1   ;REM   *

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

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DDGTD-B-D

PRODUCT NAME: GT40 ROM VERIFY

DATE CREATED: NOVEMBER 1, 1973

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: RAYMOND SHOOP

1.   ABSTRACT

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

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

2.   REQUIREMENTS

2.1   EQUIPMENT

        GT40 DISPLAY PROCESSOR WITH ROM BOOTSTRAPVERSION 1 OR 2.

2.2   STORAGE

        THIS PROGRAM USES MEMORY LOCATIONS 0-7776 + 16000-16774(8).

3.   LOADING PROCEDURE

        PROCEDURE FOR A NORMAL BINARY TAPE SHOULD BE FOLLOWED.

4.   STARTING PROCEDURE

4.1   STARTING ADDRESSES

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

4.2   SWITCH SETTINGS

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

5.   PROGRAM DESCRIPTIONS

5.1   PRG0 + LOGIC TESTS

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

5.1.1   ROUTINE DESCRIPTIONS

          ROUTINE   TESTS

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

5.2   PRG1 - ROM DATA DUMP

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

5.3   PRG2 - SINGLE ROM ADDRESS READ DATA LOOP

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

6.   ERRORS

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

7.   EXECUTION TIME

          PRG0 TAKES APPROX. 5 SECONDS PER PASS,
          PROG1 N/A
          PROG2 N/A

```
151                                    .LIST ME,BIN,SEQ,LD
152                                    .ENABL ABS,AMA
153
154                            ;LOAD ADDRESS=0200
155                            ;DEPRESS START
156                            ;STACK POINTER IS A] 500
157
164                                    .LIST
165          000034                    .=34
166  002034  002054                    SCOPEC
167  002036  000000                    0
168                            ;EQUATE STATEMENTS
169          104400                    SCOPE=TRAP
170          177564                    TPCSR=177564
171          177566                    TPDBR=177566
172          177776                    PSW=177776
173          177570                    SR=177570
174          177570                    DISPLAY=177570      ;ADDRESS OF DISPLAY REGISTER
175          000500                    STKPTR=500          ;INITIAL STACK SETTING
176          000200                    .=200
177  002200  000137  001024            JMP     PRMTRS
178  002204  000137  001506            JMP     PRG1
179  002210  000137  001670            JMP     PRG0
```

```
181
182          001000                    .=1000
183  001000  166000            ROMADD: 166000                        ;ROM ADDRESS
184  001002  001000            WORDS:  512.            ;258?          ;ROM LENGTH
185  001004  006000            IMAGE:  START           ;STARTA        ;ROM IMAGE
186  001006  172002            DSR:    172002                        ;DISPLAY STATUS REGISTER
187  001010  000010            FILLER: 10                             ;# OF FILLER CHAR
188  001012  000010            FILCNT: 10
189  001014  000000            ;CNT:   0
190  001016  000000            DUMP:   0
191  001020  000000            CHARA:  0
192  001022  000000            TERM:   0
193  001024  012706  000500    PRMTRS: MOV     #STKPTR,X6             ;SET STACK PTR
194  001030  004737  002306            JSR     PC,SWITCH              ;CHECK ROM VERSION
195
196                            ;PROGRAM 0 LOGIC TESTS
197
198  001034  005037  001014    PRG0:   CLR     ;CNT                   ;CLEAR PASS COUNT
199  001040  012706  000500    PRG0R:  MOV     #STKPTR,X6
200  001044  012737  001040  002130    MOV     #PRG0R,RETURN          ;SET RETURN ADDRESS FOR SCOPE
201  001052  013737  001014  177570    MOV     ;CNT,@#DISPLAY         ;DISPLAY PASS COUNT
202
203                            ;TEST1 TEST ABILITY TO REFERENCE ROM WITHOUT TIMING OUT
204
205  001060  013700  001000    T1:     MOV     ROMADD,%0              ;GET ROM ADDRESS
206  001064  013701  001002            MOV     WORDS,X1               ;GET ADDRESS COUNTER
207  001070  012737  001130  000004            MOV     #ERROR1,4      ;SET UP TIME OUT VECTOR
208  001076  011003            T1A:    MOV     (0),%3                 ;REFERENCE
209  001100  005720                    TST     (0)+                   ;ROM
210  001102  064037  001016            ADD     -(0),DUMP              ;
211  001106  021010                    CMP     (0),(0)                ;
212  001110  132020                    BITB    (0)+,(0)+              ;
213  001112  164037  001016            SUB     -(0),DUMP              ;
214  001116  062700  000002            ADD     #2,%0                  ;INCREMENT POINTER
215  001122  005301                    DEC     X1                     ;DECREMENT ADDRESS COUNTER
216  001124  001364                    BNE     T1A                    ;BRANCH IF NOT FINISHED
217  001126  000403                    BR      T1B                    ;GO TO SCOPE LOOP
218  001130  022626            ERROR1: CMP     (6)+,(6)+              ;REPOSITION STACK
219  001132  000000                    HALT                           ;ERROR, TIME-OUT  ON ROM ADDRESS
220  001134  000768                    BR      T1A                    ;LOOP ON ERROR
221  001136  104400            T1B:    SCOPE
222
```

```
 224
 225                                    ;TEST2  TEST THAT ROM DATA CAN BE READ RELIABLY;
 226
 227  001140  013700  001000      T2T   MOV     ROMADD,R0      ;GET ROM ADDRESS
 228  001144  013701  001002            MOV     WORDS,R1       ;GET ADDRESS COUNTER
 229  001150  012737  000006  000004     MOV     #6,@          ;INITIALIZE TIME OUT VECTOR
 230  001156  005037  001016      T2A1  CLR     DUMP           ;INITIALIZE DUMP
 231  001162  011003            MOV     (R0),R3        ;GET DATA
 232  001164  062037  001016            ADD     (R0)+,DUMP     ;ADD DATA TO DUMP
 233  001170  163703  001016            SUB     DUMP,R3        ;SUBTRACT DATA FROM DATA
 234  001174  001402            BEQ     T2B            ;BRANCH IF EQUAL
 235  001176  000000      ERROR2; HALT                 ;DATA ERROR
 236  001200  000766            BR      T2A            ;LOOP ON ERROR
 237  001202  044037  001016      T2B1  BIC     @(0)+DUMP      ;CLEAR DUMP BITS
 238  001206  001402            BEQ     T2C            ;BRANCH IF EQUAL TO 0
 239  001210  000000            HALT                   ;DATA ERROR
 240  001212  000773            BR      T2B            ;LOOP ON ERROR
 241  001214  021010      T2C1  CMP     (0),(0)        ;COMPARE DATA
 242  001216  001402            BEQ     T2D            ;BRANCH IF EQUAL
 243  001220  000000            HALT                   ;DATA ERROR
 244  001222  000774            BR      T2C            ;LOOP ON ERROR
 245  001224  122040      T2D1  CMPB    (0)+,-(0)      ;COMPARE DATA (BYTE OPERATION)
 246  001226  001402            BEQ     T2E            ;BRANCH IF EQUAL
 247  001230  000000            HALT                   ;DATA ERROR
 248  001232  000774            BR      T2D            ;LOOP ON ERROR
 249  001234  005720      T2E1  TST     (0)+           ;INCREMENT ADDRESS POINTER
 250  001236  005301            DEC     R1             ;DECREMENT ADDRESS COUNTER
 251  001240  001346            BNE     T2A            ;RETURN IF NOT DONE
 252  001242  104400            SCOPE
 253
```

```
 255
 256                                    ;TEST3  TEST THAT ROM TIMES OUT IF REFERENCED BY OTHER
 257                                    ;THAN DATI BUS CYCLE
 258
 259  001244  012706  000500      T3T   MOV     #STKPTR,R6     ;SET STACK PTR
 260  001250  013700  001000            MOV     ROMADD,R0      ;GET ROM ADDRESS
 261  001254  013701  001002            MOV     WORDS,R1       ;GET ADDRESS COUNTER
 262  001260  012737  001274  000004     T3AA1  MOV     #T3B,@         ;SET UP TIME OUT VECTOR
 263  001266  010010      T3A1  MOV     R0,(0)         ;ATTEMPT TO ALTER DATA
 264  001270  000000            HALT                   ;HERE IF DID NOT TIME OUT
 265  001272  000775            BR      T3A            ;LOOP ON ERROR
 266  001274  012737  001312  000004     T3B1  MOV     #T3C,@         ;SET UP TIME OUT VECTOR
 267  001302  022626            CMP     (0)+,(0)+      ;REPOSITION STACK
 268  001304  005210      T3C1  INC     (0)            ;ATTEMPT TO ALTER DATA
 269  001306  000000            HALT                   ;HERE IF DID NOT TIME OUT
 270  001310  000775            BR      T3C            ;LOOP ON ERROR
 271  001312  012737  001332  000004     T3D1  MOV     #T3E,@         ;SET UP TIME OUT VECTOR
 272  001320  022626            CMP     (0)+,(0)+      ;REPOSITION STACK
 273  001322  005077  177452      T3E1  CLR     @ROMADD        ;ATTEMPT TO ALTER DATA
 274  001326  000000            HALT                   ;HERE IF DID NOT TIME OUT
 275  001330  000774            BR      T3E            ;LOOP ON ERROR
 276  001332  005720      T3F1  TST     (0)+           ;INCREMENT ADDRESS POINTER
 277  001334  022626            CMP     (0)+,(0)+      ;REPOSITION STACK
 278  001336  005301            DEC     R1             ;DECREMENT ADDRESS COUNTER
 279  001340  001347            BNE     T3AA           ;RETURN IF NOT DONE
 280  001342  012737  000006  000004     MOV     #6,@          ;RESTORE TIME OUT TRAP
 281  001350  104400            SCOPE                  ;SCOPE LOOP
```

```
 283
 284                                    ;COMPARE THE ROM DATA TO THE IMAGE DATA
 285                                    ;      R0=WORD NUMBER
 286                                    ;      R1=GOOD ADDRESS
 287                                    ;      R2=GOOD DATA
 288                                    ;      R3=BAD ADDRESS
 289                                    ;      R4=BAD DATA
 290
 291   001352  012700  000000    T4T    MOV    #0,R0             ;SET UP INITIAL WORD COUNT
 292   001356  012701  001004           MOV    IMAGE,R1         ;SET UP STARTING ADDRESS OF ROM IMAGE
 293   001362  012703  001000           MOV    ROMADD,R3        ;SET UP STARTING ROM ADDRESS
 294   001366  011102           T4A:    MOV    (R1),R2          ;READ EXPECTED VALUE
 295   001370  011304                   MOV    (R3),R4          ;READ ROM VALUE
 296   001372  020204                   CMP    R2,R4            ;COMPARE EXPECTED TO THE VALUE READ
 297   001374  001402                   BEQ    T4B              ;BRANCH IF CORRECT
 298   001376  000000                   HALT                    ;ERROR; ROM VALUE FAILED TO EQUAL EXPECTED
 299   001400  000772                   BR     T4A
 300
 301   001402  022123           T4B:    CMP    (R1)+,(R3)+      ;INCREMENT ADDRESSES POINTERS
 302   001404  005200                   INC    R0               ;INCREMENT WORD COUNT
 303   001406  023700  001002           CMP    WORDS,R0         ;COMPARE IF END WORD
 304   001412  001365                   BNE    T4A              ;BRANCH IF NOT LAST WORD
 305   001414  104400           T4E:    SCOPE
 306
 307   001416  005237  001014    END:   INC    ICNT             ;INCREMENT PASS COUNT
 308   001422  012777  000001  177356          MOV    #1,@DSR          ;RING THE GT40 BELL
 309   001430  012737  000207  177566  DONE0:  MOV    #207,@#TPDBR     ;RING THE TELETYPE BELL
 310   001436  105737  177564           TSTB   @#TPCSR
 311   001442  100375                   BPL    .-4
 312   001444  012737  000207  177566          MOV    #207,@#TPDBR
 313   001452  105737  177564    1$:    TSTB   @#TPCSR
 314   001456  100375                   BPL    1$
 315   001460  013700  000042           MOV    @#44,R0          ;RETURN TO DECTAPE MONITOR?
 316   001464  001406                   BEQ    DONE1
 317   001466  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  DONE1:  JMP    PRG0
 324
```

```
 326
 327                                    ;THIS PROGRAM TYPES OUT ROM DATA
 328
 329   001506  012706  000500    PRG0:  MOV    #STKPTR,R6       ;INITIALIZE STACK
 330   001512  012737  000006  000004          MOV    #6,@#4           ;SET UP BUSS ERROR
 331   001520  004737  002306           JSR    PC,SWITCH
 332   001524  004537  001712           JSR    5,TYPEM
 333   001530  002275                   M6
 334   001532  004537  001712           JSR    5,TYPEM          ;TYPE MESSAGE
 335   001536  002252                   M7               ;ROM DATA!
 336   001540  013701  001002           MOV    WORDS,R1         ;GET # OF WORDS
 337   001544  013700  001000    PRG1A: MOV    ROMADD,R0        ;GET STARTING ADDRESS
 338   001550  012702  000010           MOV    #10,R2           ;GET ADDRESS INDICATOR
 339   001554  105737  177564           TSTB   TPCSR            ;WAIT FOR
 340   001560  100375                   BPL    .-4              ;TELEPRINTER FLAG
 341   001562  010037  002144    PRG1B: MOV    R0,@#TYP         ;GET ADDRESS
 342   001566  004737  002146           JSR    7,OBA            ;AND TYPE IT
 343   001572  004537  001712           JSR    5,TYPEM          ;TYPE
 344   001576  002301                   M9               ;CR/LF
 345   001600  012037  002144    PRG1C: MOV    (R0)+,@#TYP      ;TYPE
 346   001604  004737  002146           JSR    7,OBA            ;DATA
 347   001610  105737  177564           TSTB   TPCSR            ;WAIT FOR
 348   001614  100375                   BPL    .-4              ;TELEPRINTER FLAG
 349   001616  012737  000040  177566          MOV    #' ,TPDBR        ;TYPE SPACE
 350   001624  005301                   DEC    R1               ;ALL DATA TYPED
 351   001626  001410                   BEQ    PRG1D            ;GO TO FINISH
 352   001630  005302                   DEC    R2
 353   001632  001362                   BNE    PRG1C            ;RETURN TO PRG1B
 354   001634  012702  000010           MOV    #10,R2           ;GET ADDRESS INDICATOR
 355   001640  004537  001712           JSR    5,TYPEM          ;TYPE
 356   001644  002275                   M6               ;CR/LF
 357   001646  000745                   BR     PRG1B            ;RETURN TO PRG1B
 358   001650  004537  001712    PRG1D: JSR    5,TYPEM
 359   001654  002275                   M6
 360   001656  004537  001712           JSR    5,TYPEM
 361   001662  002275                   M6
 362   001664  000000                   HALT
 363   001666  000707                   BR     PRG1
 364
 365                                    ;ROUTINE TO LOOP ON A SINGLE ADDRESS
 366
 367   001670  012706  000500    PRG2:  MOV    #STKPTR,R6
 368   001674  012737  000006  000004          MOV    #6,@#4
 369   001702  013700  177570           MOV    SR,R0
 370   001706  011001                   MOV    (R0),R1
 371   001710  000767                   BR     PRG2
```

```
                                        ;ROUTINE TO TYPE A MESSAGE
373
374
375  001712  010026              TYPEM:  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  TYPEMA:  MOVB   (0)+,CHARA      ;GET NEXT CHARACTER
379  001726  123737      001020  001022   CMPB   CHARA,TERM      ;WAS NEXT CHARACTER THE TERM
380  001734  001005                       BNE    TYPEMB          ;CHARACTER
381  001736  014600                       MOV    -(6),X0         ;RESTORE R0
382  001740  105737      177564           TSTB   TPOBR
383  001744  100375                       BPL    .-4
384  001746  000205                       RTS    5               ;AND EXIT
385  001750  123727      001020  000045  TYPEMB:  CMPB   CHARA,#!X       ;WAS CHARACTER X
386  001756  001027                       BNE    TYPEMC
387  001760  105737      177564           TSTB   TPOBR           ;TEST TELEPRINTER FLAG
388  001764  100375                       BPL    .-4             ;AND WAIT FOR DONE
389  001766  012737      000215  177566   MOV    #215,TPOBR      ;LOAD TELEPRINTER WITH CAR, RET
390  001774  013737      001010  001012   MOV    FILLER,FILCNT          ;LOAD FILLER COUNT
391  002002  000403                       BR     1$
392  002004  012737      000006  177566  2$:  MOV    #6,TPOBR        ;PRINT FILLER CHAR
393  002012  105737      177564  1$:  TSTB   TPOBR           ;TEST TELEPRINTER FLAG
394  002016  100375                       BPL    .-4             ;AND WAIT FOR DONE
395  002020  005337      001012           DEC    FILCNT          ;FINISHED FILLERS ,
396  002024  001367                       BNE    2$              ;BR IF NOT
397  002026  012737      000212  177566   MOV    #212,TPOBR      ;LOAD TELEPRINTER WITH LINE FEED
398  002034  000732                       BR     TYPEMA          ;GET NEXT CHARACTER
399  002036  105737      177564  TYPEMC:  TSTB   TPOBR           ;TEST TELEPRINTER FLAG
400  002042  100375                       BPL    .-4             ;AND WAIT FOR DONE
401  002044  013737      001020  177566   MOV    CHARA,TPOBR     ;LOAD TELEPRINTER BUFFER
402  002052  000723                       BR     TYPEMA          ;AND GET NEXT CHARACTER
403
404                                      ;SCOPE ROUTINE. THIS ROUTINE IS ENTERED AT THE END OF EACH SUBTEST.
405
406  002054  032737      040000  177570  SCOPEC:  BIT    #40000,SR       ;TEST SR FOR SCOPE
407  002062  001023                       BNE    SCOPEB          ;YES SCOPE
408  002064  032737      004000  177570   BIT    #4000,SR        ;TEST FOR ITERATION
409  002072  001007                       BNE    SCOPEG          ;INHIBIT ITERATION
410  002074  023737      002126  002124   CMP    SCOPEF,ICOUNT   ;ITERATION COMPLETE
411  002102  001403                       BEQ    SCOPEG          ;ITERATION COMPLETE GO TO SCOPEG
412  002104  005237      002126           INC    SCOPEF          ;INCREMENT ITERATION COUNT
413  002110  000410                       BR     SCOPEB          ;GO TO SCOPEB
414  002112  005037      002126  SCOPEG:  CLR    SCOPEF          ;CLEAR ITERATION COUNT
415  002116  011637      002130           MOV    @X6,RETURN      ;GET ADDRESS OF NEXT TEST
416  002122  000002                       RTI                    ;EXIT
417  002124  000100              ICOUNT:  100
418  002126  000000              SCOPEF:  0                       ;CONTAINS SUBTEST ITERATION COUNT
419  002130  000000              RETURN:  .WORD  0               ;CONTAINS RETURN PC FOR SCOPE
420  002132  005726              SCOPEB:  TST    (6)+            ;POP PC
421  002134  012637      177776           MOV    (6)+,PSW        ;RESTORE CONDITION CODES
422  002140  000177      177764           JMP    @RETURN
423
```

```
425
426
427                                      ;THIS ROUTINE CONVERTS AN OCTAL NUMBER TO ASCII AND TYPES IT ON THE TTY.
428
429  002144  000000              D2BTYP:  0
430  002146  013746      177564  O2A:     MOV    TPCBR,-(6)      ;SAVE TPCBR
431  002152  010246                       MOV    X2,-(6)         ;SAVE R2
432  002154  010146                       MOV    X1,-(6)         ;SAVE R1
433  002156  010046                       MOV    X0,-(6)         ;SAVE R0
434  002160  013700      002144           MOV    D2BTYP,X0       ;GET DATA TO BE TYPED
435  002164  012701      000006           MOV    #6,X1           ;GET COUNTER
436  002170  005002                       CLR    X2              ;CLEAR WORKING REGISTER
437  002172  006100                       ROL    X0              ;MOV FIRST BIT (MSB) INTO
438  002174  006102                       ROL    X2              ;R2
439  002176  062702      000260  O2AA:    ADD    #260,X2         ;FORM ASCII CODE
440  002202  105737      177564           TSTB   TPCBR           ;TEST TELEPRINTER
441  002206  100375                       BPL    .-4             ;FLAG AND WAIT UNTIL DONE
442  002210  010237      177566           MOV    X2,TPOBR        ;LOAD TELEPRINTER BUFFER
443  002214  005002                       CLR    X2              ;CLEAR WORKING REGISTER
444  002216  006100                       ROL    X0              ;ROTATE THE
445  002220  006102                       ROL    X2              ;NEXT
446  002222  006100                       ROL    X0              ;OCTAL CHARACTER
447  002224  006102                       ROL    X2              ;INTO
448  002226  006100                       ROL    X0              ;REGISTER
449  002230  006102                       ROL    X2              ;TWO
450  002232  005301                       DEC    X1              ;DECREMENT COUNTER
451  002234  001360                       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)+,TPCSR      ;AND TPCSR
456  002250  000207                       RTS    7               ;AND EXIT
457
458                                      ;ASCII MESSAGES
459  002252  022500  052107  032055  M7T:  .ASCII  'MST=40 ROM DATAXXB'
     002260  020060  047522  020115
     002266  040504  040524  022445
     002274  100
460  002275  100     022445  100     M8T:  .ASCII  'MSS0'
461  002301  100     020040  100     M9T:  .ASCII  '0 0'
462  002306                                .EVEN
463
464  002306  032737  000400  177570  SWITCH:  BIT    #400,SR         ;TEST BIT 8
465  002314  001007                       BNE    1$              ;BR IF VERSION 1
466  002316  012737  001000  001002       MOV    #500,WORDS      ;SET UP VERSION 2 LENGTH
467  002324  012737  006000  001004       MOV    #START,IMAGE    ;SET UP VERSION 2 STARTING ADD.
468  002332  000406                       BR     2$
469  002334  012737  000400  001002  1$:  MOV    #290,WORDS      ;SET UP VERSION 1 LENGTH
470  002342  012737  016000  001004       MOV    #STARTA,IMAGE   ;SET UP VERSION 1 STARTING ADD.
471  002350  000207              2$:      RTS    PC
472
473                                      .SBTTL  ROM VERSION 2 VALUES
```

```
475                            ;*************************************************
476                            ;     EXCEPT FOR THE NEW ORGIN ADDRESS AND SEVERAL "1160000"
477                            ;     FOR ADDRESS FUDGING THIS IS AN EXACT COPY OF THE CONTENTS
478                            ;     OF THE GT-40 BOOTSTRAP VERSION #2
479                            ;*************************************************
480
481
482                                 .TITLE  SCROLLING ROM BOOTSTRAP FOR THE GT40
483
484
485
486
487
488
489
490            ;                         BOOTGT.T16     OCT 19, 1973
491
492
493
494
495            ;               COPYRIGHT 1973, DIGITAL EQUIPMENT CORPORATION
496            ;               146 MAIN STREET
497            ;               MAYNARD, MASSACHUSETTS
498            ;                                   01754
499
500
501
502
503            ;               WRITTEN BY JACK BURNESS.
504
505
506
507
508
509            ;               THIS PROGRAM IS THE SECOND VERSION THE THE ROM BOOTSTRAP FOR
510            ;         THE GT40 DISPLAY TERMINAL. IT INCLUDES SCROLLING AND AN END OF
511            ;         MEMORY SEARCH FOR THE LOADER.
512
513
514
515
516                 .ENABL  ABS,AMA                 ;ASSEMBLER DIRECTIVES FOR ABSOLUTE BINARY OUTPUT
517                                                 ; NOTE! USE "MACDLX" TO ASSEMBLE THIS PROGRAM.
518
519
520
521
522
523
524
525
526
527                 .SBTTL  DEFINITION SECTION
528                 .PAGE
```

```
529
530
531            ;                         REGISTER DEFINITIONS
532            ;                         ------------------
533
534
535
536            ;                         BASIC DEFINTIONS
537            ;                         ----------------
538
539
540       000000         R0=%0                      ;DEFINE STANDARD VALUES.
541       000001         R1=%1
542       000002         R2=%2
543       000003         R3=%3
544       000004         R4=%4
545       000005         R5=%5
546       000006         SP=%6
547       000007         PC=%7
548
549
550
551            ;                         GT40 DEFINTIONS
552            ;                         ---------------
553
554
555       000000         CHAR=R0                     ;CONTAINS THE INPUT CHARACTER.
556       000001         POINTR=R1                   ;POINTS TO NEXT INSERTION BYTE IN DISPLAY BUFFER
557       000002         TABCNT=R2                   ;CHARACTER COUNTER FOR THE "TAB" FEATURE.
558       000003         SCAN=R3                     ;GENERALLY CONTAINS A POINTER WHICH
559                                                  ;IS USED WHEN SCANKING MEMORY FOR SOMETHING.
560       000004         HOLD=R4                     ;TYPICALLY A TEMPORARY WHICH IS USED TO RETAIN
561                                                  ;A VALUE FOR A SHORT TIME.
562       000005         COUNTR=R5                   ;TYPICALLY USED AS A COUNTER.
563
564
565
566
567
568            ;                         LOADER DEFINITIONS
569            ;                         ------------------
570
571
572
573
574
575       000000         L.BYT=CHAR                  ;CHARACTER INPUT FOR THE LOADER.
576       000001         L.ADR=POINTR                ;CURRENT MEMORY ADDRESS TO BE LOADED.
577       000002         L.BC=TABCNT                 ;NUMBER OF DATA ITEMS TO LOAD.
578       000005         L.CKSM=COUNTR               ;CHECKSUM ON THE INPUT DATA.
579       000003         INDEX=SCAN                  ;INDICATES HOW TO ASSEMBLE THE 8 BIT CHARACTER.
580
581
582                 .PAGE
```

```
583
584
585                                        ;                        MAJOR SYSTEM DEFINITIONS
586                                        ;                        ----- ------ ----------
587
588
589
590          166000                        ORIGIN=166000                   ;ORIGIN OF THE BOOTSTRAP.
591
592          175610                        DL11IS=175610                   ;INPUT STATUS REGISTER OF DL11
593          175612                        DL11IB=DL11IS+2                  ;INPUT CHARACTER FROM DL11
594          175614                        DL11OS=DL11IB+2                  ;OUTPUT STATUS OF THE DL11
595          175616                        DL11OB=DL11OS+2                  ;OUTPUT CHARACTER TO THE DL11
596
597          177560                        KBDIS=177560                    ;KEYBOARD INPUT STATUS
598          177562                        KBDIB=KBDIS+2                    ;CURRENT CHARACTER FROM KEYBOARD.
599
600          172000                        GT40PC=172000                   ;GT40 PROGRAM COUNTER.
601          172002                        GT40SR=GT40PC+2                  ;GT40 STATUS REGISTER ADDRESS.
602
603
604          001000                        BSTART=1000                     ;START OF THE DISPLAY BUFFER.
605          007000                        BLIMIT=7000                     ;APPROXIMATE END OF THE DISPLAY BUFFER.
606          007776                        TMPEND=7776                     ;LOCATION OF INITIALIZATION STACK.
607          000004                        CORSTR=4                        ;LOCATION OF PDP-11 TRAP VECTOR.
608          007012                        JMPADD=BLIMIT+10                 ;WHERE THE POINTER IS TO FIRST CHAR ON SCREEN
609          000040                        NUMLIN=32.                       ;NUMBER OF LINES ON TEXT TO SHOW ON THE SCREEN
610
611          005015                        CRLF=5015                       ;CARRIAGE RETURN = LINE FEED
612          000175                        ALTMOD=175                       ;THE "KEY" CHARACTER (I.E. ALTMODE).
613
614          160000                        DISJMP=160000                   ;THE GT40 JMP INSTRUCTION
615          173000                        DISTOP=173000                   ;THE GT40 STOP DISPLAY INSTRUCTION.
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634                                        .SBTTL  INITIALIZATION AND RESTART CODE
635                                        .PAGE
```

```
636
637
638
639                                        ;                        GT40 BOOTSTRAP CODE
640                                        ;                        ---- --------- ----
641
642
643
644
645          006000                        .=6000
646
647                                        ;        .=ORIGIN                ;DEFINE ORIGIN OF THE BOOTSTRAP.
648
649
650
651
652
653
654                                        ;        COLD INITIALIZATION CODE
655                                        ;        ---- --------------- ----
656
657
658
659  006000  000005                        START:  RESET                   ;RESET ALL HARDWARE NOW.
660  006002  012737  000007  175610                MOV     #7,DL11IS       ;INITIALIZE DL=11 INPUT NOW.
661  006010  012706  007776                        MOV     #TMPEND,SP      ;ESTABLISH A GOOD TEMPORARY STACK
662                                                                        ;POINTER FOR CORE SEARCH.
663  006014  005237  175614                        INC     DL11OS          ;SET BREAK BIT
664  006020  004737  166652                        JSR     SCANOUT,LIT;160000 ;FOR 2 CHARACTER TIMES
665  006024  000000                                .WORD   0               ;SEND TWO ZERO'S
666
667  006026  012703  000004                        MOV     #CORSTR,SCAN    ;GET ADDRESS OF BAD CORE TRAP VECTOR.
668  006032  012723  166042                        MOV     #NOTHER!160000,(SCAN)+ ;AND INSERT A POINTER TO US THERE.
669
670  006036  005023                        ENDCOR: CLR     (SCAN)+         ;NOW CLEAR ALL OF MEMORY BEYOND THE POINTER,
671  006040  000776                                BR      ENDCOR          ;UNTIL WE RUN OUT OF MEMORY AND TRAP.
672
673
674  006042  005743                        NOTHER: TST     -(SCAN)         ;WHEN WE TRAP OUT, WE COME HERE.
675                                                                        ;WE BACK UP POINTER TO GOOD CORE.
676                                                                        ;NOTE THAT IF WE TRAP OUT AGAIN, IT
677                                                                        ;IS STILL OK, BECAUSE WE WILL LOOP
678                                                                        ;UNTIL WE GET A GOOD CORE ADDRESS.
679  006044  010306                                MOV     SCAN,SP         ;WHEN WE GET ONE, THAT IS LAST LOCATION
680                                                                        ;IN THE MACHINE, AND HENCE OUR SP.
681  006046  105737  175614                1S:     TSTB    DL11OS          ;SEE IF BREAK IS DONE
682  006052  100375                                BPL     1S              ;NO GO BACK
683  006054  005037  175614                        CLR     DL11OS          ;CLEAR BREAK BIT
684
685
686
687
688
689
```

```
690                                          ;        RESTART INITIALIZATION CODE WHEN COMMUNICATIONS IS WORKING.
691                                          ;        -------------------------------------------------
692
693
694
695   026060  052706  007776     RESTRT: BIS      #TMPEND,SP               ;FORCE THE SP TO LIMIT OF EXISTING CORE.
696
697
698   026064  012703  006700             MOV      #BLIMIT-NUMLIN-NUMLIN,SCAN    ;NOW WE WILL FILL THE KEY AREAS OF THE
699   026070  012702  000040             MOV      #NUMLIN,TABCNT            ;DISPLAY BUFFER WITH INITIAL CR-LP'S.
700
701   026074  012723  005015     SETLP1: MOV      #CRLF,(SCAN)+             ;INSERT A CRLF NOW.
702   026100  005302                     DEC      TABCNT                   ;AND LOOP UNTIL DONE.
703   026102  003374                     BGT      SETLP1                   ;THUS DISPLAY CORE IS ALMOST CORRECT.
704
705
706   026104  012703  166432             MOV      #SETUP!160000,SCAN        ;NOW WE WILL INITALIZE CORE FOR THE
707                                                                         ;DISPLAY, PICK UP POINTER TO LIST.
708
709   026110  012302     SETLP2: MOV      (SCAN)+,TABCNT            ;GET NUMBER OF ITEMS TO INSERT.
710   026112  001405                     BEQ      SETDUN                   ;IF ZERO, WE ARE DONE.
711   026114  012301                     MOV      (SCAN)+,POINTR           ;PICK UP FIRST CORE ADDRESS POINTER.
712
713   026116  012321     SETLP3: MOV      (SCAN)+,(POINTR)+        ;MOVE OVER A DATA ITEM NOW.
714   026120  005302                     DEC      TABCNT                   ;ALL DONE?
715   026122  003375                     BGT      SETLP3                   ;NOPE, MOVE OVER THE NEXT.
716   026124  000771                     BR       SETLP2                   ;YES: GET NEXT MATCH LIST TO INSERT.
