	1837*	1840	1879*	2037	2041*								
P100B = 025616	1841*	2023*	2034*	2142*									
P100C = 045776	1844*	1845	2019	2023	2024*	2032	2140						
P100S = 025614	1836*	1841	1881*	2021									
RCVEC 007764	1762*												
RC11 007720	1743*	1762											
REDCHR 002352	449*	450*	451	464*									
RELATV= 130000	182*	623	627	1489*	1573								
RES 007506	1654*	1669											
RESTRT 006060	1027*	1273											
RET1 =%000000	1812*	1820	1891*	1907	1927	1936	1937*	1938*	1939*	1941*	1943*	1945*	1947*
	1948*	1950*	1952*	1954*	1956*	1960*	1961*	1963*	1965*	1978	1980	1988*	1990
	1992	1993*	1998	2000*	2001	2003	2006*	2007*	2008	2085*	2114*	2142	2143*
	2144*												
RFVEC 007714	1740*												
RF11 007600	1691*	1740											
RKVEC 007760	1760*												
RK11 007610	1696*	1760											
ROMADO 001014	206*	524											
ROMORG= 166000	1828*												
ROTVL 001766	369*	385											
RPNT 002346	271*	448	452*	453	455*	456	462*						
RPVEC 007770	1764*												
RP11 007654	1720*	1764											
RSTRT 016034	1885*	2002	2089										
RO =%000000	186*	277*	284*	285*	286*	287*	291*	292*	293*	294*	295*	296*	297*
	309*	310*	311	313*	314*	315	340*	342*	346*	354*	356	362	555*
	874*	889	1652*	1653*	1659*	1661	1665	1671	1726*	1749*	1800*	1812	
R1 =%000001	186*	278*	280*	282*	284	286	288*	291	292	294	296	339*	341*
	344	349*	350	355*	356	359*	362	364*	365*	366*	875*	890	1616*
	1626*	1628*	1629*	1630	1635*	1654*	1655*	1657	1659	1728*	1751*	1801*	1813
R2 =%000002	186*	358*	359	360	876*	891	1617*	1619*	1624*	1630*	1634*	1636	1638*
	1639	1656*	1663*	1665*	1691*	1696*	1702*	1711*	1720*	1726	1743*	1749	1802*





GRAPH	186#	602	616		
OCTGN	183#	624	628	632	636
OCTGON	184#	639	642		
PAT1	184#	649	655		
PAT2	185#	670	675		
PAT3	186#	660	665		

ADD	285	287	293	295	297	341	506	1082	1432	1655	1931	2095			
ASL	1081	1339	1340	1349	1350	1937	1938	1946	1947						
ASLB	1341	1343	1351	1353	1355	1357	1939	1941	1948	1950	1952	1954			
BCS	379														
BCB	384	387	528	551	1042	1134	1245	1268	1410	1418	1458	1662	1667	1733	1736
BGE	1755	1991	1995	1997	2002	2004	2009	2020	2076	2083	2113				
BGT	1074	1076	1416	2081											
BHIS	1035	1047	1124	1288	1981										
BIC	1080														
BIS	450	479	505	1141	1244	1628	1994	2096							
BIT	1027	1442	1907	2040	2104										
BITB	213	1107	1152	1457	1732	1735	1754	2112							
BLO	1661														
BLOS	357	1136	1155												
BLT	363														
BMI	1286	1979													
BME	398	423	443	471	1623	1660	1664	1672	1756						
BPL	214	312	316	345	361	403	407	433	454	483	535	542	1108	1153	1264
BR	1270	1402	1452	1637	1999	2033	2069	2111	2141	2145					
BVC	546	549	1014	1241	1249	1633	2022	2027	2038						
CLR	217	290	530	1003	1048	1084	1085	1086	1087	1088	1089	1099	1109	1114	1117
CLRB	1125	1138	1251	1327	1328	1329	1423	1427	1669	1692	1697	1703	1712	1721	1744
CMP	1928	2092	2119												
CMPB	381														
DEC	220	221	224	271	272	310	314	340	365	366	434	455	484	1002	1015
DECB	1116	1144	1159	1160	1166	1396	1399	1653	1673	1757	1874	1886	1891	1893	1903
HALT	1905	1906	1929	1993	2024	2062	2066								
INC	317	318	319	2143											
INCB	311	315	344	348	356	360	362	383	386	406	432	453	482	527	532
JMP	534	1073	1075	1079	1135	1154	1163	1263	1267	1269	1285	1287	1409	1902	1978
JSR	1980	1996	1998	2001	2003	2008	2075								
MOV	1133	1636	1666												
	346	402	541	1034	1046	1123	1433	1729	2097						
	1401	2068													
	188	189	190	329	380	382	385	388	399	417	424	444	472	529	1460
	1668	2118													
	342	405	431	452	481	533	995	1098	1106	1140	1631	1635			
	1638	1663													
	195	225	333	400	418	425	445	473	561	1273	1325	1462	1639	1737	2089
	2121														
	212	222	223	279	281	283	289	491	493	495	540	556	996	1072	1097
	1105	1122	1143	1158	1262	1266	1284	1323	1337	1400	1404	1406	1411	1415	1421
	1431	1438	1440	1448	1450	1454	1892	1895	1926	1935	1977	1989	2031	2067	2070
	2072	2077	2080	2088	2094	2100	2102	2107	2109	2117	2139				
	210	211	215	216	218	219	265	266	267	268	270	277	278	280	282
	284	286	288	291	292	294	296	300	301	302	303	304	305	306	307
	308	309	313	320	321	322	323	324	325	328	332	338	339	347	350
	354	355	358	359	364	376	404	408	409	410	413	414	427	429	437
	447	448	456	458	475	476	477	485	490	497	498	504	513	522	523
	524	525	526	543	544	547	555	992	993	999	1000	1011	1030	1031	1033
	1038	1041	1043	1045	1051	1077	1093	1104	1119	1121	1128	1131	1137	1142	1156
	1157	1164	1165	1174	1243	1272	1323	1333	1338	1345	1359	1360	1394	1407	1412
	1439	1449	1535	1616	1617	1618	1619	1620	1621	1624	1625	1626	1629	1630	1652
	1654	1656	1691	1696	1702	1711	1720	1726	1728	1731	1743	1749	1751	1752	1753
	1873	1875	1879	1880	1881	1885	1887	1888	1889	1890	1899	1901	1904	1905	1927
	1936	1943	1956	1957	1988	1992	2000	2006	2007	2023	2029	2034	2041	2043	2064



