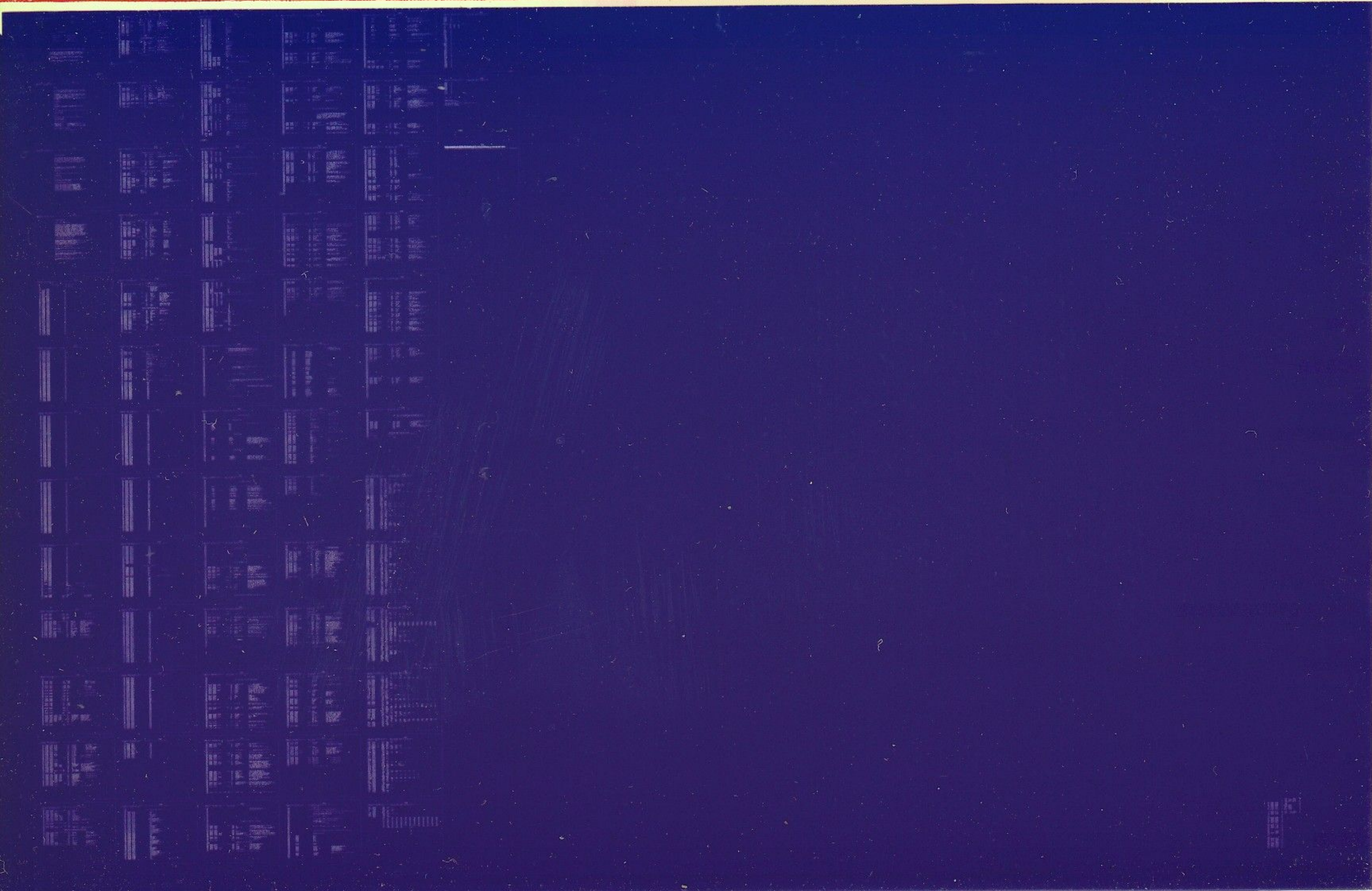


GT40

QUICK VERIFY
MD-11-DDGTE-C

EP-DDGTE-C-DL-B
COPYRIGHT © 1976
FICHE 1 OF 1

DEC 1976
digital
MADE IN USA



B01

SCROLLING ROM BOOTSTRAP FOR THE GT40
DDGTEC.P11 05-NOV-76 10:20

MAY11 27(1006) 05-NOV-76 12:10 PAGE 2

.REM

IDENTIFICATION

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

COPYRIGHT (C) 1973, 1976, 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 VERSION OF THE PROGRAM SUPPORTS NON-SWITCH REGISTER CPU'S. FOR THESE CPU'S, THE SWITCH REGISTER CAN BE CHANGED BY CHANGING THE CONTENTS OF SWREG (170).

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. REQUIREMENTS2.1 EQUIPMENT

GT40 SYSTEM (11/05, DISPLAY PROCESSOR AND VR14 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 PROCEDURE3.1 METHOD

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

4. STARTING PROCEDURE4.1 CONTROL SWITCH SETTINGS

CONSOLE SW 09 = 0	ROM PRESENT AS SPECIFIED BY SW 08
CONSOLE SW 09 = 1	NO ROM PRESENT
CONSOLE SW 08 = 0	TEST AS VERSION 2 ROM (512. WORDS)
CONSOLE SW 08 = 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. MISCELANEOUS

8.1 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.

351	000504	000506	.+2
352	000506	000000	HALT
353	000510	000512	.+2
354	000512	000000	HALT
355	000514	000516	.+2
356	000516	000000	HALT
357	000520	000522	.+2
358	000522	000000	HALT
359	000524	000526	.+2
360	000526	000000	HALT
361	000530	000532	.+2
362	000532	000000	HALT
363	000534	000536	.+2
364	000536	000000	HALT
365	000540	000542	.+2
366	000542	000000	HALT
367	000544	000546	.+2
368	000546	000000	HALT
369	000550	000552	.+2
370	000552	000000	HALT
371	000554	000556	.+2
372	000556	000000	HALT
373	000560	000562	.+2
374	000562	000000	HALT
375	000564	000566	.+2
376	000566	000000	HALT
377	000570	000572	.+2
378	000572	000000	HALT
379	000574	000576	.+2
380	000576	000000	HALT
381	000600	000602	.+2
382	000602	000000	HALT
383	000604	000606	.+2
384	000606	000000	HALT
385	000610	000612	.+2
386	000612	000000	HALT
387	000614	000616	.+2
388	000616	000000	HALT
389	000620	000622	.+2
390	000622	000000	HALT
391	000624	000626	.+2
392	000626	000000	HALT
393	000630	000632	.+2
394	000632	000000	HALT
395	000634	000636	.+2
396	000636	000000	HALT
397	000640	000642	.+2
398	000642	000000	HALT
399	000644	000646	.+2
400	000646	000000	HALT
401	000650	000652	.+2
402	000652	000000	HALT
403	000654	000656	.+2
404	000656	000000	HALT
405	000660	000662	.+2
406	000662	000000	HALT

K01

GT40 QUICK VERIFY MAINDEC-11-DDGTE-C
DDGTEC.P11 05-NOV-76 10:20

MACY11 27(1006) 05-NOV-76 12:10 PAGE 11

463	001006	175610			DLADD:	175610		:DL-11 ADDRESS
464	001010	000300			DLVCT:	300		:DL-11 VECTOR
465	001012	000240			DLBRL:	240		
466								
467	001014	166000			ROMADD:	166000		:ROM STARTING ADDRESS
468	001016	001000			WORDS:	512.		
469	001020	006000			IMAGE:	6000		
470								
471	001022	012706	000500		STARTB:	MOV	#500, SP	:LOAD THE STACK POINTER
472	001026	012777	000340	000176		MOV	#340, QPSW	:RAISE PSW
473	001034	004737	001316			JSR	PC, INITGT	:INIT DEVICE ADDRESSES
474	001040	005037	001020			CLR	IMAGE	:PRESET FOR NO ROM SELECTED
475	001044	032777	001000	000156		BIT	#1000, QSWR	:TEST FOR ROM SELECTED SWITCH
476	001052	001021				BNE	2\$:NO ROM SELECTED
477	001054	032777	000400	000146		BIT	#400, QSWR	:TEST ROM SWITCH
478	001062	001007				BNE	1\$:BR IF SET
479	001064	012737	001000	001016		MOV	#512, WORDS	:ASSUME VER. 2 ROM
480	001072	012737	006000	001020		MOV	#START, IMAGE	:LOAD IMAGE ADDRESS
481	001100	000406				BR	2\$:START TEST
482	001102	012737	000400	001016	1\$:	MOV	#256, WORDS	:SELECT VER. 1 ROM
483	001110	012737	016000	001020		MOV	#START, IMAGE	:LOAD IMAGE ADDRESS
484	001116	005077	000042		2\$:	CLR	QDLODBR	:CLEAR OUTPUT
485	001122	005077	000036			CLR	QDLODBR	
486	001126	004737	001720			JSR	PC, DDCORE	:SET UP CORE SIZE
487	001132	004737	001234			JSR	PC, PRIME	:INIT THE DEVICES
488	001136	005077	000070			CLR	QPSW	
489	001142	000137	002630			JMP	OVER	:EXECUTE BACKGROUND TASK

L01

GT40 QUICK VERIFY MAINDEC-11-DDGTE-C
DDGTEC.P11 05-NOV-76 10:20

MACY11 27(1006) 05-NOV-76 12:10 PAGE 12

```

490
491
492 001146 172000      GTPC: 172000      ;DISPLAY PC
493 001150 172002      GTSR: 172002      ;DISPLAY STATUS REG.
494 001152 172004      GTXPOS: 172004    ;DISPLAY X REGISTER
495 001154 172006      GTYPOS: 172006    ;DISSPLAY Y REGISTER
496
497 001156 175610      DLICSR: 175610    ;DL-11 STATUS
498 001160 175612      DLIDBR: 175612    ;DL-11 BUFFER
499 001162 175614      DLOCSR: 175614    ;DL-11 STATUS
500 001164 175616      DLODBR: 175616    ;DL-11 BUFFER
501
502 001166 000320      GTDONE: 320       ;DISPLAY DONE VECTOR
503 001170 000322      GTDNE1: 322
504
505 001172 000324      GTLPH: 324        ;DISPLAY LIGHT-PEN VECTOR
506 001174 000326      GTLPH1: 326
507
508 001176 000330      GTSOTM: 330       ;DISPLAY SHIFT-OUT/ TIME-OUT VECTOR
509 001200 000332      GTSOT1: 332
510
511 001202 000300      DLIVT: 300
512 001204 000302      DLIVT1: 302
513 001206 000304      DLOVT: 304
514 001210 000306      DLOVT1: 306
515
516 001212 177560      TKS: 177560
517 001214 177562      TKB: 177562
518 001216 177564      TPS: 177564
519 001220 177566      TPB: 177566
520
521 001222 000060      KRBVT: 60
522 001224 000062      KRBVT1: 62
523
524 001226 000200      KRBBRL: 200
525
526 001230 177570      SWR: 177570
527 001232 177776      PSW: 177776
528
529 001234 012777 003012 177704 PRIME: MOV #FILE00, @GTPC ;START THE DISPLAY
530 001242 012777 000100 177712      MOV #100, @DLOCSR ;ENABLE DL OUTPUT
531 001250 012777 000100 177700      MOV #100, @DLICSR ;ENABLE DL INPUT
532 001256 012777 000100 177726      MOV #100, @TKS ;ENABLE KEYBOARD
533 001264 113777 005422 177672      MOVB BUFF1, @DLODBR ;OUTPUT A CHAR
534 001272 012737 000001 002430      MOV #1, @PNT ;PRESET PRINT POINTER
535 001300 005037 002432      CLR @PNT ;CLEAR READ BUFFER
536 001304 005037 002624      CLR @PNT
537 001310 000207      RTS PC ;EXIT
538
539 001312 017476      SIZE: 17476
540 001314 000000      GTDLY0: 0