717
718
719   026126  012701  006776     SETDUN: MOV      #BLIMIT-2,POINTR         ;ESTABLISH THE BUFFER POINTER NOW.
720
721
722
723
724
725
726
727
728
729
730
731                                          .SBTTL   VT05 SIMULATOR
732                                          .PAGE
```

```
733
734
735
736                                          ;        VT05 (SCROLLING) PORTION OF THE BOOTSTRAP
737                                          ;        ------------------------------------------
738
739
740
741   026132  004737  166564     NXTCHR: JSR      PC,GETCHR!160000         ;GET A CHARACTER NOW.
742   026136  020027  000177             CMP      CHAR,#177                ;IS IT OUT OF RANGE?
743   026142  002373                     BGE      NXTCHR                   ;YEP, GET ANOTHER ONE.
744   026144  020027  000040             CMP      CHAR,#040                ;IS IT A PRINTING CHARACTER?
745   026150  002020                     BGE      NORMAL                   ;YES, IT'S A NORMAL PRINTING CHARACTER.
746   026152  010003                     MOV      CHAR,SCAN                ;MOVE IT OVER SO WE CAN PLAY WITH IT.
747   026154  162703  000007             SUB      #7,SCAN                  ;BIAS SO THAT BELL [7] IS ZERO.
748   026160  020327  000007             CMP      SCAN,#7                  ;IF CHARACTER IS LESS THEN BELL OR
749   026164  103362                     BHIS     NXTCHR                   ;GREATER THEN CR, THEN IGNORE.
750   026166  006303                     ASL      SCAN                     ;IF GOOD, MAKE IT WORD INDEX.
751   026170  060307                     ADD      SCAN,PC                  ;AND GO TO THE CORRECT ROUTINE.
752
753   026172  000426                     BR       BELL                     ;7=BELL
754   026174  000406                     BR       NORMAL                   ;10=BACKSPACE
755   026176  000411                     BR       TAB                      ;11=TAB
756   026200  000437                     BR       LF                       ;12=LINE FEED [LF]
757   026202  000420                     BR       VT                       ;13=VERTICAL TAB [VT]
758   026204  000424                     BR       FF                       ;14=FORM FEED [FF]
759                                                                         ;15=CARRIAGE RETURN [CR]
760
761
762   026206  012702  177777     CR:     MOV      #-1,TABCNT               ;RESET TAB POSITION ON A CR, AND
763                                                                         ;FALL THROUGH TO INSERT THE CHARACTER.
764
765
766   026212  004737  166350     NORMAL: JSR      PC,INSERT!160000         ;INSERT THE CHARACTER IN THE BUFFER.
767   026216  005202                     INC      TABCNT                   ;UPDATE TAB POSITION NOW.
768   026220  000744                     BR       NXTCHR                   ;AND GET NEXT CHARACTER.
769
770
771
772
773   026222  012700  000040     TAB:    MOV      #40,CHAR                 ;ON A TAB, INSERT BLANKS UNTIL THE
774   026226  004737  166350             JSR      PC,INSERT!160000         ;       NEXT CHARACTER POSITION IS A MULTIPLE
775   026232  005202                     INC      TABCNT                   ;OF 8.
776   026234  032702  000007             BIT      #7,TABCNT                ;ARE WE DONE YET?
777   026240  001370                     BNE      TAB                      ;NOPE.
778   026242  000733                     BR       NXTCHR                   ;YES.
779
780
781   026244  111705                     VT:     MOVB     (PC),COUNTR      ;THIS PUTS THE LOW BYTE OF THE
782                                                                         ;BRANCH CODE IN COUNTR-SAVE A WORD
783   026246  000405                     BR       FFLOOP
784
785   026250  005037  172002     BELL:   CLR      GT40SR                   ;RING BELL -WRITE IN GT40SR
786   026254  000726                     BR       NXTCHR                   ;AND LOOP BACK
```

```
787
788  004256  012705  000040     FF1      MOV    #NUMLIN,COUNTR          ;FORM FEED IS DONE BY INSERTING LF'S.
789
790  004262  012700  000012     FFLOOP1  MOV    #12,CHAR               ;MAKE THE CHARACTER A LINEFEED.
791  004266  004737  166304              JSR    PC,LFSUB!160000        ;DO A LINEFEED.
792  004272  005305                      DEC    COUNTR                 ;DONE?
793  004274  003372                      BGT    FFLOOP                 ;NOPE, KEEP SENDING THEM.
794  004276  000715                      BR     NXTCHR                 ;YES, NOW RETURN. (DO NOT FALL THROUGH).
795
796                                                                    ;RETURN TO NXTCHR AFTER PROCESSING
797  004300  012746  166132     LF1      MOV    #NXTCHR!160000,-(SP)   ;THE LF BY FAKING A JSR.
798
799
800  004304  013703  007012     LFSUB1   MOV    JMPADD,SCAN           ;GET POINTER TO FIRST CHAR ON SCREEN
801
802  004310  122300             LFLOOP1  CMPB   (SCAN)+,CHAR          ;AND LOOK FOR A LINEFEED.
803  004312  001406                      BEQ    LFOUND                ;GOT IT, SEARCH HAS ENDED.
804  004314  020327  007000              CMP    SCAN,#BLIMIT          ;ARE WE AT END OF BUFFER?
805  004320  103773                      BLO    LFLOOP                ;NOPE, KEEP ON LOOKING.
806  004322  012703  001000              MOV    #BSTART,SCAN          ;IF AT TOP, RESET TO BOTTOM OF BUFFER
807  004326  000770                      BR     LFLOOP                ;AND KEEP ON LOOKING.
808
809  004330  005203             LFOUND1  INC    SCAN                 ;WE'VE GOT THE LINE FEED, STOP SHOWING
810  004332  042703  000001              BIC    #1,SCAN              ;FIRST LINE BY CHANGING THE "DISJMP"
811  004336  010337  007012              MOV    SCAN,JMPADD          ;INSTRUCTION TO FIRST CHAR BEYOND LF.
812  004342  004737  166350              JSR    PC,INSERT!160000     ;INSERT THE LF IN THE BUFFER,
813  004346  005000                      CLR    CHAR                 ;AND THEN INSERT ONE NULL CHARACTER BECAUSE
814                                                                  ;THE "DISJMP" ADDRESS MUST BE EVEN, AND
815                                                                  ;THIS GUARANTEES WE WILL NOT LOSE A
816                                                                  ;A GOOD DATA CHARACTER. WE FALL THROUGH
817                                                                  ;TO INSERT THE NULL IN THE BUFFER.
818
819  004350  110021             INSERT1  MOVB   CHAR,(POINTR)+       ;STICK IN THE CHARACTER NOW.
820  004352  032701  000001              BIT    #1,POINTR            ;IS NEXT POSITION EVEN OR ODD?
821  004356  001021                      BNE    INSRTX               ;ODD, NO PROBLEMS, SPACE IS ALLOCATED.
822  004360  020127  007000              CMP    POINTR,#BLIMIT       ;EVEN, ARE WE AT THE END OF THE BUFFER?
823  004364  103410                      BLO    INSRTL               ;NO, JUST MAKE ROOM FOR ANOTHER WORD.
824  004366  010103                      MOV    POINTR,SCAN          ;AT THE END, MOVE THE STUFF TO THE
825  004370  012701  001000              MOV    #BSTART,POINTR       ;BEGINNING OF THE BUFFER.
826  004374  004737  166406              JSR    PC,INSRTL!160000     ;CALL THE ROUTINE TO SAVE SPACE,
827  004400  005023                      CLR    (SCAN)+              ;AND CLEAR UP THE INSTRUCTIONS AT THE
828  004402  005013                      CLR    (SCAN)               ;END OF THE BUFFER,
829  004404  000207                      RTS    PC                   ;AND THEN RETURN.
830
831  004406  022121             INSRTL1  CMP    (POINTR)+,(POINTR)+  ;BYPASS THE "DISJMP" BY ADDING 4 TO POINTR,
832  004410  012711  166474              MOV    #HEADER!160000,(POINTR) ;NOW INSERT THE DISJMP INSTRUCTION TO OUR HEADER
833  004414  012741  160000              MOV    #DISJMP!-(POINTR)    ;AND IT'S ADDRESS (PUT THEM IN BACKWARDS).
834  004420  005041                      CLR    -(POINTR)            ;MAKE AVAILABLE A NEW CHARACTER SPOT.
835
836  004422  000207             INSRTX1  RTS    PC                   ;FINALLY RETURN TO THE CALLER.
837
838
839
840
```

```
841
842
843  004424  012737  001000  172000  GTBUSE1  MOV  #BSTART,GT40PC    ;ON A BUS ERROR, WE MERELY RESTART THE GT40 AT
844                                                                  ;THE RTI FOR THIS ROUTINE
845                                                                  ;IS THE FIRST WORD OF THE TABLE
846                                                                  ;BELOW. IT SAVES A WORD!
847
848
849
850
851
852
853
854
855
856
857
858                                              ;           INITIALISATION TABLE FOR THE SCROLLER
859                                              ;           ------------------------------------
860                                              ;
861
862
863
864  004432  000002             SETUP1   .WORD  2                    ;INITIALIZE 2 WORDS.--ALSO RTI FROM ABOVE
865  004434  000330                      .WORD  330                  ;STARTING AT LOCATION 330
866  004436  166424                      .WORD  GTBUSE!160000        ;FIRST WORD IS POINTER TO BUS ERROR ROUT
867  004440  000200                      .WORD  200                  ;SECOND WORD IS NEW STATUS WORD ON INTERUPT.
868
869  004442  000007                      .WORD  7                    ;INITIALIZE THE END OF THE BUFFER TO
870  004444  006776                      .WORD  BLIMIT-2             ;A CLEAR SPACE TO INSERT THE CHARACTER.
871  004446  000000                      .WORD  0                    ;THIS IS THE "RUNNING" START, THIS IS
872  004450  160000  166474              .WORD  DISJMP,HEADER!160000 ;FOLLOWED BY A DISJMP TO OUR HEADER BLOC
873  004454  160000  001000              .WORD  DISJMP,BSTART        ;AND THEN A DISJMP TO THE START OF THE BUFFER
874  004460  160000  006700              .WORD  DISJMP,BLIMIT-NUMLIN-NUMLIN ;AND A DISJMP TO THE FIRST CHAR ON SCREE
875
876  004464  000001                      .WORD  1                    ;FINALLY START THE GT40 GOING AT
877  004466  172000                      .WORD  GT40PC               ;THE POSITION INSTRUCTION IN THE
878  004470  166474                      .WORD  HEADER!160000        ;HEADER BLOCK.
879
880  004472  000000                      .WORD  0                    ;END OF INIT CODE
881                                              ;
882                                              ;       HEADER BLOCK FOR THE SCROLLER
883
884  004474  103334             HEADER1  .WORD  103334               ;ENABL CHAR MODE,BLINKING
885  004476  000177                      .WORD  177                  ;A BLINKING BOX-RUB OUT!
886  004500  116124                      .WORD  116124               ;GO TO POINT MODE
887  004502  171340                      .WORD  171340               ;LOAD STATUS REGISTER
888  004504  000000  001352              .WORD  0,1352               ;POINT TO UPPER LEFT
889  004510  103324                      .WORD  103324               ;BACK TO CHAR MODE
890  004512  160000  007010              .WORD  DISJMP,JMPADD-2      ;AND TO THE CHANGING JMP INST.
891
892                                              .SBTTL  COMMUNICATIONS AND MISC. SUPPORT ROUTINES
893                                              .PAGE
```

```
894
895
896
897                                    ;                        COMMUNICATIONS HANDLING ROUTINES
898                                    ;                        -------------- -------- --------
899
900
901
902
903
904
905                                    ;                        THE DL-11 HANDLER
906                                    ;                        --- ----- -------
907
908
909
910  000516  105737  175610    GETDL:  TSTB    DL11IS          ;CHECK THE HOST INPUT STATUS.
911  000522  100011            BPL     GETDL1          ;HOST DID NOT SEND ANYTHING, YET.
912  000524  113700  175612            MOVB    DL11IB,CHAR     ;HOST SENT US A CHARACTER, PROCESS IT.
913  000530  012737  000007  175610    MOV     #7,DL11IS       ;REENABLE THE HOST TELECOMMUNICATIONS.
914  000536  042700  177600            BIC     #-200,CHAR      ;MAKE CHARACTER JUST SEVEN BITS.
915  000542  001765            BEQ     GETDL           ;IF NULL, IGNORE IT.
916  000544  000207            RTS     PC              ;ELSE RETURN NOW.
917
918  000546  105737  177560    GETDL1: TSTB    KBDIS           ;DID USER TYPE A CHARACTER?
919  000552  100361            BPL     GETDL           ;NO, GO BACK AND CHECK HOST MACHINE.
920  000554  113737  177562  175616    MOVB    KBDIB,DL11OB    ;MOVE THE CHARACTER TO THE HOST.
921  000562  000755            BR      GETDL           ;AND CHECK AGAIN FOR INPUT.
922
923
924
925
926                                    ;                        THE "GET CHARACTER" ROUTINE
927                                    ;                        --- --- --------- -------
928
929
930
931
932  000564  004737  166516    GETCHR: JSR     PC,GETDL!160000 ;GET A CHARACTER FROM THE HOST NOW.
933  000570  020027  000175            CMP     CHAR,#ALTMOD    ;IS IT AN "ALTMODE"
934  000574  001025            BNE     GETEXT          ;NO, EXIT NOW.
935
936  000576  004737  166516            JSR     PC,GETDL!160000 ;YES, GET ANOTHER ONE NOW.
937  000602  020027  000114            CMP     CHAR,#'L        ;IS IT AN "L"
938  000606  001501            BEQ     LOADER          ;YES, START LOADING NOW.
939  000610  020027  000122            CMP     CHAR,#'R        ;IS IT AN "R"
940  000614  001015            BNE     GETEXT          ;NO, IGNORE THE ALTMODE AND JUST RETURN THE CHAR
941
942  000616  012737  173000  007010    MOV     #DISTOP,JMPADD-2 ;YES, RESET, STOP DISPLAY BY INSERTING A "DISTOP
943  000624  000137  166060    PRESTR: JMP     RESTRT!160000    ;INSTRUCTION IN THE BUFFER, AND RESTART.
944
945
946
947
```

```
948
949                                    ;                        THE "GET A SIX BIT CHARACTER" ROUTINE
950                                    ;                        --- --- - --- --- ------------- -------
951
952
953
954  000630  004737  166564    GETSIX: JSR     PC,GETCHR!160000    ;GET A CHARACTER NOW.
955  000634  020027  000040            CMP     CHAR,#40        ;IS IT A LEGAL PRINTING CHARACTER?
956  000640  002517            BLT     L.BAD           ;NOPE, ABORT
957  000642  020027  000137            CMP     CHAR,#137       ;IT'S BIG ENOUGH, IS IT TOO BIG?
958  000646  003114            BGT     L.BAD           ;YEP, ABORT.
959
960  000650  000207            GETEXT: RTS     PC              ;RETURN TO THE CALLER.
961
962
963                                    ;                        THIS OUTPUTS TWO CHARACTERS VIA A
964                                    ;                        JSR SCAN,OUTLIT
965                                    ;                        'TWO CHARACTERS'
966
967  000652  112337  175616    OUTLIT: MOVB    (SCAN)+,DL11OB
968  000656  112337  175616            MOVB    (SCAN)+,DL11OB  ;DOUBLE BUFFERED
969  000662  000203            RTS     SCAN            ;RETURN
970
971
972
973
974
975
976
977                                    ;                        THE "GET AN EIGHT BIT CHARACTER" ROUTINE
978                                    ;                        --- --- -- ----- --- ------------- -------
979
980
981
982                                    ;                        THIS ROUTINE DIFFERS FROM THE PREVIOUS ROUTINES
983                                    ;                        IN THAT IT WILL TAKE SIX BIT CHARACTERS AND ASSEMBLE
984                                    ;                        THEM FOR THE LOADER TO USE. NOTE THAT FROM THIS POINT
985                                    ;                        ON WE WILL SWITCH TO THE LOADER DEFINITIONS OF THE
986                                    ;                        REGISTERS. THUS THE CHARACTER IS RETURNED IN
987                                    ;                        REGISTER "L.BYT" RATHER THAN CHAR (THOUGH THEY ARE
988                                    ;                        PHYSICALLY THE SAME).
989
990
991
992  000664  004737  166630    GET8:   JSR     PC,GETSIX!160000    ;GET A SIXBIT CHARACTER.
993  000670  010046            MOV     L.BYT,-(SP)     ;SAVE IT ON THE STACK.
994  000672  005723            TST     (INDEX)+        ;UPDATE INDEX TO NEXT ITEM (ALL ARE +2)
995  000674  000163  166676            JMP     GET8TB-2!160000(INDEX)  ;AND DISPATCH ACCORDING TO THE INDEX.
996
997  000700  000404    GET8TB: BR      GET81           ;INDEX=2: ASSEMBLE FIRST CHAR
998  000702  000416            BR      GET82           ;INDEX=4: ASSEMBLE SECOND CHAR
999  000704  000432            BR      GET83           ;INDEX=6: ASSEMBLE THIRD AND LAST CHAR
1000                                                  ;INDEX=8: RESET INDEX TO 0 (2) AND RETRY.
1001
```

```
1022
1023  006716  012703  000002       GET84:  MOV    #2,INDEX              ;THE FOURTH INDEX IS THE SAME AS THE FIRST
1024                                                                     ;INDEX, JUST RESET IT AND FALL THROUGH,
1025
1026
1027  006712  004737  166630       GET81:  JSR    PC,GETSIX;160000           ;GET ANOTHER CHARACTER NOW,
1028  006716  010004                        MOV    L,BYT,HOLD           ;AND PRESERVE IT FOR NEXT TIME THROUGH,
1029  006720  006300                        ASL    L,BYT                ;NOW THROW AWAY LEFT MOST BITS OF
1030  006722  006300                        ASL    L,BYT                ;THE 8 BIT CHARACTER, NOW MERGE IN
1031  006724  106300                        ASLB   L,BYT                ;THE LEFT TWO BITS OF THE
1032  006726  106116                        ROLB   (SP)                 ;NEW SIX BIT CHARACTER WITH THE SIX
1033  006730  106300                        ASLB   L,BYT                ;BITS FROM THE CHARACTER ON THE
1034  006732  106116                        ROLB   (SP)                 ;STACK, 1ST CHARACTER IS NOW ASSEMBLED,
1015  006734  012600                        MOV    (SP)+,L,BYT          ;SO WEILL RETURN IT TO THE USER,
1016  006736  000207                        RTS    PC                   ;AND THEN WE SHALL RETURN TO HIM,
1017
1018
1019  006740  006300       GET82:  ASL    L,BYT                         ;THE SECOND CHARACTER IS CREATED FROM
1020  006742  006300                ASL    L,BYT                         ;THE 4 RIGHT BITS OF THE PREVIOUS CHARACTER
1021  006744  106300                ASLB   L,BYT                         ;AND THE FOUR MIDDLE BITS OF THE PRESENT
1022  006746  106104                ROLB   HOLD                          ;8 BIT CHARACTER,
1023  006750  106300                ASLB   L,BYT                         ;WE WILL CREATE THE NEW 8 BIT
1024  006752  106104                ROLB   HOLD                          ;IN THIS REGISTER, SINCE IT
1025  006754  106300                ASLB   L,BYT                         ;MORE CONVIENT, WE WILL MOVE OVER THE
1026  006756  106104                ROLB   HOLD                          ;ANSWER AT THE END,
1027  006760  106300                ASLB   L,BYT                         ;ONE MORE TO GO
1028  006762  106104                ROLB   HOLD                          ;DONE,
1029  006764  010400                MOV    HOLD,L,BYT                    ;BRING OVER THE VALUE,
1030  006766  012604                MOV    (SP)+,HOLD                    ;AND REMEMBER THE LAST CHARACTER WE RECEIVED,
1031  006770  000207                RTS    PC                            ;AND RETURN TO THE CALLER,
1032
1033
1034  006772  006100       GET83:  ROL    L,BYT                         ;FINAL CHARACTER IS EASY, JUST A
1035  006774  106100                ROLB   L,BYT                         ;SIMPLE MERGER OF LEFT TWO BITS OF
1036  006776  006004                ROR    HOLD                          ;PREVIOUS VALUE WITH RIGHT SIX BITS
1037  007000  106000                RORB   L,BYT                         ;OF LAST (4TH) CHARACTER RECEIVED,
1038  007002  006004                ROR    HOLD
1039  007004  106000                RORB   L,BYT                         ;AND WE ARE DONE,
1040  007006  005726                TST    (SP)+                         ;FINALLY THROW AWAY STACK,
1041  007010  000207                RTS    PC                            ;AND RETURN TO THE CALLER,
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054                                .SBTTL   THE LOADER
1055                                .PAGE
```

```
1056
1057
1058
1059                                !                  THE LOADER
1060                                ;                  === ======
1061
1062
1063
1064
1065  007012  012737  173000  007010  LOADER:  MOV   #DISTOP;JMPADD=2     ;STOP THE GT40 BY INSERTING A "DISTOP" IN THE LI
1066
1067  007020  005003                           CLR   INDEX                ;RESET THE 8 BIT ASSEMBLER TO THE FIRST CHAR
1068
1069
1070  007022  005005               L.LD2:  CLR    L,CKSM                  ;CLEAR THE CHECKSUM
1071  007024  004737  167114              JSR    PC,L,PTR;160000          ;GET A BYTE NOW,
1072  007030  105300                      DECB   L,BYT                   ;IS IT A ONE (HEADER)?
1073  007032  001373                      BNE    L,LD2                   ;NO: WAIT FOR THE ONE,
1074
1075  007034  004737  167114              JSR    PC,L,PTR;160000          ;YES: SKIP OVER THE NEXT CHARACTER NOW,
1076
1077  007040  004737  167126              JSR    PC,L,GWRD;160000              ;ASSEMBLE A WORD NOW,
1078  007044  010002                      MOV    L,BYT,L,BC              ;MOVE OVER TO THE COUNTER,
1079  007046  162702  000004              SUB    #4,L,BC                 ;REDUCE TO ACTUAL DATA COUNT,
1080  007052  022702  000002              CMP    #2,L,BC                 ;ANY DATA AT ALL?
1081  007056  001433                      BEQ    L,JMP                   ;NO: MUST BE END
1082  007060  004737  167126              JSR    PC,L,GWRD;160000              ;YES: ASSEMBLE A DATA WORD NOW,
1083  007064  010001                      MOV    L,BYT,L,ADR             ;AND THIS MUST BE THE FIRST ADDRESS,
1084
1085
1086  007066  004737  167114       L.LD3:  JSR    PC,L,PTR;160000         ;GET A BYTE OF DATA NOW,
1087  007072  002006                       BGE    L,LD4                   ;ALL DONE?
1088  007074  105705                       TSTB   L,CKSM                  ;YEP, COUNTER IS MINUS, CHECK CHECKSUM,
1089  007076  001751                       BEQ    L,LD2                   ;CHECKSUM GOOD, GET NEXT COMMAND,
1090
1091  007100  004337  166652       L.BAD:  JSR    SCAN;OUTLIT;160000      ;BAD LOAD INFORM HOST
1092  007104     175     102               .BYTE  ALTMOD;!8               ;SEND ALTMODE B
1094  007106  000646                       BR     PRESTRT                 ;AND RESTART THE DISPLAY,
1095
1096
1097  007110  110021       L.LD4:  MOVB   L,BYT,(L,ADR)+                  ;INSERT BYTE INTO MEMORY,
1098  007112  000765               BR     L,LD3                           ;AND GET THE NEXT BYTE,
1099
1100
1101
1102  007114  004737  166664       L.PTR:  JSR    PC,GET8;160000               ;ASSEMBLE AN 8 BIT CHARACTER NOW,
1103  007120  060005                       ADD    L,BYT,L,CKSM            ;UPDATE THE CHECKSUM NOW,
1104  007122  005302                       DEC    L,BC                    ;DECREMENT THE CHARACTER COUNTER,
1105  007124  000207                       RTS    PC                      ;AND RETURN TO THE CALLER NOW,
1106
1107
1108
1109  007126  004737  167114       L.GWRD:  JSR    PC,L,PTR;160000        ;ASSEMBLE A WORD, FIRST GET A CHARACTER
```

```
1110  007132  010046                  MOV    L,BYT,-(SP)        ;AND SAVE IT.
1111  007134  004737  167114          JSR    PC,L,PTR;160000    ;AND THEN GET ANOTHER ONE.
1112  007140  000300                  SWAB   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.JMP:  JSR    PC,L,GNRD;160000          ;ALL DONE WITH THE LOAD, ASSEMBLE
1120  007152  010046                  MOV    L,BYT,-(SP)        ;THE STARTING ADDRESS NOW.
1121  007154  004737  167114          JSR    PC,L,PTR;160000    ;AND DON'T FORGET TO CHECKSUM IT,
1122  007160  105705                  TSTB   L,CKSM
1123  007162  001346                  BNE    L,BAD              ;A BAD CHECKSUM, ALL IS EVIL.
1124
1125  007164  004337  166652          JSR    SCAN#OUT,LIT;160000       ;GOOD CHKSUM,INFORM HOST
1126  007170     175     107          .BYTE  ALTMOD,:G          ;WITH ALTMOD G
1127
1128  007172  032716  000001          BIT    #1,(SP)            ;DO WE WANT TO START EXECUTION?
1129  007176  001401                  BEQ    L,JMP1             ;YES, AWAY WE GO.
1130
1131  007200  000000          L.HALT: HALT                      ;IF NOT, HALT.
1132
1133  007202  000136          L.JMP1: JMP    @(SP)+             ;IF GO, THEN GO ALREADY, WHEEEE!
1134
1135
1136
1137                                  .SBTTL THE SELF TEST
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150                                  .PAGE
```

```
1151                                          ;THIS IS GT40 QUICK TEST
1152                                          ;GIVES QUICK VISUAL TEST
1153                                          ;OF CONDITION OF MACHINE
1154                                          ;WITHOUT READING IN DIAG.
1155  100000                  CHAR=100000
1156  104000                  SHORTV=104000
1157  110000                  LONGV=110000
1158  114000                  POINT=114000
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  000020                  BLKOFF=20
1175  000030                  BLKON=30
1176
1177  000004                  LINE0=4
1178  000005                  LINE1=5
1179  000006                  LINE2=6
1180  000007                  LINE3=7
1181
1182  160000                  DJMP=160000
1183  164000                  DNOP=164000
1184  170000                  STATSA=170000
1185  173400                  DSTOP=173400            ;STOP INTERRUPT
1186
1187  000300                  LPLITE=300
1188  000200                  LPDARK=200
1189  000040                  ITAL0=40                ;ITALICS OFF
1190  000060                  ITAL1=60                ;"      ON
1191  000004                  SYNON=4                 ;SYNC ON
1192
1193
1194  174000                  STATSB=174000
1195
1196  000100                  INCR=100                ;LOAD GRAPH INCR
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=MINUSX
1202  001760                  MAXSX=1760              ;BIGGEST X IN SHORTVEC
1203  000077                  MAXSY=77                ;  "     Y IN  "
1204  000100                  MINSUY=100              ;MINUS BIT FOR Y IN SHORTVEC
```

```
1205
1206
1207   007204  012737  167214  172000        MOV     #FILE0!160000,GT40pC              ISTART THE GT40
1208   007212  000001                         WAIT                             IAND WAIT
1209
1210   007214  114020        FILE01  POINT!BLKOFF                  IPOINT=-INVISIBLE
1211   007216  000000                0
1212   007220  001377                MAXY
1213
1214   007222  112004                LONGV!INT0!LINE0              IDRAW TOP LINE
1215   007224  041777                INTX!MAXX
1216   007226  000000                0
1217
1218   007230  112405                LONGV!INT2!LINE1
1219   007232  040000                INTX                         IDRAW LINE TO RIGHT
1220   007234  021377                MINUSX!MAXY
1221
1222   007236  113006                LONGV!INT4!LINE2
1223   007240  061777                INTX,MINUSX,MAXX             IDRAW BOTTOM LINE
1224   007242  000000                0
1225
1226   007244  113407                LONGV!INT6!LINE3
1227   007246  040000                INTX
1228   007250  001377                MAXY                         IDRAW LINE TO LEFT
1229
1230   007252  114000                POINT
1231   007254  000400                400
1232   007256  000500                500
1233   007260  106200                SHORTV!INT1
1234   007262  057677                57677                        I+X+Y
1235   007264  106600                SHORTV!INT3
1236   007266  077677                77677                        I+X-Y
1237   007270  107200                SHORTV!INT5
1238   007272  077777                77777                        I-X-Y
1239   007274  107600                SHORTV!INT7
1240   007276  057777                57777                        I-X+Y
1241
1242   007300  114000                POINT
1243   007302  001400                1400
1244   007304  000500                500
1245   007306  133030                RELATV!INT4!BLKON
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                STATSB!INCR+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                FILE0!160000
1279
1280                                 .SBTTL  PAPER TAPE BOOT
```

```
1282                              ;
1283                              ; PAPER TAPE BOOT
1284                              ;
1285         177550              HSR=177550                      ;HIGH SPEED READER ADDRESS
1286         177560              LSR=177560                      ;LOW SPEED READER ADDRESS
1287                              ;         .=ORIGIN+1400
1288                              ;
1289  007400  012701  160000     PTBOOT: MOV    #160000,R1       ;SET MEMORY CHECK LIMITS
1290  007404  012702  000004             MOV    #4,R2            ;TRAP ADDRESS IS LOC. 4
1291  007410  012703  167500             MOV    #DEV+4!160000,R3 ;POINTER TO DEVICE ADDRESSES
1292  007414  010712                     MOV    PC,@R2           ;PRESET TRAP ADDRESS IN LOC. 4
1293  007416  012706  000024             MOV    #24,SP           ;STACK SET UP AT SPECIAL ADDRESS
1294  007422  014304             DEV1:   MOV    -(R3),R4         ;GET DEVICE ADDRESS
1295  007424  005714                     TST    @R4              ;CHECK AVAILABILITY OF DEVICE
1296  007426  100775                     BMI    DEV1             ;CHECK DEVICE FOR ERRORS
1297  007430  010712                     MOV    PC,@R2           ;RESET TRAP ADDRESS AT LOC. 4
1298  007432  012706  000024             MOV    #24,SP           ;SPECIAL ADDRESS USED AS MASK LATER
1299  007436  010441                     MOV    R4,-(R1)         ;DO MEM CHK;READER STATUS ADDRESS
1300                                                             ;IS MOVED
1301  007440  040601                     BIC    SP,R1            ;SET R1=X7752,MASK IN SP=24
1302  007442  010111                     MOV    R1,@R1           ;STORE OWN ADDRESS IN POINTER
1303  007444  011102             LOOP:   MOV    @R1,R2           ;GET BYTE POINTER
1304  007446  005214                     INC    @R4              ;ENABLE READER
1305  007450  105714                     TSTB   @R4              ;TEST DONE BIT
1306  007452  100376                     BPL    .-2              ;WAIT UNTIL READY
1307  007454  116412  000002             MOVB   2(R4),@R2        ;THEN PICK IT UP AND STORE IT
1308  007460  005211                     INC    @R1              ;BUMP POINTER
1309  007462  120227  000375             CMPB   R2,#375          ;STORED JUMP OFFSET?
1310  007466  001366                     BNE    LOOP             ;NOT YET
1311  007470  105222                     INCB   (R2)+            ;YES,ALL DONE
1312  007472  000142                     JMP    -(R2)            ;GO EXECUTE AS BRANCH
1313                              ;
1314                              ; DEVICE ADDRESSES FOLLOW - DO NOT CHANGE THE ORDER
1315                              ;
1316  007474  177560             DEV:    LSR                     ;LOW SPEED READER
1317  007476  177550                     HSR                     ;HIGH SPEED READER
1318
1319                                      .SBTTL  CASSETTE BOOT
```

```
1322                              ;
1323                              ; CASSETTE BOOT
1324                              ;
1325         177500              TACS=177500                     ;TA-11 CONTROL AND STATUS REGISTER
1326                              ;         .=ORIGIN+1500
1327  007500  012700  177500     TABOOT: MOV    #TACS,R0         ;SELECT UNIT #0
1328  007504  005010                     CLR    (R0)
1329  007506  010701             RES:    MOV    PC,R1            ;USE FOR PIC