MOV8	2073 269 1250	2078 430 1297	2085 436 1298	2101 449 1426	2108 451 1634	2114 457 1657	2142 478 1659	480 1665	486 2028	492 2039	494 2091	496	1112	1151	1242
NOP	557	558	559	560	1896	1897	1898								
RESET	331	553	553	554	991	1725	1748	1877							
ROL	1364	1960	1961												
ROLB	1342	1344	1352	1354	1356	1358	1365	1940	1942	1949	1951	1953	1955		
ROR	501	502	503	1356	1368	1962	1964								
RORB	378	1367	1369	1563	1965										
RTI	411	415	438	459	499										
RTS	273 1909	298 1944	326 1958	367 1967	507 1982	1161 2010	1168 2043	1246 2098	1290 2105	1299 2146	1346	1361	1371	1434	1443
SCC	375														
SUB	349	351	1078	1408	2074										
SWAB	1441	2103	2144												
TST	343 1990	397 2019	550 2032	1006 2140	1324	1370	1622	1671	1727	1730	1734	1750	1930	1966	1986
TSTB	422 2110	442	470	545	548	1013	1240	1248	1417	1451	1632	2021	2026	2037	2082
WAIT	1536														
.ASCII	649	655	733	735	746	766									
.ASCIZ	649	655	787	802											
.BYTE	660	665	670	675	748	755	768	775	788	1422	1455	2087	2116		
.DSABL	370	1770													
.ENABL	179	391	851												
.END	2189														
.EVEN	649	655	660	665	670	675	805								
.LIST	1	180	152	186											
.MACRU	183	184	185	186											
.MLIST	1	181	182	186											
.PAGE	863	910	968	1064	1224	1385	1479	1770							
.REM	1														
.REPT	190														
.S8TTL	862	967	1063	1223	1384	1466	1608	1646	1686	1769					
.TITLE	178	817													
.WORD	997	1195	1196	1197	1198	1200	1201	1202	1203	1204	1205	1207	1208	1209	1211
	1215	1216	1217	1218	1219	1220	1221	1675	1677	1679	1681	1682	1683	1971	1972
	1973	1974	2172	2173	2174	2175	2176	2177	2178	2179	2180				

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0

\*DOGTEB, DOGTEB. SEQ/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:DOGTEB.P11  
 RUN-TIME: 6 13 3 SECONDS  
 RUN-TIME RATIO: 104/24=4.3  
 CORE USED: 9K (17 PAGES)

F05

Special printing 9 000000, 00 000 100 000 000, 2 disk writes, 15 pages