```


MO1

GT40 QUICK VERIFY MAINDEC-11-DDGTE-C
DDGTEC.P11 05-NOV-76 10:20

MACY11 27(1006) 05-NOV-76 12:10 PAGE 13

541	001316	012700	001146		INITGT:	MOV	#GTFC,RO	;LOAD STARTING ADDRESS
542	001322	013701	001000			MOV	GTADD,R1	;SAVE VALUE
543	001326	004737	001400			JSR	PC,LOADRO	;LOAD GT ADDR
544	001332	013701	001006			MOV	DLADD,R1	;LOAD STARTING ADDRESS <DL-11>
545	001336	004737	001400			JSR	PC,LOADRO	;LOAD DL-11 ADDRESSES
546	001342	013701	001002			MOV	GTVCT,R1	;LOAD VECTOR VALUE
547	001346	004737	001400			JSR	PC,LOADRO	;LOAD GT-40 VECTORS
548	001352	010110				MOV	R1,(RO)	
549	001354	062720	000010			ADD	#10,(RO)+	;LOAD GT TIME-OUT
550	001360	010110				MOV	R1,(RO)	
551	001362	062720	000012			ADD	#12,(RO)+	
552	001366	013701	001010			MOV	DLVCT,R1	;LOAD VECTOR VALUE
553	001372	004737	001400			JSR	PC,LOADRO	;LOAD DL-11 VECTORS
554	001376	000436				BR	INGT	;BR
555	001400	010120				LOADRO: MOV	R1,(RO)+	;LOAD DONE
556	001402	010110				MOV	R1,(RO)	
557	001404	062720	000002			ADD	#2,(RO)+	
558	001410	010110				MOV	R1,(RO)	
559	001412	062720	000004			ADD	#4,(RO)+	;LOAD DONE
560	001416	010110				MOV	R1,(RO)	
561	001420	062720	000006			ADD	#6,(RO)+	;LOAD PSW
562	001424	013746	000004			MOV	@#ERRVEC, -(SP)	;SAVE VECTORS CONTENTS
563	001430	012737	001456	000004		MOV	#1\$, @#ERRVEC	;SET UP FOR TRAP
564	001436	012737	177570	001230		MOV	#DSWR, @#SWR	;SET UP TO TEST FOR SWITCH REGISTER
565	001444	022777	177777	177556		CMP	#-1, @#SWR	;TEST FOR SWITCH REGISTER
566	001452	001005				BNE	3\$;SWITCH REGISTER IS PRESENT
567	001454	000401				BR	2\$;NO SWITCH REGISTER
568	001456	022626			1\$:	CMP	(SP)+, (SP)+	;POP 2 WORDS OFF STACK
569	001460	012737	000170	001230	2\$:	MOV	#SWREG, @#SWR	;SET UP FOR SOFTWARE SWITCH REGISTER
570	001466	012637	000004		3\$:	MOV	(SP)+, @#ERRVEC	;RESTORE VECOTS CONTENTS
571	001472	000207				RTS	PC	;EXIT
572								
573	001474	012777	002130	177464		INGT: MOV	#GTSTOP, @GTDONE	;LOAD DONE VECTOR
574	001502	013777	001004	177460		MOV	GTBRL, @GTDNE1	
575	001510	012777	002220	177454		MOV	#GTLPEN, @GTLPH	;LOAD LIGHT-PEN VECTOR
576	001516	013777	001004	177450		MOV	GTBRL, @GTLPH1	
577	001524	012777	002236	177444		MOV	#GTSHIF, @GTSOTM	;LOAD SHIFT-OUT VECTOR
578	001532	013777	001004	177440		MOV	GTBRL, @GTSOT1	
579	001540	012737	000040	001314		MOV	#40, GTDLYD	
580	001546	012737	005752	005722		MOV	#FILEDC, FILEOA	
581	001554	012737	174104	003074		MOV	#STATSB! INCR+4, GRPINC	
582	001562	012700	005516			MOV	#BUFF2, RO	
583	001566	005020				INTD: CLR	(RO)+	
584	001570	022700	005566			CMP	#BUFF2+50, RO	
585	001574	001374				BNE	INTD	
586	001576	012700	005612			MOV	#BUFF3, RO	;SET UP KRB BUFFER


```

587 001602 005020
588 001604 022700 005662
589 001610 001374
590 001612 105037 005677
591 001616 105037 005700
592 001622 105037 005701
593 001626 012777 002326 177346
594 001634 013777 001012 177342
595 001642 012777 002244 177336
596 001650 013777 001012 177332
597 001656 012777 002440 177336
598 001664 013777 001226 177332
599 001672 000207
600
601 001674 012737 001704 000024 PWRFL: MOV #PWRUP, @#24 ;LOAD VECTOR
602 001702 000000 HALT
603
604 001704 000005 PWRUP: RESET
605 001706 012737 001674 000024 MOV #PWRFL, @#24
606 001714 000137 001022 JMP STARTB ;RESTART AT BEGINING
607
608 ;SUBROUTINE TO DETERMINE THE SIZE OF CORE
609 ; AND SET UP LOCATION SIZE WITH THE VALUE
610
611 001720 012737 001754 000004 DOCORE: MOV #2$, @#4 ;SET UP FOR NEM
612 001726 012701 017776 MOV #17776, R1 ;SET UP ADDRESS
613 001732 005000 CLR RO
614 001734 062701 020000 1$: ADD #20000, R1 ;MOVE TO THE NEXT BANK
615 001740 005200 INC RO ;INC BANK COUNTER
616 001742 005711 TST (1) ;TIMEOUT ?
617 001744 022701 157776 CMP #157776, R1 ;END ?
618 001750 001371 BNE 1$
619 001752 000404 BR 3$
620 001754 022626 2$: CMP (SP)+, (SP)+ ;POP THE STACK X2
621 001756 005300 DEC RO ;DECREMENT BANK COUNT
622 001760 162701 020000 SUB #20000, R1 ;RESTORE R1
623 001764 012737 000006 000004 3$: MOV #6, @#4 ;RESET BUSS ERROR
624 001772 010137 001312 MOV R1, SIZE
625 001776 162737 007776 001312 SUB #7776, SIZE ;BACK PAST LOADER

```


;ROUTINE TO LOAD EXCESS CORE WITH WORSE CASE MEMORY TEST

666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700

002004 013700 001312
002010 012701 017000
002014 020001
002016 103410
002020 012702 002054
002024 012221
002026 022702 002124
002032 001374
002034 020100
002036 101770
002040 012721 000207
002044 005021
002046 005021
002050 000207

002052 151456

CORTST: MOV SIZE, R0 ; GET LAST FREE CORE ADDRESS
MOV #BUFFER, R1 ; GET END OF PROGRAM
CMP R0, R1 ; TEST FOR EQUAL
BLO XMRTS ; BRANCH IF NO ROOM
XMLOP1: MOV #MEMTST, R2 ; MOVE CODE BETWEEN
XMLOP2: MOV (R2)+, (R1)+ ; MEMTST AND MEMEND TILL
CMP #MEMEND, R2 ; CORE IS FULL
BNE XMLOP2
CMP R1, R0 ; TEST FOR MORE ROOM
BLOS XMLOP1
XMRTS: MOV #207, (R1)+ ; SETUP RTS PC
CLR (R1)+
CLR (R1)+
RTS PC

ROTVAL: 151456
.DSABL AMA

; THIS IS THE BACKGROUND TASK WHICH WILL BE LOADED THRU
; THE REMAINDER OF MEMORY

002054 000277
002056 012727 123456
002062 123456
002064 106067 177773
002070 103401
002072 000000
002074 102001
002076 000000
002100 022767 151456 177754
002106 001401
002110 000000
002112 026737 177744 002052
002120 001401
002122 000000
002124 000000
002126 000000

MEMTST: SCC ; SET CARRY BIT
MOV #123456, (PC)+ ; MEMDAT CONTAINS
MEMDAT: 123456
RORB MEMDAT+1 ; ROTATE LEFT BYTE OF MEMDAT
BCS .+4
HALT ; C BIT WAS NOT SET
BVC .+4
HALT ; V BIT WAS SET
CMP #151456, MEMDAT ; CHECK HERE FOR CORRECT ROTATE
BEQ .+4
HALT ; ROTATE FAILED
CMP MEMDAT, #ROTVAL ; CHECK AGAIN REFERENCE LOW MEMORY
BEQ .+4
HALT ; REF. TO LOW MEMORY FAILED
MEMEND: D
D
.ENABL AMA


```

668
669 ;INTERRUPT SERVICE FOR THE GT STOP INTERRUPT
670
671 002130 005777 177014 GTSTOP: TST @GTSR ;TEST STOP
672 002134 100403 BMI IS
673 002136 000000 HALT ;ERROR, STOP INTERRUPT BUT NO STOP FLAG
674 002140 000137 001022 JMP STARTB ;RESTART TEST
675
676 002144 005337 001314 IS: DEC GTDLYD ;DECREMENT DELAY
677 002150 001014 BNE GTST1 ;BRANCH IF NOT
678 002152 012737 000040 001314 MOV #40,GTDLYD ;RESET DELAY
679 002160 005237 003074 INC GRPINC ;UPDATE GRAPH INCREMENT
680 002164 022737 174110 003074 CMP #STATSB!INCR+10,GRPINC ;TEST FOR INCREMENT
681 002172 001003 BNE GTST1 ;BRANCH IF NOT
682 002174 012737 174100 003074 MOV #STATSB!INCR,GRPINC ;RESET GRAPH INCREMENT
683 002202 012737 005752 005722 GTST1: MOV #FILEDC,FILEDA
684 002210 012777 000001 176730 MOV #1,@GTPC ;RESUME THE DISPLAY
685 002216 000002 RTI ;EXIT
686
687 002220 012737 005724 005722 GTLPEN: MOV #FILEOB,FILEOA
688 002226 012777 000001 176712 MOV #1,@GTPC ;RESUME THE DISPLAY
689 002234 000002 RTI
690
691 002236 000000 GTSHIF: HALT
692 002240 000137 001022 JMP STARTB ;GT-40 SHIFT-OUT/TIME-OUT ERROR

```



```

693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738

; INTERRUPT SERVICE FOR THE DL PRINTER
DLOUT: TSTB  DLLOCSR          ; TEST FOR DONE
        BMI  .+10
        HALT                    ; ERROR, PRINTER INTERRUPT BUT NO PRINTER FLAG
        JMP  STARTB            ; RESTART TEST

        MOV  R4, -(SP)

DLOUTA: MOV  PPNT, R4          ; LOAD R4 WITH BYTE POINTER
        MOVB BUFF1(R4), PUNCHR ; LOAD A CHARACTER TO BE OUTPUTTED
        INC  PPNT              ; UPDATE CHARACTER POINTER
        CMP  #40, PPNT        ; TEST FOR END
        BNE  DLOUTB
        CLR  PPNT              ; CLEAR PUNCH POINTER

DLOUTB: MOVB PUNCHR, DLLODBR  ; OUTPUT A CHARACTER
        MOV  (SP)+, R4        ; RESTORE R4
        RTI                    ; EXIT

; INTERRUPT SERVICE FOR THE DL READER
DLIN:  TSTB  DLICSR          ; TEST FOR DONE
        BMI  .+10
        HALT                    ; NOT DL INPUT FLAG
        JMP  STARTB            ; RESTART TEST

        MOV  R4, -(SP)        ; SAVE R4
        MOV  RPNT, R4
        MOVB DLIDBR, REDCHR    ; READ A CHARACTER
        BIC  #177600, REDCHR   ; MASK CHARACTER
        MOVB REDCHR, BUFF2(R4) ; PUT CHARACTER INTO THE BUFFER
        INC  RPNT              ; UPDATE READ POINTER
        CMP  #40, RPNT        ; TEST FOR END
        BNE  DLINB
        CLR  RPNT
DLINB: MOV  RPNT, R4
        MOVB #177, BUFF2(R4)   ; ADD CURSOR
        MOV  (SP)+, R4        ; RESTORE R4
        RTI                    ; EXIT

PPNT:  0
RPNT:  0
PUNCHR: 240
REDCHR: 240