1330  007510  062701  000052             ADD    #TABLE-.,R1      ;R1 HOLDS ADDR. OF COMMAND TABLE
1331  007514  012702  000375             MOV    #375,R2          ;MEMORY PTR. AND DATA FLAG
1332  007520  112103                     MOVB   (R1)+,R3         ;TEST BITS
1333                              ;
1334  007522  112110             LOOP1:  MOVB   (R1)+,(R0)       ;COMMAND FROM TABLE TO TACS
1335  007524  100413                     BMI    DONE             ;WHEN COMMAND CODE NEG., QUIT
1336  007526  130310             LOOP2:  BITB   R3,(R0)          ;TEST READY AND T-REQ BITS IN TACS
1337  007530  001776                     BEQ    LOOP2            ;LOOP 'TIL SOMETHING COMES UP
1338  007532  105202                     INCB   R2               ;ADVANCE MEMORY POINTER
1339  007534  100772                     BMI    LOOP1            ;IF MINUS, TRY NEXT COMMAND
1340  007536  116412  000002             MOVB   2(R0),(R2)       ;READ DATA INTO MEMORY
1341  007542  120337  000000             CMPB   R3,#000          ;FIRST BYTE READ SHOULD BE '240'
1342  007546  001767                     BEQ    LOOP2            ;IF O.K., GO READ ANOTHER BYTE
1343  007550  000000             STOP:   HALT                    ;HALT ON ERROR
1344  007552  000755                     BR     RES              ;RESTART ON CONTINUE
1345                              ;
1346  007554  005710             DONE:   TST    (R0)             ;CHECK FOR ERROR
1347  007556  100774                     BMI    STOP             ;HALT ON ERROR
1348  007560  005007                     CLR    PC               ;= 'JMP @#0'
1349                              ;
1350  007562  017640             TABLE:  .WORD  17640            ;.BYTE 240; READY+T-REQ,
1351                                                             ;.BYTE 37; ILBS+READY+GO
1352  007564  002415                     .WORD  2415             ;.BYTE 15; SFB+GO
1353                                                             ;.BYTE  5; READ+GO
1354  007566  112024                     .WORD  112024           ;.BYTE 24; READ+ILBS
1355                                                             ;.BYTE 224; READ+ILBS+E.O.;TABLE
1356  007570  000000  000000             .WORD  0,0              ;THESE ARE FILLER WORDS
1357  007574  167500                     .WORD  TABOOT!160000    ;     ;POWER UP VECTOR AND PRIORITY
1358  007576  000340                     .WORD  340              ;
1359
1360
1361                                      .SBTTL  MR11-DB BOOT
```

```
1363                              ;MR11-DB BULK STORAGE PROGRAM LOADER LISTING
1364
1365                              ;       .=ORIGIN+1600                          ;KEEP TRACK OF ORIGIN
1366
1367  007600  010702             RF11:   MOV PC,R2               ;FIXED HEAD DISK (256 KW)
1368  007602  000451                     BR OTHER
1369  007604  177462                     177462
1370  007606  000005                     5
1371
1372  007610  010702             RK11:   MOV PC,R2               ;MOVING HEAD DISK (CARTRIDGE)
1373  007612  000445                     BR OTHER
1374  007614  177406                     177406
1375  007616  000005                     5
1376
1377
1378  007620  010702             TC11:   MOV PC,R2
1379  007622  000417                     BR TAPES
1380  007624  177344                     177344                 ;ADDRESS OF WORD COUNT
1381  007626  000005                     5                      ;LAST COMMAND
1382  007630  004003                     4003                   ;FIRST COMMAND
1383  007632  100000                     100000                 ;DONE MASK
1384  007634  024000                     24000                  ;ERROR MASK
1385
1386
1387  007636  010702             TM11:   MOV PC,R2
1388  007640  000410                     BR TAPES
1389  007642  172524                     172524                 ;ADDRESS OF BYTE COUNT
1390  007644  060003                     60003                  ;LAST COMMAND
1391  007646  060011                     60011                  ;FIRST COMMAND
1392  007650  000200                     200                    ;DONE MASK
1393  007652  100000                     100000                 ;ERROR MASK
1394
1395
1396  007654  010702             RP11:   MOV PC,R2               ;MOVING HEAD DISK (PACK)
1397  007656  000423                     BR OTHER
1398  007660  176716                     176716
1399
1400
1401  007662  000005             TAPES:  RESET
1402  007664  010200                     MOV R2,R0              ;GET THE ADDRESS OF THE BRANCH
1403  007666  005720                     TST (0)+               ;R0 TO POINT AT LAST COMMAND
1404  007670  012001                     MOV (0)+,R1            ;GET THE WORD COUNT ADDRESS
1405  007672  005311                     DEC (1)                ;SET UP FOR ADVANCE 1 RECORD
1406  007674  005720                     TST (0)+               ;MOVE R0 TO FIRST COMMAND
1407  007676  012041                     MOV (0)+,-(1)          ;COMMAND WORD TO COMMAND REG,
1408  007700  031011                     BIT (0),(1)            ;LOOK FOR DONE INDICATORS
1409  007702  001776                     BEQ .-2                ;NONE SET, TRY AGAIN
1410  007704  005720                     TST (0)+               ;DONE FIRST COMMAND, CHECK FOR ERROR
1411  007706  031041                     BIT (0),-(1)           ;LOOK FOR SET ERROR BITS
1412  007710  001406                     BEQ OTHER              ;NO ERRORS - TRY THE READ
1413  007712  000112             AGAIN:  JMP (2)                ;RERUN FOR ERRORS
1414
1415
1416  007714  167600             RFVEC:  RF11:160000                     ;RF11 POWER UP VECTOR
```

```
1417  007716  000340                     340
1418
1419  007720  010702             RC11:   MOV PC,R2               ;FIXED HEAD DISK (64KW)
1420  007722  000401                     BR OTHER
1421  007724  177450                     177450                 ;ADRS OF WORD COUNT (COMMAND+2)
1422                                                             ;COMMAND WORD (5) IS THE RESET
1423
1424  007726  000005             OTHER:  RESET
1425  007730  010200                     MOV R2,R0              ;R0 TO POINT AT WORD COUNT ADRS
1426  007732  005720                     TST (0)+               ;POINT TO ADDRESS
1427  007734  012001                     MOV (0)+,R1            ;WORD COUNT ADDRESS TO R1
1428  007736  012711  177000             MOV #-1000,(1)         ;LOAD WORD COUNT
1429  007742  011041                     MOV (0),-(1)           ;COMMAND TO COMMAND REGISTER
1430  007744  032711  100200             BIT #100200,(1)        ;CHECK FOR ERROR OR DONE
1431  007750  001775                     BEQ .-4                ;IF NEITHER, KEEP LOOKING
1432  007752  100757                     BMI AGAIN              ;ERROR, TRY AGAIN
1433  007754  005007                     CLR PC
1434
1435  007756  000000                     0                      ;FILLER
1436  007760  167610             RKVEC:  RK11:160000                     ;RK POWER UP VECTOR
1437  007762  000340                     340
1438  007764  167720             RCVEC:  RC11:160000                     ;RC POWER UP VECTOR
1439  007766  000340                     340
1440  007770  167654             RPVEC:  RP11:160000                     ;RP POWER UP VECTOR
1441  007772  000340                     340
1442  007774  167620             TCVEC:  TC11:160000                     ;TC11 POWER UP VECTOR
1443  007776  000340                     340
1444
1445                                     .SBTTL  ROM VERSION 1 VALUES
1446                                     .PAGE
```

```
1447                             .DSABL  AMA
1448
1449                    ;DATA PATTERN STORED IN THE GT40 BOOTSTRAP VERSION 1
1450
1451                    ;       **********; THIS IS A IMAGE LISTING OF THE GT40 <VT40> BOOTSTRAP ********
1452                    !
1453                    !       THE DATA IS A MIRROR IMAGE OF THAT IN THE BOOTSTRAP   MS
1454                    ;       ONLY THE ADDRESS FIELD IS CHANGED
1455
1456                    ;BOOTVT.S89  5/2/72     <SPECIAL>
1457
1458
1459                    ;       VT-40 BOOTSTRAP LOADER, VERSION S89, RELEASE R01, 5/2/72
1460
1461                    ;       COPYRIGHT 1972, DIGITAL EQUIPMENT CORPORATION,
1462                    ;       146 MAIN STREET
1463                    ;       MAYNARD, MASSACHUSSETTS
1464                    ;                               01754
1465
1466
1467                    ;       WRITTEN BY JACK BURNESS, SENIOR SYSTEMS ARCHITECT;
1468
1469
1470
1471
1472                    ;       THIS ROUTINE IS INTENDED TO BE LOADED IN THE ROM PORTION OF THE VT-40.
1473
1474
1475                    ;       REGISTER DEFINITIONS;
1476
1477    000000                  R0=X0
1478    000001                  R1=X1
1479    000002                  R2=X2
1480    000003                  R3=X3
1481    000004                  R4=X4
1482    000005                  R5=X5
1483    000006                  R6=X6
1484    000007                  R7=X7
1485
1486    000006                  SP=R6
1487    000007                  PC=R7
1488
1489    000000                  RET1=R0                 ;RETURN OF VALUE REGISTER,
1490    000001                  INP1=R1                 ;ARGUMENT FOR CALLED FUNCTION
1491    000002                  INP2=R2                 ;SECOND ARGUMENT,
1492    000003                  WORK1=R3                ;FIRST WORK REGISTER,
1493    000004                  WORK2=R4                ;SECOND WORKING REGISTER,
1494    000005                  SCR1=R5                 ;SCRATCH REGISTER,
1495
1496    000003                  LCKSM=WORK1             ;OVERLAPPING DEFINITIONS FOR LOADER PORTION,
1497    000000                  LBYT=RET1
1498    000005                  LBC=SCR1
1499    000001                  LADR=INP1
1500
```

```
1501
1502
1503
1504    036000                  COREND=36000            ;FIRST LOCATION OF NON-CORE,
1505    166000                  ROMORG=166000           ;WHERE THE ROM PROGRAM SHOULD GO,
1506
1507    000000                  STARTX=0                ;WHERE TO START DISPLAYING THE X POSITIONS;
1508    001360                  STARTY=1360             ;WHERE TO START DISPLAYING THE Y,
1509
1510
1511    022000                  VT40PC=17200=130000     ;VT40 PROGRAM COUNTER,
1512    027560                  KBDIS=27560             ;TTY INPUT STATUS,
1513    025614                  P10S=25614              ;PDP-10 OUTPUT STATUS,
1514    025610                  P10IS=25610             ;PDP-10 INPUT STATUS,
1515
1516    027562                  KBDIB=KBDIS+2           ;TTY INPUT BUFFER,
1517    025612                  P10IB=P10IS+2           ;PDP-10 INPUT CHARACTER,
1518    025616                  P10OB=P10OS+2           ;PDP-10 OUTPUT BUFFER,
1519
1520
1521    045776                  P10OC=COREND-2+10000    ;CHARACTER TO BE SENT TO THE PDP-10
1522    045772                  P10IC=P10OC-4           ;INPUT CHARACTER FROM 10 PLUS ONE SAVE CHARACTER
1523    045770                  STKSRT=P10IC-2-30000    ;FIRST LOCATION OF STACK,
1524
1525
1526    160000                  JMPDIS=160000           ;THE VT-40 DISPLAY JUMP INSTRUCTION,
1527
1528
1529    000024                  PWRFAL=24               ;POWER FAIL RESTART LOCATION,
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541    016000                  .=16000
1542                    ;       .=ROMORG                ;SET THE ORIGIN NOW;;;;
1543
1544
1545
1546
1547
1548
1549
1550    016000  012705  000024  START4: MOV    #PWRFAL+2,SCR1   ;PICK UP POINTER TO P.F, STATUS,
1551    016004  005015          CLR    @SCR1            ;CLEAR IT OUT TO BE SURE,
1552    016006  010745          MOV    PC,-(SCR1)       ;SET UP THE RESTART LOCATION,
1553
1554    016010  000005          RESET                   ;RESET THE BUS,
```

```
1555
1556   016032  012767  000007  007570        MOV     #7,P10I$         ;INITIALIZE PDP-10 INPUT
1557   016040  012767  000001  011532        MOV     #1,KBDI$         ;INITIALIZE TTY INPUT,
1558   016046  012767  000201  007560        MOV     #201,P10OS       ;INITIALIZE PDP-10 OUTPUT,
1559
1560
1561
1562   016054  012706  015770      RSTRT:    MOV     #STKSRT,SP       ;SET UP THE STACK NOW!
1563   016040  005001                        CLR     LADR             ;CLEAR ADDRESS POINTER,
1564   016042  012702  160000                MOV     #JMPDIS,INP2      ;PLACE A DISPLAY JUMP INSTRUCTION IN A REGISTER,
1565   016046  010221                        MOV     INR2,(LADR)+      ;MOVE IT TO LOCATION 0,
1566   016050  012711  166756                MOV     #DISPRG+150000,(LADR)    ;MOVE ADDRESS POINTER INTO 2,
1567   016054  012701  000030                MOV     #PWRFAL+4,LADR    ;SET UP WHERE WE WILL STORE CHARACTERS,
1568   016060  005000                        CLR     RET$             ;PREPARE TO INSERT A ZERO CHARACTER,
1569   016062  004767  000022                JSR     PC,DOCHAR        ;INSERT IT NOW,
1570   016066  005067  003706                CLR     VT40PC           ;CLEAR THE DISPLAY PROGRAM COUNTER AND START,
1571
1572   016072  004767  000210      MAJOR:    JSR     PC,GTCHR         ;GT A CHARACTER NOW,
1573   016076  000240                        NOP
1574   016100  000240                        NOP
1575   016102  000240                        NOP
1576   016104  012746  166072                MOV     #MAJOR+150000,-(SP)      ;INSERT IN DISPLAY BUFFER NOW,
1577
1578   016110  010105      DOCHAR:   MOV     LADR,SCR1        ;GT CURRENT BUFFER POSITION NOW,
1579   016112  022525                        CMP     (SCR1)+,(SCR1)+  ;BYPASS CURRENT DISPLAY JUMP,
1580   016114  005025                        CLR     (SCR1)+          ;CLEAR FUTURE ADDRESS FOR JUMP,
1581   016116  010225                        MOV     INR2,(SCR1)+     ;STICK IN TEMPORARY JUMP WHILE WE REPLACE CURREN
1582   016120  005015                        CLR     (SCR1)           ;A DISPLAY JUMP TO ZERO,
1583   016122  005011                        CLR     (LADR)           ;NOW REPLACE CURRENT DISPLAY JUMP BY THE CHARACT
1584   016124  050021                        BIS     RET$,(LADR)+     ;IT'S DONE THIS WAY TO WASTE 2 CYCLES,
1585   016126  010211                        MOV     INR2,(LADR)      ;TO AVOID TIMING PROBLEMS WITH THE VT40,
1586   016130  000207                        RTS     PC               ;AND FINALLY RETURN,
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603   016132  004767  000124      GT8:      JSR     PC,GTSIX         ;GT SIX BITS NOW,
1604   016136  010046                        MOV     RET$,-(SP)       ;SAVE THE CHARACTER NOW,
1605   016140  000401                        BR      GTP84            ;BYPASS THE 8'ER
1606   016142  005002      GT84:     CLR     INR2             ;RESET THE MAGIC REGISTER NOW,
1607   016144  005722      GTP84:    TST     (INP2)+          ;INCREMENT WHERE TO GO,
1608   016146  066207  166250                ADD     GT8T8+150000(INP2),PC    ;UPDATE PC NOW,
```

```
1609
1610           016152              GT8P8.
1611
1612   016152  004767  000104      GT81:     JSR     PC,GTSIX         ;GT A CHARACTER NOW,
1613   016156  010004                        MOV     RET$,WORK2       ;SAVE FOR A SECOND,
1614   016160  006300                        ASL     RET$
1615   016162  006300                        ASL     RET$             ;SHIFT TO LEFT OF BYTE
1616   016164  106300                        ASLB    RET$
1617   016166  106116                        ROLB    @SP              ;PACK THEM IN,
1618   016170  106300                        ASLB    RET$
1619   016172  106116                        ROLB    @SP              ;A GOOD 8 BIT THING,
1620   016174  012600                        MOV     (SP)+,RET$       ;POP AND RETURN NOW,
1621   016176  000207                        RTS     PC
1622
1623   016200  006300      GT82:     ASL     RET$             ;WORST CASE: SHIFT 4
1624   016202  006300                        ASL     RET$
1625   016204  106300                        ASLB    RET$
1626   016206  106104                        ROLB    WORK2
1627   016210  106300                        ASLB    RET$
1628   016212  106104                        ROLB    WORK2
1629   016214  106300                        ASLB    RET$
1630   016216  106104                        ROLB    WORK2
1631   016220  106300                        ASLB    RET$
1632   016222  106104                        ROLB    WORK2
1633   016224  010400                        MOV     WORK2,RET$
1634   016226  012604                        MOV     (SP)+,WORK2
1635   016230  000207                        RTS     PC
1636
1637   016232  006100      GT83:     ROL     RET$
1638   016234  006100                        ROL     RET$
1639   016236  006004                        ROR     WORK2
1640   016240  106000                        RORB    RET$
1641   016242  006004                        ROR     WORK2
1642   016244  106000                        RORB    RET$
1643   016246  005726                        TST     (SP)+            ;FINAL CHARACTER ASSEMBLED,
1644   016250  000207                        RTS     PC               ;FUDGE STACK,
1645                                                                   ;AND RETURN NOW,
1646           016250              GT8T8     =       .-2              ;PUSH ZERO CONDITION BACK INTO NEVER-NEVER LAND,
1647
1648   016252  000000                        .WORD   GT81-GT8P
1649   016254  000026                        .WORD   GT82-GT8P
1650   016256  000060                        .WORD   GT83-GT8P
1651   016260  177770                        .WORD   GT84-GT8P
1652
1653
1654   016262  004767  000020      GTSIX:    JSR     PC,GTCHR
1655   016266  020027  000040                CMP     RET$,#40
1656   016272  002546                        BLT     LBAD
1657   016274  020027  000137                CMP     RET$,#137
1658   016300  003143                        BGT     LBAD
1659   016302  000207                        RTS     PC
1660
1661
1662
```

```
1663  016304  005726          GTCMP:  TST     (SP)+               ;UPDATE THE STACK,
1664
1665  016306  012700  015772  GTCHR:  MOV     #P1810-30000,RET1       ;SET UP POINTER TO THE INPUT CHARACTER.
1666  016312  004767  000064  GTCHL:  JSR     PC,CHECK
1667  016316  005710          TST     @RET1               ;ANY CHARACTERS THERE?
1668  016320  001774          BEQ     GTCHL
1669  016322  011046          MOV     @RET1,=(SP)         ;PUSH THE CHAR ON THE STACK,
1670  016324  005020          CLR     (RET1)+             ;CLEAR THE CHAR GOT FLAG NOW.
1671  016326  042716  177600  BIC     #=200,(SP)          ;CLEAR AWAY PARITY NOW,
1672  016332  001764          BEQ     GTCMP               ;IF ZERO, GT ANOTHER
1673  016334  022716  000177  CMP     #177,(SP)
1674  016340  001761          BEQ     GTCMP               ;ALSO IGNORE RUBOUTS,
1675  016342  022710  000175  CMP     #175,@RET1          ;WAS IT A "175"
1676  016346  001007          BNE     GTNP                ;NOPE,
1677  016350  011610          MOV     (SP),@RET1          ;YEP, RESET IN CASE OF ABORT.
1678  016352  021027  000122  CMP     @RET1,#122          ;IS IT AN R
1679  016356  001626          BEQ     RSTRT               ;YEP, RESTART
1680  016360  021027  000114  CMP     @RET1,#114          ;IS IT AN L
1681  016364  001455          BEQ     LOAD                ;YEP, LOAD,
1682
1683  016366  011610          GTNP:   MOV     (SP),@RET1          ;NOW DO THE FOUGING,
1684  016370  012600          MOV     (SP)+,RET1
1685  016372  020027  000175  CMP     RET1,#175
1686  016376  001743          BEQ     GTCHR               ;IF ALTMODE, LOOP
1687  016400  000207          RTS     PC
1688
1689
1690
1691
1692
1693
1694
1695
1696  016402  005767  027370  CHECK:  TST     P180C               ;DO WE WANT TO OUTPUT?
1697  016406  001410          BEQ     CHECK1              ;NO,
1698  016410  105767  007200  TSTB    P180S               ;WE DO, IS THE 10 READY?
1699  016414  100005          BPL     CHECK1              ;NOT QUITE,
1700  016416  016767  027354  007172  MOV     P180C,P180B         ;IT'S READY, SEND THE CHARACTER,
1701  016424  005067  027346  CLR     P180C               ;AND THE SAVED CHARACTER,
1702
1703  016430  105767  011124  CHECK1: TSTB    KBDIS               ;HEY! IS THE KEYBOARD READY?
1704  016434  100014          BPL     CHECK3              ;NOPE, NO LUCK,
1705  016436  116746  011120  MOVB    KBDIB,=(SP)         ;YEP, SAVE THE CHARACTER NOW,
1706  016442  012767  000001  011110  MOV     #1,KBDIS            ;AND REENABLE THE COMMUNICATIONS DEVICE.
1707
1708  016450  004767  177726  CHECK2: JSR     PC,CHECK            ;IS THE OUTPUT READY?
1709  016454  005767  027316  TST     P180C
1710  016460  001373          BNE     CHECK2              ;IF NOT, WAIT TILL DONE,
1711  016462  012667  007130  MOV     (SP)+,P180B         ;AND THEN SEND OUT THE CHARACTER,
1712
1713
1714  016466  105767  007116  CHECK3: TSTB    P1013               ;IS THE 10 TALKING TO ME,
1715  016472  100011          BPL     CHECK4              ;NOPE, EXIT,
1716  016474  116767  007112  027270  MOVB    P1018,P1810C        ;GT THE CHARACTER NOW,
```

```
1717  016502  052767  177400  027262  BIS     #=400,P181C         ;MAKE SURE IT'S NONE ZERO,
1718  016510  012767  000007  007072  MOV     #7,P1019            ;REINITIALIZE COMMUNICATION LINE.
1719
1720  016516  000207          CHECK4: RTS     PC                  ;AND RETURN,
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737  ;       THE   L O A D E R
1738
1739  016520  005002          LOAD:   CLR     INP2                ;RESET TO FIRST 8 BIT CHARACTER.
1740  016522  012712  172000  MOV     #172000,(INP2)      ;AND ALSO CLEVERLY STOP THE VT40.
1741  016526  012706  015770  MOV     #STKSRT,SP          ;RESET STACK POINTER NOW,
1742
1743  016532  005003          LLD2:   CLR     LCKSM               ;CLEAR THE CHECKSUM
1744  016534  004767  000070  JSR     PC,LPTR             ;GT A BYTE NOW,
1745  016540  105303          DECB    LBYT                ;IS IT ONE?
1746  016542  001373          BNE     LLD2                ;NOPE, WAIT AWHILE,
1747  016544  004767  000060  JSR     PC,LPTR             ;YEP, GT NEXT CHARACTER,
1748
1749  016550  004767  000072  JSR     PC,LGWRD            ;GT A WORD,
1750  016554  010005          MOV     LBYT,LBC            ;GT THE COUNTER NOW,
1751  016556  162705  000004  SUB     #4,LBC              ;CHOP OFF EXTRA STUFF,
1752  016562  022705  000002  CMP     #2,LBC              ;NULL?
1753  016566  001437          BEQ     LJMP                ;YEP, MUST BE END,
1754  016570  004767  000052  JSR     PC,LGWRD            ;NOPE, GT THE ADDRESS,
1755  016574  010001          MOV     LBYT,LADR           ;AND REMEMBER FOR OLD TIMES SAKE.
1756
1757  016576  004767  000026  LLD3:   JSR     PC,LPTR             ;GT A BYTE (DATA)
1758  016602  002010          BGE     LLD5                ;ALL DONE WITH THE COUNTER?
1759  016604  105703          TSTB    LCKSM               ;YEP, GOOD CHECK SUM?
1760  016606  001751          BEQ     LLD2                ;NOPE, LOAD ERROR,
1761
1762  016610  012700          LBAD:   MOV     (PC)+,RET1          ;SEND OUT SOME CHARACTERS NOW,
1763  ;       .BYTE   175,102             ;"CTRL BAD"
1764  016612  102  175  .BYTE   102,175             ;"BAD CTRL"
1765  016614  004767  000116  JSR     PC,SENDIT
1766  016620  000167  177210  JMP     RSTRT
1767
1768  016624  116021          LLD4:   MOVB    LBYI,(LADR)+        ;PLACE THE BYTE IN CORE,
1769  016626  000763          BR      LLD3                ;GT ANOTHER ONE,
1770
```

```
1771  016630  004767  177276   LPTR:  JSR   PC,GT8           ;GT 8 BITS NOW.
1772  016634  060003          ADD   LBYT,LCKSM       ;UPDATE CHECKSUM
1773  016636  042700  177400          BIC   #177400,LBYT     ;CLEAN UP THE BYTE NOW.
1774  016642  005305          DEC   LBC              ;UPDATE THE COUNTER.
1775  016644  000207          RTS   PC               ;RETURN NOW.
1776
1777  016646  004767  177756   LGWRD: JSR   PC,LPTR          ;GT A CHARACTER.
1778  016652  010046          MOV   LBYT,-(SP)       ;SAVE FOR A SECOND.
1779  016654  004767  177750          JSR   PC,LPTR          ;GT ANOTHER CHARACTER.
1780  016660  000300          SWAB  LBYT             ;NOW ASSEMBLE THE WORD.
1781  016662  052600          BIS   (SP)+,LBYT       ;AND RETURN WITH A 16 BITER.
1782  016664  000207          RTS   PC
1783
1784  016666  004767  177754   LJMP:  JSR   PC,LGWRD         ;GT A WORD
1785  016672  010046          MOV   LBYT,-(SP)       ;SAVE ON THE STACK.
1786  016674  004767  177730          JSR   PC,LPTR          ;GT A CHARACTER.
1787  016700  105703          TSTB  LCKSM            ;IS IT ZERO?
1788  016702  001342          BNE   LBAD             ;YEP, WHAT CRAP.
1789  016704  032716  000001          BIT   #1,(SP)          ;IS IT ODD?
1790  016710  001406          BEQ   LJMP1            ;YEP. START PROGRAM GOING NOW.
1791  016712  012700          MOV   (PC)+,RET1       ;TELL PDP-10 WE'VE LOADED OK.
1792                           :    .BYTE 175,107          ;"CTRL GOOD"
1793  016714    107    175    .BYTE 107,175          ;"GOOD CTRL"
1794  016716  004767  000006          JSR   PC,SENDIT
1795  016722  000000          HALT
1796  016724  000776          BR    .-2
1797
1798  016726  000136   LJMP1: JMP   @(SP)+           ;AND AWAY WE GO.
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816  016730  004767  177446   SENDIT: JSR  PC,CHECK         ;POLL THE OUTPUT DEVICE NOW.
1817  016734  005767  027036          TST   P10PC            ;OUTPUT CLEAR?
1818  016740  001373          BNE   SENDIT           ;NOPE, LOOP AWHILE LONGER.
1819  016742  010067  006650          MOV   RET1,P10DB       ;SEND OUT THE CHARACTER.
1820  016746  105000          CLRB  RET1             ;CLEAR THE BYTE.
1821  016750  000300          SWAB  RET1             ;AND SWAP THEM NOW.
1822  016752  001366          BNE   SENDIT           ;IF NOT EQUAL, REPEAT.
1823  016754  000207          RTS   PC
1824
```

```
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842                           :              THIS IS THE INITIALIZING VT40 PROGRAM WHICH WILL
1843                           :              JUMP TO THE PROGRAM AFTER THE POWER FAIL LOCATIONS
1844                           :              WHICH WILL JUMP TO ZERO WHICH WILL JUMP BACK TO HERE.
1845
1846
1847
1848
1849  016756  170256   DISPRG: .WORD 170256           ;LOAD STATUS REGISTER FOR NORMAL OPERATION.
1850  016760  115124          .WORD 115124           ;SET POINT MODE, "NORMAL".
1851  016762  000000          .WORD STARTX           ;X COORDINATE
1852  016764  001360          .WORD STARTY           ;Y COORDINATE
1853  016766  100000          .WORD 100000           ;SET CHARACTER MODE.
1854  016770  160000          .WORD JMPDIS           ;THEN JUMP TO THE POWERFAIL LOCATION.
1855  016772  000030          .WORD PWRFAL+4         ;TO DISPLAY USERS CHARACTERS.
1856  016774  000000          .WORD 0
1857  016776  000000          .WORD 0
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871        000001                   .END
```

```
AGAIN    007712    ALTMOD = 000175    BELL     000250    BLIMIT = 007000
BLKOFF = 000020    BLKON  = 000030    BSTART = 001000    CHAR   = 100000
CHARA    001020    CHECK    016402    CHECK1   016430    CHECK2   016450
CHECK3   016466    CHECK4   016516    COREND = 030000    CORSTR = 000004
COUNTR =X000005    CR       006206    CRLF   = 006212    DEV      007474
DEV1     007422    DISJMP = 160000    DISPLA = 177570    DISPRG   016756
DISTCP = 173000    DJMP   = 160000    DL11IB = 175612    DL11IS = 175610
DL11CB = 175616    DL110S = 175614    DNOP   = 164000    DOCHAR   016110
DONE     007554    DONE0    001430    DONE1    001502    DSR      001006
DSTOP  = 173400    DUMP     001016    D2BTYP   002144    END      001416
FNCCCR   016036    ERROR1   001130    ERROR2   001176    FF       000256
FFLOOP   006262    FILCNT   001012    FILE0    007214    FILLER   001010
GETCHR   006964    GETOL    006516    GETOL1   006546    GETEXT   006650
GETSIX   006630    GET8     006664    GET8TB   006700    GET81    006712
GET82    006740    GET83    006772    GET84    006706    GRAPHX = 120000
GRAPHY = 124000    GTBUSE   006424    GTCHL    016012    GTCHP    016304
GTCHR    016306    GTNP     016366    GTP84    016144    GTBIX    016262
GT40PC = 172000    GT40SR = 172002    GT8      016132    GT8P     016152
GT8TB  = 016250    GT81     016152    GT82     016200    GT83     016232
GT84     016142    HEADER   006474    HOLD   =X000004    HSR    = 177550
ICNT     001014    ICOUNT   002124    IMAGE    001004    INCR   = 000100
INDEX  =X000003    INP1   =X000001    INP2   =X000002    INSERT   006350
INSRTL   006406    INSRTX   006422    INTX     040000    INT0   = 002000
INT1   = 002200    INT2   = 002400    INT3   = 002600    INT4   = 003000
INT5   = 003200    INT6   = 003400    INT7   = 003600    ITAL0  = 000040
ITAL1  = 000060    JMPADD = 007012    JMPDIS = 160000    KBDIB  = 027562
KBDIS  = 027560    LADR   =X000001    LBAD     016610    LBC    =X000005
LBYT   =X000000    LCKSM  =X000003    LF       000300    LFLOOP   006310
LFOUND   006330    LFSUB    006304    LGWRD    016046    LINE0  = 000004
LINE1  = 000005    LINE2  = 000006    LINE3  = 000007    LJMP     016666
LJMP1    016726    LLD2     016532    LLD3     016576    LLD4     016624
LOAD     016520    LOADER   007012    LONGV  = 110000    LOOP     007444
LOOP1    007922    LOOP2    007526    LPDARK = 000200    LPLITE = 000300
LPCFF  = 000100    LPON     000140    LPTR     016630    LSR    = 177560
L.ADR  =X000001    L.BAD    007100    L.BC   =X000002    L.BYT  =X000000
L.CKSM =X000005    L.GWRD   007126    L.HALT   007200    L.JMP    007144
L.JMP1   007202    L.LD2    007022    L.LD3    007066    L.LD4    007110
L.PTR    007114    MAJOR    016072    MAXSX  = 017600    MAXSY  = 000077
MAXX   = 001777    MAXY   = 001377    MINSUY = 000100    MINUSX = 020000
MINUSY = 020000    M7       002252    M8       002275    M9       002301
NORMAL   016212    NOTHER   006042    NUMLIN = 000040    NXTCHR   006132
ORIGIN = 166000    OTHER    007726    OUTLIT   006652    O2A      002146
O2AA     002176    PC     =X000007    POINT  = 110000    POINTR =X000001
PRESTR   006624    PRG0     001034    PRG0R    001040    PRG1     001506
PRG1A    001544    PRG1B    001562    PRG1C    001600    PRG1D    001650
PRG2     001670    PRMTRS   001024    PSW    = 177776    PTBOOT   007400
PWRFAL = 000024    P10IB  = 025612    P10IC  = 045772    P10IS  = 025610
P10OP  = 025616    P100C  = 045776    P100S  = 025614    RCVEC    007764
RC11     007720    RELATV = 130000    RES      007506    RESTRT   006068
RETURN   002130    RET1   =X000000    RFVEC    007714    RF11     007600
RKVEC    007760    RK11     007610    ROMADD   001000    ROMORG = 166000
RPVEC    007770    RP11     007654    RSTRT    016034    R0     =X000000
R1     =X000001    R2     =X000002    R3     =X000003    R4     =X000004
R5     =X000005    R6     =X000006    R7     =X000007    SCAN   =X000003
```

```
SCOPE  = 124400    SCOPEB   002132    SCOPEC   002054    SCOPEF   002124
SCOPEG   002112    SCR1   =X000005    SENDIT   012730    SETDUN   006120
SETLP1   006074    SETLP2   006110    SETLP3   006116    SETUP    006432
SWORTV = 104400    SP     =X000006    SR     = 177570    START    006000
STARTA   016000    STARTX = 000000    STARTY = 002360    STATSA = 174000
STATSB = 174000    STKPTR = 000500    STKSRT = 002770    STOP     007550
SWITCH   002306    SYNON  = 000004    TAB      000222    TABCNT =X000002
TABLE    007562    TABOOT   007500    TACS   = 177500    TAPES    007662
TCVEC    007774    TC11     007620    TERM     002022    TMPEND = 007776
TM11     007638    TPCSR  = 177564    TPDBR  = 177566    TYPEM    001712
TYPEMA   001722    TYPEMB   001750    TYPEMC   002030    T1       001060
T1A      001078    T1B      001136    T2       001140    T2A      001156
T2B      001202    T2C      001214    T2D      001224    T2E      001234
T3       001244    T3A      001266    T3AA     001260    T3B      001274
T3C      001304    T3D      001312    T3E      001322    T3F      001332
T4       001352    T4A      001366    T4B      001402    T4E      001414
VT       016244    VT40PC = 022000    WORDS    002002    WORK1  =X000003
WORK2  =X000004    .      = 017000
```

