

1

REM

TITLE GT40 ROM BOOTSTRAP TEST MAINDEC-11-DDGTD-B

IDENTIFICATION

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

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 OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

1.

ABSTRACT

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

THE AVAILABLE TESTS ARE

PRG0 = LOGIC TESTS
PRG1 = ROM DATA DUMP TO THE CONSOLE TELETYPE
PRG2 = SINGLE ROM ADDRESS READ DATA LOOP

2.

REQUIREMENTS

2.1

EQUIPMENT

GT40 DISPLAY PROCESSOR WITH ROM BOOTSTRAP VERSION 1 OR 2.

2.2

STORAGE

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

3.

LOADING PROCEDURE

PROCEDURE FOR A NORMAL BINARY TAPE SHOULD BE FOLLOWED.

4.

STARTING PROCEDURE

4.1

STARTING ADDRESSES

0200 PROGRAM 0, ROM LOGIC TEST,
0204 PROGRAM 1, ROM DATA DUMP ON CONSOLE TTY,
0210 PROGRAM 2, SINGLE ROM READ,

4.2

SWITCH SETTINGS

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

5'' PROGRAM DESCRIPTIONS

5:1 PRG0 - LOGIC TESTS

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

5:1:1 ROUTINE DESCRIPTIONS

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

5:2 PRG1 - ROM DATA DUMP

THIS PROGRAM TYPES OUT THE 512,288, WORDS OF ROM DATA ON THE
CONSOLE TELETYPE AND HALTS.

5:3 PRG2 - SINGLE ROM ADDRESS READ DATA LOOP

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

6'' ERRORS

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

7'' EXECUTION TIME

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

151
152
153
154
155
156
157
164
165 000034
166 002034 002034
167 002036 000000
168
169 104400
170 177564
171 177564
172 177776
173 177570
174 177570
175 000500
176 000200
177 002200 000137 001024
178 002204 000137 001506
179 002210 000137 001670

.LIST ME,BIN#SEQ,LD
.ENABL ABS,AMA
;LOAD ADDRESS=0200
;DEPRESS START
;STACK POINTER IS AT 000

.LIST
.034
SCOPEC
0
;EQUATE STATEMENTS
SCOPESTRAP
TPCSR=177564
TPDBR=177564
PSW=177776
SR=177570
DISPLAY=177570 ;ADDRESS OF DISPLAY REGISTER
STKPTR=500 ;INITIAL STACK SETTING
.0200
JMP PRMTRS
JMP PRG1
JMP PRG2

181
182 001000
183 001000 166000
184 001002 001000
185 001004 000000
186 001006 172002
187 001010 000010
188 001012 000010
189 001014 000000
190 001016 000000
191 001020 000000
192 001022 000000
193 001024 012704 000500
194 001030 004737 002306
195
196
197
198 001034 005037 001014
199 001040 012706 000500
200 001044 012739 001040 002130
201 001052 013757 001014 177570
202
203
204
205 001060 013700 001000
206 001064 013701 001002
207 001070 012737 001130 000004
208 001076 011003
209 001100 009720
210 001102 064037 001016
211 001106 021010
212 001110 132020
213 001112 164037 001016
214 001116 062700 000002
215 001122 009301
216 001124 001304
217 001126 000403
218 001100 022026
219 001132 000000
220 001134 000700
221 001136 104000
222

.#1000
ROMADD 166000 ;ROM ADDRESS
WORDS 512, 1250 ;ROM LENGTH
;ROM IMAGE
;ROM IMAGE
DSR 172002 ;DISPLAY STATUS REGISTER
FILLER 10 ;# OF FILLER CHAR
FILLCNT 10
ICNT 0
DUMP 0
CHARA 0
TERM 0
PRMTRS MOV #STKPTR,NS ;SET STACK PTR
JSR PC,BNIT0N ;CHECK ROM VERSION
;PROGRAM LOGIC TESTS
PRG0 CLR ICNT ;CLEAR PASS COUNT
PRG0 MOV #STKPTR,NS
MOV #PRGR,RTURN ;SET RETURN ADDRESS FOR SCOPE
MOV ICNT,#0DISP ;DISPLAY PASS COUNT
;TEST1 TEST ABILITY TO REFERENCE ROM WITHOUT TIMING OUT
T1F MOV ROMADD,NS ;GET ROM ADDRESS
MOV WORDS,NS ;GET ADDRESS COUNTER
MOV #ERROR1,4 ;SET UP TIME OUT VECTOR
T1A MOV (0),NS ;REFERENCE
TST ;ROM
ADD -(0),DUMP
CMP (0),(0)
BITB (0),(0)
SUB -(0),DUMP
ADD #2,NS
DEC NS
BNE T1A ;INCREMENT POINTER
BR T1B ;INCREMENT ADDRESS COUNTER
;BRANCH IF NOT FINISHED
;GO TO SCORE LOOP
;REPOSITION STACK
;ERROR, TIME-OUT ON ROM ADDRESS
;LOOP ON ERROR
ERROR1 CMP (0),(0)
HALT T1A
BR T1A
T1B SCOPE

```

224
225
226
227 001140 013700 001000
228 001144 013701 001002
229 001130 012737 000004 000004
230 001196 000037 001016
231 001142 011003
232 001144 060037 001016
233 001170 163703 001016
234 001174 001402
235 001176 000000
236 001200 000766
237 001202 044037 001016
238 001206 001402
239 001210 000000
240 001232 000773
241 001234 021010
242 001236 001402
243 001240 000000
244 001282 000774
245 001224 122040
246 001226 001402
247 001230 000000
248 001232 000774
249 001234 000720
250 001236 000301
251 001240 001346
252 001242 104400
253

```

TESTS TEST THAT ROM DATA CAN BE READ RELIABLY.

T27 MOV ROMADD, #0
T28 MOV WORDS, #2
T29 MOV #0, #0
T30 CLR DUMP
T31 MOV (0), #0
ADD (0), #DUMP
SUB DUMP, #0
BEG T28
T32 BR T2A
T33 BIC = (0), #DUMP
T34 BEQ T2C
T35 HALT
T36 BR T2A
T37 T3C
T38 T3C
T39 T3C
T40 CMP (0), (0)
T41 BEQ T2D
T42 HALT
T43 BR T2C
T44 T3C
T45 CMPB (0), #- (0)
T46 BEQ T2E
T47 HALT
T48 BR T2D
T49 T3C
T50 DEC #1
T51 BNE T2A
T52 SCOPE

SET ROM ADDRESS
GET ADDRESS COUNTER
INITIALISE TIME OUT VECTOR
INITIALISE DUMP
GET DATA
ADD DATA TO DUMP
SUBTRACT DATA FROM DATA
BRANCH IF EQUAL
DATA ERROR
LOOP ON ERROR
CLEAR DUMP BIT
BRANCH IF EQUAL TO 0
DATA ERROR
LOAD ON ERROR
COMPARE DATA
BRANCH IF EQUAL
DATA ERROR
LOAD ON ERROR
COMPARE DATA (BYTE OPERATION)
BRANCH IF EQUAL
DATA ERROR
LOAD ON ERROR
INCREMENT ADDRESS POINTER
DECREMENT ADDRESS COUNTER
RETURN IF NOT DONE

```

255
256
257
258
259 001244 012706 000500
260 001230 013700 001000
261 001234 013701 001002
262 001200 012737 001274 000004
263 001206 010010
264 001270 000000
265 001272 000775
266 001274 012737 001312 000004
267 001302 022626
268 001374 000210
269 001326 000000
270 001330 000775
271 001332 012737 001332 000004
272 001330 022626
273 001322 000777 177492
274 001326 000000
275 001330 000774
276 001332 000720
277 001334 022626
278 001336 000301
279 001340 001347
280 001342 012737 000006 000004
281 001330 104400

```

TESTS TEST THAT ROM TIMES OUT IF REFERENCED BY OTHER
ITMAN DATA BUS CYCLE

T37 MOV #STMPTR, #0
T38 MOV ROMADD, #0
T39 MOV WORDS, #2
T40 MOV #T30, #4
T41 MOV #0, #0
T42 HALT
T43 BR T3A
T44 MOV #T30, #4
T45 CMP (0), # (0)
T46 INC (0)
T47 HALT
T48 BR T3C
T49 MOV #T30, #4
T50 CMP (0), # (0)
T51 CLR #ROMADD
T52 HALT
T53 BR T3E
T54 T3C
T55 CMP (0), # (0)
T56 DEC #1
T57 BNE T3AA
T58 MOV #6, #4
T59 SCOPE

SET STACK PTR
GET ROM ADDRESS
GET ADDRESS COUNTER
SET UP TIME OUT VECTOR
ATTEMPT TO ALTER DATA
HERE IF DID NOT TIME OUT
LOOP ON ERROR
SET UP TIME OUT VECTOR
REPOSITION STACK
ATTEMPT TO ALTER DATA
HERE IF DID NOT TIME OUT
LOOP ON ERROR
SET UP TIME OUT VECTOR
REPOSITION STACK
ATTEMPT TO ALTER DATA
HERE IF DID NOT TIME OUT
LOOP ON ERROR
INCREMENT ADDRESS POINTER
REPOSITION STACK
DECREMENT ADDRESS COUNTER
RETURN IF NOT DONE
RESTORE TIME OUT TRAP
SCOPE LOOP

```

283
284
285
286
287
288
289
290
291 001392 012700 000000          T4T  MOV  #0, #0          ;SET UP INITIAL WORD COUNT
292 001396 013701 001004          MOV  IMAGE, X1        ;SET UP STARTING ADDRESS OF ROM IMAGE
293 001342 013703 001000          MOV  ROMADD, X3       ;SET UP STARTING ROM ADDRESS
294 001346 011102                   T4A1 MOV  (X1), X2       ;READ EXPECTED VALUE
295 001370 011304                   MOV  (X3), X4        ;READ ROM VALUE
296 001372 020204                   CMP  X2, X4           ;COMPARE EXPECTED TO THE VALUE READ
297 001374 001402                   BEQ  #0, #0           ;BRANCH IF CORRECT
298 001376 000000                   HALT                    ;ERROR, ROM VALUE FAILED TO EQUAL EXPECTED
299 001400 000772                   BR  T4A

300
301 001402 022123                   T4B1 CMP  (X1)+, (X3)+ ;INCREMENT ADDRESSES POINTERS
302 001404 005200                   INC  X2               ;INCREMENT WORD COUNT
303 001406 023700 001002                   CMP  WORDS, X0        ;COMPARE IF END WORD
304 001422 001300                   BNE  T4A              ;BRANCH IF NOT LAST WORD
305 001424 104400                   T4E1 SCOPE

306
307 001416 005237 001014          END1  INC  ICNT         ;INCREMENT PASS COUNT
308 001422 012777 000001 177356  MOV  #1, #0DR        ;RING THE STAR BELL
309 001430 012737 000207 177566  DONEB1 MOV  #207, #TPDR    ;RING THE TELETYPE BELL
310 001436 105737 177564          TSTB #TPDR
311 001442 100375                   BPL  #=4
312 001444 012737 000207 177566  IST  MOV  #207, #TPDR
313 001422 105737 177564          TSTB #TPDR
314 001426 100375                   BPL  15
315 001400 013700 000042          MOV  #40, X0         ;RETURN TO DECTAPE MONITOR
316 001404 001406                   BEQ  DONE1
317 001406 000005                   RESET
318 001470 000005                   RESET
319 001472 004710                   JSR  7, 0            ;RETURN
320 001474 000240                   NOP
321 001476 000240                   NOP
322 001500 000240                   NOP
323 001502 000137 001034          DONE11 JMP  PROG

324

```

```

326
327
328
329 001506 012706 000500          PRG11 MOV  #SYNTRX6        ;INITIALIZE STACK
330 001512 012737 000006 000004  MOV  #6, #4          ;SET UP BUSS ERROR
331 001520 004737 002306          JSR  PC, #WHITCH
332 001524 004537 001712          JSR  5, #TYPEM
333 001530 002275                   M0
334 001532 004537 001712          JSR  5, #TYPEM
335 001536 002252                   M7
336 001540 013701 001002          PRG1A1 MOV  WORDS, X1        ;GET # OF WORDS
337 001544 013700 001000          MOV  ROMADD, X0      ;GET STARTING ADDRESS
338 001530 012702 000010          MOV  #10, X2         ;GET ADDRESS INDICATOR
339 001534 105737 177564          TSTB #PCDR
340 001500 100375                   BPL  #=4
341 001562 010037 002144          PRG1B1 MOV  X0, #0B2TYP     ;TELEPRINTER FLAG
342 001566 004737 002146          JSR  7, 0A           ;GET ADDRESS
343 001572 004537 001712          JSR  5, #TYPEM
344 001576 002304                   M0
345 001600 012037 002144          PRG1C1 MOV  (0)+, #0B2TYP   ;TYPE
346 001604 004737 002146          JSR  7, 0A           ;DATA
347 001610 105737 177564          TSTB #PCDR
348 001614 100375                   BPL  #=4
349 001616 012737 000040 177566  MOV  #1, #TPDR
350 001624 005301                   DEC  X1
351 001626 001410                   BEQ  PRG1D
352 001630 005302                   DEC  X2
353 001632 001342                   BNE  PRG1C
354 001634 012702 000010          MOV  #10, X2         ;RETURN TO PRG1B
355 001640 004537 001712          JSR  5, #TYPEM
356 001644 002275                   M0
357 001646 000745                   BR  PRG1D
358 001630 004537 001712          PRG1D1 JSR  5, #TYPEM
359 001634 002275                   M0
360 001636 004537 001712          JSR  5, #TYPEM
361 001602 002275                   M0
362 001604 000000                   HALT
363 001606 000707                   BR  PRG1
364
365
366
367 001670 012706 000500          PRG21 MOV  #SYNTRX6
368 001674 012737 000006 000004  MOV  #6, #4
369 001702 013700 177570          MOV  #R, #R
370 001706 011001                   MOV  (0), X1
371 001710 000707                   BR  PRG2

```

```

ROUTINE TO TYPE A MESSAGE
373
374
375 001712 010026 TYPEM1 MOV X0,(6)* ;SAVE REGISTER 0
376 001714 012500 MOV (5)*,X0 ;PLACE MESSAGE ADDRESS IN R0
377 001716 112037 001022 MOVB (0)*,TERM ;GET TERMINATOR CHARACTER
378 001722 112037 001020 TYPEM1+ MOVB (0)*,CHARA ;GET NEXT CHARACTER
379 001726 123737 001020 001022 CHMFB CHARA,TERM ;HAS NEXT CHARACTER THE TERM
380 001734 001005 BNE TYPEMS ;CHARACTER
381 001736 014000 MOV (6),X0 ;RESTORE R0
382 001740 109737 177564 TSTB TPCBR
383 001744 100375 BPL ;=4
384 001746 000205 RTS ;
385 001750 123727 001020 000045 TYPEM1+ CHMFB CHARA,#IX ;AND EXIT
386 001756 001027 BNE TYPEMC ;HAS CHARACTER X
387 001760 109737 177564 TSTB TPCBR ;TEST TELEPRINTER FLAG
388 001764 100375 BPL ;=4 ;AND WAIT FOR DONE
389 001766 012737 000215 MOV #25,TPOBR ;LOAD TELEPRINTER WITH CAR, RET
390 001774 013737 001010 001012 BR FILLER,FILCNT ;LOAD FILLER COUNT
391 001802 000403 MOV #0,TPOBR ;PRINT FILLER CHAR
392 001804 012737 000006 177564 TSTB TPCBR ;TEST TELEPRINTER FLAG
393 001806 109737 177564 BPL ;=4 ;AND WAIT FOR DONE
394 001806 100375 DEC FILCNT ;FINISHED FILLERS ?
395 001808 009337 001012 BNE ;=25 ;IF NOT
396 001808 001367 BNE ;=25 ;IF NOT
397 001806 012737 000212 177564 MOV #25,TPOBR ;LOAD TELEPRINTER WITH LINE FEED
398 001804 000732 BR TYPEM1 ;GET NEXT CHARACTER
399 001806 109737 177564 TYPEM1+ TSTB TPCBR ;TEST TELEPRINTER FLAG
400 001804 100375 BPL ;=4 ;AND WAIT FOR DONE
401 001804 013737 001020 177564 MOV CHARA,TPOBR ;LOAD TELEPRINTER BUFFER
402 001802 000723 BR TYPEM1 ;AND GET NEXT CHARACTER
403
404
405
406 002004 032737 040000 177570 SCOPEC1 BIT #40000,SR ;TEST SR FOR SCOPE
407 002002 001023 BNE SCOPEB ;YES SCOPE
408 002004 032737 004000 177570 BIT #40000,SR ;TEST FOR ITERATION
409 002002 001007 BNE SCOPEC ;INHIBIT ITERATION
410 002004 023737 002126 002124 CHMFB SCOPEC,ICOUNT ;ITERATION COMPLETE
411 002102 001403 BEO SCOPEC ;ITERATION COMPLETE GO TO SCOPEC
412 002104 005237 002126 INC SCOPEC ;INCREMENT ITERATION COUNT
413 002110 000410 BR SCOPEB ;GO TO SCOPEB
414 002112 005037 002126 SCOPEC1 CLR SCOPEC ;CLEAR ITERATION COUNT
415 002116 011637 002130 MOV #X6,RETURN ;GET ADDRESS OF NEXT TEST
416 002132 000002 RTI ;EXIT
417 002124 000100 ICOUNT1 100 ;
418 002116 000000 SCOPEC1 0 ;CONTAINS SUBTEST ITERATION COUNT
419 002130 000000 RETURN1 ,WORD 0 ;CONTAINS RETURN PC FOR SCOPE
420 002132 005726 SCOPEC1 TST (6)* ;POP PC
421 002134 012637 177776 MOV (6)*,PSW ;RESTORE CONDITION CODES
422 002140 000177 177764 JMP @RETURN
423

```

```

THIS ROUTINE CONVERTS AN OCTAL NUMBER TO ASCII AND TYPES IT ON THE TTY.
425
426
427
428
429 002144 000000 D2BTYP1 0
430 002146 013746 177564 O2A1 MOV TPCBR,(6) ;SAVE TPCBR
431 002132 010246 MOV X2,(6) ;SAVE R2
432 002134 010146 MOV X1,(6) ;SAVE R1
433 002136 010046 MOV X0,(6) ;SAVE R0
434 002140 013700 002144 MOV D2BTYP1,X0 ;GET DATA TO BE TYPED
435 002144 012701 000006 MOV #6,X1 ;GET COUNTER
436 002170 005002 CLR X2 ;CLEAR WORKING REGISTER
437 002172 004100 ROL X0 ;MOV FIRST BIT (MSB) INTO
438 002174 004100 ROL X2 ;R2
439 002176 042702 000260 O2AA1 ADD #20,X2 ;FORM ASCII CODE
440 002202 109737 177564 TSTB TPCBR ;TEST TELEPRINTER
441 002206 100375 BPL ;=4 ;FLAG AND WAIT UNTIL DONE
442 002210 010237 177564 MOV X2,TPOBR ;LOAD TELEPRINTER BUFFER
443 002214 005002 CLR X2 ;CLEAR WORKING REGISTER
444 002216 004100 ROL X0 ;ROTATE THE
445 002218 004100 ROL X2 ;NEXT
446 002222 004100 ROL X0 ;OCTAL CHARACTER
447 002224 004100 ROL X2 ;INTO
448 002226 004100 ROL X0 ;REGISTER
449 002228 004100 ROL X2 ;TWO
450 002230 005102 DEC X1 ;DECREMENT COUNTER
451 002234 001300 BNE O2AA ;GO TO O2AA IF NOT 0
452 002236 012600 MOV (6)*,X0 ;FINISHED, RESTORE REGISTERS
453 002240 012601 MOV (6)*,X1
454 002242 012602 MOV (6)*,X2
455 002244 012637 177564 MOV (6)*,TPCBR ;AND TPCBR
456 002250 000207 RTS ;AND EXIT
457
458
459 002252 022500 052107 032095 IASCII MESSAGES ;ASCII MESSAGES
460 002260 020000 049522 020115 M9T ,ASCII ;M9T=00 ROM DATAXX0
461 002266 040504 040524 022449 ;
462 002274 100 ;
463 002276 100 022445 100 M9T ,ASCII ;M9T=00
464 002278 100 020040 100 M9T ,ASCII ;M9T=00
465 002280 022306 ;EVEN
466 002306 032737 000400 177570 SWITCH1 BIT #400,SR ;TEST BIT 0
467 002314 001007 BNE ;=1 ;IF VERSION 1
468 002316 012737 001000 001002 MOV #530,Y,WOROS ;SET UP VERSION 2 LENGTH
469 002324 012737 000000 001004 MOV #START,IMAGE ;SET UP VERSION 2 STARTING ADDR
470 002332 000406 BR ;=25 ;
471 002334 012737 000400 001002 IST MOV #250,Y,WOROS ;SET UP VERSION 1 LENGTH
472 002342 012737 010000 001004 MOV #START,IMAGE ;SET UP VERSION 1 STARTING ADDR
473 002350 000207 TST RTS ;PC

```

.SBTTL ROM VERSION 2 VALUES

475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528

EXCEPT FOR THE NEW ORIGIN ADDRESS AND SEVERAL "1160000"
FOR ADDRESS PADDING THIS IS AN EXACT COPY OF THE CONTENTS
OF THE GT-48 BOOTSTRAP VERSION #2

.TITLE SCROLLING ROM BOOTSTRAP FOR THE GT48

BOOTGT.T16 OCT 18, 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 GT48 DISPLAY TERMINAL. IT INCLUDES SCROLLING AND AN END OF
MEMORY SEARCH FOR THE LOADER.

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

.SBTTL DEFINITION SECTION
.PAGE

529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582

REGISTER DEFINITIONS

BASIC DEFINITIONS

!DEFINE STANDARD VALUES.

R00X0
R10X1
R20X2
R30X3
R40X4
R50X5
SP0X6
PC0X7

GT48 DEFINITIONS

CHAR0R0
POINTR01
TABCNT0R2
SCAN0R3
HOLD0R4
COUNT0R5

!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.

LOADER DEFINITIONS

L.BYTBCHAR
L.ADR=POINTR
L.BC=TABCNT
L.CKSH=COUNT
INDEX=SCAN

!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.

.PAGE

```

583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635

```

MAJOR SYSTEM DEFINITIONS

```

166000      ORIGIN=166000      |ORIGIN OF THE BOOTSTRAP.
175610      DL11IS=175610      |INPUT STATUS REGISTER OF DL11
175612      DL11IS=DL11IS*2    |INPUT CHARACTER FROM DL11
175614      DL11OS=DL11IS*2    |OUTPUT STATUS OF THE DL11
175616      DL11OS=DL11OS*2    |OUTPUT CHARACTER TO THE DL11
177560      KBDIS=177560      |KEYBOARD INPUT STATUS
177562      KBDIS=KBDIS*2      |CURRENT CHARACTER FROM KEYBOARD.
172000      GT40PC=172000      |GT40 PROGRAM COUNTER.
172002      GT40SR=GT40PC*2    |GT40 STATUS REGISTER ADDRESS.
001000      START=1000        |START OF THE DISPLAY BUFFER.
007000      BLIMIT=7000       |APPROXIMATE END OF THE DISPLAY BUFFER.
007776      TMPE=007776       |LOCATION OF INITIALIZATION STACK.
000004      CORSTR=4          |LOCATION OF PCP-13 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=0015         |CARRIAGE RETURN = LINE FEED
000175      ALTMOD=175        |THE "KEY" CHARACTER (I.E., ALTMODE).
160000      DISJMP=160000     |THE GT40 JMP INSTRUCTION
173000      DISTOP=173000     |THE GT40 STOP DISPLAY INSTRUCTION.

```

.BOTTL INITIALIZATION AND RESTART CODE
.PAGE

INITIALIZATION AND RESTART CODE

```

636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689

```

GT40 BOOTSTRAP CODE

```

006000      .ORIGIN          |DEFINE ORIGIN OF THE BOOTSTRAP.

```

CPU0 INITIALIZATION CODE

```

000000      000000 000000 175610  STARTI  RESET      #7,DL11IS      |RESET ALL HARDWARE NOW.
000002      012737 000007 175610  MOV       #7,DL11IS |INITIALIZE DL=11 INPUT NOW.
000003      012706 007776          MOV       #TMPE,SP  |ESTABLISH A GOOD TEMPORARY STACK
000014      005237 175614          INC       DL11OS   |POINTER FOR CORE SEARCH.
000020      004337 166692          JSR      #SCANOUTLIT|160000 |SET BREAK BIT
000024      000000          .WORD 0      |FOR 2 CHARACTER TIMES
000026      012703 000004          MOV       #CORSTR,SCAN |SET ADDRESS OF BAD CORE TRAP VECTOR.
000032      012723 166042          MOV       #NOTHERE|160000,(SCAN)+ |AND INSERT A POINTER TO US THERE.
000036      005023          ENDCOR  CLR      (SCAN)+  |NOW CLEAR ALL OF MEMORY BEYOND THE POINTER,
000040      000776          BR       ENDCOR    |UNTIL WE RUN OUT OF MEMORY AND TRAP.
000042      005743          NOTHERE  TST     =(SCAN)  |WHEN WE TRAP OUT, WE COME HERE.
000044      010306          MOV       SCANVSP   |WE BACK UP POINTER TO GOOD CORE.
000046      105737 175614          ISL     TSTB     DL11OS  |NOTE THAT IF WE TRAP OUT AGAIN, IT
000052      100375          CPL     IS        |IS STILL OK, BECAUSE WE WILL LOOP
000054      005037 175614          CLR     DL11OS    |UNTIL WE GET A GOOD CORE ADDRESS.
000056          ISL     IS        |WHEN WE GET ONE, THAT IS LAST LOCATION
000058          CPL     IS        |IN THE MACHINE, AND HENCE OUR SP.
000060          CLR     DL11OS  |SEE IF BREAK IS OCNE
000062          ISL     IS        |INGO BACK
000064          CLR     DL11OS  |CLEAR BREAK BIT

```

```

690 RESTART INITIALIZATION CODE WHEN COMMUNICATIONS IS WORKING.
691 -----
692
693
694
695 001060 052706 007776 RESTRTO BIS #TMPENDISP IFORCE THE SP TO LIMIT OF EXISTING CORE.
696
697
698 001064 012703 006700 MOV #BLIMIT=NUMLIN=NUMLIN,SCAN INOW WE WILL FILL THE KEY AREAS OF THE
699 001070 012702 006040 MOV #NUMLIN,TABCNT ION DISPLAY BUFFER WITH INITIAL CR-LP'S.
700
701 001074 012723 005015 SETLP1 MOV #CRLF,(SCAN)+ IINSERT A CRLF NOW.
702 001100 005302 TABCNT IAND LOOP UNTIL DONE.
703 001102 005374 BGT SETLP1 ITHUS DISPLAY CORE IS ALMOST CORRECT.
704
705
706 001104 012703 166432 MOV #SETUP1160000,SCAN INOW WE WILL INITIALIZE CORE FOR THE
707 ION DISPLAY, PICK UP POINTER TO LIST.
708
709 001110 012302 SETLP2 MOV (SCAN)+,TABCNT IGET NUMBER OF ITEMS TO INSERT.
710 001112 001405 SETDUN BEQ #1405 IIF ZERO, WE ARE DONE.
711 001114 012301 MOV (SCAN)+,POINTR IPICK UP FIRST CORE ADDRESS POINTER.
712
713 001116 012321 SETLP3 MOV (SCAN)+,(POINTR)+ IMOVE OVER A DATA ITEM NOW.
714 001120 005302 TABCNT IALL DONE?
715 001122 005375 BGT SETLP3 INOPE, MOVE OVER THE NEXT.
716 001124 000771 BR SETLP2 IYES, GET NEXT MANCH LIST TO INSERT.
717
718
719 001126 012701 006776 SETDUN MOV #BLIMIT=2,POINTR IESTABLISH THE BUFFER POINTER NOW.
720
721
722
723
724
725
726
727
728
729
730
731
732

```

.8BTTL VT02 SIMULATOR
 .PAGE

```

733
734
735
736
737
738
739
740
741 001132 004737 166564 NXTCHR JSR PC,GETCHR1160000 IGET A CHARACTER NOW.
742 001136 020027 000177 CMP CHAR,#127 IIS IT OUT OF RANGE?
743 001142 002373 BGE NXTCHR IYES, GET ANOTHER ONE.
744 001144 020027 000040 CMP CHAR,#00 IIS IT A PRINTING CHARACTER?
745 001150 002020 BGE NORMAL IYES, IT'S A NORMAL PRINTING CHARACTER.
746 001152 010003 MOV CHAR,SCAN IMOVE IT OVER SO WE CAN PLAY WITH IT.
747 001154 162703 000007 SUB #7,SCAN IBIAS SO THAT BELL [7] IS ZERO.
748 001160 020327 000007 CMP SCAN,#7 IIF CHARACTER IS LESS THEN BELL OR
749 001164 103302 BHS NXTCHR IOREATER THEN CR, THEN IGNORE.
750 001166 004303 SCAN ASL IIF GOOD, MAKE IT WORD INDEX.
751 001170 060307 ADD SCAN,#PC IAND GO TO THE CORRECT ROUTINE.
752
753 BR BELL IBELL
754 001174 000406 BR NORMAL IIS BACKSPACE
755 001176 000411 BR TAB IBTAB
756 001200 000437 BR LF IIS LINE FEED [LF]
757 001202 000420 BR VT IIS VERTICAL TAB [VT]
758 001204 000424 BR FF IIS FORM FEED [FF]
759 IIS CARRIAGE RETURN [CR]
760
761
762 001206 012702 177777 CRF MOV #1,TABCNT IRESET TAB POSITION ON A CR, AND
763 IFALL THROUGH TO INSERT THE CHARACTER.
764
765
766 001212 004737 166350 NORMAL JSR PC,INSERT1160000 IINSERT THE CHARACTER IN THE BUFFER.
767 001216 005202 INC TABCNT IUPDATE TAB POSITION NOW.
768 001220 000544 BR NXTCHR IAND GET NEXT CHARACTER.
769
770
771
772
773 001202 012700 000040 TABI MOV #0,CHAR ION A TAB, INSERT BLANKS UNTIL THE
774 001206 004737 166350 JSR PC,INSERT1160000 INEXT CHARACTER POSITION IS A MULTIPLE
775 001212 005202 INC TABCNT IOF 0.
776 001234 032702 000007 BIT #7,TABCNT IARE WE DONE YET?
777 001240 004370 BNE TAB INOPE.
778 001242 000733 BR NXTCHR IYES.
779
780
781 001244 111705 VFI MOVB (PC)+,COUNTR ITHIS PUTS THE LOW BYTE OF THE
782 IBRANCH CODE IN COUNTR.SAVE A WORD
783 001246 000405 BR FFLOOP
784
785 001250 005037 172002 BELL CLR GT40SR IIRING BELL -WRITE IN GT40SR
786 001254 000726 BR NXTCHR IAND LOOP BACK

```


894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
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

COMMUNICATIONS HANDLING ROUTINES

THE DL=11 HANDLE

```

GETDL1 TSTB DL11IS      ;CHECK THE HOST INPUT STATUS.
      SPL  GETDL1      ;HOST DID NOT SEND ANYTHING, YET.
      MOV  DL11IB,CHAR ;HOST SENT US A CHARACTER, PROCESS IT.
      MOV  #7,DL11IS  ;REENABLE THE HOST TELECOMMUNICATIONS.
      BIC  #288,CHAR  ;MAKE CHARACTER JUST SEVEN BITS.
      BEQ  GETDL      ;IF NULL, IGNORE IT.
      RTS  PC         ;ELSE RETURN NOW.
  
```

```

GETDL1 TSTB KBDIS      ;DID USER TYPE A CHARACTER?
      SPL  GETDL      ;NO, GO BACK AND CHECK HOST MACHINE.
      MOV  KBDIB,DL11OB ;MOVE THE CHARACTER TO THE HOST.
      BR   GETDL      ;AND CHECK AGAIN FOR INPUT.
  
```

THE "GET CHARACTER" ROUTINE

```

GETCHR1 JSR  PC,GETDL100000 ;GET A CHARACTER FROM THE HOST NOW.
      CMP  CHAR#ALTMOD ;IS IT AN "ALTMODE"?
      BNE  GETEXT      ;NO, EXIT NOW.
      JSR  PC,GETDL100000 ;YES, GET ANOTHER ONE NOW.
      CMP  CHAR#L      ;IS IT AN "L"?
      BEQ  LOADER      ;YES, START LOADING NOW.
      CMP  CHAR#R      ;IS IT AN "R"?
      BNE  GETEXT      ;NO, IGNORE THE ALTMODE AND JUST RETURN THE CHAR
      MOV  #D1STOP,JMPADD=2 ;YES, REEF, STOP DISPLAY BY INSERTING A "D1STOP
      PRESTR1 JMP  RESTRT100000 ;INSTRUCTION IN THE BUFFER, AND RESTART.
  
```

948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001

THE "GET A SIX BIT CHARACTER" ROUTINE

```

GETSIX1 JSR  PC,GETCHR100000 ;GET A CHARACTER NOW.
      CMP  CHAR#400 ;IS IT A LEGAL PRINTING CHARACTER?
      BLT  L,BAD      ;NOPE, ABORT
      CMP  CHAR#237 ;IS IT BIG ENOUGH, IS IT TOO BIG?
      BGT  L,BAD      ;YEP, ABORT?
      RTS  PC         ;RETURN TO THE CALLER.
  
```

THIS OUTPUTS TWO CHARACTERS VIA A JSR SCAB,OUTLIT

```

OUTLIT1 MOV  (SCAN)*DL11OB ;DOUBLE BUFFERED
      MOV  (SCAN)*DL11OB ;RETURN
      RTS
  
```

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).

```

GET81 JSR  PC,GETSIX100000 ;GET A SIXBIT CHARACTER.
      MOV  L,BYT,=SP ;SAVE IT ON THE STACK.
      TST (INDEX) ;UPDATE INDEX TO NEXT ITEM (ALL ARE *2)
      JMP GET8B=2;100000(INDEX) ;AND DISPATCH ACCORDING TO THE INDEX.
  
```

```

GET8TB1 BR  GET81 ;INDEX#2: ASSEMBLE FIRST CHAR
      BR  GET82 ;INDEX#4: ASSEMBLE SECOND CHAR
      BR  GET83 ;INDEX#6: ASSEMBLE THIRD AND LAST CHAR
      BR  GET84 ;INDEX#8: RESET INDEX TO 8 (2) AND RETRY.
  