```



```

739
740
741
742
743
744 002440 105777 176546      KBIN:  TSTB  @TKS      ;TEST FOR DONE
745 002444 100403           BMI    .+10
746 002446 000000           HALT
747 002450 000137 001022      JMP    STARTB      ;NOT KRB INPUT FLAG
748                                     ;RESTART
749 002454 010346           MOV    R3,-(SP)    ;SAVE R3
750 002456 010446           MOV    R4,-(SP)    ;SAVE R4
751 002460 013704 002624      MOV    KPNT,R4
752 002464 117737 176524 002626  MOVB  @TKB,KBCHR    ;READ CHARACTER
753 002472 042737 177600 002626  BIC   #177600,KBCHR ;MASK
754 002500 113764 002626 005612  MOVB  KBCHR,BUFF3(4) ;SAVE THE CHAR
755 002506 005237 002624           INC   KPNT         ;UPDATE POINTER
756 002512 022737 000050 002624  CMP   #40.,KPNT    ;TEST FOR END
757 002520 001002           BNE   1$
758 002522 005037 002624      CLR   KPNT        ;CLEAR POINTER
759 002526 013704 002624      1$:  MOV   KPNT,R4
760 002532 112764 000177 005612  MOVB  #177,BUFF3(R4) ;ADD CURSOR
761
762                                     ;UPDATE OCTAL READOUT
763
764 002540 013703 002626      MOV   KBCHR,R3    ;GET CHAR
765 002544 004737 002610      JSR   PC,10$      ;LOAD BITS
766 002550 110437 005701      MOVB  R4,OCTA+2   ;SAVE BITS
767 002554 004737 002602      JSR   PC,11$      ;MOVE BITS
768 002560 110437 005700      MOVB  R4,OCTA+1   ;SAVE BITS
769 002564 004737 002602      JSR   PC,11$      ;MOVE BITS
770 002570 110437 005677      MOVB  R4,OCTA     ;SAVE BITS
771 002574 012604           MOV   (SP)+,R4    ;RESTORE R4
772 002576 012603           MOV   (SP)+,R3    ;RESTORE R3
773 002600 000002           RTI               ;EXIT
774
775 002602 006003      11$:  ROR   R3
776 002604 006003      ROR   R3
777 002606 006003      ROR   R3
778 002610 010304      10$:  MOV   R3,R4
779 002612 042704 177770  BIC   #177770,R4   ;LOAD R4
780 002616 062704 000060  ADD   #60,R4       ;MASK BITS
781 002622 000207           RTS              ;MAKE A NUMBER
782
783 002624 000000      KPNT:  0
784 002626 000240      KBCHR: 240

```



```

785 ;PART 1 OF THE BACKGROUND TASK
786
787 002630 012737 001000 003010 OVER: MOV #1000,PCOUNT ;SET UP EXECUTION COUNT
788
789 : COMPARE THE ROM DATA TO THE IMAGE DATA
790 : R0=WORD NUMBER
791 : R1=GOOD DATA
792 : R2=GOOD DATA
793 : R3=BAD ADDRESS
794 : R4=BAD DATA
795
796 002636 012700 000000 BACK: MOV #0,%0 ;SETUP INITIAL WORD NUMBER
797 002642 013701 001020 MOV IMAGE,%1 ;SET UP BUFFER
798 002646 001415 BEQ TMEM ;NO ROM SELECTED
799 002650 013703 001014 MOV ROMADD,%3 ;SET UP ROM ADDRESS
800 002654 011102 BACK1: MOV (%1),%2 ;READ A IMAGE WORD
801 002656 011304 MOV (%3),%4 ;READ A ROM WORD
802 002660 020204 CMP %2,%4 ;TEST FOR EQUAL
803 002662 001402 BEQ BACK2 ;BRANCH IF OK
804 002664 000000 HALT ;ERROR ROM VALUE FAILED TO EQUAL THE
805 002666 000772 BR BACK1 ; THE EXPECTED
806
807 002670 022123 BACK2: CMP (%1)+,(%3)+ ;BUMP BOTH REGISTERS
808 002672 005200 INC %0 ;UPDATE WORD COUNTER
809 002674 023700 001016 CMP WORDS,%0 ;TEST FOR LAST WORD
810 002700 001365 BNE BACK1 ;BRANCK IF NOT LAST
811
812 ;PART 2 OF THE BACKGROUND TASK
813 ; EXECUTE WORSE CASE NOISE TEST THRU MEMORY
814
815 002702 004737 017000 TMEM: JSR PC,BUFFER ;EXECUTE NOISE TEST
816 002706 005337 003010 DEC PCOUNT ;DONE PASS ?
817 002712 001351 BNE BACK ;NO
818 002714 012777 000001 176226 MOV #1,%GTSR ;YES RING THE BELL
819 002722 012777 000207 176270 MOV #207,%TPB ;RING THE BELL
820 002730 105777 176262 1$: TSTB %TPS
821 002734 100375 BPL 1$
822 002736 012777 000207 176254 MOV #207,%TPB
823 002744 105777 176246 2$: TSTB %TPS
824 002750 100375 BPL 2$
825 002752 005737 000042 TST %42 ;TEST LOC. 42
826 002756 001724 BEQ OVER ;BR IF =0
827 002760 000005 RESET
828 002762 000005 RESET
829 002764 000005 RESET
830 002766 013700 000042 MOV %42,R0 ;READ VALUE
831 002772 004710 LOGICAL: JSR PC,(0)
832 002774 000240 NOP
833 002776 000240 NOP
834 003000 000240 NOP
835 003002 000240 NOP
836 003004 000137 001022 JMP STARTB
837
838 003010 000000 PCOUNT: 0

```



```

0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095

```

```

003012 114140
003014 000000
003016 001377
003020 174300

003022 113004
003024 041777
003026 000000
003030 110005
003032 040000
003034 021377
003036 110006
003040 061777
003042 000000
003044 110007
003046 040000
003050 001377

003052 114004
003054 000400
003056 000200
003060 110000
003062 041200
003064 000000
003066 114000
003070 000440
003072 000200
003074 174104

003076 124000
003100 000200
003102 000205
003104 000212
003106 000217
003110 000224
003112 000231
003114 000236
003116 000243
003120 000247
003122 000253
003124 000257
003126 000262
003130 000265
003132 000270
003134 000272
003136 000274
003140 000276
003142 000277
003144 000277

```

```

FILE00: POINT!LPON
0
MAXY
STATSB!LPLITE

;LINE THE EDGES OF THE SCREEN

LONGV!INT4!LINED ;TOP LINE
INTX!MAXX
0
LONGV!LINE1 ;RIGHT LINE
INTX
MINUSX!MAXY
LONGV!LINE2 ;BOTTOM LINE
INTX!MINUSX!MAXX
0
LONGV!LINE3 ;LEFT LINE
INTX
MAXY

;SETUP THE X SINEWAVE

POINT!LINED
400
200
LONGV
INTX+1200 ;DRAW X AXIS
0
POINT
440
200
GRPINC: STATSB!INCR+4 ;GRAPHPLOT THE X SINEWAVE

GRAPHY
0200
0205
0212
0217
0224
0231
0236
0243
0247
0253
0257
0262
0265
0270
0272
0274
0276
0277
0277

```


896	003146	000277	0277
897	003150	000277	0277
898	003152	000276	0276
899	003154	000275	0275
900	003156	000274	0274
901	003160	000272	0272
902	003162	000267	0267
903	003164	000264	0264
904	003166	000261	0261
905	003170	000256	0256
906	003172	000252	0252
907	003174	000246	0246
908	003176	000241	0241
909	003200	000235	0235
910	003202	000230	0230
911	003204	000223	0223
912	003206	000216	0216
913	003210	000211	0211
914	003212	000203	0203
915	003214	000176	0176
916	003216	000171	0171
917	003220	000163	0163
918	003222	000156	0156
919	003224	000151	0151
920	003226	000144	0144
921	003230	000137	0137
922	003232	000133	0133
923	003234	000127	0127
924	003236	000123	0123
925	003240	000117	0117
926	003242	000114	0114
927	003244	000111	0111
928	003246	000106	0106
929	003250	000104	0104
930	003252	000102	0102
931	003254	000101	0101
932	003256	000100	0100
933	003260	000100	0100
934	003262	000100	0100
935	003264	000100	0100
936	003266	000101	0101
937	003270	000102	0102
938	003272	000104	0104
939	003274	000106	0106
940	003276	000111	0111
941	003300	000113	0113
942	003302	000117	0117
943	003304	000122	0122
944	003306	000126	0126
945	003310	000132	0132
946	003312	000137	0137
947	003314	000144	0144
948	003316	000151	0151
949	003320	000156	0156
950	003322	000163	0163
951	003324	000170	0170

952	003326	000175	0175
953	003330	000203	0203
954	003332	000210	0210
955	003334	000215	0215
956	003336	000222	0222
957	003340	000227	0227
958	003342	000234	0234
959	003344	000241	0241
960	003346	000245	0245
961	003350	000252	0252
962	003352	000255	0255
963	003354	000261	0261
964	003356	000264	0264
965	003360	000267	0267
966	003362	000271	0271
967	003364	000274	0274
968	003366	000275	0275
969	003370	000276	0276
970	003372	000277	0277
971	003374	000277	0277
972	003376	000277	0277
973	003400	000277	0277
974	003402	000276	0276
975	003404	000274	0274
976	003406	000273	0273
977	003410	000270	0270
978	003412	000266	0266
979	003414	000263	0263
980	003416	000257	0257
981	003420	000254	0254
982	003422	000247	0247
983	003424	000243	0243
984	003426	000237	0237
985	003430	000232	0232
986	003432	000225	0225
987	003434	000220	0220
988	003436	000213	0213
989	003440	000205	0205
990	003442	000200	0200
991	003444	000173	0173
992	003446	000165	0165
993	003450	000160	0160
994	003452	000153	0153
995	003454	000146	0146
996	003456	000141	0141
997	003460	000135	0135
998	003462	000130	0130
999	003464	000124	0124
1000	003466	000120	0120
1001	003470	000115	0115
1002	003472	000112	0112
1003	003474	000107	0107
1004	003476	000105	0105
1005	003500	000103	0103
1006	003502	000101	0101
1007	003504	000100	0100

1008	003506	000100	0100
1009	003510	000100	0100
1010	003512	000100	0100
1011	003514	000100	0100
1012	003516	000102	0102
1013	003520	000103	0103
1014	003522	000105	0105
1015	003524	000107	0107
1016	003526	000112	0112
1017	003530	000115	0115
1018	003532	000121	0121
1019	003534	000125	0125
1020	003536	000131	0131
1021	003540	000135	0135
1022	003542	000142	0142
1023	003544	000147	0147
1024	003546	000154	0154
1025	003550	000161	0161
1026	003552	000166	0166
1027	003554	000173	0173

;SETUP THE Y SINEWAVE

1031	003556	114000	POINT
1032	003560	000200	200
1033	003562	000040	40
1034	003564	110000	LONGV
1035	003566	040000	INTX
1036	003570	001200	1200
1037	003572	114000	POINT
1038	003574	000200	200
1039	003576	000100	100
1040	003600	120000	GRAPHX

;DRAW Y AXIS

;GRAPHPLOT THE Y SINEWAVE

1042	003602	000200	0200
1043	003604	000205	0205
1044	003606	000212	0212
1045	003610	000217	0217
1046	003612	000224	0224
1047	003614	000231	0231
1048	003616	000236	0236
1049	003620	000243	0243
1050	003622	000247	0247
1051	003624	000253	0253
1052	003626	000257	0257
1053	003630	000262	0262
1054	003632	000265	0265
1055	003634	000270	0270
1056	003636	000272	0272
1057	003640	000274	0274
1058	003642	000276	0276
1059	003644	000277	0277
1060	003646	000277	0277
1061	003650	000277	0277
1062	003652	000277	0277
1063	003654	000276	0276

K02

GT40 QUICK VERIFY MAINDEC-11-DDGTE-C
DDGTEC.P11 05-NOV-76 10:20

MACY11 27(1006) 05-NOV-76 12:10 PAGE 24

1064	003656	000275	0275
1065	003660	000274	0274
1066	003662	000272	0272
1067	003664	000267	0267
1068	003666	000264	0264
1069	003670	000261	0261
1070	003672	000256	0256
1071	003674	000252	0252
1072	003676	000246	0246
1073	003700	000241	0241
1074	003702	000235	0235
1075	003704	000230	0230
1076	003706	000223	0223
1077	003710	000216	0216
1078	003712	000211	0211
1079	003714	000203	0203
1080	003716	000176	0176
1081	003720	000171	0171
1082	003722	000163	0163
1083	003724	000156	0156
1084	003726	000151	0151
1085	003730	000144	0144
1086	003732	000137	0137
1087	003734	000133	0133
1088	003736	000127	0127
1089	003740	000123	0123
1090	003742	000117	0117
1091	003744	000114	0114
1092	003746	000111	0111
1093	003750	000106	0106
1094	003752	000104	0104
1095	003754	000102	0102
1096	003756	000101	0101
1097	003760	000100	0100
1098	003762	000100	0100
1099	003764	000100	0100
1100	003766	000100	0100
1101	003770	000101	0101
1102	003772	000102	0102
1103	003774	000104	0104
1104	003776	000106	0106
1105	004000	000111	0111
1106	004002	000113	0113
1107	004004	000117	0117
1108	004006	000122	0122
1109	004010	000126	0126
1110	004012	000132	0132
1111	004014	000137	0137
1112	004016	000144	0144
1113	004020	000151	0151
1114	004022	000156	0156
1115	004024	000163	0163
1116	004026	000170	0170
1117	004030	000175	0175
1118	004032	000203	0203
1119	004034	000210	0210