ERRORS DETECTED: 0

            ;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

      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   EQUIPTMENT

GT40 DISPLAY SYSTEM (REF. 7.) OR
GT44 DISPLAY SYSTEM

2.2   STORAGE

THIS PROGRAM USED MEMORY LOCATIONS 0-14000 (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 <BR, NPR AND INTERRUPT>
204   SUB-TEST 2, BASIC VISUAL DISPLAY PATTERNS
        <SELECTED BY SW 00-02>

        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 SUBTESTS 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 FACILITES 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.   MISCELANEOUS

8.1   EXECUTION TIME

        SUB-TEST 1 TAKES APPROXIMATELY 15 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' <DSTEP> 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 'NULL','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 06-02, THE PROGRAM
        MUST BE RESTARTED TO SELECT ANOTHER PATTERN.

```
                              .ENABL  ABS.AMA
                              .TITLE  GT-40/GT-44 INSTRUCTION TEST II MAINDEC-11-DDGTB-B
256                                   .LIST   ME,BIN,SEQ
257                                   .NLIST  MC,MD,CND
258
274
275            000000                 .=0
276   000000   000000         HALT
277   000002   000000         HALT
278                                   ;0 THRU 776 IS FILLED WITH A TRAP CATCHER
285            000024                 .=24
286   000024   012364         LOWPWR
287   000026   000340         340
288            000030                 .=30
289   000030   012320         .WORD   SCOPEA          ;EMT RETURN
290   000032   000340         340
291
292            000200                 .=200
293   000200   000137  001336  JMP     START
294   000204   000137  012464  JMP     START1
295
296            001000                 .=1000
297   001000   172000         GSADD:  172000          ;GS DISPLAY STARTING ADDRESS
298   001002   000320         GSVCT:  320             ;GS DISPLAY STARTING VECTOR
299   001004   000200         DSRBR:  200             ;GS DISPLAY INTERRUPT LEVEL
300   001006   000016         GSCHSZ: 16              ;CHARACTER SIZE (14-16)
301   001010   000030         GSLFSZ: 30              ;LINE FEED SIZE (30-32)
302   001012   001777         GSYAXS: 1777            ;+Y AXIS CUTOFF LOCATION
303   001014   000177         GSSEND: 177             ;SKIP-OUT END CHARACTER
304
305   001016   000000         ICNT:   0               ;PASS COUNTER
306   001020   177776         PSW:    177776
307   001022   013464         DBUF:   BUFFER          ;FIRST WORD IN THE DISPLAY BUFFER
308   001024   013466         DBUF1:  BUFFER+2        ;SECOND WORD
309   001026   013470         DBUF2:  BUFFER+4        ;THIRD WORD
310   001030   013472         DBUF3:  BUFFER+6        ;FOURTH WORD
311   001032   013474         DBUF4:  BUFFER+10       ;FIFTH WORD
312   001034   013476         DBUF5:  BUFFER+12       ;SIX WORD
313   001036   000000         DSAVE:  0               ;TEMP REG.
314   001040   177570         SWR:    177570
315   001042   017476         SIZE:   17476           ;BUFFER SIZE FOR 4K <WORD LENGTH>
316   001044   000000         CNTR:   0
317   001046   000750         LFSIZE: 750             ;LINE FEED DELTA Y SIZE
318   001050   000762         CHSIZE: 762             ;BACK SPACE CHARACTER DELTA X SIZE
```

```
323
327
332                              ;GS ADDRESSES AND VECTORS
333
334   001052   172000         DPC:    172000          ;DISPLAY PC REGISTER
335   001054   172002         DSR:    172002          ;DISPLAY STATUS REGISTER
336   001056   172004         XPOS:   172004          ;X AXIS REGISTER <READ ONLY>
337   001060   172006         YPOS:   172006          ;Y AXIS REGISTER AND GRAPHPLOT REGISTER <READ ONLY>
338
339   001062   000320         DDONE:  320             ;DISPLAY STOP <DONE> VECTOR
340   001064   000322         DDONE1: 322             ;
341
342   001066   000324         LPVCT:  324             ;DISPLAY LIGHT PEN VECTOR
343   001070   000326         LPVCT1: 326             ;
344
345   001072   000330         TIMEVT: 330             ;DISPLAY TIME-OUT <NXM> ERROR VECTOR
346   001074   000332         TMEVT1: 332             ; OR "SHIFT-OUT" VECTOR
347
348                              ;GS INITIALIZATION ROUTINE
349
350   001076   012700  001052  SETUP:  MOV     #DPC,R0         ;SET UP POINTER
351   001102   013701  001000          MOV     GSADD,R1
352   001106   010120         SETUPA: MOV     R1,(0)+
353   001110   062701  000002          ADD     #2,R1
354   001114   022700  001062          CMP     #DPC+10,R0
355   001120   001372                  BNE     SETUPA
356   001122   012700  001062          MOV     #DDONE,R0
357   001126   013701  001002          MOV     GSVCT,R1
358   001132   010120         SETUPB: MOV     R1,(0)+
359   001134   062701  000002          ADD     #2,R1
360   001140   022700  001076          CMP     #DDONE+14,R0
361   001144   001372                  BNE     SETUPB
362   001146   013737  001010  001046  MOV     GSLFSZ,LFSIZE   ;SET UP DELTA LF
363   001154   005437  001046          NEG     LFSIZE          ;NEGATE IT
364   001160   042737  177000  001046  BIC     #177000,LFSIZE  ;MASK IT
365   001166   013737  001006  001050  MOV     GSCHSZ,CHSIZE   ;SET UP DELTA CHAR
366   001174   005437  001050          NEG     CHSIZE          ;NEGATE IT
367   001200   004737  001252          JSR     PC,DOCORE       ;SET UP CORE SIZE
368   001204   042737  177000  001050  BIC     #177000,CHSIZE  ;MASK IT
369   001212   013777  001064  177642  MOV     DDONE1,@DDONE   ;LOAD DONE VECTOR
370   001220   005077  177640          CLR     @DDONE1
371   001224   013777  001070  177634  MOV     LPVCT1,@LPVCT   ;LOAD LIGHT-PEN VECTOR
372   001232   005077  177632          CLR     @LPVCT1
373   001236   013777  001074  177626  MOV     TMEVT1,@TIMEVT  ;LOAD TIME-OUT VECTOR
374   001244   005077  177624          CLR     @TMEVT1
375   001250   000207         RTS     PC
```

```
377                                  ;SUBROUTINE TO DETERMINE THE SIZE OF CORE
378                                  ; AND SET UP LOCATION SIZE WITH THE VALUE
379
380   001252  012737  001304  000004  DOCORE: MOV   #25,@#4          ;SET UP FOR NEW
381   001260  012701  017776          MOV   #17776,R1        ;SET UP ADDRESS
382   001264  005000          CLR   R0
383   001266  062701  020000   1ST:  ADD   #20000,R1        ;MOVE TO THE NEXT BANK
384   001272  005200          INC   R0              ;INC BANK COUNTER
385   001274  005711          TST   (1)             ;TIMEOUT ?
386   001276  022701  177776          CMP   #177776,R1      ;END ?
387   001302  001371          BNE   1S
388   001304  005300   2ST:  DEC   R0              ;DECREMENT BANK COUNT
389   001306  012737  000006  000004          MOV   #6,@#4          ;RESET BUSS ERROR
390   001314  022626          CMP   (SP)+,(SP)+     ;POP THE STACK X2
391   001316  162701  020000          SUB   #20000,R1
392   001322  010137  001042          MOV   R1,SIZE         ;LOAD SIZE
393   001326  162737  007776  001042          SUB   #7776,SIZE      ;BYPASS LOADER
394   001334  000207          RTS   PC              ;EXIT
395
```

```
397
398   001336  012777  000340  177494  START: MOV   #340,@#PSW
399   001344  012706  000500          MOV   #STKPTR,SP
400   001350  004737  001076          JSR   PC,SETUP
401   001354  005037  001016          CLR   IONT
402   001360  012701  001366          MOV   #GTBUSS+2,R1

404                                  ;TEST FOR BUSS ERRORS ON DISPLAY ADDRESSES

406   001364  104000          GTBUSS: SCOPE
407   001366  000005          RESET
408   001370  005077  177460          CLR   @DSB            ; ON DISPLAY STATUS
409   001374  000240          NOP
410   001376  005077  177494          CLR   @XRS            ; ON DISPLAY X REGISTER
411   001402  000240          NOP
412   001404  005077  177450          CLR   @YRS            ; ON DISPLAY Y REGISTERS
413   001410  000005          RESET

415                                  ;INCREMENT P.C. TEST
416                                  ;COMPLEX - BUFFER LENGTH

418   001412  104000          GTRC:  SCOPE
419   001414  013702  001022          MOV   DBUF,R2         ;SET UP POINTER
420   001420  012722  170000   1ST:  MOV   #170000,(23+    ;MOVE DSTOP INTO THE BUFFER
421   001424  023702  001042          CMP   SIZE,R2         ;FINISHED FILLING THE BUFFER?
422   001430  001373          BNE   1S              ;NO

424   001432  104000          SCOPE
425   001434  013777  001022  177410          MOV   DBUF,@DPC       ;YES, START THE DISPLAY
426   001442  013737  001022  001036          MOV   DBUF,DSAVE
427   001450  013702  001042          MOV   SIZE,R2         ;SETUP A COUNT
428   001454  005302          DEC   R2
429   001456  017704  177372          GTPCA: MOV   @DSR,R4
430   001462  100402          BMI   1S
431   001464  000000          HALT                  ;ERROR, STOP FLAG FAILED TO SET
432   001466  000421          BR    GTS
433
434   001470  062737  000002  001036  1ST:  ADD   #2,DSAVE
435   001476  017700  177350          MOV   @DSR,R0         ;READ DISPLAY P.C.
436   001502  023700  001036          CMP   DSAVE,R0        ;DID IT INCREMENT BY 2?
437   001506  001402          BEQ   2S              ;YES
438   001510  000000          HALT                  ;DISPLAY PC FAILED TO INCREMENT
439   001512  000407          BR    GTS             ;PROPERLY
440
441   001514  020037  001036   2ST:  CMP   R0,DSAVE        ;FINISHED THE BUFFER ?
442   001520  001404          BEQ   GTS             ;BR IF YES
443   001522  012777  000001  177322          MOV   #1,@DPC         ;SINGLE STEP THE DISPLAY
444   001530  000752          BR    GTPCA           ;TRY AGAIN
445
```

```
447                                         ;TESTED BY "LOAD STATUS REGISTER A"
448
449   001532  104000                  GT0:  SCOPE
450   001534  012777  172040  177260        MOV   #172040,@DBUF   ;ITALICS ENABLE=1 ITALICS=0
451   001542  013777  001022  177302         MOV   DBUF,@DPC      ;LOAD DISPLAY P.C.
452   001550  017700  177300               MOV   @DSR,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  .+4            ;ITALICS BIT FAILED TO RESET
457
458   001570  104000                  GT1:  SCOPE
459   001572  012777  172060  177222        MOV   #172060,@DBUF   ;ITALICS ENABLE=1 ITALICS=1
460   001600  013777  001022  177244        MOV   DBUF,@DPC       ;LOAD DISPLAY P.C.
461   001606  017700  177242               MOV   @DSR,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                  GT2:  SCOPE
468   001630  012777  172000  177164        MOV   #172000,@DBUF   ;ITALICS ENABLE=0 ITALICS=0
469   001636  013777  001022  177206        MOV   DBUF,@DPC       ;LOAD DISPLAY P.C.
470   001644  017700  177204               MOV   @DSR,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                  GT3:  SCOPE
478   001666  012777  172002  177126        MOV   #172002,@DBUF   ;COLOR ENABLE=1 COLOR=0
479   001674  013777  001022  177150        MOV   DBUF,@DPC       ;LOAD DISPLAY P.C.
480   001702  004737  012352               JSR   7,DLAY1         ;EXECUTE A PROGRAM DELAY
      (1)
481   001706  017700  177142               MOV   @DSR,R0         ;READ DISPLAY STATUS REGISTER
482   001712  042700  177773               BIC   #177773,R0      ;MASK TO BIT 2
483   001716  022700  000000               CMP   #0,R0           ;TEST R0
484   001722  001401                        BEQ   .+4            ;
485   001724  000240                        NOP                  ;COLOR BIT FAILED TO RESET
486
487
488   001726  104000                  GT4:  SCOPE
489   001730  012777  172003  177064        MOV   #172003,@DBUF   ;COLOR ENABLE=1 COLOR=1
490   001736  013777  001022  177106        MOV   DBUF,@DPC       ;LOAD DISPLAY P.C.
491   001744  004737  012352               JSR   7,DLAY1         ;EXECUTE A PROGRAM DELAY
      (1)
492   001750  017700  177100               MOV   @DSR,R0         ;READ DISPLAY STATUS REGISTER
493   001754  042700  177773               BIC   #177773,R0      ;MASK TO BIT 2
494   001760  022700  000004               CMP   #4,R0           ;TEST R0
495   001764  001401                        BEQ   .+4            ;
496   001766  000240                        NOP                  ;COLOR BIT FAILED TO SET
```

```
498
499   001770  104000                  GT5:  SCOPE
500   001772  012777  172000  177022        MOV   #172000,@DBUF   ;COLOR ENABLE=0 COLOR=0
501   002000  013777  001022  177044        MOV   DBUF,@DPC       ;LOAD DISPLAY P.C.
502   002006  017700  177042               MOV   @DSR,R0         ;READ DISPLAY STATUS REGISTER
503   002012  042700  177773               BIC   #177773,R0      ;MASK TO BIT 2
504   002016  022700  000004               CMP   #4,R0           ;TEST R0
505   002022  001401                        BEQ   .+4            ;
506   002024  000240                        NOP                  ;COLOR ENABLE FAILED TO INHIBIT
507                                                               ;RESETTING OF COLOR BIT
508
509                                         ;TESTED BY "LOAD CHARACTER" MODE
510
511   002026  104000                  GT6:  SCOPE
512   002030  012777  100004  176764        MOV   #100004,@DBUF   ;LOAD LINE TYPE ENABLE =1 AND LINE TYPE VALUE =0
513   002036  012777  172000  176760        MOV   #172000,@DBUF1
514   002044  013777  001022  177000        MOV   DBUF,@DPC       ;LOAD DISPLAY P.C.
515   002052  017700  176776               MOV   @DSR,R0         ;READ DISPLAY STATUS REGISTER
516   002056  042700  177774               BIC   #177774,R0      ;MASK TO BITS 1-0
517   002062  022700  000000               CMP   #0,R0           ;TEST R0
518   002066  001401                        BEQ   .+4            ;
519   002070  000000                        HALT                 ;LINE BITS 1-0 FAILED TO RESET
520
521   002072  104000                  GT7:  SCOPE
522   002074  012777  100007  176720        MOV   #100007,@DBUF   ;LINE TYPE ENABLE =1 LINE TYPE =3
523   002102  012777  172000  176714        MOV   #172000,@DBUF1  ;LOAD STOP
524   002110  013777  001022  176734        MOV   DBUF,@DPC       ;LOAD DISPLAY P.C.
525   002116  017700  176732               MOV   @DSR,R0         ;READ DISPLAY STATUS REGISTER
526   002122  042700  177774               BIC   #177774,R0      ;MASK TO BITS 1-0
527   002126  022700  000003               CMP   #3,R0           ;TEST R0
528   002132  001401                        BEQ   .+4            ;
529   002134  000000                        HALT                 ;LINE BITS 1-0 FAILED TO SET
530
531   002136  104000                  GT8:  SCOPE
532   002140  012777  100005  176654        MOV   #100005,@DBUF   ;LINE TYPE ENABLE =1 LINE TYPE =1
533   002146  012777  172000  176650        MOV   #172000,@DBUF1  ;LOAD STOP
534   002154  013777  001022  176670        MOV   DBUF,@DPC       ;LOAD DISPLAY P.C.
535   002162  017700  176666               MOV   @DSR,R0         ;READ DISPLAY STATUS REGISTER
536   002166  042700  177774               BIC   #177774,R0      ;MASK TO BITS 1-0
537   002172  022700  000001               CMP   #1,R0           ;TEST R0
538   002176  001401                        BEQ   .+4            ;
539   002200  000000                        HALT                 ;LINE BIT 0 FAILED TO SET
540
541
542   002202  104000                  GT9:  SCOPE
543   002204  012777  100006  176610        MOV   #100006,@DBUF   ;LINE TYPE ENABLE =1 LINE TYPE =2
544   002212  012777  172000  176604        MOV   #172000,@DBUF1
545   002220  013777  001022  176624        MOV   DBUF,@DPC       ;LOAD DISPLAY P.C.
546   002226  017700  176622               MOV   @DSR,R0         ;READ DISPLAY STATUS REGISTER
547   002232  042700  177774               BIC   #177774,R0      ;MASK TO BITS 1-0
548   002236  022700  000002               CMP   #2,R0           ;TEST R0
549   002242  001401                        BEQ   .+4            ;
550   002244  000000                        HALT                 ;LINE BIT 1 FAILED TO SET
```

```
552
553   022246  104000                       GT801  SCOPE
554   002250  012777  100003  176544        MOV    #100003,@DBUF    ;LINE TYPE ENABLE #0 LINE TYPE #3
555   002256  012777  172000  176540        MOV    #172000,@DBUF1
556   002264  013777  001022  176560        MOV    DBUF,@DPC        ;LOAD DISPLAY P.C.
557   002272  017700  176556               MOV    @DSR,R0          ;READ DISPLAY STATUS REGISTER
558   002276  042700  177774               BIC    #177774,R0       ;MASK TO BITS 1-0
559   002302  022700  000002               CMP    #2,R0            ;TEST R0
560   002306  001401                       BEQ    .+4              ;SHOULD NOT CHANGE LT VALUE
561   002310  000000                       HALT                    ;LINE TYPE ENABLE FAILED TO INHIBIT
562                                                                 ;CHANGING OF LINETYPE VALUE
563
564   002312  104000                       GT811  SCOPE
565   002314  012777  100020  176500        MOV    #100020,@DBUF    ;BLINK ENABLE #1  BLINK #0
566   002322  012777  172000  176474        MOV    #172000,@DBUF1
567   002330  013777  001022  176514        MOV    DBUF,@DPC        ;LOAD DISPLAY P.C.
568   002336  017700  176512               MOV    @DSR,R0          ;READ DISPLAY STATUS REGISTER
569   002342  042700  177767               BIC    #177767,R0       ;MASK TO BIT 3
570   002346  022700  000000               CMP    #0,R0            ;TEST R0
571   002352  001401                       BEQ    .+4
572   002354  000000                       HALT                    ;BLINK BIT FAILED TO RESET
573
574
575   002356  104000                       GT821  SCOPE
576   002360  012777  100030  176434        MOV    #100030,@DBUF    ;BLINK ENABLE #1  BLINK #1
577   002366  012777  172000  176430        MOV    #172000,@DBUF1
578   002374  013777  001022  176450        MOV    DBUF,@DPC        ;LOAD DISPLAY P.C.
579   002402  017700  176446               MOV    @DSR,R0          ;READ DISPLAY STATUS REGISTER
580   002406  042700  177767               BIC    #177767,R0       ;MASK TO BIT 3
581   002412  022700  000010               CMP    #10,R0           ;TEST R0
582   002416  001401                       BEQ    .+4
583   002420  000000                       HALT                    ;BLINK BIT FAILED TO SET
584
585
586   002422  104000                       GT831  SCOPE
587   002424  012777  100000  176370        MOV    #100000,@DBUF    ;BLINK ENABLE #0  BLINK #0
588   002432  012777  172000  176364        MOV    #172000,@DBUF1
589   002440  013777  001022  176404        MOV    DBUF,@DPC        ;LOAD DISPLAY P.C.
590   002446  017700  176402               MOV    @DSR,R0          ;READ DISPLAY STATUS REGISTER
591   002452  042700  177767               BIC    #177767,R0       ;MASK TO BIT 3
592   002456  022700  000010               CMP    #10,R0           ;TEST R0
593   002462  001401                       BEQ    .+4
594   002464  000000                       HALT                    ;BLINK ENABLE FAILED TO INHIBIT
595                                                                 ;CHANGING OF THE BLINK BIT
596
597   002466  104000                       GT841  SCOPE
598   002470  012777  100100  176324        MOV    #100100,@DBUF    ;LP ENABLE #1 LP=0
599   002476  012777  172000  176320        MOV    #172000,@DBUF1
600   002504  013777  001022  176340        MOV    DBUF,@DPC        ;LOAD DISPLAY P.C.
601   002512  017700  176336               MOV    @DSR,R0          ;READ STATUS
602   002516  032700  000200               BIT    #200,R0          ;
603   002522  001401                       BEQ    .+4
604   002524  000000                       HALT                    ;LIGHT PEN FLAG SET IN ERROR
```

```
606
607   002526  104000                       GT851  SCOPE
608   002530  012777  100140  176264        MOV    #100140,@DBUF    ;LP ENABLE #1 LP=1
609   002536  012777  172000  176260        MOV    #172000,@DBUF1
610   002544  013777  001022  176300        MOV    DBUF,@DPC        ;LOAD DISPLAY P.C.
611   002552  017700  176276               MOV    @DSR,R0          ;READ STATUS
612   002556  032700  000200               BIT    #200,R0          ;
613   002562  001401                       BEQ    .+4
614   002564  000000                       HALT                    ;LIGHT PEN FLAG SET IN ERROR
615
616   002566  104000                       GT861  SCOPE
617   002570  012777  102000  176224        MOV    #102000,@DBUF    ;INTENSITY LEVEL ENABLE #1 LEVEL #0
618   002576  012777  172000  176220        MOV    #172000,@DBUF1
619   002604  013777  001022  176240        MOV    DBUF,@DPC        ;LOAD DISPLAY P.C.
620   002612  017700  176236               MOV    @DSR,R0          ;READ DISPLAY STATUS REGISTER
621   002616  042700  174377               BIC    #174377,R0       ;MASK TO BITS 8-10
622   002622  022700  000000               CMP    #0,R0            ;TEST R0
623   002626  001401                       BEQ    .+4
624   002630  000000                       HALT                    ;INTENSITY LEVEL BITS 8-10 FAILED TO RESET
625
626
627   002632  104000                       GT871  SCOPE
628   002634  012777  103600  176160        MOV    #103600,@DBUF    ;INTENSITY LEVEL ENABLE #1  LEVEL #7
629   002642  012777  172000  176154        MOV    #172000,@DBUF1
630   002650  013777  001022  176174        MOV    DBUF,@DPC        ;LOAD DISPLAY P.C.
631   002656  017700  176172               MOV    @DSR,R0          ;READ DISPLAY STATUS REGISTER
632   002662  042700  174377               BIC    #174377,R0       ;MASK TO BITS 8-10
633   002666  022700  003400               CMP    #3400,R0         ;TEST R0
634   002672  001401                       BEQ    .+4
635   002674  000000                       HALT                    ;INTENSITY LEVEL BITS 8-10 FAILED TO SET
636
637
638   002676  104000                       GT881  SCOPE
639   002700  012777  103000  176114        MOV    #103000,@DBUF    ;INTENSITY LEVEL ENABLE #1  LEVEL #4
640   002706  012777  172000  176110        MOV    #172000,@DBUF1
641   002714  013777  001022  176130        MOV    DBUF,@DPC        ;LOAD DISPLAY P.C.
642   002722  017700  176126               MOV    @DSR,R0          ;READ DISPLAY STATUS REGISTER
643   002726  042700  174377               BIC    #174377,R0       ;MASK TO BITS 8-10
644   002732  022700  002000               CMP    #2000,R0         ;TEST R0
645   002736  001401                       BEQ    .+4
646   002740  000000                       HALT                    ;INTENSITY LEVEL BIT 10 FAILED
647
648
649   002742  104000                       GT891  SCOPE
650   002744  012777  102400  176050        MOV    #102400,@DBUF    ;INTENSITY LEVEL ENABLE #1  LEVEL #2
651   002752  012777  172000  176044        MOV    #172000,@DBUF1
652   002760  013777  001022  176064        MOV    DBUF,@DPC        ;LOAD DISPLAY P.C.
653   002766  017700  176062               MOV    @DSR,R0          ;READ DISPLAY STATUS REGISTER
654   002772  042700  174377               BIC    #174377,R0       ;MASK TO BITS 8-10
655   002776  022700  001000               CMP    #1000,R0         ;TEST R0
656   003002  001401                       BEQ    .+4
657   003004  000000                       HALT                    ;INTENSITY LEVEL BIT 9 FAILED
```

```
659
660  003076  104000              GT20I   SCOPE
661  003010  012777   102200 176004       MOV    #102200,@DBUF   ;INTENSITY LEVEL ENABLE #1  LEVEL #1
662  003016  012777   172000 176000       MOV    #172000,@DBUF1
663  003024  013777   001022 176020       MOV    DBUF,@DPC       ;LOAD DISPLAY P.C.
664  003032  017700   176016              MOV    @DSR,R0         ;READ DISPLAY STATUS REGISTER
665  003036  042700   174377              BIC    #174377,R0      ;MASK TO BITS 8-10
666  003042  022700   000400              CMP    #400,R0         ;TEST R0
667  003046  001401                       BEQ    .+4
668  003050  000000                       HALT                   ;INTENSITY LEVEL BIT 8 FAILED
669
670
671  003052  104000              GT21I   SCOPE
672  003054  012777   101600 175740       MOV    #101600,@DBUF   ;INTENSITY LEVEL ENABLE #0  LEVEL #7
673  003062  012777   172000 175734       MOV    #172000,@DBUF1
674  003070  013777   001022 175754       MOV    DBUF,@DPC       ;LOAD DISPLAY P.C.
675  003076  017700   175752              MOV    @DSR,R0         ;READ DISPLAY STATUS REGISTER
676  003102  042700   174377              BIC    #174377,R0      ;MASK TO BITS 8-10
677  003106  022700   000400              CMP    #400,R0         ;TEST R0
678  003112  001401                       BEQ    .+4
679  003114  000000                       HALT                   ;INTENSITY LEVEL ENABLE FAILED TO INHIBIT
680                                                               ;INTENSITY LEVEL CHANGE
681
682                                      ;GRAPHPLOT INCREMENT REGISTER TEST
683
684  003116  104000              GT22I   SCOPE
685  003120  012777   174100 175674       MOV    #174100,@DBUF   ;LOAD GRAPHPLOT COUNTER
686  003126  012777   172000 175670       MOV    #172000,@DBUF1
687  003134  013777   001022 175710       MOV    DBUF,@DPC       ;START DISPLAY
688  003142  017700   175710              MOV    @XPOS,R0        ;READ INCREMENT REGISTER
689  003146  042700   001777              BIC    #1777,R0        ;MASK TO BITS 15-10
690  003152  022700   000000              CMP    #0,R0
691  003156  001401                       BEQ    .+4
692  003160  000000                       HALT                   ;GRAPHPLOT REGISTER IN ERROR
693
694  003162  104000              GT23I   SCOPE
695  003164  012777   174177 175630       MOV    #174177,@DBUF   ;LOAD GRAPHPLOT COUNTER
696  003172  012777   172000 175624       MOV    #172000,@DBUF1
697  003200  013777   001022 175644       MOV    DBUF,@DPC       ;START DISPLAY
698  003206  017700   175644              MOV    @XPOS,R0        ;READ INCREMENT REGISTER
699  003212  042700   001777              BIC    #1777,R0        ;MASK TO BITS 15-10
700  003216  022700   174000              CMP    #174000,R0
701  003222  001401                       BEQ    .+4
702  003224  000000                       HALT                   ;GRAPHPLOT REGISTER IN ERROR
703
704  003226  104000              GT24I   SCOPE
705  003230  012777   174152 175564       MOV    #174152,@DBUF   ;LOAD GRAPHPLOT COUNTER
706  003236  012777   172000 175560       MOV    #172000,@DBUF1
707  003244  013777   001022 175600       MOV    DBUF,@DPC       ;START DISPLAY
708  003252  017700   175600              MOV    @XPOS,R0        ;READ INCREMENT REGISTER
709  003256  042700   001777              BIC    #1777,R0        ;MASK TO BITS 15-10
710  003262  022700   124000              CMP    #124000,R0
711  003266  001401                       BEQ    .+4
712  003270  000000                       HALT                   ;GRAPHPLOT REGISTER IN ERROR
```

```
714
715  003272  104000              GT25I   SCOPE
716  003274  012777   174125 175520       MOV    #174125,@DBUF   ;LOAD GRAPHPLOT COUNTER
717  003302  012777   172000 175514       MOV    #172000,@DBUF1
718  003310  013777   001022 175534       MOV    DBUF,@DPC       ;START DISPLAY
719  003316  017700   175534              MOV    @XPOS,R0        ;READ INCREMENT REGISTER
720  003322  042700   001777              BIC    #1777,R0        ;MASK TO BITS 15-10
721  003326  022700   052000              CMP    #52000,R0
722  003332  001401                       BEQ    .+4
723  003334  000000                       HALT                   ;GRAPHPLOT REGISTER IN ERROR
724
725  003336  104000              GT26I   SCOPE
726  003340  012777   174100 175454       MOV    #174100,@DBUF   ;LOAD GRAPHPLOT COUNTER WITH 0
727  003346  012777   172000 175450       MOV    #172000,@DBUF1
728  003354  013777   001022 175470       MOV    DBUF,@DPC       ;START DISPLAY
729  003362  004737   012340              JSR    7,DLAY          ;EXECUTE A PROGRAM DELAY
730  003366  012777   174077 175426       MOV    #174077,@DBUF   ;LOAD GRAPHPLOT NO ENABLE
731  003374  013777   001022 175450       MOV    DBUF,@DPC       ;START DISPLAY
732  003402  017700   175450              MOV    @XPOS,R0        ;READ INCREMENT REGISTER
733  003406  042700   001777              BIC    #1777,R0        ;MASK TO BITS 15-10
734  003412  022700   000000              CMP    #0,R0           ;ARE THEY EQUAL ?
735  003436  001401                       BEQ    .+4
736  003420  000000                       HALT                   ;GRAPHPLOT REGISTER CHANGED WITHOUT
737                                                               ; THE ENABLE BEING SET
738
739                                      ;TEST THAT THE X POSITION REGISTER CAN BE LOADED CORRECTLY
740                                      ;USING GRAPHPLOT X
741
742  003422  104000              GT27I   SCOPE
743  003424  012777   122000 175370       MOV    #122000,@DBUF   ;LOW INTENSITY - SET GRAPHPLOT X MODE
744  003432  012777   001252 175364       MOV    #1252,@DBUF1    ;SET X POSITION
745  003440  012777   172000 175360       MOV    #172000,@DBUF2  ;LOAD STOP
746  003446  013777   001022 175376       MOV    DBUF,@DPC       ;START THE DISPLAY
747  003454  004737   012340              JSR    7,DLAY          ;EXECUTE A PROGRAM DELAY
748  003460  017700   175372              MOV    @XPOS,R0        ;READ X POSITION
749  003464  022700   001252              CMP    #1252,R0        ;
750  003470  001401                       BEQ    .+4
751  003472  000000                       HALT                   ;X POSITION REGISTER FAILED TO LOAD
752                                                               ;PROPERLY USING GRAPHPLOT X MODE
```

```
754
755                                          ;TEST THAT THE X POSITION REGISTER CAN BE LOADED CORRECTLY
756                                          ;USING GRAPHPLOT X
757
758   008474  104000                 GT281   SCOPE
759   008476  012777  122000  175316          MOV     #122000,.DBUF    ;LOW INTENSITY - SET GRAPH PLOT X MODE
760   008504  012777  000525  175312          MOV     #525,.DBUF1      ;SET X POSITION
761   008512  012777  172000  175306          MOV     #172000,.DBUF+2  ;LOAD STATUS REGISTER A, STOP
762   008520  013777  001022  175324          MOV     DBUF+.DPC        ;LOAD THE DISPLAY P.C.
763   008526  004737  012340                  JSR     7,DLAY           ;EXECUTE A PROGRAM DELAY
764   008532  017700  175320                  MOV     @XPOS,R0         ;READ X POSITION
765   008536  022700  000525                  CMP     #525,R0          ;
766   008542  001401                          BEQ     .+4              ;
767   008544  000000                          HALT                     ;X POSITION REGISTER FAILED TO LAD
768                                                                    ;PROPERLY USING GRAPH PLOT X MODE
769
770                                          ;TEST THAT THE Y POSITION REGISTER CAN BE LOADED CORRECTLY
771                                          ;USING GRAPHPLOT Y MODE
772
773   008546  104000                 GT891   SCOPE
774   008550  012777  126000  175244          MOV     #126000,.DBUF    ;LOW INTENSITY - SET GRAPHPLOT Y
775   008556  012777  001252  175240          MOV     #1252,.DBUF1     ;SET Y POSITION
776   008564  012777  172000  175234          MOV     #172000,.DBUF2   ;LOAD STATUS REGISTER A, STOP
777   008572  013777  001022  175252          MOV     DBUF+.DPC        ;LOAD THE DISPLAY P.C.
778   008600  004737  012340                  JSR     7,DLAY           ;EXECUTE A PROGRAM DELAY
779   008604  017700  175250                  MOV     @YPOS,R0         ;READ Y POSITION
780   008610  022700  001252                  CMP     #1252,R0         ;
781   008614  001401                          BEQ     .+4              ;
782   008616  000000                          HALT                     ;Y POSITION REGISTER FAILED TO LOAD
783                                                                    ;PROPERLY USING GRAPHPLOT Y MODE
784
```

```
786
787                                          ;TEST THAT THE Y POSITION REGISTER CAN BE LOADED CORRECTLY
788                                          ;USING GRAPHPLOT Y MODE
789
790   008620  104000                 GT381   SCOPE
791   008622  012777  126000  175172          MOV     #126000,.DBUF    ;LOW INTENSITY - SET GRAPHPLOT Y MODE
792   008630  012777  000525  175146          MOV     #525,.DBUF1      ;SET Y POSITION
793   008636  012777  172000  175162          MOV     #172000,.DBUF2   ;LOAD STATUS REGISTER A, STOP
794   008644  013777  001022  175200          MOV     DBUF+.DPC        ;LOAD THE DISPLAY P.C.
795   008652  004737  012340                  JSR     7,DLAY           ;EXECUTE A PROGRAM DELAY
796   008656  017700  175176                  MOV     @YPOS,R0         ;READ Y POSITION
797   008662  022700  000525                  CMP     #525,R0          ;
798   008666  001401                          BEQ     .+4              ;
799   008670  000000                          HALT                     ;Y POSITION REGISTER FAILED TO LOAD
800                                                                    ;PROPERLY USING GRAPHPLOT Y MODE
801
802                                          ;TEST THAT THE X + Y POSITION REGISTERS CAN BE LOADED CORRECTLY
803                                          ;USING GRAPHPLOT X + Y MODE
804                                          ;TEST FOR PROPER SELECTION OF X AND Y REGISTERS
805
806   008672  104000                 GT311   SCOPE
807   008674  012777  122000  175120          MOV     #122000,.DBUF    ;LOW INTENSITY - SET GRAPHPLOT X MODE
808   008702  012777  001234  175114          MOV     #1234,.DBUF1     ;SET X POSITION
809   008710  012777  126000  175110          MOV     #126000,.DBUF+2  ;SET GRAPHPLOT Y MODE
810   008716  012777  001432  175104          MOV     #1432,.DBUF3     ;SET Y POSITION
811   008724  012777  172000  175100          MOV     #172000,.DBUF4   ;LOAD STATUS REGISTER A, STOP
812   008732  013777  001022  175112          MOV     DBUF+.DPC        ;LOAD THE DISPLAY P.C.
813   008740  004737  012340                  JSR     7,DLAY           ;EXECUTE A PROGRAM DELAY
814   008744  017700  175106                  MOV     @XPOS,R0         ;READ X POSITION
815   008750  022700  001234                  CMP     #1234,R0         ;
816   008754  001402                          BEQ     .+6              ;
817   008756  000000                          HALT                     ;GRAPHPLOT X MODE FAILED TO SELECT
818   008760  000406                          BR      GT33             ;X POSITION PROPERLY
819
820   008762  017700  175072                  MOV     @YPOS,R0         ;READ Y POSITION
821   008766  022700  001432                  CMP     #1432,R0         ;
822   008772  001401                          BEQ     .+4              ;
823   008774  000000                          HALT                     ;Y POSITION REGISTER FAILED TO LOAD
824                                                                    ;PROPERLY USING GRAPHPLOT Y MODE
825
```

```
827
828                                      ;TEST THAT THE X-Y POSITION REGISTERS CAN BE RESET
829                                      ;USING POINT DATA MODE.
830
831    002776  104000           GT32:   SCOPE
832    004000  012777  116000  175014    MOV   #116000,@DBUF   ;LOW INTENSITY - POINT MODE
833    004006  005077  175012            CLR   @DBUF1         ;CLEAR X POSITION
834    004012  005077  175010            CLR   @DBUF2         ;CLEAR Y POSITION
835    004016  012777  172000  175004    MOV   #172000,@DBUF3  ;LOAD STATUS "A" REGISTER, STOP
836    004024  013777  001022  175020    MOV   DBUF,@DPC      ;LOAD DISPLAY P.C.
837    004032  004737  012340            JSR   7,DLAY         ;EXECUTE A PROGRAM DELAY
838    004036  017700  175014            MOV   @XPOS,R0       ;READ X POSITION
839    004042  001402                    BEQ   .+6            ;WAS IT 0?
840    004044  000000                    HALT                 ;X POSITION REGISTER FAILED TO RESET
841    004046  000404                    BR    GT33           ;USING POINT DATA MODE
842
843    004050  017700  175004            MOV   @YPOS,R0       ;READ Y POSITION
844    004054  001401                    BEQ   .+4            ;WAS IT 0?
845    004056  000000                    HALT                 ;Y POSITION REGISTER FAILED TO RESET
846                                                           ;USING POINT DATA MODE
847
848                                      ;TEST THAT THE X-Y POSITION REGISTERS CAN BE SET
849                                      ;USING POINT DATA MODE.
850
851    004060  104000           GT33:   SCOPE
852    004062  012777  116000  174732    MOV   #116000,@DBUF   ;LOW INTENSITY - POINT MODE
853    004070  012777  001777  174726    MOV   #1777,@DBUF1    ;SET X POSITION
854    004076  012777  001777  174722    MOV   #1777,@DBUF2    ;SET Y POSITION
855    004104  012777  172000  174716    MOV   #172000,@DBUF3  ;LOAD STATUS A REGISTER, STOP
856    004112  013777  001022  174732    MOV   DBUF,@DPC      ;LOAD DISPLAY P.C.
857    004120  004737  012340            JSR   7,DLAY         ;EXECUTE A PROGRAM DELAY
858    004124  017700  174726            MOV   @XPOS,R0       ;READ X POSITION
859    004130  022700  001777            CMP   #1777,R0       ;WAS IT SET?
860    004134  001402                    BEQ   .+6
861    004136  000000                    HALT                 ;X POSITION REGISTER FAILED TO SET
862    004140  000404                    BR    GT34           ;USING POINT DATA MODE
863
864    004142  017700  174712            MOV   @YPOS,R0       ;READ Y POSITION
865    004146  022700  001777            CMP   #1777,R0       ;WAS IT SET?
866    004152  001401                    BEQ   .+4
867    004154  000000                    HALT                 ;Y POSITION REGISTER FAILED TO SET
868                                                           ;USING POINT DATA MODE
869
```

```
871
872                                      ;TEST THAT THE X-Y POSITION REGISTERS CAN BE LOADED CORRECTLY
873                                      ;USING POINT DATA MODE
874
875    004156  104000           GT34:   SCOPE
876    004160  012777  116000  174634    MOV   #116000,@DBUF   ;LOW INTENSITY - POINT MODE
877    004166  012777  001252  174630    MOV   #1252,@DBUF1    ;SET X POSITION
878    004174  012777  001252  174624    MOV   #1252,@DBUF2    ;SET Y POSITION
879    004202  012777  172000  174620    MOV   #172000,@DBUF3  ;LOAD STATUS REGISTER A, STOP
880    004210  013777  001022  174634    MOV   DBUF,@DPC      ;
881    004216  004737  012340            JSR   7,DLAY         ;EXECUTE A PROGRAM DELAY
882    004222  017700  174630            MOV   @XPOS,R0       ;READ X POSITION
883    004226  022700  001252            CMP   #1252,R0       ;
884    004232  001402                    BEQ   .+6
885    004234  000000                    HALT                 ;X POSITION REGISTER FAILED
886    004236  000406                    BR    GT35           ;USING POINT DATA MODE
887
888    004240  017700  174614            MOV   @YPOS,R0       ;READ Y POSITION
889    004244  022700  001252            CMP   #1252,R0       ;
890    004250  001401                    BEQ   .+4
891    004252  000000                    HALT                 ;Y POSITION REGISTER FAILED
892                                                           ;USING POINT DATA MODE
893
894                                      ;TEST THAT THE X-Y POSITION REGISTERS CAN BE LOADED CORRECTLY
895                                      ;USING POINT DATA MODE
896
897    004254  104000           GT35:   SCOPE
898    004256  012777  116000  174536    MOV   #116000,@DBUF   ;LOW INTENSITY - POINT MODE
899    004264  012777  000525  174532    MOV   #525,@DBUF1     ;SET X POSITION
900    004272  012777  000525  174526    MOV   #525,@DBUF2     ;SET Y POSITION
901    004300  012777  172000  174522    MOV   #172000,@DBUF3  ;LOAD STATUS REGISTER A, STOP
902    004306  013777  001022  174536    MOV   DBUF,@DPC      ;
903    004314  004737  012340            JSR   7,DLAY         ;EXECUTE A PROGRAM DELAY
904    004320  017700  174532            MOV   @XPOS,R0       ;READ X POSITION
905    004324  022700  000525            CMP   #525,R0        ;
906    004330  001402                    BEQ   .+6
907    004332  000000                    HALT                 ;X POSITION REGISTER FAILED
908    004334  000406                    BR    GT36           ;USING POINT DATA MODE
909
910    004336  017700  174516            MOV   @YPOS,R0       ;READ Y POSITION
911    004342  022700  000525            CMP   #525,R0        ;
912    004346  001401                    BEQ   .+4
913    004350  000000                    HALT                 ;Y POSITION REGISTER FAILED
914                                                           ;USING POINT DATA MODE
915
```

```
917                                    ;TEST THAT THE X-Y POSITION REGISTERS CAN BE LOADED CORRECTLY
918                                    ;USING POINT DATA MODE
919
920
921  004352  104000          GT361   SCOPE
922  004354  012777  116000  174440          MOV    #110000,@DBUP    ;LOW INTENSITY - POINT MODE
923  004362  012777  000000  174434          MOV    #0,@DBUF1        ;SET X POSITION
924  004370  012777  001777  174430          MOV    #1777,@DBUF2     ;SET Y POSITION
925  004376  012777  172000  174424          MOV    #172000,@DBUF3   ;LOAD STATUS REGISTER A, STOP
926  004404  013777  001022  174440          MOV    DBUF,@DPC
927  004412  004737  012340                  JSR    7,DLAY           ;EXECUTE A PROGRAM DELAY
928  004416  017700  174434                  MOV    @XPOS,R0         ;READ X POSITION
929  004422  022700  000000                  CMP    #0,R0
930  004426  001402                          BEQ    .+4
931  004430  000000                          HALT                    ;X POSITION REGISTER FAILED
932  004432  000406                          BR     GT37             ;USING POINT DATA MODE
933
934  004434  017700  174420                  MOV    @YPOS,R0         ;READ Y POSITION
935  004440  022700  001777                  CMP    #1777,R0
936  004444  001401                          BEQ    .+4
937  004446  000000                          HALT                    ;Y POSITION REGISTER FAILED
938                                                                  ;USING POINT DATA MODE
939
```

```
941                                    ;TEST THAT LONG VECTOR MODE INCREMENTS X AND Y AXIS PROPERLY
942                                    ;COUNT 1
943
944
945  004450  104000          GT371   SCOPE
946  004452  013700  001022          MOV    DBUF,R0
947  004456  012720  116000          MOV    #110000,(R0)+    ;LOAD "POINT MODE"
948  004462  005020                  CLR    (R0)+            ;CLEAR X AXIS
949  004464  005020                  CLR    (R0)+            ;CLEAR Y AXIS
950  004466  012720  110000          MOV    #110000,(R0)+    ;LOAD "LONG VECTOR MODE"
951  004472  012720  000001          MOV    #1,(R0)+         ;PRESET "DELTA X AXIS"
952  004476  012720  000001          MOV    #1,(R0)+         ;PRESET "DELTA Y AXIS"