```

1022								
1023	004776	012703	000002	GET84:	MOV	#2,INDEX		ITHE FOURTH INDEX IS THE SAME AS THE FIRST INDEX, JUST RESET IT AND FALL THROUGH,
1024								
1025								
1026								
1027	004712	004737	166630	GET81:	JSR	PC,GETSIX 160000		IGET ANOTHER CHARACTER NOW,
1028	004716	010004			MOV	L,BYT,HOLD		IAND PRESERVE IT FOR NEXT TIME THROUGH,
1029	004720	006300			ASL	L,BYT		INOW THROW AWAY LEFT MOST BITS OF
1030	004722	006300			ASL	L,BYT		ITHE 8 BIT CHARACTER, NOW MERGE IN
1031	004724	106300			ASLB	L,BYT		ITHE LEFT TWO BITS OF THE
1032	004726	106116			ROLB	(SP)		INOW SIX BIT CHARACTER WITH THE SIX
1033	004730	106300			ASLB	L,BYT		IBITS FROM THE CHARACTER ON THE
1034	004732	106116			ROLB	(SP)		ISTACK. 1ST CHARACTER IS NOW ASSEMBLED,
1035	004734	012600			MOV	(SP)+,L,BYT		ISO WE'LL RETURN IT TO THE USER,
1036	004736	000207			RTB	PC		IAND THEN WE SHALL RETURN TO HIM,
1037								
1038								
1039								
1040	004740	006300		GET82:	ASL	L,BYT		ITHE SECOND CHARACTER IS CREATED FROM
1041	004742	006300			ASL	L,BYT		ITHE 4 RIGHT BITS OF THE PREVIOUS CHARACTER
1042	004744	106300			ASLB	L,BYT		IAND THE FOUR MIDDLE BITS OF THE PRESENT
1043	004746	106104			ROLB	HOLD		8 BIT CHARACTER,
1044	004750	106300			ASLB	L,BYT		WE WILL CREATE THE NEW 8 BIT
1045	004752	106104			ROLB	HOLD		IN THIS REGISTER, SINCE IT
1046	004754	106300			ASLB	L,BYT		IS MORE CONVENIENT, WE WILL MOVE OVER THE
1047	004756	106104			ROLB	HOLD		ANSWER AT THE END,
1048	004760	106300			ASLB	L,BYT		IS ONE MORE TO GO
1049	004762	106104			ROLB	HOLD		IS DONE,
1050	004764	010400			MOV	HOLD,L,BYT		IBRING OVER THE VALUE,
1051	004766	012604			MOV	(SP)+,HOLD		IAND REMEMBER THE LAST CHARACTER WE RECEIVED,
1052	004770	000207			RTS	PC		IAND RETURN TO THE CALLER,
1053								
1054	004772	006100		GET83:	ROL	L,BYT		IFINAL CHARACTER IS EASY, JUST A
1055	004774	106100			ROLB	L,BYT		ISIMPLE MERGER OF LEFT TWO BITS OF
1056	004776	006004			ROR	HOLD		IPREVIOUS VALUE WITH RIGHT SIX BITS
1057	007000	106000			ROR	L,BYT		IOF LAST (4TH) CHARACTER RECEIVED,
1058	007002	006004			ROR	HOLD		
1059	007004	106000			ROR	L,BYT		IAND WE ARE DONE,
1060	007006	005726			TST	(SP)+		IFINALLY THROW AWAY STACK,
1061	007010	000207			RTS	PC		IAND RETURN TO THE CALLER,
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								

,SBTTL THE LOADER
 ,PAGE

1056								
1057								
1058								
1059								
1060								
1061								
1062								
1063								
1064								
1065	007012	012737	173000	007010	LOADER:	MOV	#DISTOP,JMPADD=2	ISTOP THE GT48 BY INSERTING A "DISTOP" IN THE LI
1066								
1067	007020	005003				CLR	INDEX	IRESET THE 8 BIT ASSEMBLER TO THE FIRST CHAR
1068								
1069								
1070	007022	005005		L.LD2:	CLR	L,CXSM		ICLEAR THE CHECKSUM
1071	007024	004737	167114		JSR	PC,L PTR 160000		IGET A BYTE NOW,
1072	007030	105300			DECB	L,BYT		IS IT A ONE (HEADER)?
1073	007032	001373			BNE	L.LD2		INO, WAIT FOR THE ONE,
1074								
1075	007034	004737	167114		JSR	PC,L PTR 160000		IFYES, SKIP OVER THE NEXT CHARACTER NOW,
1076								
1077	007040	004737	167126		JSR	PC,L G W D 160000		IASSEMBLE A WORD NOW,
1078	007044	010002			MOV	L,BYT,L BC		IMOVE OVER TO THE COUNTER,
1079	007046	162702	000004		SUB	#4,L BC		IREDUCE TO ACTUAL DATA COUNT,
1080	007052	022702	000002		CMP	#2,L BC		IANY DATA AT ALL?
1081	007056	001433			BEQ	L JMP		INO, MUST BE END
1082	007060	004737	167126		JSR	PC,L G W D 160000		IFYES, ASSEMBLE A DATA WORD NOW,
1083	007064	010001			MOV	L,BYT,L ADR		IAND THIS MUST BE THE FIRST ADDRESS,
1084								
1085								
1086	007066	004737	167114	L.LD3:	JSR	PC,L PTR 160000		IGET A BYTE OF DATA NOW,
1087	007072	002006			L.LD4			IFALL DONE?
1088	007074	105700			L,CXSM			IFYP, COUNTER IS MINUS; CHECK CHECKSUM,
1089	007076	001751			L.LD2			ICHECKSUM GOOD, GET NEXT COMMAND,
1090								
1091								
1092	007100	004337	166652	L.BAD:	JSR	SCAN OUT L 160000		IBAD LOAD INFORM HCBT
1093	007104	175	102		,BYTE	ALTRD ,B		ISEND ALTRD= B
1094	007106	000646			,BR	PRETRT		IAND RESTART THE DISPLAY,
1095								
1096								
1097	007110	110021		L.LD4:	MOV	L,BYT,(L ADR)+		INSERBYE INTO MEMORY,
1098	007112	000765			,BR	L.LD3		IAND GET THE NEXT BYTE,
1099								
1100								
1101								
1102	007114	004737	166664	L.PTR:	JSR	PC,GETB 160000		IASSEMBLE AN 8 BIT CHARACTER NOW,
1103	007120	000005			ADD	L,BYT,L CXSM		IUPDATE THE CHECKSUM NOW,
1104	007122	005302			DEC	L,BYT		IDECREMENT THE CHARACTER COUNTER,
1105	007124	000207			RTS	PC		IAND RETURN TO THE CALLER NOW,
1106								
1107								
1108								
1109	007126	004737	167114	L.GWRD:	JSR	PC,L PTR 160000		IASSEMBLE A WORD, FIRST GET A CHARACTER

```

1110 007132 010046      MOV      L, BYT, = {SP}      ;AND SAVE IT;
1111 007134 004737 167114 JSR      PC, L, PTR: 100000 ;AND THEN GET ANOTHER ONE;
1112 007140 000300      SHAB    L, BYT              ;AND THEN REASSEMBLE THE MESS;
1113 007142 052600      BIS     {SP}, L, BYT        ;WITH THE FEARSOME POWER OF THE 11;
1114 007144 000207      RTS     PC                  ;AND RETURN TO THE CALLER;
1115
1116
1117
1118
1119 007146 004737 167126 L.JMPI   JSR      PC, L, GNRD: 100000 ;ALL DONE WITH THE LOAD. ASSEMBLE
1120 007132 010046      MOV     PC, L, PTR: 100000 ;THE STARTING ADDRESS NOW;
1121 007134 004737 167114 JSR      PC, L, PTR: 100000 ;AND DON'T FORGET TO CHECKSUM IT;
1122 007100 105705      TSTB   L, CASH              ;
1123 007102 001346      BNE    L, BAD                ;A BAD CHECKSUM, ALL IS EVIL;
1124
1125 007104 004337 166692 JSR      SCAN: OUT: 11160000 ;GOOD CHECKSUM, INFORM HOST
1126 007170 175          ALTNOD: IC ;WITH ALTNOD G
1127
1128 007172 032716 000001 BTT     #1, {SP}             ;DO WE WANT TO START EXECUTION?
1129 007176 001401      BEQ    L, JMP1               ;YES, AWAY WE GO;
1130
1131 007200 000000      L.MALT; MALT                ;IF NOT, MALT;
1132
1133 007202 000136      L.JMP1; JMP     0({SP})+    ;IF GO, THEN GO ALREADY, WHEEEEE!
1134
1135
1136
1137          .SBTTL THE SELF TEST
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150          .PAGE

```

```

1151
1152
1153
1154
1155          100000      CHAR#100000 ;THIS IS G40 QUICK TEST
1156          104000      SHORTV#104000 ;GIVES QUICK VISUAL TEST
1157          110000      LONGV#110000 ;TOP CONDITION OF MACHINE
1158          114000      POINT#114000 ;WITHOUT READING IN DIAG;
1159          120000      GRAPHX#120000
1160          124000      GRAPHY#124000
1161          130000      RELATV#130000
1162
1163          002000      INT0#2000
1164          002200      INT1#2200
1165          002400      INT2#2400
1166          002600      INT3#2600
1167          003000      INT4#3000
1168          003200      INT5#3200
1169          003400      INT6#3400
1170          003600      INT7#3600 ;BRIGHTEST
1171
1172          000100      LPOFF#100
1173          000140      LPON#140
1174          000200      BLKOFF#20
1175          000300      BLKON#30
1176
1177          000004      LINE0#4
1178          000005      LINE1#5
1179          000006      LINE2#6
1180          000007      LINE3#7
1181
1182          160000      DJMP#100000
1183          164000      ONOP#104000
1184          170000      STAYS#170000
1185          173400      DSTOP#173400 ;STOP INTERRUPT
1186
1187          000300      LPLITE#300
1188          000200      LDARK#200
1189          000400      ITAL#40
1190          000600      ITAL1#60 ;ITALICS OFF
1191          000004      SYNON#4 ;" ON
1192
1193
1194          174000      STATSD#174000 ;SYNC ON
1195
1196          000100      INC#100 ;LOAD GRAPH INC#
1197          040000      INTX#40000 ;INTENSIFY BIT
1198          001777      MAXX#1777 ;BIGGEST X VECTOR
1199          001377      MAXY#1377 ;BIGGEST Y VECTOR
1200          020000      MINUSX#20000 ;THE MINUS BIT
1201          020000      MINUSY#20000
1202          017600      MAXSX#17600 ;BIGGEST X IN SHORTVEC
1203          000077      MAXSY#77 ;" " Y IN "
1204          000100      MINUSY#100 ;MINUS BIT FOR Y IN SHORTVEC

```

```

1205
1206
1207 007204 012737 167214 172000 MOV #FILE01160000,GT48PC ISTART THE GT48
1208 007212 000001 WAIT IAND WAIT
1209
1210 007214 114020 FILE01 POINTIBLKOFF IPOINT=;INVISIBLE
1211 007216 000000 0
1212 007220 001377 MAXY
1213
1214 007222 112004 LONGVIINT01,LINE0 IDRAW TOP LINE
1215 007224 041777 INTXIMAXX
1216 007226 000000 0
1217
1218 007230 112405 LONGVIINT21,LINE1
1219 007232 040000 INTX IDRAW LINE TO RIGHT
1220 007234 021377 MINUSXIMAXY
1221
1222 007236 113004 LONGVIINT41,LINE2
1223 007240 061777 INTX,MINUSXIMAXX IDRAW BOTTOM LINE
1224 007242 000000 0
1225
1226 007244 113407 LONGVIINT61,LINE3
1227 007246 040000 INTX IDRAW LINE TO LEFT
1228 007250 001377 MAXY
1229
1230 007252 114000 POINT
1231 007254 000400 400
1232 007256 000500 500
1233 007260 104200 SHORTVIINT1
1234 007262 057677 57677 I=X+Y
1235 007264 104600 SHORTVIINT3
1236 007266 077677 77677 I=X+Y
1237 007270 107200 SHORTVIINT5
1238 007272 077777 77777 I=X+Y
1239 007274 107600 SHORTVIINT7
1240 007276 057777 57777 I=X+Y
1241
1242 007300 114000 POINT
1243 007302 001400 1400
1244 007304 000500 500
1245 007306 133030 RELATIVINT4IBLKON
1246 007310 057677 57677 I=X+Y
1247 007312 077677 77677 I=X+Y
1248 007314 077777 77777 I=X+Y
1249 007316 057777 57777 I=X+Y
1250
1251 007320 114000 POINT
1252 007322 000400 400
1253 007324 000100 100
1254 007326 174120 STATSBINCR=20 ITRY GRAPH MODES
1255 007330 114000 POINT
1256 007332 001000 1000
1257 007334 000200 200
1258
  
```

```

1259 007336 120000 GRAPHX
1260 007340 001010 1010
1261 007342 001020 1020
1262 007344 001030 1030
1263 007346 001040 1040
1264 007350 001050 1050
1265
1266 007352 114000 POINT
1267 007354 001000 1000
1268 007356 001200 1200
1269
1270 007360 124000 GRAPHY
1271 007362 001020 1020
1272 007364 001030 1030
1273 007366 001040 1040
1274 007370 001050 1050
1275 007372 001060 1060
1276
1277 007374 160000 DJMP
1278 007376 167214 FILE01160000
1279
1280 .SBTTL PAPER TAPE BOOT
  
```

```

1282
1283
1284
1285      177500
1286      177500
1287
1288
1289      027400 012701 160000
1290      027404 012702 000004
1291      027408 012703 167500
1292      027414 010712
1293      027416 012706 000024
1294      027422 014304
1295      027424 005714
1296      027426 100775
1297      027430 010712
1298      027432 012706 000024
1299      027436 010441
1300
1301      027440 040601
1302      027442 010111
1303      027444 011102
1304      027446 005214
1305      027450 105714
1306      027452 100376
1307      027454 116412 000002
1308      027460 005211
1309      027462 120227 000375
1310      027466 001366
1311      027470 105222
1312      027472 000142
1313
1314
1315
1316      027474 177500
1317      027476 177500
1318
1319
  
```

```

;
; PAPER TAPE BOOT
;
; HIGH SPEED READER ADDRESS
; LOW SPEED READER ADDRESS
;
; .ORIGIN=1400
;
PTBOOT: MOV #160000,R1 ;SET MEMORY CHECK LIMITS
        MOV #4,R2 ;TRAP ADDRESS IS LOC. 4
        MOV #DEV+41160000,R3 ;POINTER TO DEVICE ADDRESSES
        MOV PC,R2 ;PRESET TRAP ADDRESS IN LOC. 4
        MOV #24,SP ;STACK SET UP AT SPECIAL ADDRESS
        MOV -(R3),R4 ;GET DEVICE ADDRESS
        TST R4 ;CHECK AVAILABILITY OF DEVICE
        BHI DEV1 ;CHECK DEVICE FOR ERRORS
        MOV PC,R2 ;RESET TRAP ADDRESS AT LOC. 4
        MOV #24,SP ;SPECIAL ADDRESSES USED AS MASK LATER
        MOV R4,=(R1) ;DO MEM CHK:READER STATUS ADDRESS
        ;IS MOVED
        BIC SP,R1 ;SET R1=X7792;MASK IN SP=24
        MOV R1,R1 ;STORE OWN ADDRESS IN POINTER
        MOV RR1,R2 ;GET BYTE POINTER
        INC R4 ;ENABLE READER
        TSTB R4 ;TEST DONE BIT
        BPL =2 ;WAIT UNTIL READY
        MOVB 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 -(R3) ;GO EXECUTE AS BRANCH
;
; DEVICE ADDRESSES FOLLOW - DO NOT CHANGE THE ORDER
;
DEVI: LSR ;LOW SPEED READER
      HSR ;HIGH SPEED READER
;
;SBTTL CASSETTE BOOT
  
```

```

1322
1323
1324
1325      177500
1326
1327      027500 012700 177500
1328      027504 005010
1329      027506 010701
1330      027510 062701 000052
1331      027514 012702 000375
1332      027520 112103
1333
1334      027522 112110
1335      027524 100413
1336      027526 130310
1337      027528 001776
1338      027532 105202
1339      027534 100772
1340      027536 116012 000002
1341      027542 120337 000000
1342      027546 001767
1343      027550 000000
1344      027552 000755
1345
1346      027554 005710
1347      027556 100774
1348      027560 005007
1349
1350      027502 017640
1351
1352      027504 002415
1353
1354      027506 112024
1355
1356      027570 000000 000000
1357      027574 167500
1358      027576 000340
1359
1360
1361
  
```

```

;
; CASSETTE BOOT
;
; IACS=177500 ;IACS-11 CONTROL AND STATUS REGISTER
; .ORIGIN=1900
TABOOT: MOV #TACS,R0
        CLR (R0) ;SELECT UNIT #0
        MOV PC,R1 ;USE FOR PIC
        ADD #TABLE+,R1 ;R1 HOLDS ADDR. OF COMMAND TABLE
        MOVB #375,R2 ;MEMORY PTR. AND DATA FLAG
        MOVB (R1)+,R3 ;TEST BITS
;
LOOP1: MOVB (R1)+,(R0) ;COMMAND FROM TABLE TO IACS
        BHI DONE ;WHEN COMMAND CODE NEG., QUIT
        BITB R3,(R0) ;TEST READY AND T-REQ BITS IN IACS
        BEQ LOOP2 ;LOOP 'TIL SOMETHING COMES UP
        INCB R2 ;ADVANCE MEMORY POINTER
        BHI 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 0,K., GO READ ANOTHER BYTE
        STOP: HALT ;HALT ON ERROR
        BR RES ;RESTART ON CONTINUE
;
DONE: TST (R0) ;CHECK FOR ERROR
       BHI STOP ;HALT ON ERROR
       CLR PC ;JUMP #00'
;
TABLE: .WORD 17640 ;.BYTE 240: READY+T-REQ;
        .WORD 2410 ;.BYTE 371: ILS+READY+00
        .WORD 2420 ;.BYTE 15: SFB+00
        .WORD 112024 ;.BYTE 5: READ+00
        .WORD 0,0 ;.BYTE 24: READ+ILS;
        .WORD TABOOT+160000 ;.BYTE 224: READ+ILS+2.0;TABLE
        .WORD 340 ;THESE ARE FILLER WORDS
;
;SBTTL MR13=00 BOOT
  
```

1363
1364
1365
1366
1367 007600 010702
1368 007602 000451
1369 007604 177462
1370 007606 000005
1371
1372 007610 010702
1373 007612 000445
1374 007614 177406
1375 007616 000005
1376
1377
1378 007600 010702
1379 007602 000417
1380 007604 177344
1381 007606 000005
1382 007608 004003
1383 007610 100000
1384 007612 024000
1385
1386
1387 007616 010702
1388 007618 000410
1389 007620 172524
1390 007622 000003
1391 007624 000011
1392 007626 000200
1393 007628 100000
1394
1395
1396 007634 010702
1397 007636 000423
1398 007638 176716
1399
1400
1401 007642 000005
1402 007644 010200
1403 007646 005720
1404 007648 012001
1405 007650 005311
1406 007652 005720
1407 007654 012041
1408 007656 031011
1409 007658 001776
1410 007660 005720
1411 007662 031041
1412 007664 001406
1413 007666 000112
1414
1415
1416 007714 167600

MR11-DB BULK STORAGE PROGRAM LOADER LISTING
i :=ORIGIN+1600 ;KEEP TRACK OF ORIGIN
RF11: MOV PC,R2 ;FIXED HEAD DISK (256 KH)
BR OTHER
177462
5
RK11: MOV PC,R2 ;MOVING HEAD DISK (CARTRIDGE)
BR OTHER
177406
5
TC11: MOV PC,R2
BR TAPES
177344 ;ADDRESS OF WORD COUNT
5 ;LAST COMMAND
4003 ;FIRST COMMAND
100000 ;DONE MARK
24000 ;ERROR MASK
TM11: MOV PC,R2
BR TAPES
172524 ;ADDRESS OF BYTE COUNT
00003 ;LAST COMMAND
00011 ;FIRST COMMAND
200 ;DONE MARK
100000 ;ERROR MASK
RP11: MOV PC,R2 ;MOVING HEAD DISK (PACK)
BR OTHER
176716
TAPES: RESET
MOV R2,R0 ;GET THE ADDRESS OF THE BRANCH
TST (0)+ ;RB TO POINT AT LAST COMMAND
MOV (0)+,R1 ;GET THE WORD COUNT ADDRESS
DEC (1) ;SET UP FOR ADVANCE 1 RECORD
TST (0)+ ;MOVE RB TO FIRST COMMAND
MOV (0)+,(1) ;COMMAND WORD TO COMMAND REG.
BIT (0),(1) ;LOOK FOR DONE INDICATORS
BEQ =E ;NONE SET, TRY AGAIN
TST (0)+ ;DONE FIRST COMMAND, CHECK FOR ERROR
BIT (0),(1) ;LOOK FOR SET ERROR BITS
BEQ OTHER ;NO ERRORS = TRY THE READ
AGAIN: JMP (2) ;RERUN FOR ERRORS
RPVEC: RF11,160000 ;RF11 POWER UP VECTOR

1417 007716 000340
1418
1419 007720 010702
1420 007722 000401
1421 007724 177450
1422
1423
1424 007726 000005
1425 007730 010200
1426 007732 005720
1427 007734 012001
1428 007736 012711 177000
1429 007738 011041
1430 007740 032711 100200
1431 007742 001775
1432 007744 100797
1433 007746 005007
1434
1435 007750 000000
1436 007752 167610
1437 007754 000340
1438 007756 167720
1439 007758 000340
1440 007760 167654
1441 007762 000340
1442 007764 167620
1443 007766 000340
1444
1445
1446

340
RC11: MOV PC,R2 ;FIXED HEAD DISK (64KH)
BR OTHER
177450 ;ADRS OF WORD COUNT (COMMAND+2)
;COMMAND WORD (3) IS THE RESET
OTHER: RESET
MOV R2,R0 ;RB TO POINT AT WORD COUNT ADRS
TST (0)+ ;POINT TO ADDRESS
MOV (0)+,R1 ;WORD COUNT ADDRESS TO R1
MOV #-1000,(1) ;LOAD WORD COUNT
MOV (0),(1) ;COMMAND TO COMMAND REGISTER
BIT #100200,(1) ;CHECK FOR ERROR OR DONE
BEQ =4 ;IF NEITHER, KEEP LOOKING
BNI AGAIN ;ERROR, TRY AGAIN
CLR PC
B ;FILLER
RKVEC: RK11,160000 ;RK POWER UP VECTOR
340
RVVEC: RC11,160000 ;RV 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
;PAGE

1447
 1448
 1449
 1450
 1451
 1452
 1453
 1454
 1455
 1456
 1457
 1458
 1459
 1460
 1461
 1462
 1463
 1464
 1465
 1466
 1467
 1468
 1469
 1470
 1471
 1472
 1473
 1474
 1475
 1476
 1477
 1478
 1479
 1480
 1481
 1482
 1483
 1484
 1485
 1486
 1487
 1488
 1489
 1490
 1491
 1492
 1493
 1494
 1495
 1496
 1497
 1498
 1499
 1500

```

.DSABL AMA
;DATA PATTERN STORED IN THE GT40 BOOTSTRAP VERSION 1
;
;***** THIS IS A IMAGE LISTING OF THE GT40 <V140> BOOTSTRAP *****
;
; THE DATA IS A MIRROR IMAGE OF THAT IN THE BOOTSTRAP .MS
; ONLY THE ADDRESS FIELD IS CHANGED
;BOOTVT.389 5/2/72 <SPECIAL>
;
; VT40 BOOTSTRAP LOADER, VERSION 800, RELEASE R01, 5/2/72
;
; COPYRIGHT 1972, DIGITAL EQUIPMENT CORPORATION,
; 146 MAIN STREET
; MAYNARD, MASSACHUSETTS
; 01754
;
; WRITTEN BY JACK BURNES, SENIOR SYSTEMS ARCHITECT
;
; THIS ROUTINE IS INTENDED TO BE LOADED IN THE ROM PORTION OF THE VT40.
;
; REGISTER DEFINITIONS:
;
;R0=R0
;R1=R1
;R2=R2
;R3=R3
;R4=R4
;R5=R5
;R6=R6
;R7=R7
;
;SP=R6
;PC=R7
;
;RET=R0
;INP1=R1
;INP2=R2
;WORK1=R3
;WORK2=R4
;SCR1=R5
;
;LCKM=WORK1
;LBYT=RET1
;LBC=SCR1
;LADR=INP1
;
;RETURN OF VALUE REGISTER,
;ARGUMENT FOR CALLED FUNCTION
;SECOND ARGUMENT,
;FIRST WORK REGISTER,
;SECOND WORKING REGISTER,
;SCRATCH REGISTER,
;
;OVERLAPPING DEFINITIONS FOR LOADER PORTION.
  
```

1501
 1502
 1503
 1504
 1505
 1506
 1507
 1508
 1509
 1510
 1511
 1512
 1513
 1514
 1515
 1516
 1517
 1518
 1519
 1520
 1521
 1522
 1523
 1524
 1525
 1526
 1527
 1528
 1529
 1530
 1531
 1532
 1533
 1534
 1535
 1536
 1537
 1538
 1539
 1540
 1541
 1542
 1543
 1544
 1545
 1546
 1547
 1548
 1549
 1550
 1551
 1552
 1553
 1554

```

;C0REND=36000
;ROMORG=166000
;
;STARTX=0
;STARTY=1360
;
;VT40=CR172000-150000
;KBDIS=27560
;P1000=25614
;P1010=25610
;
;KBDIS=KBDIS+2
;P1010=P1010+2
;P1000=P1000+2
;
;P1000=COREND-2+10000
;P1010=P1000+4
;STKRT=P1010-2+30000
;
;JMPDIS=16000
;
;PWRFL=24
;
;.16000
;.ROMORG
;
;SET THE ORIGIN NOW!!!
;
;STARTAI MOV #PWRFL+2,SCR1
;CLR #SCR1
;MOV PC,#(SCR1)
;
;RESEY
;
;PICK UP POINTER TO P.F. STATUS.
;CLEAR IT OUT TO BE SURE.
;SET UP THE RESTART LOCATION.
;
;RESET THE BUS.
  
```

```

1855
1856 010052 012767 000007 007570      MOV    #7,#01015      INITIALIZE PDP-10 INPUT
1857 010080 012767 000001 011532      MOV    #1,#00100      INITIALIZE TTY INPUT;
1858 010086 012767 000201 007500      MOV    #2017P100S     INITIALIZE PDP-10 OUTPUT;
1859
1860
1861
1862 010034 012706 019770      RSTRTI MOV    #STKRT:SP      ISET UP THE STACK NOW!
1863 010040 005001          CLR    LADR              ICLEAR ADDRESS POINTER;
1864 010042 012702 160000      MOV    #JMPDIS,INP2     IPLACE A DISPLAY JUMP INSTRUCTION IN A REGISTER,
1865 010046 010221          MOV    INR#;(LADR)      IMOVE IT TO LOCATION 0;
1866 010050 012711 164756      MOV    #DISPRG+15000,(LADR) IMOVE ADDRESS POINTER INTO 2,
1867 010054 012701 000030      MOV    #PWFAL+4,LADR    ISET UP WHERE WE WILL STORE CHARACTERS,
1868 010060 005000          CLR    RET;             IPREPARE TO INSERT A ZERO CHARACTER,
1869 010062 004787 000022      JSR    PC;DOCHAR        IINSERT IT NOW;
1870 010066 005007          CLR    VT40PC          ICLEAR THE DISPLAY PROGRAM COUNTER AND START,
1871
1872 010072 004787 000210      MAJORI JSR    PC;GTCHR      IGET A CHARACTER NOW;
1873 010076 000240      NOP
1874 010080 000240      NOP
1875 010102 000240      NOP
1876 010104 012746 166072      MOV    #MAJOR+5000,-(SP) IINSERT IN DISPLAY BUFFER NOW;
1877
1878 010110 010105      DOCHARI MOV    LADR;SCR1        IGET CURRENT BUFFER POSITION NOW,
1879 010112 022529      CMP    (SCR1)+,(SCR1)+ IBYPASS CURRENT DISPLAY JUMP,
1880 010114 005025          CLR    (SCR1)+         ICLEAR FUTURE ADDRESS FOR JUMP,
1881 010116 010225      MOV    INR#;(SCR1)+    ISTICK IN TEMPORARY JUMP WHILE WE REPLACE CURREN
1882 010120 005015          CLR    (SCR1)         IA DISPLAY JUMP TO ZERO,
1883 010122 005011          CLR    (LADR)         INOW REPLACE CURRENT DISPLAY JUMP BY THE CHARACT
1884 010124 050021      BIS    RET;+(LADR)+    ITIS DONE THIS WAY TO WASTE 2 CYCLES;
1885 010126 010211      MOV    INR#+(LADR)     ITO AVOID TIMING PROBLEMS WITH THE VT40,
1886 010130 000207          RTS    PC              IAND FINALLY RETURN;
1887
1888
1889
1890
1891
1892
1893
1894
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 010132 004787 000124      GT8I   JSR    PC;GT8IX      IGET SIX BITS NOW,
1924 010136 010046      MOV    RET;,-(SP)      ISAVE THE CHARACTER NOW,
1925 010140 000401      BR     GT8I4           IBYPASS THE BITR
1926 010142 005002      GT8I4 CLR    INR#        IRESET THE MAGIC REGISTER NOW;
1927 010144 005722      GT8I4 TST    (INP2)+    IINCREMENT WHERE TO GO;
1928 010146 066207 166250      ADD    GT8I8+15000(INP2),PC IUPDATE PC NOW;

```

```

1929
1930 010152
1931
1932 010152 004787 000104      GT8P8. JSR    PC;GT8IX      IGET A CHARACTER NOW;
1933 010156 010004      MOV    RET;+WORK2     ISAVE FOR A SECOND,
1934 010160 006300      ASL    RET;           ISHIFT TO LEFT OF BYTE
1935 010162 006300      ASL    RET;           ISHIFT TO LEFT OF BYTE
1936 010164 106300      ASLB   RET;           IPACK THEM IN,
1937 010166 106116      ROLB   #SP            IA GOOD 8 BIT THING;
1938 010170 106300      ASLB   RET;           IPOP AND RETURN NOW.
1939 010172 106116      ROLB   #SP
1940 010174 012600      MOV    (SP)+,RET;
1941 010176 000207      RTS    PC
1942
1943
1944
1945
1946
1947
1948 010200 006300      GT82I ASL    RET;           INORST CASE, SHIFT 4
1949 010202 006300      ASL    RET;
1950 010204 106300      ASLB   RET;
1951 010206 106104      ROLB   WORK2
1952 010210 106300      ASLB   RET;
1953 010212 106104      ROLB   WORK2
1954 010214 106300      ASLB   RET;
1955 010216 106104      ROLB   WORK2
1956 010220 106300      ASLB   RET;
1957 010222 106104      ROLB   WORK2
1958 010224 010400      MOV    WORK2,RET;
1959 010226 012604      MOV    (SP)+,WORK2
1960 010230 000207      RTS    PC
1961
1962
1963
1964
1965
1966
1967
1968 010232 006100      GT83I ROL    RET;           IFINAL CHARACTER ASSEMBLED;
1969 010234 006100      ROL    RET;           IFUDGE STACK,
1970 010236 006004      ROR    WORK2          IAND RETURN NOW,
1971 010240 106000      ROR    RET;
1972 010242 006004      ROR    WORK2
1973 010244 106000      ROR    RET;
1974 010246 005726      TST    (SP)+
1975 010250 000207      RTS    PC
1976
1977
1978
1979
1980
1981 010250 016250      GT8TB =    #=2          IPUSH ZERO CONDITION BACK INTO NEVER-NEVER LAND.
1982
1983
1984
1985 010252 000000      #WORD  GT8I=GT8P
1986 010254 000026      #WORD  GT8I=GT8P
1987 010256 000060      #WORD  GT8I=GT8P
1988 010260 177770      #WORD  GT8I=GT8P
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003 010282 004787 000020      GT5IXI JSR    PC;GTCHR        IGET SIX BITS NOW,
2004 010286 020027 000040      CMP    RET;#40
2005 010290 002546      BLY   LBA0
2006 010294 020027 000137      CMP    RET;#137
2007 010300 003143      BGY   LBA0
2008 010302 000207      RTS    PC

```

1663	01E304	005726	GTCHP1	TST	(SP)0	UPDATE THE STACK,
1664						
1665	01E306	012700	GTCHR1	MOV	#P1000,RET1	SET UP POINTER TO THE INPUT CHARACTER.
1666	01E312	004767	GTCHL1	JSR	PC,CHECK	
1667	01E316	005710		TST	0RE11	ANY CHARACTERS THERE?
1668	01E320	001774		BEQ	GTCHL	
1669	01E322	011046		MOV	0RE11,=1SP	PUSH THE CHAR ON THE STACK,
1670	01E324	005020		CLR	(RE11)0	CLEAR THE CHAR GOT TO NOW.
1671	01E326	042716		BIC	#200,(SP)	CLEAR AWAY PARITY NO.
1672	01E332	001764		BEQ	GTCHP	IF ZERO, GET ANOTHER
1673	01E334	022716		CMP	#177,(0P)	ALSO IGNORE SUBROUTS,
1674	01E340	001761		BEQ	GTCHP	WAS IT A "1,5"
1675	01E342	022710		CMP	#177,0RE11	NOPE.
1676	01E346	001007		MOV	(SP)0RE11	YEP, RESET IN CASE OF ABORT.
1677	01E350	011610		CMP	0RE11,#22	IS IT AN R
1678	01E352	021027		BEQ	RSTR1	YEP, RESTART
1679	01E356	001626		CMP	0RE11,#14	IS IT AN L
1680	01E360	021027		BEQ	LOAD	YEP, LOAD.
1681	01E364	001455				
1682						
1683	01E386	011610	GTNPI	MOV	(SP)0RE11	DO THE FOCUSING.
1684	01E370	012600		MOV	(SP)0,RET1	
1685	01E372	020027		CMP	RET1,#175	
1686	01E376	001743		BEQ	GTCHP	IF ALTMODE, LOOP
1687	01E400	000207		RTS	PC	
1688						
1689						
1690						
1691						
1692						
1693						
1694						
1695						
1696	01E402	005767	CHECK1	TST	P100C	DO WE WANT TO OUTPUT?
1697	01E406	001410		BEQ	CHECK1	NO.
1698	01E410	105767		TSTB	P100S	WE DO, IS THE 10 READY?
1699	01E414	100005		BPL	CHECK1	NOT QUITE.
1700	01E416	016767		MOV	P100C,P100B	IT'S READY, SEND THE CHARACTER,
1701	01E424	005067		CLR	P100C	AND THE SAVED CHARACTER.
1702						
1703	01E430	105767	CHECK11	TSTB	KBD1S	KEY, IS THE KEYBOARD READY?
1704	01E434	100014		BPL	CHECK13	NOPE, NO LUCK.
1705	01E436	116746		MOVB	KBD1S,=1SP	YEP, SAVE THE CHARACTER NOW.
1706	01E442	012767		MOV	#1,KBD1S	AND REENABLE THE COMMUNICATIONS DEVICE.
1707						
1708	01E450	004767	CHECK21	JSR	PC,CHECK	IS THE OUTPUT READY?
1709	01E454	005767		TST	P100C	
1710	01E460	001373		BNE	CHECK2	IF NOT, WAIT TILL DONE.
1711	01E402	012667		MOV	(SP)0,P100B	AND THEN SEND OUT THE CHARACTER.
1712						
1713						
1714	01E466	105767	CHECK31	TSTB	P101S	IS THE 10 TALKING TO ME?
1715	01E472	100014		BPL	CHECK4	NOPE, EXIT.
1716	01E474	116767		MOVB	P101S,P101C	GET THE CHARACTER NOW.

1717	01E502	002767	01S	#400,P101C	MAKE SURE IT'S NONE ZERO,	
1718	01E510	012767	MOV	#7,P101S	REINITIALISE COMMUNICATION LINE.	
1719						
1720	01E516	000207	CHECK41	RTS	AND RETURN.	
1721						
1722						
1723						
1724						
1725						
1726						
1727						
1728						
1729						
1730						
1731						
1732						
1733						
1734						
1735						
1736						
1737						
1738						
1739	01E520	005002	LOAD1	CLR	INP	RESET TO FIRST 8-BIT CHARACTER.
1740	01E522	012712		MOV	#17000,(INP2)	AND ALSO CLEVERLY STOP THE VT48.
1741	01E526	012706		MOV	#STRRT,SP	RESET STACK POINTER NOW.
1742						
1743	01E532	005003	LL021	CLR	LCHKH	CLEAR THE CHECKSUM
1744	01E534	004767		JSR	PC,LPTN	GET A BYTE NOW.
1745	01E536	105306		DECB	LBV1	IS IT ONE?
1746	01E542	001373		BNE	LL02	NOPE, WAIT ANHILE
1747	01E544	004767		JSR	PC,LPTN	YEP, GET NEXT CHARACTER.
1748						
1749	01E550	004767		JSR	PC,LOADR	GET A WORD.
1750	01E554	010005		MOV	LBV1,LCB	GET THE COUNTER NOW.
1751	01E556	102705		SUB	#4,LCB	CHOP OFF EXTRA STUFF.
1752	01E502	022705		CMP	#2,LCB	NULL?
1753	01E506	001437		BEQ	LJMP	YEP, MUST BE END.
1754	01E570	004767		JSR	PC,LOADR	NOPE, GET THE ADDRESS.
1755	01E574	010001		MOV	LBV1,LOADR	AND REMEMBER FOR OLD TIMES SAKE.
1756						
1757	01E576	004767	LL031	JSR	PC,LPTN	GET A BYTE (DATA)
1758	01E602	002010		BGE	LL03	ALL DONE WITH THE COUNTER?
1759	01E604	105703		TSTB	LCHKH	YEP, GOOD CHECK SUM?
1760	01E606	001701		BEQ	LL03	NOPE, LOAD ERROR.
1761						
1762	01E610	012700	LOAD1	MOV	(PC)0,RET1	SEND OUT ONE CHARACTER NOW.
1763						
1764	01E612	102		.BYTE	175,102	"CTRL BAD"
1765	01E614	004767		.BYTE	102,175	"BAD CTRL"
1766	01E620	000167		JSR	PC,ENDIT	
1767				JMP	RSTR1	
1768	01E624	110021	LL041	MOVB	LBV1,(LADR)0	PLACE THE BYTE IN CORE.
1769	01E626	000763		BR	LL03	GET ANOTHER ONE.
1770						

1771 016630 004767 177276
 1772 016634 060003
 1773 016636 042700 177400
 1774 016642 005305
 1775 016644 000207
 1776
 1777 016646 004767 177756
 1778 016652 010046
 1779 016654 004767 177750
 1780 016660 000300
 1781 016662 052600
 1782 016664 000207
 1783
 1784 016666 004767 177754
 1785 016672 010046
 1786 016674 004767 177730
 1787 016700 105703
 1788 016702 001342
 1789 016704 032716 000001
 1790 016710 001406
 1791 016712 012700
 1792
 1793 016714 107 175
 1794 016716 004767 000006
 1795 016722 000000
 1796 016724 000776
 1797
 1798 016726 000136
 1799
 1800
 1801
 1802
 1803
 1804
 1805
 1806
 1807
 1808
 1809
 1810
 1811
 1812
 1813
 1814
 1815
 1816 016730 004767 177446
 1817 016734 005767 027036
 1818 016740 001373
 1819 016742 010067 006600
 1820 016744 105000
 1821 016750 000300
 1822 016752 001366
 1823 016754 000207
 1824

LPTR1 JSR PC,5TB ;GET 8 BITS NOW,
 ADD LBYPFLCKSM ;UPDATE CHECKSUM
 BIC #177400,LBYP ;CLEAN UP THE BYTE NOW,
 DEC LBC ;UPDATE THE COUNTER,
 RTS PC ;RETURN NOW,
 LGHRD JSR PC,LPTR ;GET A CHARACTER,
 MOV LBYP=(SP) ;SAVE FOR A SECOND,
 JSR PC,LPTR ;GET ANOTHER CHARACTER,
 SWAB LBYP ;NOW ASSEMBLE THE WORD,
 BIS (SP)+,LBYP ;AND RETURN WITH A 16 BITER,
 RTS PC
 LJMP1 JSR PC,LGHRD ;GET A WORD
 MOV LBYP=(SP) ;SAVE ON THE STACK,
 JSR PC,LPTR ;GET A CHARACTER,
 TSTB LCKSM ;IS IT ZERO?
 BNE LBA0 ;YEP, WHAT CRAP,
 BIT #1,(SP) ;IS IT ODD?
 BEQ LJMP1 ;YEP, START PROGRAM GOING NOW,
 MOV (PC)+,RET1 ;TELL RDP=10 WE'VE LOADED OK,
 .BYTE 175,107 ;"CTRL GOOD"
 .BYTE 107,175 ;"GOOD CTRL"
 JSR PC,SENDIT
 MALT
 BR 1=2
 LJMP1 JMP 0(SP)+ ;AND AWAY WE GO.

SENDIT JSR PC,CHECK ;POLL THE OUTPUT DEVICE NOW,
 TST P100C ;OUTPUT CLEAR?
 BNE SENDIT ;NOPE, LOOP ANHILE 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

1825
 1826
 1827
 1828
 1829
 1830
 1831
 1832
 1833
 1834
 1835
 1836
 1837
 1838
 1839
 1840
 1841
 1842
 1843
 1844
 1845
 1846
 1847
 1848
 1849 016756 170256
 1850 016760 115124
 1851 016762 000000
 1852 016764 001360
 1853 016766 100000
 1854 016770 160000
 1855 016772 000030
 1856 016774 000000
 1857 016776 000000
 1858
 1859
 1860
 1861
 1862
 1863
 1864
 1865
 1866
 1867
 1868
 1869
 1870
 1871 000001

THIS IS THE INITIALIZING V940 PROGRAM WHICH WILL
 JUMP TO THE PROGRAM AFTER THE POWER FAIL OCCATIONS
 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 000000 ;STARTX
 WORD 001360 ;IX COORDINATE
 WORD 100000 ;IY COORDINATE
 WORD 160000 ;SET CHARACTER MODE,
 WORD 000030 ;THEN JUMP TO THE POWERFAIL LOCATION,
 WORD 000000 ;ITO DISPLAY USERS CHARACTERS.
 WORD 0
 .END

AGAIN = 027712	ALYMOD = 000175	BELL = 000250	BLIMIT = 007000
ALNOFF = 020020	BLKON = 000030	BSTART = 001000	CHAR = 100000
CHARA = 021020	CHECK = 016402	CHECK1 = 010430	CHECK2 = 016450
CHECK3 = 016466	CHECK4 = 016516	COREND = 030000	CORSTR = 000004
COUNTN = 000000	CR = 006206	CRLF = 000013	DEV = 007474
DEV1 = 027422	DISJMP = 160000	DISPLA = 177070	DISPRG = 046756
DISTCP = 173000	DJMP = 160000	DL11IB = 170012	DL11IS = 175010
DL11OB = 175016	DL11OS = 175614	DNOP = 160000	DOCHAR = 010110
DOKE = 027954	DONE0 = 001430	DONE1 = 001002	DSR = 001006
DSTOP = 173400	DUMP = 001016	D2BTYP = 000144	END = 001416
FNCCCR = 026036	ERROR1 = 001430	ERROR2 = 001170	FF = 006206
FFLOOP = 026262	FILCNT = 001012	FILE0 = 007214	FILLER = 001010
GETCHR = 026964	GETDL = 006916	GETDL1 = 000046	GETEXT = 006650
GETSIX = 026630	GETB = 006664	GETBTS = 000700	GETB1 = 000712
GETB2 = 026740	GETB3 = 006772	GETB4 = 000706	GRAPFX = 120000
GRAPHY = 114000	GTBUSE = 006424	GTCHL = 010012	GTCHP = 016304
GTCHR = 016306	GTNP = 016366	GTPE4 = 010144	GTPIX = 016260
GT40PC = 172000	GT40SR = 172002	GTB = 010132	GTB1 = 010132
GT8TB = 016250	GT81 = 016152	GT82 = 000200	GT83 = 016232
GT84 = 016142	HEADER = 006474	HOLD = 000004	WSR = 177500
ICNT = 021014	ICOUNT = 002124	IMAGE = 001004	INCR = 000100
INDEX = 000003	INP1 = 000001	INP2 = 000002	INSERT = 006350
INSRTL = 026406	INSRTX = 006422	INTX = 000000	INT0 = 002000
INT1 = 022200	INT2 = 002400	INT3 = 002600	INT4 = 003000
INT5 = 023200	INT6 = 003400	INT7 = 003900	ITAL0 = 000040
ITAL1 = 010060	JMPADD = 007012	JMPDIS = 160000	KBDIB = 027562
KBDIS = 027960	LADR = 000001	LBAD = 010010	LBC = 000000
LBYT = 000000	LCKSM = 000003	LF = 000300	LFLOOP = 006310
LFOUND = 026330	LFSUB = 006304	LGWRD = 010046	LINE0 = 000004
LINE1 = 000004	LINE2 = 000006	LINE3 = 000007	LJMP = 016664
LJMP1 = 016726	LLO2 = 016532	LLO3 = 010070	LLO4 = 016624
LOAD = 016920	LOADER = 007012	LONGV = 110000	LOOP = 007444
LOOP1 = 027922	LOOP2 = 007526	LPOARK = 000200	LPLITE = 000300
LPFFF = 000100	LPOU = 000140	LPTR = 010030	LSR = 177500
LADR = 000001	LBAD = 007100	LBC = 000002	LBYT = 000000
LCKSM = 000003	LCHWRD = 007126	LHALT = 007200	LJMP = 007146
LJMP1 = 007202	LLO2 = 007022	LLO3 = 007066	LLO4 = 007110
LPTR = 027414	MAJOR = 016072	MAXBX = 017600	MAXBY = 000077
MAXX = 021777	MAXY = 001377	MINSUJ = 000100	MINUSX = 020000
MINUSY = 020000	M7 = 002292	M8 = 000270	M9 = 002301
NORMAL = 026212	NOTHER = 006042	NUMLIN = 000040	NXTCHR = 000132
ORIGIN = 166000	OTWER = 007226	OUTLIT = 000052	O2A = 002146
O2A = 02176	PC = 000007	POINT = 110000	POINTR = 000001
PRESTR = 026024	PRG0 = 010034	PRG0R = 001040	PRG1 = 001006
PRG1A = 021944	PRG1B = 001062	PRG1C = 001000	PRG1D = 001000
PRG2 = 021070	PRMTRS = 001024	PSW = 177770	PTBOOT = 007400
PWFAL = 020024	P10TB = 025012	P10IC = 000772	P10IS = 025014
P10P = 025016	P10OC = 004976	P10OS = 020014	RCVEC = 007764
RC11 = 027720	RELATV = 130000	RES = 007906	RESTR = 000600
RETURN = 021230	RET1 = 000000	RFVEC = 007714	RF11 = 007600
RKVEC = 027760	RK11 = 007610	ROMADD = 001000	ROMORG = 166000
RPVEC = 027770	RP11 = 007654	RSTRY = 010034	R0 = 000000
R1 = 002000	R2 = 000002	R3 = 000003	R4 = 000004
R5 = 002000	R6 = 000006	R7 = 000007	SCAN = 000003

SCOPE = 114400	SCOPEB = 002132	SCOPEC = 000054	SCOPEF = 002126
SCOPEG = 022112	SCR1 = 000000	SENDIT = 010730	SETDUN = 000120
SETLP1 = 026074	SETLP2 = 006110	SETLP3 = 000110	SETUP = 006432
SWORTY = 114000	SP = 000006	SR = 177070	START = 000000
STARTA = 016000	STARTX = 000000	STARTY = 001360	STATSA = 170000
STATSB = 174000	STKPTR = 000000	STKSRT = 000770	STOP = 007500
SWITCH = 022306	SYNON = 000004	TAB = 000222	TASGNT = 000002
TABLE = 027962	TABDOT = 000000	TACS = 177000	TAPES = 007662
TCVEC = 027774	TC11 = 007620	TERM = 001022	TEMPEND = 007776
TM11 = 027630	TPCSR = 177564	TPDBR = 177060	TYPEM = 007712
TYPEMA = 021722	TYPEMB = 001750	TYPEMC = 000030	T1 = 001060
T1A = 021070	T1B = 001136	T2 = 001140	T2A = 001156
T2B = 021402	T2C = 001214	T2D = 001224	T2E = 001234
T3 = 021244	T3A = 001260	T3AA = 001260	T3B = 001274
T3C = 021304	T3D = 001312	T3E = 001322	T3F = 001332
T4 = 021352	T4A = 001366	T4B = 001402	T4E = 001416
VT = 026244	VT40PC = 022000	WORDS = 001002	WORK1 = 000003
WORK2 = 000004	.	.	.