1120	004036	000215	0215
1121	004040	000222	0222
1122	004042	000227	0227
1123	004044	000234	0234
1124	004046	000241	0241
1125	004050	000245	0245
1126	004052	000252	0252
1127	004054	000255	0255
1128	004056	000261	0261
1129	004060	000264	0264
1130	004062	000267	0267
1131	004064	000271	0271
1132	004066	000274	0274
1133	004070	000275	0275
1134	004072	000276	0276
1135	004074	000277	0277
1136	004076	000277	0277
1137	004100	000277	0277
1138	004102	000277	0277
1139	004104	000276	0276
1140	004106	000274	0274
1141	004110	000273	0273
1142	004112	000270	0270
1143	004114	000266	0266
1144	004116	000263	0263
1145	004120	000257	0257
1146	004122	000254	0254
1147	004124	000247	0247
1148	004126	000243	0243
1149	004130	000237	0237
1150	004132	000232	0232
1151	004134	000225	0225
1152	004136	000220	0220
1153	004140	000213	0213
1154	004142	000205	0205
1155	004144	000200	0200
1156	004146	000173	0173
1157	004150	000165	0165
1158	004152	000160	0160
1159	004154	000153	0153
1160	004156	000146	0146
1161	004160	000141	0141
1162	004162	000135	0135
1163	004164	000130	0130
1164	004166	000124	0124
1165	004170	000120	0120
1166	004172	000115	0115
1167	004174	000112	0112
1168	004176	000107	0107
1169	004200	000105	0105
1170	004202	000103	0103
1171	004204	000101	0101
1172	004206	000100	0100
1173	004210	000100	0100
1174	004212	000100	0100
1175	004214	000100	0100

M02

GT40 QUICK VERIFY MAINDEC-11-DDGTE-C
DDGTEC.P11 05-NOV-76 10:20

MACY11 27(1006) 05-NOV-76 12:10 PAGE 26

1176	004216	000100	0100
1177	004220	000102	0102
1178	004222	000103	0103
1179	004224	000105	0105
1180	004226	000107	0107
1181	004230	000112	0112
1182	004232	000115	0115
1183	004234	000121	0121
1184	004236	000125	0125
1185	004240	000131	0131
1186	004242	000135	0135
1187	004244	000142	0142
1188	004246	000147	0147
1189	004250	000154	0154
1190	004252	000161	0161
1191	004254	000166	0166
1192	004256	000173	0173
1193			

;SETUP TO DISPLAY THE OCTAGONS

1194			
1195			
1196			
1197	004260	114000	POINT
1198	004262	001434	1434
1199	004264	000724	724
1200	004266	130030	RELATV:BLKON
1201	004270	041600	INTX+1600
1202	004272	041607	INTX+1600+7
1203	004274	040007	INTX+7
1204	004276	061607	INTX!MINUSX+1600+7
1205	004300	061600	INTX!MINUSX+1600
1206	004302	061707	INTX!MINUSX+1600+MINSUY+7
1207	004304	040107	INTX+MINSUY+7
1208	004306	041707	INTX+1600+MINSUY+7
1209	004310	114000	POINT
1210	004312	001430	1430
1211	004314	000710	710
1212	004316	130020	RELATV:BLKOFF
1213	004320	043600	INTX+3600
1214	004322	043617	INTX+3600+17
1215	004324	040017	INTX+17
1216	004326	063617	INTX!MINUSX+3600+17
1217	004330	063600	INTX!MINUSX+3600
1218	004332	063717	INTX!MINUSX+3600+MINSUY+17
1219	004334	040117	INTX+MINSUY+17
1220	004336	043717	INTX+3600+MINSUY+17
1221	004340	114000	POINT
1222	004342	001420	1420
1223	004344	000660	660
1224	004346	104030	SHORTV:BLKON
1225	004350	047600	INTX+7600
1226	004352	047637	INTX+7600+37
1227	004354	040037	INTX+37
1228	004356	067637	INTX!MINUSX+7600+37
1229	004360	067600	INTX!MINUSX+7600
1230	004362	067737	INTX!MINUSX+7600+MINSUY+37
1231	004364	040137	INTX+MINSUY+37
1232	004366	047737	INTX+7600+MINSUY+37
1233	004370	114000	POINT
1234	004372	001400	1400
1235	004374	000600	600
1236	004376	104020	SHORTV:BLKOFF
1237	004400	057600	INTX+17600
1238	004402	057677	INTX+17600+77
1239	004404	040077	INTX+77
1240	004406	077677	INTX!MINUSX+17600+77
1241	004410	077600	INTX!MINUSX+17600
1242	004412	077777	INTX!MINUSX+17600+MINSUY+77
1243	004414	040177	INTX+MINSUY+77
1244	004416	057777	INTX+17600+MINSUY+77
1245	004420	114030	POINT:BLKON
1246	004422	001360	1360
1247	004424	000520	520
1248	004426	110000	LONGV
1249	004430	040137	INTX+137

;OCTOGON BY LENGTH OF 137

1250	004432	000000		
1251	004434	040137		
1252	004436	000137		
1253	004440	040000		
1254	004442	000137		
1255	004444	060137		
1256	004446	000137		
1257	004450	060137		
1258	004452	000000		
1259	004454	060137		
1260	004456	020137		
1261	004460	040000		
1262	004462	020137		
1263	004464	040137		
1264	004466	020137		
1265	004470	114120		
1266	004472	001340		
1267	004474	000440		
1268	004476	110000		
1269	004500	040177		
1270	004502	000000		
1271	004504	040177		
1272	004506	000177		
1273	004510	040000		
1274	004512	000177		
1275	004514	060177		
1276	004516	000177		
1277	004520	060177		
1278	004522	000000		
1279	004524	060177		
1280	004526	020177		
1281	004530	040000		
1282	004532	020177		
1283	004534	040177		
1284	004536	020177		
1285				
1286	004540	114140		
1287	004542	000100		
1288	004544	001277		
1289	004546	164000		
1290	004550	170040		
1291	004552	100000		
1292	004554	040500	041502	042504
1293	004556	043506	044510	045512
1294	004570	046514	047516	050520
1295	004576	051522	052524	053526
1296	004604	054530	132	
1297	004607	040	021041	022043
1298	004614	023045	024047	025051
1299	004622	026053	027055	030057
1300	004630	031061	032063	033065
1301	004636	034067	035071	036053
1302	004644	037075	000077	
1303				
1304	004650	164000		
1305	004652	170060		

```

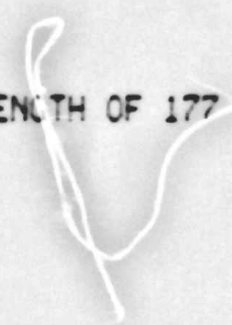
0
INTX+137
137
INTX
137
INTX!MINUSX+137
137
INTX!MINUSX+137
0
INTX!MINUSX+137
MINUSX+137
INTX
MINUSX+137
INTX+137
MINUSX+137
POINT!BLKOFF!LPOFF
1340
440
LONGV
INTX+177
0
INTX+177
177
INTX
177
INTX!MINUSX+177
177
INTX!MINUSX+177
0
INTX!MINUSX+177
MINUSX+177
INTX
MINUSX+177
INTX+177
MINUSX+177
POINT!LPON
100
MAXY-100
DNOP
STATSA!ITALO
CHAR
.ASCII ' @ABCDEFGHIJKLMNPOQRSTUVWXYZ'

.ASCIIZ @ !"#%&'()*+,-./0123456789:;<=>?@

.EVEN
DNOP
STATSA!ITALI

```

;OCTOGON BY LENGTH OF 177



1306	004654	114000		
1307	004656	000100		
1308	004660	001247		
1309	004662	100000		
1310	004664	040500	041502	042504
1311	004672	043506	044510	045512
1312	004700	046514	047516	050520
1313	004706	051522	052524	053526
1314	004714	054530	132	
1315	004717	040	021041	022043
1316	004724	023045	024047	025051
1317	004732	026053	027055	030057
1318	004740	031061	032063	033065
1319	004746	034067	035071	036053
1320	004754	037075	000077	
1321				
1322	004760	170040		
1323	004762	114000		
1324	004764	000220		
1325	004766	001177		
1326	004770	100000		
1327	004772	140	141	142
1328	004775	143	144	145
1329	005000	146	147	
1330	005002	150	151	152
1331	005005	153	154	155
1332	005010	156	157	
1333	005012	160	161	162
1334	005015	163	164	165
1335	005020	166	167	
1336	005022	170	171	172
1337	005025	173	174	175
1338	005030	176	177	
1339				
1340	005032	170060		
1341	005034	114000		
1342	005036	000220		
1343	005040	001147		
1344	005042	100000		
1345	005044	140	141	142
1346	005047	143	144	145
1347	005052	146	147	
1348	005054	150	151	152
1349	005057	153	154	155
1350	005062	156	157	
1351	005064	160	161	162
1352	005067	163	164	165
1353	005072	166	167	
1354	005074	170	171	172
1355	005077	173	174	175
1356	005102	176	177	
1357				
1358	005104	170040		
1359	005106	114000		
1360	005110	000220		
1361	005112	001077		

```

POINT
100
MAXY-130
CHAR
.ASCII ' @ABCDEFGHIJKLMNPOQRSTUVWXYZ'

.ASCIIZ @ !"#%&'()*+,-./0123456789:+(=) ?@

.EVEN
STATSA:ITALO
POINT
220
MAXY-200
CHAR
.BYTE 140,141,142,143,144,145,146,147

.BYTE 150,151,152,153,154,155,156,157

.BYTE 160,161,162,163,164,165,166,167

.BYTE 170,171,172,173,174,175,176,177

.EVEN
STATSA:ITALI
POINT
220
MAXY-230
CHAR
.BYTE 140,141,142,143,144,145,146,147

.BYTE 150,151,152,153,154,155,156,157

.BYTE 160,161,162,163,164,165,166,167

.BYTE 170,171,172,173,174,175,176,177

.EVEN
STATSA:ITALO
POINT
220
MAXY-300

```


1362	005114	100000		
1363	005116	016	000	001
1364	005121	002	003	004
1365	005124	005	006	007
1366	005127	010	011	012
1367	005132	013	014	015
1368	005135	016		
1369	005136	020	021	022
1370	005141	023	024	025
1371	005144	026	027	030
1372	005147	031	032	033
1373	005152	034	035	036
1374	005155	037	017	000
1375				
1376	005160	170060		
1377	005162	114000		
1378	005164	000220		
1379	005166	001047		
1380	005170	100000		
1381	005172	016	000	001
1382	005175	002	003	004
1383	005200	005	006	007
1384	005203	010	011	012
1385	005206	013	014	015
1386	005211	016		
1387	005212	020	021	022
1388	005215	023	024	025
1389	005220	026	027	030
1390	005223	031	032	033
1391	005226	034	035	036
1392	005231	037	017	000
1393				
1394	005234	170040		
1395				
1396				
1397	005236	114000		
1398	005240	000340		
1399	005242	001000		
1400	005244	113604		
1401	005246	040400		
1402	005250	000000		
1403	005252	114000		
1404	005254	000340		
1405	005256	000740		
1406	005260	113400		
1407	005262	040400		
1408	005264	000000		
1409	005266	114000		
1410	005270	000340		
1411	005272	000700		
1412	005274	113200		
1413	005276	040400		
1414	005300	000000		
1415	005302	114000		
1416	005304	000340		
1417	005306	000640		

```

CHAR
.BYTE 16,0,1,2,3,4,5,6,7,10,11,12,13,14,15,16

.EVEN
STATSA:ITAL1
POINT
220
MAXY-330
CHAR
.BYTE 16,0,1,2,3,4,5,6,7,10,11,12,13,14,15,16

.EVEN
STATSA:ITAL0
;SETUP INTENSITY LEVEL TEST
POINT
340
1000
LONGV!INT7!LINE0
INTX+400
0
POINT
340
740
LONGV!INT6
INTX+400
0
POINT
340
700
LONGV!INT5
INTX+400
0
POINT
340
640