953  004502  012710  172000          MOV    #172000,(R0)     ;LOAD "DISPLAY STOP"
954  004506  013777  001022  174336          MOV    DBUF,@DPC        ;LOAD THE DISPLAY P.C.
955  004514  004737  012340                  JSR    7,DLAY           ;EXECUTE A PROGRAM DELAY
956
957  004520  017700  174332                  MOV    @XPOS,R0         ;READ X AXIS
958  004524  022700  000001                  CMP    #1,R0            ;DID IT INCREMENT BY 1
959  004530  001402                          BEQ    .+4              ;YES
960  004532  000000                          HALT                    ;NO, INCREMENT X AXIS BY
961  004534  000406                          BR     GT38             ;LONG VECTOR MODE FAILED
962
963  004536  017700  174316                  MOV    @YPOS,R0         ;READ Y AXIS
964  004542  022700  000001                  CMP    #1,R0            ;DID IT INCREMENT BY 1
965  004546  001401                          BEQ    .+4              ;YES
966  004550  000000                          HALT                    ;NO, INCREMENT Y AXIS BY
967                                                                  ;LONG VECTOR MODE FAILED
968
969                                    ;TEST THAT LONG VECTOR MODE DECREMENT X AND Y AXIS PROPERLY
970                                    ;COUNT 1
971
972  004552  104000          GT381   SCOPE
973  004554  013700  001022          MOV    DBUF,R0
974  004560  012720  116000          MOV    #110000,(R0)+    ;LOAD "POINT MODE"
975  004564  005020                  CLR    (R0)+            ;CLEAR X AXIS
976  004566  005020                  CLR    (R0)+            ;CLEAR Y AXIS
977  004570  012720  110000          MOV    #110000,(R0)+    ;LOAD "LONG VECTOR MODE"
978  004574  012720  020001          MOV    #20001,(R0)+     ;PRESET "DELTA X AXIS"
979  004600  012720  020001          MOV    #20001,(R0)+     ;PRESET "DELTA Y AXIS"
980  004604  012710  172000          MOV    #172000,(R0)     ;LOAD "DISPLAY STOP"
981  004610  013777  001022  174234          MOV    DBUF,@DPC        ;LOAD THE DISPLAY P.C.
982  004616  004737  012340                  JSR    7,DLAY           ;EXECUTE A PROGRAM DELAY
983
984  004622  017700  174230                  MOV    @XPOS,R0         ;READ X AXIS
985  004626  022700  001777                  CMP    #1777,R0         ;DID IT DECREMENT BY 1
986  004632  001402                          BEQ    .+4              ;YES
987  004634  000000                          HALT                    ;NO, DECREMENT X AXIS BY
988  004636  000406                          BR     GT39             ;LONG VECTOR MODE FAILED
989
990  004640  017700  174214                  MOV    @YPOS,R0         ;READ Y AXIS
991  004644  022700  001777                  CMP    #1777,R0         ;DID IT DECREMENT BY 1
992  004650  001401                          BEQ    .+4              ;YES
993  004652  000000                          HALT                    ;NO, DECREMENT Y AXIS BY
994                                                                  ;LONG VECTOR MODE FAILED
```

```
 996
 997                                    ;TEST THAT LONG VECTOR MODE INCREMENT X AND Y AXIS PROPERLY
 998                                    ;COUNT 0-1777
 999
1000  004654  104000          GT391   SCOPE
1001  004656  012703  001777          MOV     #1777,R3        ;SET UP A COUNTER
1002  004662  012704  000001          MOV     #1,R4           ;PRESET THE COMPARED VALUE
1003
1004  004666  104000                  SCOPE
1005  004670  013700  001022  GT39A:  MOV     DBUF,R0         ;SET UP R0
1006  004674  012720  116000          MOV     #116000,(R0)+   ;LOAD "POINT MODE"
1007  004700  005020                  CLR     (R0)+           ;CLEAR X AXIS
1008  004702  005020                  CLR     (R0)+           ;CLEAR Y AXIS
1009  004704  012720  110000          MOV     #110000,(R0)+   ;LOAD "LONG VECTOR MODE"
1010  004710  010420                  MOV     R4,(R0)+        ;PRESET "DELTA X AXIS"
1011  004712  010420                  MOV     R4,(R0)+        ;PRESET "DELTA Y AXIS"
1012  004714  012720  172000          MOV     #172000,(R0)+
1013  004720  013777  001022  174124  MOV     DBUF,@#DPC      ;LOAD THE DISPLAY P.C.
1014  004726  004737  012340          JSR     7,DLAY          ;EXECUTE A PROGRAM DELAY
1015
1016  004732  017700  174120          MOV     @#XPOS,R0       ;READ X AXIS
1017  004736  020400                  CMP     R4,R0           ;ARE THEY EQUAL?
1018  004740  001402                  BEQ     .+6             ;YES
1019  004742  000000                  HALT                    ;NO, INCREMENT X AXIS VIA
1020  004744  000411                  BR      GT40            ;LONG VECTOR MODE FAILED
1021
1022  004746  017700  174106          MOV     @#YPOS,R0       ;READ Y AXIS
1023  004752  020400                  CMP     R4,R0           ;ARE THEY EQUAL?
1024  004754  001402                  BEQ     .+6             ;YES
1025  004756  000000                  HALT                    ;NO, INCREMENT Y AXIS VIA
1026  004760  000403                  BR      GT40            ;LONG VECTOR MODE FAILED
1027
1028  004762  005204                  INC     R4              ;INCREMENT EXPECTED VALUE
1029  004764  005303                  DEC     R3              ;FINISHED?
1030  004766  001340                  BNE     GT39A           ;NO, TEST MORE DATA
```

```
1032
1033
1034                                   ;TEST THAT LONG VECTOR MODE DECREMENTS X AND Y AXIS PROPERLY
1035                                   ;COUNT 1777-0
1036  004770  104000          GT401   SCOPE
1037  004772  012703  002000          MOV     #2000,R3        ;SET UP A COUNTER
1038  004776  012704  001777          MOV     #1777,R4        ;PRESET THE COMPARED VALUE
1039  005002  012705  020001          MOV     #20001,R5
1040
1041  005006  104000                  SCOPE
1042  005010  013700  001022  GT40A:  MOV     DBUF,R0         ;SET UP R0
1043  005014  012720  116000          MOV     #116000,(R0)+   ;LOAD "POINT MODE"
1044  005020  005020                  CLR     (R0)+           ;CLEAR X AXIS
1045  005022  005020                  CLR     (R0)+           ;CLEAR Y AXIS
1046  005024  012720  110000          MOV     #110000,(R0)+   ;LOAD "LONG VECTOR MODE"
1047  005030  010520                  MOV     R5,(R0)+        ;PRESET "DELTA X AXIS"
1048  005032  010520                  MOV     R5,(R0)+        ;PRESET "DELTA Y AXIS"
1049  005034  012710  172000          MOV     #172000,(R0)
1050  005040  013777  001022  174004  MOV     DBUF,@#DPC      ;LOAD THE DISPLAY P.C.
1051  005046  004737  012340          JSR     7,DLAY          ;EXECUTE A PROGRAM DELAY
1052
1053  005052  017700  174000          MOV     @#XPOS,R0       ;READ X AXIS
1054  005056  020400                  CMP     R4,R0           ;ARE THEY EQUAL?
1055  005060  001402                  BEQ     .+6             ;YES
1056  005062  000000                  HALT                    ;NO, DECREMENT X AXIS VIA
1057  005064  000412                  BR      GT41            ;LONG VECTOR MODE FAILED
1058
1059  005066  017700  173766          MOV     @#YPOS,R0       ;READ Y AXIS
1060  005072  020400                  CMP     R4,R0           ;ARE THEY EQUAL?
1061  005074  001402                  BEQ     .+6             ;YES
1062  005076  000000                  HALT                    ;NO, DECREMENT Y AXIS VIA
1063  005100  000404                  BR      GT41            ;LONG VECTOR MODE FAILED
1064
1065  005102  005205                  INC     R5              ;INCREMENT "DELTA X-Y"
1066  005104  005304                  DEC     R4              ;DECREMENT EXPECTED VALUE
1067  005106  005303                  DEC     R3              ;FINISHED?
1068  005110  001337                  BNE     GT40A           ;NO, TEST MORE DATA
```

```
1070                                    ;TEST THAT X AND Y AXIS INCREMENTS PROPERLY
1071                                    ;USING SHORT VECTOR MODE
1072                                    ;COUNT 1
1073
1074    005112  104000          GT41:   SCOPE
1075    005114  013700  001022          MOV     DBUF,R0         ;SET UP R0
1076    005120  012720  116000          MOV     #116000,(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     #104000,(0)+     ;LOAD "SET SHORT VECTOR MODE"
1080    005134  012720  000201          MOV     #201,(0)+        ;PRESET "DELTA X AND DELTA Y"
1081    005140  012710  172000          MOV     #172000,(0)      ;LOAD THE DISPLAY PC
1082    005144  013777  001022  173700  MOV     DBUF,@DPC
1083    005152  004737  012340          JSR     7,DLAY          ;EXECUTE A PROGRAM DELAY
1084
1085    005156  017700  173674          MOV     @XPOS,R0        ;READ X AXIS
1086    005162  022700  000001          CMP     #1,R0           ;ARE THEY EQUAL?
1087    005166  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  173660          MOV     @YPOS,R0        ;READ Y AXIS
1092    005200  022700  000001          CMP     #1,R0           ;ARE THEY EQUAL?
1093    005204  001401                  BEQ     .+4             ;YES
1094    005206  000000                  HALT                    ;NO INCREMENT Y AXIS FAILED
1095                                                            ;USING SHORT VECTOR MODE
1096
1097                                    ;TEST THAT X AND Y AXIS DECREMENT PROPERLY
1098                                    ;USING SHORT VECTOR MODE
1099                                    ;COUNT 1
1100
1101    005210  104000          GT42:   SCOPE
1102    005212  013700  001022          MOV     DBUF,R0         ;SET UP R0
1103    005216  012720  116000          MOV     #116000,(0)+     ;LOAD "SET POINT MODE"
1104    005222  005020                  CLR     (0)+            ;CLEAR X AXIS
1105    005224  005020                  CLR     (0)+            ;CLEAR Y AXIS
1106    005226  012720  106000          MOV     #106000,(0)+     ;LOAD "SET SHORT VECTOR MODE"
1107    005232  012720  020301          MOV     #20301,(0)+      ;PRESET "DELTA X AND DELTA Y"
1108    005236  012710  172000          MOV     #172000,(0)      ;
1109    005242  013777  001022  173602  MOV     DBUF,@DPC       ;LOAD THE DISPLAY PC
1110    005250  004737  012340          JSR     7,DLAY          ;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     .+4             ;YES
1121    005304  000000                  HALT                    ;NO DECREMENT Y AXIS FAILED
1122                                                            ;USING SHORT VECTOR MODE
```

```
1124                                    ;TEST THAT X AND Y AXIS INCREMENT PROPERLY
1125                                    ;USING SHORT VECTOR MODE
1126                                    ;COUNT 0-77
1127
1128
1129    005306  104000          GT43:   SCOPE
1130    005310  012703  000077          MOV     #77,R3          ;SET UP A COUNT LOCATION
1131    005314  012702  000001          MOV     #1,R2           ;SET UP THE COMPARED LOCATION
1132    005320  012704  000201          MOV     #201,R4         ;SET UP "DELTA X+Y"
1133
1134    005324  104000                  SCOPE
1135    005326  013700  001022  GT43A:  MOV     DBUF,R0         ;SET UP R0
1136    005332  012720  116000          MOV     #116000,(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  106000          MOV     #106000,(0)+     ;LOAD "SET SHORT VECTOR MODE"
1140    005346  010420                  MOV     R4,(0)+         ;PRESET "DELTA X AND DELTA Y"
1141    005350  012710  172000          MOV     #172000,(0)      ;
1142    005354  013777  001022  173470  MOV     DBUF,@DPC       ;LOAD THE DISPLAY P.C.
1143    005362  004737  012340          JSR     7,DLAY          ;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    005412  001402                  BEQ     .+6             ;YES
1154    005412  000000                  HALT                    ;INCREMENT Y AXIS FAILED USING
1155    005414  000405                  BR      GT44            ;SHORT VECTOR MODE
1156
1157    005416  062704  000201          ADD     #201,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                                              ;TEST THAT X AND Y AXIS DECREMENT PROPERLY
1164                                              ;USING SHORT VECTOR MODE
1165                                              ;COUNT 77-0
1166
1167  005430  104000              GT441    SCOPE
1168  005432  012703  000077               MOV     #77,R3          ;SET UP A COUNT LOCATION
1169  005436  012702  001777               MOV     #1777,R2        ;SET UP THE COMPARED LOCATION
1170  005442  012704  020301               MOV     #20301,R4       ;PRESET THE "DELTA X.Y"
1171
1172  005446  104000                       SCOPE
1173  005450  013700  001022      GT44A:   MOV     DBUF,R0         ;SET UP R0
1174  005454  012720  116000               MOV     #116000,(0)+    ;LOAD "SET POINT DATA MODE"
1175  005460  005020                        CLR     (0)+            ;CLEAR X AXIS
1176  005462  005020                        CLR     (0)+            ;CLEAR Y AXIS
1177  005464  012720  106000               MOV     #106000,(0)+    ;LOAD "SET SHORT VECTOR MODE"
1178  005470  010420                        MOV     R4,(0)+         ;PRESET "DELTA X AND DELTA Y"
1179  005472  012710  172000               MOV     #172000,(0)
1180  005476  013777  001022  173346        MOV     DBUF,@#PC       ;LOAD THE DISPLAY P.C.
1181  005504  004737  012340               JSR     7,DLAY          ;EXECUTE A PROGRAM DELAY
1182
1183  005510  017700  173342               MOV     @XPOS,R0        ;READ X POSITION
1184  005514  020200                        CMP     R2,R0           ;ARE THEY EQUAL
1185  005516  001402                        BEQ     .+6             ;YES
1186  005520  000000                        HALT                    ;DECREMENT X AXIS FAILED USING
1187  005522  000413                        BR      GT48            ;SHORT VECTOR MODE
1188
1189  005524  017700  173330               MOV     @YPOS, R0       ;READ Y POSITION
1190  005530  020200                        CMP     R2,R0           ;ARE THEY EQUAL ?
1191  005532  001402                        BEQ     .+6             ;YES DECREMENT
1192  005534  000000                        HALT                    ;DECREMENT Y AXIS FAILED USING
1193  005536  000405                        BR      GT48            ;SHORT VECTOR MODE
1194
1195  005540  062704  000201               ADD     #201,R4         ;ADD "DELTA X-Y"
1196  005544  005302                        DEC     R2              ;DECREMENT EXPECTED VALUE
1197  005546  005303                        DEC     R3              ;DECREMENT COUNT, FINISHED?
1198  005550  001337                        BNE     GT44A           ;NO, TEST MORE DATA
1199
```

```
1201                                              ;TEST THAT X AND Y AXIS INCREMENTS PROPERLY
1202                                              ;USING RELATIVE POINT MODE
1203                                              ;COUNT 1
1204
1205  005552  104000              GT451    SCOPE
1206  005554  013700  001022               MOV     DBUF,R0         ;SET UP R0
1207  005560  012720  116000               MOV     #116000,(0)+    ;LOAD "SET POINT MODE"
1208  005564  005020                        CLR     (0)+            ;CLEAR X AXIS
1209  005566  005020                        CLR     (0)+            ;CLEAR Y AXIS
1210  005570  012720  130000               MOV     #130000,(0)+    ;LOAD "SET RELATIVE POINT MODE"
1211  005574  012720  000201               MOV     #201,(0)+       ;PRESET "DELTA X AND DELTA Y"
1212  005600  012710  172000               MOV     #172000,(0)
1213  005604  013777  001022  173240        MOV     DBUF,@#PC       ;LOAD THE DISPLAY PC
1214  005612  004737  012340               JSR     7,DLAY          ;EXECUTE A PROGRAM DELAY
1215
1216  005616  017700  173234               MOV     @XPOS,R0        ;READ X AXIS
1217  005622  022700  000001               CMP     #1,R0           ;ARE THEY EQUAL?
1218  005626  001402                        BEQ     .+6             ;YES
1219  005630  000000                        HALT                    ;NO, INCREMENT X AXIS FAILED USING
1220  005632  000406                        BR      GT44            ;RELATIVE POINT MODE
1221
1222  005634  017700  173220               MOV     @YPOS,R0        ;READ Y AXIS
1223  005640  022700  000001               CMP     #1,R0           ;ARE THEY EQUAL?
1224  005644  001401                        BEQ     .+4             ;YES
1225  005646  000000                        HALT                    ;NO INCREMENT Y AXIS FAILED
1226                                                                 ;USING RELATIVE POINT MODE
1227
1228                                              ;TEST THAT X AND Y AXIS DECREMENT PROPERLY
1229                                              ;USING RELATIVE POINT MODE
1230                                              ;COUNT 1
1231
1232  005650  104000              GT461    SCOPE
1233  005652  013700  001022               MOV     DBUF,R0         ;SET UP R0
1234  005656  012720  116000               MOV     #116000,(0)+    ;LOAD "SET POINT MODE"
1235  005662  005020                        CLR     (0)+            ;CLEAR X AXIS
1236  005664  005020                        CLR     (0)+            ;CLEAR Y AXIS
1237  005666  012720  130000               MOV     #130000,(0)+    ;LOAD "SET RELATIVE POINT MODE"
1238  005672  012720  020301               MOV     #20301,(0)+     ;PRESET "DELTA X AND DELTA Y"
1239  005676  012710  172000               MOV     #172000,(0)
1240  005702  013777  001022  173142        MOV     DBUF,@#PC       ;LOAD THE DISPLAY PC
1241  005710  004737  012340               JSR     7,DLAY          ;EXECUTE A PROGRAM DELAY
1242
1243  005714  017700  173136               MOV     @XPOS,R0        ;READ X AXIS
1244  005720  022700  001777               CMP     #1777,R0        ;ARE THEY EQUAL?
1245  005724  001402                        BEQ     .+6             ;YES
1246  005726  000000                        HALT                    ;NO, DECREMENT X AXIS FAILED USING
1247  005730  000406                        BR      GT47            ;RELATIVE POINT MODE
1248
1249  005732  017700  173122               MOV     @YPOS,R0        ;READ Y AXIS
1250  005736  022700  001777               CMP     #1777,R0        ;ARE THEY EQUAL?
1251  005742  001401                        BEQ     .+4             ;YES
1252  005744  000000                        HALT                    ;NO DECREMENT Y AXIS FAILED
1253                                                                 ;USING RELATIVE POINT MODE
```

```
1255                                        ;TEST THAT X AND Y AXIS INCREMENT PROPERLY
1256                                        ;USING RELATIVE POINT MODE
1257                                        ;COUNT 0-77
1258
1259
1260    005746  104000           GT471    SCOPE
1261    005750  012703  000077            MOV     #77,R3          ;SET UP A COUNT LOCATION
1262    005754  012702  000001            MOV     #1,R2           ;SET UP THE COMPARED LOCATION
1263    005760  012704  000201            MOV     #201,R4         ;SET UP "DELTA X-Y"
1264
1265    005764  104000            SCOPE
1266    005766  013700  001022   GT47A:   MOV     DBUF,R0         ;SET UP R0
1267    005772  012720  116000            MOV     #116000,(0)+    ;LOAD "SET POINT DATA MODE"
1268    005776  005020            CLR     (0)+            ;CLEAR X AXIS
1269    006000  005020            CLR     (0)+            ;CLEAR Y AXIS
1270    006002  012720  130000            MOV     #130000,(0)+    ;LOAD "SET RELATIVE POINT MODE"
1271    006006  010420            MOV     R4,(0)+         ;PRESET "DELTA X AND DELTA Y"
1272    006010  012710  172000            MOV     #172000,(0)     ;LOAD THE DISPLAY P.C.
1273    006014  013777  001022  173030    MOV     DBUF,@#DPC      ;EXECUTE A PROGRAM DELAY
1274    006022  004737  012340            JSR     7,DLAY
1275
1276    006026  017700  173024            MOV     @XPOS,R0        ;READ X POSITION
1277    006032  020200            CMP     R2,R0           ;ARE THEY EQUAL
1278    006034  001402            BEQ     .+6             ;YES
1279    006036  000000            HALT                    ;INCREMENT X AXIS FAILED USING
1280    006040  000413            BR      GT40            ;RELATIVE POINT MODE
1281
1282    006042  017700  173012            MOV     @YPOS,R0        ;READ Y POSITION
1283    006046  020200            CMP     R2,R0           ;ARE THEY EQUAL ?
1284    006050  001402            BEQ     .+6             ;YES
1285    006052  000000            HALT                    ;INCREMENT Y AXIS FAILED USING
1286    006054  000405            BR      GT40            ;RELATIVE POINT MODE
1287
1288    006056  062704  000201            ADD     #201,R4         ;ADD DELTA X-Y
1289    006062  005202            INC     R2              ;INCREMENT EXPECTED VALUE
1290    006064  005303            DEC     R3              ;DECREMENT COUNT, FINISHED?
1291    006066  001337            BNE     GT47A           ;NO, TEST MORE DATA
```

```
1293                                       ;TEST THAT X AND Y AXIS DECREMENT PROPERLY
1294                                       ;USING RELATIVE POINT MODE
1295                                       ;COUNT 77-0
1296
1297
1298    006070  104000           GT48:    SCOPE
1299    006072  012703  000077            MOV     #77,R3          ;SET UP A COUNT LOCATION
1300    006076  012702  001777            MOV     #1777,R2        ;SET UP THE COMPARED LOCATION
1301    006102  012704  020301            MOV     #20301,R4       ;PRESET THE "DELTA X-Y"
1302
1303    006106  104000            SCOPE
1304    006110  013700  001022   GT48A:   MOV     DBUF,R0         ;SET UP R0
1305    006114  012720  116000            MOV     #116000,(0)+    ;LOAD "SET POINT DATA MODE"
1306    006120  005020            CLR     (0)+            ;CLEAR X AXIS
1307    006122  005020            CLR     (0)+            ;CLEAR Y AXIS
1308    006124  012720  130000            MOV     #130000,(0)+    ;LOAD "SET RELATIVE POINT MODE"
1309    006130  010420            MOV     R4,(0)+         ;PRESET "DELTA X AND DELTA Y"
1310    006132  012710  172000            MOV     #172000,(0)     ;LOAD THE DISPLAY P.C.
1311    006136  013777  001022  172706    MOV     DBUF,@#DPC      ;EXECUTE A PROGRAM DELAY
1312    006144  004737  012340            JSR     7,DLAY
1313
1314    006150  017700  172702            MOV     @XPOS,R0        ;READ X POSITION
1315    006154  020200            CMP     R2,R0           ;ARE THEY EQUAL
1316    006156  001402            BEQ     .+6             ;YES
1317    006160  000000            HALT                    ;DECREMENT X AXIS FAILED USING
1318    006162  000413            BR      GT40            ;RELATIVE POINT MODE
1319
1320    006164  017700  172670            MOV     @YPOS,R0        ;READ Y POSITION
1321    006170  020200            CMP     R2,R0           ;ARE THEY EQUAL ?
1322    006172  001402            BEQ     .+6             ;YES DECREMENT
1323    006174  000000            HALT                    ;DECREMENT Y AXIS FAILED USING
1324    006176  000405            BR      GT40            ;RELATIVE POINT MODE
1325
1326    006200  062704  000201            ADD     #201,R4         ;ADD "DELTA X-Y"
1327    006204  005302            DEC     R2              ;DECREMENT EXPECTED VALUE
1328    006206  005303            DEC     R3              ;DECREMENT COUNT, FINISHED?
1329    006210  001337            BNE     GT48A           ;NO, TEST MORE DATA
1330
```

```
1332
1333
1334                                    ;LOAD STATUS B TEST
1335                                    ;USE GRAPHPLOT X MODE TO TEST Y AXIS IS INCREMENTED BY
1336                                    ;"SCALE" REGISTER
1337
1338  001212  104000              GT49:  SCOPE
1339  001214  012703  000077            MOV     #77,R3            ;SET UP EXECUTION COUNTER
1340  001220  012704  000001            MOV     #1,R4             ;SET UP COMPARED DATA
1341  001224  012737  174101  001036     MOV     #174101,DSAVE    ;SET UP BASIC "LOAD STATUS B"
1342
1343  001232  104000              SCOPE
1344  001234  013700  001022     GT49A:  MOV     DBUF,R0          ;SET UP R0
1345  001240  012720  116000            MOV     #116000,(0)+     ;LOAD "POINT MODE"
1346  001244  005020                    CLR     (0)+             ;CLEAR X AXIS
1347  001246  005020                    CLR     (0)+             ;CLEAR Y AXIS
1348  001250  013720  001036            MOV     DSAVE,(0)+       ;LOAD "SET STATUS B"
1349  001254  012720  120000            MOV     #120000,(0)+     ;LOAD "SET GRAPHPLOT X MODE"
1350  001260  005020                    CLR     (0)+             ;LOAD "X GRAPHPLOT DATA"
1351  001262  012710  172000            MOV     #172000,(0)
1352  001266  013777  001022  172556     MOV    DBUF,@#DPC       ;LOAD THE DISPLAY P.C.
1353  001274  004737  012340            JSR     7,DLAY           ;EXECUTE A PROGRAM DELAY
1354
1355  001320  017700  172554            MOV     @YRDS,R0         ;READ Y AXIS
1356  001304  020400                    CMP     R4,R0            ;COMPARE TO EXPECTED VALUE
1357  001306  001402                    BEQ     .+6              ;ARE THEY EQUAL?
1358  001310  000000                    HALT                     ;LOAD "STATUS B" FAILED TO LOAD
1359  001312  000405                    BR      GT50             ;THE Y AXIS CORRECTLY
1360
1361  001314  005237  001036            INC     DSAVE            ;INCREMENT THE STATUS B COUNT
1362  001320  005204                    INC     R4
1363  001322  005303                    DEC     R3               ;DECREMENT THE EXECUTION COUNT
1364  001324  001343                    BNE     GT49A            ;TEST MORE DATA
```

```
1366
1367                                    ;LOAD STATUS B TEST
1368                                    ;USE GRAPHPLOT Y MODE TO TEST X AXIS IS INCREMENTED BY
1369                                    ;"SCALE" REGISTER
1370
1371  001326  104000              GT50:  SCOPE
1372  001330  012703  000077            MOV     #77,R3            ;SET UP EXECUTION COUNTER
1373  001334  012704  000001            MOV     #1,R4             ;SET UP COMPARED DATA
1374  001340  012737  174101  001036     MOV     #174101,DSAVE    ;SET UP BASIC "LOAD STATUS B"
1375
1376  001346  104000              SCOPE
1377  001350  013700  001022     GT50A:  MOV     DBUF,R0          ;SET UP R0
1378  001354  012720  116000            MOV     #116000,(0)+     ;LOAD "POINT MODE"
1379  001360  005020                    CLR     (0)+             ;CLEAR X AXIS
1380  001362  005020                    CLR     (0)+             ;CLEAR Y AXIS
1381  001364  013720  001036            MOV     DSAVE,(0)+       ;LOAD "SET STATUS B"
1382  001370  012720  124000            MOV     #124000,(0)+     ;LOAD "SET GRAPHPLOT Y MODE"
1383  001374  005020                    CLR     (0)+             ;LOAD "Y GRAPHPLOT DATA"
1384  001376  012710  172000            MOV     #172000,(0)
1385  001402  013777  001022  172442     MOV    DBUF,@#DPC       ;LOAD THE DISPLAY P.C.
1386  001410  004737  012340            JSR     7,DLAY           ;EXECUTE A PROGRAM DELAY
1387
1388  001414  017700  172436            MOV     @XRDS,R0         ;READ X AXIS
1389  001420  042700  176000            BIC     #176000,R0       ;MASK TO BITS 0-9
1390  001424  020400                    CMP     R4,R0            ;COMPARE TO EXPECTED VALUE
1391  001426  001402                    BEQ     .+6              ;ARE THEY EQUAL?
1392  001430  000000                    HALT                     ;LOAD "STATUS B" FAILED TO LOAD
1393  001432  000405                    BR      GT50B            ;THE X AXIS CORRECTLY
1394
1395  001434  005237  001036            INC     DSAVE            ;INCREMENT THE STATUS B COUNT
1396  001440  005204                    INC     R4
1397  001442  005303                    DEC     R3               ;DECREMENT THE EXECUTION COUNT
1398  001444  001341                    BNE     GT50A            ;TEST MORE DATA
1399
1400  001446  012777  174100  172346 GT50B: MOV  #174100,@DBUF
1401  001454  012777  172000  172342     MOV    #172000,@DBUF1
1402  001462  013777  001022  172362     MOV    DBUF,@#DPC
```

```
1404                              ;TEST THAT THE EDGE FLAG IS NOT SET AT 1777,0
1405
1406   006470   104000             GT51:   SCOPE
1407   006472   013700  001022             MOV    DBUF,R0
1408   006476   012720  116000             MOV    #116000,(0)+      ;LOAD POINT
1409   006502   012720  001777             MOV    #1777,(0)+        ;LOAD X
1410   006506   012720  000000             MOV    #0,(0)+           ;LOAD Y
1411   006512   012720  172000             MOV    #172000,(0)+      ;LOAD STOP
1412   006516   013777  001022  172326      MOV    DBUF,@#DPC        ;START
1413   006524   004737  012340             JSR    7,DLAY            ;EXECUTE A PROGRAM DELAY
1414
1415   006530   032777  000040  172316      BIT    #40,@DSR
1416   006536   001401                      BEQ    .+4
1417   006540   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   006542   104000             GT52:   SCOPE
1423   006544   013700  001022             MOV    DBUF,R0
1424   006550   012720  116000             MOV    #116000,(0)+      ;LOAD POINT
1425   006554   012720  001777             MOV    #1777,(0)+        ;LOAD MAX X
1426   006560   012720  000000             MOV    #0,(0)+           ;LOAD Y
1427   006564   012720  110000             MOV    #110000,(0)+      ;LOAD LONG VECTOR
1428   006570   012720  000001             MOV    #1,(0)+           ;LOAD DELTA X
1429   006574   012720  000000             MOV    #0,(0)+           ;LOAD DELTA Y
1430   006600   012720  172000             MOV    #172000,(0)+      ;LOAD STOP
1431   006604   013777  001022  172240      MOV    DBUF,@#DPC        ;START DISPLAY
1432   006612   004737  012340             JSR    7,DLAY            ;EXECUTE A PROGRAM DELAY
1433
1434   006616   032777  000040  172230      BIT    #40,@DSR          ;TEST BIT 5
1435   006624   001002                      BNE    .+6
1436   006626   000000                      HALT                     ;EDGE FLAG FAILED TO SET
1437   006630   000424                      BR     GT53
1438
1439
1440                              ;SUB-TEST, TEST THAT THE EDGE FLAG  CLEARS
1441
1442   006632   013700  001022             MOV    DBUF,R0
1443   006636   012720  116000             MOV    #116000,(0)+      ;LOAD POINT
1444   006642   012720  001777             MOV    #1777,(0)+        ;LOAD X
1445   006646   012720  000000             MOV    #0,(0)+           ;LOAD Y
1446   006652   012720  172000             MOV    #172000,(0)+      ;LOAD STOP
1447   006656   013777  001022  172166      MOV    DBUF,@#DPC        ;START DISPLAY
1448   006664   004737  012340             JSR    7,DLAY            ;EXECUTE A PROGRAM DELAY
1449
1450   006670   032777  000040  172156      BIT    #40,@DSR
1451   006676   001401                      BEQ    .+4
1452   006700   000000                      HALT                     ;ERROR, EDGE FLAG FAILED TO CLEAR
```

```
1454
1455
1456                              ;EDGE FLAG TEST
1457                              ;TEST THAT EXCEEDING -X AXIS SETS EDGE FLAG
1458
1459   006702   104000             GT53:   SCOPE
1460   006704   013700  001022             MOV    DBUF,R0
1461   006710   012720  116000             MOV    #116000,(0)+      ;LOAD POINT
1462   006714   012720  000000             MOV    #0,(0)+           ;LOAD MAX X
1463   006720   012720  000000             MOV    #0,(0)+           ;LOAD Y
1464   006724   012720  110000             MOV    #110000,(0)+      ;LOAD LONG VECTOR
1465   006730   012720  020001             MOV    #20001,(0)+       ;LOAD DETA X
1466   006734   012720  000000             MOV    #0,(0)+           ;LOAD DELTA Y
1467   006740   012720  172000             MOV    #172000,(0)+      ;LOAD STOP
1468   006744   013777  001022  172100      MOV    DBUF,@#DPC        ;START DISPLAY
1469   006752   004737  012340             JSR    7,DLAY            ;EXECUTE A PROGRAM DELAY
1470
1471   006756   032777  000040  172070      BIT    #40,@DSR          ;TEST BIT 5
1472   006764   001002                      BNE    .+6
1473   006766   000000                      HALT                     ;EDGE FLAG FAILED TO SET
1474   006770   000424                      BR     GT54
1475
1476                              ;SUB-TEST, TEST THAT THE EDGE FLAG CLEARS
1477
1478   006772   013700  001022             MOV    DBUF,R0
1479   006776   012720  116000             MOV    #116000,(0)+      ;LOAD POINT
1480   007002   012720  000000             MOV    #0,(0)+           ;LOAD X
1481   007006   012720  000000             MOV    #0,(0)+           ;LOAD Y
1482   007012   012720  172000             MOV    #172000,(0)+      ;LOAD STOP
1483   007016   013777  001022  172026      MOV    DBUF,@#DPC        ;START DISPLAY
1484   007024   004737  012340             JSR    7,DLAY            ;EXECUTE A PROGRAM DELAY
1485
1486   007030   032777  000040  172016      BIT    #40,@DSR
1487   007036   001401                      BEQ    .+4
1488   007040   000000                      HALT                     ;ERROR, DEGE FLAG FAILED TO CLEAR
```

```
1490                                    ;EDGE FLAG TEST
1491                                    ;TEST THAT EXCEEDING +Y AXIS SETS EDGE FLAG
1492
1493   007042 104000          GT54:     SCOPE
1494   007044 013700 001022             MOV     DBUF,R0
1495   007050 012720 116000             MOV     #116000,(0)+      ;LOAD POINT
1496   007054 012720 000000             MOV     #0,(0)+           ;LOAD X
1497   007060 013720 001012             MOV     GSYAXS,(0)+       ;LOAD MAX Y
1498   007064 012720 110000             MOV     #110000,(0)+      ;LOAD LONG VECTOR
1499   007070 012720 000000             MOV     #0,(0)+           ;LOAD DELTA X
1500   007074 012720 000001             MOV     #1,(0)+           ;LOAD DELTA Y
1501   007100 012720 172000             MOV     #172000,(0)+      ;LOAD STOP
1502   007104 013777 001022 171740      MOV     DBUF,@DPC         ;START DISPLAY
1503   007112 004737 012340             JSR     7,DLAY            ;EXECUTE A PROGRAM DELAY
1504
1505   007116 032777 000040 171730      BIT     #40,@DSR          ;TEST BIT 5
1506   007124 001002                    BNE     .+4
1507   007126 000000                    HALT                      ;EDGE FLAG FAILED TO SET
1508   007130 000424                    BR      GT55
1509
1510                                    ;SUB-TEST, TEST THAT THE EDGE FLAG CLEARS
1511
1512   007132 013700 001022             MOV     DBUF,R0
1513   007136 012720 116000             MOV     #116000,(0)+      ;LOAD POINT
1514   007142 012720 000000             MOV     #0,(0)+           ;LOAD X
1515   007146 012720 000000             MOV     #0,(0)+  ;LOAD Y
1516   007152 012720 172000             MOV     #172000,(0)+      ;LOAD STOP
1517   007156 013777 001022 171666      MOV     DBUF,@DPC         ;START DISPLAY
1518   007164 004737 012340             JSR     7,DLAY            ;EXECUTE A PROGRAM DELAY
1519
1520   007170 032777 000040 171656      BIT     #40,@DSR
1521   007176 001401                    BEQ     .+4
1522   007200 000000                    HALT                      ;ERROR, EDGE FLAG FAILED TO CLEAR
1523
1524                                    ;EDGE FLAG TEST
1525                                    ;TEST THAT EXCEEDING -Y AXIS SETS EDGE FLAG
1526
1527   007202 104000          GT55:     SCOPE
1528   007204 013700 001022             MOV     DBUF,R0
1529   007210 012720 116000             MOV     #116000,(0)+      ;LOAD POINT
1530   007234 012720 000000             MOV     #0,(0)+           ;LOAD X
1531   007220 012720 000000             MOV     #0,(0)+           ;LOAD Y
1532   007234 012720 110000             MOV     #110000,(0)+      ;LOAD LONG VECTOR
1533   007230 012720 000000             MOV     #0,(0)+           ;LOAD DELTA X
1534   007234 012720 020001             MOV     #20001,(40)+      ;LOAD DELTA Y
1535   007240 012720 172000             MOV     #172000,(0)+      ;LOAD STOP
1536   007244 013777 001022 171600      MOV     DBUF,@DPC         ;START DISPLAY
1537   007252 004737 012340             JSR     7,DLAY            ;EXECUTE A PROGRAM DELAY
1538
1539   007256 032777 000040 171570      BIT     #40,@DSR          ;TEST BIT 5
1540   007264 001001                    BNE     .+4
1541   007266 000000                    HALT                      ;EDGE FLAG FAILED TO SET
1542
```

```
1544                                    ;TEST THAT THE CHARACTER REGISTER IS LOADED PROPERLY
1545                                    ; CODE 00
1546
1547   007270 104000          GT56:     SCOPE
1548   007272 012777 100000 171522      MOV     #100000,@DBUF     ;LOAD "CHARACTER MODE"
1549   007300 012777 000000 171516      MOV     #0,@DBUF1         ;LOAD "NULL" CHARACTER
1550   007306 012777 172000 171512      MOV     #172000,@DBUF2
1551   007314 013777 001022 171530      MOV     DBUF,@DPC         ;START DISPLAY
1552   007322 004737 012340             JSR     7,DLAY            ;EXECUTE A PROGRAM DELAY
1553   007326 017700 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                      ;ERROR, CHARACTER REGISTER LOADED IN ERROR
1558
1559                                    ;TEST THAT THE CHARACTER REGISTER IS LOADED PROPERLY
1560                                    ; CODE 77
1561
1562   007346 104000          GT57:     SCOPE
1563   007350 012777 100000 171444      MOV     #100000,@DBUF     ;LOAD "CHARACTER MODE"
1564   007356 012777 000077 171440      MOV     #77,@DBUF1        ;LOAD CHARACTER
1565   007364 012777 172000 171434      MOV     #172000,@DBUF2
1566   007372 013777 001022 171452      MOV     DBUF,@DPC         ;START DISPLAY
1567   007400 004737 012340             JSR     7,DLAY            ;EXECUTE A PROGRAM DELAY
1568   007404 017700 171450             MOV     @YRDS,R0          ;READ CHARACTER REG.
1569   007410 042700 001777             BIC     #1777,R0          ;MASK TO BITS 10-15
1570   007414 022700 176000             CMP     #176000,R0
1571   007420 001401                    BEQ     .+4
1572   007422 000000                    HALT                      ;ERROR, CHARACTER REGISTER LOADED IN ERROR
1573
1574                                    ;TEST THAT THE CHARACTER REGISTER IS LOADED PROPERLY
1575                                    ; CODE 25
1576
1577   007424 104000          GT58:     SCOPE
1578   007426 012777 100000 171366      MOV     #100000,@DBUF     ;LOAD "CHARACTER MODE"
1579   007434 012777 000025 171362      MOV     #25,@DBUF1        ;LOAD CHARACTER
1580   007442 012777 172000 171356      MOV     #172000,@DBUF2
1581   007450 013777 001022 171374      MOV     DBUF,@DPC         ;START DISPLAY
1582   007456 004737 012340             JSR     7,DLAY            ;EXECUTE A PROGRAM DELAY
1583   007462 017700 171372             MOV     @YRDS,R0          ;READ CHARACTER REG.
1584   007466 042700 001777             BIC     #1777,R0          ;MASK TO BITS 10-15
1585   007472 022700 052000             CMP     #52000,R0
1586   007476 001401                    BEQ     .+4
1587   007500 000000                    HALT                      ;ERROR, CHARACTER REGISTER LOADED IN ERROR
```

```
1589
1590                                          ;TEST THAT THE CHARACTER REGISTER IS LOADED PROPERLY
1591                                          ; CODE 52
1592
1593  007502  104000           GT591   SCOPE
1594  007504  012777  100000  171310         MOV   #100000,@DBUP   ;LOAD "CHARACTER MODE"
1595  007512  012777  000052  171304         MOV   #52,@DBUF1      ;LOAD CHARACTER
1596  007520  012777  172000  171300         MOV   #172000,@DBUF2
1597  007526  013777  001022  171316         MOV   DBUF,@DPC       ;START DISPLAY
1598  007534  004737  012340                 JSR   7,DLAY          ;EXECUTE A PROGRAM DELAY
1599  007540  017700  171314                 MOV   @YPOS,R0        ;READ CHARACTER REG.
1600  007544  042700  001777                 BIC   #1777,R0        ;MASK TO BITS 10-15
1601  007550  022700  124000                 CMP   #124000,R0
1602  007554  001401                         BEQ   .+4
1603  007556  000000                         HALT                  ;ERROR, CHARACTER REGISTER LOADED IN ERROR
1604
1605                                          ;TEST THAT CHARACTER MODE DOES NOT HANG THE DISPLAY PROCESSOR
1606                                          ;TEST THAT "NULL" DOES NOT CHANGE X OR Y AXIS
1607
1608  007560  104000           GT601   SCOPE
1609  007562  012777  110000  171232         MOV   #110000,@DBUF   ;POINT MODE
1610  007570  012777  001000  171226         MOV   #1000,@DBUF1
1611  007576  012777  001000  171222         MOV   #1000,@DBUF2    ;1000,1000
1612  007604  012777  100000  171216         MOV   #100000,@DBUP3  ;LOAD "CHARACTER MODE"
1613  007612  005077  171214                 CLR   @DBUF4          ;NULL CHARACTER
1614  007616  012777  172000  171210         MOV   #172000,@DBUP5
1615  007624  013777  001022  171220         MOV   DBUF,@DPC       ;LOAD THE DISPLAY P.C.
1616  007632  004737  012340                 JSR   7,DLAY          ;EXECUTE A PROGRAM DELAY
1617
1618  007636  017700  171216                 MOV   @YPOS,R0        ;READ CHARACTER REGISTER
1619  007642  042700  001777                 BIC   #1777,R0        ;MASK TO BITS 10-15
1620  007646  022700  000000                 CMP   #0,R0
1621  007652  001402                         BEQ   .+6
1622  007654  000000                         HALT                  ;CHARACTER REGISTER IN ERROR
1623  007656  000417                         BR    GT61
1624
1625  007660  017700  171172                 MOV   @XPOS,R0        ;READ X AXIS
1626  007664  022700  001000                 CMP   #1000,R0        ;ARE THEY EQUAL ?
1627  007670  001402                         BEQ   .+6             ;YES
1628  007672  000000                         HALT                  ;"NULL" CHARACTER CHANGED X AXIS
1629  007674  000410                         BR    GT61
1630
1631  007676  017700  171156                 MOV   @YPOS,R0        ;READ Y AXIS
1632  007702  042700  176000                 BIC   #176000,R0      ;MASK TO BITS 0-9
1633  007706  022700  001000                 CMP   #1000,R0        ;ARE THEY EQUAL ?
1634  007712  001401                         BEQ   .+4             ;YES
1635  007714  000000                         HALT                  ;"NULL" CHARACTER CHANGED Y AXIS
1636
```

```
1638
1639
1640                                          ;TEST THAT CHARACTER MODE DOES NOT HANG THE DISPLAY PROCESSOR
1641                                          ;TEST THAT "CR" DOES CHANGE X AND DOES NOT CHANGE Y AXIS
1642
1643  007716  104000           GT611   SCOPE
1644  007720  012777  110000  171074         MOV   #110000,@DBUF   ;POINT MODE
1645  007726  012777  001000  171070         MOV   #1000,@DBUF1
1646  007734  012777  001000  171064         MOV   #1000,@DBUF2    ;1000,1000
1647  007742  012777  100000  171060         MOV   #100000,@DBUP3  ;LOAD "CHARACTER MODE"
1648  007750  012777  000015  171054         MOV   #15,@DBUF4      ;LOAD "CR"
1649  007756  012777  172000  171050         MOV   #172000,@DBUP5  ;LOAD STOP
1650  007764  013777  001022  171040         MOV   DBUF,@DPC       ;LOAD THE DISPLAY P.C.