ERRORS DETECTED: 8

REM *

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DDGTB-B-D
PRODUCT NAME:	GT40/GT44 INSTRUCTION TEST II
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 IS A TWO PART LOGIC TEST OF THE ALPHAGRAPHIC TERMINAL.
FOR THIS TEST THE TWO MAINTENANCE SWITCH WILL NOT BE USED.
THIS TEST IS DESIGNED TO TEST ALL FUNCTIONAL REGISTERS AND INTERRUPT
VECTOR IN THE ALPHAGRAPHIC DISPLAY CONTROL.
THIS PROGRAM DOES NOT TYPEOUT OR DISPLAY ANY MESSAGES.
THE PROGRAM WILL ONLY HALT ON AN ERROR.

2. REQUIREMENTS

2.1 EQUIPMENT

GT40 DISPLAY SYSTEM (REF. 71) OR
GT44 DISPLAY SYSTEM

2.2 STORAGE

THIS PROGRAM USED MEMORY LOCATIONS 0014000 <LESS THAN 4K OF MEMORY>.

3. LOADING PROCEDURE

3.1 METHOD

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SWITCH BIT 14 = 1 LOOP ON TEST

4.2 STARTING ADDRESS OR ADDRESSES

200 SUB-TEST 1, COMPLEX LOGIC TEST <DR, NPR AND INTERRUPT>
204 SUB-TEST 2, BASIC VISUAL DISPLAY PATTERNS
 <SELECTED BY SW 00002>

0 # POSITIVE HORIZONTAL LINE FROM CENTER SCREEN
1 # NEGATIVE HORIZONTAL LINE FROM CENTER SCREEN
2 # POSITIVE VERTICAL LINE FROM CENTER SCREEN
3 # NEGATIVE VERTICAL LINE FROM CENTER SCREEN
4 # RECTANGLE AROUND SCREEN EDGE
5 # OCTAGON PATTERN IN RELATIVE POINT AND SHORT VECTOR
6 # CHARACTER SET
7 # LIGHT PEN TEST

5. OPERATING PROCEDURE

NONE. ONCE STARTED BOTH SUB-TESTS WILL RUN IN THEIR NORMAL MANNER WITHOUT OPERATOR INTERVENTION OR SWITCH SELECTION.

6. ERRORS

THE PROGRAM WILL ONLY HALT ON AN ERROR. THE PROGRAM DOES NOT CONTAIN FACILITIES FOR REPORTING MESSAGES OR ERROR CONDITIONS. TO PLACE THE PROGRAM INTO A SCOPE LOOP, REPLACE THE ERROR HALT WITH A NOP, SET SWITCH 14 # 1 AND DEPRESS CONT.

7. RESTRICTIONS

BOTH SUB-TESTS DO NOT USE THE MAINTENANCE SWITCHES. IF VR14 SCOPE, LOCATION "GSYAXS" (LOC. 1012) MUST BE CHANGED TO 1377.

8. MISCELLANEOUS

8.1 EXECUTION TIME

SUB-TEST 1 TAKES APPROXIMATELY 35 SECONDS,
N/A OPERATOR INTERVENTION ONLY.

8.2 DEVICE ADDRESS PROGRAM LOCATIONS

LOCATION 1000 CONTAINS THE GT40/GT44 DEVICE ADDRESS
LOCATION 1002 CONTAINS THE GT40/GT44 INTERRUPT VECTOR,
LOCATION 1004 CONTAINS THE GT40/GT44 INTERRUPT LEVEL,
LOCATION 1006 CONTAINS THE GT40/GT44 CHARACTER SIZE,
LOCATION 1010 CONTAINS THE GT40/GT44 LINE FEED SIZE,
LOCATION 1012 CONTAINS THE GT40/GT44 *Y AXIS CUTOFF LOCATION;
(LOC. 1012 = 1377 IF VR14 SCOPE)
(LOC. 1012 = 1777 IF VR17 SCOPE)

9. PROGRAM DESCRIPTION

9.1 SUBTEST 1

THIS SUBTEST IS A COMPLEX TEST OF THE DISPLAY STATUS, X AXIS AND Y AXIS REGISTERS, THE PROGRAM ALSO TESTS STOP<DONE>, LIGHT-PEN, TIME-OUT AND SHIFT-OUT INTERRUPTS AND VECTORS, ALSO INCLUDED ARE TESTS FOR MODE, LINE-TYPE, BLINK, INTENSITY LEVELS, ITALICS AND COLOR CHANGE, THE 'RESUME' <STEP> INSTRUCTION IS USED TO SINGLE STEP THRU THE DISPLAY FILE, ALL DISPLAY INSTRUCTIONS ARE TESTED FOR PROPER OPERATION, TESTS ARE ALSO MADE FOR SETTING OF THE 'EDGE' FLAG, WHEN EXCEEDING ALL FOUR DISPLAY EDGES, TESTS ARE ALSO MADE THAT INULL, 'CR', 'LF' AND 'BS' CHANGE X OR Y AXIS CORRECTLY.

9.2 SUBTEST 2

THIS SUBTEST CONSISTS OF SEVERAL BASIC VISUAL DISPLAY PATTERNS TO AID IN THE REPAIR AND ALIGNMENT OF THE GT-40 TERMINAL, ONCE A PATTERN HAS BEEN SELECTED BY SW 00-02, THE PROGRAM MUST BE RESTARTED TO SELECT ANOTHER PATTERN.

```

ENABL ABS,AMA
TITLE GT=40/GT=44 INSTRUCTION TEST II MAINDEC=11=DDGTB=B
LIST ME,WIN,SEQ
NLIST MC,PO,CND

256
257
258
274
275          000000          ,#0
276 001000 000000          MALT
277 001002 000000          MALT
278          000000          JG THRU 776 IS FILLED WITH A TRAP CATCHER
285          000024          ,#24
286 001024 012364          LOWPHR
287 001026 000348          348
288          000030          ,#30
289 001030 012328          ,WORD SCOPEA      JEMT RETURN
290 001032 000348          348
291
292          000200          ,#200
293 001200 000137 001336          JMP      START
294 001204 000137 012464          JMP      START1
295
296          001000          ,#1000
297 001000 172000          OSADD; 172000          ;OS DISPLAY STARTING ADDRESS
298 001002 000320          OSVCT; 320           ;OS DISPLAY STARTING VECTOR
299 001004 000200          OSBRI; 200          ;OS DISPLAY INTERRUPT LEVEL
300 001006 000016          OSCHER; 16          ;CHARACTER SIZE (14=16)
301 001010 000030          OSLPER; 30          ;LINE FEED SIZE (30=32)
302 001012 001777          OSYXSF; 1777       ;Y AXIS CUTOFF LOCATION
303 001014 000177          OSSEND; 177       ;SHIFT-OUT END CHARACTER
304
305 001016 000000          ICNT; 0             ;PASS COUNTER
306 001020 177776          PSW; 177776        ;
307 001022 013444          OSBFI; BUFFER       ;FIRST WORD IN THE DISPLAY BUFFER
308 001024 013466          OSBFI; BUFFER+2     ;SECOND WORD
309 001026 013478          OSBFI; BUFFER+4     ;THIRD WORD
310 001030 013472          OSBFI; BUFFER+6     ;FOURTH WORD
311 001032 013474          OSBFI; BUFFER+8     ;FIFTH WORD
312 001034 013476          OSBFI; BUFFER+10    ;SIX WORD
313 001036 000000          OSAVE; 0            ;TEMP REG.
314 001040 177700          SWR; 177700        ;
315 001042 017476          SIZE; 17476        ;BUFFER SIZE FOR 4K <WORD LENGTH>
316 001044 000000          CNTR; 0             ;
317 001046 000700          LPSIZE; 700        ;LINE FEED DELTA Y SIZE
318 001048 000762          CHSIZE; 762        ;BACK SPACE CHARACTER DELTA X SIZE
    
```

```

323
327
332          ;OS ADDRESSES AND VECTORS
333
334 001002 172000          DPC; 172000         ;OS DISPLAY PC REGISTER
335 001004 172002          OSRI; 172002        ;OS DISPLAY STATUS REGISTER
336 001006 172004          XPOS; 172004       ;X AXIS REGISTER <READ ONLY>
337 001008 172006          YPOS; 172006       ;Y AXIS REGISTER AND GRAPH/LOT REGISTER <READ ONLY>
338
339 001002 000320          DDONE; 320         ;OS DISPLAY STOP <DONE> VECTOR
340 001004 000322          DDONE; 322         ;
341
342 001006 000324          LPVCT; 324         ;OS DISPLAY LIGHT PEN VECTOR
343 001008 000326          LPVCT; 326         ;
344
345 001072 000330          TIMEVT; 330        ;OS DISPLAY TIME-OUT <NXH> ERROR VECTOR
346 001074 000332          TIMEVT; 332        ;OR "SHIFT-OUT" VECTOR
347
348          ;OS INSTALLATION ROUTINE
349
350 001076 012700 001052          SETUP; MOV  #0,RO      ;SET UP POINTER
351 001102 013701 001000          SETUP; MOV  OSADD,R0
352 001106 010120          SETUP; MOV  R1,(R0)+
353 001110 002701 000002          SETUP; ADD  #2,R1
354 001114 022700 001062          SETUP; CMP  #0,RO+1,RO
355 001120 001372          SETUP; BNE  SETUPA
356 001122 012700 001062          SETUP; MOV  #DDONE,R0
357 001126 013701 001002          SETUP; MOV  OSVCT,R1
358 001132 010120          SETUP; MOV  R1,(R0)+
359 001134 002701 000002          SETUP; ADD  #2,R1
360 001140 022700 001076          SETUP; CMP  #DDONE+4,R0
361 001144 001372          SETUP; BNE  SETUPB
362 001146 013737 001010 001046          SETUP; MOV  OSLESS,LPSIZE ;SET UP DELTA LF
363 001154 005437 001046          SETUP; NEG  LPSIZE      ;NEGATE IT
364 001160 042737 177000 001046          SETUP; BIC  #177000,LPSIZE ;MASK IT
365 001166 013737 001006 001050          SETUP; MOV  OSCHER,CHSIZE ;SET UP DELTA CHAR
366 001174 005437 001050          SETUP; NEG  CHSIZE     ;NEGATE IT
367 001200 004737 001252          SETUP; JSR  PC,CORE     ;SET UP CORE SIZE
368 001204 042737 177000 001050          SETUP; BIC  #177000,CHSIZE ;MASK IT
369 001212 013777 001064 177642          SETUP; MOV  DDONE1,DDONE ;LOAD DONE VECTOR
370 001220 005077 177640          SETUP; CLR  DDONE1
371 001224 013777 001070 177634          SETUP; MOV  LPVCT1,LPVCT ;LOAD LIGHT-PEN VECTOR
372 001232 005077 177632          SETUP; CLR  LPVCT1
373 001236 013777 001074 177626          SETUP; MOV  TIMEVT1,TIMEVT ;LOAD TIME-OUT VECTOR
374 001244 005077 177624          SETUP; CLR  TIMEVT1
375 001250 000207          SETUP; RTS
    
```

```

377          ;SUBROUTINE TO DETERMINE THE SIZE OF CORE
378          ; AND SET UP LOCATION SIZE WITH THE VALUE
379
380 001272 012737 001304 000004 DD CORE1 MOV #25,004 ;SET UP FOR NEM
381 001260 012701 017776          MOV #17776,R1 ;SET UP ADDRESS
382 001264 000000          CLR R2
383 001266 062701 020000          IST ADD #20000,R1 ;MOVE TO THE NEXT BANK
384 001272 000200          INC R2 ;INC BANK COUNTER
385 001274 000011          TST (1) ;TIMEOUT ?
386 001276 022701 177776          CMP #17776,R1 ;END ?
387 001302 001371          BNE IS
388 001304 000300          IS DEC R2
389 001306 012737 000004 000004 DD CORE1 MOV #6,004 ;DECREMENT BANK COUNT
390 001314 022626          CMP (SP)*,(SP)* ;RESET BUSS ERROR
391 001316 162701 020000          SUB #20000,R1 ;PROP THE STACK X2
392 001322 010137 001042          MOV R1,010 ;LOAD SIZE
393 001326 162737 007776 001042 DD CORE1 SUB #7776,010C ;BYPASS LOADER
394 001334 000207          RTS PC ;EXIT
395

```

```

397
398 001336 012777 000340 177454 START: MOV #340,003H
399 001344 012704 000000          MOV #0,000H ;SP
400 001350 004737 001074          JBR PC,RETURN
401 001354 000037 001016          CLR ION
402 001360 012701 001366          MOV #0,000+2,R2
403
404          ;TEST FOR BUSS ERRORS ON DISPLAY ADDRESSES
405
406 001364 104000          OTBUSS: SCOPE
407 001366 000000          RESET
408 001370 000077 177460          CLR #000 ; ON DISPLAY STATUS
409 001374 000240          NOP
410 001376 000077 177454          CLR #XRRS ; ON DISPLAY X REGISTER
411 001402 000240          NOP
412 001404 000077 177450          CLR #YRRS ; ON DISPLAY Y REGISTERS
413 001410 000000          RESET
414
415          ;INCREMENT P.C. TEST
416          ;COMPLEX = BUFFER LENGTH
417
418 001412 104000          OTRC: SCOPE
419 001414 013702 001022          MOV DBUFFR2
420 001420 012022 170000          IST MOV #170000,(2)* ;SET UP POINTER
421 001404 023702 001042          CMP $1000R2 ;MOVE OSTOP INTO THE BUFFER
422 001430 001373          BNE INO ;FINISHED FILLING THE BUFFER
423
424 001402 104000          SCOPE
425 001404 013777 001022 177410          MOV DBUFFR0PC ;YES, START THE DISPLAY
426 001442 013737 001022 001036          MOV DBUFFR0SAVE
427 001450 013702 001042          MOV $1000R2 ;SETUP A COUNT
428 001494 000300          DEC R2
429 001496 017704 177372          OTRCA: MOV #0,000R4
430 001402 100402          BMI IS
431 001404 000000          HALT
432 001406 000421          BR OTS ;ERROR, STOP FLAG FAILED TO SET
433
434 001470 062737 000002 001036 IST ADD #2,000SAVE
435 001476 017700 177350          MOV #0,000R0 ;READ DISPLAY P.C.
436 001502 023700 001036          CMP 000SAVE,R0 ;DID IT INCREMENT BY 2?
437 001506 001402          BEQ IS ;YES
438 001510 000000          HALT ;DISPLAY PC FAILED TO INCREMENT
439 001512 000407          BR OTS ;PROPERLY
440
441 001514 020037 001036          IST CMP R0,000SAVE ;FINISHED THE BUFFER ?
442 001516 001404          BEQ OTS ;OR IF YES
443 001522 012777 000001 177322          MOV #1,000PC ;SINGLE STEP THE DISPLAY
444 001530 000792          BR OTRCA ;TRY AGAIN
445

```

```

447                                     iTSTED BY "LOAD STATUS REGISTER A"
448
449 001532 104000                                GT01 SCOPE
450 001534 012777 172040 177260             MOV #172040,0DBUF ;ITALICS ENABLE=1 ITALICS=0
451 001542 013777 001022 177302             MOV DBUF,0DFC ;LOAD DISPLAY P,C,
452 001550 017700 177300             MOV 0DSR,R0 ;READ DISPLAY STATUS REGISTER
453 001554 042700 177757             BIC #177757,R0 ;MASK TO BIT 4
454 001560 022700 000000             CMP #0,R0 ;TEST R0
455 001564 001401                               BEQ ;=4
456 001566 000000                               HALT ;ITALICS BIT FAILED TO RESET
457
458 001570 104000                                GT01 SCOPE
459 001572 012777 172060 177222             MOV #172060,0DBUF ;ITALICS ENABLE=1 ITALICS=1
460 001600 013777 001022 177244             MOV DBUF,0DFC ;LOAD DISPLAY P,C,
461 001606 017700 177242             MOV 0DSR,R0 ;READY DISPLAY STATUS REGISTER
462 001612 042700 177757             BIC #177757,R0 ;MASK TO BIT 4
463 001616 022700 000020             CMP #20,R0 ;TEST R0
464 001622 001401                               BEQ ;=4
465 001624 000000                               HALT ;ITALICS BIT FAILED TO SET
466
467 001626 104000                                GT21 SCOPE
468 001630 012777 172000 177164             MOV #172000,0DBUF ;ITALICS ENABLE=0 ITALICS=0
469 001636 013777 001022 177206             MOV DBUF,0DFC ;LOAD DISPLAY P,C,
470 001644 017700 177204             MOV 0DSR,R0 ;READ DISPLAY STATUS REGISTER
471 001650 042700 177757             BIC #177757,R0 ;MASK TO BITS 4
472 001654 022700 000020             CMP #20,R0 ;TEST R0
473 001660 001401                               BEQ ;=4
474 001662 000000                               HALT ;ITALICS ENABLE FAILED TO INHIBIT
475                                     ;CLEARING OF ITALICS BIT
476
477 001664 104000                                GT31 SCOPE
478 001666 012777 172002 177126             MOV #172002,0DBUF ;COLOR ENABLE=1 COLOR=0
479 001674 013777 001022 177150             MOV DBUF,0DFC ;LOAD DISPLAY P,C,
480 001702 004737 012352             JSR 7,0LAY1 ;EXECUTE A PROGRAM DELAY
481                                     (1)
482 001706 017700 177142             MOV 0DSR,R0 ;READ DISPLAY STATUS REGISTER
483 001712 042700 177773             BIC #177773,R0 ;MASK TO BIT 2
484 001716 022700 000000             CMP #0,R0 ;TEST R0
485 001722 001401                               BEQ ;=4
486 001724 000240                               NOP ;COLOR BIT FAILED TO RESET
487
488 001726 104000                                GT41 SCOPE
489 001730 012777 172003 177064             MOV #172003,0DBUF ;COLOR ENABLE=1 COLOR=1
490 001736 013777 001022 177106             MOV DBUF,0DFC ;LOAD DISPLAY P,C,
491 001744 004737 012352             JSR 7,0LAY1 ;EXECUTE A PROGRAM DELAY
492                                     (1)
493 001750 017700 177100             MOV 0DSR,R0 ;READ DISPLAY STATUS REGISTER
494 001754 042700 177773             BIC #177773,R0 ;MASK TO BIT 2
495 001760 022700 000004             CMP #4,R0 ;TEST R0
496 001766 001401                               BEQ ;=4
497 001768 000240                               NOP ;COLOR BIT FAILED TO SET
    
```

```

498
499 001770 104000                                GT51 SCOPE
500 001772 012777 172000 177022             MOV #172000,0DBUF ;COLOR ENABLE=0 COLOR=0
501 001800 013777 001022 177044             MOV DBUF,0DFC ;LOAD DISPLAY P,C,
502 001806 017700 177042             MOV 0DSR,R0 ;READ DISPLAY STATUS REGISTER
503 001812 042700 177773             BIC #177773,R0 ;MASK TO BIT 2
504 001816 022700 000004             CMP #4,R0 ;TEST R0
505 001822 001401                               BEQ ;=4
506 001824 000240                               NOP ;COLOR ENABLE FAILED TO INHIBIT
507                                     ;RESETTING OF COLOR BIT
508
509                                     iTSTED BY "LOAD CHARACTER MODE"
510
511 001826 104000                                GT61 SCOPE
512 001830 012777 100004 176704             MOV #100004,0DBUF ;LOAD LINE TYPE ENABLE #1 AND LINE TYPE VALUE =0
513 001836 012777 172000 176700             MOV #172000,0DBUF ;LOAD STOP
514 001844 013777 001022 177000             MOV DBUF,0DFC ;LOAD DISPLAY P,C,
515 001852 017700 176776             MOV 0DSR,R0 ;READ DISPLAY STATUS REGISTER
516 001856 042700 177774             BIC #177774,R0 ;MASK TO BITS 1-0
517 001862 022700 000000             CMP #0,R0 ;TEST R0
518 001866 001401                               BEQ ;=4
519 001870 000000                               HALT ;LINE BITS 1-0 FAILED TO RESET
520
521 001872 104000                                GT71 SCOPE
522 001874 012777 100007 176720             MOV #100007,0DBUF ;LINE TYPE ENABLE #1 LINE TYPE #3
523 001882 012777 172000 176714             MOV #172000,0DBUF ;LOAD STOP
524 001890 013777 001022 176734             MOV DBUF,0DFC ;LOAD DISPLAY P,C,
525 001896 017700 176732             MOV 0DSR,R0 ;READ DISPLAY STATUS REGISTER
526 001902 042700 177774             BIC #177774,R0 ;MASK TO BITS 1-0
527 001906 022700 000003             CMP #3,R0 ;TEST R0
528 001912 001401                               BEQ ;=4
529 001914 000000                               HALT ;LINE BITS 1-0 FAILED TO SET
530
531 001916 104000                                GT81 SCOPE
532 001920 012777 100005 176654             MOV #100005,0DBUF ;LINE TYPE ENABLE #1 LINE TYPE #1
533 001926 012777 172000 176650             MOV #172000,0DBUF ;LOAD STOP
534 001934 013777 001022 176670             MOV DBUF,0DFC ;LOAD DISPLAY P,C,
535 001942 017700 176666             MOV 0DSR,R0 ;READ DISPLAY STATUS REGISTER
536 001948 042700 177774             BIC #177774,R0 ;MASK TO BITS 1-0
537 001954 022700 000001             CMP #1,R0 ;TEST R0
538 001960 001401                               BEQ ;=4
539 001962 000000                               HALT ;LINE BIT 0 FAILED TO SET
540
541 001964 104000                                GT91 SCOPE
542 001966 012777 100006 176610             MOV #100006,0DBUF ;LINE TYPE ENABLE #1 LINE TYPE #2
543 001974 012777 172000 176604             MOV #172000,0DBUF ;LOAD STOP
544 001982 013777 001022 176624             MOV DBUF,0DFC ;LOAD DISPLAY P,C,
545 001990 017700 176622             MOV 0DSR,R0 ;READ DISPLAY STATUS REGISTER
546 001996 042700 177774             BIC #177774,R0 ;MASK TO BITS 1-0
547 002002 022700 000002             CMP #2,R0 ;TEST R0
548 002008 001401                               BEQ ;=4
549 002010 000000                               HALT ;LINE BIT 1 FAILED TO SET
550
    
```

552									
553	022246	104000			GT801	SCOPE			
554	022230	012777	100003	176544	MOV	#100003,0DBUF		;LINE TYPE ENABLE #0 LINE TYPE #3	
555	022256	012777	172000	176540	MOV	#172000,0DBUF1			
556	022244	013777	001022	176500	MOV	DBUF#0DFC		;LOAD DISPLAY P.C.	
557	022272	017700	176556		MOV	0DSR,R0		;READ DISPLAY STATUS REGISTER	
558	022276	042700	177774		BIC	#177774,R0		;MASK TO BITS 1-0	
559	022302	022700	000002		CMP	#2,R0		;TEST R0	
560	022306	001401			BEG	.+4		;SHOULD NOT CHANGE LT VALUE	
561	022310	000000			HALT			;LINE TYPE ENABLE FAILED TO INHIBIT	
562								;CHANGING OF LINETYPE VALUE	
563									
564	022312	104000			GT811	SCOPE			
565	022334	012777	100020	176500	MOV	#100020,0DBUF		;BLINK ENABLE #1 BLINK #0	
566	022322	012777	172000	176474	MOV	#172000,0DBUF1			
567	022330	013777	001022	176514	MOV	DBUF#0DFC		;LOAD DISPLAY P.C.	
568	022336	017700	176512		MOV	0DSR,R0		;READ DISPLAY STATUS REGISTER	
569	022342	042700	177767		BIC	#177767,R0		;MASK TO BIT 3	
570	022346	022700	000000		CMP	#0,R0		;TEST R0	
571	022352	001401			BEG	.+4			
572	022354	000000			HALT			;BLINK BIT FAILED TO RESET	
573									
574									
575	022356	104000			GT821	SCOPE			
576	022300	012777	100030	176434	MOV	#100030,0DBUF		;BLINK ENABLE #1 BLINK #1	
577	022306	012777	172000	176430	MOV	#172000,0DBUF1			
578	022374	013777	001022	176490	MOV	DBUF#0DFC		;LOAD DISPLAY P.C.	
579	022402	017700	176446		MOV	0DSR,R0		;READ DISPLAY STATUS REGISTER	
580	022406	042700	177767		BIC	#177767,R0		;MASK TO BIT 3	
581	022412	022700	000010		CMP	#10,R0		;TEST R0	
582	022416	001401			BEG	.+4			
583	022420	000000			HALT			;BLINK BIT FAILED TO SET	
584									
585									
586	022422	104000			GT831	SCOPE			
587	022424	012777	100000	176370	MOV	#100000,0DBUF		;BLINK ENABLE #0 BLINK #0	
588	022432	012777	172000	176364	MOV	#172000,0DBUF1			
589	022440	013777	001022	176404	MOV	DBUF#0DFC		;LOAD DISPLAY P.C.	
590	022446	017700	176402		MOV	0DSR,R0		;READ DISPLAY STATUS REGISTER	
591	022452	042700	177767		BIC	#177767,R0		;MASK TO BIT 3	
592	022456	022700	000010		CMP	#10,R0		;TEST R0	
593	022462	001401			BEG	.+4			
594	022464	000000			HALT			;BLINK ENABLE FAILED TO INHIBIT	
595								;CHANGING OF THE BLINK BIT	
596									
597	022466	104000			GT841	SCOPE			
598	022470	012777	100100	176384	MOV	#100100,0DBUF		;LP ENABLE #1 LP#0	
599	022476	012777	172000	176320	MOV	#172000,0DBUF1			
600	022504	013777	001022	176340	MOV	DBUF#0DFC		;LOAD DISPLAY P.C.	
601	022512	017700	176336		MOV	0DSR,R0		;READ STATUS	
602	022516	032700	000200		BIT	#200,R0			
603	022522	001401			BEG	.+4			
604	022524	000000			HALT			;LIGHT PEN FLAG SET IN ERROR	

606									
607	022536	104000			GT851	SCOPE			
608	022530	012777	100140	176264	MOV	#100140,0DBUF		;LP ENABLE #1 LP#1	
609	022506	012777	172000	176260	MOV	#172000,0DBUF1			
610	022544	013777	001022	176300	MOV	DBUF#0DFC		;LOAD DISPLAY P.C.	
611	022532	017700	176276		MOV	0DSR,R0		;READ STATUS	
612	022536	032700	000200		BIT	#200,R0			
613	022542	001401			BEG	.+4			
614	022544	000000			HALT			;LIGHT PEN FLAG SET IN ERROR	
615									
616	022546	104000			GT861	SCOPE			
617	022570	012777	102000	176224	MOV	#102000,0DBUF		;INTENSITY LEVEL ENABLE #1 LEVEL #0	
618	022576	012777	172000	176220	MOV	#172000,0DBUF1			
619	022604	013777	001022	176240	MOV	DBUF#0DFC		;LOAD DISPLAY P.C.	
620	022612	017700	176236		MOV	0DSR,R0		;READ DISPLAY STATUS REGISTER	
621	022616	042700	174377		BIC	#174377,R0		;MASK TO BITS 0-10	
622	022622	022700	000000		CMP	#0,R0		;TEST R0	
623	022626	001401			BEG	.+4			
624	022630	000000			HALT			;INTENSITY LEVEL BITS 0-10 FAILED TO RESET	
625									
626									
627	022632	104000			GT871	SCOPE			
628	022634	012777	103600	176160	MOV	#103600,0DBUF		;INTENSITY LEVEL ENABLE #1 LEVEL #7	
629	022642	012777	172000	176154	MOV	#172000,0DBUF1			
630	022690	013777	001022	176174	MOV	DBUF#0DFC		;LOAD DISPLAY P.C.	
631	022696	017700	176172		MOV	0DSR,R0		;READ DISPLAY STATUS REGISTER	
632	022662	042700	174377		BIC	#174377,R0		;MASK TO BITS 0-10	
633	022666	022700	003400		CMP	#3400,R0		;TEST R0	
634	022672	001401			BEG	.+4			
635	022674	000000			HALT			;INTENSITY LEVEL BITS 0-10 FAILED TO SET	
636									
637									
638	022676	104000			GT881	SCOPE			
639	022700	012777	103000	176114	MOV	#103000,0DBUF		;INTENSITY LEVEL ENABLE #1 LEVEL #4	
640	022706	012777	172000	176110	MOV	#172000,0DBUF1			
641	022714	013777	001022	176130	MOV	DBUF#0DFC		;LOAD DISPLAY P.C.	
642	022722	017700	176126		MOV	0DSR,R0		;READ DISPLAY STATUS REGISTER	
643	022726	042700	174377		BIC	#174377,R0		;MASK TO BITS 0-10	
644	022730	022700	002000		CMP	#2000,R0		;TEST R0	
645	022736	001401			BEG	.+4			
646	022740	000000			HALT			;INTENSITY LEVEL BIT 10 FAILED	
647									
648									
649	022742	104000			GT891	SCOPE			
650	022744	012777	102400	176050	MOV	#102400,0DBUF		;INTENSITY LEVEL ENABLE #1 LEVEL #2	
651	022752	012777	172000	176044	MOV	#172000,0DBUF1			
652	022760	013777	001022	176064	MOV	DBUF#0DFC		;LOAD DISPLAY P.C.	
653	022766	017700	176062		MOV	0DSR,R0		;READ DISPLAY STATUS REGISTER	
654	022772	042700	174377		BIC	#174377,R0		;MASK TO BITS 0-10	
655	022776	022700	001000		CMP	#1000,R0		;TEST R0	
656	022782	001401			BEG	.+4			
657	022784	000000			HALT			;INTENSITY LEVEL BIT 9 FAILED	

```

659
660 000076 104000
661 000100 012777 102200 176004 GT201 SCOPE
662 000106 012777 172000 176000 MOV #102200,0DBUF ;INTENSITY LEVEL ENABLE =1 LEVEL #1
663 000124 013777 001022 176020 MOV #172000,0DBUF1 ;
664 000302 017700 176016 MOV DBUF,0DPC ;LOAD DISPLAY P.C.
665 000306 042700 174377 MOV #0000,RO ;READ DISPLAY STATUS REGISTER
666 000342 022700 000400 BIC #174377,RO ;MASK TO BITS 0-10
667 000346 001401 CMP #400,RO ;TEST RO
668 000350 000000 BEQ ;+4
669
670
671 000352 104000
672 000354 012777 101600 175740 GT211 SCOPE
673 000362 012777 172000 175734 MOV #101600,0DBUF ;INTENSITY LEVEL ENABLE =0 LEVEL #7
674 000370 013777 001022 175754 MOV #172000,0DBUF1 ;
675 000376 017700 175752 MOV DBUF,0DPC ;LOAD DISPLAY P.C.
676 000402 042700 174377 MOV #0000,RO ;READ DISPLAY STATUS REGISTER
677 000406 022700 000400 BIC #174377,RO ;MASK TO BITS 0-10
678 000412 001401 CMP #400,RO ;TEST RO
679 000414 000000 BEQ ;+4
680
681
682
683
684 000416 104000
685 000420 012777 174100 175674 GT221 SCOPE
686 000426 012777 172000 175670 MOV #174100,0DBUF ;LOAD GRAPHPLOT COUNTER
687 000434 013777 001022 175710 MOV #172000,0DBUF1 ;
688 000442 017700 175710 MOV DBUF,0DPC ;START DISPLAY
689 000446 042700 001777 MOV #0000,RO ;READ INCREMENT REGISTER
690 000452 022700 000000 BIC #1777,RO ;MASK TO BITS 10-18
691 000456 001401 CMP #0,RO
692 000460 000000 BEQ ;+4
693
694 000462 104000
695 000464 012777 174177 175630 GT231 SCOPE
696 000472 012777 172000 175624 MOV #174177,0DBUF ;LOAD GRAPHPLOT COUNTER
697 000480 013777 001022 175644 MOV #172000,0DBUF1 ;
698 000486 017700 175644 MOV DBUF,0DPC ;START DISPLAY
699 000492 042700 001777 MOV #0000,RO ;READ INCREMENT REGISTER
700 000496 022700 174000 BIC #1777,RO ;MASK TO BITS 10-18
701 000502 001401 CMP #17000,RO
702 000506 000000 BEQ ;+4
703
704 000508 104000
705 000510 012777 174152 175564 GT241 SCOPE
706 000516 012777 172000 175560 MOV #174152,0DBUF ;LOAD GRAPHPLOT COUNTER
707 000524 013777 001022 175600 MOV #172000,0DBUF1 ;
708 000532 017700 175600 MOV DBUF,0DPC ;START DISPLAY
709 000536 042700 001777 MOV #0000,RO ;READ INCREMENT REGISTER
710 000542 022700 124000 BIC #1777,RO ;MASK TO BITS 10-18
711 000546 001401 CMP #10000,RO
712 000550 000000 BEQ ;+4

```

```

714
715 000272 104000
716 000274 012777 174125 175500 GT251 SCOPE
717 000302 012777 172000 175514 MOV #174125,0DBUF ;LOAD GRAPHPLOT COUNTER
718 000310 013777 001022 175534 MOV #172000,0DBUF1 ;
719 000316 017700 175534 MOV DBUF,0DPC ;START DISPLAY
720 000322 042700 001777 MOV #0000,RO ;READ INCREMENT REGISTER
721 000326 022700 052000 BIC #1777,RO ;MASK TO BITS 10-18
722 000332 001401 CMP #52000,RO
723 000334 000000 BEQ ;+4
724
725 000336 104000
726 000340 012777 174100 175494 GT261 SCOPE
727 000346 012777 172000 175490 MOV #174100,0DBUF ;LOAD GRAPHPLOT COUNTER WITH 0
728 000354 013777 001022 175470 MOV #172000,0DBUF1 ;
729 000362 004737 012340 MOV DBUF,0DPC ;START DISPLAY
730 000366 012777 174077 175486 JSR 7,DELAY ;EXECUTE A PROGRAM DELAY
731 000374 013777 001022 175480 MOV #174077,0DBUF ;LOAD GRAPHPLOT NO ENABLE
732 000382 017700 175480 MOV DBUF,0DPC ;START DISPLAY
733 000386 042700 001777 MOV #0000,RO ;READ INCREMENT REGISTER
734 000392 022700 000000 BIC #1777,RO ;MASK TO BITS 10-18
735 000396 001401 CMP #0,RO ;ARE THEY EQUAL ?
736 000402 000000 BEQ ;+4
737
738
739
740
741
742 000422 104000
743 000424 012777 122000 175370 GT271 SCOPE
744 000432 012777 001252 175394 MOV #122000,0DBUF ;LOW INTENSITY - SET GRAPHPLOT X MODE
745 000440 012777 172000 175300 MOV #122000,0DBUF1 ;SET X POSITION
746 000446 013777 001022 175376 MOV DBUF,0DPC ;LOAD STOP
747 000454 004737 012340 JSR 7,DELAY ;START THE DISPLAY
748 000460 017700 175372 MOV #0000,RO ;EXECUTE A PROGRAM DELAY
749 000464 022700 001252 MOV #0000,RO ;READ X POSITION
750 000470 001401 CMP #1000,RO
751 000472 000000 BEQ ;+4
752