```

1418	005310	113000			LONGV!INT4
1419	005312	040400			INTX+400
1420	005314	000000			0
1421	005316	114000			POINT
1422	005320	000340			340
1423	005322	000600			600
1424	005324	112600			LONGV!INT3
1425	005326	040400			INTX+400
1426	005330	000000			0
1427	005332	114000			POINT
1428	005334	000340			340
1429	005336	000540			540
1430	005340	112400			LONGV!INT2
1431	005342	040400			INTX+400
1432	005344	000000			0
1433	005346	114000			POINT
1434	005350	000340			340
1435	005352	000500			500
1436	005354	112200			LONGV!INT1
1437	005356	040400			INTX+400
1438	005360	000000			0
1439	005362	114000			POINT
1440	005364	000340			340
1441	005366	000440			440
1442	005370	112000			LONGV!INT0
1443	005372	040400			INTX+400
1444	005374	000000			0
1445					
1446					
1447					
1448	005376	117000			POINT!INT4
1449	005400	000400			400
1450	005402	000020			20
1451	005404	100000			CHAR
1452	005406	047503	020115	052517	.ASCII /COM OUTPUT /
1453	005414	050124	052125	020040	
1454	005422	042504	043503	040522	BUFF1: .ASCII /DECGRAPHIC-11 DISPLAY TERMINAL GT40 VR14/
1455	005430	044120	041511	030455	
1456	005436	020061	044504	050123	
1457	005444	040514	020131	042524	
1458	005452	046522	047111	046101	
1459	005460	043440	032124	020060	
1460	005466	051126	032061		
1461	005472	114000			POINT
1462	005474	000400			400
1463	005476	000320			320
1464	005500	100000			CHAR
1465	005502	047503	027115	044440	.ASCII /COM. INPUT /
1466	005510	050116	052125	020040	
1467	005516	000	000	000	BUFF2: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1468	005521	000	000	000	
1469	005524	000	000	000	
1470	005527	000	000	000	
1471	005532	000	000	000	
1472	005535	000	000	000	
1473	005540	000	000	000	

1474	005542	000	000	000	.BYTE	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1475	005545	000	000	000		
1476	005550	000	000	000		
1477	005553	000	000	000		
1478	005556	000	000	000		
1479	005561	000	000	000		
1480	005564	000	000	000		
1481	005566	114000			POINT	
1482	005570	000400			400	
1483	005572	000350			350	
1484	005574	100000			CHAR	
1485	005576	051113	027102	044440	.ASCII	/KRB. INPUT /
1486	005604	050116	052125	020040		
1487	005612	000	000	000	BUFF3: .BYTE	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1488	005615	000	000	000		
1489	005620	000	000	000		
1490	005623	000	000	000		
1491	005626	000	000	000		
1492	005631	000	000	000		
1493	005634	000	000	000		
1494	005636	000	000	000	.BYTE	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1495	005641	000	000	000		
1496	005644	000	000	000		
1497	005647	000	000	000		
1498	005652	000	000	000		
1499	005655	000	000	000		
1500	005660	000	000	000		
1501						
1502	005662	114000			POINT	
1503	005664	000400			400	
1504	005666	000400			400	
1505	005670	100000			CHAR	
1506	005672	041517	020124	000	.ASCIZ	/OCT /
1507	005677	000	000	000	OCTA: .BYTE	0,0,0
1508	005702	164000			DNOP	
1509	005704	164000			DNOP	
1510	005706	164000			DNOP	
1511	005710	164000			DNOP	
1512	005712	164000			DNOP	
1513	005714	164000			DNOP	
1514	005716	164000			DNOP	
1515	005720	160000			DJMP	
1516	005722	005752			FILEDA: FILEOC	
1517	005724	114000			FILEOB: POINT	
1518	005726	001000			1000	
1519	005730	000440			440	
1520	005732	100000			CHAR	
1521	005734	044514	044107	026524	.ASCIZ	/LIGHT-PEN HIT/
1522	005742	042520	020116	044510		
1523	005750	000124				
1524					FILEOC: .EVEN	
1525	005752	173400			DSTOP	
1526	005754	160000			DJMP	
1527	005756	003012			FILEOO	
1528						

1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700

```
*****  
: 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
```


16000
16001
16002
16003
16004
16005
16006
16007
16008
16009
16010
16011
16012
16013
16014
16015
16016
16017
16018
16019
16020
16021
16022
16023
16024
16025
16026
16027
16028
16029
16030
16031
16032
16033
16034

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.
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.

MAJOR SYSTEM DEFINITIONS

```

1635
1636
1637
1638
1639
1640
1641
1642      166000      ORIGIN=166000      ;ORIGIN OF THE BOOTSTRAP.
1643
1644      175610      DL11IS=175610      ;INPUT STATUS REGISTER OF DL11
1645      175612      DL11IB=DL11IS+2    ;INPUT CHARACTER FROM DL11
1646      175614      DL11OS=DL11IB+2    ;OUTPUT STATUS OF THE DL11
1647      175616      DL11OB=DL11OS+2    ;OUTPUT CHARACTER TO THE DL11
1648
1649      177560      KBDIS=177560      ;KEYBOARD INPUT STATUS
1650      177562      KBDIB=KBDIS+2    ;CURRENT CHARACTER FROM KEYBOARD.
1651
1652      172000      GT4OPC=172000      ;GT40 PROGRAM COUNTER.
1653      172002      GT4OSR=GT4OPC+2    ;GT40 STATUS REGISTER ADDRESS.
1654
1655
1656      001000      BSTART=1000      ;START OF THE DISPLAY BUFFER
1657      007000      BLIMIT=7000      ;APPROXIMATE END OF THE DISPLAY BUFFER.
1658      007776      TMPEND=7776      ;LOCATION OF INITIALIZATION STACK.
1659      000004      CORSTR=4      ;LOCATION OF PDP-11 TRAP VECTOR.
1660      007012      JMPADD=BLIMIT+10.  ;WHERE THE POINTER IS TO FIRST CHAR ON SCREEN
1661      000040      NUMLIN=32.      ;NUMBER OF LINES ON TEXT TO SHOW ON THE SCREEN
1662
1663      005015      CRLF=5015      ;CARRIAGE RETURN - LINE FEED
1664      000175      ALTMOD=175      ;THE "KEY" CHARACTER [I.E. ALTMODE].
1665
1666      160000      DISJMP=160000      ;THE GT40 JMP INSTRUCTION
1667      173000      DISTOP=173000      ;THE GT40 STOP DISPLAY INSTRUCTION.
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686

```

.SBTTL INITIALIZATION AND RESTART CODE


```

1687
1688
1689
1690          :          GT40 BOOTSTRAP CODE
1691          :          -----
1692
1693
1694
1695
1696          006000          .=6000
1697
1698          :          .=ORIGIN          ;DEFINE ORIGIN OF THE BOOTSTRAP.
1699
1700
1701
1702
1703
1704
1705          :          COLD INITIALIZATION CODE
1706          :          -----
1707
1708
1709
1710 006000 000005          START: RESET          ;RESET ALL HARDWARE NOW.
1711 006002 012737 000007 175610 MOV          #7,DL11IS          ;INITIALIZE DL-11 INPUT NOW.
1712 006010 012706 007776          MOV          #TMPEND,SP          ;ESTABLISH A GOOD TEMPORARY STACK
1713          ;POINTER FOR CORE SEARCH.
1714 006014 005237 175614          INC          DL110S          ;SET BREAK BIT
1715 006020 004337 166652          JSR          SCAN,OUTLIT!160000 ;FOR 2 CHARACTER TIMES
1716 006024 000000          .WORD 0          ;SEND TWO ZERO'S
1717
1718 006026 012703 000004          MOV          #CORSTR,SCAN          ;GET ADDRESS OF BAD CORE TRAP VECTOR.
1719 006032 012723 166042          MOV          #NOTHERE!160000,(SCAN)+ ;AND INSERT A POINTER TO US THERE.
1720
1721 006036 005023          ENDCOR: CLR          (SCAN)+          ;NOW CLEAR ALL OF MEMORY BEYOND THE POINTER,
1722 006040 000776          BR          ENDCOR          ;UNTIL WE RUN OUT OF MEMORY AND TRAP.
1723
1724
1725 006042 005743          NOTHER: TST          -(SCAN)          ;WHEN WE TRAP OUT, WE COME HERE.
1726          ;WE BACK UP POINTER TO GOOD CORE.
1727          ;NOTE THAT IF WE TRAP OUT AGAIN, IT
1728          ;IS STILL OK, BECAUSE WE WILL LOOP
1729          ;UNTIL WE GET A GOOD CORE ADDRESS.
1730 006044 010306          MOV          SCAN,SP          ;WHEN WE GET ONE, THAT IS LAST LOCATION
1731          ;IN THE MACHINE, AND HENCE OUR SP.
1732 006046 105737 175614          1$: TSTB          DL110S          ;SEE IF BREAK IS DONE
1733 006052 100375          BPL          1$          ;NO GO BACK
1734 006054 005037 175614          CLR          DL110S          ;CLEAR BREAK BIT
1735
1736
1737
1738
1739
1740
1741          :          RESTART INITIALIZATION CODE WHEN COMMUNICATIONS IS WORKING.
1742          :          -----

```

```

1743
1744
1745
1746 006060 052706 007776      RESTRT: BIS      #TMPEND,SP      ;FORCE THE SP TO LIMIT OF EXISTING CORE.
1747
1748
1749 006064 012703 006700      MOV      #BLIMIT-NUMLIN-NUMLIN,SCAN      ;NOW WE WILL FILL THE KEY AREAS OF THE
1750 006070 012702 000040      MOV      #NUMLIN,TABCNT      ;DISPLAY BUFFER WITH INITIAL CR-LF'S.
1751
1752 006074 012723 005015      SETLP1: MOV      #CRLF,(SCAN)+      ;INSERT A CRLF NOW.
1753 006100 005302      DEC      TABCNT      ;AND LOOP UNTIL DONE.
1754 006102 003374      BGT      SETLP1      ;THUS DISPLAY CORE IS ALMOST CORRECT.
1755
1756
1757 006104 012703 166432      MOV      #SETUP!160000,SCAN      ;NOW WE WILL INITIALIZE CORE FOR THE
1758      ;DISPLAY. PICK UP POINTER TO LIST.
1759
1760 006110 012302      SETLP2: MOV      (SCAN)+,TABCNT      ;GET NUMBER OF ITEMS TO INSERT.
1761 006112 001405      BEQ      SETDUN      ;IF ZERO, WE ARE DONE.
1762 006114 012301      MOV      (SCAN)+,POINTR      ;PICK UP FIRST CORE ADDRESS POINTER.
1763
1764 006116 012321      SETLP3: MOV      (SCAN)+,(POINTR)+      ;MOVE OVER A DATA ITEM NOW.
1765 006120 005302      DEC      TABCNT      ;ALL DONE?
1766 006122 003375      BGT      SETLP3      ;NOPE. MOVE OVER THE NEXT.
1767 006124 000771      BR       SETLP2      ;YES. GET NEXT MAJOR LIST TO INSERT.
1768
1769
1770 006126 012701 006776      SETDUN: MOV      #BLIMIT-2,POINTR      ;ESTABLISH THE BUFFER POINTER NOW.
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
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838

006132 004737 166564
006136 020027 000177
006142 002373
006144 020027 000040
006150 002020
006152 010003
006154 162703 000007
006160 020327 000007
006164 103362
006166 006303
006170 060307

006172 000426
006174 000406
006176 000411
006200 000437
006202 000420
006204 000424

006206 012702 177777

006212 004737 166350
006216 005202
006220 000744

006222 012700 000040
006226 004737 166350
006232 005202
006234 032702 000007
006240 001370
006242 000733

006244 111705
006246 000405
006250 005037 172002
006254 000726
006256 012705 000040

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

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

CR: MOV #-1,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 GT40SR
BR NXTCHR

FF: MOV #NUMLIN,COUNTR

VTOS (SCROLLING) PORTION OF THE BOOTSTRAP

;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 GT40SR
;AND LOOP BACK

;FORM FEED IS DONE BY INSERTING LF'S.