1651  007772  004737  012340                 JSR   7,DLAY          ;EXECUTE A PROGRAM DELAY
1652
1653  007776  017700  171056                 MOV   @YPOS,R0        ;READ Y AXIS
1654  010002  042700  001777                 BIC   #1777,R0        ;MASK TO BITS 10-15
1655  010006  022700  032000                 CMP   #32000,R0
1656  010012  001402                         BEQ   .+6
1657  010014  000000                         HALT                  ;CHARACTER REGISTER FAILED TO LOAD CORRECTLY
1658  010016  000417                         BR    GT62
1659
1660  010020  017700  171032                 MOV   @XPOS,R0        ;READ X AXIS
1661  010024  022700  000000                 CMP   #0,R0           ;ARE THEY EQUAL ?
1662  010030  001402                         BEQ   .+6             ;YES
1663  010032  000000                         HALT                  ;"CR" CHARACTER FAILED TO CHANGED X AXIS CORRECTLY
1664  010034  000410                         BR    GT62
1665
1666  010036  017700  171016                 MOV   @YPOS,R0        ;READ Y AXIS
1667  010042  042700  176000                 BIC   #176000,R0      ;MASK TO BITS 0-9
1668  010046  022700  001000                 CMP   #1000,R0        ;ARE THEY EQUAL ?
1669  010052  001401                         BEQ   .+4             ;YES
1670  010054  000000                         HALT                  ;"CR" CHARACTER CHANGED Y AXIS
1671
```

```
1673
1674
1675                                    ;TEST THAT CHARACTER MODE DOES NOT HANG THE DISPLAY PROCESSOR
1676                                    ;TEST THAT "LF" DOES NOT CHANGE X BUT DOES CHANGE Y AXIS
1677
1678  010056 104000           GT02:    SCOPE
1679  010060 012777 116000 170734       MOV    #116000,@DBUF    ;POINT MODE
1680  010066 012777 001000 170730        MOV    #1000,@DBUF1
1681  010074 012777 001000 170724        MOV    #1000,@DBUF2     ;1000,1000
1682  010102 012777 100000 170720        MOV    #100000,@DBUF3   ;LOAD "CHARACTER MODE"
1683  010110 012777 000012 170714        MOV    #12,@DBUF4
1684  010116 012777 172000 170710        MOV    #172000,@DBUF5
1685  010124 013777 001022 170720        MOV    DBUF,@DPC        ;LOAD THE DISPLAY P.C.
1686  010132 004737 012340               JSR    7,DLAY           ;EXECUTE A PROGRAM DELAY
1687
1688  010136 017700 170716               MOV    @YPOS,R0         ;READ CHARACTER REG.
1689  010142 042700 001777               BIC    #1777,R0         ;MASK TO BITS 10-15
1690  010146 022700 024000               CMP    #24000,R0
1691  010152 001402                      BEQ    .+6
1692  010154 000000                      HALT                    ;CHARACTER REGISTER IN ERROR
1693  010156 000477                      BR     GT03
1694
1695  010160 017700 170672               MOV    @XPOS,R0         ;READ X AXIS
1696  010164 022700 001000               CMP    #1000,R0         ;ARE THEY EQUAL ?
1697  010170 001402                      BEQ    .+6              ;YES
1698  010172 000000                      HALT                    ;"LF" CHARACTER CHANGED X AXIS
1699  010174 000470                      BR     GT03
1700
1701  010176 017700 170656               MOV    @YPOS,R0         ;READ Y AXIS
1702  010202 042700 176000               BIC    #176000,R0       ;MASK TO BITS 10-15
1703  010206 023700 001046               CMP    LFSIZE,R0        ;ARE THEY EQUAL ?
1704  010212 001401                      BEQ    .+4              ;YES
1705  010214 000000                      HALT                    ;"LF" CHARACTER FAILED TO CHANGED Y AXIS CORRECTLY
1706
```

```
1708
1709
1710                                    ;TEST THAT CHARACTER MODE DOES NOT HANG THE DISPLAY PROCESSOR
1711                                    ;TEST THAT "A" DOES CHANGE X BUT NOT Y AXIS
1712
1713  010216 104000           GT02A:   SCOPE
1714  010220 012777 116000 170574       MOV    #116000,@DBUF    ;POINT MODE
1715  010226 012777 000000 170570        MOV    #0,@DBUF1
1716  010234 012777 001000 170564        MOV    #1000,@DBUF2     ;0,1000
1717  010242 012777 100000 170560        MOV    #100000,@DBUF3   ;LOAD "CHARACTER MODE"
1718  010250 012777 000101 170554        MOV    #101,@DBUF4      ;LOAD AN "A"
1719  010256 012777 172000 170550        MOV    #172000,@DBUF5
1720  010264 013777 001022 170560        MOV    DBUF,@DPC        ;LOAD THE DISPLAY P.C.
1721  010272 004737 012340               JSR    7,DLAY           ;EXECUTE A PROGRAM DELAY
1722
1723  010276 017700 170556               MOV    @YPOS,R0         ;READ CHARACTER REG
1724  010302 042700 001777               BIC    #1777,R0         ;MASK TO BITS 10-15
1725  010306 022700 002000               CMP    #2000,R0
1726  010312 001402                      BEQ    .+6
1727  010314 000000                      HALT                    ;CHARACTER REGISTER IN ERROR
1728  010316 000417                      BR     GT03
1729
1730  010320 017700 170532               MOV    @XPOS,R0         ;READ X AXIS
1731  010324 023700 001006               CMP    GSCHSZ,R0        ;ARE THEY EQUAL ?
1732  010330 001402                      BEQ    .+6              ;YES
1733  010332 000000                      HALT                    ;"A" CHARACTER FAILED TO CHANGED X AXIS CORRECTLY
1734  010334 000410                      BR     GT03
1735
1736  010336 017700 170516               MOV    @YPOS,R0         ;READ Y AXIS
1737  010342 042700 176000               BIC    #176000,R0       ;MASK TO BITS 0-9
1738  010346 022700 001000               CMP    #1000,R0         ;ARE THEY EQUAL ?
1739  010352 001401                      BEQ    .+4              ;YES
1740  010354 000000                      HALT                    ;"A" CHARACTER CHANGED Y AXIS
1741
```

```
1743
1744
1745                                    ;TEST THAT CHARACTER MODE DOES NOT HANG THE DISPLAY PROCESSOR
1746                                    ;TEST THAT "BS" DOES CHANGE X BUT NOT Y AXIS
1747
1748   012356  104000          GT631   SCOPE
1749   012360  012777  116000  170434          MOV     #110000,.DBUF   ;POINT MODE
1750   012366  012777  001000  170430          MOV     #1000,.DBUF1
1751   012374  012777  001000  170424          MOV     #1000,.DBUF2    ;1000,1000
1752   012402  012777  100000  170420          MOV     #100000,.DBUF3  ;LOAD "CHARACTER MODE"
1753   012410  012777  000010  170414          MOV     #10,.DBUF4
1754   012416  012777  172000  170410          MOV     #172000,.DBUF5
1755   012424  013777  001022  170420          MOV     DBUF,.DPC       ;LOAD THE DISPLAY P.C.
1756   012432  004737  012340                  JSR     7,DLAY          ;EXECUTE A PROGRAM DELAY
1757
1758   012436  017700  170416                  MOV     @YPOS,R0        ;READ CHARACTER REG
1759   012442  042700  001777                  BIC     #1777,R0        ;MASK TO BITS 10-15
1760   012446  022700  020000                  CMP     #20000,R0
1761   012452  001402                          BEQ     .+6
1762   012454  000000                          HALT                    ;CHARACTER REGISTER IN ERROR
1763   012456  000425                          BR      GT64
1764
1765   012460  017700  170372                  MOV     @XPOS,R0        ;READ X AXIS
1766   012464  023700  001050                  CMP     CHSIZE,R0       ;ARE THEY EQUAL ?
1767   012470  001402                          BEQ     .+6             ;YES
1768   012472  000000                          HALT                    ;"BS" CHARACTER FAILED TO CHANGED X AXIS CORRECTLY
1769   012474  000416                          BR      GT64
1770
1771   012476  017700  170356                  MOV     @YPOS,R0        ;READ Y AXIS
1772   012502  042700  176000                  BIC     #176000,R0      ;MASK TO BITS 0-9
1773   012506  022700  001000                  CMP     #1000,R0        ;ARE THEY EQUAL ?
1774   012512  001401                          BEQ     .+4             ;YES
1775   012514  000000                          HALT                    ;"BS" CHARACTER CHANGED Y AXIS
1776
1777                                    ;TEST THAT "SHIFT-OUT" STATUS BIT IS NOT SET
1778
1779   012516  017700  170332          GT63A1  MOV     @DSR,R0         ;READ STATUS
1780   012522  032700  000100                  BIT     #100,R0
1781   012526  001401                          BEQ     .+4
1782   012530  000000                          HALT                    ;SHIFT OUT STATUS BIT IS SET
1783
```

```
1785
1786                                    ;TEST THAT "SHIFT-OUT" GENERATES A STATUS BIT
1787                                    ;SHIFT-OUT <LOW BYTE>, FOLLOWED BY CODE 77 <HIGH BYTE>
1788
1789   012532  104000          GT641   SCOPE
1790   012534  012777  116000  170260          MOV     #110000,.DBUF   ;POINT MODE
1791   012542  012777  001000  170254          MOV     #1000,.DBUF1
1792   012550  012777  001000  170250          MOV     #1000,.DBUF2    ;1000,1000
1793   012556  012777  100000  170244          MOV     #100000,.DBUF3  ;LOAD "CHARACTER MODE"
1794   012564  012777  037416  170240          MOV     #037416,.DBUF4  ;"SHIFT-OUT" IN LOW BYTE #77 IN HIGH BYTE
1795   012572  012777  172000  170234          MOV     #172000,.DBUF5  ;LOAD STOP
1796   012600  013777  001022  170244          MOV     DBUF,.DPC       ;START DISPLAY
1797   012606  004737  012340                  JSR     7,DLAY          ;EXECUTE A PROGRAM DELAY
1798
1799   012612  017700  170242                  MOV     @YPOS,R0        ;READ CHARACTER REG
1800   012616  042700  001777                  BIC     #1777,R0        ;MASK TO BITS 10-15
1801   012622  022700  176000                  CMP     #176000,R0
1802   012626  001402                          BEQ     .+6
1803   012630  000000                          HALT                    ;CHARACTER REGISTER IN ERROR
1804   012632  000426                          BR      GT64            ; AFTER A SHIFT-OUT COMMAND
1805
1806   012634  017700  170214                  MOV     @DSR,R0         ;READ STATUS REGISTER
1807   012640  032700  000100                  BIT     #100,R0
1808   012644  001002                          BNE     .+6
1809   012646  000000                          HALT                    ;SHIFT OUT STATUS BIT FAILED TO SET
1810   012650  000417                          BR      GT64
1811
1812   012652  017700  170200                  MOV     @XPOS,R0        ;READ X POS
1813   012656  022700  001000                  CMP     #1000,R0
1814   012662  001402                          BEQ     .+6
1815   012664  000000                          HALT                    ;SHIFT-OUT CHARACTER CHANGED X AXIS
1816   012666  000410                          BR      GT64
1817
1818   012670  017700  170164                  MOV     @YPOS,R0        ;READ Y POS
1819   012674  042700  176000                  BIC     #176000,R0      ;MASK
1820   012700  022700  001000                  CMP     #1000,R0
1821   012704  001401                          BEQ     .+4
1822   012706  000000                          HALT                    ;SHIFT-OUT CHARACTER CHANGED Y AXIS
```

```
1824
1825                                              ;TEST THAT "SHIFT-OUT" DOES NOT GENERATE A STATUS BIT
1826                                              ;("SHIFT-OUT" FOLLOWED BY CODE 0 THRU 37 EXCEPT #17)
1827
1828  212710  104000              GT65:    SCOPE
1829  010712  000005                       RESET
1830  010714  005003                       CLR      R3
1831  010716  012777  100000  170076        MOV      #100000,@DBUF     ;SET 'CHAR' MODE
1832  010724  012777  000016  170072        MOV      #16,@DBUF1        ;LOAD "SHIFT-OUT" INTO THE LOW BYTE
1833  010732  012777  172000  170066        MOV      #172000,@DBUF2
1834  010740  110339  013467       GT65A:   MOVB     R3,BUFFER+3       ;LOAD HIGH BYTE
1835  010744  000240                        NOP
1836  010746  013777  001022  170076        MOV      DBUF+@DPC         ;START THE DISPLAY
1837  010754  004737  012340               JSR      7,DLAY            ;EXECUTE A PROGRAM DELAY
1838
1839  010760  032777  000100  170066        BIT      #100,@DSR         ;TEST FOR SHIFT BIT
1840  010766  001402                        BEQ      .+6
1841  010770  000000                        HALT                       ;SHIFT STATUS BIT SET IN ERROR
1842  010772  000407                        BR       GT65              ; CHARACTER IS IN R3
1843
1844  010774  005203              GT65B:   INC      R3
1845  010776  022703  000017               CMP      #17,R3            ;TEST FOR "SHIFT-IN"
1846  011002  001774                        BEQ      GT65B
1847  011004  022703  000040               CMP      #40,R3            ;TEST FOR #40
1848  011010  001353                        BNE      GT65A             ;IS IT #40
1849                                                                   ;YES, NEXT TEST?
1850
1851                                              ;TEST THAT "SHIFT-OUT" FOLLOWED BY CODE 40 GENERATE A
1852                                              ;SHIFT STATUS BIT
1853
1854  011012  104000              GT66:    SCOPE
1855  011014  000005                       RESET
1856  011016  012777  100000  167776        MOV      #100000,@DBUF     ;LOAD SET CHAR MODE
1857  011024  012777  000016  167772        MOV      #16,@DBUF1        ;LOAD "SHIFT-OUT" INTO THE LOW BYTE
1858  011032  012777  172000  167766        MOV      #172000,@DBUF2
1859  011040  112737  000040  013467 GT66A: MOVB     #40,BUFFER+3      ;LOAD HIGH BYTE
1860  011046  000240                        NOP
1861  011050  013777  001022  167774        MOV      DBUF+@DPC         ;START THE DISPLAY
1862  011056  004737  012340               JSR      7,DLAY            ;EXECUTE A PROGRAM DELAY
1863
1864  011062  032777  000100  167764        BIT      #100,@DSR         ;TEST 'SHIFT' STATUS BIT
1865  011070  001002                        BNE      .+6
1866  011072  000000                        HALT                       ;"SHIFT-OUT" STATUS BIT FAILED TO SET
1867  011074  000407                        BR       GT67              ;ON CHARACTER IN R3
1868
```

```
1870
1871  011076  000005                       RESET
1872  011100  032777  000100  167746        BIT      #100,@DSR         ;TEST SHIFT-OUT BIT
1873  011106  001402                        BEQ      GT67              ;BR IF CLEARED
1874  011110  000000                        HALT                       ;SHIFT OUT STATUS BIT FAILED TO CLEAR
1875  011112  000406                        BR       GT67
1876
1877                                              ;TEST THAT 'SHIFT-OUT' IN THE HIGH BYTE FOLLOWED BY A CHARACTER
1878                                              ; IN THE NEXT LOW BYTE GENERATES A STATUS BIT
1879
1880  011114  104000              GT67:    SCOPE
1881  011116  012777  100000  167676        MOV      #100000,@DBUF     ;LOAD SET 'CHAR' MODE
1882  011124  005077  167674                CLR      @DBUF1
1883  011130  012777  007000  167666        MOV      #7000,@DBUF1      ;LOAD 'SHIFT-OUT' INTO THE HIGH BYTE
1884  011136  012777  000040  167662        MOV      #40,@DBUF2        ;LOAD A SHIFT-OUT CHARACTER IN THE NEXT
1885                                                                   ;WORD <LOW BYTE>
1886  011144  012777  172000  167656        MOV      #172000,@DBUF3
1887  011152  000005                        RESET
1888  011154  000240                        NOP
1889  011156  013777  001022  167666        MOV      DBUF+@DPC         ;START THE DISPLAY
1890  011164  004737  012340               JSR      7,DLAY            ;EXECUTE A PROGRAM DELAY
1891
1892  011170  032777  000100  167656        BIT      #100,@DSR         ;TEST THE STATUS REGISTER
1893  011176  001002                        BNE      .+6
1894  011200  000000                        HALT                       ;SHIFT-OUT IN THE HIGH BYTE FAILED TO
1895  011202  000410                        BR       GT68              ;SET A STATUS BIT
1896
1897  011204  017700  167650                MOV      @YPOS,R0          ;READ Y POS
1898  011210  042700  001777                BIC      #1777,R0          ;MASK TO BITS 15-10
1899  011214  022700  100000                CMP      #100000,R0        ;TEST FOR CHAR #40
1900  011220  001401                        BEQ      .+4
1901  011222  000000                        HALT                       ;CHARACTER REGISTER IN ERROR AFTER A
1902                                                                   ;"SHIFT-OUT" <HIGH BYTE> FOLLOWED BY
1903                                                                   ; #40 <LOW BYTE NEXT WORD>
1904
```

```
1906                                  ;STOP INTERRUPT TEST
1907                                  ;TEST FOR NO INTERRUPT
1908
1909  011224  104000          GT68:   SCOPE
1910  011226  000005                  RESET
1911  011230  012777  011312  167624  MOV  #GT68A,@DDONE    ;LOAD RETURN FROM DONE INTERRUPT
1912  011236  012777  011312  167626  MOV  #GT68A,@TIMEVT   ;LOAD RETURN FROM TIME-OUT INTERRUPT
1913  011244  012777  011312  167614  MOV  #GT68A,@LPVCT    ;LOAD RETURN FROM LIGHT-PEN INTERRUPT
1914  011252  012777  164000  167542  MOV  #164000,@DBUP    ;LOAD "DISPLAY NOP"
1915  011260  012777  173000  167536  MOV  #173000,@DBUP1   ;LOAD "STATUS A","STOP","STOP INT; ENABLE"
1916  011266  005077  167526          CLR  @PSW             ;LOWER MACHINE PRIORITY
1917  011272  013777  001022  167552  MOV  DBUF,@DPC        ;LOAD DISPLAY P.C.
1918  011300  000240                  NOP
1919  011302  000240                  NOP
1920  011304  000240                  NOP
1921  011306  000240                  NOP
1922  011310  000401                  BR   .+4
1923
1924  011312  000000          GT68A:  HALT                  ;GT-40 INTERRUPTED IN ERROR
1925
1926                                  ;STOP INTERRUPT TEST
1927                                  ;TEST FOR INTERRUPT
1928
1929  011314  104000          GT69:   SCOPE
1930  011316  000005                  RESET
1931  011320  012777  011402  167534  MOV  #GT69A,@DDONE    ;LOAD RETURN ADDRESS FROM INTERRUPT
1932  011326  012777  011414  167532  MOV  #GT69B,@LPVCT    ;LOAD LP VECTOR
1933  011334  012777  011422  167530  MOV  #GT69C,@TIMEVT   ;LOAD TO VECTOR
1934  011342  012777  164000  167452  MOV  #164000,@DBUP    ;LOAD "DISPLAY NOP"
1935  011350  012777  173400  167446  MOV  #173400,@DBUP1   ;LOAD "STATUS A","STOP","STOP INT; ENABLE-INT"
1936  011356  005077  167436          CLR  @PSW
1937  011362  013777  001022  167462  MOV  DBUF,@DPC
1938  011370  000240                  NOP
1939  011372  000240                  NOP
1940  011374  000240                  NOP
1941  011376  000240                  NOP
1942  011400  000000                  HALT                  ;GT-40 FAILED TO GENERATE A STOP INTERRUPT
1943  011402  013777  001064  167452  GT69A:  MOV  DDONE1,@DDONE
1944  011410  022626                  CMP  (SP)+,(SP)+
1945  011412  000405                  BR   GT70
1946
1947  011414  022626          GT69B:  CMP  (SP)+,(SP)+
1948  011416  000000                  HALT                  ;GT-40 STOP (DONE) INTERRUPTED TO THE
1949                                                         ; LIGHT-PEN VECTOR
1950  011420  000402                  BR   GT70
1951
1952  011422  022626          GT69C:  CMP  (SP)+,(SP)+
1953  011424  000000                  HALT                  ;GT-40 STOP (DONE) INTERRUPTED TO THE
1954                                                         ; TIME-OUT VECTOR
```

```
1956
1957                                  ;SHIFT OUT INTERRUPT TEST
1958                                  ;TEST FOR INTERRUPT
1959
1960  011426  104000          GT70:   SCOPE
1961  011430  000005                  RESET
1962  011432  012777  011536  167422  MOV  #GT70B,@DDONE    ;LOAD DONE VECTOR
1963  011440  012777  011544  167420  MOV  #GT70C,@LPVCT    ;LOAD LIGHT-PEN VECTOR
1964  011446  012777  011522  167416  MOV  #GT70A,@TIMEVT   ;LOAD RETURN ADDRESS
1965  011454  012777  100000  167340  MOV  #100000,@DBUP    ;LOAD "CHARACTER MODE"
1966  011462  012777  020016  167334  MOV  #020016,@DBUF1   ;LOAD "SHIFT-OUT"
1967  011470  012777  173000  167330  MOV  #173000,@DBUP2
1968  011476  005077  167316          CLR  @PSW
1969  011502  013777  001022  167342  MOV  DBUF,@DPC        ;START DISPLAY
1970  011510  000240                  NOP
1971  011512  000240                  NOP
1972  011514  000240                  NOP
1973  011516  000240                  NOP
1974  011520  000000                  HALT                  ;GT-40 FAILED TO INTERRUPT ON SHIFT-OUT
1975  011522  000240          GT70A:  NOP
1976  011524  013777  001074  167340  MOV  THEVT1,@TIMEVT
1977  011532  022626                  CMP  (SP)+,(SP)+
1978  011534  000405                  BR   GT71
1979
1980  011536  022626          GT70B:  CMP  (SP)+,(SP)+
1981  011540  000000                  HALT                  ;GT-40 SHIFT-OUT INTERRUPTED TO THE
1982                                                         ; DONE VECTOR
1983  011542  000402                  BR   GT71
1984
1985  011544  022626          GT70C:  CMP  (SP)+,(SP)+
1986  011546  000000                  HALT                  ;GT-40 SHIFT-OUT INTERRUPTED TO THE
1987                                                         ; LIGHT PEN VECTOR
1988
```

```
1990
1991
1992                                    ;TIME-OUT INTERRUPT TEST
1993
1994   011550  104000            GT71:   SCOPE
1995   011552  000005                    RESET
1996   011554  013777  001064  167300    MOV    DDONE1,@DDONE
1997   011562  013777  001070  167276    MOV    LPVCT1,@LPVCT
1998   011570  012777  011616  167274    MOV    #GT71A,@TIMEVT   ;LOAD RETURN ADDRESS
1999   011576  005077  167216            CLR    @PSW
2000   011602  012777  177776  167242    MOV    #177776,@DPC     ;LOAD DISPLAY P.C.
2001   011610  004737  012340            JSR    7,DLAY           ;EXECUTE A PROGRAM DELAY
2002   011614  000000                    HALT                    ;GT-40 FAILED TO INTERRUPT ON TIME-OUT
2003
2004   011616  000240            GT71A:  NOP
2005   011620  013777  001074  167244    MOV    TMEVT1,@TIMEVT
2006   011626  022626                    CMP    (SP)+,(SP)+
2007
2008                                    ;NO LIGHT PEN INTERRUPT TEST
2009
2010   011630  104000            GT72:   SCOPE
2011   011632  000005                    RESET
2012   011634  012777  011676  167224    MOV    #GT72A,@LPVCT    ;LOAD RETURN ADDRESS
2013   011642  012777  100140  167152    MOV    #100140,@DBUF    ;LOAD DISPLAY BUFFER
2014   011650  012777  173000  167146    MOV    #173000,@DBUF1
2015   011656  005077  167136            CLR    @PSW
2016   011662  013777  001022  167162    MOV    DBUF,@DPC
2017   011670  004737  012340            JSR    7,DLAY           ;EXECUTE A PROGRAM DELAY
2018   011674  000401                    BR     .+4
2019   011676  000000            GT72A:  HALT                    ;GT-40 INTERRUPTED ON FALSE LIGHT PEN FLAG
2020   011700  013777  001070  167160    MOV    LPVCT1,@LPVCT
```

```
2022                                    ;PRE BB LEVEL SETUP
2023
2024   011706  042737  177437  001004    BIC    #177437,DSPBR    ;MASK TO BITS
2025   011714  001001                    BNE    .+4
2026   011716  000000                    HALT                    ;BR LEVEL WAS 6
2027   011720  022737  000340  001004    CMP    #340,DSPBR
2028   011726  001001                    BNE    .+4
2029   011730  000000                    HALT                    ;BR LEVEL WAS 7
2030
2031   011732  013737  001004  011756    MOV    DSPBR,BRLEV1
2032   011740  162737  000040  011756    SUB    #40,BRLEV1
2033   011746  013737  001004  011760    MOV    DSPBR,BRLEV2
2034   011754  000402                    BR     GT73
2035
2036   011756  000140            BRLEV1: 140
2037   011760  000200            BRLEV2: 200
2038
2039                                    ;BR LEVEL TEST (BR=1)
2040                                    ;TEST FOR INTERRUPT
2041
2042   011762  104000            GT73:   SCOPE
2043   011764  000005                    RESET
2044   011766  012777  012030  167046    MOV    #GT73A,@DDONE    ;LOAD RETURN ADDRESS
2045   011774  012777  173400  167020    MOV    #173400,@DBUF    ;LOAD "STATUS A"-NO INTERRUPT ENABLE
2046   012002  013777  011756  167010    MOV    BRLEV1,@PSW
2047   012010  013777  001022  167034    MOV    DBUF,@DPC        ;LOAD THE DISPLAY P.C.
2048   012016  000240                    NOP
2049   012020  000240                    NOP
2050   012022  000240                    NOP
2051   012024  000240                    NOP
2052   012026  000000                    HALT                    ;NO STOP INTERRUPT ON BR LEVEL INDICATED -1
2053                                                              ;CHECK TO SEE IF PROPER BR LEVEL
2054   012030  022626            GT73A:  CMP    (SP)+,(SP)+
2055
2056                                    ;BR LEVEL TEST (BR)
2057                                    ;TEST THAT THE GT-40 DOES NOT INTERRUPT AT THE LEVEL INDICATED
2058
2059   012032  104000            GT74:   SCOPE
2060   012034  000005                    RESET
2061   012036  012777  012100  167016    MOV    #GT74A,@DDONE    ;LOAD RETURN ADDRESS
2062   012044  012777  173400  166750    MOV    #173400,@DBUF    ;LOAD "STATUS A- STOP- STOP INT ENABLE
2063   012052  013777  011760  166740    MOV    BRLEV2,@PSW      ;LOWER MACHINE PRIORITY TO INDICATED LEVEL
2064   012060  013777  001022  166764    MOV    DBUF,@DPC
2065   012066  000240                    NOP
2066   012070  000240                    NOP
2067   012072  000240                    NOP
2068   012074  000240                    NOP
2069   012076  000401                    BR     .+4              ;NEXT TEST
2070   012100  000000            GT74A:  HALT                    ;GT-40 INTERRUPTED ON THE WRONG BR LEVEL
2071   012102  013777  001064  166752    MOV    DDONE1,@DDONE    ;LOAD INTERRUPT VECTOR
2072   012110  000005                    RESET
2073   012112  000005                    RESET
```

```
2075
2076
2077                                     ;RESET TEST
2078                                     ;DOES RESET CLEAR ALL DISPLAY PC AND STATUS BITS
2079
2080  012114  104000              GT75:  SCOPE
2081  012116  012777  117637 166676      MOV    #117637,@DBUF   ;POINT INTENSITY=7,BLINK=1,LINETYPE=3
2082  012124  005077  166674            CLR    @DBUF1          ; X = 0
2083  012130  005077  166672            CLR    @DBUF2          ; Y = 0
2084  012134  012777  172077 166666      MOV    #172077,@DBUF3  ;ITALIC=1,SYNC=1,COLOR=1
2085  012142  013777  001022 164702      MOV    DBUF#@DPC       ;LOAD DISPLAY P.C.
2086  012150  004737  012340            JSR    PC,DLAY         ;DELAY
2087  012154  000005                    RESET                  ;GENERATE "INIT"
2088  012156  005777  166670            TST    @DPC
2089  012162  001402                    BEQ    .+6
2090  012164  000000                    HALT                   ;RESET FAILED TO CLEAR DISPLAY PC
2091  012166  000406                    BR     END
2092
2093  012170  017700  166660            MOV    @DSR,R0         ;READ DISPLAY STATUS
2094  012174  042700  074000            BIC    #74000,R0       ;MASK TO BIT 11-14
2095  012200  001401                    BEQ    .+4             ;IS THE STATUS CLEARED ?
2096  012202  000000                    HALT                   ;"INIT" FAILED TO RESET DISPLAY STATUS REGISTER
2097
2098  012204  104000              END:   SCOPE
2099  012206  005237  001016            INC    ICNT            ;UPDATE COUNT
2100  012212  022737  000010 001016      CMP    #10,ICNT        ;FINISHED ?
2101  012220  001402                    BEQ    HERE            ;BR IF YES
2102  012222  000137  001412            JMP    GT75            ;NO RE-DO
2103  012226  000005              HERE:  RESET
2104  012230  013700  000042            MOV    @#42,R0
2105  012234  001410                    BEQ    HERE1           ;BRANCH IF OFF LINE
2106  012236  000005                    RESET
2107  012240  004710              LOGICAL: JSR  PC,@0)
2108  012242  000240                    NOP
2109  012244  000240                    NOP
2110  012246  000240                    NOP
2111  012250  000240                    NOP
2112  012252  000240                    NOP
2113  012254  000240                    NOP
2114  012256  012777  000002 164570 HERE1: MOV  #2,@DSR         ;RING THE BELL
2115  012264  012737  000207 177566      MOV    #207,TPDBR      ;PRINT THE BELL
2116  012272  105737  177564      1S:    TSTB   TPDBR           ;WAIT
2117  012276  100375                    BPL    1S
2118  012300  012737  000207 177566      MOV    #207,TPDBR      ;PRINT BELL
2119  012306  105737  177564      2S:    TSTB   TPDBR
2120  012312  100375                    BPL    2S
2121  012314  000137  001336            JMP    START
```

```
2123                                     ;SCOPE ROUTINE
2124
2125  012320  032737  040000 177570 SCOPEA: BIT #40000,@#DISPLAY    ;TEST "SCOPE" SWITCH
2126  012326  001001                    BNE    SCOPEB
2127  012330  011601                    MOV    (SP),R1
2128  012332  012706  000500      SCOPEB: MOV  #STKPTR,SP
2129  012336  000111                    JMP    (1)
2130
2131  012340  012700  000200      DLAY:  MOV    #200,R0
2132  012344  005300              DLAYA: DEC    R0
2133  012346  001376                    BNE    DLAYA
2134  012350  000207                    RTS    7
2135
2136  012352  012700  001000      DLAY1: MOV    #1000,R0
2137  012356  005300              DLAY1A: DEC   R0
2138  012360  001376                    BNE    DLAY1A
2139  012362  000207                    RTS    7
2140
2141  012364  010046              LOWPWR: MOV   R0,-(SP)
2142  012366  010146                    MOV    R1,-(SP)
2143  012370  010246                    MOV    R2,-(SP)
2144  012372  010346                    MOV    R3,-(SP)
2145  012374  010446                    MOV    R4,-(SP)
2146  012376  010546                    MOV    R5,-(SP)
2147  012400  010637  012462            MOV    SP,LOWSV
2148  012404  012737  012414 000024      MOV    #HIGPWR,@#24
2149  012412  000000                    HALT
2150  012414  013706  012462      HIGPWR: MOV   LOWSV,SP
2151  012420  012605                    MOV    (SP)+,R5
2152  012422  012604                    MOV    (SP)+,R4
2153  012424  012603                    MOV    (SP)+,R3
2154  012426  012602                    MOV    (SP)+,R2
2155  012430  012601                    MOV    (SP)+,R1
2156  012432  012600                    MOV    (SP)+,R0
2157  012434  012737  012364 000024      MOV    #LOWPWR,@#24
2158  012442  012706  000500            MOV    #STKPTR,SP
2159  012446  000240                    NOP
2160  012450  000240                    NOP
2161  012452  000000                    HALT
2162  012454  000240                    NOP
2163  012456  000240                    NOP
2164  012460  000111                    JMP    (R1)
2165
2166  012462  000000              LOWSV: 0
2167
```

```
2169
2170  012464  012777  000340  166326  START1: MOV    #340,@PSW
2171  012472  012706  000500                  MOV    #STKPTR,SP
2172  012476  004737  001076                  JSR    PC,SETUP
2173  012502  012701  012464                  MOV    #START1,R1
2174  012506  012777  012542  166346          MOV    #SPCEA,@DDONE   ;SET UP DONE VECTOR
2175  012514  013777  001004  166342          MOV    DSPBR,@DDONE1
2176  012522  012777  012616  166336          MOV    #SPACEB,@LPVCT  ;SET UP LIGHT-PEN VECTOR
2177  012530  013777  001004  166332          MOV    DSPBR,@LPVCT1
2178  012536  000240                          NOP
2179  012540  000240                          NOP
2180  012542  012706  000500          SPCEA:  MOV    #STKPTR,SP
2181  012546  012737  173400  013426          MOV    #DSTOP,FRM7A
2182  012554  017700  166260                  MOV    @SWR,R0
2183  012560  006100                          ROL    R0
2184  012562  042700  177761                  BIC    #177761,R0
2185  012566  016002  012636                  MOV    DSPTCH(0),R2
2186  012572  010277  166254                  MOV    R2,@DPC         ;START THE DISPLAY
2187
2188  012576  000240                          NOP
2189  012600  000240                          NOP
2190  012602  005077  166212          SPCEC:  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    #DNOP,FRM7A
2197  012624  012777  000001  166220          MOV    #1,@DPC         ;SINGLE STEP THE DISPLAY
2198  012632  022626                          CMP    (SP)+,(SP)+
2199  012634  000762                          BR     SPCEC
2200
2201
2202  012636  012656                  DSPTCH: FRAME0
2203  012640  012704                          FRAME1
2204  012642  012732                          FRAME2
2205  012644  012760                          FRAME3
2206  012646  013006                          FRAME4
2207  012650  013044                          FRAME5
2208  012652  013132                          FRAME6
2209  012654  013356                          FRAME7
2210
2211
```

```
2213
2214
2215  012656  117004                  FRAME0: POINT;INT4;LINE0
2216  012660  041600                          INTX+1600
2217  012662  000600                          600
2218  012664  041000                          INTX+1000
2219  012666  000600                          600
2220  012670  112400                          LONGV;INT2
2221  012672  040600                          INTX+600
2222  012674  000000                          0
2223  012676  173400                          DSTOP
2224  012700  160000                          DJMP
2225  012702  012656                          FRAME0
2226
2227  012704  117004                  FRAME1: POINT;INT4;LINE0
2228  012706  040200                          INTX+200
2229  012710  000600                          600
2230  012712  041000                          INTX+1000
2231  012714  000600                          600
2232  012716  112400                          LONGV;INT2
2233  012720  060600                          INTX;MINUSX+600
2234  012722  000000                          0
2235  012724  173400                          DSTOP
2236  012726  160000                          DJMP
2237  012730  012704                          FRAME1
2238
2239  012732  117004                  FRAME2: POINT;INT4;LINE0
2240  012734  041000                          INTX+1000
2241  012736  001200                          1200
2242  012740  041000                          INTX+1000
2243  012742  000600                          600
2244  012744  112400                          LONGV;INT2
2245  012746  040000                          INTX
2246  012750  000400                          400
2247  012752  173400                          DSTOP
2248  012754  160000                          DJMP
2249  012756  012732                          FRAME2
2250
2251  012760  117004                  FRAME3: POINT;INT4;LINE0
2252  012762  041000                          INTX+1000
2253  012764  000200                          200
2254  012766  041000                          INTX+1000
2255  012770  000600                          600
2256  012772  112400                          LONGV;INT2
2257  012774  040000                          INTX
2258  012776  020400                          MINUSX+400
2259  013000  173400                          DSTOP
2260  013002  160000                          DJMP
2261  013004  012760                          FRAME3
2262
```

```
2264
2265  013006  117004          FRAME4: POINT:INT4:LINE0
2266  013010  000000                  0
2267  013012  000000                  0
2268  013014  110000                  LONGV
2269  013016  041777                  INTX:MAXX
2270  013020  000000                  0
2271  013022  040000                  INTX
2272  013024  001377                  MAXY
2273  013026  061777                  INTX:MINUSX:MAXX
2274  013030  000000                  0
2275  013032  040000                  INTX
2276  013034  021377                  MINUSX:MAXY
2277  013036  173400                  DSTOP
2278  013040  160000                  DJMP
2279  013042  013006                  FRAME4
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  000760                  760
2295  013100  000620                  620
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
2311  013132  117004                FRAME6: POINT:INT4:LINE0
2312  013134  000200                        200
2313  013136  000600                        600
2314  013140  100000                        CHAR
2315
 (1)  013142  040500  041502  042504        .ASCII  ' @ABCDEFGHIJKLMNOPQRSTUVWXYZ!
 (1)  013150  043506  044510  045512
 (1)  013156  046514  047516  050520
 (1)  013164  051522  052524  053526
 (1)  013172  054530    132
 (1)  013175    040  021041  022043        .ASCIZ   @ ["#$%&'()*+,-./0123456789:;<=>?@
 (1)  013202  023045  024047  025051
 (1)  013210  026053  027055  030057
 (1)  013216  031061  032063  033065
 (1)  013224  034067  035071  036053
 (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   16,0,1,2,3,4,5,6,7,10,11,12,13,14,15,16
 (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   20,21,22,23,24,25,26,27,30,31,32,33,34,35,36,37,17,0
 (1)  013331    023    024    025
 (1)  013334    026    027    030
 (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  013356  117004                FRAME7: POINT:INT4:LINE8
2329  013360  000470                        400
2330  013362  000700                        700
2331  013364  170200                        STATSAI:LPLITE
2332  013366  110140                        LONGVI:LPON
2333  013370  041000                        INTX+1000
2334  013372  000000                        0
2335  013374  114000                        POINT
2336  013376  000400                        400
2337  013400  000600                        600
2338  013402  170300                        STATSAI:LPDARK
2339  013404  110140                        LONGVI:LPON
2340  013406  041000                        INTX+1000
2341  013410  000000                        0
2342  013412  114000                        POINT
2343  013414  000400                        400
2344  013416  000500                        500
2345  013420  110100                        LONGVI:LPOFF
2346  013422  041000                        INTX+1000
2347  013424  000000                        0
2348  013426  173400                 PRM7A: DSTOP
2349  013430  114000                        POINT
2350  013432  000700                        700
2351  013434  001000                        1000
2352  013436  100000                        CHAR
2353  013440  044514  044107  026524        .ASCIZ  /LIGHT-BEN HIT/
      013446  042520  020116  044510
      013454  000124
2354                                        .EVEN
2355  013456  173400                        DSTOP
2356  013460  160000                        DJMP
2357  013462  013356                        FRAME7
2358
2359  013464  000000                 BUFFER: 0
2360
2361          000001                        .END
```

```
BLKOFF = 000020      BLKON  = 000030      BRLEV1   011756      BRLEV2   011760
BUFFER   013464      CHAR   = 100000      CHSIZE = 001050      CNTR     001044
DBUF     001022      DBUF1    001024      DBUF2    001026      DBUF3    001030
DBUF4    001032      DBUF5    001034      DDONE    001062      DDONE1   001064
DISPLA = 177570      DJMP   = 160000      DLAY     013340      DLAYA    013344
DLAY1    013352      DLAY1A   013356      DNOP   = 140000      DOCORE   001252
DRC      001052      DSAVE    001036      DSPBR    001004      DSPTCH   013636
DSR      001050      DSTOP  = 173400      END      013204      FRAME0   013656
FRAME1   013704      FRAME2   013732      FRAME3   013760      FRAME4   013000
FRAME5   013044      FRAME6   013132      FRAME7   013356      PRM7A    013426
GRAPHX = 120000      GRAPHY = 124000      GREEN  = 000002      GBADD    001000
GSCHSZ   001006      GSLFSZ   001010      GSEND    001014      GSVCT    001002
GSYAXS   001012      GTBUSS   001364      GTPC     001412      GTPCA    001456
GT0      001932      GT1      001570      GT10     002240      GT11     002312
GT12 ·   002356      GT13     002422      GT14     002460      GT15     002526
GT16     002566      GT17     002632      GT18     002676      GT19     002742
GT2      001626      GT20     003004      GT21     003052      GT22     003116
GT23     003162      GT24     003226      GT25     003272      GT26     003336
GT27     003422      GT28     003474      GT29     003540      GT3      001664
GT30     003620      GT31     003672      GT32     003776      GT33     004060
GT34     004156      GT35     004254      GT36     004352      GT37     004450
GT38     004452      GT39     004654      GT39A    004670      GT4      001726
GT40     004570      GT40A    005010      GT41     005112      GT42     005210
GT43     005306      GT43A    005326      GT44     005430      GT44A    005450
GT45     005552      GT46     005650      GT47     005746      GT47A    005766
GT48     006070      GT48A    006110      GT49     006212      GT49A    006234
GT5      001770      GT50     006326      GT50A    006350      GT50B    006446
GT51     006470      GT52     006542      GT53     006702      GT54     007040
GT55     007200      GT56     007270      GT57     007346      GT58     007424
GT59     007502      GT6      007626      GT60     007560      GT61     007710
GT62     010056      GT62A    010216      GT63     010356      GT63A    010516
GT64     010632      GT65     010710      GT65A    010740      GT66B    010770
GT66     011012      GT66A    011040      GT67     011114      GT68     011226
GT68A    011312      GT69     011314      GT69A    011402      GT69B    011410
GT69C    011420      GT7      007672      GT70     011424      GT70A    011512
GT70B    011530      GT70C    011544      GT71     011556      GT71A    011610
GT72     011630      GT72A    011676      GT73     011762      GT73A    012030
GT74     012032      GT74A    012100      GT75     012114      GT76     012136
GT9      002202      HERE     012226      HERE1    012256      HIGPWR   012414
ICNT     001016      INCR   = 000100      INTX   = 040000      INT0   = 002000
INT1   = 002200      INT2   = 002400      INT3   = 002600      INT4   = 003000
INT5   = 003200      INT6   = 003400      INT7   = 003600      ITAL0  = 000040
ITAL1  = 000060      LFSIZE   001046      LINE0  = 000004      LINE1  = 000006
LINE2  = 000006      LINE3  = 000007      LINE8  = 000004      LOGIC0   001240
LOWPWR   012364      LOWSV    012462      LOGICA   012240      LONGV  = 110000
LPOFF  = 000100      LPON   = 000140      LPDARK = 000300      LPLITE = 000200
LPCFF  = 000100      MAXSY  = 000077      LPVCT    001066      LPVCT1   001070
MAXSX  = 017600      MAXXY  = 000077      MAXX   = 001777      MAXY   = 001377
MINSUY = 000100      MINUSX = 020000      MINUSY = 020000      PC     =X000007
POINT  = 114000      PSW      001020      RED    = 000003      RELATV = 130000
R0     =X000000      R1     =X000001      R2     =X000002      R3     =X000003
R4     =X000004      R5     =X000005      SCOPE  = 100000      SCOPEA   012320
SCOPEB   012332      SETUP    001076      SETUPA   001106      SETUPB   001132
SHORTV = 124000      SIZE     001042      SIZE0  = 000400      SIZE1  = 000600
SP     =X000006      SPACEB   012616      SPCEA    012542      SPCEC    012602
```