```

```
754
755
756
757
758 000474 104000
759 000476 012777 122000 175316 G*281 SCOPE
760 000504 012777 000525 175312 MOV #12000,00BUF ;LOW INTENSITY - SET GRAPH PLOT X MODE
761 000532 012777 172000 175306 MOV #500,00BUF1 ;SET X POSITION
762 000520 013777 001022 175324 MOV #17000,00BUF2 ;LOAD STATUS REGISTER A, STOP
763 000526 004737 012340 DBUF,00PC ;LOAD THE DISPLAY P.C.
764 000532 017700 175320 JSR 7,DELAY ;EXECUTE A PROGRAM DELAY
765 000536 022700 000525 MOV #XPOS,R0 ;READ X POSITION
766 000542 001401 CMP #500,R0
767 000544 000000 BEQ ;
768
769
770
771
772
773 000546 104000
774 000550 012777 126000 175244 G*291 SCOPE
775 000556 012777 001252 175240 MOV #10000,00BUF ;LOW INTENSITY - SET GRAPH PLOT Y
776 000504 012777 172000 175234 MOV #500,00BUF1 ;SET Y POSITION
777 000532 013777 001022 175252 MOV #17000,00BUF2 ;LOAD STATUS REGISTER A, STOP
778 000600 004737 012340 DBUF,00PC ;LOAD THE DISPLAY P.C.
779 000604 017700 175250 JSR 7,DELAY ;EXECUTE A PROGRAM DELAY
780 000608 022700 001252 MOV #YPOS,R0 ;READ Y POSITION
781 000614 001401 CMP #1000,R0
782 000616 000000 BEQ ;
783
784
```

```
786
787
788
789
790 000600 104000
791 000602 012777 126000 175172 G*301 SCOPE
792 000600 012777 000525 175166 MOV #10000,00BUF ;LOW INTENSITY - SET GRAPH PLOT Y MODE
793 000606 012777 172000 175102 MOV #500,00BUF1 ;SET Y POSITION
794 000644 013777 001022 175200 MOV #17000,00BUF2 ;LOAD STATUS REGISTER A, STOP
795 000632 004737 012340 DBUF,00PC ;LOAD THE DISPLAY P.C.
796 000636 017700 175176 JSR 7,DELAY ;EXECUTE A PROGRAM DELAY
797 000602 022700 000525 MOV #YPOS,R0 ;READ Y POSITION
798 000606 001401 CMP #500,R0
799 000670 000000 BEQ ;
800
801
802
803
804
805
806 000672 104000
807 000674 012777 122000 175120 G*311 SCOPE
808 000700 012777 001234 175114 MOV #10000,00BUF ;LOW INTENSITY - SET GRAPH PLOT X MODE
809 000710 012777 126000 175110 MOV #500,00BUF1 ;SET X POSITION
810 000716 012777 001432 175104 MOV #10000,00BUF2 ;SET GRAPH PLOT Y MODE
811 000724 012777 172000 175100 MOV #500,00BUF3 ;SET Y POSITION
812 000732 013777 001022 175112 MOV #17000,00BUF4 ;LOAD STATUS REGISTER A, STOP
813 000740 004737 012340 DBUF,00PC ;LOAD THE DISPLAY P.C.
814 000744 017700 175106 JSR 7,DELAY ;EXECUTE A PROGRAM DELAY
815 000730 022700 001234 MOV #XPOS,R0 ;READ X POSITION
816 000734 001402 CMP #500,R0
817 000736 000000 BEQ ;
818 000760 000406 HALT BR G*30
819
820 000762 017700 175072 MOV #YPOS,R0 ;READ Y POSITION
821 000766 022700 001432 CMP #1432,R0
822 000772 001401 BEQ ;
823 000774 000000 HALT ;
824
825
```

```

827
828
829
830
831 003776 104000
832 004000 012777 116000 175014 G9321 SCOPE
833 004000 005077 175012 MOV #110000,0DBUF ;LOW INTENSITY - POINT MODE
834 004012 005077 175010 CLR 0DBUF1 ;CLEAR X POSITION
835 004016 012777 172000 CLR 0DBUF2 ;CLEAR Y POSITION
836 004024 013777 001022 MOV #170000,0DBUF3 ;LOAD STATUS "A" REGISTER, STOP
837 004032 004737 012340 DBUF,0DPC ;LOAD DISPLAY R,C,
838 004036 017700 175014 JSR 7,DELAY ;EXECUTE A PROGRAM DELAY
839 004042 001400 MOV 0XRD3,R0 ;READ X POSITION
840 004044 000000 BEQ +06 ;WAS IT 0?
841 004046 000400 BR G933 ;X POSITION REGISTER FAILED TO RESET
;USING POINT DATA MODE
842
843 004050 017700 175004 MOV 0YRD3,R0 ;READ Y POSITION
844 004054 001400 BEQ +04 ;WAS IT 0?
845 004056 000000 HALT ;Y POSITION REGISTER FAILED TO RESET
;USING POINT DATA MODE
846
847
848
849
850
851 004060 104000
852 004062 012777 116000 174732 G9331 SCOPE
853 004070 012777 001777 174726 MOV #110000,0DBUF ;LOW INTENSITY - POINT MODE
854 004076 012777 001777 174722 MOV #0777,0DBUF1 ;SET X POSITION
855 004104 012777 172000 174716 MOV #1777,0DBUF2 ;SET Y POSITION
856 004112 013777 001022 174732 MOV #170000,0DBUF3 ;LOAD STATUS A REGISTER, STOP
857 004120 004737 012340 DBUF,0DPC ;LOAD DISPLAY R,C,
858 004124 017700 174726 JSR 7,DELAY ;EXECUTE A PROGRAM DELAY
859 004130 022700 001777 MOV 0XRD3,R0 ;READ X POSITION
860 004134 001400 CMP #1777,R0 ;WAS IT SET?
861 004136 000000 BEQ +06 ;X POSITION REGISTER FAILED TO SET
862 004140 000400 BR G933 ;USING POINT DATA MODE
863
864 004142 017700 174712 MOV 0YRD3,R0 ;READ Y POSITION
865 004146 022700 001777 CMP #1777,R0 ;WAS IT SET?
866 004152 001400 BEQ +04 ;Y POSITION REGISTER FAILED TO SET
867 004154 000000 HALT ;USING POINT DATA MODE
868
869

```

```

871
872
873
874
875 004156 104000
876 004160 012777 116000 174634 G9341 SCOPE
877 004166 012777 001292 174630 MOV #110000,0DBUF ;LOW INTENSITY - POINT MODE
878 004174 012777 001292 174624 MOV #1292,0DBUF1 ;SET X POSITION
879 004202 012777 172000 174620 MOV #1292,0DBUF2 ;SET Y POSITION
880 004210 013777 001022 174634 MOV #170000,0DBUF3 ;LOAD STATUS REGISTER A, STOP
881 004216 004737 012340 DBUF,0DPC ;LOAD DISPLAY R,C,
882 004222 017700 174630 JSR 7,DELAY ;EXECUTE A PROGRAM DELAY
883 004226 022700 001292 MOV 0XRD3,R0 ;READ X POSITION
884 004232 001400 CMP #1292,R0 ;WAS IT 0?
885 004234 000000 BEQ +06 ;X POSITION REGISTER FAILED
886 004236 000400 BR G933 ;USING POINT DATA MODE
887
888 004240 017700 174614 MOV 0YRD3,R0 ;READ Y POSITION
889 004244 022700 001292 CMP #1292,R0 ;WAS IT 0?
890 004250 001400 BEQ +04 ;Y POSITION REGISTER FAILED
891 004252 000000 HALT ;USING POINT DATA MODE
892
893
894
895
896
897 004254 104000
898 004256 012777 116000 174536 G9351 SCOPE
899 004264 012777 000525 174532 MOV #110000,0DBUF ;LOW INTENSITY - POINT MODE
900 004272 012777 000525 174526 MOV #0525,0DBUF1 ;SET X POSITION
901 004300 012777 172000 174522 MOV #0525,0DBUF2 ;SET Y POSITION
902 004306 013777 001022 174536 MOV #170000,0DBUF3 ;LOAD STATUS REGISTER A, STOP
903 004314 004737 012340 DBUF,0DPC ;LOAD DISPLAY R,C,
904 004320 017700 174532 JSR 7,DELAY ;EXECUTE A PROGRAM DELAY
905 004324 022700 000525 MOV 0XRD3,R0 ;READ X POSITION
906 004330 001400 CMP #0525,R0 ;WAS IT 0?
907 004332 000000 BEQ +06 ;X POSITION REGISTER FAILED
908 004334 000400 BR G933 ;USING POINT DATA MODE
909
910 004336 017700 174516 MOV 0YRD3,R0 ;READ Y POSITION
911 004342 022700 000525 CMP #0525,R0 ;WAS IT 0?
912 004346 001400 BEQ +04 ;Y POSITION REGISTER FAILED
913 004350 000000 HALT ;USING POINT DATA MODE
914
915

```



```

    996
    997
    998
    999
    1000 004654 104000
    1001 004656 012783 001777
    1002 004662 012784 000001
    1003
    1004 004666 104000
    1005 004670 013700 001022
    1006 004674 012720 110000
    1007 004700 005020
    1008 004702 005020
    1009 004704 012720 110000
    1010 004706 010420
    1011 004712 010420
    1012 004714 012720 172000
    1013 004716 013777 001022 174124
    1014 004720 004737 012340
    1015
    1016 004732 017700 174120
    1017 004736 020400
    1018 004740 001402
    1019 004742 000000
    1020 004744 000411
    1021
    1022 004746 017700 174106
    1023 004752 020400
    1024 004754 001402
    1025 004756 000000
    1026 004760 000403
    1027
    1028 004762 005204
    1029 004764 005303
    1030 004766 001340

    ;TEST THAT LONG VECTOR MODE INCREMENT X AND Y AXIS PROPERLY
    ;COUNT 0=1777
    GT39I SCOPE
    MOV #1777,R3
    MOV #1,R4 ;SET UP A COUNTER
    ;PRESET THE COMPARED VALUE

    GT39A SCOPE
    MOV DBUF,R0 ;SET UP R0
    MOV #110000,(R0) ;LOAD "POINT MODE"
    CLR (R0) ;CLEAR X AXIS
    CLR (R0) ;CLEAR Y AXIS
    MOV #110000,(R0) ;LOAD "LONG VECTOR MODE"
    MOV R4,(R0) ;PRESET "DELTA X AXIS"
    MOV R4,(R0) ;PRESET "DELTA Y AXIS"
    MOV #172000,(R0)
    MOV DBUF,ODPC ;LOAD THE DISPLAY P.C.
    JSR 7,DELAY ;EXECUTE A PROGRAM DELAY

    MOV @XPOS,R0 ;READ X AXIS
    CMP R4,R0 ;ARE THEY EQUAL?
    BEQ +06 ;YES
    HALT ;NO, INCREMENT X AXIS VIA
    BR GT40 ;LONG VECTOR MODE FAILED

    MOV @YPOS,R0 ;READ Y AXIS
    CMP R4,R0 ;ARE THEY EQUAL?
    BEQ +06 ;YES
    HALT ;NO, INCREMENT Y AXIS VIA
    BR GT40 ;LONG VECTOR MODE FAILED

    INC R4 ;INCREMENT EXPECTED VALUE
    DEC R3 ;FINISHED?
    BNE GT39A ;NO, TEST MORE DATA
    
```

```

    1032
    1033
    1034
    1035
    1036 004770 104000
    1037 004772 012703 002000
    1038 004776 012704 001777
    1039 004802 012705 020001
    1040
    1041 005006 104000
    1042 005010 013700 001022
    1043 005014 012720 110000
    1044 005020 005020
    1045 005022 005020
    1046 005024 012720 110000
    1047 005030 010420
    1048 005032 010420
    1049 005034 012710 172000
    1050 005040 013777 001022 174004
    1051 005046 004737 012340
    1052
    1053 005052 017700 174000
    1054 005056 020400
    1055 005060 001402
    1056 005062 000000
    1057 005064 000412
    1058
    1059 005066 017700 173766
    1060 005072 020400
    1061 005074 001402
    1062 005076 000000
    1063 005100 000404
    1064
    1065 005102 005205
    1066 005104 005304
    1067 005106 005303
    1068 005110 001339

    ;TEST THAT LONG VECTOR MODE DECREMENTS X AND Y AXIS PROPERLY
    ;COUNT 1777=0
    GT40I SCOPE
    MOV #2000,R3
    MOV #1777,R4
    MOV #2000,R5 ;SET UP A COUNTER
    ;PRESET THE COMPARED VALUE

    GT40A SCOPE
    MOV DBUF,R0 ;SET UP R0
    MOV #110000,(R0) ;LOAD "POINT MODE"
    CLR (R0) ;CLEAR X AXIS
    CLR (R0) ;CLEAR Y AXIS
    MOV #110000,(R0) ;LOAD "LONG VECTOR MODE"
    MOV R5,(R0) ;PRESET "DELTA X AXIS"
    MOV R5,(R0) ;PRESET "DELTA Y AXIS"
    MOV #172000,(R0)
    MOV DBUF,ODPC ;LOAD THE DISPLAY P.C.
    JSR 7,DELAY ;EXECUTE A PROGRAM DELAY

    MOV @XPOS,R0 ;READ X AXIS
    CMP R4,R0 ;ARE THEY EQUAL?
    BEQ +06 ;YES
    HALT ;NO, DECREMENT X AXIS VIA
    BR GT41 ;LONG VECTOR MODE FAILED

    MOV @YPOS,R0 ;READ Y AXIS
    CMP R4,R0 ;ARE THEY EQUAL?
    BEQ +06 ;YES
    HALT ;NO, DECREMENT Y AXIS VIA
    BR GT41 ;LONG VECTOR MODE FAILED

    INC R5 ;INCREMENT "DELTA X=Y"
    DEC R4 ;DECREMENT EXPECTED VALUE
    DEC R3 ;FINISHED?
    BNE GT40A ;NO, TEST MORE DATA
    
```

```
TEST THAT X AND Y AXIS INCREMENTS PROPERLY
USING SHORT VECTOR MODE
ICOUNT 1

1070
1071
1072
1073
1074 005112 104000 GT411 SCOPE
1075 005114 013700 001022 MOV DBUF,R0 ;SET UP R0
1076 005120 012720 116000 MOV #11000,(0)+ ;LOAD "SET POINT MODE"
1077 005124 005020 CLR (0)+ ;CLEAR X AXIS
1078 005126 005020 CLR (0)+ ;CLEAR Y AXIS
1079 005130 012720 104000 MOV #10000,(0)+ ;LOAD "SET SHORT VECTOR MODE"
1080 005134 012720 000201 MOV #20,(0)+ ;PRESET "DELTA X AND DELTA Y"
1081 005140 012710 172000 MOV #17000,(0)
1082 005144 013777 001022 173700 MOV DBUF,ODPC ;LOAD THE DISPLAY PC
1083 005152 004737 012340 JSR 7,DELAY ;EXECUTE A PROGRAM DELAY

1084
1085 005156 017700 173674 MOV #XPOS,R0 ;READ X AXIS
1086 005102 022700 000001 CMP #1,R0 ;ARE THEY EQUAL?
1087 005106 001402 BEQ +6 ;YES
1088 005170 000000 HALT ;NO, INCREMENT X AXIS FAILED USING
1089 005172 000406 BR GT42 ;SHORT VECTOR MODE

1090
1091 005174 017700 173600 MOV #YPOS,R0 ;READ Y AXIS
1092 005200 022700 000001 CMP #1,R0 ;ARE THEY EQUAL?
1093 005204 001401 BEQ +6 ;YES
1094 005206 000000 HALT ;NO INCREMENT Y AXIS FAILED
1095 ;USING SHORT VECTOR MODE

TEST THAT X AND Y AXIS DECREMENT PROPERLY
USING SHORT VECTOR MODE
ICOUNT 1

1100
1101 005210 104000 GT421 SCOPE
1102 005212 013700 001022 MOV DBUF,R0 ;SET UP R0
1103 005216 012720 116000 MOV #11000,(0)+ ;LOAD "SET POINT MODE"
1104 005222 005020 CLR (0)+ ;CLEAR X AXIS
1105 005224 005020 CLR (0)+ ;CLEAR Y AXIS
1106 005226 012720 104000 MOV #10000,(0)+ ;LOAD "SET SHORT VECTOR MODE"
1107 005232 012720 000301 MOV #20,(0)+ ;PRESET "DELTA X AND DELTA Y"
1108 005236 012710 172000 MOV #17000,(0)
1109 005242 013777 001022 173602 MOV DBUF,ODPC ;LOAD THE DISPLAY PC
1110 005250 004737 012340 JSR 7,DELAY ;EXECUTE A PROGRAM DELAY

1111
1112 005254 017700 173576 MOV #XPOS,R0 ;READ X AXIS
1113 005260 022700 001777 CMP #1777,R0 ;ARE THEY EQUAL?
1114 005264 001402 BEQ +6 ;YES
1115 005266 000000 HALT ;NO, DECREMENT X AXIS FAILED USING
1116 005270 000406 BR GT43 ;SHORT VECTOR MODE

1117
1118 005272 017700 173562 MOV #YPOS,R0 ;READ Y AXIS
1119 005276 022700 001777 CMP #1777,R0 ;ARE THEY EQUAL?
1120 005302 001401 BEQ +6 ;YES
1121 005304 000000 HALT ;NO DECREMENT Y AXIS FAILED
1122 ;USING SHORT VECTOR MODE
```

```
TEST THAT X AND Y AXIS INCREMENT PROPERLY
USING SHORT VECTOR MODE
ICOUNT 0=77

1124
1125
1126
1127
1128
1129 005306 104000 GT431 SCOPE
1130 005310 012700 000077 MOV #77,R3 ;SET UP A COUNT LOCATION
1131 005314 012702 000001 MOV #1,R2 ;SET UP THE COMPARED LOCATION
1132 005318 012704 000201 MOV #20,R4 ;SET UP "DELTA X=Y"

1133
1134 005324 104000 GT43A1 SCOPE
1135 005326 013700 001022 MOV DBUF,R0 ;SET UP R0
1136 005332 012720 116000 MOV #11000,(0)+ ;LOAD "SET POINT DATA MODE"
1137 005336 005020 CLR (0)+ ;CLEAR X AXIS
1138 005340 005020 CLR (0)+ ;CLEAR Y AXIS
1139 005342 012720 104000 MOV #10000,(0)+ ;LOAD "SET SHORT VECTOR MODE"
1140 005346 010420 MOV R4,(0)+ ;PRESET "DELTA X AND DELTA Y"
1141 005350 012710 172000 MOV #17000,(0)
1142 005354 013777 001022 173470 MOV DBUF,ODPC ;LOAD THE DISPLAY P.C.
1143 005362 004737 012340 JSR 7,DELAY ;EXECUTE A PROGRAM DELAY

1144
1145 005366 017700 173464 MOV #XPOS,R0 ;READ X POSITION
1146 005372 020200 CMP R2,R0 ;ARE THEY EQUAL?
1147 005374 001402 BEQ +6 ;YES
1148 005376 000000 HALT ;INCREMENT X AXIS FAILED USING
1149 005400 000413 BR GT44 ;SHORT VECTOR MODE

1150
1151 005402 017700 173452 MOV #YPOS,R0 ;READ Y POSITION
1152 005406 020200 CMP R2,R0 ;ARE THEY EQUAL?
1153 005410 001402 BEQ +6 ;YES
1154 005412 000000 HALT ;INCREMENT Y AXIS FAILED USING
1155 005414 000403 BR GT44 ;SHORT VECTOR MODE

1156
1157 005426 002704 000201 ADD #20,R4 ;ADD DELTA X=Y
1158 005422 005202 INC R2 ;INCREMENT EXPECTED VALUE
1159 005424 005303 DEC R3 ;DECREMENT COUNT, FINISHED?
1160 005426 001337 BNE GT43A ;NO, TEST MORE DATA
```

```

1162
1163
1164
1165
1166
1167 005430 104000
1168 005432 012703 000077
1169 005436 012702 001777
1170 005442 012704 020301
1171
1172 005446 104000
1173 005450 013700 001022
1174 005454 012720 116000
1175 005460 005020
1176 005462 005020
1177 005464 012720 106000
1178 005470 010400
1179 005472 012710 172000
1180 005476 013777 001022 173346
1181 005504 004737 012340
1182
1183 005510 017700 173342
1184 005514 020200
1185 005516 001402
1186 005520 000000
1187 005522 000413
1188
1189 005524 017700 173330
1190 005530 020200
1191 005532 001402
1192 005534 000000
1193 005536 000405
1194
1195 005540 062704 000201
1196 005544 005302
1197 005546 005303
1198 005550 001337
1199

;TEST THAT X AND Y AXIS DECREMENT PROPERLY
;USING SHORT VECTOR MODE
;COUNT 77=8
GT441 SCOPE
MOV #77,R3 ;SET UP A COUNT LOCATION
MOV #177,R2 ;SET UP THE COMPARED LOCATION
MOV #20301,R4 ;PRESET THE "DELTA X=Y"

SCOPE
MOV DBUF#R0 ;SET UP R0
MOV #11000,(R0) ;LOAD "SET POINT DATA MODE"
CLR (R0) ;CLEAR X AXIS
CLR (R0) ;CLEAR Y AXIS
MOV #10000,(R0) ;LOAD "SET SHORT VECTOR MODE"
MOV R4,(R0) ;PRESET "DELTA X AND DELTA Y"
MOV #17000,(R0)
MOV DBUF#ODPC ;LOAD THE DISPLAY P.C.
JSR 7,DELAY ;EXECUTE A PROGRAM DELAY

MOV #XPOS,R0 ;READ X POSITION
CMP R2,R0 ;ARE THEY EQUAL?
BCQ .+6 ;YES
HALT ;DECREMENT X AXIS FAILED USING
BR GT4R ;SHORT VECTOR MODE

MOV #YPOS,R0 ;READ Y POSITION
CMP R2,R0 ;ARE THEY EQUAL?
BCQ .+6 ;YES DECREMENT
HALT ;DECREMENT Y AXIS FAILED USING
BR GT4R ;SHORT VECTOR MODE

ADD #20301,R4 ;ADD "DELTA X=Y"
DEC R2 ;DECREMENT EXPECTED VALUE
DEC R3 ;DECREMENT COUNT, FINISHED?
BNK GT49A ;NO, TEST MORE DATA
    
```

```

1201
1202
1203
1204
1205 005532 104000
1206 005534 013700 001022
1207 005536 012720 116000
1208 005544 005020
1209 005546 005020
1210 005550 012720 130000
1211 005574 012720 000201
1212 005600 012710 172000
1213 005604 013777 001022 173240
1214 005612 004737 012340
1215
1216 005616 017700 173234
1217 005622 022700 000001
1218 005626 001402
1219 005630 000000
1220 005632 000406
1221
1222 005634 017700 173220
1223 005640 022700 000001
1224 005644 001401
1225 005646 000000
1226
1227
1228
1229
1230
1231
1232 005650 104000
1233 005652 013700 001022
1234 005656 012720 116000
1235 005662 005020
1236 005664 005020
1237 005666 012720 130000
1238 005672 012720 020301
1239 005676 012710 172000
1240 005702 013777 001022 173142
1241 005710 004737 012340
1242
1243 005714 017700 173136
1244 005720 022700 001777
1245 005724 001402
1246 005726 000000
1247 005730 000406
1248
1249 005732 017700 173122
1250 005736 022700 001777
1251 005742 001401
1252 005744 000000
1253

;TEST THAT X AND Y AXIS INCREMENTS PROPERLY
;USING RELATIVE POINT MODE
;COUNT 1
GT451 SCOPE
MOV DBUF#R0 ;SET UP R0
MOV #11000,(R0) ;LOAD "SET POINT MODE"
CLR (R0) ;CLEAR X AXIS
CLR (R0) ;CLEAR Y AXIS
MOV #13000,(R0) ;LOAD "SET RELATIVE POINT MODE"
MOV #20301,(R0) ;PRESET "DELTA X AND DELTA Y"
MOV #17000,(R0)
MOV DBUF#ODPC ;LOAD THE DISPLAY PC
JSR 7,DELAY ;EXECUTE A PROGRAM DELAY

MOV #XPOS,R0 ;READ X AXIS
CMP #1,R0 ;ARE THEY EQUAL?
BCQ .+6 ;YES
HALT ;NO, INCREMENT X AXIS FAILED USING
BR GT4R ;RELATIVE POINT MODE

MOV #YPOS,R0 ;READ Y AXIS
CMP #1,R0 ;ARE THEY EQUAL?
BCQ .+6 ;YES
HALT ;NO INCREMENT Y AXIS FAILED
;USING RELATIVE POINT MODE

;TEST THAT X AND Y AXIS DECREMENT PROPERLY
;USING RELATIVE POINT MODE
;COUNT 1
GT461 SCOPE
MOV DBUF#R0 ;SET UP R0
MOV #11000,(R0) ;LOAD "SET POINT MODE"
CLR (R0) ;CLEAR X AXIS
CLR (R0) ;CLEAR Y AXIS
MOV #13000,(R0) ;LOAD "SET RELATIVE POINT MODE"
MOV #20301,(R0) ;PRESET "DELTA X AND DELTA Y"
MOV #17000,(R0)
MOV DBUF#ODPC ;LOAD THE DISPLAY PC
JSR 7,DELAY ;EXECUTE A PROGRAM DELAY

MOV #XPOS,R0 ;READ X AXIS
CMP #177,R0 ;ARE THEY EQUAL?
BCQ .+6 ;YES
HALT ;NO, DECREMENT X AXIS FAILED USING
BR GT47 ;RELATIVE POINT MODE

MOV #YPOS,R0 ;READ Y AXIS
CMP #177,R0 ;ARE THEY EQUAL?
BCQ .+6 ;YES
HALT ;NO DECREMENT Y AXIS FAILED
;USING RELATIVE POINT MODE
    
```



```

1332
1333
1334
1335
1336
1337
1338 001212 104000
1339 001214 012703 000077
1340 001220 012704 000001
1341 001224 012737 174101 001036
1342
1343 001232 104000
1344 001234 013700 001022
1345 001240 012720 116000
1346 001244 005020
1347 001246 005020
1348 001250 013720 001036
1349 001254 012720 120000
1350 001200 005020
1351 001202 012710 172000
1352 001204 013777 001022 172554
1353 001274 004737 012340
1354
1355 001370 017700 172554
1356 001304 020400
1357 001306 001402
1358 001310 000000
1359 001312 000405
1360
1361 001314 005237 001036
1362 001320 005204
1363 001322 005303
1364 001324 001343

                                ILOAD STATUS B TEST
                                IUSE GRAPH PLOT X MODE TO TEST Y AXIS IS INCREMENTED BY
                                I"SCALE" REGISTER

GT491  SCOPE
      MOV #77,R3 ;SET UP EXECUTION COUNTER
      MOV #1,R4 ;SET UP COMPARED DATA
      MOV #173101,DSAVE ;SET UP BASIC ILOAD STATUS B"

GT49A1 SCOPE
      MOV DBUF,R0 ;SET UP R0
      MOV #110000,(R)+ ;LOAD "POINT MODE"
      CLR (R)+ ;CLEAR X AXIS
      CLR (R)+ ;CLEAR Y AXIS
      MOV DSAVE,(R)+ ;LOAD "SET STATUS B"
      MOV #120000,(R)+ ;LOAD "SET GRAPH PLOT X MODE"
      CLR (R)+ ;LOAD "X GRAPH PLOT DATA"
      MOV #172000,(R)
      MOV DBUF,DDPC ;LOAD THE DISPLAY P.C.
      JSR 7,DELAY ;EXECUTE A PROGRAM DELAY

      MOV @YRDS,R0 ;READ Y AXIS
      CMP R4,R0 ;COMPARE TO EXPECTED VALUE
      BEQ ,+6 ;ARE THEY EQUAL?
      HALT ;LOAD "STATUS B" FAILED TO LOAD
      BR GT50B ;THE Y AXIS CORRECTLY

      INC DSAVE
      INC R4 ;INCREMENT THE STATUS B COUNT
      DEC R3 ;DECREMENT THE EXECUTION COUNT
      BNE GT49A ;TEST MORE DATA
    
```

```

1366
1367
1368
1369
1370
1371 001326 104000
1372 001330 012703 000077
1373 001334 012704 000001
1374 001340 012737 174101 001036
1375
1376 001346 104000
1377 001350 013700 001022
1378 001354 012720 116000
1379 001360 005020
1380 001362 005020
1381 001364 013720 001036
1382 001370 012720 124000
1383 001374 000000
1384 001376 012710 172000
1385 001402 013777 001022 172442
1386 001410 004737 012340
1387
1388 001414 017700 172436
1389 001420 042700 176000
1390 001424 020400
1391 001426 001402
1392 001430 000000
1393 001432 000405
1394
1395 001434 005237 001036
1396 001440 005204
1397 001442 005303
1398 001444 001341
1399
1400 001446 012777 174100 172346 GT50B1 MOV #173100,0DBUF
1401 001454 012777 172000 172342 MOV #172000,0DBUF
1402 001462 013777 001022 172342 MOV DBUF,DDPC

                                ILOAD STATUS B TEST
                                IUSE GRAPH PLOT Y MODE TO TEST X AXIS IS INCREMENTED BY
                                I"SCALE" REGISTER

GT501  SCOPE
      MOV #77,R3 ;SET UP EXECUTION COUNTER
      MOV #1,R4 ;SET UP COMPARED DATA
      MOV #173101,DSAVE ;SET UP BASIC ILOAD STATUS B"

GT50A1 SCOPE
      MOV DBUF+R0 ;SET UP R0
      MOV #110000,(R)+ ;LOAD "POINT MODE"
      CLR (R)+ ;CLEAR X AXIS
      CLR (R)+ ;CLEAR Y AXIS
      MOV DSAVE,(R)+ ;LOAD "SET STATUS B"
      MOV #120000,(R)+ ;LOAD "SET GRAPH PLOT Y MODE"
      CLR (R)+ ;LOAD "Y GRAPH PLOT DATA"
      MOV #172000,(R)
      MOV DBUF,DDPC ;LOAD THE DISPLAY P.C.
      JSR 7,DELAY ;EXECUTE A PROGRAM DELAY

      MOV @XPRDS,R0 ;READ X AXIS
      BIC #100000,R0 ;MASK TO BITS 0-9
      CMP R4,R0 ;COMPARE TO EXPECTED VALUE
      BEQ ,+6 ;ARE THEY EQUAL?
      HALT ;LOAD "STATUS B" FAILED TO LOAD
      BR GT50B ;THE X AXIS CORRECTLY

      INC DSAVE
      INC R4 ;INCREMENT THE STATUS B COUNT
      DEC R3 ;DECREMENT THE EXECUTION COUNT
      BNE GT50A ;TEST MORE DATA
    
```

```

1404 ;TEST THAT THE EDGE FLAG IS NOT SET AT 1777,0
1405
1406 001470 104000 001022 GT511 SCOPE
1407 001472 013700 001022 MOV DBUF,RO
1408 001476 012720 116000 MOV #110000,(B)+ ;LOAD POINT
1409 001502 012720 001777 MOV #1777,(B)+ ;LOAD X
1410 001506 012720 000000 MOV #0,(B)+ ;LOAD Y
1411 001542 012720 172000 MOV #172000,(B)+ ;LOAD STOP
1412 001546 013777 001022 172326 MOV DBUF,ROPC ;START
1413 001554 004737 012340 JSR 7,DLAY ;EXECUTE A PROGRAM DELAY
1414
1415 001530 032777 000040 172316 BIT #40,RODR ;TEST BIT 5
1416 001536 001401 BEQ .+4 ;EDGE FLAG SET ON 1777,0
1417 001540 000000 HALT ;ERROR, EDGE FLAG SET ON 1777,0
1418
1419 ;EDGE FLAG TEST
1420 ;TEST THAT EXCEEDING +X AXIS SETS EDGE FLAG
1421
1422 001542 104000 001022 GT521 SCOPE
1423 001544 013700 001022 MOV DBUF,RO
1424 001550 012720 116000 MOV #110000,(B)+ ;LOAD POINT
1425 001594 012720 001777 MOV #1777,(B)+ ;LOAD MAX X
1426 001500 012720 000000 MOV #0,(B)+ ;LOAD Y
1427 001504 012720 110000 MOV #110000,(B)+ ;LOAD LONG VECTOR
1428 001570 012720 000001 MOV #1,(B)+ ;LOAD DELTA X
1429 001574 012720 000000 MOV #0,(B)+ ;LOAD DELTA Y
1430 001600 012720 172000 MOV #172000,(B)+ ;LOAD STOP
1431 001604 013777 001022 172240 MOV DBUF,ROPC ;START DISPLAY
1432 001612 004737 012340 JSR 7,DLAY ;EXECUTE A PROGRAM DELAY
1433
1434 001616 032777 000040 172230 BIT #40,RODR ;TEST BIT 5
1435 001624 001002 BNE .+6 ;EDGE FLAG FAILED TO SET
1436 001626 000000 HALT ;EDGE FLAG FAILED TO SET
1437 001630 000424 BR GT53
1438
1439 ;SUB=TEST, TEST THAT THE EDGE FLAG CLEARS
1440
1441
1442 001632 013700 001022 MOV DBUF,RO
1443 001636 012720 116000 MOV #110000,(B)+ ;LOAD POINT
1444 001642 012720 001777 MOV #1777,(B)+ ;LOAD X
1445 001646 012720 000000 MOV #0,(B)+ ;LOAD Y
1446 001692 012720 172000 MOV #172000,(B)+ ;LOAD STOP
1447 001696 013777 001022 172100 MOV DBUF,ROPC ;START DISPLAY
1448 001604 004737 012340 JSR 7,DLAY ;EXECUTE A PROGRAM DELAY
1449
1450 001670 032777 000040 172150 BIT #40,RODR ;TEST BIT 5
1451 001676 001401 BEQ .+4 ;ERROR, EDGE FLAG FAILED TO CLEAR
1452 001700 000000 HALT ;ERROR, EDGE FLAG FAILED TO CLEAR

```

```

1454 ;EDGE FLAG TEST
1455 ;TEST THAT EXCEEDING -X AXIS SETS EDGE FLAG
1456
1457
1458
1459 001702 104000 001022 GT531 SCOPE
1460 001704 013700 001022 MOV DBUF,RO
1461 001710 012720 116000 MOV #110000,(B)+ ;LOAD POINT
1462 001714 012720 000000 MOV #0,(B)+ ;LOAD MAX X
1463 001720 012720 000000 MOV #0,(B)+ ;LOAD Y
1464 001724 012720 110000 MOV #110000,(B)+ ;LOAD LONG VECTOR
1465 001730 012720 000001 MOV #20001,(B)+ ;LOAD DELTA X
1466 001734 012720 000000 MOV #0,(B)+ ;LOAD DELTA Y
1467 001740 012720 172000 MOV #172000,(B)+ ;LOAD STOP
1468 001744 013777 001022 172100 MOV DBUF,ROPC ;START DISPLAY
1469 001752 004737 012340 JSR 7,DLAY ;EXECUTE A PROGRAM DELAY
1470
1471 001756 032777 000040 172070 BIT #40,RODR ;TEST BIT 5
1472 001784 001002 BNE .+6 ;EDGE FLAG FAILED TO SET
1473 001786 000000 HALT ;EDGE FLAG FAILED TO SET
1474 001790 000424 BR GT59
1475
1476 ;SUB=TEST, TEST THAT THE EDGE FLAG CLEARS
1477
1478 001772 013700 001022 MOV DBUF,RO
1479 001776 012720 116000 MOV #110000,(B)+ ;LOAD POINT
1480 001802 012720 000000 MOV #0,(B)+ ;LOAD X
1481 001806 012720 000000 MOV #0,(B)+ ;LOAD Y
1482 001812 012720 172000 MOV #172000,(B)+ ;LOAD STOP
1483 001816 013777 001022 172020 MOV DBUF,ROPC ;START DISPLAY
1484 001824 004737 012340 JSR 7,DLAY ;EXECUTE A PROGRAM DELAY
1485
1486 001830 032777 000040 172010 BIT #40,RODR ;TEST BIT 5
1487 001836 001401 BEQ .+4 ;ERROR, EDGE FLAG FAILED TO CLEAR
1488 001840 000000 HALT ;ERROR, EDGE FLAG FAILED TO CLEAR

```

```

1490          JEDGE FLAG TEST
1491          ITEST THAT EXCEEDING +Y AXIS SETS EDGE FLAG
1492
1493 007042 104000          G9541 SCOPE
1494 007044 013700 001022      MOV      DBUF,R0
1495 007050 012720 116000      MOV      #110000,(R0)  ;LOAD POINT
1496 007054 012720 000000      MOV      #0,(R0)      ;LOAD X
1497 007060 013720 001012      MOV      GSYARS,(R0)  ;LOAD MAX Y
1498 007064 012720 110000      MOV      #110000,(R1)  ;LOAD LONG VECTOR
1499 007070 012720 000000      MOV      #0,(R0)      ;LOAD DELTA X
1500 007074 012720 000001      MOV      #1,(R0)      ;LOAD DELTA Y
1501 007100 012720 172000      MOV      #170000,(R0)  ;LOAD STOP
1502 007104 013777 001022 171740  MOV      DBUF,OPDC     ;START DISPLAY
1503 007112 004737 012340      JSR      7,DLAY       ;EXECUTE A PROGRAM DELAY
1504
1505 007116 032777 000040 171730      BIT      #40,ODSR     ;TEST BIT 5
1506 007124 001002          BNE      ,+4
1507 007126 000000          HALT
1508 007130 000424          BR      G950
1509
1510          ISUB=TEST; TEST THAT THE EDGE FLAG CLEARS
1511
1512 007132 013700 001022      MOV      DBUF,R0
1513 007136 012720 116000      MOV      #110000,(R0)  ;LOAD POINT
1514 007142 012720 000000      MOV      #0,(R0)      ;LOAD X
1515 007146 012720 000000      MOV      #0,(R0)      ;LOAD Y
1516 007152 012720 172000      MOV      #170000,(R1)  ;LOAD LONG VECTOR
1517 007156 013777 001022 171660  MOV      DBUF,OPDC     ;LOAD STOP
1518 007164 004737 012340      JSR      7,DLAY       ;START DISPLAY
1519                                ;EXECUTE A PROGRAM DELAY
1520
1521 007170 032777 000040 171650      BIT      #40,ODSR     ;TEST BIT 5
1522 007176 001401          BEQ      ,+4
1523 007200 000000          HALT
1524                                ;ERROR, EDGE FLAG FAILED TO CLEAR
1525
1526          JEDGE FLAG TEST
1527          ITEST THAT EXCEEDING -Y AXIS SETS EDGE FLAG
1528
1529 007202 104000          G9551 SCOPE
1530 007204 013700 001022      MOV      DBUF,R0
1531 007210 012720 116000      MOV      #110000,(R0)  ;LOAD POINT
1532 007214 012720 000000      MOV      #0,(R0)      ;LOAD X
1533 007218 012720 000000      MOV      #0,(R0)      ;LOAD Y
1534 007224 012720 110000      MOV      #110000,(R1)  ;LOAD LONG VECTOR
1535 007228 012720 000000      MOV      #0,(R0)      ;LOAD DELTA X
1536 007234 012720 000001      MOV      #1,(R0)      ;LOAD DELTA Y
1537 007240 012720 172000      MOV      #170000,(R0)  ;LOAD STOP
1538 007244 013777 001022 171600  MOV      DBUF,OPDC     ;LOAD STOP
1539 007252 004737 012340      JSR      7,DLAY       ;START DISPLAY
1540                                ;EXECUTE A PROGRAM DELAY
1541
1542 007256 032777 000040 171570      BIT      #40,ODSR     ;TEST BIT 5
1543 007264 001001          BNE      ,+4
1544 007266 000000          HALT
1545                                ;EDGE FLAG FAILED TO SET
  