```

1839
1840 006262 012700 000012      FFLOOP: MOV      #12,CHAR      ;MAKE THE CHARACTER A LINEFEED.
1841 006266 004737 166304      JSR      PC,LFSUB!160000      ;DO A LINEFEED.
1842 006272 005305              DEC      COUNTR              ;DONE?
1843 006274 003372              BGT      FFLOOP              ;NOPE. KEEP SENDING THEM.
1844 006276 000715              BR       NXTCHR              ;YES. NOW RETURN. DO NOT FALL THROUGH.
1845
1846
1847 006300 012746 166132      LF:      MOV      #NXTCHR!160000,-(SP) ;RETURN TO NXTCHR AFTER PROCESSING
1848                                     ;THE LF BY FAKING A JSR.
1849
1850 006304 013703 007012      LFSUB:  MOV      JMPADD,SCAN      ;GET POINTER TO FIRST CHAR ON SCREEN
1851
1852 006310 122300              LFLOOP: CMPB     (SCAN)+,CHAR      ;AND LOOK FOR A LINEFEED.
1853 006312 001406              BEQ      LFOUND              ;GOT IT. SEARCH HAS ENDED.
1854 006314 020327 007000      CMP      SCAN,#BLIMIT        ;ARE WE AT END OF BUFFER?
1855 006320 103773              BLO      LFLOOP              ;NOPE. KEEP ON LOOKING.
1856 006322 012703 001000      MOV      #BSTART,SCAN        ;IF AT TOP, RESET TO BOTTOM OF BUFFER
1857 006326 000770              BR       LFLOOP              ;AND KEEP ON LOOKING.
1858
1859 006330 005203              LFOUND: INC      SCAN          ;WE'VE GOT THE LINE FEED. STOP SHOWING
1860 006332 042703 000001      BIC      #1,SCAN            ;FIRST LINE BY CHANGING THE "DISJMP"
1861 006336 010337 007012      MOV      SCAN,JMPADD         ;INSTRUCTION TO FIRST CHAR BEYOND LF.
1862 006342 004737 166350      JSR      PC,INSERT!160000      ;INSERT THE LF IN THE BUFFER.
1863 006346 005000              CLR      CHAR                ;AND THEN INSERT ONE NULL CHARACTER BECAUSE
1864                                     ;THE "DISJMP" ADDRESS MUST BE EVEN, AND
1865                                     ;THIS GUARANTEES WE WILL NOT LOSE A
1866                                     ;A GOOD DATA CHARACTER. WE FALL THROUGH
1867                                     ;TO INSERT THE NULL IN THE BUFFER.
1868
1869
1870 006350 110021              INSERT: MOVVB     CHAR,(POINTR)+    ;STICK IN THE CHARACTER NOW.
1871 006352 032701 000001      BIT      #1,POINTR          ;IS NEXT POSITION EVEN OR ODD?
1872 006356 001021              BNE      INSRTX              ;ODD. NO PROBLEMS. SPACE IS ALLOCATED.
1873 006360 020127 007000      CMP      POINTR,#BLIMIT      ;EVEN. ARE WE AT THE END OF THE BUFFER?
1874 006364 103410              BLO      INSRTL              ;NO. JUST MAKE ROOM FOR ANOTHER WORD.
1875 006366 010103              MOV      POINTR,SCAN         ;AT THE END. MOVE THE STUFF TO THE
1876 006370 012701 001000      MOV      #BSTART,POINTR      ;BEGINNING OF THE BUFFER.
1877 006374 004737 166406      JSR      PC,INSRTL!160000      ;CALL THE ROUTINE TO SAVE SPACE.
1878 006400 005023              CLR      (SCAN)+            ;AND CLEAR UP THE INSTRUCTIONS AT THE
1879 006402 005013              CLR      (SCAN)              ;END OF THE BUFFER.
1880 006404 000207              RTS      PC                  ;AND THEN RETURN.
1881
1882 006406 022121              INSRTL: CMP      (POINTR)+,(POINTR)+ ;BYPASS THE "DISJMP" BY ADDING 4 TO POINTR.
1883 006410 012711 166474      MOV      #HEADER!160000,(POINTR) ;NOW INSERT THE DISJMP INSTRUCTION TO OUR HEADER
1884 006414 012741 160000      MOV      #DISJMP,-(POINTR)    ;AND IT'S ADDRESS (PUT THEM IN BACKWARDS).
1885 006420 005041              CLR      -(POINTR)          ;MAKE AVAILABLE A NEW CHARACTER SPOT.
1886
1887 006422 000207              INSRTX: RTS      PC          ;FINALLY RETURN TO THE CALLER.
1888
1889
1890
1891
1892
1893 006424 012737 001000 172000  GTBUSE: MOV      #BSTART,GT40PC      ;ON A BUS ERROR, WE MERELY RESTART THE GT40 AT
1894

```


SCROLLING ROM BOOTSTRAP FOR THE GT40
DDGTEC.P11 05-NOV-76 10:20

MACY11 27(1006) 05-NOV-76 12:10 PAGE 40
VT05 SIMULATOR

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

1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942

INITIALIZATION TABLE FOR THE SCROLLER

```

:
:
SETUP: .WORD 2 ; INITIALIZE 2 WORDS.--ALSO RTI FROM ABOVE
        .WORD 330 ; STARTING AT LOCATION 330
        .WORD GTBASE!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 ; FOLLOWED BY A DISJMP TO OUR HEADER BLOC
        .WORD DISJMP,BSTART ; AND THEN A DISJMP TO THE START OF THE BUFFER
        .WORD DISJMP,BLIMIT-NUMLIN-NUMLIN ; 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: .WORD 103334 ; ENABL CHAR MODE, BLINKING
        .WORD 177 ; A BLINKING BOX-RUB OUT!
        .WORD 116124 ; GO TO POINT MODE
        .WORD 171340 ; LOAD STATUS REGISTER
        .WORD 0,1352 ; POINT TO UPPER LEFT
        .WORD 103324 ; BACK TO CHAR MODE
        .WORD DISJMP,JMPADD-2 ; AND TO THE CHANGING JMP INST.
.SBTTL COMMUNICATIONS AND MISC. SUPPORT ROUTINES
    
```

1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998

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.
006520	012737	000007	175610	MOV	#7,DL11IS	:REENABLE THE HOST TELECOMMUNICATIONS.
006536	042700	177600		BIC	#-200,CHAR	:MAKE CHARACTER JUST SEVEN BITS.
006542	001765			BEG	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			BEG	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	PRES'R: MOV	#DISTOP JMPADD-2	:YES. RESET. STOP DISPLAY BY INSERTING A "DISTOP
006624	000137	166060		JMP	RESTR!160000	:INSTRUCTION IN THE BUFFER, AND RESTART.

THE "GET A SIX BIT CHARACTER" ROUTINE

1999
 0000
 0001
 0002
 0003
 0004
 0005
 0006
 0007
 0008
 0009
 0010
 0011
 0012
 0013
 0014
 0015
 0016
 0017
 0018
 0019
 0020
 0021
 0022
 0023
 0024
 0025
 0026
 0027
 0028
 0029
 0030
 0031
 0032
 0033
 0034
 0035
 0036
 0037
 0038
 0039
 0040
 0041
 0042
 0043
 0044
 0045
 0046
 0047
 0048
 0049
 0050
 0051
 0052
 0053
 0054
 0055
 0056
 0057
 0058
 0059
 0060

006630 004737 166564
 006634 020027 000040
 006640 002517
 006642 020027 000137
 006646 003114

```

GETSIX: JSR PC,GETCHR!160000 ;GET A CHARACTER NOW.
        CMP CHAR,#40 ;IS IT A LEGAL PRINTING CHARACTER?
        BLT L.BAD ;NOPE. ABORT
        CMP CHAR,#137 ;IT'S BIG ENOUGH. IS IT TOO BIG?
        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
 006656 112337 175616
 006662 000203

```

OUTLIT: MOVB (SCAN)+,DL110B
        MOVB (SCAN)+,DL110B ;DOUBLE BUFFERED
        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
 006670 010046
 006672 005723
 006674 000163 166676

```

GETB: JSR PC,GETSIX!160000 ;GET A SIXBIT CHARACTER.
      MOV L.BYT,-(SP) ;SAVE IT ON THE STACK.
      TST (INDEX)+ ;UPDATE INDEX TO NEXT ITEM (ALL ARE *2)
      JMP GET8TB-2!160000(INDEX) ;AND DISPATCH ACCORDING TO THE INDEX.
  
```

006700 000404
 006702 000416
 006704 000432

```

GET8TB: BR GET81 ;INDEX=2: ASSEMBLE FIRST CHAR
        BR GET82 ;INDEX=4: ASSEMBLE SECOND CHAR
        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.
  
```

2055
2056 006712 004737 166630
2057 006716 010004
2058 006720 006300
2059 006722 006300
2060 006724 106300
2061 006726 106116
2062 006730 106300
2063 006732 106116
2064 006734 012600
2065 006736 000207
2066
2067
2068 006740 006300
2069 006742 006300
2070 006744 106300
2071 006746 106104
2072 006750 106300
2073 006752 106104
2074 006754 106300
2075 006756 106104
2076 006760 106300
2077 006762 106104
2078 006764 010400
2079 006766 012604
2080 006770 000207
2081
2082
2083 006772 006100
2084 006774 106100
2085 006776 006004
2086 007000 106000
2087 007002 006004
2088 007004 106000
2089 007006 005726
2090 007010 000207
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103

GET81: JSR PC,GETSIX!160000
MOV L,BYT,HOLD
ASL L,BYT
ASL L,BYT
ASLB L,BYT
ROLB (SP)
ASLB L,BYT
ROLB (SP)
MOV (SP)+,L,BYT
RTS PC

:GET ANOTHER 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.

GET82: ASL L,BYT
ASL L,BYT
ASLB L,BYT
ROLB HOLD
ASLB L,BYT
ROLB HOLD
ASLB L,BYT
ROLB HOLD
ASLB L,BYT
ROLB HOLD
MOV HOLD,L,BYT
MOV (SP)+,HOLD
RTS PC

: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.

GET83: ROL L,BYT
ROLB L,BYT
ROR HOLD
RORB L,BYT
ROR HOLD
RORB L,BYT
TST (SP)+
RTS PC

: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.

160	007140	000300		SWAB	L.BYT						
161	007142	052600		BIS	(SP)+,L.BYT						:AND THEN REASSEMBLE THE MESS.
162	007144	000207		RTS	PC						:WITH THE FEARSOME POWER OF THE 11.
											:AND RETURN TO THE CALLER.
167	007146	004737	167126	L.JMP:	JSR	PC,L.GWRD!160000					:ALL DONE WITH THE LOAD. ASSEMBLE
168	007152	010046			MOV	L.BYT, -(SP)					:THE STARTING ADDRESS NOW.
169	007154	004737	167114		JSR	PC,L.PTR!160000					:AND DON'T FORGET TO CHECKSUM IT.
170	007160	105705			TSTB	L.CKSM					
171	007162	001346			BNE	L.BAD					:A BAD CHECKSUM. ALL IS EVIL.
172											
173	007164	004337	166852		JSR	SCAN,OUTLIT!160000					:GOOD CHKSUM, INFORM HOST
174	007170	175	107		.BYTE	ALTMOD,'G					:WITH ALTMOD G
175											
176	007172	032716	000001		BIT	#1,(SP)					:DO WE WANT TO START EXECUTION?
177	007176	001401			BEG	L.JMP1					:YES. AWAY WE GO.
178											
179	007200	000000		L.HALT:	HALT						:IF NOT, HALT.
180											
181	007202	000136		L.JMP1:	JMP	@(SP)+					:IF GO, THEN GO ALREADY. WHEEEE!

.SBTTL THE SELF TEST

160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

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

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100

100000
104000
110000
114000
120000
124000
130000

002000
002200
002400
002600
003000
003200
003400
003600

000100
000140
000020
000030

000004
000005
000006
000007

160000
164000
170000
173400

000300
000200
000040
000060
000004

174000

000100
040000
001777
001377
020000
020000
017600
000077
000100

CHAR=100000
SHORTV=104000
LONGV=110000
POINT=114000
GRAPHX=120000
GRAPHY=124000
RELATV=130000

INT0=2000
INT1=2200
INT2=2400
INT3=2600
INT4=3000
INT5=3200
INT6=3400
INT7=3600

LPOFF=100
LPON=140
BLKOFF=20
BLKON=30

LINE0=4
LINE1=5
LINE2=6
LINE3=7

DJMP=160000
DNOP=164000
STATSA=170000
DSTOP=173400

LPLITE=300
LPDARK=200
ITAL0=40
ITAL1=60
SYNON=4

STATSB=174000

INCR=100
INTX=40000
MAXX=1777
MAXY=1377
MINUSX=20000
MINUSY=MINUSX
MAXSX=17600
MAXSY=77
MINSUY=100

;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