START   001336   START1   012464   STATSA = 170000   STATSB = 174000
STKPTR = 000500   SWR   001040   SYNOFF = 000010   SYNON = 000014
TIMEVT   001072   TMEVT1   001074   TPCSR = 177564   TPDBR = 177566
XPOS   001056   YPOS   001060   .   = 013466

ERRORS DETECTED! 0

CREM   .

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DDGTG-A - D
PRODUCT NAME:	GT40/GT44 VISUAL DISPLAY TEST WITH VR17 DISPLAY
DATE CREATED:	NOVEMBER 15 1973
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	RAYMOND SHOOP

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   EQUIPTMENT

        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 0200
        START WITH SWITCHES 7=0, 8=0 FOR AUTO SEQUENCING
        THRU ALL NON-OPERATOR INVERVENTION PATTERNS.
        START WITH SWITCH BIT 7=0, 8=1 FOR SWITCH REGISTER PATTERN
        CONTROL (REF 4.2).
        START WITH SWITCH BIT 7=1, 8=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 REPEATIBILITY
           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 <HORIZ>
           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 REPEATIBILITY
C	PINCUSHION (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 'CR' AFTER SELECTING A FRAME WILL
SELECT SUB-PICTURE 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   PINCUSHION TEST

         SW 6 = 0   DISPLAY PINCUSHION
         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   GRAPHPLOT INCREMENT TEST

         SW 6 = 0   USE GRAPHPLOT X
         SW 6 = 1   USE GRAPHPLOT 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 CORNOR AND A DOT IN THE
         CENTER OF THE SCREEN. THIS TEST IS USED TO VERIFY DOT REPEATIBILITY.

9.3   PINCUSHION 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 DRAWNED USING
         DIFFERENT VECTOR LENGTHS (7,17,37,77,177,377 AND 777).
         THIS TEST IS USED TO TEST THE END POINT MATCHING OF THE VECTORS.

9.5   CHARACTER SET (ADJUSTMENT OF THE CHARACTER POT'S)

        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 DRAWNED
        FROM THE SCREEN ORGIN TO THE OPPOSITE EDGE OF THE SCREEN,
        THESE VECTORS SHOULD TERMINATE ON THE LINE DRAWNED AT THE
        VIEWING EDGE, IF THE VECTORS DO NOT END ON THE LINE, ADJUST
        THE APPROPIATE 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 SCOPE,

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
        VISABLE. 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 DRAWNED 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 INTINIFIED 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
REPEATIBILITY 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 OUTTER OCTAGON IS DRAWN USING SHORT VECTOR MOODE
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   GRAPHPLOT INCREMENT TEST

A SERIES OF POINTS ARE PLOTTED WITH EACH POSSIBLE VALUE IN THE GRAPHPLOT
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 DISPLABLE CHARACTER
KEY IS DEPERSSED THE CHARACTER IS DISPLAYED ON THE SCREEN,
IN SELECTING THE SHIFT-OUT MODE, IF THE KEY DEPRESSED
IS NOT A CONTROL CHARACTER, THE PROGRAM WILL TRAP TO
THE SHIFT-OUT VECTOR, AN OCIAL CHARACTER VALUE READOUT
IS ALSO DISPLAYED AS AN AID IN ADJUSTING THE TTY CLOCK,