```

```

1544          ITEST THAT THE CHARACTER REGISTER IS LOADED PROPERLY
1545          I CODE 00
1546
1547 007270 104000          G9561 SCOPE
1548 007272 012777 100000 171522      MOV      #100000,OBUP  ;LOAD "CHARACTER MODE"
1549 007300 012777 000000 171516      MOV      #0,OBUP1     ;LOAD "NULL" CHARACTER
1550 007306 012777 172000 171512      MOV      #170000,OBUP2 ;LOAD CHARACTER
1551 007314 013777 001022 171530      MOV      DBUF,OPDC     ;START DISPLAY
1552 007322 004737 012340      JSR      7,DLAY       ;EXECUTE A PROGRAM DELAY
1553 007326 017000 171526      MOV      @YRDS,R0     ;READ CHARACTER REG.
1554 007332 042700 001777          BIC      #1777,R0     ;MASK TO BITS 10-15
1555 007336 022700 000000          CMP      #0,R0
1556 007342 001401          BEQ      ,+4
1557 007344 000000          HALT
1558                                ;ERROR, CHARACTER REGISTER LOADED IN ERROR
1559
1560          ITEST THAT THE CHARACTER REGISTER IS LOADED PROPERLY
1561          I CODE 77
1562
1563 007346 104000          G9571 SCOPE
1564 007348 012777 100000 171444      MOV      #100000,OBUP  ;LOAD "CHARACTER MODE"
1565 007354 012777 000077 171440      MOV      #77,OBUP1    ;LOAD CHARACTER
1566 007362 012777 172000 171434      MOV      #170000,OBUP2 ;LOAD CHARACTER
1567 007372 013777 001022 171492      MOV      DBUF,OPDC     ;START DISPLAY
1568 007400 004737 012340      JSR      7,DLAY       ;EXECUTE A PROGRAM DELAY
1569 007404 017000 171490      MOV      @YRDS,R0     ;READ CHARACTER REG.
1570 007410 042700 001777          BIC      #1777,R0     ;MASK TO BITS 10-15
1571 007414 022700 176000          CMP      #170000,R0
1572 007420 001401          BEQ      ,+4
1573 007422 000000          HALT
1574                                ;ERROR, CHARACTER REGISTER LOADED IN ERROR
1575
1576          ITEST THAT THE CHARACTER REGISTER IS LOADED PROPERLY
1577          I CODE 29
1578
1579 007424 104000          G9581 SCOPE
1580 007426 012777 100000 171366      MOV      #100000,OBUP  ;LOAD "CHARACTER MODE"
1581 007434 012777 000029 171362      MOV      #29,OBUP1    ;LOAD CHARACTER
1582 007442 012777 172000 171356      MOV      #170000,OBUP2 ;LOAD CHARACTER
1583 007450 013777 001022 171374      MOV      DBUF,OPDC     ;START DISPLAY
1584 007460 004737 012340      JSR      7,DLAY       ;EXECUTE A PROGRAM DELAY
1585 007464 017000 171372      MOV      @YRDS,R0     ;READ CHARACTER REG.
1586 007470 042700 001777          BIC      #1777,R0     ;MASK TO BITS 10-15
1587 007476 022700 052000          CMP      #52000,R0
1588 007482 001401          BEQ      ,+4
1589 007500 000000          HALT
1590                                ;ERROR, CHARACTER REGISTER LOADED IN ERROR
  
```

```

1589
1590
1591
1592
1593 007502 104000
1594 007504 012777 100000 171310
1595 007512 012777 000052 171304
1596 007520 012777 172000 171300
1597 007526 013777 001022 171316
1598 007534 004737 012340
1599 007540 017700 171314
1600 007544 042700 001777
1601 007550 022700 124000
1602 007554 001401
1603 007556 000000
1604
1605
1606
1607
1608 007500 104000
1609 007502 012777 116000 171232
1610 007570 012777 001000 171206
1611 007572 012777 001000 171222
1612 007574 012777 100000 171216
1613 007612 000077 171214
1614 007616 012777 172000 171210
1615 007644 013777 001022 171200
1616 007632 004737 012340
1617
1618 007636 017700 171216
1619 007642 042700 001777
1620 007646 022700 000000
1621 007652 001402
1622 007654 000000
1623 007656 000417
1624
1625 007660 017700 171172
1626 007644 022700 001000
1627 007670 001402
1628 007672 000000
1629 007674 000410
1630
1631 007676 017700 171156
1632 007702 042700 170000
1633 007706 022700 001000
1634 007732 001401
1635 007714 000000
1636

```

TEST THAT THE CHARACTER REGISTER IS LOADED PROPERLY
CODE 52

GT591 SCOPE
MOV #100000,00BUF ;LOAD "CHARACTER MODE"
MOV #2,00BUF1 ;LOAD CHARACTER
MOV #170000,00BUF2
MOV 00BUF,00PC ;START DISPLAY
JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
MOV 0YPC0,R0 ;READ CHARACTER REG.
BIC #1777,R0 ;MASK TO BITS 10-15
CMP #124000,R0
BEQ ,+4
HALT ;ERROR, CHARACTER REGISTER LOADED IN ERROR

TEST THAT CHARACTER MODE DOES NOT HANG THE DISPLAY PROCESSOR
TEST THAT "NULL" DOES NOT CHANGE X OR Y AXIS

GT601 SCOPE
MOV #110000,00BUF ;POINT MODE
MOV #1000,00BUF1
MOV #1000,00BUF2 ;1000,1000
MOV #100000,00BUF3 ;LOAD "CHARACTER MODE"
CLR 00BUF4 ;NULL CHARACTER
MOV #170000,00BUF5
MOV 00BUF,00PC ;LOAD THE DISPLAY P.C.
JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
MOV 0YPC0,R0 ;READ CHARACTER REGISTER
BIC #1777,R0 ;MASK TO BITS 10-15
CMP #0,R0
BEQ ,+6
HALT ;CHARACTER REGISTER IN ERROR
BR GT61
MOV 0YPC0,R0 ;READ X AXIS
CMP #1000,R0 ;ARE THEY EQUAL ?
BEQ ,+6 ;YES
HALT ;"NULL" CHARACTER CHANGED X AXIS
BR GT61
MOV 0YPC0,R0 ;READ Y AXIS
BIC #170000,R0 ;MASK TO BITS 0-9
CMP #1000,R0 ;ARE THEY EQUAL ?
BEQ ,+4 ;YES
HALT ;"NULL" CHARACTER CHANGED Y AXIS

```

1638
1639
1640
1641
1642
1643 007716 104000
1644 007720 012777 116000 171074
1645 007726 012777 001000 171070
1646 007734 012777 001000 171064
1647 007742 012777 100000 171060
1648 007750 012777 000015 171054
1649 007756 012777 172000 171050
1650 007764 013777 001022 171040
1651 007772 004737 012340
1652
1653 007776 017700 171056
1654 010002 042700 001777
1655 010006 022700 032000
1656 010012 001402
1657 010014 000000
1658 010016 000417
1659
1660 010020 017700 171032
1661 010024 022700 000000
1662 010030 001402
1663 010032 000000
1664 010034 000410
1665
1666 010036 017700 171016
1667 010042 042700 170000
1668 010046 022700 001000
1669 010052 001401
1670 010054 000000
1671

```

TEST THAT CHARACTER MODE DOES NOT HANG THE DISPLAY PROCESSOR
TEST THAT "CR" DOES CHANGE X AND DOES NOT CHANGE Y AXIS

GT611 SCOPE
MOV #110000,00BUF ;POINT MODE
MOV #1070,00BUF1
MOV #1000,00BUF2 ;1000,1000
MOV #100000,00BUF3 ;LOAD "CHARACTER MODE"
MOV #0,00BUF4 ;LOAD "CR"
MOV #170000,00BUF5 ;LOAD STOP
MOV 00BUF,00PC ;LOAD THE DISPLAY P.C.
JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
MOV 0YPC0,R0 ;READ Y AXIS
BIC #1777,R0 ;MASK TO BITS 10-15
CMP #32000,R0
BEQ ,+6
HALT ;CHARACTER REGISTER FAILED TO LOAD CORRECTLY
BR GT62
MOV 0YPC0,R0 ;READ X AXIS
CMP #0,R0 ;ARE THEY EQUAL ?
BEQ ,+6 ;YES
HALT ;"CR" CHARACTER FAILED TO CHANGE X AXIS CORRECTLY
BR GT62
MOV 0YPC0,R0 ;READ Y AXIS
BIC #170000,R0 ;MASK TO BITS 0-9
CMP #1000,R0 ;ARE THEY EQUAL ?
BEQ ,+4 ;YES
HALT ;"CR" CHARACTER CHANGED Y AXIS

```

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
1701
1702
1703
1704
1705
1706

```

TEST THAT CHARACTER MODE DOES NOT HANG THE DISPLAY PROCESSOR
TEST THAT "LF" DOES NOT CHANGE X BUT DOES CHANGE Y AXIS

```

GT621 SCOPE
MOV #11000,0BUF ;POINT MODE
MOV #1000,0BUF1 ;1000,1000
MOV #1000,0BUF2 ;LOAD "CHARACTER MODE"
MOV #10000,0BUFP3
MOV #12,0BUF4
MOV #17000,0BUFP5
MOV 0BUF,0DPC ;LOAD THE DISPLAY P.C.
JSR 7,DLAY ;EXECUTE A PROGRAM DELAY

MOV 0YPR3,R0 ;READ CHARACTER REG.
BIC #1777,R0 ;MASK TO BITS 10-15
CMP #20000,R0
BEQ ,+6
HALT ;CHARACTER REGISTER IN ERROR
BR GT63

MOV 0XPR3,R0 ;READ X AXIS
CMP #1000,R0 ;ARE THEY EQUAL ?
BEQ ,+6 ;YES
HALT ;"LF" CHARACTER CHANGED X AXIS
BR GT63

MOV 0YPR3,R0 ;READ Y AXIS
BIC #7000,R0 ;MASK TO BITS 10-15
CMP #1000,R0 ;ARE THEY EQUAL ?
BEQ ,+6 ;YES
HALT ;"LF" CHARACTER FAILED TO CHANGE Y AXIS CORRECTLY

```

```

1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741

```

TEST THAT CHARACTER MODE DOES NOT HANG THE DISPLAY PROCESSOR
TEST THAT "A" DOES CHANGE X BUT NOT Y AXIS

```

GT62A1 SCOPE
MOV #11000,0BUF ;POINT MODE
MOV #0,0BUF1 ;0,1000
MOV #1000,0BUF2 ;LOAD "CHARACTER MODE"
MOV #10000,0BUFP3 ;LOAD AN "A"
MOV #12,0BUFP4
MOV #17000,0BUFP5
MOV 0BUF,0DPC ;LOAD THE DISPLAY P.C.
JSR 7,DLAY ;EXECUTE A PROGRAM DELAY

MOV 0YPR3,R0 ;READ CHARACTER REG
BIC #1777,R0 ;MASK TO BITS 10-15
CMP #2000,R0
BEQ ,+6
HALT ;CHARACTER REGISTER IN ERROR
BR GT63

MOV 0XPR3,R0 ;READ X AXIS
CMP 00000,R0 ;ARE THEY EQUAL ?
BEQ ,+6 ;YES
HALT ;"A" CHARACTER FAILED TO CHANGE X AXIS CORRECTLY
BR GT63

MOV 0YPR3,R0 ;READ Y AXIS
BIC #7000,R0 ;MASK TO BITS 0-9
CMP #1000,R0 ;ARE THEY EQUAL ?
BEQ ,+6 ;YES
HALT ;"A" CHARACTER CHANGED Y AXIS

```

```

1743
1744
1745
1746
1747
1748 #10356 104000
1749 010300 012777 116000 170434
1750 010306 012777 001000 170430
1751 010374 012777 001000 170424
1752 010402 012777 100000 170420
1753 010410 012777 000010 170414
1754 010416 012777 172000 170410
1755 010424 013777 001022 170420
1756 010432 004737 012340
1757
1758 010436 017700 170416
1759 010442 042700 001777
1760 010446 022700 020000
1761 010492 001402
1762 010494 000000
1763 010496 000425
1764
1765 010400 017700 170372
1766 010404 023700 001050
1767 010470 001402
1768 010472 000000
1769 010474 000416
1770
1771 010476 017700 170356
1772 010502 042700 176000
1773 010506 022700 001000
1774 010512 001401
1775 010514 000000
1776
1777
1778
1779 010516 017700 170332
1780 010522 032700 000100
1781 010526 001401
1782 010530 000000
1783

```

;TEST THAT CHARACTER MODE DOES NOT HAND THE DISPLAY PROCESSOR
;TEST THAT "BS" DOES CHANGE X BUT NOT Y AXIS

```

GT031 SCOPE
MOV #11000,0DBUF ;POINT MODE
MOV #1000,0DBUF1
MOV #1000,0DBUF2
MOV #1000,0DBUF3
MOV #10,0DBUF4
MOV #17000,0DBUF5
MOV DBUF1,0DBUF ;LOAD THE DISPLAY P.C.
JSR 7,DELAY ;EXECUTE A PROGRAM DELAY
MOV 0YPOS,R0 ;READ CHARACTER REG
BIC #177,R0 ;MASK TO BITS 10-19
CMP #2000,R0
BEQ ,+6
HALT ;CHARACTER REGISTER IN ERROR
BR GT03
MOV 0XPOS,R0 ;READ X AXIS
CMP 0MS1E,R0 ;ARE THEY EQUAL ?
BEQ ,+6
HALT ;"BS" CHARACTER FAILED TO CHANGE X AXIS CORRECTLY
BR GT03
MOV 0YPOS,R0 ;READ Y AXIS
BIC #17000,R0 ;MASK TO BITS 0-9
CMP #1000,R0 ;ARE THEY EQUAL ?
BEQ ,+6
HALT ;"BS" CHARACTER CHANGED Y AXIS

```

;TEST THAT "SHIFT-OUT" STATUS BIT IS NOT SET

```

GT03A1 MOV 0DBR,R0 ;READ STATUS
BIT #100,R0
BEQ ,+4
HALT ;SHIFT OUT STATUS BIT IS SET

```

```

1785
1786
1787
1788
1789 #10532 104000
1790 010534 012777 116000 170260
1791 010542 012777 001000 170254
1792 010550 012777 001000 170250
1793 010556 012777 100000 170244
1794 010564 012777 037410 170240
1795 010572 012777 170000 170234
1796 010600 013777 001022 170244
1797 010606 004737 012340
1798
1799 010612 017700 170242
1800 010616 042700 001777
1801 010622 022700 176000
1802 010626 001402
1803 010630 000000
1804 010632 000426
1805
1806 010634 017700 170214
1807 010640 032700 000100
1808 010644 001002
1809 010646 000000
1810 010650 000417
1811
1812 010652 017700 170200
1813 010656 022700 001000
1814 010662 001402
1815 010664 000000
1816 010666 000410
1817
1818 010670 017700 170164
1819 010674 042700 176000
1820 010700 022700 001000
1821 010704 001401
1822 010706 000000

```

;TEST THAT "SHIFT-OUT" GENERATES A STATUS BIT
;SHIFT-OUT (LOW BYTE), FOLLOWED BY CODE 77 (HIGH BYTE)

```

GT041 SCOPE
MOV #11000,0DBUF ;POINT MODE
MOV #1000,0DBUF1
MOV #1000,0DBUF2
MOV #1000,0DBUF3
MOV #37410,0DBUF4
MOV #17000,0DBUF5
MOV DBUF1,0DBUF ;LOAD "CHARACTER MODE"
;LOAD "SHIFT-OUT" IN LOW BYTE #77 IN HIGH BYTE
;LOAD STOP
MOV DBUF1,0DBUF
JSR 7,DELAY ;START DISPLAY
;EXECUTE A PROGRAM DELAY
MOV 0YPOS,R0 ;READ CHARACTER REG
BIC #177,R0 ;MASK TO BITS 10-19
CMP #2000,R0
BEQ ,+6
HALT ;CHARACTER REGISTER IN ERROR
; AFTER A SHIFT-OUT COMMAND
BR GT03
MOV 0DBR,R0 ;READ STATUS REGISTER
BIT #100,R0
BNE ,+6
HALT ;SHIFT OUT STATUS BIT FAILED TO SET
BR GT03
MOV 0XPOS,R0 ;READ X POS
CMP #1000,R0
BEQ ,+6
HALT ;SHIFT-OUT CHARACTER CHANGED X AXIS
BR GT03
MOV 0YPOS,R0 ;READ Y POS
BIC #17000,R0 ;MASK
CMP #1000,R0
BEQ ,+6
HALT ;SHIFT-OUT CHARACTER CHANGED Y AXIS

```

```

1824
1825
1826
1827
1828 210710 104000
1829 010712 000005
1830 010714 005003
1831 010716 012777 100000 170076
1832 010720 012777 000016 170072
1833 010722 012777 172000 170066
1834 010724 110337 013467
1835 010726 002240
1836 010728 013777 001022 170076
1837 010730 004737 012340
1838
1839 010740 032777 000100 170066
1840 010742 001402
1841 010744 000000
1842 010746 000407
1843
1844 010748 002203
1845 010750 022703 000017
1846 010752 001774
1847 010754 022703 000040
1848 010756 001353
1849
1850
1851
1852
1853
1854 011012 104000
1855 011014 000005
1856 011016 012777 100000 167776
1857 011018 012777 000016 167772
1858 011020 012777 172000 167766
1859 011022 12737 000040 013467
1860 011024 002240
1861 011026 013777 001022 167774
1862 011028 004737 012340
1863
1864 011030 032777 000100 167764
1865 011032 001002
1866 011034 000000
1867 011036 000407
1868
  
```

TEST THAT "SHIFT-OUT" DOES NOT GENERATE A STATUS BIT
 ("SHIFT-OUT" FOLLOWED BY CODE 0 THRU 37 EXCEPT #17)

GT05I SCOPE
 RESET
 CLR R3
 MOV #100000,00BUF ;SET 'CHAR! MODE
 MOV #16,00BUF1 ;LOAD "SHIFT-OUT" INTO THE LOW BYTE
 MOV #170000,00BUF2 ;
 GT05A1 MOVB R3,00FFER03 ;LOAD HIGH BYTE
 NOP
 MOV 00BUF00PC ;START THE DISPLAY
 JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
 BIT #100,00SR ;TEST FOR SHIFT BIT
 BEQ .+6
 HALT ;SHIFT STATUS BIT SET IN ERROR
 BR GT06 ; CHARACTER IS IN R3
 GT05B1 INC R3
 CHP #17,R3 ;TEST FOR "SHIFT-IN"
 BEQ GT000
 CHP #00,R3 ;TEST FOR #0
 BNE GT00A ;IS IT #0
 ;YES, NEXT TEST

TEST THAT "SHIFT-OUT" FOLLOWED BY CODE 40 GENERATE A
 SHIFT STATUS BIT

GT06I SCOPE
 RESET
 MOV #100000,00BUF ;LOAD SET 'CHAR! MODE
 MOV #16,00BUF1 ;LOAD "SHIFT-OUT" INTO THE LOW BYTE
 MOV #170000,00BUF2 ;
 GT06A1 MOVB #40,00FFER03 ;LOAD HIGH BYTE
 NOP
 MOV 00BUF00PC ;START THE DISPLAY
 JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
 BIT #100,00SR ;TEST 'SHIFT! STATUS BIT
 BNE .+6
 HALT ;"SHIFT-OUT" STATUS BIT FAILED TO SET
 BR GT07 ;ON CHARACTER IN R3

```

1870
1871 011076 000005
1872 011078 032777 000100 167746
1873 011080 001402
1874 011082 000000
1875 011084 000400
1876
1877
1878
1879
1880 011114 104000
1881 011116 012777 100000 167676
1882 011118 002077 167674
1883 011120 012777 007000 167666
1884 011122 012777 000040 167662
1885
1886 011124 012777 172000 167656
1887 011126 000005
1888 011128 002240
1889 011130 013777 001022 167666
1890 011132 004737 012340
1891
1892 011134 032777 000100 167656
1893 011136 001002
1894 011138 000000
1895 011140 000410
1896
1897 011204 017700 167650
1898 011206 042700 001777
1899 011208 022700 100000
1900 011210 001401
1901 011212 000000
1902
1903
1904
  
```

RESET
 BIT #100,00SR ;TEST SHIFT-OUT BIT
 BEQ GT07 ;IF CLEARED
 HALT ;SHIFT OUT STATUS BIT FAILED TO CLEAR
 BR GT07

TEST THAT 'SHIFT-OUT' IN THE HIGH BYTE FOLLOWED BY A CHARACTER
 IN THE NEXT LOW BYTE GENERATES A STATUS BIT

GT07I SCOPE
 MOV #100000,00BUF ;LOAD SET 'CHAR! MODE
 CLR 00BUF1
 MOV #7000,00BUF1 ;LOAD 'SHIFT-OUT' INTO THE HIGH BYTE
 MOV #40,00BUF2 ;LOAD A SHIFT-OUT CHARACTER IN THE NEXT
 ;WORD <LOW BYTE>
 MOV #170000,00BUF3
 RESET
 NOP
 MOV 00BUF00PC ;START THE DISPLAY
 JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
 BIT #100,00SR ;TEST THE STATUS REGISTER
 BNE .+6
 HALT ;SHIFT-OUT IN THE HIGH BYTE FAILED TO
 BR GT08 ;SET A STATUS BIT
 MOV 0Y003,R0 ;READ Y POS
 BIC 0177,R0 ;MASK TO BITS 19-18
 CMP #100000,R0 ;TEST FOR CHAR #40
 BEQ .+6
 HALT ;CHARACTER REGISTER IN ERROR AFTER A
 ;"SHIFT-OUT" <HIGH BYTE> FOLLOWED BY
 ; #40 <LOW BYTE NEXT WORD>

```
1976 ;STOP INTERRUPT TEST  
1977 ;TEST FOR NO INTERRUPT  
1978  
1979  
1980  
1981 011224 104000 G768I SCOPE  
1982 011226 000000 RESET  
1983 011230 012777 011312 167624 MOV #G768A,0DDONE ;LOAD RETURN FROM DONE INTERRUPT  
1984 011236 012777 011312 167626 MOV #G768A,0TIMEVT ;LOAD RETURN FROM TIME-OUT INTERRUPT  
1985 011244 012777 011312 167614 MOV #G768A,0LPVCT ;LOAD RETURN FROM LIGHT-PEN INTERRUPT  
1986 011292 012777 164000 167542 MOV #160000,0DBUF ;LOAD "DISPLAY NOP"  
1987 011200 012777 173000 167536 MOV #173000,0DBUF1 ;LOAD "STATUS A"-"STOP"-"STOP INT", ENABLE"  
1988 011206 005077 167526 CLR 0PSH ;LOWER MACHINE PRIORITY  
1989 011272 013777 001022 167552 MOV 0BUF,0DPC ;LOAD DISPLAY P.O.  
1990 011300 000240 NOP  
1991 011302 000240 NOP  
1992 011304 000240 NOP  
1993 011306 000240 NOP  
1994 011310 000401 BR ;*  
1995  
1996  
1997  
1998 011312 000000 G768A; HALT ;GT-40 INTERRUPTED IN ERROR  
1999  
2000  
2001 ;STOP INTERRUPT TEST  
2002 ;TEST FOR INTERRUPT  
2003  
2004  
2005 011314 104000 G769I SCOPE  
2006 011316 000000 RESET  
2007 011320 012777 011402 167534 MOV #G769A,0DDONE ;LOAD RETURN ADDRESS FROM INTERRUPT  
2008 011326 012777 011414 167532 MOV #G769B,0LPVCT ;LOAD LP VECTOR  
2009 011334 012777 011422 167530 MOV #G769C,0TIMEVT ;LOAD TO VECTOR  
2010 011342 012777 164000 167492 MOV #164000,0DBUF ;LOAD "DISPLAY NOP"  
2011 011350 012777 173400 167446 MOV #173400,0DBUF1 ;LOAD "STATUS A"-"STOP"-"STOP INT", ENABLE-INT"  
2012 011356 005077 167436 CLR 0PSH  
2013 011362 013777 001022 167462 MOV 0BUF,0DPC  
2014 011370 000240 NOP  
2015 011372 000240 NOP  
2016 011374 000240 NOP  
2017 011376 000240 NOP  
2018 011400 000000 HALT ;GT-40 FAILED TO GENERATE A STOP INTERRUPT  
2019 011402 013777 001064 167492 G769A; MOV 0D0001,0DDONE  
2020 011410 022626 G769A; CMP (SP)+,(SP)+  
2021 011412 000405 BR G770  
2022  
2023  
2024 011414 022626 G769B; CMP (SP)+,(SP)+  
2025 011416 000000 HALT ;GT-40 STOP (DONE) INTERRUPTED TO THE  
2026 ; LIGHT-PEN VECTOR  
2027  
2028  
2029 011400 000402 BR G770  
2030  
2031  
2032 011402 022626 G769C; CMP (SP)+,(SP)+  
2033 011404 000000 HALT ;GT-40 STOP (DONE) INTERRUPTED TO THE  
2034 ; TIME-OUT VECTOR  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054
```

```
1956  
1957 ;SHIFT OUT INTERRUPT TEST  
1958 ;TEST FOR INTERRUPT  
1959  
1960  
1961 011426 104000 G770I SCOPE  
1962 011430 000000 RESET  
1963 011432 012777 011536 167422 MOV #G770B,0DDONE ;LOAD DONE VECTOR  
1964 011440 012777 011544 167420 MOV #G770C,0LPVCT ;LOAD LIGHT-PEN VECTOR  
1965 011446 012777 011522 167416 MOV #G770A,0TIMEVT ;LOAD RETURN ADDRESS  
1966 011494 012777 100000 167340 MOV #100000,0DBUF ;LOAD "CHARACTER MODE"  
1967 011462 012777 020016 167334 MOV #020016,0DBUF1 ;LOAD "SHIFT-OUT"  
1968 011470 012777 173000 167330 MOV #173000,0DBUF2  
1969 011476 005077 167316 CLR 0PSH  
1970 011502 013777 001022 167342 MOV 0BUF,0DPC ;START DISPLAY  
1971 011510 000240 NOP  
1972 011512 000240 NOP  
1973 011514 000240 NOP  
1974 011516 000240 NOP  
1975 011518 000000 HALT ;GT-40 FAILED TO INTERRUPT ON SHIFT-OUT  
1976 011522 000240 G770A; NOP  
1977 011524 013777 001074 167340 MOV #THEVT1,0TIMEVT  
1978 011532 022626 G770A; CMP (SP)+,(SP)+  
1979 011534 000405 BR G771  
1980  
1981 011536 022626 G770B; CMP (SP)+,(SP)+  
1982 011540 000000 HALT ;GT-40 SHIFT-OUT INTERRUPTED TO THE  
1983 ; DONE VECTOR  
1984  
1985  
1986 011542 000402 BR G771  
1987  
1988 011544 022626 G770C; CMP (SP)+,(SP)+  
1989 011546 000000 HALT ;GT-40 SHIFT-OUT INTERRUPTED TO THE  
1990 ; LIGHT PEN VECTOR  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998
```

```

1990
1991
1992
1993
1994 011580 104000
1995 011582 000000
1996 011584 013777 001064 167300
1997 011586 013777 001070 167276
1998 011570 012777 011616 167274
1999 011576 000077 167216
2000 011602 012777 177776 167242
2001 011610 004737 012340
2002 011614 000000
2003
2004 011616 000240
2005 011620 013777 001074 167244
2006 011626 022626
2007
2008
2009
2100 011630 104000
2101 011632 000000
2102 011634 012777 011676 167224
2103 011642 012777 100140 167192
2104 011630 012777 173000 167146
2105 011636 000077 167136
2106 011642 013777 001022 167102
2107 011670 004737 012340
2108 011674 000401
2109 011676 000000
2120 011700 013777 001070 167100
    
```

;TIME-OUT INTERRUPT TEST

```

G771:  SCOPE
      RESET
      MOV DDONE1,DDONE
      MOV LPVGT1,0LPVCT
      MOV #G771A,0TIMEVT
      CLR 0PSH
      MOV #177776,0DPC
      JSR 7,0LAY
      HALT
    
```

;LOAD RETURN ADDRESS
;LOAD DISPLAY P.C.
;EXECUTE A PROGRAM DELAY
;GT-40 FAILED TO INTERRUPT ON TIME-OUT

```

G771A: NOP
      MOV TMEVT1,0TIMEVT
      CMP (SP)+,(SP)+
    
```

;NO LIGHT PEN INTERRUPT TEST

```

G772:  SCOPE
      RESET
      MOV #G772A,0LPVCT
      MOV #100140,0DBUF
      MOV #173000,000BUP1
      CLR 0PSH
      MOV 0BUP1,0DPC
      JSR 7,0LAY
      BR 04
G772A: HALT
      MOV LPVGT1,0LPVCT
    
```

;LOAD RETURN ADDRESS
;LOAD DISPLAY BUFFER
;EXECUTE A PROGRAM DELAY
;GT-40 INTERRUPTED ON FALSE LIGHT PEN FLAG

```

2022
2023
2024 011706 042737 177437 001004
2025 011714 001001
2026 011716 000000
2027 011720 022737 000340 001004
2028 011726 001001
2029 011730 000000
2030
2031 011732 013737 001004 011756
2032 011740 162737 000040 011756
2033 011746 013737 001004 011700
2034 011754 000402
2035
2036 011756 000140
2037 011700 000200
2038
2039
2040
2041
2042 011702 104000
2043 011704 000000
2044 011706 012777 010030 167046
2045 011774 012777 173400 167020
2046 010002 013777 011756 167010
2047 010010 013777 001022 167034
2048 010014 000240
2049 010020 000240
2050 010022 000240
2051 010024 000240
2052 010026 000000
2053
2054 010030 022626
2055
2056
2057
2058
2059 010032 104000
2060 010034 000000
2061 010036 012777 012100 167016
2062 010044 012777 173400 166750
2063 010052 013777 011700 166740
2064 010000 013777 001022 166764
2065 010006 000240
2066 010010 000240
2067 010012 000240
2068 010014 000240
2069 010016 000401
2070 010020 000000
2071 010022 013777 001064 166752
2072 010110 000000
2073 010112 000000
    
```

;PRE BR LEVEL SETUP

```

      BIC #177437,0DSPBR
      BNE 04
      MALT
      CMP #340,0DSPBR
      BNE 04
      MALT
    
```

;MARK TO BIT 0
;BR LEVEL WAS 0
;BR LEVEL WAS 7

```

      MOV 0DSPBR,0RLEV1
      SUB #40,0RLEV1
      MOV 0DSPBR,0RLEV2
      BR 0773
    
```

;BRLEV1: 140
;BRLEV2: 200

;BR LEVEL TEST (BR=1)
;TEST THAT THE GT-40

```

G773:  SCOPE
      RESET
      MOV #G773A,0DDONE
      MOV #173400,0DBUF
      MOV 0RLEV1,0PSH
      MOV 0BUP1,0DPC
      NOP
      NOP
      NOP
      HALT
    
```

;LOAD RETURN ADDRESS
;LOAD "STATUS A"-NO INTERRUPT ENABLE
;LOAD THE DISPLAY P.C.
;NO STOP INTERRUPT ON BR LEVEL INDICATED =1
;CHECK TO SEE IF PROPER BR LEVEL

```

G773A: CMP (SP)+,(SP)+
    
```

;BR LEVEL TEST (BR)
;TEST THAT THE GT-40 DOES NOT INTERRUPT AT THE LEVEL INDICATED

```

G774:  SCOPE
      RESET
      MOV #G774A,0DDONE
      MOV #173400,0DBUF
      MOV 0RLEV2,0PSH
      MOV 0BUP1,0DPC
      NOP
      NOP
      NOP
      BR 04
G774A: HALT
      MOV DDONE1,DDONE
      RESET
    
```

;LOAD RETURN ADDRESS
;LOAD "STATUS A"-STOP- STOP INT ENABLE
;LOWER MACHINE PRIORITY TO INDICATED LEVEL
;NEXT TEST
;GT-40 INTERRUPTED ON THE WRONG BR LEVEL
;LOAD INTERRUPT VECTOR

```

2075
2076
2077
2078
2079
2080 011114 104000
2081 011116 012777 117637 166676
2082 011124 009077 166674
2083 011130 009077 166672
2084 011134 012777 172077 166666
2085 011142 013777 001022 166602
2086 011190 004737 012340
2087 011194 000000
2088 011196 005777 166670
2089 011102 001402
2090 011104 000000
2091 011106 000000
2092
2093 011170 017700 166660
2094 011174 042700 074000
2095 011200 001401
2096 011202 000000
2097
2098 011204 104000
2099 011206 005237 001016
2100 011212 022737 000010 001016
2101 011208 001402
2102 011202 000137 001412
2103 011206 000000
2104 011230 013700 000042
2105 011234 001410
2106 011236 000000
2107 011240 004710
2108 011242 000240
2109 011244 000240
2110 011246 000240
2111 011208 000240
2112 011234 000240
2113 011234 000240
2114 011276 012777 000002 166670
2115 011204 012737 000207 177566
2116 011272 105737 177564
2117 011276 100375
2118 011300 012737 000207 177566
2119 011306 105737 177564
2120 011312 100375
2121 011314 000137 001336

```

IRESET TEST
DOES RESET CLEAR ALL DISPLAY PC AND STATUS BITS

```

GT75: SCOPE
MOV #117637,DOBUP ;POINT INTENSITY;BLINKS;LINETYPEs
CLR DOBUP1 ; X S B
CLR DOBUP2 ; Y S B
MOV #178077,DOBUP3 ;ITALICS;SYNCS;COLORS;
MOV DOBUP0DFC ;LOAD DISPLAY S,S;
JBR PC,DLAY ;DELAY
RESET ;GENERATE "INIT"
TST #DPE
BEQ #06 ;RESET FAILED TO CLEAR DISPLAY PC
HALT
BR END
MOV #000,RO ;READ DISPLAY STATUS
BIC #74000,RO ;MASK TO BIT 15-16
BEQ #04 ;IS THE STATUS CLEARED ?
HALT ;"INIT" FAILED TO RESET DISPLAY STATUS REGISTER

END: SCOPE
INC ICNT ;UPDATE COUNT
CMP #10,ICNT ;FINISHED ?
BEQ HERE ;BR IF YES
JMP GTRG ;NO RE-DO

HERE: RESET
MOV #40,RO ;BRANCH IF OFF LINE
BEQ HERE1
RESET

LOGICAL: JBR PC,00)

NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
MOV #2,ROSR ;RING THE BELL
MOV #207,TPDBR ;PRINT THE BELL
TSTB TPDBR ;WAIT
SPL 15
MOV #207,TPDBR ;PRINT BELL
TSTB TPDBR
SPL 25
JMP START

```

```

2123
2124
2125 011320 032737 040000 177570 SCOPEA) BIT #40000,0#0,0PLAY ;TEST "SCOPE" SWITCH
2126 011336 001001 BNE SCOPES
2127 011330 011601 MOV (#0)R1
2128 011302 012706 SCOPEB) MOV #STNPTR,SP
2129 011336 000111 JMP (#1)
2130
2131 011340 012700 000200 DLAY) MOV #200,RO
2132 011344 005300 R0 DEC
2133 011346 001376 BNE DLAYA
2134 011330 000207 RTS
2135
2136 011302 012700 001000 DLAY1) MOV #1000,RO
2137 011306 005300 DLAY1A) DEC R0
2138 011300 001376 BNE DLAY1A
2139 011302 000207 RTS
2140
2141 011304 010046 LONPHR) MOV R0,(SP)
2142 011306 010146 MOV R1,(SP)
2143 011370 010246 MOV R2,(SP)
2144 011372 010346 MOV R3,(SP)
2145 011374 010446 MOV R4,(SP)
2146 011376 010546 MOV R5,(SP)
2147 011400 010637 012462 MOV #0,LOMSV
2148 011404 012737 012414 000024 MOV #HIOPHR,0004
2149 011412 000000 HALT
2150 011414 013706 012462 HIOPHR) MOV (SP)+,R0
2151 011420 012609 MOV (SP)+,R1
2152 011422 012604 MOV (SP)+,R2
2153 011424 012603 MOV (SP)+,R3
2154 011426 012602 MOV (SP)+,R4
2155 011430 012601 MOV (SP)+,R5
2156 011432 012600 MOV (SP)+,R0
2157 011434 012737 012364 000024 LONPHR) MOV #STNPTR,SP
2158 011442 012706 000500
2159 011446 000240
2160 011430 000240
2161 011452 000000 HALT
2162 011454 000240
2163 011456 000240
2164 011400 000111
2165
2166 011462 000000 LOMSV) 0
2167

```

```

2169
2170 012464 012777 000340 166326 START1) MOV #340,PPSW
2171 012472 012706 000500 MOV #STKPTR,SP
2172 012476 004737 001076 JSR PC,SETUP
2173 012502 012704 012464 MOV #START1,R1
2174 012506 012777 012542 166346 MOV #SPCA,DDONE ISET UP DONE VECTOR
2175 012514 013777 001004 166342 MOV #SPRR,DDONE1
2176 012522 012777 012616 166336 MOV #SPACE,PLPVCT ISET UP LIGHT=PEN VECTOR
2177 012530 013777 001004 166332 MOV #SPRR,PLPVCT1
2178 012536 000240 NOP
2179 012540 000240 NOP
2180 012542 012706 000500 SPCEA1 MOV #STKPTR,SP
2181 012546 012737 173400 013426 MOV #DSTOP,FRM7A
2182 012554 017700 166260 MOV #SWR,R0
2183 012560 000100 ROL R0
2184 012502 042700 177761 BIC #177761,R0
2185 012506 016002 012636 MOV #DSP(CH(0)),R2
2186 012572 010277 166254 MOV R2,DDPC ISTART THE DISPLAY
2187
2188 012576 000240 NOP
2189 012600 000240 NOP
2190 012602 005077 166212 SPCEC1 CLR #PSW
2191 012606 000001 WAIT
2192 012610 000240 NOP
2193 012612 000240 NOP
2194 012614 000752 BR SPCEA
2195
2196 012616 012737 164000 013426 SPACEB) MOV #DNDP,FRM7A
2197 012624 012777 000001 166220 MOV #1,DDPC I SINGLE STEP THE DISPLAY
2198 012632 022626 CMP #SP1,(SP)+
2199 012634 000702 BR SPCEC
2200
2201
2202 012636 012656 DSPTCH0 FRAME0
2203 012640 012704 FRAME1
2204 012642 012732 FRAME2
2205 012644 012760 FRAME3
2206 012646 013000 FRAME4
2207 012650 013044 FRAME5
2208 012652 013132 FRAME6
2209 012654 013356 FRAME7
2210
2211

```

```

2213
2214
2215 012686 117004 FRAME0) POINT,INT4,LINE0
2216 012690 041000 INTX+1000
2217 012692 000600 000
2218 012694 041000 INTX+1000
2219 012696 000600 000
2220 012698 112400 LONGV,INT2
2221 012702 040600 INTX+600
2222 012704 000000 0
2223 012706 173400 DSTOP
2224 012708 160000 DJMP
2225 012710 012656 FRAME0
2226
2227 012714 117004 FRAME1) POINT,INT4,LINE0
2228 012716 040200 INTX+200
2229 012718 000600 000
2230 012720 041000 INTX+1000
2231 012722 000600 000
2232 012724 112400 LONGV,INT2
2233 012726 000600 INTX+MINUS+600
2234 012728 000000 0
2235 012730 173400 DSTOP
2236 012732 160000 DJMP
2237 012734 012704 FRAME1
2238
2239 012736 117004 FRAME2) POINT,INT4,LINE0
2240 012738 041000 INTX+1000
2241 012740 001200 1200
2242 012742 041000 INTX+1000
2243 012744 000600 000
2244 012746 112400 LONGV,INT2
2245 012748 040000 INTX
2246 012750 000400 400
2247 012752 173400 DSTOP
2248 012754 160000 DJMP
2249 012756 012732 FRAME2
2250
2251 012758 117004 FRAME3) POINT,INT4,LINE0
2252 012760 041000 INTX+1000
2253 012762 000200 200
2254 012764 041000 INTX+1000
2255 012766 000600 000
2256 012768 112400 LONGV,INT2
2257 012770 040000 INTX
2258 012772 020400 MINUS+400
2259 012774 173400 DSTOP
2260 012776 160000 DJMP
2261 012778 012760 FRAME3
2262

```

2264					FRAME4 POINT:INT4:LINE0
2265	013006	117004			0
2266	013010	000000			0
2267	013012	000000			LONGV
2268	013014	110000			INTX:MAXX
2269	013016	041777			0
2270	013020	000000			INTX
2271	013022	040000			MAXY
2272	013024	001377			INTX:MINUSX:MAXX
2273	013026	061777			0
2274	013030	000000			INTX
2275	013032	040000			MINUSX:MAXY
2276	013034	021377			DSTOP
2277	013036	173400			DJMP
2278	013040	160000			FRAME4
2279	013042	013006			
2280					
2281	013044	117004			FRAME5 POINT:INT4:LINE0
2282	013046	000740			740
2283	013050	000540			540
2284	013052	104000			SHORTV
2285	013054	057600			INTX*17600
2286	013056	057677			INTX*17677
2287	013060	040077			INTX*77
2288	013062	077677			INTX:MINUSX*17677
2289	013064	077600			INTX:MINUSX*17600
2290	013066	077777			INTX:MINUSX*17777
2291	013070	040177			INTX*177
2292	013072	057777			INTX*17777
2293	013074	114000			POINT
2294	013076	000740			740
2295	013100	000620			020
2296	013102	130000			RELATV
2297	013104	047600			INTX*7600
2298	013106	047637			INTX*7637
2299	013110	040037			INTX*37
2300	013112	067637			INTX:MINUSX*7637
2301	013114	067600			INTX:MINUSX*7600
2302	013116	067737			INTX:MINUSX*7737
2303	013120	040137			INTX*137
2304	013122	047737			INTX*7737
2305	013124	173400			DSTOP
2306	013126	160000			DJMP
2307	013130	013044			FRAME5
2308					