```

2254 007204 012737 167214 172000      MOV      #FILED!160000,GT40PC      ;START THE GT40
2255 007212 000001                      WAIT                               ;AND WAIT
2256                                     FILED: POINT!BLKOFF                ;POINT--INVISIBLE
2257 007214 114020                      0
2258 007216 000000                      MAXY
2259 007220 001377
2260
2261 007222 112004                      LONGV!INT0!LINE0                  ;DRAW TOP LINE
2262 007224 041777                      INTX!MAXX
2263 007226 000000                      0
2264
2265 007230 112405                      LONGV!INT2!LINE1
2266 007232 040000                      INTX                               ;DRAW LINE TO RIGHT
2267 007234 021377                      MINUSX!MAXY
2268
2269 007236 113006                      LONGV!INT4!LINE2
2270 007240 061777                      INTX!MINUSX!MAXX                 ;DRAW BOTTOM LINE
2271 007242 000000                      0
2272
2273 007244 113407                      LONGV!INT6!LINE3
2274 007246 040000                      INTX
2275 007250 001377                      MAXY                               ;DRAW LINE TO LEFT
2276
2277 007252 114000                      POINT
2278 007254 000400                      400
2279 007256 000500                      500
2280 007260 106200                      SHORTV!INT1
2281 007262 057677                      57677                             ;+X+Y
2282 007264 106600                      SHORTV!INT3
2283 007266 077677                      77677                             ;+X-Y
2284 007270 107200                      SHORTV!INT5
2285 007272 077777                      77777                             ;-X-Y
2286 007274 107600                      SHORTV!INT7
2287 007276 057777                      57777                             ;-X+Y
2288
2289 007300 114000                      POINT
2290 007302 001400                      1400
2291 007304 000500                      500
2292 007306 133030                      RELATV!INT4!BLKON
2293 007310 057677                      57677                             ;+X+Y
2294 007312 077677                      77677                             ;+X-Y
2295 007314 077777                      77777                             ;-X-Y
2296 007316 057777                      57777                             ;-X+Y
2297
2298 007320 114000                      POINT
2299 007322 000400                      400
2300 007324 000100                      100
2301 007326 174120                      STATSB!INCR+20                   ;TRY GRAPH MODES
2302 007330 114000                      POINT
2303 007332 001000                      1000
2304 007334 000200                      200
2305
2306 007336 120000                      GRAPHX
2307 007340 001010                      1010
2308 007342 001020                      1020
2309 007344 001030                      1030

```


SCROLLING ROM BOOTSTRAP FOR THE GT40
DDGTEC.P11 05-NOV-76 10:20

MACY11 27(1006) 05-NOV-76 12:10 PAGE 48
THE SELF TEST

2310	007346	001040	1040
2311	007350	001050	1050
2312			
2313	007352	114000	POINT
2314	007354	001000	1000
2315	007356	001200	1200
2316			
2317	007360	124000	GRAPHY
2318	007362	001020	1020
2319	007364	001030	1030
2320	007366	001040	1040
2321	007370	001050	1050
2322	007372	001060	1060
2323			
2324	007374	160000	DJMP
2325	007376	167214	FILEO!160000
2326			
2327			.SBTTL PAPER TAPE BOOT

```

2328
2329
2330
2331      177550
2332      177560
2333
2334
2335      007400  012701  160000
2336      007404  012702  000004
2337      007410  012703  167500
2338      007414  010712
2339      007416  012706  000024
2340      007422  014304
2341      007424  005714
2342      007426  100775
2343      007430  010712
2344      007432  012706  000024
2345      007436  010441
2346
2347      007440  040601
2348      007442  010111
2349      007444  011102
2350      007446  005214
2351      007450  105714
2352      007452  100376
2353      007454  116412  000002
2354      007460  005211
2355      007462  120227  000375
2356      007466  001366
2357      007470  105222
2358      007472  000142
2359
2360
2361
2362      007474  177560
2363      007476  177550
2364
2365

```

```

: PAPER TAPE BOOT
HSR=177550      ;HIGH SPEED READER ADDRESS
LSR=177560      ;LOW SPEED READER ADDRESS
:
:      .=ORIGIN+1400
PTBOOT: MOV      #160000,R1      ;SET MEMORY CHECK LIMITS
:TRAP ADDRESS IS LOC. 4
MOV      #4,R2
MOV      #DEV+4!160000,R3      ; POINTER TO DEVICE ADDRESSES
PC,@R2      ;PRESET TRAP ADDRESS IN LOC. 4
MOV      #24,SP      ;STACK SET UP AT SPECIAL ADDRESS
DEV1:  MOV      -(R3),R4      ;GET DEVICE ADDRESS
TST      @R4      ;CHECK AVAILABILITY OF DEVICE
BMI      DEV1      ;CHECK DEVICE FOR ERRORS
MOV      PC,@R2      ;RESET TRAP ADDRESS AT LOC. 4
MOV      #24,SP      ;SPECIAL ADDRESS USED AS MASK LATER
MOV      R4,-(R1)      ;DO MEM CHK:READER STATUS ADDRESS
:IS MOVED
BIC      SP,R1      ;SET R1=X7752,MASK IN SP=24
MOV      R1,@R1      ;STORE OWN ADDRESS IN POINTER
LOOP:  MOV      @R1,R2      ;GET BYTE POINTER
INC      @R4      ;ENABLE READER
TSTB     @R4      ;TEST DONE BIT
BPL      #-2      ;WAIT UNTIL READY
MOV      2(R4),@R2      ;THEN PICK IT UP AND STORE IT
INC      @R1      ;BUMP POINTER
CMPB     R2,#375      ;STORED JUMP OFFSET?
BNE      LOOP      ;NOT YET
INCB     (R2)+      ;YES,ALL DONE
JMP      -(R2)      ;GO EXECUTE AS BRANCH
:
: DEVICE ADDRESSES FOLLOW - DO NOT CHANGE THE ORDER
DEV:  LSR      ;LOW SPEED READER
HSR      ;HIGH SPEED READER
:
.SBTTL CASSETTE BOOT

```



```

2366
2367
2368
2369          177500
2370
2371 007500 012700 177500
2372 007504 005010
2373 007506 010701
2374 007510 062701 000052
2375 007514 012702 000375
2376 007520 112103
2377
2378 007522 112110
2379 007524 100413
2380 007526 130310
2381 007530 001776
2382 007532 105202
2383 007534 100772
2384 007536 116012 000002
2385 007542 120337 000000
2386 007546 001767
2387 007550 000000
2388 007552 000755
2389
2390 007554 005710
2391 007556 100774
2392 007560 005007
2393
2394 007562 017640
2395
2396 007564 002415
2397
2398 007566 112024
2399
2400 007570 000000 000000
2401 007574 167500
2402 007576 000340
2403
2404
2405

; CASSETTE BOOT
TACS=177500 ;TA-11 CONTROL AND STATUS REGISTER
      =ORIGIN+1500
TABOOT: 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-REG 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-REG.
        .WORD 2415 ;.BYTE 37: ILBS+READY+GO
        .WORD 112024 ;.BYTE 15: SFB+GO
        .WORD 0,0 ;.BYTE 5: READ+GO
        .WORD TABOOT!160000 ;.BYTE 24: READ+ILBS
        .WORD 340 ;.BYTE 224: READ+ILBS+E.O.TABLE
        ;THESE ARE FILLER WORDS
        ;POWER UP VECTOR AND PRIORITY

.SBTTL MR11-DB BOOT

```


M04

SCROLLING ROM BOOTSTRAP FOR THE GT40
DDGTEC.P11 05-NOV-76 10:20

MACY11 27(1006) 05-NOV-76 12:10 PAGE 52
MR11-DB BOOT

2462 007720 010702
2463 007722 000401
2464 007724 177450
2465
2466
2467 007726 000005
2468 007730 010200
2469 007732 005720
2470 007734 012001
2471 007736 012711 177000
2472 007742 011041
2473 007744 032711 100200
2474 007750 001775
2475 007752 100757
2476 007754 005007
2477
2478 007756 000000
2479 007760 167610
2480 007762 000340
2481 007764 167720
2482 007766 000340
2483 007770 167654
2484 007772 000340
2485 007774 167620
2486 007776 000340
2487
2488

RC11: MOV PC,R2 ;FIXED HEAD DISK (64KW)
BR OTHER ;ADRS OF WORD COUNT (COMMAND+2)
177450 ;COMMAND WORD (5) IS THE RESET

OTHER: RESET ;RD TO POINT AT WORD COUNT ADRS
MOV R2,R0 ;POINT TO ADDRESS
TST (0)+ ;WORD COUNT ADDRESS TO R1
MOV (0)+,R1 ;LOAD WORD COUNT
MOV #-1000,(1) ;COMMAND TO COMMAND REGISTER
MOV (0)-,(1) ;CHECK FOR ERROR OR DONE
BIT #100200,(1) ;IF NEITHER, KEEP LOOKING
BEQ -4 ;ERROR, TRY AGAIN
BMI AGAIN
CLR PC

0 ;FILLER
RKVEC: RK11!160000 ;RK POWER UP VECTOR
340
RCVEC: RC11!160000 ;RC POWER UP VECTOR
340
RPVEC: RP11!160000 ;RP POWER UP VECTOR
340
TCVEC: TC11!160000 ;TC11 POWER UP VECTOR
340

.SBTTL ROM VERSION 1 VALUES

2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544

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 ;RETURN OF VALUE REGISTER.
;INP1=R1 ;ARGUMENT FOR CALLED FUNCTION
;INP2=R2 ;SECOND ARGUMENT.
;WORK1=R3 ;FIRST WORK REGISTER.
;WORK2=R4 ;SECOND WORKING REGISTER.
;SCR1=R5 ;SCRATCH REGISTER.
;
;LCKSM=WORK1 ;OVERLAPPING DEFINITIONS FOR LOADER PORTION.
;LBYT=RET1
;LBC=SCR1
;LADR=INP1

```


B05

SCROLLING ROM BOOTSTRAP FOR THE GT40
DDGTEC.P11 05-NOV-76 10:20

MACY11 27(1006) 05-NOV-76 12:10 PAGE 54
ROM VERSION 1 VALUES

016000 012705 000026
016004 005015
016006 010745
016010 000005
016012 012767 000007 007570
016020 012767 000001 011532
016026 012767 000201 007560

036000
166000

000000
001360

022000
027560
025614
025610

027562
025612
025616

045776
045772
015770

160000

000024

016000

COREND=36000
ROMORG=166000

STARTX=0
STARTY=1360

VT40PC=172000-150000
KBDIS=27560
P100S=25614
P10IS=25610

KBDIB=KBDIS+2
P10IB=P10IS+2
P10OB=P100S+2

P100C=COREND-2+10000
P10IC=P100C-4
STKSRT=P10IC-2-30000

JMPDIS=160000

PWRFAL=24

. =16000
.=ROMORG

STARTA: MOV #PWRFAL+2, SCR1
CLR @SCR1
MOV PC, -(SCR1)

RESET

MOV #7, P10IS
MOV #1, KBDIS
MOV #201, P100S

; FIRST LOCATION OF NON-CORE.
; WHERE THE ROM PROGRAM SHOULD GO.
; WHERE TO START DISPLAYING THE X POSITIONS.
; WHERE TO START DISPLAYING THE Y.

; VT40 PROGRAM COUNTER.
; TTY INPUT STATUS.
; PDP-10 OUTPUT STATUS.
; PDP-10 INPUT STATUS.

; TTY INPUT BUFFER.
; PDP-10 INPUT CHARACTER.
; PDP-10 OUTPUT BUFFER.

; CHARACTER TO BE SENT TO THE PDP-10
; INPUT CHARACTER FROM 10 PLUS ONE SAVE CHARACTER
; FIRST LOCATION OF STACK.

; THE VT-40 DISPLAY JUMP INSTRUCTION.

; POWER FAIL RESTART LOCATION.

; SET THE ORIGIN NOW!!!!

; PICK UP POINTER TO P.F. STATUS.
; CLEAR IT OUT TO BE SURE.
; SET UP THE RESTART LOCATION.

; RESET THE BUS.

; INITIALIZE PDP-10 INPUT
; INITIALIZE TTY INPUT.
; INITIALIZE PDP-10 OUTPUT.

016034	012706	015770	RSTRT:	MOV	#STKSRT, SP	:SET UP THE STACK NOW!
016040	005001			CLR	LADR	:CLEAR ADDRESS POINTER.
016042	012702	160000		MOV	#JMPDIS, INP2	:PLACE A DISPLAY JUMP INSTRUCTION IN A REGISTER.
016046	010221			MOV	INP2, (LADR)+	:MOVE IT TO LOCATION 0.
016050	012711	166756		MOV	#DISPRG+150000, (LADR)	:MOVE ADDRESS POINTER INTO 2.
016054	012701	000030		MOV	#PWFAL+4, LADR	:SET UP WHERE WE WILL STORE CHARACTERS.
016060	005000			CLR	RET1	:PREPARE TO INSERT A ZERO CHARACTER.
016062	004767	000022		JSR	PC, DOCHAR	:INSERT IT NOW.
016066	005067	003706		CLR	VT40PC	:CLEAR THE DISPLAY PROGRAM COUNTER AND START.
016072	004767	000210	MAJOR:	JSR	PC, GTCHR	:GT A CHARACTER NOW.
016076	000240			NOP		
016100	000240			NOP		
016102	000240			NOP		
016104	012746	166072		MOV	#MAJOR+150000, -(SP)	:INSERT IN DISPLAY BUFFER NOW.
016110	010105		DOCHAR:	MOV	LADR, SCR1	:GT CURRENT BUUFER POSITION NOW.
016112	022525			CMP	(SCR1)+, (SCR1)+	:BYPASS CURRENT DISPLAY JUMP.
016114	005025			CLR	(SCR1)+	:CLEAR FUTURE ADDRESS FOR JUMP.
016116	010225			MOV	INP2, (SCR1)+	:STICK IN TEMPORARY JUMP WHILE WE REPLACE CURREN
016120	005015			CLR	(SCR1)	:A DISPLAY JUMP TO ZERO.
016122	005011			CLR	(LADR)	:NOW REPLACE CURRENT DISPLAY JUMP BY THE CHARACT
016124	050021			BIS	RET1, (LADR)+	:IT'S DONE THIS WAY TO WASTE 2 CYCLES.
016126	010211			MOV	INP2, (LADR)	:TO AVOID TIMING PROBLEMS WITH THE VT40.
016130	000207			RTS	PC	:AND FINALLY RETURN.
016132	004767	000124	GT8:	JSR	PC, GTSIX	:GT SIX BITS NOW.
016136	010046			MOV	RET1, -(SP)	:SAVE THE CHARACTER NOW.
016140	000401			BR	GTP84	:BYPASS THE 8'ER
016142	005002		GT84:	CLR	INP2	:RESET THE MAGIC REGISTER NOW.
016144	005722		GTP84:	TST	(INP2)+	:INCREMENT WHERE TO GO.
016146	066207	166250		ADD	GT8TB+150000(INP2), PC	:UPDATE PC NOW.
	016152		GT8P=.			
016152	004767	000104	GT81:	JSR	PC, GTSIX	:GT A CHARACTER NOW.
016156	010004			MOV	RET1, WORK2	:SAVE FOR A SECOND.
016160	006300			ASL	RET1	

2657	016162	006300		ASL	RET1	;SHIFT TO LEFT OF BYTE
2658	016164	106300		ASLB	RET1	
2659	016166	106116		ROLB	QSP	;PACK THEM IN.
2660	016170	106300		ASLB	RET1	
2661	016172	106116		ROLB	QSP	;A GOOD 8 BIT THING.
2662	016174	012600		MOV	(SP)+,RET1	;POP AND RETURN NOW.
2663	016176	000207		RTS	PC	
2664						
2665	016200	006300	GT82:	ASL	RET1	;WORST CASE. SHIFT 4
2666	016202	006300		ASL	RET1	
2667	016204	106300		ASLB	RET1	
2668	016206	106104		ROLB	WORK2	
2669	016210	106300		ASLB	RET1	
2670	016212	106104		ROLB	WORK2	
2671	016214	106300		ASLB	RET1	
2672	016216	106104		ROLB	WORK2	
2673	016220	106300		ASLB	RET1	
2674	016222	106104		ROLB	WORK2	
2675	016224	010400		MOV	WORK2,RET1	
2676	016226	012604		MOV	(SP)+,WORK2	
2677	016230	000207		RTS	PC	
2678						
2679	016232	006100	GT83:	ROL	RET1	
2680	016234	006100		ROL	RET1	
2681	016236	006004		ROR	WORK2	
2682	016240	106000		RORB	RET1	
2683	016242	006004		ROR	WORK2	
2684	016244	106000		RORB	RET1	;FINAL CHARACTER ASSEMBLED.
2685	016246	005726		TST	(SP)+	;FUDGE STACK.
2686	016250	000207		RTS	PC	;AND RETURN NOW.
2687						
2688		016250	GT8TB	=	.-2	;PUSH ZERO CONDITION BACK INTO NEVER-NEVER LAND.
2689						
2690	016252	000000		.WORD	GT81-GT8P	
2691	016254	000026		.WORD	GT82-GT8P	
2692	016256	000060		.WORD	GT83-GT8P	
2693	016260	177770		.WORD	GT84-GT8P	
2694						
2695						
2696	016262	004767	000020	JSR	PC,GTCHR	
2697	016266	020027	000040	CMP	RET1,#40	
2698	016272	002546		BLT	LBAD	
2699	016274	020027	000137	CMP	RET1,#137	
2700	016300	003143		BGT	LBAD	
2701	016302	000207		RTS	PC	
2702						
2703						
2704						
2705	016304	005726	GTCHP:	TST	(SP)+	;UPDATE THE STACK.
2706						
2707	016306	012700	015772	GTCHR:	MOV	#P10IC-30000,RET1
2708	016312	004767	000064	GTCHL:	JSR	PC,CHECK
2709	016316	005710			TST	QRET1
2710	016320	001774			BEQ	GTCHL
2711	016322	011046			MOV	QRET1,-(SP)
2712	016324	005020			CLR	(RET1)+
						;CLEAR THE CHAR GOT FLAG NOW.

2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824

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

016610 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 " L O A D E R

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.

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

LLD4: MOVBR 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

2825						
2826	016666	004767	177754	LJMP:	JSR	PC,LGWRD ;GT A WORD
2827	016672	010046			MOV	LBYT, -(SP) ;SAVE ON THE STACK.
2828	016674	004767	177730		JSR	PC,LPTR ;GT A CHARACTER.
2829	016700	105703			TSTB	LCKSM ;IS IT ZERO?
2830	016702	001342			BNE	LBAD ;YEP. WHAT CRAP.
2831	016704	032716	000001		BIT	#1,(SP) ;IS IT ODD?
2832	016710	001406			BEQ	LJMP1 ;YEP. START PROGRAM GOING NOW.
2833	016712	012700			MOV	(PC)+,RET1 ;TELL PDP-10 WE'VE LOADED OK.
2834					.BYTE	175,107 ;"CTRL GOOD"
2835	016714	107	175		.BYTE	107,175 ;"GOOD CTRL"
2836	016716	004767	000006		JSR	PC,SENDIT
2837	016722	000000			HALT	
2838	016724	000776			BR	.-2
2839						
2840	016726	000136		LJMP1:	JMP	2(SP)+ ;AND AWAY WE GO.
2841						
2842						
2843						
2844						
2845						
2846						
2847						
2848						
2849						
2850						
2851						
2852						
2853						
2854						
2855						
2856						
2857						
2858	016730	004767	177446	SENDIT:	JSR	PC,CHECK ;POLL THE OUTPUT DEVICE NOW.
2859	016734	005767	027036		TST	P100C ;OUTPUT CLEAR?
2860	016740	001373			BNE	SENDIT ;NOPE. LOOP AWHILE LONGER.
2861	016742	010067	006650		MOV	RET1,P100B ;SEND OUT THE CHARACTER.
2862	016746	105000			CLRB	RET1 ;CLEAR THE BYTE.
2863	016750	000300			SWAB	RET1 ;AND SWAP THEM NOW.
2864	016752	001366			BNE	SENDIT ;IF NOT EQUAL, REPEAT.
2865	016754	000207			RTS	PC
2866						
2867						
2868						
2869						
2870						
2871						
2872						
2873						
2874						
2875						
2876						
2877						
2878						
2879						
2880						

2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908

016756 170256
016760 115124
016762 000000
016764 001360
016766 100000
016770 160000
016772 000030
016774 000000
016776 000000

017000 000000
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

DSWR =	177570	449#	564				
ENDCOR	006036	1721#	1722				
ERRVEC=	000004	448#	562	563*	570*		
FF	006256	1808	1838#				
FFLOOP	006262	1833	1840#	1843			
FILED	007214	2254	2257#	2325			
FILEOA	005722	580*	683*	687*	1516#		
FILEOB	005724	687	1517#				
FILEOC	005752	580	683	1516	1525#		
FILEOD	003012	529	843#	1527			
GETCHR	006564	1791	1981#	2003			
GETDL	006516	1959#	1964	1968	1970	1981	1985
GETDL1	006546	1960	1967#				
GETEXT	006650	1983	1989	2009#			
GETSIX	006630	2003#	2041	2056			
GETB	006664	2041#	2150				
GETBTB	006700	2044	2046#				
GETB1	006712	2046	2056#				
GETB2	006740	2047	2068#				
GETB3	006772	2048	2083#				
GETB4	006706	2052#					
GRAPHX=	120000	187#	1040	2206#	2306		
GRAPHY=	124000	187#	876	2207#	2317		
GRPINC	003074	581*	679*	680	682*	874#	
GTADD	001000	459#	542				
GTBRL	001004	461#	574	576	578		
GTBUSE	006424	1893#	1916				
GTCHL	016312	2708#	2710				
GTCHP	016304	2705#	2714	2716			
GTCHR	016306	2614	2696	2707#	2728		
GTDLYO	001314	540#	579*	676*	678*		
GTDNE1	001170	503#	574*				
GTDONE	001166	502#	573*				
GTLPEN	002220	575	687#				
GTLPH	001172	505#	575*				
GTLPH1	001174	506#	576*				
GTNP	016366	2718	2725#				
GTPC	001146	492#	529*	541	684*	688*	
GTPB4	016144	2647	2649#				
GTSHIF	002236	577	691#				
GTSIX	016262	2645	2654	2696#			
GTSOTM	001176	508#	577*				
GTSOT1	001200	509#	578*				
GTSR	001150	493#	671	818*			
GTSTOP	002130	573	671#				
GTST1	002202	677	681	683#			
GTVCT	001002	460#	546				
GTXP05	001152	494#					
GTYP05	001154	495#					
GT40PC=	172000	1652#	1653	1893*	1927	2254*	
GT40SR=	172002	1653#	1835*				
GTB	016132	2645#	2813				
GTBP =	016152	2652#	2690	2691	2692	2693	
GTBTB =	016250	2650	2688#				
GTB1	016152	2654#	2690				
GTB2	016200	2665#	2691				

COMMEN	107				
ENDCOM	107				
ESCAPP	107				
PTPR	107				
PTSWR	107				
GRAPH	107	877	1042		
MULT	107				
NEWST	107				
OCTGN	107	1201	1213	1225	1237
OCTGN	107	1204	1206		
PAT1	107	1209	1210		
PAT2	107	1209	1210		
PAT3	107	1209	1210		
POP	107	1207	1208		
PUSH	107				
REPORT	107				
PTPR	107				
PTUP	107				
SKIP	107				
BLASH	107				
STARS	107				
WRSU	107				
TYPBIN	107				
TYPDEC	107				
TYPNAM	107				
TYPNUM	107				
TYPPCS	107				
TYP OCT	107				
TYPTXT	107				
SS ESCA	107				
SS NEWT	107				
SS SKIP	107				
.EQUAT	107				
.HERDE	107				
.KT11	107				
.SETUP	107				
.SWAH	107				
.SACT1	107				
.SACTB	107				
.SACTH	107				
.SACTY	107				
.ASTA	107				
.CASC	107				
.CMTR	107				
.DB2D	107				
.DB2O	107				
.DIV	107				
.FOR	107				
.FRRO	107				
.FRRT	107				
.MULT	107				
.POWE	107				
.SRAND	107				
.SRDE	107				
.SRDOC	107				
.SREAD	107				

SCROLLING ROM BOOTSTRAP FOR THE GT40
DDGTEC.P11 05-NOV-76 10:20

MACY11 27(1006) 05-NOV-76 12:10 PAGE 70
CROSS REFERENCE TABLE -- MACRO NAMES

..SR2A7	18
..SR2AV	18
..SR2B2D	18
..SR2B2O	18
..SR2C09	18
..SR2I27	18
..SR2R2	18
..SR2TYPB	18
..SR2TYPD	18
..SR2TYPD	18
..SR2TYP	18
..SR40CA	18
..1170	18

. ABS. 017002 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DDGTEC.BIN,DDGTEC.SEG/SOL/CRF/NL:TOC=DDGTEC.SML,DDGTEC.P11
RUN-TIME: 24 32 1 SECONDS
RUN-TIME RATIO: 123/59=2.0
CORE USED: 33K (65 PAGES)