2310					FRAME6 POINT:INT4:LINE0
2311	013132	117004			200
2312	013134	000200			600
2313	013136	000600			CHAR
2314	013140	100000			
2315					
(1)	013142	040500	041502	042504	.ASCII 'ABCDEFGHIJKLWNPQRSTUVWXYZ'
(1)	013150	043506	044510	045512	
(1)	013156	046514	047516	048520	
(1)	013164	051522	052524	053526	
(1)	013172	054530	132		
(1)	013175	040	021041	022043	.ASCIZ 'INSTR(())*,./0123456789!*<>?@
(1)	013202	023048	024047	025051	
(1)	013210	026053	027055	028057	
(1)	013216	031061	032063	033065	
(1)	013224	034067	035071	036073	
(1)	013232	037075	000077		
(1)					.EVEN
2316	013236	114000			POINT
2317	013240	000200			200
2318	013242	000540			540
2319	013244	100000			CHAR
2320					
(1)	013246	140	141	142	.BYTE 140,141,142,143,144,145,146,147
(1)	013251	143	144	145	
(1)	013254	146	147		
(1)	013256	150	151	152	.BYTE 150,151,152,153,154,155,156,157
(1)	013261	153	154	155	
(1)	013264	156	157		
(1)	013266	160	161	162	.BYTE 160,161,162,163,164,165,166,167
(1)	013271	163	164	165	
(1)	013274	166	167		
(1)	013276	170	171	172	.BYTE 170,171,172,173,174,175,176,177
(1)	013301	173	174	175	
(1)	013304	176	177		
2321					
(1)	013306	016	000	001	.BYTE 16,001,2,3,4,5,6,7,10,11,12,13,14,15,16
(1)	013311	002	003	004	
(1)	013314	005	006	007	
(1)	013317	010	011	012	
(1)	013322	013	014	015	
(1)	013325	016			
(1)	013326	020	021	022	.BYTE 20,01,22,23,24,25,26,27,30,31,32,33,34,35,36,37,17,0
(1)	013331	023	024	025	
(1)	013334	026	027	028	
(1)	013337	031	032	033	
(1)	013342	034	035	036	
(1)	013345	037	017	000	
(1)					.EVEN
2322	013350	173400			DSTOP
2323	013352	160000			DJMP
2324	013354	013132			FRAME6
2325					

```

2327
2328 013396 117004 FRAME7 POINTINT4LINE0
2329 013390 000400 400
2330 013392 000700 700
2331 013304 170200 STAYSAILPLIIE
2332 013306 110140 LONGVILPON
2333 013370 041000 INTX+1000
2334 013372 000000 0
2335 013374 114000 POINT
2336 013376 000400 400
2337 013400 000600 600
2338 013402 170300 STAYSAILPDARK
2339 013404 110140 LONGVILPON
2340 013406 041000 INTX+1000
2341 013410 000000 0
2342 013412 114000 POINT
2343 013414 000400 400
2344 013416 000500 500
2345 013420 110100 LONGVILPOFF
2346 013422 041000 INTX+1000
2347 013424 000000 0
2348 013426 173400 PRM7A1 DSTOP
2349 013430 114000 POINT
2350 013432 000700 700
2351 013434 001000 1000
2352 013436 100000 CHAR
2353 013440 044514 044107 026524 .ASCIZ /LIGHT=REN HIT/
      013446 042520 020116 044910
      013494 000124
2354
2355 013496 173400 .EVEN
2356 013498 160000 DSTOP
2357 013402 013356 DJMP
2358
2359 013464 000000 BUFFER4 0
2360
2361 000001 .END
    
```

BLKOFF	020020	BLKON	000030	BRLEV1	011750	BRLEV2	011760
BUFFER	013464	CHAR	100000	CHSIRE	001050	CNTR	001044
DBUF	001020	DBUF1	001024	DBUF2	001020	DBUF3	001020
DBUF4	001032	DBUF5	001034	DDONE	001002	DBUF4	001020
DIBPLA	177970	DJMP	100000	DLAY	013340	DDONE1	001006
DLAY1	012352	DLAY1A	012356	DNOP	100000	DLAYA	012344
DRC	001052	DSAVE	001036	DSPDR	001004	DCCCR	001220
DNR	001054	DSTOP	173400	END	011204	DBRCH	010030
FRAME1	012704	FRAME2	012732	FRAME3	012760	FRAME5	012656
FRAME5	013044	FRAME6	013132	FRAME7	013336	FRAME4	013006
GRAPHX	120000	GRAPHY	104000	GREEN	000002	PRM7A	013426
GSCHSZ	001006	GSLPSE	001010	GSEND	001014	GSADD	001000
GSYAXS	001012	GTBUSS	001044	GTPC	001014	GSVCT	001002
GT2	001032	GT1	001030	GT2	001012	GTPOA	001054
GT12	002350	GT13	002420	GT14	001046	GT11	002312
GT16	002366	GT17	002432	GT15	001046	GT10	002026
GT2	001026	GT20	003006	GT16	001046	GT19	002742
GT23	003260	GT24	003226	GT21	001050	GT22	003116
GT27	003420	GT28	003474	GT22	001072	GT26	003336
GT30	003600	GT31	003672	GT23	001076	GT3	001064
GT34	004356	GT32	004254	GT24	001076	GT33	004000
GT38	004550	GT33	004054	GT25	001076	GT37	004490
GT40	004770	GT40A	005010	GT26	001076	GT4	001726
GT43	005306	GT43A	005326	GT27	001076	GT42	005210
GT45	005350	GT46	005000	GT28	001076	GT44A	005450
GT48	006070	GT48A	006110	GT29	001076	GT47A	005766
GT5	001077	GT50	006326	GT29A	001076	GT49A	006234
GT51	006470	GT52	006340	GT30	001076	GT50B	006444
GT55	007200	GT56	007270	GT31	001076	GT54	007040
GT59	007500	GT6	002026	GT32	001076	GT58	007424
GT62	010056	GT62A	010016	GT33	001076	GT61	007710
GT64	010030	GT65	010070	GT34	001076	GT63A	010010
GT66	011010	GT66A	011040	GT35	001076	GT60B	010070
GT68A	011312	GT69	011314	GT36	001076	GT60	011224
GT69C	011420	GT7	002072	GT37	001076	GT69B	011410
GT70R	011934	GT70C	011944	GT38	001076	GT70A	011920
GT72	011630	GT72A	011076	GT39	001076	GT71A	011010
GT74	012032	GT74A	012100	GT39A	001076	GT73A	012030
GT9	002200	HERE	012226	GT40	001076	GT8	002136
ICNT	001016	INCR	000100	GT41	001076	WISPR	010416
INT1	002200	INT2	002400	GT42	001076	INT0	002000
INT5	003200	INT6	003400	GT43	001076	INT4	003000
ITAL1	000060	LFSIZE	001046	GT44	001076	ITAL0	000040
LINE2	000006	LINE3	000007	GT45	001076	LINE1	000000
LOWPWR	012864	LOWSV	012462	GT46	001076	LONGV	110000
LPVCT	000200	LPON	000140	GT47	001076	LPLITE	000200
MAXSX	001760	MAXSY	000077	GT48	001076	LVPCT1	001070
MINBUY	000100	MINUSX	000000	GT49	001076	MAXY	001377
POINT	114000	PSW	001020	GT50	001076	PC	000000
R0	000000	R1	000000	GT51	001076	RELATV	130000
R4	000004	R5	000000	GT52	001076	R3	000000
SCOPEB	012332	SETUP	001076	GT53	001076	SCOPEA	012320
SHORTV	124000	SIRE	001042	GT54	001076	SETUPB	001132
SP	000006	SPACEB	012016	GT55	001076	SIRE1	000000
				GT56	001076	SPCEC	012002

BT=42/RT=44 INSTRUCTION TEST II MAINDEC=11-DDGTB=B MACY11.024 25-NOV-73 10190 PAGE 93.2
DDGTB=011 SYMBOL TABLE

START	001336	START1	012464	STARTSA	= 170000	STARTSB	= 174000
SYKPTR	= 000000	SWR	001040	SYNOFF	= 000010	SYNON	= 000014
TMEVT	001070	TMEVT1	001074	TPCSR	= 177564	TPDR	= 177566
YPOS	001056	YPOS	001060	.	= 013460		

ERRORS DETECTED: 0

REM *

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DDGTG-A-D
PRODUCT NAME: GT48/GT44 VISUAL DISPLAY TEST
WITH VR17 DISPLAY
DATE CREATED: NOVEMBER 11 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 CONTAINS A SERIES OF PATTERNS THAT ARE USED AS AIDS IN THE ALIGNMENT AND ADJUSTMENT OF THE GT40/GT44 DISPLAY SYSTEM WITH A VR17. FOR THIS TEST THE MAINTENCE SWITCHES ARE NOT USED (NORMAL POSITION).

2. REQUIREMENTS

2.1 EQUIPMENT

GT40 DISPLAY SYSTEM WITH VR17 DISPLAY SCOPE OR
GT44 DISPLAY SYSTEM WITH VR17 DISPLAY SCOPE.

2.2 STORAGE

THIS PROGRAM USES LESS THAN 4K OF MEMORY.

2.3 PRELIMINARY PROGRAMS

ALL PROCESSOR MAINDECS, GT40/GT44 INSTRUCTION TEST I AND
GT40/GT44 INSTRUCTION TEST II MUST HAVE RUN IN THEIR
ENTIRETY BEFORE ATTEMPTING TO RUN THIS TEST.

3. LOADING PROCEDURE

3.1 METHOD

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

4. STARTING PROCEDURE

4.1 STARTING ADDRESS

LOAD ADDRESS 0000
START WITH SWITCHES 7#0, 0#0 FOR AUTO SEQUENCING
THRU ALL NON-OPERATOR INTERVENTION PATTERNS.
START WITH SWITCH BIT 7#0, 0#1 FOR SWITCH REGISTER PATTERN
CONTROL (REF 4.2).
START WITH SWITCH BIT 7#1, 0#0 OR 1 FOR KEYBOARD PATTERN
CONTROL (REF 4.3).

4.2 CONTROL SWITCH SETTINGS (SWITCH REGISTER)

SWITCH REGISTER BITS 0,1,2,3 ARE USED TO SELECT EACH OF THE TESTS.

NON-OPERATOR INTERVENTION TESTS

SW 3 = 0 00 /DIRECTORY
 01 /DOT REPEATABILITY
 02 /PINCUSHION (X AND Y OFFSET ADJ.)
 03 /OCTAGONS OR SQUARES
 04 /CHARACTER SET (CHAR ADJ.)
 05 /DASH LINES AND BLINK
 06 /VECTOR LENGTH TEST < X VECTOR LENGTH ADJ.>
 07 /VECTOR LENGTH TEST < Y VECTOR LENGTH ADJ.>
 10 /PHOSPHOR TEST (SHORTER)
 11 /PHOSPHOR TEST (VERT)
 12 /INTENSITY LEVELS, SYNC AND LIGHT-PEN TEST
 13 /EDGE TEST
 14 /SHORT VECTOR AND RELATIVE POINT TEST
 15 /GRAPHPLOT INCREMENT TEST

OPERATOR INTERVENTION TESTS

 16 /LIGHT-PEN FOLLOW TEST
 17 /KEYBOARD ECHO

SW 6 = 0 SELECT SUB-PICTURE 0
SW 6 = 1 SELECT SUB-PICTURE 1 OR
 STOP DISPLAY FRAME MOTION

SW 8 = 0 EXECUTE ALL NON-OPERATOR INTERVENTION FRAMES;
SW 8 = 1 EXECUTE THE DISPLAY FRAME SPECIFIED BY SW 0-3.

4.3 CONTROL SWITCH SETTINGS (DISPLAY KEYBOARD)

ALPHA CHARACTERS (A) THRU (P) ARE USED TO SELECT EACH OF THE TESTS.

CHARACTER	TEST
A	DIRECTORY
B	DOT REPEATABILITY
C	PINGUSHION (X AND Y OFFSET ADJ.)
D	OCTAGONS OR SQUARES
E	CHARACTER SET (CHAR. ADJ.)
F	DASH LINES AND BLINK
G	VECTOR LENGTH TEST (X VECTOR LENGTH ADJ.)
H	VECTOR LENGTH TEST (Y VECTOR LENGTH ADJ.)
I	PHOSPHOR TEST (HORIZ)
J	PHOSPHOR TEST (VERT)
K	INTENSITY LEVELS, SYNC AND LIGHT-PEN TEST
L	EDGE TEST
M	SHORT VECTOR AND RELATIVE POINT
N	GRAPHPLOT INCREMENT TEST
O	LIGHT-PEN FOLLOW TEST
P	KEYBOARD ECHO

DEPRESSING A 'RUBOUT' AFTER SELECTING A FRAME WILL LOCK ON THE SELECT FRAME.

DEPRESSING A 'ER' AFTER SELECTING A FRAME WILL SELECT SUBPICTURE 1 OR STOP DISPLAY FRAME MOTION.

TO CONTINUE AFTER DEPRESSING A 'CR' OR 'RUBOUT' DEPRESS ANY KEY OTHER THAN 'CR' OR 'RUBOUT'.

DEPRESSING 'CONTROL C (<C>)' WHEN EXECUTING THE KEYBOARD ECHO TEST, WILL RETURN CONTROL TO THE DIRECTORY FRAME.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCHES

ALL OF THE TEST WILL RUN IN THEIR NORMAL MANNER WITHOUT ANY OPERATIONAL SWITCHES SELECTED, HOWEVER, SOME OF THE TESTS HAVE ADDITIONAL FEATURES AND THE ARE SELECTED BY USING SWITCH BIT 06 OR "CR" KEYBOARD KEY.

5.1.1 PINGUSHION TEST

SW 6 # 0 DISPLAY PINGUSHION
SW 6 # 1 DISPLAY CROSSHATCH <IN-HOUSE TEST ONLY>

5.1.2 OCTAGON OR SQUARES

SW 6 # 0 DISPLAY OCTAGONS
SW 6 # 1 DISPLAY SQUARES

5.1.3 VECTOR LENGTH TEST

SW 6 # 0 SWEEP MOVEMENT
SW 6 # 1 STOP MOVEMENT

5.1.4 PHOSPHOR TEST

SW 6 # 0 SWEEP ACROSS THE SCREEN
SW 6 # 1 STOP MOVEMENT

5.1.5 INTENSITY TEST

SW 6 # 0 ENABLE SYNC 'OFF'
SW 6 # 1 ENABLE SYNC 'ON'

5.1.6 GRAPHPLT INCREMENT TEST

SW 6 # 0 USE GRAPHPLT X
SW 6 # 1 USE GRAPHPLT Y

5.1.7 LIGHT PEN FOLLOW

SW 6 # 0 DISPLAY LIGHT PEN FOLLOW
SW 6 # 1 DISPLAY LIGHT PEN FIELD OF VIEW
 <IN-HOUSE TEST ONLY>

6. ERRORS

THE PROGRAM WILL ONLY HALT ON ERROR,
THE PROGRAM DOES NOT CONTAIN FACILITES FOR THE REPORTING OF ERROR
CONDITIONS;

7. RESTRICTIONS

IF USING THE SWITCH REGISTER (REF #,2) TO CONTROL THE PROGRAM, THERE WILL BE A DELAY BEFORE THE NEW TEST IS SELECTED.

8. MISCELLANEOUS

8.1 DEVICE ADDRESS PROGRAM LOCATIONS

LOCATION 1000 CONTAINS THE GT40/GT44 DEVICE ADDRESS;
LOCATION 1002 CONTAINS THE GT40/GT44 INTERRUPT VECTOR,
LOCATION 1004 CONTAINS THE GT40/GT44 INTERRUPT BR LEVEL.

9. PROGRAM DESCRIPTION

9.1 DIRECTORY

THIS TEST USES THE CHARACTER MODE TO DISPLAY A DIRECTORY OF THE TESTS THAT ARE AVAILABLE.

9.2 DOT REPEATIBILITY

THIS TEST INTENSIFIES A DOT IN EACH CORNER AND A DOT IN THE CENTER OF THE SCREEN; THIS TEST IS USED TO VERIFY DOT REPEATIBILITY.

9.3 PINGUISHION AND VECTOR CURVATURE TEST (ADJUSTMENT OF X AND Y OFFSET POTS)

THIS TEST OUTLINES THE FULL SCREEN AREA, IT IS USEFUL IN CENTERING THE VIEWING AREA IN THE DISPLAY MASK, THIS TEST ALSO DRAWS A DIAGONAL LINE FROM LOWER LEFT CORNER TO THE UPPER RIGHT AND THEN RETURNS IN THE OPPOSITE DIRECTION, A SIMILAR SEQUENCE IS REPEATED STARTING AT LOWER RIGHT CORNER TO THE UPPER LEFT CORNER AND BACK. THE PURPOSE IS TO MAKE CERTAIN THAT THE VECTORS ARE LINEAR OVER THEIR ENTIRE LENGTH, WITH PROPER LENGTH VECTORS ONLY TWO DIAGONAL LINES SHOULD BE SEEN IN THE CENTER OF THE SCREEN, DO NOT ADJUST THE VECTOR LENGTH POTS WITH THIS DISPLAY PATTERN, SINGLE LINES SHOULD BE VISABLE AT THE TOP AND BOTTOM OF THE SCREEN, IF NOT ADJUST THE Y OFFSET POT, SINGLE LINES SHOULD BE VISABLE AT THE RIGHT AND LEFT EDGE OF THE SCREEN IF NOT ADJUST THE X OFFSET POT..

9.4 OCTAGONS OR SQUARES

A SERIES OF DIFFERENT SIZE OCTAGONS OR SQUARES ARE DRAWN TO DEMONSTRATE THAT CLOSED FIGURES CAN BE DRAWN USING DIFFERENT VECTOR LENGTHS (7,17,37,73,177,377 AND 797), THIS TEST IS USED TO TEST THE END POINT MATCHING OF THE VECTORS.

- 9.5 CHARACTER SET (ADJUSTMENT OF THE CHARACTER POTIS)
TWO COMPLETE SETS OF ASCII CHARACTERS AVAILABLE FROM THE CHARACTER GENERATOR ARE DISPLAYED. THE CHARACTERS ARE DISPLAYED IN FOUR LINES OF TEXT. THE FIRST HALF OF A LINE IS IN 'NORMAL' FONT. THE SECOND HALF OF A LINE IS IN 'ITALICS' FONT.
- 9.6 DASH LINES AND BLINK TEST
THIS TEST IS USED TO TEST THE FOUR TYPES OF VECTOR LINES. FOUR VECTORS ARE PLOTTED USING EACH OF THE FOUR LINE REGISTER VALUES. THIS TEST ALSO ENABLES THE BLINK OPTION. THE FIRST VECTOR ON A LINE SHOULD NOT BLINK. THE SECOND VECTOR ON A LINE SHOULD BLINK.
- 9.7 VECTOR LENGTH TEST (ADJUSTMENT OF X AND Y VECTOR LENGTH)
A SERIES OF INCREMENTING ANGLE VECTORS ARE DRAWN FROM THE SCREEN ORIGIN TO THE OPPOSITE EDGE OF THE SCREEN. THESE VECTORS SHOULD TERMINATE ON THE LINE DRAWN AT THE VIEWING EDGE. IF THE VECTORS DO NOT END ON THE LINE, ADJUST THE APPROPRIATE VECTOR LENGTH POT.
- 9.8 PHOSPHOR TEST
A WIDE BAND OF INTENSIFIED VECTORS IS DISPLAYED TO ALLOW FOR VISUAL INSPECTION OF THE CRT PHOSPHOR. THIS TEST ALSO TEST FOR ANY DISTORTION IN DEFLECTION CROSS-OVER IN THE SCORE.
- 9.9 INTENSITY LEVEL, SYNC AND LIGHT-PEN SENSITIVITY TEST
EIGHT VECTORS ARE DRAWN USING EACH OF THE EIGHT INTENSITY LEVELS. THE INTENSITY SHOULD BE ADJUSTED SO THAT THE LEVEL 0 IS BARELY VISIBLE. THIS TEST IS ALSO USED TEST THE LIGHT PEN SENSITIVITY. ALL LINES ARE SET TO ALLOW A LIGHT PEN HIT. THEN HIT THE MESSAGE 'LIGHT PEN HIT' WILL BE DISPLAYED ON THE LINE HIT. THIS TEST IS ALSO USED TO TEST THE 'SYNC' LOGIC IF SELECTED.
- 9.10 EDGE SQUARES TEST
THIS TEST IS USED TO TEST FOR PROPER EDGE BLANKING AND REENTRY SETTLE TIME. THE SCREEN IS OUTLINED AND FOUR RECTANGLES ARE DRAWN AS TO EXCEED THE EDGE OF THE SCREEN. ONLY HALF OF EACH RECTANGLE SHOULD BE VISIBLE.

9.11 SHORT VECTOR AND RELATIVE POINT TEST

THIS TEST IS USED TO VERIFY PROPER DECODING OF THE SHORT VECTOR AND RELATIVE POINT. A SERIES OF INTENSIFIED VERTICAL LINES ARE PLOTTED USING SHORT VECTOR MODE. THE TEST THEN REPEATS USING RELATIVE POINT. THE RESULTS IS THAT A SINGLE HORIZONTAL LINE APPEARS TO THE RIGHT OF THE VERTICAL LINES. ALSO INCLUDED IS A RELATIVE POINT REPEATABILITY TEST. FOUR SETS OF THREE OCTAGONS EACH WILL BE DISPLAYED. THE INNER OCTAGON IS DRAWN USING SHORT VECTOR MODE WITH A DELTA X, Y OF 71 OCT. THE MIDDLE OCTAGON IS DRAWN USING RELATIVE POINT MODE WITH A DELTA X, Y OF 74 OCT. THE OUTER OCTAGON IS DRAWN USING SHORT VECTOR MODE WITH AN DELTA X, Y OF 77 OCT. THE MIDDLE OCTAGON SHOULD BE EQUAL DISTANCE FROM THE OUTER OCTAGONS AND SHOULD NOT MOVE.

9.12 GRAPHLOT INCREMENT TEST

A SERIES OF POINTS ARE PLOTTED WITH EACH POSSIBLE VALUE IN THE GRAPHLOT INCREMENT REGISTER FROM 0-777. THE RESULTING PATTERN USED SHOULD APPEAR TO BE A SERIES OF POINTS AT AN INCREASING ANGLE.

9.13 LIGHT-PEN FOLLOW TEST

IN THIS OPERATOR INTERVENTION TEST A TRACKING CROSS IS DISPLAYED. THE OPERATOR MAY MOVE ACROSS THE SCREEN WITH THE LIGHT PEN. AN X AND Y OCTAL READOUT IS ALSO DISPLAYED TO THE OPERATOR.

9.14 KEYBOARD ECHO TEST

THIS IS AN OPERATOR INTERVENTION TEST USED TO INSURE PROPER OPERATION OF THE DISPLAY KEYBOARD. WHEN A DISPLAYABLE CHARACTER KEY IS DEPRESSSED THE CHARACTER IS DISPLAYED ON THE SCREEN. IN SELECTING THE SHIFT-OUT MODE, IF THE KEY DEPRESSSED IS NOT A CONTROL CHARACTER, THE PROGRAM WILL TRAP TO THE SHIFT-OUT VECTOR. AN OCTAL CHARACTER VALUE READOUT IS ALSO DISPLAYED AS AN AID IN ADJUSTING THE TTY CLOCK.

.LIST

368
369
370

```

372
373
374          :ENABL  ABS: AHA
          :TITLE  GT=44/RT=44 WITH VR17 VISUAL DISPLAY TEST MAINDEC=11-DDGTG=A
375          :LIST  ME
376          :NLIST  MC,MD,CND
377
378          R0=X0
379          R1=X1
380          R2=X2
381          R3=X3
382          R4=X4
383          R5=X5
384          SP=X6
385          PC=X7
386          SCOPE=ENT
387          STKPTR=300
388          DISPLAY=177570          ;11/45 LIGHT DISPLAY REGISTER
389
390          ;0=776 IS FILLED WITH ;02, HALT
391          :LIST
392
401          ;=24
402          :WORD  LOWPWR
403          340
404
405          ;=30
406          :WORD  SCOPEA          ;EMT RETURN
407          340
408

```

```

410
411          ;=200
412          JMP  START          ;DISPLAY TEST
413
414          ;=1000
415          GSADD; 172000          ;DISPLAY STARTING ADDRESS
416          GSVCT; 320          ;DISPLAY INTERRUPT VECTOR STARTING ADDRESS
417          GSRLL; 200          ;DISPLAY BR LEVEL
418
419          ICNTI  0
420          PSW; 177776
421          TKS; 177560
422          TKD; 177562
423          DBUF1; BUFFER          ;FIRST WORD IN THE DISPLAY BUFFER
424          DBUF2; BUFFER+2          ;SECOND WORD
425          DBUF3; BUFFER+4          ;THIRD WORD
426          DBUF4; BUFFER+6          ;FOURTH WORD
427          DBUF5; BUFFER+10         ;FIFTH WORD
428          DBUF6; BUFFER+12
429          DSAVE1; 0          ;TEMP REG.
430          DSAVE2; 0
431          DSAVE3; 0
432          DSAVE4; 0
433          HOLD1; 0
434          TSAVE1; 0
435          CNTRI; 0
436          CHANGE1; 0
437          LOKRB; 0
438
439
440          ;GS ADDRESSES AND INTERRUPT VECTORS
441
442          DPCI; 172000          ;DISPLAY PROGRAM COUNTER
443          DSRI; 172002          ;DISPLAY STATUS REGISTER
444          XPSI; 172004          ;DISPLAY X AXIS REGISTER
445          YPSI; 172006          ;DISPLAY Y AXIS REGISTER
446
447          DDONE1; 320          ;DISPLAY INTERRUPT VECTOR FOR STOP
448          DDONE1; 322
449
450          LPVCT1; 324          ;DISPLAY INTERRUPT VECTOR FOR LIGHT-PEN
451          LPVCT1; 326
452
453          TIMEVT1; 330          ;DISPLAY INTERRUPT VECTOR FOR TIME=QUI OR SHIFT-OUT
454          TIMEVT1; 332
455

```

```

461                                ;MONITOR ROUTINE
462
463 001100 005737 002046      SCOPEA1 TST   KR00          ;TEST IF SW OR "KR0"
464 001104 001014                BNF   SCOPEF          ;BR IF "KR0"
465 001106 005037 005556      CLR   SWITCH        ;CLEAR "SWITCH"
466 001102 032737 000100 177570 BIT   #10070#DISPLAY ;TEST FOR "HOLD/STOP SWITCH"
467 001120 001402                BEQ   SCOPEE          ;BR IF CLEARED
468 001122 005137 005556      COM   SWITCH        ;SET SWITCH
469 001126 032737 000400 177570 SCOPEE1 BIT   #4008#DISPLAY ;TEST BIT 8
470 001134 001010                BNE   SCOPEB          ;BR IF SET
471 001136 005737 001042      SCOPEF1 TST   HOLD          ;TEST FOR "HOLD/STOP"
472 001142 001012                BNF   SCOPEO          ;BR IF SET
473 001144 000240                NOP
474 001146 004737 001536      JSR   PC,SETUP        ;RESET HOUSEKEEPING
475 001152 000240                NOP
476 001154 000002                RTI
477 001156 013704 177570      SCOPEB1 MOV   #0#DISPLAY,R4 ;READ SWITCHES
478 001162 042704 177570      SCOPEC1 BIC   #177760,R4 ;MASK TO BITS 4-15
479 001166 006304                ASL   R4              ;MOVE LEFT
480 001170 012706 000500      SCOPED1 MOV   #STKPTR,SP ;RESET STACK
481 001174 000240                NOP
482 001176 004737 001536      JSR   PC,SETUP        ;RESET HOUSEKEEPING
483 001202 000240                NOP
484 001204 000174 001210      JMP   #DIPTR(R4)     ;JMP TO THAT TEST
485
486 001210 002052      DISPTC1 FILE0+2      ;DIRECTORY
487 001212 002064      FILE1+2              ;DOT REPEATABILITY
488 001214 002076      FILE2+2              ;PINCUSHION
489 001216 002342      FILE3+2              ;OCTAGONS OR SQUARES
490 001220 002416      FILE4+2              ;CHARACTER SET
491 001222 003026      FILE5+2              ;DASH LINES AND BLINK
492 001224 003040      FILE6+2              ;X VECTOR LENGTH
493 001226 003172      FILE7+2              ;Y VECTOR LENGTH
494 001230 003324      FILE10+2             ;X PHOSPHOR TEST
495 001232 003400      FILE11+2             ;Y PHOSPHOR TEST
496 001234 003454      FILE12+2             ;INTENSITY LEVEL AND LIGHTPEN
497 001236 003616      FILE13+2             ;EDGE SQUARES
498 001240 003630      FILE14+2             ;SHORT VECTOR RELATIVE POINT TEST
499 001242 004110      FILE15+2             ;GRAPH PLOT TEST
500 001244 004344      FILE16+2             ;LIGHT-PEN FOLLOW
501 001246 005054      FILE17+2             ;KEY BOARD ECHO
502

```

```

504
505
506 001250 010046      LONPHR1 MOV   R0,#(SP)
507 001252 010146      MOV   R1,#(SP)
508 001254 010246      MOV   R2,#(SP)
509 001256 010346      MOV   R3,#(SP)
510 001260 010446      MOV   R4,#(SP)
511 001262 010546      MOV   R5,#(SP)
512 001264 010637 001300      MOV   SP,LONSV
513 001270 012737 001302 000024      MOV   #HIGPHR,#24
514 001276 000000      HALT
515
516 001300 000000      LONSV1  0
517
518 001302 013706 001300      HIGPHR1 MOV   LONSV,SP
519 001306 012605      MOV   (SP)+,R5
520 001310 012604      MOV   (SP)+,R6
521 001312 012603      MOV   (SP)+,R3
522 001314 012602      MOV   (SP)+,R2
523 001316 012601      MOV   (SP)+,R1
524 001320 012600      MOV   (SP)+,R0
525 001322 012737 001250 000024      MOV   #LONPHR,#24
526 001330 012706 000500      MOV   #STKPTR,SP
527 001334 000240      NOP
528 001336 000240      NOP
529 001340 000240      NOP
530 001342 000000      HALT
531 001344 000240      NOP
532 001346 000240      NOP
533 001350 000240      NOP
534 001352 000137 001170      JMP   SCOPEO

```

```

536 001336 012706 000500          STARTI  MOV    #STKPTR,SP      ;GET UP THE STACK
537 001342 012777 000340 177420  MOV    #340,#PSW      ;RAISE PSW
538 001370 012700 001034          MOV    #DR0,R0       ;GET POINTER
539 001374 013701 001000          MOV    GSA00,R1      ;GET SUPPLIED ADDRESS
540 001400 010120          SYRAI  MOV    R1,(0)*      ;UPDATE
541 001402 062701 000002          ADD    #2,R1         ;ITHE
542 001406 022700 001064          CMP    #DR0*10,R0   ;ADDRESSES
543 001412 001372          BNE    SYRA          ;UNTIL DONE
544 001424 012700 001064          MOV    #DDONE,R0    ;GET POINTER
545 001420 013701 001002          MOV    GSVT,R1      ;GET SUPPLIED VECTOR
546 001424 010120          STRBI  MOV    R1,(0)*      ;UPDATE
547 001426 062701 000002          ADD    #2,R1         ;ITHE VECTORS
548 001432 022700 001100          CMP    #DDONE*14,R0 ;STRB
549 001436 001372          BNE    STRB         ;SWITCH
550 001440 005037 005556          CLR    SWITCH      ;HOUSEKEEP
551 001444 005037 001042          CLR    M0LQ        ;
552 001440 005004          CLR    R4          ;
553 001432 005037 001044          CLR    TSAVE       ;
554 001446 004737 001536          STRCI  JSR    PC,SETUP ;SET UP VECTORS
555 001442 005037 001042          CLR    M0LQ        ;
556 001446 012737 001000 012004  MOV    #1000,RAY14A ;HOUSEKEEP X,Y ORIGIN FOR LIGHTPEN
557 001474 012737 000600 012006  MOV    #600,RAY14B ;
558 001502 012737 030060 011764  MOV    #3000,DLT14A ;INITIALIZE X READOUT
559 001510 012737 030060 011766  MOV    #3000,DLT14A*2 ;
560 001516 012737 030060 011776  MOV    #3000,DLT14B ;INITIALIZE Y READOUT
561 001524 012737 030060 012000  MOV    #3000,DLT14B*2 ;
562 001532 000137 002050          JMP    FILE0       ;START THE TEST
563
564 001536 012737 000062 000060  SETUPI  MOV    #62,#000      ;RESET KRB VECTOR
565 001544 012737 000000 000062  MOV    #0,#62       ;
566 001532 042777 000100 177232  BIC    #100,#TKS    ;CLEAR INT ENABLE
567 001500 005037 002046          CLR    KRBQ        ;
568 001504 032737 000200 177570  BIT    #200,#DISPLAY ;TEST FOR "KRB" CONTROL
569 001572 001413          BEQ    SETUPA      ;BR IF NOT
570 001574 005137 002046          COM    KRBQ        ;SET "KRB" CONTROL
571 001600 012737 001700 000060  MOV    #REIB,#60    ;SET UP "KRB" INT
572 001606 012737 000340 000062  MOV    #340,#002    ;
573 001614 002777 000100 177170  BIS    #100,#TKS    ;ENABLE "KRB" INT
574 001632 012777 001664 177234  SETUPA  MOV    #SETUPB,DDONE ;SET UP GT DONE VECTOR
575 001630 012777 000340 177230  MOV    #340,#DDONE1 ;
576 001636 013777 001072 177224  MOV    LPVCT1,0LPVCT ;RESET LIGHT-PEN VECTOR
577 001644 005077 177222          CLR    0LPVCT1     ;
578 001600 013777 001076 177216  MOV    TIMEVT1,TIMEVT ;RESET TIME-OUT/SHIFT OUT VECTOR
579 001636 005077 177214          CLR    TIMEVT1     ;
580 001662 000207          RTS                ;
581
582 001664 005777 177166          SETUPB  TST    00SR  ;TEST FOR STOP
583 001670 100401          BMI    ,+4         ;
584 001672 000000          HALT              ;ERROR, INTERRUPT OCCURRED TO THE STOP
585
586 001674 000002          RTI              ;VECTOR BUT STOP WAS NOT SET
587 001676 000000          HALT

```

```

589
590
591 001700 117737 177110 001044  RETBI  MOVB   #TKR,TSAVE ;READ THE CHARACTER
592 001706 042737 177600 001044  BIC    #177000,TSAVE ;MASK TO 7 BITS
593 001714 022737 000015 001044  CMP    #15,TSAVE    ;TEST FOR "CR"
594 001722 001440          BEQ    KYT0        ;BR IF
595 001724 005037 005556          CLR    SWITCH     ;CLEAR "SWITCH"
596 001730 162737 000101 001044  SUB    #100,TSAVE   ;MAKE 0-77
597 001736 100426          BMI    KYT1        ;KA
598 001740 022737 000017 001044  CMP    #17,TSAVE   ;
599 001746 100412          BMI    KYT2        ;>
600 001750 013704 001044          MOV    TSAVE,R4    ;
601 001754 012737 177777 001030  MOV    #-1,CHANGE  ;
602 001762 005037 005556          CLR    SWITCH     ;
603 001766 005037 001042          CLR    M0LQ       ;
604 001772 000002          RTI                ;EXIT
605 001774 022737 000076 001044  KYT2I  CMP    #76,TSAVE   ;
606 002002 001015          BNE    KYT4        ;
607 002004 012737 177777 001042  MOV    #-1,M0LQ    ;RUBOUT
608 002012 000002          RTI                ;EXIT
609 002014 005037 001042          KYT1I  CLR    M0LQ     ;
610 002020 000002          RTI                ;
611 002022 000000          HALT              ;FATAL ERROR RTI FAILED
612
613 002024 012737 177777 005556  KYT3I  MOV    #-1,SWITCH  ;
614 002032 000002          RTI                ;
615 002034 000000          HALT              ;FATAL ERROR, RTI FAILED
616
617 002036 162737 000040 001044  KYT4I  SUB    #40,TSAVE   ;CONVERT LC TO UC
618 002044 000734          BR                ;
619 002046 000000          KRBDI  0          ;
620

```

```

        .LIST
685      IEXECUTE DIRECTORY FRAME
689
690
691      002090 104000      FILE01 SCOPE
692      002092 004537 005412      JSR      5,MSG      IEXIT TO DISPLAY A FRAME
693      002096 001000      JSR      1000
694      0020A0 003540      FRM00      IUSING THE DIR. FRAME
695
696      IEXECUTE DOT REPEATABILITY FRAME
697
698      0020A2 104000      FILE11 SCOPE
699      0020A4 004537 005412      JSR      5,MSG      IEXIT TO DISPLAY A FRAME
700      002070 100000      JSR      100000
701      002072 007140      FRM01      IUSING THE DOT REPEAT FRAME
702
703      IEXECUTE PINCUSHION FRAME
704
705      002074 104000      FILE21 SCOPE
706      002076 012700 012470      MOV      #BUFFER,R0      ILOAD START ADDRESS
707      002102 004737 002252      JSR      PC,SETPNT      ILOAD 0,0 ORGIN
708      002106 012701 000020      MOV      #20,R1      ISETUP COUNT
709      002112 012720 040000      MOV      #INTX,(R0)+      ILOAD INT LINE
710      002116 012720 001777      MOV      #MAXY,(R0)+      I MAX Y
711      002122 012720 000100      MOV      #100,(R0)+      ILOAD DELTA X
712      002126 012720 021777      MOV      #MINUS*MAXY,(R0)+ ILOAD * MAX Y
713      002132 005301      DEC      R1      IFINISHED ?
714      002134 001366      BNE      1$      IBR IF NOT
715      002136 012720 020001      MOV      #MINUS*1,(R0)+      IGO BACK 1 UNIT
716      002142 012720 000000      MOV      #0,(R0)+
717      002146 012720 040000      MOV      #INTX,(R0)+
718      002152 012720 001777      MOV      #MAXY,(R0)+      IPLOT LAST LINE
719      002156 004737 002252      JSR      PC,SETPNT      ISET ORGIN
720      002102 012701 000020      MOV      #MAXY+1,(R0)+      ISETUP COUNT
721      002106 012720 041777      MOV      #INTX*MAXX,(R0)+ ILOAD DELTA X MAX
722      002112 012720 000000      MOV      #0,(R0)+      ILOAD DELTA Y = 0
723      002116 012720 021777      MOV      #MINUS*MAXX,(R0)+ IRETRACE
724      002122 012720 000100      MOV      #100*(R0)+      ILOAD DELTA Y OF 100
725      002126 005301      DEC      R1
726      002128 001366      BNE      2$      IBR IF NOT
727      002132 012720 000000      MOV      #0,(R0)+
728      002136 012720 020001      MOV      #MINUS*1,(R0)+
729      002142 012720 041777      MOV      #INTX*MAXX,(R0)+      IPLOT LAST LINE
730      002146 012720 000000      MOV      #0,(R0)+
731      002152 012720 173400      MOV      #DSJOP,(R0)+      ILOAD STOP
732      002156 012720 100000      MOV      #DJRP,(R0)+      ILOAD JUMP
733      002162 012710 012470      MOV      #BUFFER,(R0)
734      002166 000137 002274      JMP      FILE2A
735
736      002292 012720 117000      SETPNT1 MOV      #POINT1,INT*(R0)+      ILOAD POINT
737      002296 012720 000000      MOV      #0,(R0)+      I AT X
738      002202 012720 000000      MOV      #0,(R0)+      I AT Y
739      002206 012720 110000      MOV      #LONGV,(R0)+      I LONG VECTOR
740      002272 000207      RTS      PC      IEXIT
    
```

```

742
743      002274 012737 004000 001046 FILE2A1 MOV      #4000,CNTR      ILOAD COUNTER
744      002302 007737 005556 FILE2B1 TST      SWITCH      ITEST SWITCH
745      002306 001405      BEQ      FILE2C      IBR IF SUBTEST NOT SELECTED
746      002310 004537 005412      JSR      5,MSG      IEXIT TO DISPLAY FRAME
747      002314 000001      I
748      002316 012470      BUFFER
749      002320 004004      BR      FILE2D      IUSING THE CROSS HATCH PATTERN
750
751      002322 004537 005412 FILE2C1 JSR      5,MSG      IEXIT TO DISPLAY FRAME
752      002326 000001      I
753      002330 007230      FRM02
754      002332 005337 001046 FILE2D1 DEC      CNTR      IUSING THE OFFSET PATTERN
755      002336 001361      BNE      FILE2B      IFINISHED ?
756
757      IEXECUTE OCTAGONS OR SQUARES
758
759      002340 104000      FILE31 SCOPE
760      002342 012737 014000 001046      MOV      #14000,CNTR      ISET UP A COUNTER
761      002350 005737 005556 FILE3A1 TST      SWITCH
762      002354 001010      BNE      FILE3B      IBRANCH IF SUB+TEST
763      002356 004537 005412      JSR      5,MSG      IDISPLAY TEST
764      002360 000001      I
765      002364 007334      FRM03      IFRAME # 3
766      002366 005337 001046      DEC      CNTR      IDECREMENT COUNTER
767      002372 001366      BNE      FILE3A      IBRANCH IF NOT COMPLETE
768      002374 000407      BR      FILE4      IEXIT TO NEXT TEST
769
770      002376 004537 005412 FILE3B1 JSR      5,MSG      IDISPLAY TEST
771      002402 000001      I
772      002404 007724      FRM0A      IFRAME # 3
773      002406 005337 001046      DEC      CNTR      IDECREMENT COUNTER
774      002412 001356      BNE      FILE3A      IBRANCH IF NOT COMPLETE
    
```

```

776
777
778      DISPLAY FILE
779      CHARACTER AND ITALICS TEST
780      SET UP THE BUFFER FOR THIS TEST
781      FILE4: SCOPE
782      MOV      #BUFFER,R0
783      MOV      #STAT$SIZE0,(R0)+
784      MOV      #STAT$ITAL0$SYNOFFGREEN,(R0)+
785      MOV      #P$INT$INT4$LOFF$BLKOFF$LINE0,(R0)+      ILCAD POINT MPDE
786      MOV      #0,(R0)+
787      MOV      #MAXY=77,(R0)+
788      MOV      #CHAR,(R0)+
789      MOV      #17,(R0)+
790      MOV      #17,(R0)+
791      MOV      #10$,STCHAR      ILOAD INITIAL CHAR,
792      JSR      PC,LOADBF
793      MOV      #14$,STCHAR      ILOAD INITIAL LG CHAR
794      JSR      PC,LOADBF      ILOAD LINE
795      MOV      #4$,STCHAR      ILOAD NUMBERS AND PUNCT
796      JSR      PC,LOADBF      ILOAD LINE
797      MOV      #STAT$ITAL0,(R0)+      ILOAD NORMAL FONT
798      JSR      PC,LOADSP      ILOAD SPECIAL CHARS
799      JSR      PC,$SPACE      IINSERT SPACES
800      MOV      #STAT$ITAL1,(R0)+      ILOAD ITALICS FONT
801      JSR      PC,LOADSP      ILOAD SPECIAL
802      MOV      #D$TOP,(R0)+      ILOAD D$TOP
803      MOV      #D$FP,(R0)+
804      MOV      #BUFFER,(R0)+
805      JMP      FILE4A
806
807      MOV      #1,(R0)+
808      MOV      #0,R2
809      MOV      #37,R3      ISET INITIAL SHIFT OUT CHAR
810      MOV      R2,(R0)+      ILOAD COUNT
811      INC      R2      ILOAD CHAR
812      CMP      #17,R2
813      BEQ      2$      ITEST FOR $I
814      DEC      R3      I$R IF $I "17"
815      BNE      1$      I$FINISHED ?
816      MOV      #20$,R3      I$R IF NOT
817      RTS      PC      ILOAD SHIFT-IN SPACE
818
819      MOV      #STAT$ITAL0,(R0)+      ILOAD NORMAL FONT
820      MOV      #STCHAR,R2      ISET STARTING CHAR
821      JSR      PC,FILLIT      ILOAD THE CHARACTERS
822      JSR      PC,$SPACE      IINSERT SPACES
823      MOV      #STAT$ITAL1,(R0)+      ILOAD ITALICS FONT
824      MOV      #STCHAR,R2      ISET STARTING CHARACTER
825      JSR      PC,FILLIT      ILOAD THE CHARACTERS
826      JSR      PC,$RLF      IINSERT CR/LF
827      RTS      PC      IEXIT
828
829      STCHAR: 0
  
```

```

830
831      CR/LF: MOV      #15,(R0)+
832      MOV      #12,(R0)+
833      MOV      #12,(R0)+
834      MOV      #12,(R0)+
835      RTS      PC      IEXIT
836
837      FILLIT: MOV      #4$,R3
838      FILLAI: MOV      R2,(R0)+
839      INC      R2
840      DEC      R3
841      BNE      FILLAI
842      RTS      7
843
844      SPACE: MOV      #1$,R3
845      I$T: MOV      #4$,R3      ILOAD A SPACE
846      DEC      R3
847      BNE      1$      I$R IF NOT DONE
848      RTS      PC      IEXIT
849
850      IACTUAL DISPLAY ROUTINE
851
852      FILE4A: MOV      #100,10$      ILOAD A COUNTER
853      MOV      #MAXY=77,BUFFER+10      ILOAD STARTING POINT
854      JSR      R5,MSG
855      I
856      BUFFER
857
858      MOV      #40$,BUFFER+10
859      JSR      R5,MSG
860      I
861      BUFFER
862
863      DEC      10$      I$FINISHED ?
864      BNE      4$      I$R IF NOT
865      JMP      FILE5      IGO TO NEXT TEST
866
867      10$: 0
868
869      IEXECUTE DASH LINES AND BLINK
870
871      FILE5: SCOPE
872      JSR      5,MSG      IEXIT TO DISPLAY A FRAME
873      I0000
874      FRM5      IUSING THE DASH AND BLINK FRAME
  
```

```

076
077
078
079 003036 104000 FILE6: SCOPE
080 003040 012737 041777 010472 MOV #INX,DELTX6 ISETUP VERTICAL HEIGHT
081 003046 012737 000010 001036 MOV #10,DSAVE2 ISETUP T1,ER
082 003054 012737 000000 001034 MOV #0,DSAVE1
083 003062 012737 000040 001046 LOOPA1 MOV #40,CNTR ISETUP EXECUTION COUNT
084 003070 012737 000200 001032 LOOPA1 MOV #MAXY*1/10,DSAVE ISETUP
085 003076 013737 001034 010474 MOV DSAVE1,DELTY6
086 003104 004537 005412 JSR 5,MESG IEXIT TO DISPLAY FRAME
087 003110 000001 1
088 003112 010426 FRME6 IVECTOR LENGTH FRAME
089 003114 004537 005412 LOOPA2 JSR 5,MESG IEXIT TO DISPLAY FRAME
090 003120 000001 1
091 003122 010462 FRME6A IVECTOR LENGTH FRAME
092 003124 062737 000010 010474 ADD #10,DELTY6 IUPDATE ANGLE
093 003132 005337 001032 DEC DSAVE IFINISHED ALL THE ANGLES
094 003136 001366 BNE LOOPA2 IBR IF NOT
095 003140 005337 001046 LOOPA3 DEC CNTR IDONE COUNT?
096 003144 001351 BNE LOOPA1 IBR IF NOT
097 003146 000240 NOP
098 003150 005737 005556 TST SWITCH ITEST SWITCH
099 003154 001342 BNE LOOPA IBR IF HALT MOTION
100 003156 005237 001034 INC DSAVE1 IUPDATE INITIAL ANGLE
101 003162 005337 001036 DEC DSAVE2 IFINISHED ALL?
102 003166 001335 BNE LOOPA IBR IF NOT
103
104
105
106 003170 104000 FILE7: SCOPE
107 003172 012737 040000 001034 MOV #INX,DSAVE1 ISETUP INITIAL X
108 003200 012737 001777 010474 MOV #MAXY,DELTY6 ISETUP INITIAL Y
109 003206 012737 000010 001036 MOV #10,DSAVE2 ISETUP EXECUTION COUNT
110 003214 012737 000040 001046 LOOPB1 MOV #40,CNTR ISETUP DELAY
111 003222 012737 000200 001032 LOOPB1 MOV #200,DSAVE
112 003230 013737 001034 010472 MOV DSAVE1,DELTX6 IEXIT TO DISPLAY FRAME
113 003236 004537 005412 JSR 5,MESG IVECTOR LENGTH TEST FRAME
114 003242 000001 1
115 003244 010426 FRME6 IEXIT TO DISPLAY FRAME
116 003246 004537 005412 LOOPB2 JSR 5,MESG IEXIT TO DISPLAY FRAME
117 003252 000001 1
118 003254 010462 FRME6A IVECTOR LENGTH FRAME
119 003256 062737 000010 010472 ADD #10,DELTX6 IUPDATE ANGLE
120 003264 005337 001032 DEC DSAVE IFINISHED ALL THE ANGLES
121 003270 001366 BNE LOOPB2 IBR IF NOT
122 003272 005337 001046 LOOPB3 DEC CNTR IDONE COUNT?
123 003276 001351 BNE LOOPB1 IBR IF NOT
124 003300 000240 NOP
125 003302 005737 005556 TST SWITCH ITEST SWITCH
126 003306 001342 BNE LOOPB IBR IF HALT MOTION
127 003310 005237 001034 INC DSAVE1 IUPDATE INITIAL ANGLE
128 003314 005337 001036 DEC DSAVE2 IFINISHED ALL?
129 003320 001335 BNE LOOPB IBR IF NOT

```

```

931
932
933
934 003322 104000 FILE10: SCOPE
935 003324 005037 010506 CLR DELTX7
936 003330 004537 005412 D7A1 JSR 5,MESG IEXIT TO DISPLAY A FRAME
937 003334 000050 50
938 003336 010504 FRME10 IUSING THE HORIE FRAME
939 003340 004537 005412 JSR 5,MESG IEXIT TO DISPLAY A FRAME
940 003344 000001 1
941 003346 010604 FRM10 IUSING THE PERIMETER BOX
942 003350 000240 NOP
943 003352 005737 005556 TST SWITCH ITEST THE "SWITCH"
944 003356 001364 BNE D7A IBR IF FREEZE THE MOVEMENT
945 003360 062737 000001 010506 D7C1 ADD #1,DELTX7 IUPDATE THE X ORIGIN
946 003366 022737 002000 010506 CMP #200,DELTX7 ITEST IF THE END
947 003374 001355 BNE D7A IBR IF NOT
948
949
950
951 003376 104000 FILE11: SCOPE
952 003400 005037 010550 CLR DELTY7
953 003404 004537 005412 D7D1 JSR 5,MESG IEXIT TO DISPLAY A FRAME
954 003410 000050 50
955 003412 010544 FRME11 IUSING THE VERTY FRAME
956 003414 004537 005412 JSR 5,MESG IEXIT TO DISPLAY A FRAME
957 003420 000001 1
958 003422 010604 FRM10 IUSING THE PERIMETER BOX
959 003424 000240 NOP
960 003426 005737 005556 TST SWITCH ITEST THE "SWITCH"
961 003432 001364 BNE D7D IBR IF FREEZE THE MOVEMENT
962 003434 062737 000001 010550 D7F1 ADD #1,DELTY7 IUPDATE THE Y ORIGIN
963 003442 022737 002000 010550 CMP #MAXY*1,DELTY7 ITEST IF THE END
964 003448 001355 BNE D7D IBR IF NOT
965

```

```

967
968
969
970 003492 104000 FILE12) SCOPE
971 003494 012777 003550 175406 MOV #RETL,0LPCVCT ;SET UP LIGHT-PEN VECTOR
972 003402 013777 001004 175402 MOV GSRL,0LPVCT1 ;SET UP BR LEVEL
973 003470 012737 004000 001032 MOV #4000,DSAVE ;SET UP A EXECUTION COUNT
974 003476 003737 003556 FILE12A) TST SWITCH ;TEST THE "SWITCH"
975 003502 001004 BNE FILE12B ;BR IF SET "SYNC"
976 003504 042737 000004 010650 BIC #4,SYN12 ;ENSURE CLEAR "SYNC"
977 003532 004003 BR FILE12C ;BY PASS
978 003534 052737 000004 010650 FILE12B) BIS #4,SYN12 ;SET THE "SYNC"
979 003532 004937 003412 FILE12C) JSR 5,MESG ;EXIT TO DISPLAY FRAME
980 003526 000001 1 ;
981 003530 010442 FRME12 ;USING THE "INTENSITY" FRAME
982 003532 003337 001032 DEC ;FINISHED?
983 003536 001403 BEQ FILE12D ;YES, EXIT
984 003540 012737 173400 011250 MOV #DSTOP,RAYLPA ;NO, RESET MESSAGE
985 003546 000733 BR FILE12A ;BR BACK
986 003530 012737 164000 011250 RETLPI MOV #DASP,RAYLPA ;LIGHT-PEN HIT
987 003536 017737 173300 011262 MOV #YPOS,LPPNT ;READ Y POSITION
988 003544 042737 176000 011262 BIC #17000,LPPNT ;MASK THE BITS
989 003572 022626 CMP (SP),*(SP)+ ;POP THE STACK
990 003574 012777 000001 175252 MOV #1,OPPC ;SINGLE STEP THE DISPLAY
991 003602 000137 003430 JMP ;JUMP TO WAIT
992 003606 013777 001072 175254 FILE12D) MOV LPVCT1,0LPVCT ;RESET THE LIGHT-PEN VECTOR
993
994
995
996
997 003614 104000 FILE13) SCOPE
998 003616 004537 003412 JSR 5,MESG ;EXIT TO DISPLAY FRAME
999 003622 010000 10000 ;
1000 003624 011312 FRME13 ;USING THE "EDGE" FRAME

```

```

1002
1003
1004
1005 003626 104000 FILE14) SCOPE
1006 003630 012700 012470 MOV #BUFFER,0 ;SET UP 0
1007 003634 012720 114000 MOV #PCINT,(0)+ ;SET UP INITIAL
1008 003640 012720 000240 MOV #200,(0)+ ;X POSITION
1009 003644 012720 001000 MOV #MAXY+1/2,(0)+ ;Y POSITION
1010 003630 012720 107004 MOV #SHORTVINT4*LINE,(0)+ ;LOAD "SHORT VECTOR"
1011 003634 004737 003706 PC,LOADVT ;LOAD THE DISPLAY PATTERN
1012 003660 012720 130000 MOV #RELATV,(0)+ ;LOAD "RELATIVE POINT"
1013 003664 004737 003706 PC,LOADVT ;LOAD THE DISPLAY PATTERN
1014 003670 012720 173000 MOV #DSTOP,(0)+ ;LOAD "DISPLAY STOP"
1015 003674 012720 160000 MOV #DJMP,(0)+ ;LOAD "DISPLAY JUMP"
1016 003700 012720 012470 MOV #BUFFER,(0)+ ;TO THE BUFFER ADDRESS
1017 003704 000413 BR FILE14A ;BR TO THE FRAME
1018
1019 003736 012737 000024 001046 LOADVT) MOV #24,CNTR ;LOAD A COUNTER
1020 003714 012720 040077 LADVT) MOV #INTX+77,(0)+ ;LOAD A DELTA Y
1021 003720 012720 004177 MOV #4177,(0)+ ;LOAD A DELTA X,Y
1022 003724 003337 001046 DEC CNTR ;FINISHED?
1023 003730 001371 BNE LADVT ;BR IF NOT
1024 003732 000207 RTS PC ;EXIT
1025
1026 003734 012737 004000 004104 FILE14A) MOV #4000,105 ;LOAD COUNTER
1027 003742 012737 000200 011572 ISI MOV #200,FRM14A ;LOAD FIRST OCTAGON
1028 003750 012737 000200 011574 MOV #200,FRM14B ;
1029 003756 004537 005412 JSR R5,MESG ;DISPLAY OCT.
1030 003762 000001 1 ;
1031 003764 011566 FRME14 ;
1032 003766 012737 001400 011572 MOV #1400,FRM14A ;LOAD SECOND OCTAGON
1033 003774 012737 000200 011574 MOV #200,FRM14B ;
1034 004002 004537 005412 JSR R5,MESG ;DISPLAY 2ND OCT.
1035 004006 000001 1 ;
1036 004010 011566 FRME14 ;
1037 004012 012737 001400 011572 MOV #1400,FRM14A ;LOAD THIRD OCTAGON
1038 004020 012737 001400 011574 MOV #MAXY-377,FRM14B ;
1039 004026 004537 005412 JSR R5,MESG ;
1040 004032 000001 1 ;
1041 004034 011566 FRME14 ;
1042 004036 012737 000200 011572 MOV #200,FRM14A ;LOAD FOURTH OCTAGON
1043 004044 012737 001400 011574 MOV #MAXY-377,FRM14B ;
1044 004052 004537 005412 JSR R5,MESG ;DISPLAY 4TH OCT.
1045 004056 000001 1 ;
1046 004060 011566 FRME14 ;
1047 004062 004537 005412 JSR R5,MESG ;DISPLAY BAR
1048 004066 000001 1 ;
1049 004070 012470 BUFFER ;
1050 004072 003337 004104 DEC 105 ;FINISHED?
1051 004076 001321 BNE 15 ;BR IF NOT
1052 004100 000137 004106 JMP FILE15 ;NEXT TEST
1053 004104 000000 105) 0 ;
1054

```

```

1056
1057
1058
1059 004156 104000 IGRAPH PLOT X+Y TEST
1060 004110 012700 012470 FILE151 SCOPE
1061 004114 012720 117600 MOV #BUFFER,R0 ILOAD R0
1062 004120 012720 000000 MOV #POINT:INT7,(0)* ILOAD INITIAL POINT
1063 004124 012720 000000 MOV #0,(0)*
1064 004130 012720 170052 MOV #STATSA[ITAL0]SYNOFFIGREEN,(R0)* IRESET THE STATUS A
1065 004134 012720 174100 MOV #STATSB[INCR,(0)* ILOAD INITIAL STATUS B
1066 004140 012720 120000 MOV #GRAPHX,(0)* ILOAD GRAPH X INST
1067 004144 012705 000040 DFL1501 MOV #40,R5 ILOAD STARTUP COUNT
1068 004150 012737 000000 001032 MOV #0,DSAVE ILOAD INITIAL PLOT
1069 004156 000403 BR 25
1070 004100 062737 000020 001032 1ST ADD #20,DSAVE IUPDATE PLOT POINT
1071 004166 043720 001032 2ST MOV DSAVE,(0)* ISAVE THE POINT
1072 004172 005305 DEC R5 IFINISHED?
1073 004174 001371 BNE 15 IBR IF NOT
1074 004176 012720 173400 MOV #DSTOP,(0)* ILOAD "DSTOP"
1075 004202 012720 160000 MOV #DJMP,(0)* ILOAD "DJMP"
1076 004206 012720 012470 MOV #BUFFER,(0)* ILOAD RETURN
1077 004212 012737 000200 001032 MOV #200,DSAVE ILOAD POINT COUNT
1078 004230 042777 004000 174602 DFL1501 BIC #4000,#DBUF5 IENSURE "GRAPHX"
1079 004226 005737 005556 TST SWITCH ITEST SWITCH
1080 004232 001403 BEQ DFL1501 IBR IF GRAPHX
1081 004234 052777 004000 174566 BIS #4000,#DBUF5 ISET GRAPHY
1082 004242 004537 005412 DFL1501 JSR 5,MSG IEXIT TO DISPLAY A FRAME
1083 004246 000001 I
1084 004250 012470 I BUFFER IUSING THE GENERATED PATTERN
1085 004292 062777 000001 174546 ADD #1,#DBUF4 IUPDATE INCREMENT
1086 004260 022777 174200 174540 CMP #STATSB,200,#DBUF4 ITEST IF LAST INCREMENT
1087 004266 001365 BNE DFL1501 IBR IF NOT
1088 004270 012777 174100 174530 MOV #STATSB[INCR,#DBUF4 IRELOAD INCREMENT
1089 004276 005337 001032 DEC DSAVE IFINISHED 10 SEC?
1090 004302 001346 BNE 15 IBR IF NOT
1091
1092 004304 013700 000042 MOV #42,R0
1093 004310 001407 BEQ HERE IACT=11/DDP=11
1094 004312 000005 RESET
1095 004314 000005 RESET
1096 004316 004710 LOGICAL1 JSR PC,(R0)
1097 004320 000240 NOP
1098 004322 000240 NOP
1099 004324 000240 NOP
1100 004326 000240 NOP
1101 004330 000137 002050 HERE1 JMP FILE0
1102 004334 000240 NOP
1103 004336 000240 NOP
1104 004340 000240 NOP
  
```

```

1106
1107
1108
1109 004342 104000 IOPERATOR OPERATOR INTERVENTION TESTS
1110 004344 012777 004614 174516 FILE161 SCOPE
1111 004352 013777 001004 174512 MOV #REI14,0LPVCT
1112 004360 012737 000100 001034 MOV #000,0LPVCT1
1113 004366 012700 012470 MOV #100,DSAVE1 ISET UP COUNT
1114 004372 012737 000100 001032 1ST MOV #BUFFER,R0 ILOAD START ADDR,
1115 004400 012720 117144 MOV #100,DSAVE
1116 004404 012720 000700 MOV #POINT:INT4[LPONILINED,(R0)* ILOAD POINT
1117 004410 012720 000474 MOV #700,(R0)* ILOAD X POINT
1118 004414 004737 004556 MOV #470,(R0)* ILOAD Y POINT
1119 004420 012720 173400 JSR PC,LOADUP ILOAD UP THE BUFFER
1120 004424 012720 160000 MOV #DSTOP,(R0)* ILOAD DSTOP
1121 004430 012720 012470 MOV #DJMP,(R0)* ILOAD DJUMP
1122 004434 005037 005050 CLR #BUFFER,(R0)* ILOAD RETURN ADDRESS
1123 004440 012737 030000 012374 HITENT ICLEAR HIT COUNT
1124 004446 012737 030000 012372 MOV #30000,FRM10B=2 IPRESET THE READOUT
1125
1126 004494 005737 005556 4ST TST SWITCH ITEST SWITCH BIT
1127 004460 001005 BNE 65 IBR IF SUBTEST
1128
1129 004462 004537 005412 JSR R5,MSG IEXIT TO DISPLAY FRAME
1130 004466 000100 I00
1131 004470 011714 FRM16. IJUNT THE LIGHT=PEN FRAME
1132 004472 000770 BR 45 IBR BACK
1133
1134 004474 004537 005412 6ST JSR R5,MSG IEXIT TO DISPLAY FRAME
1135 004500 000001 I1
1136 004502 012302 FRM16A IASCII SUBTITLE
1137
1138 004504 004537 005412 JSR R5,MSG IEXIT TO DISPLAY FRAME
1139 004510 000001 I1
1140 004512 012470 BUFFER
1141
1142
1143 004514 005337 001032 DEC DSAVE IFINISHED ?
1144 004520 001355 BNE 45 IBR IF NOT MINI-LOOP
1145
1146 004522 005337 001034 DEC DSAVE1 IFINISHED ?
1147 004526 001317 BNE 15 IBR IF NOT
1148 004530 000137 004342 JMP FILE16 IRESTART
1149
  
```

```

1151
1152 004534 012701 000030 LOADACI MOV #24,R1 ;LOAD COUNT
1153 004540 012720 130000 MOV #RELATV,(R0)+ ;LOAD RELATIVE POINT
1154 004544 012720 040004 1ST MOV #INIX*4,(R0)+ ;LOAD INTEN BIT
1155 004550 005301 DEC R1 ;FINISHED ?
1156 004552 001374 BNE 1S ;BR IF NOT
1157 004554 000207 RTS PC ;EXIT
1158
1159 004556 012737 000030 001046 LOADUPI MOV #24,CNTR ;LOAD COUNT
1160 004564 004737 004534 1ST JSR PC,LOADAC ;LOAD ACROSS
1161 004570 012720 110000 MOV #LOADV,(R0)+ ;LOAD LONG VECTOR
1162 004574 012720 000004 MOV #4,(R0)+ ;LOAD VECTOR OVER
1163 004600 012720 020140 MOV #MINUS*140,(R0)+ ;AND UP
1164 004604 005337 001046 DEC CNTR
1165 004610 001365 BNE 1S ;BR IF NOT DONE
1166 004612 000207 RTS PC ;EXIT
1167
1168
1169 004614 017737 174242 004754 RET14; MOV #YPOS,40S
1170 004622 042737 176000 004754 BIC #170000,40S
1171 004630 017737 174224 004756 MOV #XPOS,41S
1172 004636 042737 176000 004756 BIC #170000,41S
1173 004644 005737 005556 TST SWITCH ;TEST SW
1174 004650 001411 BEQ 1S ;BR IF LIGHT PEN FOLLOW
1175 004692 005237 INC HITCNT ;UPDATE LIGHT PEN HIT COUNT
1176 004696 013703 005050 MOV HITCNT,R3 ;LOAD R3
1177 004662 012702 012376 MOV #FRM160,R2 ;LOAD ADDRESS
1178 004666 004737 005332 JSR PC,R0CHR ;CONVERT OCTAL
1179 004672 000432 BR 20S ;BR
1180 004674 013703 004756 1ST MOV 41S,R3 ;LOAD R3
1181 004700 012702 011770 MOV #DLI144*4,R2 ;LOAD ADDRESS
1182 004704 004737 005332 JSR PC,R0CHR ;LOAD X READOUT
1183 004710 013703 004754 MOV 40S,R3 ;LOAD R3
1184 004714 012702 012002 MOV #DLI140*4,R2 ;LOAD ADDRESS
1185 004720 004737 005332 JSR PC,R0CHR ;LOAD Y READOUT
1186 004724 013737 004754 012006 MOV #00,RAT14B ;LOAD NEW Y POSITION
1187 004732 013737 004756 012004 MOV 41S,RAT14A ;LOAD NEW X POSITION
1188 004740 10S;
(1) 004740 012777 000001 174106 MOV #1,00PC ;SINGLE STEP THE DISPLAY
1189 004746 022626 CHP (SP)+,(SP)+
1190 004750 000137 005430 JMP MESSAGE
1191
1192 004754 000000 40S; 0
1193 004756 000000 41S; 0
1194

```

```

1196
1197 004760 005001 20S; CLR R1
1198 004762 005002 CLR R2
1199 004764 013700 004756 MOV 41S,R0 ;GET X AXIS
1200 004770 162700 000700 SUB #700,R0 ;GET A BASE ADDRESS
1201 004774 000200 ASR R0
1202 004776 000200 ASR R0
1203 005000 001404 BEQ 30S
1204 005002 062701 000070 21S; ADD #700,R1 ;UPDATE OFFSET
1205 005006 005300 DEC R0
1206 005010 001374 BNE 21S ;BR UNTIL DONE
1207
1208 005012 013700 004754 30S; MOV 40S,R0 ;GET X AXIS
1209 005016 162700 000500 SUB #500,R0 ;MAKE BASE ADDRESS
1210 005022 000200 ASR R0
1211 005024 000200 ASR R0 ;SHIFT RIGHT
1212 005026 001404 BEQ 32S
1213 005030 062701 000002 31S; ADD #2,R1
1214 005034 005300 DEC R0
1215 005036 001374 BNE 31S
1216 005040 042761 040000 012500 32S; BIC #INIX,BUFFER*10(R1) ;CLEAR THE BIT
1217 005046 000734 BR 10S
1218
1219 005050 000000 HITCNT; 0

```

```

1221                                     ECHO ROUTINE KEYBOARD TO DISPLAY
1222
1223 005052 104000                                     FILE17 SCOPE
1224 005054 012700 012470                                     MOV #BUFFER,R0 ILOAD R0
1225 005060 012720 173400 ECHOA: MOV #DSTOP,(0)+ IMOV "DSTOPS"
1226 005064 022700 013470 CMP #BUFFER+100,R0 ITHRUOUT THE
1227 005070 001373 BNE ECHQA I;BUFFER
1228 005072 005037 001052 CLR LOKRB IHOUSE
1229 005076 005037 001032 CLR DSAVE IKEEPING
1230 005102 112737 000060 012461 MOVB #60;KBDOCT-4
1231 005110 112737 000060 012462 MOVB #60;KBDOCT-3 IPRESET READOUT
1232 005116 112737 000060 012463 MOVB #60;KBDOCT-2
1233 005124 112737 000060 012464 MOVB #60;KBDOCT-1
1234 005132 012737 005206 000060 MOV #REI117;R#60 ILOAD KEYBOARD VECTOR
1235 005140 012737 000340 000062 MOV #340;R#02
1236 005146 052777 000100 173636 BIS #100;R0K IENABLE INTERRUPT
1237 005154 012737 000700 005330 MOV #700;CHRCNT ILOAD CHAR COUNT
1238 005162 012700 012470 MOV #BUFFER,R0 IRESET R0
1239 005166 004537 005412 ECHO: JSR 0;MSG IEXIT TO DISPLAY A FRAME
1240 005172 000001 1 FRME17
1241 005174 012404 TST LOKRB IUSING THE KEYBOARD HEADER
1242 005176 005737 001052 BNE RET#1 IUPDATE A CHAR?
1243 005202 001012 RET#1 I;R IF YES
1244 005204 000770 ECHQC
1245 005206 017701 173602 RET117: MOV #TK#R1 IGET A CHAR
1246 005212 042701 177600 BIC #17700;R1 IMASK
1247 005216 012737 177777 001052 MOV #1;LOKRB ISET (FLAG)
1248 005224 000002 RTI IEXIT
1249 005226 000000 HALT
1250 005230 005037 001052 RET21: CLR LOKRB ICLEAR (FLAG)
1251 005234 022701 000003 CMP #3,R1 ITEST FOR PC
1252 005240 001002 BNE RET#0 I;R IF NOT
1253 005242 000137 001456 JMP STRC IRESTART
1254 005246 005337 005330 RET#0: DEC CHRCNT IFINISHED COUNT?
1255 005252 001002 BNE 1$ I;R IF NOT
1256 005254 000137 005054 JMP FILE17+2 IRESTART
1257 005260 012702 012465 1$ MOV #KBDOCT,R2 ILOAD ADDRESS
1258 005264 010103 MOV R1,R3
1259 005266 004737 005332 JSR PC,RBCHR ILOAD THE OCTAL VALUE
1260 005272 005737 001032 TST DSAVE ITEST HIGH/LOW BYTE
1261 005276 001007 BNE ECHQB
1262 005300 110120 MOVB R1,(R0)+ ISAVE BYTE
1263 005302 112710 000017 MOVB #17;(R0) ISHIFT-IN
1264 005306 005137 001032 COM DSAVE ICMP FLAG
1265 005312 000137 005166 JWP ECHQC I;R BACK
1266 005316 110120 ECHOB: MOVB R1,(R0)+ ISAVE CHAR
1267 005320 005037 001032 CLR DSAVE ICLEAR FLAG
1268 005324 005137 005166 JWP ECHQC I;R BACK
1269
1270
1271 005330 000200 CHRCNT: 200

```

```

1273                                     IUPDATE OCTAL READOUT
1274
1275 005332 042703 176000 KBCHR: BIC #17700;R3
1276 005336 004737 005376 JSR PC,R0 ILOAD BITS
1277 005342 110442 005370 MOVB R4,(R2) ISAVE BITS
1278 005344 004737 005370 JSR PC,R3 IMOVE BITS
1279 005350 110442 005370 MOVB R4,(R2) ISAVE BITS
1280 005352 004737 005370 JSR PC,R3 IMOVE BITS
1281 005356 110442 005370 MOVB R4,(R2) ISAVE BITS
1282 005360 004737 005370 JSR PC,R3
1283 005364 110442 005370 MOVB R4,(R2)
1284 005366 000207 RTS PC
1285 005370 000003 11$: ROR R3
1286 005372 000003 ROR R3
1287 005374 000003 ROR R3
1288 005376 010304 10$: MOV R3,R4 ILOAD R4
1289 005400 042704 177770 BIC #17770;R4 IMASK BITS
1290 005404 062704 000060 ADD #60;R4 IMAKE A NUMBER
1291 005410 000207 RTS PC
1292
1293 005412 012537 005552 MSG1: MOV (5);COUNT
1294 005416 010537 005554 MOV (5);FILE
1295 005422 013777 005554 173424 MOV FILE;DOPC ISTART DISPLAY
1296 005430 005077 173354 MSGA: CLR #PSH
1297 005434 000001 WAIT
1298 005436 005737 002046 TST KRDB
1299 005442 001025 BNE MSGAB
1300 005444 005337 005552 MSGAA: DEC COUNT
1301 005450 001405 BEQ MSGB
1302 005452 012777 000001 173374 MOV #1;DOPC ISINGLE STEP THE DISPLAY
1303 005460 000137 005430 JMP MSGA
1304 005464 000240 MSGB: NOP
1305 005466 005737 002046 TST KRDB
1306 005472 001010 BNE MSGBA
1307 005474 005037 005556 CLR SWITCH
1308 005500 032737 000100 177570 BIT #011;#DISPLAY
1309 005506 001402 BEQ MSGBA
1310 005510 005137 005556 COM SWITCH
1311 005514 000205 MSGBA: RTS 5
1312 005516 005737 005556 MSGAB: TST SWITCH
1313 005522 001350 BNE MSGAA
1314 005524 005737 001050 TST CHANGE
1315 005530 001745 BEQ MSGAA
1316 005532 005037 001050 CLR CHANGE
1317 005536 005037 005556 CLR SWITCH
1318 005542 005037 001042 CLR HOLD
1319 005546 000137 001162 JMP SCOPEC
1320 005552 000000 COUNT: 0
1321 005554 000000 FILE: 0
1322 005556 000000 SWITCH: 0

```

```

1324
1325 005560 114000          FRME01 POINT
1326 005562 000000          0
1327 005564 001500          MAXY=277
1328 005566 170052          STATAIITALEISYNOFFIGREEN
1329 005570 103124          CHARINT4LPOFFIBLKOFFILINE0
1330 005572 017 017          .BYTE 17,17
1331 005574 052107 032055 020060          .ASCII /GT=40 OR GT=44 WITH VR17 VISUAL TEST (MD=11-DDGTG-A)/
005576 051117 043440 026524
005578 032064 053440 052111
005580 020110 051126 033461
005582 053040 051511 040525
005584 020114 042524 052123
005586 020040 046474 026504
005588 030461 042055 043504
005590 043524 040455 000076
1332 005662 015 012 012          .BYTE 15,12,12
1333 005665 040 020040 044504          .ASCII / DIRECTORY/
005667 042522 052103 051117
005669 131
005671 015 012 012          .BYTE 15,12,12
1334 005701 030060 036440 040440          .ASCII /00 = A = DIRECTORY/
1335 005712 036440 042040 051111
005714 041505 047524 054522
005716 015 012          .BYTE 15,12
1336 005726 030460 036440 041040          .ASCII /01 = B = DOT REPEATIBILITY/
1337 005736 036440 042040 052117
005738 051040 050105 040505
005740 044524 044502 044514
005742 054524
005744 015 012          .BYTE 15,12
1338 005762 031060 036440 041440          .ASCII /02 = C = PINCUSHION AND VECTOR CURVATURE (X OR Y OFFSET ADJ.)/
1339 005772 036440 050040 047111
005774 052503 041123 047511
005776 020116 047101 020104
005778 042526 052103 051117
005780 041440 051125 040526
005782 052524 042522 036040
005784 020130 051117 054440
005786 047440 043106 042523
005788 020124 042101 027112
005790 076
1340 006061 015 012          .BYTE 15,12
1341 006063 060 020063 020075          .ASCII /03 = D = OCTAGONS OR SQUARES/
006065 020104 020075 041517
006067 040524 047507 051516
006069 047440 020122 050523
006071 040525 042522 123
1342 006127 015 012          .BYTE 15,12
1343 006121 060 020064 020075          .ASCII /04 = E = CHARACTER SET (CHAR, ADJ.)/
006123 020105 020075 041103
006125 051101 041501 042524
006127 020122 042523 020124
006129 041474 040510 027122

```

```

1344 006136 040440 045104 037056          .BYTE 15,12
1345 006134 015 012          .ASCII /05 = F = DASH LINES AND BLINK/
006136 032460 036440 043040
006138 036440 042040 051501
006140 020110 044514 042516
006142 020123 047101 020104
006144 046102 047111 113
1346 006223 015 012          .BYTE 15,12
1347 006225 060 020066 020075          .ASCII /06 = G = HORIZONTAL VECTOR ANGLE (ADJ, X VECTOR LENGTH)/
006227 052107 020075 047510
006229 044522 047532 052116
006231 046101 053040 041505
006233 047524 020122 047101
006235 046107 020105 040474
006237 045104 020056 020130
006239 042526 052103 051117
006241 046040 047105 052107
006243 037110
1348 006314 015 012          .BYTE 15,12
1349 006316 033460 036440 044040          .ASCII /07 = H = VERTICAL VECTOR ANGLE (ADJ, Y VECTOR LENGTH)/
006318 036440 053040 051105
006320 044524 040503 020114
006322 042526 052103 051117
006324 040440 043516 042514
006326 036040 042101 027112
006328 054440 053040 041505
006330 047524 020122 042514
006332 043516 044124 076
1350 006403 015 012          .BYTE 15,12
1351 006405 061 020060 020075          .ASCII /10 = I = HORIZONTAL PHOSPHOR TEST/
006407 020111 020075 047510
006409 044522 047532 052116
006411 046101 050040 047510
006413 050123 047510 020122
006415 042524 052123
1352 006446 015 012          .BYTE 15,12
1353 006448 030461 036440 045040          .ASCII /11 = J = VERTICAL PHOSPHOR TEST/
006450 036440 053040 051105
006452 044524 040503 020114
006454 044120 051517 044120
006456 051117 052040 051505
006458 124
1354 006507 015 012          .BYTE 15,12
1355 006511 061 020062 020075          .ASCII /12 = K = INTENSITY LEVEL AND LIGHT-PEN TEST/
006513 050113 020075 047111
006515 042524 051516 052111
006517 020131 042514 042526
006519 020114 047101 020104
006521 044514 044107 026524
006523 042520 020116 042524
006525 052123
1356 006504 015 012          .BYTE 15,12
1357 006506 031461 036440 046040          .ASCII /13 = L = EDGE FLAG TEST/
006508 036440 042440 043504

```

1358	020105	046106	043501		
1359	052040	051505	124		
	015	012		.BYTE	15,12
	061	020064	020075	.ASCII	/14 = M = SHORT VECTORS AND RELATIVE POINT,
	020115	020075	044123		
	051117	020124	042526		
	052103	051117	020123		
	047101	020104	042522		
	040514	044524	042526		
	050040	044517	052116		
1360	015	012		.BYTE	15,12
1361	032461	036440	047040	.ASCII	/15 = N = GRAPHLOT TEST/
	036440	043440	040522		
	044120	046120	052117		
	052040	051505	124		
1362	015	012		.BYTE	15,12
1363	001	020066	020075	.ASCII	/16 = O = LIGHT PEN FOLLOW,
	020117	020075	044514		
	044107	020124	042520		
	020116	047506	046114		
	053517				
1364	015	012		.BYTE	15,12
1365	033461	036440	050040	.ASCII	/17 = P = KEYBOARD ECHO TEST/
	036440	045440	054505		
	047502	051101	020104		
	041505	047510	052040		
	051505	124			
1366	015	012	012	.BYTE	15,12,12
1367	020040	052522	047502	.ASCII	/ RUBOUT TO REMAIN ON THE PATTERN,
	052125	052040	020117		
	042522	040515	047111		
	047440	020116	044124		
	020105	040520	052124		
	051105	116			
1368	015	012		.BYTE	15,12
1369	040	041440	020122	.ASCII	/ Q R TO SELECT SUB-PICTURE OR STOP MOTION /
	047524	051440	046105		
	041505	020124	052523		
	026502	044520	052103		
	051125	020105	051117		
	051440	047524	020120		
	047515	044524	047117		
	000040				
1370				.EVEN	
1371	027132	173400		DSTOP	
1372	027134	160000		DJMP	
1373	027136	005560		FRME0	
1374					
1375	027140			FRME1	
(1)	027140	170052		STATS:ITAL0;SYNOFF:GREEN	
1376	027142	116124		POINT:INT0;LPOFF:BLKOFF:LINE0	
1377	027144	041000		INTX=1000	
1378	027146	001000		MAXY=1/2	
1379	027130	040000		INTX=0	

1380	027132	000000		0	
1381	027134	041000		INTX=1000	
1382	027136	001000		MAXY=1/2	
1383	027140	041777		INTX=1777	
1384	027142	000000		0	
1385	027144	041000		INTX=1000	
1386	027146	001000		MAXY=1/2	
1387	027170	041777		INTX=1777	
1388	027172	001777		MAXY	
1389	027174	041000		INTX=1000	
1390	027176	001000		MAXY=1/2	
1391	027230	040000		INTX	
1392	027232	001777		MAXY	
1393	027234	164000		DNOP	
1394	027236	164000		DNOP	
1395	027210	164000		DNOP	
1396	027212	164000		DNOP	
1397	027214	164000		DNOP	
1398	027216	164000		DNOP	
1399	027218	164000		DNOP	
1400	027220	164000		DNOP	
1401	027222	173400		DSTOP	
1402	027224	160000		DJMP	
1403	027226	007140		FRME1	
1404					
1405					
1406	027230	116524		FRME2	
1407	027232	000000		POINT:INT2;LPOFF:BLKOFF:LINE0	
1408	027234	000000		0	
1409	027236	170052		STATS:ITAL0;SYNOFF:GREEN	
1410	027240	110000		LONGV	
1411	027242	041777		INTX:IMAXX	
1412	027244	000000		0	
1413	027246	040000		INTX	J X; +Y
1414	027230	001777		MAXY	J X; +Y
1415	027232	061777		INTX:IMINUSX:IMAXX	J X; +Y
1416	027234	000000		0	
1417	027236	040000		INTX	J X; -Y
1418	027238	021777		MINUSY:MAXY	J X; -Y
1419	027240	041777		INTX:IMAXX	J X; -Y
1420	027242	020000		MINUSY	J X; -Y
1421	027244	060000		INTX:IMINUSX	J X; -Y
1422	027270	001777		MAXY	J X; -Y
1423	027272	061777		INTX:IMINUSX:IMAXX	J X; -Y
1424	027274	020000		MINUSY	J X; -Y
1425	027276	060000		INTX:IMINUSX	J X; -Y
1426	027300	021777		MINUSY:IMAXX	
1427	027302	041777		INTX:IMAXX	
1428	027304	001777		MAXY	
1429	027306	061777		INTX:IMINUSX:IMAXX	
1430	027310	021777		MINUSY:IMAXX	
1431	027312	001777		MAXX	
1432	027314	000000		0	
1433	027316	061777		INTX:IMINUSX:IMAXX	

1434	007320	001777	MAXY
1435	007322	041777	INTXIMAXX
1436	007324	021777	MINUSXIMAXY
1437	007326	173400	DSTOP
1438	007330	160000	DJMP
1439	007332	007230	FRME2
1440			
1441			JOCTAGONS
1442			
1443	007334	117124	FRME3 POINT;INT4;LPOFF;BLKOFF;LINE0
1444	007336	000774	774
1445	007340	000564	564
1446	007342	170052	STATSA;ITAL;ISYNOFF;GREEN
1466	007344	110000	LONGV JOCTOGON BY LENGTH OF 7
(1)	007346	040007	INTX=7
(1)	007350	000000	0
(1)	007352	040007	INTX=7
(1)	007354	000007	7
(1)	007356	040000	INTX
(1)	007360	000007	7
(1)	007362	060007	INTXIMINUSX=7
(1)	007364	000007	7
(1)	007366	060007	INTXIMINUSX=7
(1)	007370	000000	0
(1)	007372	060007	INTXIMINUSX=7
(1)	007374	020007	MINUSX=7
(1)	007376	040000	INTX
(1)	007400	020007	MINUSX=7
(1)	007402	040007	INTX=7
(1)	007404	020007	MINUSX=7
1467	007406	114000	POINT
1468	007410	000770	770
1469	007412	000550	550
1470	007414	110000	LONGV JOCTOGON BY LENGTH OF 17
(1)	007416	040017	INTX=17
(1)	007420	000000	0
(1)	007422	040017	INTX=17
(1)	007424	000017	17
(1)	007426	040000	INTX
(1)	007430	000017	17
(1)	007432	060017	INTXIMINUSX=17
(1)	007434	000017	17
(1)	007436	060017	INTXIMINUSX=17
(1)	007440	000000	0
(1)	007442	060017	INTXIMINUSX=17
(1)	007444	020017	MINUSX=17
(1)	007446	040000	INTX
(1)	007450	020017	MINUSX=17
(1)	007452	040017	INTX=17
(1)	007454	020017	MINUSX=17
1471	007456	114000	POINT
1472	007460	000760	760
1473	007462	000520	520
1474	007464	110000	LONGV JOCTOGON BY LENGTH OF 37

(1)	007466	040037	INTX=37
(1)	007470	000000	0
(1)	007472	040037	INTX=37
(1)	007474	000037	37
(1)	007476	040000	INTX
(1)	007500	000037	37
(1)	007502	060037	INTXIMINUSX=37
(1)	007504	000037	37
(1)	007506	060037	INTXIMINUSX=37
(1)	007510	000000	0
(1)	007512	060037	INTXIMINUSX=37
(1)	007514	020037	MINUSX=37
(1)	007516	040000	INTX
(1)	007520	020037	MINUSX=37
(1)	007522	040037	INTX=37
(1)	007524	020037	MINUSX=37
1475	007526	114000	POINT
1476	007530	000740	740
1477	007532	000440	440
1478	007534	110000	LONGV JOCTOGON BY LENGTH OF 77
(1)	007536	040077	INTX=77
(1)	007540	000000	0
(1)	007542	040077	INTX=77
(1)	007544	000077	77
(1)	007546	040000	INTX
(1)	007550	000077	77
(1)	007552	060077	INTXIMINUSX=77
(1)	007554	000077	77
(1)	007556	060077	INTXIMINUSX=77
(1)	007560	000000	0
(1)	007562	060077	INTXIMINUSX=77
(1)	007564	020077	MINUSX=77
(1)	007566	040000	INTX
(1)	007570	020077	MINUSX=77
(1)	007572	040077	INTX=77
(1)	007574	020077	MINUSX=77
1479	007576	114000	POINT
1480	007600	000700	700
1481	007602	000300	300
1482	007604	110000	LONGV JOCTOGON BY LENGTH OF 177
(1)	007606	040177	INTX=177
(1)	007610	000000	0
(1)	007612	040177	INTX=177
(1)	007614	000177	177
(1)	007616	040000	INTX
(1)	007620	000177	177
(1)	007622	060177	INTXIMINUSX=177
(1)	007624	000177	177
(1)	007626	060177	INTXIMINUSX=177
(1)	007630	000000	0
(1)	007632	060177	INTXIMINUSX=177
(1)	007634	020177	MINUSX=177
(1)	007636	040000	INTX
(1)	007640	020177	MINUSX=177

(1)	007642	040177	INTX+177	
(1)	007644	020177	MINUSX+177	
1483	007646	114000	POINT	
1484	007650	000600	00P	
1485	007652	000000	0	
1486	007654	110000	LONGV	JOCTOGON BY LENGTH OF 377
(1)	007656	040377	INTX+377	
(1)	007660	000000	0	
(1)	007662	040377	INTX+377	
(1)	007664	000377	377	
(1)	007666	040000	INTX	
(1)	007670	000377	377	
(1)	007672	060377	INTX!MINUSX+377	
(1)	007674	000377	377	
(1)	007676	060377	INTX!MINUSX+377	
(1)	007700	000000	0	
(1)	007702	040377	INTX!MINUSX+377	
(1)	007704	020377	MINUSX+377	
(1)	007706	040000	INTX	
(1)	007710	020377	MINUSX+377	
(1)	007712	040377	INTX+377	
(1)	007714	020377	MINUSX+377	
1487	007716	173400	DSTOP	
1488	007720	160000	DJMP	
1489	007722	067334	FRMES	
1490			SQUARES	7,17,37,77,177,377,777 WIDE
1491				
1492	007724	117124	FRMESA	POINT!INT4!POFF!BLKOFF!LINE0
1493	007726	001000	1000	BY 7
1494	007730	000600	000	
1495	007732	170052	STATSAIITL@!SYH0FF!GREEN	
1509		000007	Q=7	
1510		000004	R=4	
1517	007734	110000	LONGV	BY 7 AND 4
(2)	007736	040007	INTX+7	
(2)	007740	000000	0	
(2)	007742	040000	INTX	
(2)	007744	000007	7	
(2)	007746	060007	INTX!MINUSX+7	
(2)	007750	000000	0	
(2)	007752	040000	INTX	
(2)	007754	020007	MINUSX+7	
(2)	007756	020004	MINUSX+4	
(2)	007760	020004	MINUSX+4	
(1)			.LIST	
(2)	007762	110000	LONGV	BY 17 AND 7
(2)	007764	040017	INTX+17	
(2)	007766	000000	0	
(2)	007770	040000	INTX	
(2)	007772	000017	17	
(2)	007774	060017	INTX!MINUSX+17	
(2)	007776	000000	0	
(2)	010000	040000	INTX	
(2)	010002	020017	MINUSX+17	

(2)	010004	020007	MINUSX+7	
(2)	010006	020007	MINUSX+7	
(1)			.LIST	
(2)	010010	110000	LONGV	BY 37 AND 17
(2)	010012	040037	INTX+37	
(2)	010014	000000	0	
(2)	010016	040000	INTX	
(2)	010020	000037	37	
(2)	010022	060037	INTX!MINUSX+37	
(2)	010024	000000	0	
(2)	010026	040000	INTX	
(2)	010030	020037	MINUSX+37	
(2)	010032	020017	MINUSX+17	
(2)	010034	020017	MINUSX+17	
(1)			.LIST	
(2)	010036	110000	LONGV	BY 77 AND 37
(2)	010040	040077	INTX+77	
(2)	010042	000000	0	
(2)	010044	040000	INTX	
(2)	010046	000077	77	
(2)	010050	060077	INTX!MINUSX+77	
(2)	010052	000000	0	
(2)	010054	040000	INTX	
(2)	010056	020077	MINUSX+77	
(2)	010060	020037	MINUSX+37	
(2)	010062	020037	MINUSX+37	
(1)			.LIST	
(2)	010064	110000	LONGV	BY 177 AND 77
(2)	010066	040177	INTX+177	
(2)	010070	000000	0	
(2)	010072	040000	INTX	
(2)	010074	000177	177	
(2)	010076	060177	INTX!MINUSX+177	
(2)	010080	000000	0	
(2)	010082	040000	INTX	
(2)	010084	020177	MINUSX+177	
(2)	010086	020077	MINUSX+77	
(2)	010088	020077	MINUSX+77	
(1)			.LIST	
(2)	010092	110000	LONGV	BY 377 AND 177
(2)	010094	040377	INTX+377	
(2)	010096	000000	0	
(2)	010098	040000	INTX	
(2)	010102	000377	377	
(2)	010104	060377	INTX!MINUSX+377	
(2)	010106	000000	0	
(2)	010108	040000	INTX	
(2)	010110	020377	MINUSX+377	
(2)	010112	020177	MINUSX+177	
(2)	010114	020177	MINUSX+177	
(2)	010116	000000	0	
(2)	010118	040000	INTX	
(2)	010122	000377	377	
(2)	010124	060377	INTX!MINUSX+377	
(2)	010126	000000	0	
(2)	010130	040000	INTX	
(2)	010132	020377	MINUSX+377	
(2)	010134	020177	MINUSX+177	
(2)	010136	020177	MINUSX+177	
(1)			.LIST	
(2)	010140	110000	LONGV	BY 777 AND 377
(2)	010142	040777	INTX+777	
(2)	010144	000000	0	

(2)	011146	040000				INTX
(2)	011150	000777				777
(2)	011152	060777				INTX MINUSX*777
(2)	011154	080000				0
(2)	011156	040000				INTX
(2)	011160	020777				MINUSX*777
(2)	011162	020377				MINUSX*377
(2)	011164	020377				MINUSX*377
(1)						.LIST
1518	011166	173400				DSTOP
1519	011170	160000				DJMP
1520	011172	007724				FRMESA
1521						
1522						DASH LINE TEST
1523						
1524	011174	117000				FRMESA; POINT;INT4
1525	011176	000000				0
1526	011200	001000				1000
1527	012202	174400				STATSA ITAL SYNOFF GREEN
1528	012204	170052				CHAR LINE0
1529	012206	100004				.BYTE 17,17
1530	012210	017				.ASCII /SOLID /
1531	012212	047523	044514	020104		
			020040			
1532	012224	110004				LONGV LINE0
1533	012226	040400				40400
1534	012230	000000				0
1535	012232	000400				400
1536	012234	000000				0
1537	012236	110030				LONGV BLK0N
1538	012240	040400				40400
1539	012242	000000				0
1540	012244	100020				CHAR BLKOFF
1541	012246	015	012	012		.BYTE 15,12,12,12,12,12
			012	012		
1542	012254	040504	044123	044440		.ASCII /DASH I /
			020040			
1543	012266	110005				LONGV LINE1
1544	012270	040400				40400
1545	012272	000000				0
1546	012274	000400				400
1547	012276	000000				0
1548	012300	110030				LONGV BLK0N
1549	012302	040400				40400
1550	012304	000000				0
1551	012306	100020				CHAR BLKOFF
1552	012310	015	012	012		.BYTE 15,12,12,12,12,12
			012	012		
1553	012316	040504	044123	044440		.ASCII /DASH II /
			020040			
1554	012330	110006				LONGV LINE2
1555	012332	040400				40400
1556	012334	000000				0
1557	012336	000400				400

1558	012340	000000				0
1559	012342	110030				LONGV BLK0N
1560	012344	040400				40400
1561	012346	000000				0
1562	012350	100020				CHAR BLKOFF
1563	012352	015	012	012		.BYTE 15,12,12,12,12,12
			012	012		
1564	012360	040504	044123	044440		.ASCII /DASH III /
			020040			
1565	012372	110007				LONGV LINE3
1566	012374	040400				40400
1567	012376	000000				0
1568	012400	000400				400
1569	012402	000000				0
1570	012404	110030				LONGV BLK0N
1571	012406	040400				40400
1572	012410	000000				0
1573	012412	110024				LONGV BLKOFF LINE0
1574	012414	000000				0
1575	012416	000000				0
1576	012420	173400				DSTOP
1577	012422	160000				DJMP
1578	012424	010174				FRMESA
1579						
1580						VECTOR LENGTH TEST <FILE 6 AND 7>
1581						
1582	012426	114000				FRMESA; POINT
1583	012430	001777				MAXX
1584	012432	000000				0
1585	012434	170052				STATSA ITAL SYNOFF GREEN
1586	012436	113724				LONGV INT7 BLKOFF LINE0
1587	012440	040000				INTX
1588	012442	001777				MAXY
1589	012444	114000				POINT
1590	012446	000000				0
1591	012450	001777				MAXY
1592	012452	110000				LONGV
1593	012454	041777				INTX MAXX
1594	012456	000000				0
1595	012460	173400				DSTOP
1596	012462	114000				FRMESA; POINT
1597	012464	000000				0
1598	012466	000000				0
1599	012470	110000				LONGV
1600	012472	000000				DELTX6; 0
1601	012474	000000				DELT6; 0
1602	012476	173400				DSTOP
1603	012500	160000				DJMP
1604	012502	010462				FRMESA
1605						
1606						
1607						PHOSPHOR TEST
1608						
1609	012504	114000				FRMESA; POINT

1610	010506	000000	DELTX7:	0
1611	010510	000000		0
1612	010512	170052	STATS:	ITALB SYNOFF GREEN
1613	010514	113724	DF110A:	LONGV INT7 LPOFF BLKOFF LINE0
1614	010516	040000		INTX
1615	010520	001777		MAXY
1616	010522	000002		2
1617	010524	000000		0
1618	010526	040000		INTX
1619	010530	021777		MINUSY MAXY
1620	010532	000002		2
1621	010534	000000		0
1622	010536	173400		DSTOP
1623	010540	160000		DJMP
1624	010542	010514		DF110A
1625				
1626				
1627				
1628	010544	114000	IPHOSPHOR TEST	
1629	010546	000000	FRM11:	POINT
1630	010550	000000		0
1631	010552	170052	DELY7:	0
1632	010554	113724	STATS:	ITALB SYNOFF GREEN
1633	010556	041777	DF111C:	LONGV INT7 LPOFF BLKOFF LINE0
1634	010560	000000		INTX MAXX
1635	010562	000000		0
1636	010564	000002		2
1637	010566	061777		INTX MINUSX MAXX
1638	010570	000000		0
1639	010572	000000		0
1640	010574	000002		2
1641	010576	173400		DSTOP
1642	010600	160000		DJMP
1643	010602	010554		DF111C
1644				
1645	010604	117604	FRM10:	POINT INT7 LINE0
1646	010606	000000		0
1647	010610	000000		0
1648	010612	110000		LONGV
1649	010614	041777		INTX MAXX
1650	010616	000000		0
1651	010620	040000		INTX
1652	010622	001777		MAXY
1653	010624	061777		INTX MINUSX MAXX
1654	010626	000000		0
1655	010630	040000		INTX
1656	010632	021777		MINUSX MAXY
1657	010634	173400		DSTOP
1658	010636	160000		DJMP
1659	010640	010604		FRM10
1660				
1661				
1662				
1663	010642	114164	INTENSITY TEST	
			FRM12:	POINT LINE0 LPOFF BLKOFF

1664	010644	000000		0
1665	010646	001200		1200
1666	010650	170252	SYN12:	STATS LPL R SYNOFF ITALB GREEN
1667	010652	103600		CHAR INT7
1668	010654	017		,BYTE 17,17
1669	010656	047111	042524	051516
	010660	052111	020131	020067
	010672	020040		
1670	010674	110000		LONGV
1671	010676	041000		41000
1672	010700	000000		0
1673	010702	130000		RELATV
1674	010704	057600		57600
1675	010706	103400		CHAR INT6
1676	010710	015	012	012
	010713	012		
1677	010714	047111	042524	051516
	010722	052111	020131	020066
	010730	020040		
1678	010732	110000		LONGV
1679	010734	041000		41000
1680	010736	000000		0
1681	010740	130000		RELATV
1682	010742	057600		57600
1683	010744	103200		CHAR INT5
1684	010746	015	012	012
	010751	012		
1685	010752	047111	042524	051516
	010760	052111	020131	020065
	010766	020040		
1686	010770	110000		LONGV
1687	010772	041000		41000
1688	010774	000000		0
1689	010776	130000		RELATV
1690	011000	057600		57600
1691	011002	102600		CHAR INT4
1692	011004	015	012	012
	011007	012		
1693	011010	047111	042524	051516
	011016	052111	020131	020064
	011024	020040		
1694	011026	110000		LONGV
1695	011030	041000		41000
1696	011032	000000		0
1697	011034	130000		RELATV
1698	011036	057600		57600
1699	011040	102600		CHAR INT3
1700	011042	015	012	012
	011045	012		
1701	011046	047111	042524	051516
	011054	052111	020131	020063
	011062	020040		
1702	011064	110000		LONGV
1703	011066	041000		41000

1704 011070 000000 0
 1705 011072 130000 RELATV
 1706 011074 057600 57600
 1707 011076 102400 CHARINT2
 1708 011100 015 012 012 ,BYTE 15,12,12,12
 1709 011103 012 012
 011104 047111 042524 051516 ,ASCII /INTENSITY 2 /
 011112 052111 020131 020002
 011120 020040
 1710 011122 110000 LONGV
 1711 011124 041000 41000
 1712 011126 000000 0
 1713 011130 130000 RELATV
 1714 011132 057600 57600
 1715 011134 102200 CHARINT1
 1716 011136 015 012 012 ,BYTE 15,12,12,12
 011141 012 012
 1717 011142 047111 042524 051516 ,ASCII /INTENSITY 1 /
 011150 052111 020131 020001
 011156 020040
 1718 011160 110000 LONGV
 1719 011162 041000 41000
 1720 011164 000000 0
 1721 011166 130000 RELATV
 1722 011170 057600 57600
 1723 011172 102000 CHARINT0
 1724 011174 015 012 012 ,BYTE 15,12,12,12
 011177 012 012
 1725 011200 047111 042524 051516 ,ASCII /INTENSITY 0 /
 011206 052111 020131 020000
 011214 020040
 1726 011216 110000 LONGV
 1727 011220 041000 41000
 1728 011222 000000 0
 1729 011224 130000 RELATV
 1730 011226 057600 57600
 1731 011230 164000 DNOP
 1732 011232 164000 DNOP
 1733 011234 164000 DNOP
 1734 011236 164000 DNOP
 1735 011240 164000 DNOP
 1736 011242 164000 DNOP
 1737 011244 164000 DNOP
 1738 011246 164000 DNOP
 1739 011250 173400 RAYLPA\$ DSTOP
 1740 011252 164000 DNOP
 1741 011254 164000 DNOP
 1742
 1743 011296 117100 DF12A\$ POINT:INT4:LP OFF
 1744 011200 001500 1500
 1745 011202 001200 LPPNT: 1200
 1746 011204 100000 CHAR
 1747 011206 044514 044124 ,ASCII /LIGHT PEN HIT/
 011274 042520 020116 044810

1748 011302 000124 ,EVEN
 1749 011304 173400 DSTOP
 1750 011306 160000 DJMP
 1751 011310 010642 FRME12
 1752
 1753 ,EDGE FILE
 1754
 1755 011332 117124 FRME13\$ POINT:INT4:LP OFF:BLK OFF:LINE0
 1756 011334 000000 0
 1757 011336 000000 0
 1758 011320 170052 STATSA:ITAL0:SYN0:FF:GREEN
 1759 011322 100000 CHAR
 1760 011324 017 017 ,BYTE 17,17
 1761 011326 110000 LONGV
 1762 011330 041777 INTX:MAXX
 1763 011332 000000 0
 1764 011334 040000 INTX
 1765 011336 001777 MAXY
 1766 011340 061777 INTX:MINUSX:MAXX
 1767 011342 000000 0
 1768 011344 040000 INTX
 1769 011346 021777 MINUSY:MAXY
 1770 011350 114000 POINT
 1771 011352 000100 100
 1772 011354 000300 300 ,LEFT SIDE
 1773 011356 110000 LONGV
 1774 011340 040000 INTX
 1775 011362 000400 400
 1776 011364 060200 INTX:MINUSX*200
 1777 011356 000000 0
 1778 011370 040000 INTX
 1779 011372 020400 MINUSY*400
 1780 011374 040200 INTX*200
 1781 011376 000000 0
 1782 011400 114000 POINT
 1783 011402 000200 200 ,TOP SIDE
 1784 011404 001700 MAXY*1=100
 1785 011406 110000 LONGV
 1786 011410 040400 INTX*400
 1787 011412 000000 0
 1788 011414 040000 INTX
 1789 011416 000200 200
 1790 011420 060400 INTX:MINUSX*400
 1791 011422 000000 0
 1792 011424 040000 INTX
 1793 011426 020200 MINUSY*200
 1794 011430 114000 POINT
 1795 011432 001700 1700 ,RIGHT SIDE
 1796 011434 001500 MAXY*1=300
 1797 011436 110000 LONGV
 1798 011440 040000 INTX
 1799 011442 020400 MINUSY*400
 1800 011444 040200 INTX*200

1001	011446	000000	0		
1002	011490	040000	INTX		
1003	011492	000400	400		
1004	011494	060200	INTX:MINUSX*200		
1005	011496	000000	0		
1006	011480	114000	POINT		
1007	011482	001600	1600		IBOTTOM SIDE
1008	011464	000100	100		
1009	011466	110000	LONGV		
1010	011470	060400	INTX:MINUSX*400		
1011	011472	000000	0		
1012	011474	040000	INTX		
1013	011476	020200	MINUSY*200		
1014	011500	040400	INTX*400		
1015	011502	000000	0		
1016	011504	040000	INTX		
1017	011506	000200	200		
1018	011510	114000	POINT		
1019	011512	001777	MAXX		
1020	011514	000400	400		
1021	011516	110000	LONGV		
1022	011520	000200	20		
1023	011522	000000	0		
1024	011524	100000	CHAR		
1025	011526	015	,BYTE 15,101		"CR" AND AN "A"
1026	011530	114000	POINT		
1027	011532	000000	0		
1028	011534	000500	500		
1029	011536	110000	LONGV		
1030	011540	020012	MINUSX*12		
1031	011542	000000	0		
1032	011544	100000	CHAR		
1033	011546	040	,BYTE 40,102		"SPACE" AND AN "B"
1034	011550	164000	DNOP		
1035	011552	164000	DNOP		
1036	011554	173400	DSTOP		
1037	011556	164000	DNOP		
1038	011560	164000	DNOP		
1039	011562	160000	DJMP		
1040	011564	011312	FRME13		
1041					

1043					
1057					
1058	011566	170052	FRME14	STATS:ITAL:ISYNOFF:IGREEN	
1059	011570	117124	POINT	INT4:BLKOFF:LP:OFF:LINE0	
1060	011572	000000	FRM14A	0	
1061	011574	000000	FRM14B	0	
1062	011576	104000			
1063	011600	056200	SHORTV		
(1)	011602	056271	INTX*16200		
(1)	011604	040071	INTX*16200*71		
(1)	011606	076271	INTX*71		
(1)	011610	076200	INTX:MINUSX*16200*71		
(1)	011614	076371	INTX:MINUSX*16200		
(1)	011616	040171	INTX:MINUSX*16200*MINUSY*71		
(1)	011618	056371	INTX*MINUSY*71		
(1)	011620	020504	INTX*16200*MINUSY*71		
(1)	011622	164000	20504		
(1)	011624	164000	DNOP		
1064	011606	130000	DNOP		
1065	011630	057000	RELATV		
(1)	011632	057074	INTX*17000		
(1)	011634	040074	INTX*17000*74		
(1)	011636	077074	INTX*74		
(1)	011640	077000	INTX:MINUSX*17000*74		
(1)	011642	077174	INTX:MINUSX*17000		
(1)	011644	040174	INTX:MINUSX*17000*MINUSY*74		
(1)	011646	057174	INTX*MINUSY*74		
(1)	011650	020504	INTX*17000*MINUSY*74		
(1)	011652	164000	20504		
(1)	011654	164000	DNOP		
1066	011636	104000	DNOP		
1067	011680	057600	SHORTV		
(1)	011682	057677	INTX*17600		
(1)	011684	040077	INTX*17600*77		
(1)	011686	077677	INTX*77		
(1)	011690	077600	INTX:MINUSX*17600*77		
(1)	011692	077777	INTX:MINUSX*17600		
(1)	011694	040177	INTX:MINUSX*17600*MINUSY*77		
(1)	011696	057777	INTX*MINUSY*77		
(1)	011700	020504	INTX*17600*MINUSY*77		
(1)	011702	164000	20504		
(1)	011704	164000	DNOP		
1068	011706	173400	DNOP		
1069	011710	160000	DSTOP		
1070	011712	011566	DJMP		
			FRME14		

1872							
1873	011714	117124			FRME16#	POINTINT4ILPOFFIBLKOFFILIN0	
1874	011716	000000				0	
1875	011720	001600				MAXY=177	
1876	011722	170052				STATSAITAL@ISYNOFFIGREEN	
1877	011724	100000				CHAR	
1878	011726	017				.BYTE 17,17	
1879	011730	044514	044107	020124		.ASCII /LIGHT REN FOLLOW TEST /	
	011736	042520	020116	047506			
	011744	046114	053517	052040			
	011752	051500	020124				
1880	011756	019		012		.BYTE 15,12,12	
1881	011761	130		020075		.ASCII /X= /	
1882	011764	030061		030060	DLT14A#	.ASCII /1000/	
1883	011770	015		012		.BYTE 15,12,12	
1884	011773	131		020075		.ASCII /Y= /	
1885	011776	033060		030060	DLT14B#	.ASCII /0000/	
1886	012002	114140				POINTILPON	
1887	012004	001000			RAT14A#	1000	
1888	012006	000600			RAY14B#	600	
1889	012010	164000				DNOP	
1890	012012	164000				DNOP	
1891	012014	130000				RELATV	
1892	012016	074000			TAB16A#	INTXIMINUSX*14000	
1895	012020	041000				INTX+1000	
(1)	012022	041000				INTX+1000	
(1)	012024	041000				INTX+1000	
(1)	012026	041000				INTX+1000	
(1)	012030	041000				INTX+1000	
(1)	012032	041000				INTX+1000	
(1)	012034	041000				INTX+1000	
(1)	012036	041000				INTX+1000	
(1)	012040	041000				INTX+1000	
(1)	012042	041000				INTX+1000	
(1)	012044	041000				INTX+1000	
(1)	012046	041000				INTX+1000	
(1)	012050	041000				INTX+1000	
(1)	012052	041000				INTX+1000	
(1)	012054	041000				INTX+1000	
(1)	012056	041000				INTX+1000	
(1)	012060	041000				INTX+1000	
(1)	012062	041000				INTX+1000	
(1)	012064	041000				INTX+1000	
(1)	012066	041000				INTX+1000	
(1)	012070	041000				INTX+1000	
(1)	012072	041000				INTX+1000	
(1)	012074	041000				INTX+1000	
(1)	012076	041000				INTX+1000	
1896	012100	164000				DNOP	
1897	012102	164000				DNOP	
1898	012104	164000				DNOP	
1899	012106	164000				DNOP	
1900	012110	164000				DNOP	
1901	012112	164000				DNOP	

1902	012114	164000				DNOP	
1903	012116	164000				DNOP	
1904	012120	164000				DNOP	
1905	012122	164000				DNOP	
1906	012124	130000				RELATV	
1907	012126	034000				MINUSX*14000	
1908	012130	040160			TAB16B#	INTXIMINSUV*60	
1911	012132	040004				INTX+4	
(1)	012134	040004				INTX+4	
(1)	012136	040004				INTX+4	
(1)	012140	040004				INTX+4	
(1)	012142	040004				INTX+4	
(1)	012144	040004				INTX+4	
(1)	012146	040004				INTX+4	
(1)	012150	040004				INTX+4	
(1)	012152	040004				INTX+4	
(1)	012154	040004				INTX+4	
(1)	012156	040004				INTX+4	
(1)	012160	040004				INTX+4	
(1)	012162	040004				INTX+4	
(1)	012164	040004				INTX+4	
(1)	012166	040004				INTX+4	
(1)	012170	040004				INTX+4	
(1)	012172	040004				INTX+4	
(1)	012174	040004				INTX+4	
(1)	012176	040004				INTX+4	
(1)	012200	040004				INTX+4	
(1)	012202	040004				INTX+4	
(1)	012204	040004				INTX+4	
(1)	012206	040004				INTX+4	
(1)	012210	040004				INTX+4	
1912	012212	164000				DNOP	
1913	012214	164000				DNOP	
1914	012216	164000				DNOP	
1915	012220	164000				DNOP	
1916	012222	164000				DNOP	
1917	012224	164000				DNOP	
1918	012226	164000				DNOP	
1919	012230	164000				DNOP	
1920	012232	110000				LONGV	
1921	012234	000000				0	
1922	012236	000077				77	
1923	012240	040160			PNT16A#	INTX+160	
1924	012242	020160				MINUSX*160	
1925	012244	060160			PNT16B#	INTXIMINUSX*160	
1926	012246	020160				MINUSX*160	
1927	012250	060160			PNT16C#	INTXIMINUSX*160	
1928	012252	000160				160	
1929	012254	040160			PNT16D#	INTX+160	
1930	012256	000160				160	
1931	012260	164000				DNOP	
1932	012262	164000				DNOP	
1933	012264	164000				DNOP	
1934	012266	164000				DNOP	

```

1935 012270 164000      DNOP
1936 012272 164000      DNOP
1937 012274 173400      DSTOP
1938 012276 160000      DJMP
1939 012300 011714      FRME16
1940
1941 012302 117124      FRM16A POINT;INT4;LPOFF;BLKOFF;LINE0
1942 012304 000000      0
1943 012306 001600      MAXY=177
1944 012308 170002      STATAI;ITAL;ISYNOFF;GREEN
1945 012312 100000      CHAR
1946 012314      017      .BYTE 17,17
1947 012316 044514 044107 020124      .ASCII /LIGHT PEN FIELD OF VIEW /
      012324 042520 020116 044506
      012302 040105 020104 043117
      012308 053040 042511 020127
1948 012346      015      012      012      .BYTE 15,12,12
1949 012391      116 046525 042502      .ASCII /NUMBER OF HITS = 0000/
      012396 020122 043117 044040
      012304 052111 020123 020075
      012372 030060 030060
1950 012376 173400      FRM16B DSTOP
1951 012400 160000      DJMP
1952 012402 012302      FRM16A
1953
1954 012404 114124      FRME17 POINT;LPOFF;BLKOFF;LINE0      MUST BE JUST BEFORE THE BUFFER
1955 012406 000000      0
1956 012408 001600      MAXY=177
1957 012412 170002      STATAI;ITAL;ISYNOFF;GREEN
1958 012414 103000      CHAR;INT4
1959 012416      017      017      .BYTE 17,17
1960 012420 042513 041131 040517      .ASCII /KEYBOARD ECHO TEST/
      012426 042122 040440 044103
      012434 020117 042524 052123
      012442      000
1961 012443      015      012      012      .BYTE 15,12,12
1962 012446 044103 051101 047440      .ASCII /CHAR OCT = /
      012454 052103 036440 040
1963 012401      000      000      .BYTE 0,0,0,0
      012404      000
1964 012405      015      012      012 KBOCT; .BYTE 15,12,12
1965
1966 012470 164000      BUFFER; DNOP
1967
1968      000001      .END
    
```

SYMBOL TABLE

BIF	= 000200	BLKOFF	= 000020	BLKON	= 000030	BUFFER	012470
CHANGE	001000	CHAR	100000	CHRCNT	000330	CNTR	001040
COUNT	005052	CRLF	002070	DBUF	001010	DBUF1	001020
DBUF2	001022	DBUF3	001024	DBUF4	001020	DBUF5	001030
DDCNE	001064	DDONE1	001066	DELTX6	010472	DELTX7	010500
DELTY6	010474	DELTY7	010550	DFI10A	010514	DFI11C	010550
DPI12A	011250	DFL15B	004242	DFL15C	004144	DFL15D	004220
DSPLA	117970	DISPTC	001210	DJMP	160000	DLT140	011700
DLT14B	011776	DNOP	164000	DPC	001054	OSAVE	001032
OSAVE1	001030	OSAVE2	001036	OSAVE3	001040	OSR	001050
DSTOP	173400	O7A	003330	O7C	003360	O7D	003400
O7F	003430	ECHOA	005060	ECHOB	005310	ECHOC	005100
FILE	005050	FILE0	002050	FILE1	002062	FILE10	003322
FILE11	003370	FILE12	003452	FILE13	003014	FILE14	003020
FILE15	004100	FILE16	004342	FILE17	005052	FILE2	002074
FILE2A	002270	FILE2B	002302	FILE2C	004322	FILE2D	002330
FILE3	002340	FILE3A	002350	FILE3B	004370	FILE4	002414
FILE4A	002740	FILE5	003024	FILE6	003030	FILE7	003170
FILLA	002710	FILLIT	002712	FIL14A	003730	FLE12A	003470
FILE12B	003914	FILE12C	003522	FILE12D	003600	FRME0	005560
FRME1	007140	FRME10	010504	FRME11	010544	FRME12	010640
FRME13	011312	FRME14	011566	FRME16	011714	FRME17	012400
FRME2	007230	FRME3	007334	FRME3A	007724	FRME5	010174
FRME6	010420	FRME6A	010462	FRM10	010000	FRM14A	011570
FRM14B	011974	FRM16A	012302	FRM16B	010370	GRAPHX	120000
GRAPHY	124000	GREEN	000002	GSADD	001000	GSBRL	001004
GSVCT	001002	HERE	004330	HITGWR	001302	HITGNT	005050
HOLD	001042	ICNT	001006	INCR	000100	INTX	040000
INT0	002000	INT1	002200	INT2	002400	INT3	002600
INT4	003000	INT5	003200	INT6	004400	INT7	003600
ITAL0	000040	ITAL1	000060	KBCHR	000332	KBOCT	012465
KRBD	002040	KYT1	002014	KYT2	001774	KYT3	002024
KYT4	002030	KYT5	001736	LADVT	003710	LINE0	000004
LINE1	000000	LINE2	000006	LINE3	000007	LOADAC	004534
LOADBF	002024	LOADSP	002564	LOADUP	001556	LOADVT	003700
LOGICA	004310	LOKR8	001052	LONGV	110000	LOOPA	003062
LOOPA1	003070	LOOPA2	003114	LOOPA3	003140	LOOPB	003214
LOOPB1	003222	LOOPB2	003246	LOOPB3	003272	LOWPWR	001250
LOH5V	001300	LPDARK	000300	LPLITE	000200	LPOFF	000100
LPON	000140	LPPNT	011262	LPVCT	001070	LPVCT1	001072
MAXSX	017400	MAXSY	000077	MAXX	001777	MAXY	001777
MESG	005412	MESGA	005430	MESGAA	005444	MESGAB	005510
MESGB	005464	MESGBA	005514	MINSUY	000100	MINUSX	020000
MINSUY	010000	PC	X000007	PNT16A	012240	PNT16B	012244
PNT16C	012250	PNT16D	012254	POINT	110000	PSW	001010
Q	001777	R	000777	RAY10A	011250	RAY14A	012004
RAY14B	012006	RED	000003	RELATV	130000	RET0	001700
RETL0	013950	RET17	005206	RET14	000014	RET20	005240
RET21	005230	R0	X000000	R1	X000001	R2	X000002
R3	X000003	R4	X000004	R5	X000005	SCOPE	100000
SCOPEA	001100	SCOPEB	001156	SCOPEC	001162	SCOPEE	001170
SCOPEE	001120	SCOPEF	001136	SETPNT	002252	SETUP	001936
SETUPA	001022	SETUPB	001664	SHORTV	100000	SIZE0	000400
SIZE1	000000	SP	X000006	SPACE	002730	START	001350

SYTSA = 17000*	SYTSA = 174000	SYCHAR = 000060	SYKPTR = 000000
SYRA = 001400	SYRB = 001424	SYRC = 001450	SWITCH = 000000
SYNOFF = 000010	SYNON = 000014	SYN12 = 010050	TAB16A = 012010
TAB16B = 012130	TIMEVT = 001074	TKB = 001014	TKS = 001010
TNEVT1 = 001076	TSAVE = 001044	XPOS = 001060	VPOS = 001000
" = 012472			

ERRORS DETECTED: 0