368
369
370

.LIST

```
372
373                                          .ENABL  ABS,AMA
374                                          .TITLE  GT-40/GT-44 WITH VR17 VISUAL DISPLAY TEST MAINDEC-11-DDGTG-A
375                                          .LIST   ME
376                                          .NLIST  MC,MD,CND
377
378          000000                          R0=%0
379          000001                          R1=%1
380          000002                          R2=%2
381          000003                          R3=%3
382          000004                          R4=%4
383          000005                          R5=%5
384          000006                          SP=%6
385          000007                          PC=%7
386          104000                          SCOPE=EMT
387          000500                          STKPTR=500
388          177570                          DISPLAY=177570          ;11/45 LIGHT DISPLAY REGISTER
389
390                                          10-776 IS FILLED WITH .+2, HALT
399                                          .LIST
400
401          000024                          .=24
402  000024  001250                          .WORD   LOWPWR
403  000026  000340                          340
404
405          000030                          .=30
406  000030  001100                          .WORD   SCOPEA      ;EMT RETURN
427  000032  000340                          340
428
```

```
410
411          000200                          .=200
412  000200  000137  001356                  JMP     START       ;DISPLAY TEST
413
414          001000                          .=1000
415  001000  172000                  GSADD:  172000              ;DISPLAY STARTING ADDRESS
416  001002  000320                  GSVCT:  320                 ;DISPLAY INTERRUPT VECTOR STARTING ADDRESS
417  001004  000200                  GSBRL:  200                 ;DISPLAY BR LEVEL
418
419  001006  000000                  ICNT:   0
420  001010  177776                  PSW:    177776
421  001012  177560                  TKS:    177560
422  001014  177562                  TKB:    177562
423  001016  012470                  DBUF:   BUFFER              ;FIRST WORD IN THE DISPLAY BUFFER
424  001020  012472                  DBUF1:  BUFFER+2            ;SECOND WORD
425  001022  012474                  DBUF2:  BUFFER+4            ;THIRD WORD
426  001024  012476                  DBUF3:  BUFFER+6            ;FOURTH WORD
427  001026  012500                  DBUF4:  BUFFER+10           ;FIFTH WORD
428  001030  012502                  DBUF5:  BUFFER+12
429  001032  000000                  DSAVE:  0                   ;TEMP REG.
430  001034  000000                  DSAVE1: 0
431  001036  000000                  DSAVE2: 0
432  001040  000000                  DSAVE3: 0
433  001042  000000                  HOLD:   0
434  001044  000000                  TSAVE:  0
435  001046  000000                  CNTR:   0
436  001050  000000                  CHANGE: 0
437  001052  000000                  LOKRB:  0
438
439
443
444                                  ;GS ADDRESSES AND INTERRUPT VECTORS
445
446  001054  172000                  DPC:    172000              ;DISPLAY PROGRAM COUNTER
447  001056  172002                  DSR:    172002              ;DISPLAY STATUS REGISTER
448  001060  172004                  XPOS:   172004              ;DISPLAY X AXIS REGISTER
449  001062  172006                  YPOS:   172006              ;DISPLAY Y AXIS REGISTER
450
451  001064  000320                  DDONE:  320                 ;DISPLAY INTERRUPT VECTOR FOR STOP
452  001066  000322                  DDONE1: 322
453
454  001070  000324                  LPVCT:  324                 ;DISPLAY INTERRUPT VECTOR FOR LIGHT-PEN
455  001072  000326                  LPVCT1: 326
456
457  001074  000330                  TIMEVT: 330                 ;DISPLAY INTERRUPT VECTOR FOR TIME-OUT OR SHIFT-OUT
458  001076  000332                  TMEVT1: 332
459
```

```
461                                    ;MONITOR ROUTINE
462
463   001100  005737  002046          SCOPEA: TST     KRBU             ;TEST IF SW OR "KRB"
464   001104  001014                          BNE     SCOPEF           ;BR IF "KRB"
465   001106  005037  005556                  CLR     SWITCH           ;CLEAR "SWITCH"
466   001112  032737  000100  177570          BIT     #100,@#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  SCOPEE: BIT     #400,@#DISPLAY   ;TEST BIT 8
470   001134  001010                          BNE     SCOPEB
471   001136  005737  001042          SCOPEF: TST     HOLD             ;TEST FOR "HOLD/STOP"
472   001142  001012                          BNE     SCOPED           ;BR IF SET
473   001144  000240                          NOP
474   001146  004737  001536                  JSR     PC,SETUP         ;RESET HOUSEKEEPING
475   001152  000240                          NOP
476   001154  000002                          RTI                      ;EXIT
477   001156  013704  177570          SCOPEB: MOV     @#DISPLAY,R4     ;READ SWITCHES
478   001162  042704  177760          SCOPEC: BIC     #177760,R4       ;MASK TO BITS 4-15
479   001166  006304                          ASL     R4               ;MOVE LEFT
480   001170  012706  000500          SCOPED: 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     @DISPTC(R4)      ;JMP TO THAT TEST
485
486   001210  002052          DISPTC: FILE0+2                  ;DIRECTORY
487   001212  002064                  FILE1+2                  ;DOT REPEATIBILITY
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                 ;GRAPHPLOT TEST
500   001244  004344                  FILE16+2                 ;LIGHT-PEN FOLLOW
501   001246  005054                  FILE17+2                 ;KEY BOARD ECHO
502
```

```
504
505
506   001250  010046          LOWPWR: 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,LOWSV
513   001270  012737  001302  000024  MOV     #HIGPWR,@#24
514   001276  000000                  HALT
515
516   001300  000000          LOWSV:  0
517
518   001302  013706  001300  HIGPWR: MOV     LOWSV,SP
519   001306  012605                  MOV     (SP)+,R5
520   001310  012604                  MOV     (SP)+,R4
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     #LOWPWR,@#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     SCOPED
```

```
536  001356  012706  000500          START:  MOV    #STKPTR,SP    ;SET UP THE STACK
537  001362  012777  000340  177420          MOV    #340,@PSW     ;RAISE PSW
538  001370  012700  001054                  MOV    #DRG,R0       ;GET POINTER
539  001374  013701  001000                  MOV    GSADD,R1      ;GET SUPPLIED ADDRESS
540  001400  010120          STRA:   MOV    R1,(0)+       ;UPDATE
541  001402  062701  000002                  ADD    #2,R1         ;THE
542  001406  022700  001064                  CMP    #DRG+10,R0    ;ADDRESSES
543  001412  001372                          BNE    STRA          ;UNTIL DONE
544  001414  012700  001064                  MOV    #DDONE,R0     ;GET POINTER
545  001420  013701  001002                  MOV    GSVCT,R1      ;GET SUPPLIED VECTOR
546  001424  010120          STRB:   MOV    R1,(0)+       ;UPDATE
547  001426  062701  000002                  ADD    #2,R1         ;THE VECTORS
548  001432  022700  001100                  CMP    #DDONE+14,R0
549  001436  001372                          BNE    STRB
550  001440  005037  005556                  CLR    SWITCH        ;HOUSEKEEP
551  001444  005037  001042                  CLR    HOLD          ;
552  001450  005004                          CLR    R4            ;
553  001452  005037  001044                  CLR    TSAVE         ;
554  001456  004737  001536          STRC:   JSR    PC,SETUP      ;SET UP VECTORS
555  001462  005037  001042                  CLR    HOLD
556  001466  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    #30060,DLT14A ;INITIALIZE X READOUT
559  001510  012737  030060  011766          MOV    #30060,DLT14A+2
560  001516  012737  030060  011776          MOV    #30060,DLT14B ;INITIALIZE Y READOUT
561  001524  012737  030060  012000          MOV    #30060,DLT14B+2
562  001532  000137  002050                  JMP    FILE0         ;START THE TEST
563
564  001536  012737  000062  000060  SETUP:  MOV    #62,@#60      ;RESET KRB VECTOR
565  001544  012737  000000  000062          MOV    #0,@#62
566  001552  042777  000100  177232          BIC    #100,@TKS     ;CLEAR INT ENABLE
567  001560  005037  002046                  CLR    KRBD
568  001564  032737  000200  177570          BIT    #200,@#DISPLAY ;TEST FOR "KRB" CONTROL
569  001572  001413                          BEQ    SETUPA        ;BR IF NOT
570  001574  005137  002046                  COM    KRBD          ;SET "KRB" CONTROL
571  001600  012737  001700  000060          MOV    #RETB,@#60    ;SET UP "KRB" INT
572  001606  012737  000340  000062          MOV    #340,@#62
573  001614  052777  000100  177170          BIS    #100,@TKS     ;ENABLE "KRB" INT
574  001622  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,@LPVCT ;RESET LIGHT-PEN VECTOR
577  001644  005077  177222                  CLR    @LPVCT1
578  001650  013777  001076  177216          MOV    TMEVT1,@TIMEVT ;RESET TIME-OUT/SHIFT OUT VECTOR
579  001656  005077  177214                  CLR    @TMEVT1
580  001662  000207                          RTS    PC            ;EXIT
581
582  001664  005777  177166          SETUPB: TST    @DSR          ;TEST FOR STOP
583  001670  100401                          BMI    .+4
584  001672  000000                          HALT                 ;ERROR, INTERRUPT OCCURRED TO THE STOP
585                                                                ;VECTOR BUT STOP WAS NOT SET
586  001674  000002                          RTI
587  001676  000000                          HALT
```

```
589
590
591  001700  117737  177110  001044  RETB:   MOVB   @TKB,TSAVE    ;READ THE CHARACTER
592  001706  042737  177600  001044          BIC    #177600,TSAVE ;MASK TO 7 BITS
593  001714  022737  000015  001044          CMP    #15,TSAVE     ;TEST FOR "CR"
594  001722  001440                          BEQ    KYT3          ;BR IF
595  001724  005037  005556                  CLR    SWITCH        ;CLEAR "SWITCH"
596  001730  162737  000101  001044          SUB    #101,TSAVE    ;MAKE 0-77
597  001736  100426                  KYT5:   BMI    KYT1          ;<A
598  001740  022737  000017  001044          CMP    #17,TSAVE
599  001746  100412                          BMI    KYT2          ;>P
600  001750  013704  001044                  MOV    TSAVE,R4
601  001754  012737  177777  001050          MOV    #-1,CHANGE
602  001762  005037  005556                  CLR    SWITCH
603  001766  005037  001042                  CLR    HOLD
604  001772  000002                          RTI                  ;EXIT
605  001774  022737  000076  001044  KYT2:   CMP    #76,TSAVE
606  002002  001015                          BNE    KYT4
607  002004  012737  177777  001042          MOV    #-1,HOLD      ; RUBOUT
608  002012  000002                          RTI                  ;EXIT
609  002014  005037  001042          KYT1:   CLR    HOLD
610  002020  000002                          RTI
611  002022  000000                          HALT                 ;FATAL ERROR RTI FAILED
612
613  002024  012737  177777  005556  KYT3:   MOV    #-1,SWITCH
614  002032  000002                          RTI
615  002034  000000                          HALT                 ;FATAL ERROR, RTI FAILED
616
617  002036  162737  000040  001044  KYT4:   SUB    #40,TSAVE     ;CONVERT LC TO UC
618  002044  000734                          BR     KYT5
619  002046  000000                  KRBD:   0
620
```

```
685                                       .LIST
689                             ;EXECUTE DIRECTORY FRAME
690
691  022050  104000            FILE0:  SCOPE
692  022052  004537  005412            JSR     5,MSG           ;EXIT TO DISPLAY A FRAME
693  022056  001000                    1000
694  022060  005560                    FRME0                   ;USING THE DIR. FRAME
695
696                             ;EXECUTE DOT REPEATIBILITY FRAME
697
698  022062  104000            FILE1:  SCOPE
699  022064  004537  005412            JSR     5,MSG           ;EXIT TO DISPLAY A FRAME
700  022070  100000                    100000
701  022072  007140                    FRME1                   ;USING THE DOR REPEAT FRAME
702
703                             ;EXECUTE PINCUSHION FRAME
704
705  022074  104000            FILE2:  SCOPE
706  022076  012700  012470            MOV     #BUFFER,R0      ;LOAD START ADDRESS
707  022102  004737  002252            JSR     PC,SETPNT       ;LOAD 0,0 ORGIN
708  022106  012701  000020            MOV     #20,R1          ;SETUP COUNT
709  022112  012720  040000    1S:     MOV     #INIX,(R0)+     ;LOAD INT LINE
710  022116  012720  001777            MOV     #MAXY,(R0)+     ; MAX Y
711  022122  012720  000100            MOV     #100,(R0)+      ;LOAD DELTA X
712  022126  012720  021777            MOV     #MINUSX+MAXY,(R0)+  ;LOAD - MAX Y
713  022132  005301                    DEC     R1              ;FINISHED ?
714  022134  001366                    BNE     1S              ;BR IF NOT
715  022136  012720  020001            MOV     #MINUSX+1,(R0)+ ;GO BACK 1 UNIT
716  022142  012720  000000            MOV     #0,(R0)+
717  022146  012720  040000            MOV     #INIX,(R0)+
718  022152  012720  001777            MOV     #MAXY,(R0)+     ;PLOT LAST LINE
719  022156  004737  002252            JSR     PC,SETPNT       ;SET ORGIN
720  022162  012701  000020            MOV     #MAXY+1/100,R1  ;SETUP COUNT
721  022166  012720  041777    2S:     MOV     #INIX+MAXX,(R0)+  ;LOAD DELTA X MAX
722  022172  012720  000000            MOV     #0,(R0)+        ;LOAD DELTA Y = 0
723  022176  012720  021777            MOV     #MINUSX+MAXX,(R0)+ ;RETRACE
724  022202  012720  000100            MOV     #100,(R0)+      ;LOAD DELTA Y OF 100
725  022206  005301                    DEC     R1              ;FINISHED ?
726  022210  001366                    BNE     2S              ;BR IF NOT
727  022212  012720  000000            MOV     #0,(R0)+
728  022216  012720  020001            MOV     #MINUSX+1,(R0)+
729  022222  012720  041777            MOV     #INIX+MAXX,(R0)+ ;PLOT LAST LINE
730  022226  012720  000000            MOV     #0,(R0)+
731  022232  012720  173400            MOV     #DSTOP,(R0)+    ;LOAD STOP
732  022236  012720  160000            MOV     #DJMP,(R0)+     ;LOAD JUMP
733  022242  012710  012470            MOV     #BUFFER,(R0)
734  022246  000137  002274            JMP     FILE2A
735
736  022252  012720  117000    SETPNT: MOV     #POINT+INT+,(R0)+  ;LOAD POINT
737  022256  012720  000000            MOV     #0,(R0)+        ; AT X
738  022262  012720  000000            MOV     #0,(R0)+        ; AT Y
739  022266  012720  110000            MOV     #LONGV,(R0)+    ;LONG VECTOR
740  022272  000207                    RTS     PC              ;EXIT
```

```
742
743  022274  012737  004000  001046  FILE2A: MOV     #4000,CNTR      ;LOAD COUNTER
744  022302  005737  005556          FILE2B: TST     SWITCH          ;TEST SWITCH
745  022306  001405                          BEQ     FILE2C          ;BR IF SUBTEST NOT SELECTED
746  022310  004537  005412                  JSR     R5,MSG          ;EXIT TO DISPLAY FRAME
747  022314  000001                          1
748  022316  012470                          BUFFER                  ;USING THE CROSS HATCH PATTERN
749  022320  000404                          BR      FILE2D          ;BR
750
751  022322  004537  005412          FILE2C: JSR     R5,MSG          ;EXIT TO DISPLAY FRAME
752  022326  000001                          1
753  022330  007230                          FRME2                   ;USING THE OFFSET PATTERN
754  022332  005337  001046          FILE2D: DEC     CNTR            ;FINISHED ?
755  022336  001361                          BNE     FILE2B          ;BR IF NOT
756
757                                  ;EXECUTE OCTAGONS OR SQUARES
758
759  022340  104000                  FILE3:  SCOPE
760  022342  012737  014000  001046          MOV     #14000,CNTR     ;SET UP A COUNTER
761  022350  005737  005556          FILE3A: TST     SWITCH
762  022354  001010                          BNE     FILE3B          ;BRANCH IF SUB-TEST
763  022356  004537  005412                  JSR     5,MSG           ;DISPLAY TEST
764  022362  000001                          1
765  022364  007334                          FRME3                   ;FRAME # 3
766  022366  005337  001046                  DEC     CNTR            ;DECREMENT COUNTER
767  022372  001366                          BNE     FILE3A          ;BRANCH IF NOT COMPLETE
768  022374  000407                          BR      FILE4           ;EXIT TO NEXT TEST
769
770  022376  004537  005412          FILE3B: JSR     5,MSG           ;DISPLAY TEST
771  022402  000001                          1
772  022404  007724                          FRME3A                  ;FRAME # 3A
773  022406  005337  001046                  DEC     CNTR            ;DECREMENT COUNTER
774  022412  001356                          BNE     FILE3A          ;BRANCH IF NOT COMPLETE
```

```
776
777                                      ;DISPLAY FILE
778                                      ;CHARACTER AND ITALICS TEST
779                                      ;SET UP THE BUFFER FOR THIS TEST
780
781   002414  104000            FILE4:  SCOPE
782   002416  012700  012470            MOV     #BUFFER,R0
783   002422  012720  174400            MOV     #STATSB!SIZE0,(0)+
784   002426  012720  170052            MOV     #STATSA!ITAL0!SYNOFF!GREEN,(0)+
785   002432  012720  117124            MOV     #POINT!INT4!LPOFF!BLKOFF!LINE0,(0)+      ;LOAD POINT MODE
786   002436  012720  000000            MOV     #0,(0)+
787   002442  012720  001700            MOV     #MAXY=77,(0)+
788   002446  012720  100000            MOV     #CHAR,(0)+
789   002452  112720  000017            MOVB    #17,(0)+
790   002456  112720  000017            MOVB    #17,(0)+
791   002462  012737  000100  002666     MOV    #100,STCHAR               ;LOAD INITIAL CHAR,
792   002470  004737  002624            JSR     PC,LOADBF
793   002474  012737  000140  002666     MOV    #140,STCHAR               ;LOAD INITIAL LC CHAR
794   002502  004737  002624            JSR     PC,LOADBF                 ;LOAD LINE
795   002506  012737  000040  002666     MOV    #40,STCHAR                ;LOAD NUMBERS AND PUNCT
796   002514  004737  002624            JSR     PC,LOADBF                 ;LOAD LINE
797   002520  012720  170040            MOV     #STATSA!ITAL0,(R0)+       ;LOAD NORMAL FONT
798   002524  004737  002564            JSR     PC,LOADSP                 ;LOAD SPECIAL CHARS
799   002530  004737  002730            JSR     PC,SPACE                  ;INSERT SPACES
800   002534  012720  170060            MOV     #STATSA!ITAL1,(R0)+       ;LOAD ITALICS FONT
801   002540  004737  002564            JSR     PC,LOADSP                 ;LOAD SPIECAL
802   002544  012720  173400            MOV     #DSTOP,(R0)+              ;LOAD DSTOP
803   002550  012720  160000            MOV     #DJMP,(R0)+
804   002554  012720  012470            MOV     #BUFFER,(R0)+
805   002560  000137  002746            JMP     FILE4A
806
807   002564  112720  000016     LOADSP: MOVB   #16,(R0)+
808   002570  012702  000000            MOV     #0,R2                     ;SET INITIAL SHIFT OUT CHAR
809   002574  012703  000037            MOV     #37,R3                    ;LOAD COUNT
810   002600  110220            1S:     MOVB    R2,(R0)+                  ;LOAD CHAR
811   002602  005202            2S:     INC     R2
812   002604  022702  000017            CMP     #17,R2                    ;TEST FOR SI
813   002610  001774                    BEQ     2S                        ;BR IF SI "17"
814   002612  005303                    DEC     R3                        ;FINISHED ?
815   002614  001371                    BNE     1S                        ;BR IF NOT
816   002616  012720  020017            MOV     #20!17,(R0)+              ;LOAD SHIFT-IN SPACE
817   002622  000207                    RTS     PC                        ;EXIT
818
819   002624  012720  170040     LOADBF: MOV    #STATSA!ITAL0,(R0)+       ;LOAD NORMAL FONT
820   002630  013702  002666            MOV     STCHAR,R2                 ;GET STARTING CHAR
821   002634  004737  002712            JSR     PC,FILLIT                 ;LOAD THE CHARACTERS
822   002640  004737  002730            JSR     PC,SPACE                  ;INSERT SPACES
823   002644  012720  170060            MOV     #STATSA!ITAL1,(R0)+       ;LOAD ITALICS FONT
824   002650  013702  002666            MOV     STCHAR,R2                 ;GET STARTING CHARACTER
825   002654  004737  002712            JSR     PC,FILLIT                 ;LOAD THE CHARACTERS
826   002660  004737  002670            JSR     PC,CRLF                   ;INSERT CR-LF
827   002664  000207                    RTS     PC                        ;EXIT
828
829   002666  000000            STCHAR: 0
```

```
830
831   002670  112720  000015     CRLF:   MOVB   #15,(0)+
832   002674  112720  000012            MOVB    #12,(0)+
833   002700  112720  000012            MOVB    #12,(0)+
834   002704  112720  000012            MOVB    #12,(0)+
835   002710  000207                    RTS     PC                        ;EXIT
836
837   002712  012703  000040     FILLIT: MOV    #40,R3
838   002716  110220            FILLA:  MOVB    R2,(0)+
839   002720  005202                    INC     R2
840   002722  005303                    DEC     R3
841   002724  001374                    BNE     FILLA
842   002726  000207                    RTS     7
843
844   002730  012703  000010     SPACE:  MOV    #10,R3
845   002734  112720  000040     1S:     MOVB   #40,(R0)+                 ;LOAD A SPACE
846   002740  005303                    DEC     R3
847   002742  001374                    BNE     1S                        ;BR IF NOT DONE
848   002744  000207                    RTS     PC                        ;EXIT
849
850                                      ;ACTUAL DISPLAY ROUTINE
851
852   002746  012737  001000  003022 FILE4A: MOV #1000,10S                ;LOAD A COUNTER
853   002754  012737  001700  012500   4S:  MOV  #MAXY=77,BUFFER+10       ;LOAD STARTING POINT
854   002762  004537  005412            JSR     R5,MESG
855   002766  000001                    1
856   002770  012470                    BUFFER
857
858   002772  012737  000400  012500     MOV    #40!,BUFFER+10
859   003000  004537  005412            JSR     R5,MESG
860   003004  000001                    1
861   003006  012470                    BUFFER
862
863   003010  005337  003022            DEC     10S                       ;FINISHED ?
864   003014  001357                    BNE     4S                        ;BR IF NOT
865   003016  000137  003024            JMP     FILE5                     ;GO TO NEXT TEST
866
867   003022  000000            10S:    0
868
869                                      ;EXECUTE DASH LINES AND BLINK
870
871   003024  104200            FILE5:  SCOPE
872   003026  004537  005412            JSR     5,MESG                    ;EXIT TO DISPLAY A FRAME
873   003032  010000                    10000
874   003034  012174                    FRME5                            ;USING THE DASH AND BLINK FRAME
```

```
876
877                                      ;EXECUTE VECTOR LENGTH TEST <HORIZ>
878
879  003036  104000                FILE6:  SCOPE
880  003040  012737  041777  010472        MOV   #INIXIMAXX,DELTX6  ;SET UP VERTICAL HEIGHT
881  003046  012737  000010  001036        MOV   #10,DSAVE2   ;SET UP TIMER
882  003054  012737  000000  001034        MOV   #0,DSAVE1
883  003062  012737  000040  001046  LOOPA:  MOV   #40,CNTR   ;SET UP EXECUTION COUNT
884  003070  012737  000200  001032  LOOPA1: MOV   #MAXY+1/10,DSAVE  ;SET UP
885  003076  013737  001034  010474        MOV   DSAVE1,DELTY6
886  003104  004537  005412        JSR   5,MESG   ;EXIT TO DISPLAY FRAME
887  003110  000001                1
888  003112  010426                FRME6    ;VECTOR LENGTH FRAME
889  003114  004537  005412  LOOPA2: JSR   5,MESG   ;EXIT TO DISPLAY FRAME
890  003120  000001                1
891  003122  010462                FRME6A   ;VECTOR LENGTH FRAME
892  003124  062737  000010  010474        ADD   #10,DELTY6   ;UPDATE ANGLE
893  003132  005337  001032        DEC   DSAVE   ;FINISHED ALL THE ANGLES
894  003136  001366        BNE   LOOPA2   ;BR IF NOT
895  003140  005337  001046  LOOPA3: DEC   CNTR   ;DONE COUNT?
896  003144  001351        BNE   LOOPA1   ;BR IF NOT
897  003146  000240        NOP
898  003150  005737  005556        TST   SWITCH   ;TEST SWITCH
899  003154  001342        BNE   LOOPA   ;BR IF HALT MOTION
900  003156  005237  001034        INC   DSAVE1   ;UPDATE INITIAL ANGLE
901  003162  005337  001036        DEC   DSAVE2   ;FINISHED ALL?
902  003166  001335        BNE   LOOPA   ;BR IF NOT
903
904                                      ;EXECUTE VECTOR LENGTH TEST <VERT>
905
906  003170  104000                FILE7:  SCOPE
907  003172  012737  040000  001034        MOV   #INIX,DSAVE1   ;SETUP INITIAL X
908  003200  012737  001777  010474        MOV   #MAXY,DELTY6   ;SETUP INITIAL Y
909  003206  012737  000010  001036        MOV   #10,DSAVE2   ;SETUP EXECUTION COUNT
910  003214  012737  000040  001046  LOOPB:  MOV   #40,CNTR   ;SETUP DELAY
911  003222  012737  000200  001032  LOOPB1: MOV   #200,DSAVE
912  003230  013737  001034  010472        MOV   DSAVE1,DELTX6  ;EXIT TO DISPLAY FRAME
913  003236  004537  005412        JSR   5,MESG   ;VECTOR LENGTH TEST FRAME
914  003242  000001                1
915  003244  010426                FRME6
916  003246  004537  005412  LOOPB2: JSR   5,MESG   ;EXIT TO DISPLAY FRAME
917  003252  000001                1
918  003254  010462                FRME6A   ;VECTOR LENGTH FRAME
919  003256  062737  000010  010472        ADD   #10,DELTX6   ;UPDATE ANGLE
920  003264  005337  001032        DEC   DSAVE   ;FINISHED ALL THE ANGLES
921  003270  001366        BNE   LOOPB2   ;BR IF NOT
922  003272  005337  001046  LOOPB3: DEC   CNTR   ;DONE COUNT?
923  003276  001351        BNE   LOOPB1   ;BR IF NOT
924  003300  000240        NOP
925  003302  005737  005556        TST   SWITCH   ;TEST SWITCH
926  003306  001342        BNE   LOOPB   ;BR IF HALT MOTION
927  003310  005237  001034        INC   DSAVE1   ;UPDATE INITIAL ANGLE
928  003314  005337  001036        DEC   DSAVE2   ;FINISHED ALL?
929  003320  001335        BNE   LOOPB   ;BR IF NOT
```

```
931
932                                      ;PHOSPHOR TEST <HORIZONTAL>
933
934  003322  104000                FILE10:  SCOPE
935  003324  005037  010506        CLR   DELTX7
936  003330  004537  005412  D7A:  JSR   5,MESG   ;EXIT TO DISPLAY A FRAME
937  003334  000050                50
938  003336  010504                FRME10   ;USING THE HORIZ FRAME
939  003340  004537  005412        JSR   5,MESG   ;EXIT TO DISPLAY A FRAME
940  003344  000001                1
941  003346  010604                FRM10   ;USING THE PERIMETER BOX
942  003350  000240        NOP
943  003352  005737  005556        TST   SWITCH   ;TEST THE "SWITCH"
944  003356  001364        BNE   D7A   ;BR IF FREEZE THE MOVEMENT
945  003360  062737  000001  010506  D7C:  ADD   #1,DELTX7   ;UPDATE THE X ORIGIN
946  003366  022737  002000  010506        CMP   #2000,DELTX7  ;TEST IF THE END
947  003374  001355        BNE   D7A   ;BR IF NOT
948
949                                      ;PHOSPHOR TEST  <VERTICAL>
950
951  003376  104000                FILE11:  SCOPE
952  003400  005037  010550        CLR   DELIY7
953  003404  004537  005412  D7D:  JSR   5,MESG   ;EXIT TO DISPLAY A FRAME
954  003410  000050                50
955  003412  010544                FRME11   ;USING THE VERT FRAME
956  003414  004537  005412        JSR   5,MESG   ;EXIT TO DISPLAY A FRAME
957  003420  000001                1
958  003422  010604                FRM10   ;USING THE PERIMETER BOX
959  003424  000240        NOP
960  003426  005737  005556        TST   SWITCH   ;TEST THE "SWITCH"
961  003432  001364        BNE   D7D   ;BR IF FREEZE THE MOVEMENT
962  003434  062737  000001  010550  D7F:  ADD   #1,DELTY7   ;UPDATE THE Y ORIGIN
963  003442  022737  002000  010550        CMP   #MAXY+1,DELTY7  ;TEST IF THE END
964  003450  001355        BNE   D7D   ;BR IF NOT
965
```

```
 967
 968                                      ;INTENSITY LEVEL TEST
 969
 970  003452  104000           FILE12: SCOPE
 971  003454  012777  003550  175406        MOV     #RETLP,@LPVCT       ;SET UP LIGHT-PEN VECTOR
 972  003462  013777  001004  175402        MOV     GSBRL,@LPVCT1       ;SET UP BR LEVEL
 973  003470  012737  004000  001032        MOV     #4000,DSAVE         ;SET UP A EXECUTION COUNT
 974  003476  005737  005556  FLE12A: TST   SWITCH              ;TEST THE "SWITCH"
 975  003502  001004           BNE     FLE12B              ;BR IF SET "SYNC"
 976  003504  042737  000004  010650        BIC     #4,SYN12            ;ENSURE CLEAR ?SYNC?
 977  003512  000403           BR      FLE12C              ;BY PASS
 978  003514  052737  000004  010650  FLE12B: BIS   #4,SYN12            ;SET THE "SYNC"
 979  003522  004537  005412  FLE12C: JSR   5,MESG              ;EXIT TO DISPLAY FRAME
 980  003526  000001           1
 981  003530  010642           FRME12              ;USING THE "INTENSITY" FRAME
 982  003532  005337  001032        DEC     DSAVE               ;FINISHED?
 983  003536  001423           BEG     FLE12D              ;YES, EXIT
 984  003540  012737  173400  011250        MOV     #DSTOP,RAYLPA       ;NO, RESET MESSAGE
 985  003546  000753           BR      FLE12A              ;BR BACK
 986  003550  012737  164000  011250  RETLP: MOV   #DNOP,RAYLPA        ;LIGHT-PEN HIT
 987  003556  017737  175300  011262        MOV     @YPOS,LPPNT         ;READ Y POSITION
 988  003564  042737  176000  011262        BIC     #176000,LPPNT       ;MASK THE BITS
 989  003572  022626           CMP     (SP)+,(SP)+         ;POP THE STACK
 990  003574  012777  000001  175252        MOV     #1,@DPC             ;SINGLE STEP THE DISPLAY
 991  003602  000137  005430        JMP     MESGA               ;JUMP TO WAIT
 992  003606  013777  001072  175254  FLE12D: MOV  LPVCT1,@LPVCT       ;RESET THE LIGHT-PEN VECTOR
 993
 994
 995                                      ;EXECUTE EDGE TEST
 996
 997  003614  104000           FILE13: SCOPE
 998  003616  004537  005412        JSR     5,MESG              ;EXIT TO DISPLAY FRAME
 999  003622  010000           10000
1000  003624  011312           FRME13              ;USING THE "EDGE" FRAME
```

```
1002
1003                                      ;SHORT VECTOR AND RELATIVE POINT TEST
1004
1005  003626  104000           FILE14: SCOPE
1006  003630  012700  012470        MOV     #BUFFER,R0          ;SET UP R0
1007  003634  012720  114000        MOV     #POINT,(0)+         ;SET UP INITIAL
1008  003640  012720  000240        MOV     #240,(0)+           ;X POSITION
1009  003644  012720  001000        MOV     #MAXY+1/2,(0)+      ;Y POSITION
1010  003650  012720  107004        MOV     #SHORTV!INT4!LINE0,(0)+ ;LOAD "SHORT VECTOR"
1011  003654  004537  003706        JSR     PC,LOADVT           ;LOAD THE DISPLAY PATTERN
1012  003660  012720  130000        MOV     #RELATV,(0)+        ;LOAD "RELATIVE POINT"
1013  003664  004537  003706        JSR     PC,LOADVT           ;LOAD THE DISPLAY PATTERN
1014  003670  012720  173400        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      FIL14A              ;BR TO THE FRAME
1018
1019  003706  012737  000024  001046  LOADVT: MOV  #24,CNTR            ;LOAD A COUNTER
1020  003714  012720  040077  LADVT: MOV   #INIX+77,(0)+       ;LOAD A DELTA Y
1021  003720  012720  004177        MOV     #4177,(0)+          ;LOAD A DELTA X,Y
1022  003724  005337  001046        DEC     CNTR                ;FINISHED?
1023  003730  001371           BNE     LADVT               ;BR IF NOT
1024  003732  000207           RTS     PC                  ;EXIT
1025
1026  003734  012737  004000  004104  FIL14A: MOV  #4000,10S           ;LOAD COUNTER
1027  003742  012737  000200  011572  1S:  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             ;DISPLAY 3RD OCT.
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  005337  004104        DEC     10S                 ;FINISHED ?
1051  004076  001321           BNE     1S                  ;BR IF NOT
1052  004100  000137  004106        JMP     FILE15              ;NEXT TEST
1053  004104  000000           10S: 0
1054
```

```
1056
1057                                    ;GRAPHPLOT X-Y TEST
1058
1059   004176  104000          FILE15: SCOPE
1060   004110  012700  012470          MOV     #BUFFER,R0        ;LOAD R0
1061   004114  012720  117600          MOV     #POINT!INT7,(0)+  ;LOAD INITIAL POINT
1062   004120  012720  000000          MOV     #0,(0)+
1063   004124  012720  000000          MOV     #0,(0)+
1064   004130  012720  170052          MOV     #STATSA!!TAL0!SYNOFF!GREEN,(R0)+ ;RESET THE STATUS A
1065   004134  012720  174100          MOV     #STATSB!INCR,(0)+ ;LOAD INITIAL STATUS B
1066   004140  012720  120000          MOV     #GRAPHX,(0)+      ;LOAD GRAPH X INST
1067   004144  012705  000040          MOV     #40,R5            ;LOAD STARTUP COUNT
1068   004150  012737  000000  001032  DFL15C: MOV   #0,DSAVE    ;LOAD INITIAL PLOT
1069   004156  000403          BR      2S
1070   004160  062737  000020  001032  1S:   ADD   #20,DSAVE     ;UPDATE PLOT POINT
1071   004166  013720  001032  2S:   MOV   DSAVE,(0)+    ;SAVE THE POINT
1072   004172  005305          DEC     R5                ;FINISHED?
1073   004174  001371          BNE     1S                ;BR IF NOT
1074   004176  012720  173400          MOV     #DSTOP,(0)+       ;LOAD "DSTOP"
1075   004202  012720  160000          MOV     #DJMP,(0)+        ;LOAD "DJMP"
1076   004206  012720  012470          MOV     #BUFFER,(0)+      ;LOAD RETURN
1077   004212  012737  000200  001032          MOV   #200,DSAVE  ;LOAD POINT COUNT
1078   004220  042777  004000  174602  DFL150: BIC   #4000,@DBUF5  ;ENSURE "GRAPHX"
1079   004226  005737  005556          TST     SWITCH            ;TEST SWITCH
1080   004232  001403          BEQ     DFL15B            ;BR IF GRAPHX
1081   004234  052777  004000  174566          BIS   #4000,@DBUF5  ;SET GRAPHY
1082   004242  004537  005412  DFL15B: JSR   5,MESG    ;EXIT TO DISPLAY A FRAME
1083   004246  000001          1
1084   004250  012470          BUFFER                    ;USING THE GENERATED PATTERN
1085   004252  062777  000001  174546          ADD   #1,@DBUF4   ;UPDATE INCREMENT
1086   004260  022777  174200  174540          CMP   #STATSB+200,@DBUF4  ;TEST IF LAST INCREMENT
1087   004266  001365          BNE     DFL15B            ;BR IF NOT
1088   004270  012777  174100  174530          MOV   #STATSB!INCR,@DBUF4  ;RELOAD INCREMENT
1089   004276  005537  001032          DEC     DSAVE             ;FINISHED 10 SEC?
1090   004302  001346          BNE     DFL15D            ;BR IF NOT
1091
1092   004304  013700  000042          MOV     @#42,R0
1093   004310  001407          BEQ     HERE              ;ACT-11/DDP-11
1094   004312  000005          RESET
1095   004314  000005          RESET
1096   004316  004710          LOGICAL: JSR    PC,(R0)
1097   004320  000240          NOP
1098   004322  000240          NOP
1099   004324  000240          NOP
1100   004326  000240          NOP
1101   004330  000137  002050  HERE:   JMP     FILE0
1102   004334  000240          NOP
1103   004336  000240          NOP
1104   004340  000240          NOP
```

```
1106
1107                                    ;OPERATOR OPERATOR INTERVENTION TESTS
1108
1109   004342  104000          FILE16: SCOPE
1110   004344  012777  004614  174516          MOV   #REI14,@LPVCT
1111   004352  013777  001004  174512          MOV   GSBAL,@LPVCT1
1112   004360  012737  000100  001034          MOV   #100,DSAVE1   ;SET UP COUNT
1113   004366  012700  012470  1S:   MOV   #BUFFER,R0    ;LOAD START ADDR,
1114   004372  012737  000100  001032          MOV   #100,DSAVE
1115   004400  012720  117144          MOV     #POINT!INT4!LPON!LINE0,(R0)+  ;LOAD POINT
1116   004404  012720  000700          MOV     #700,(R0)+        ;LOAD X POINT
1117   004410  012720  000474          MOV     #474,(R0)+        ;LOAD Y POINT
1118   004414  004737  004556          JSR     PC,LOADUP         ;LOAD UP THE BUFFER
1119   004420  012720  173400          MOV     #DSTOP,(R0)+      ;LOAD DSTOP
1120   004424  012720  160000          MOV     #DJMP,(R0)+       ;LOAD DJUMP
1121   004430  012720  012470          MOV     #BUFFER,(R0)+     ;LOAD RETURN ADDRESS
1122   004434  005037  005050          CLR     HITCNT            ;CLEAR HIT COUNT
1123   004440  012737  030060  012374          MOV   #30060,FRM16B-2  ;PRESET THE READOUT
1124   004446  012737  030060  012372          MOV   #30060,FRM16B-4
1125
1126   004454  005737  005556  4S:   TST   SWITCH        ;TEST SWITCH BIT
1127   004460  001005          BNE     6S                ;BR IF SUBTEST
1128
1129   004462  004537  005412          JSR     R5,MESG           ;EXIT TO DISPLAY FRAME
1130   004466  000100          100
1131   004470  011714          FRME16                    ;USINT THE LIGHT-PEN FRAME
1132   004472  000770          BR      4S                ;BR BACK
1133
1134   004474  004537  005412  6S:   JSR   R5,MESG       ;EXIT TO DISPLAY FRAME
1135   004500  000001          1
1136   004502  012302          FRM16A                    ;ASCII SUBTITLE
1137
1138   004504  004537  005412          JSR     R5,MESG           ;EXIT TO DISPLAY FRAME
1139   004510  000001          1
1140   004512  012470          BUFFER
1141
1142
1143   004514  005337  001032          DEC     DSAVE             ;FINISHED ?
1144   004520  001355          BNE     4S                ;BR IF NOT MINI-LOOP
1145
1146   004522  005337  001034          DEC     DSAVE1            ;FINISHED ?
1147   004526  001317          BNE     1S                ;BR IF NOT
1148   004530  000137  004342          JMP     FILE16            ;RESTART
1149
```

```
1151
1152  004534  012701  000030           LOADAC: MOV   #24,R1              ;LOAD COUNT
1153  004540  012720  130000                   MOV   #RELATV,(R0)+       ;LOAD RELATIVE POINT
1154  004544  012720  040004           1S:     MOV   #INTX+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   LOADUP: MOV   #24,CNTR            ;LOAD COUNT
1160  004564  004737  004534           1S:     JSR   PC,LOADAC           ;LOAD ADDRESS
1161  004570  012720  110000                   MOV   #LONGV,(R0)+        ;LOAD LONG VECTOR
1162  004574  012720  000004                   MOV   #4,(R0)+            ;LOAD VECTOR OVER
1163  004600  012720  020140                   MOV   #MINUSX+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   #176000,40S
1171  004630  017737  174224  004756           MOV   @XPOS,41S
1172  004636  042737  176000  004756           BIC   #176000,41S
1173  004644  005737  005556                   TST   SWITCH              ;TEST SW
1174  004650  001411                           BEQ   1S                  ;BR IF LIGHT PEN FOLLOW
1175  004652  005237  005050                   INC   HITCNT              ;UPDATE LIGHT PEN HIT COUNT
1176  004656  013703  005050                   MOV   HITCNT,R3           ;LOAD R3
1177  004662  012702  012376                   MOV   #FRM16B,R2          ;LOAD ADDRESS
1178  004666  004737  005332                   JSR   PC,KBCHR            ;CONVERT OCTAL
1179  004672  000432                           BR    20S                 ;BR
1180  004674  013703  004756           1S:     MOV   41S,R3              ;LOAD R3
1181  004700  012702  011770                   MOV   #DLI14A+4,R2        ;LOAD ADDRESS
1182  004704  004737  005332                   JSR   PC,KBCHR            ;LOAD X READOUT
1183  004710  013703  004754                   MOV   40S,R3              ;LOAD R3
1184  004714  012702  012002                   MOV   #DLI14B+4,R2        ;LOAD ADDRESS
1185  004720  004737  005332                   JSR   PC,KBCHR            ;LOAD Y READOUT
1186  004724  013737  004754  012006           MOV   40S,RAY14B          ;LOAD NEW Y POSITION
1187  004732  013737  004756  012004           MOV   41S,RAY14A          ;LOAD NEW X POSITION
1188  004740                           10S:
(1)   004740  012777  000001  174106           MOV   #1,@PC              ;SINGLE STEP THE DISPLAY
1189  004746  022626                           CMP   (SP)+,(SP)+
1190  004750  000137  005430                   JMP   MESGA
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  006200                           ASR   R0
1202  004776  006200                           ASR   R0
1203  005000  001404                           BEQ   30S
1204  005002  062701  000070           21S:    ADD   #70,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  006200                           ASR   R0
1211  005024  006200                           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   #INTX,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        ;LOAD R0
1225  005060  012720  173400    ECHOA:  MOV    #DSTOP,(0)+       ;MOV "DSTOPS"
1226  005064  022700  013470            CMP    #BUFFER+1000,R0   ;THRUOUT THE
1227  005070  001373                    BNE    ECHOA             ;BUFFER
1228  005072  005037  001052            CLR    LOKRB             ;HOUSE
1229  005076  005037  001032            CLR    DSAVE             ;KEEPING
1230  005102  112737  000060  012461    MOVB   #60,KBOCT-4
1231  005110  112737  000060  012462    MOVB   #60,KBOCT-3                ;PRESET READOUT
1232  005116  112737  000060  012463    MOVB   #60,KBOCT-2
1233  005124  112737  000060  012464    MOVB   #60,KBOCT-1
1234  005132  012737  005206  000060    MOV    #RET117,@#60      ;LOAD KEYBOARD VECTOR
1235  005140  012737  000340  000062    MOV    #340,@#62
1236  005146  052777  000100  173636    BIS    #100,@TKS         ;ENABLE INTERRUPT
1237  005154  012737  000700  005330    MOV    #700,CHRCNT       ;LOAD CHAR COUNT
1238  005162  012700  012470            MOV    #BUFFER,R0        ;RESET R0
1239  005166  004537  005412    ECHOC:  JSR    5,MESG            ;EXIT TO DISPLAY A FRAME
1240  005172  000001                    1
1241  005174  012404                    FRME17                   ;USING THE KEYBOARD HEADER
1242  005176  005737  001052            TST    LOKRB             ;UPDATE A CHAR?
1243  005202  001012                    BNE    RET01             ;BR IF YES
1244  005204  000770                    BR     ECHOC
1245  005206  017701  173602    RET117: MOV    @TKB,R1           ;GET A CHAR
1246  005212  042701  177600            BIC    #177600,R1        ;MASK
1247  005216  012737  177777  001052    MOV    #-1,LOKRB         ;SET (FLAG)
1248  005224  000002                    RTI                      ;EXIT
1249  005226  000000                    HALT
1250  005230  005037  001052    RET21:  CLR    LOKRB             ;CLEAR (FLAG)
1251  005234  022701  000003            CMP    #3,R1             ;TEST FOR PC
1252  005240  001002                    BNE    RET20             ;BR IF NOT
1253  005242  000137  001456            JMP    STRC              ;RESTART
1254  005246  005337  005330    RET20:  DEC    CHRCNT            ;FINISHED COUNT?
1255  005252  001002                    BNE    1S                ;BR IF NOT
1256  005254  000137  005054            JMP    FILE17+2          ;RESTART
1257  005260  012702  012465    1S:     MOV    #KBOCT,R2                 ;LOAD ADDRESS
1258  005264  010103                    MOV    R1,R3
1259  005266  004737  005332            JSR    PC,KBCHR                  ;LOAD THE OCTAL VALUE
1260  005272  005737  001032            TST    DSAVE             ;TEST HIGH/LOW BYTE
1261  005276  001007                    BNE    ECHOB
1262  005300  110120                    MOVB   R1,(R0)+          ;SAVE BYTE
1263  005302  112710  000017            MOVB   #17,(R0)          ;SHIFT-IN
1264  005306  005137  001032            COM    DSAVE             ;COMP FLAG
1265  005312  000137  005166            JMP    ECHOC             ;BR BACK
1266  005316  110120            ECHOB:  MOVB   R1,(0)+           ;SAVE CHAR
1267  005320  005037  001032            CLR    DSAVE             ;CLEAR FLAG
1268  005324  000137  005166            JMP    ECHOC             ;BR BACK
1269
1270
1271  005330  000200            CHRCNT: 200
```

```
1273                                    ;UPDATE OCTAL READOUT
1274
1275  005332  042703  176000    KBCHR:  BIC    #176000,R3        ;LOAD BITS
1276  005336  004737  005376            JSR    PC,10S
1277  005342  110442                    MOVB   R4,-(R2)          ;SAVE BITS
1278  005344  004737  005370            JSR    PC,11S            ;MOVE BITS
1279  005350  110442                    MOVB   R4,-(R2)          ;SAVE BITS
1280  005352  004737  005370            JSR    PC,11S            ;MOVE BITS
1281  005356  110442                    MOVB   R4,-(R2)          ;SAVE BITS
1282  005360  004737  005370            JSR    PC,11S            ;MOVE BITS
1283  005364  110442                    MOVB   R4,-(R2)
1284  005366  000207                    RTS    PC
1285  005370  006003            11S:    ROR    R3
1286  005372  006003                    ROR    R3
1287  005374  006003                    ROR    R3
1288  005376  010304            10S:    MOV    R3,R4             ;LOAD R4
1289  005400  042704  177770            BIC    #177770,R4        ;MASK BITS
1290  005404  062704  000060            ADD    #60,R4            ;MAKE A NUMBER
1291  005410  000207                    RTS    PC
1292
1293  005412  012537  005552    MESG:   MOV    (5)+,COUNT
1294  005416  012537  005554            MOV    (5)+,FILE
1295  005422  013777  005554  173424    MOV    FILE,@DPC         ;START DISPLAY
1296  005430  005077  173354    MESGA:  CLR    @PSR
1297  005434  000001                    WAIT
1298  005436  005737  002046            TST    KRB
1299  005442  001025                    BNE    MESGAB
1300  005444  005337  005552    MESGAA: DEC    COUNT
1301  005450  001405                    BEQ    MESGB
1302  005452  012777  000001  173374    MOV    #1,@DPC           ;SINGLE STEP THE DISPLAY
1303  005460  000137  005430            JMP    MESGA
1304  005464  000240            MESGB:  NOP
1305  005466  005737  002046            TST    KRB
1306  005472  001010                    BNE    MESGBA
1307  005474  005037  005556            CLR    SWITCH
1308  005500  032737  000100  177570    BIT    #BIT6,@#DISPLAY
1309  005506  001402                    BEQ    MESGBA
1310  005510  005137  005556            COM    SWITCH
1311  005514  000205            MESGBA: RTS    5
1312  005516  005737  005556    MESGAB: TST    SWITCH
1313  005522  001350                    BNE    MESGAA
1314  005524  005737  001050            TST    CHANGE
1315  005530  001745                    BEQ    MESGAA
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              FRME0:  POINT
1326  005562  000000                      0
1327  005564  001500                      MAXY=277
1328  005566  170052                      STATSA!ITALE!SYNOFF!GREEN
1329  005570  105124                      CHAR!INT4!LPOFF!BLKOFF!LINE0
1330  005572  017     017                 .BYTE  17,17
1331  005574  052107  032055  020060      .ASCIZ /GT-40 OR GT-44 WITH VR17 VISUAL TEST  <MD-11-DDGTG-A>/
      005602  051117  043440  026524
      005610  032064  053440  052111
      005616  020110  051126  033461
      005624  053040  051511  040525
      005632  020114  042524  052123
      005640  020040  046474  026504
      005646  030461  042055  043504
      005654  043524  040455  000076
1332  005662  015     012     012         .BYTE  15,12,12
1333  005665  040     020040  044504      .ASCII /  DIRECTORY/
      005672  042522  052103  051117
      005700  131
1334  005701  015     012     012         .BYTE  15,12,12
1335  005704  030060  036440  040440      .ASCII /00 = A = DIRECTORY/
      005712  036440  042040  051111
      005720  041505  047524  054522
1336  005726  015     012                 .BYTE  15,12
1337  005730  030460  036440  041040      .ASCII /01 = B = DOT REPEATIBILITY/
      005736  036440  042040  052117
      005744  051040  050105  040505
      005752  044524  044502  044514
      005760  054524
1338  005762  015     012                 .BYTE  15,12
1339  005764  031060  036440  041440      .ASCII /02 = C = PINCUSHION AND VECTOR CURVATURE <X OR Y OFFSET ADJ.>/
      005772  036440  050040  047111
      006000  052503  044123  047511
      006006  020116  047101  020104
      006014  042526  052103  051117
      006022  041440  051125  040526
      006030  052524  042522  036040
      006036  020130  051117  054440
      006044  047440  043106  042523
      006052  020124  042101  027112
      006060  076
1340  006061  015     012                 .BYTE  15,12
1341  006063  060     020063  020075      .ASCII /03 = D = OCTAGONS OR SQUARES/
      006070  020104  020075  041517
      006076  040524  047507  051516
      006104  047440  020122  050523
      006112  040525  042522  123
1342  006117  015     012                 .BYTE  15,12
1343  006121  060     020064  020075      .ASCII /04 = E = CHARACTER SET <CHAR, ADJ.>/
      006126  020105  020075  044103
      006134  051101  041501  042524
      006142  020122  042523  020124
      006150  041474  040510  027122
```

```
      006156  040440  045104  037056
1344  006164  015     012                 .BYTE  15,12
1345  006166  032460  036440  043040      .ASCII /05 = F = DASH LINES AND BLINK/
      006174  036440  042040  051501
      006202  020110  044514  042516
      006210  020123  047101  020104
      006216  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>/
      006232  020107  020075  047510
      006240  044522  047532  052116
      006246  046101  053040  041505
      006254  047524  020122  047101
      006262  046107  020105  040474
      006270  045104  020056  020130
      006276  042526  052103  051117
      006304  046040  047105  052107
      006312  037110
1348  006314  015     012                 .BYTE  15,12
1349  006316  033460  036440  044040      .ASCII /07 = H = VERTICAL VECTOR ANGLE <ADJ. Y VECTOR LENGTH>/
      006324  036440  053040  051105
      006332  044524  040503  020114
      006340  042526  052103  051117
      006346  040440  043516  042514
      006354  036040  042101  027112
      006362  054440  053040  041505
      006370  047524  020122  042514
      006376  043516  044124  076
1350  006403  015     012                 .BYTE  15,12
1351  006405  061     020060  020075      .ASCII /10 = I = HORIZONTAL PHOSPHOR TEST/
      006412  020111  020075  047510
      006420  044522  047532  052116
      006426  046101  050040  047510
      006434  050123  047510  020122
      006442  042524  052123
1352  006446  015     012                 .BYTE  15,12
1353  006450  030461  036440  045040      .ASCII /11 = J = VERTICAL PHOSPHOR TEST/
      006456  036440  053040  051105
      006464  044524  040503  020114
      006472  044120  051517  044120
      006500  051117  052040  051505
      006506  124
1354  006507  015     012                 .BYTE  15,12
1355  006511  061     020062  020075      .ASCII /12 = K = INTENSITY LEVEL AND LIGHT-PEN TEST/
      006516  020113  020075  047111
      006524  042524  051516  052111
      006532  020131  042514  042526
      006540  020114  047101  020104
      006546  044514  044107  026524
      006554  042520  020116  042524
      006562  052123
1356  006564  015     012                 .BYTE  15,12
1357  006566  031461  036440  046040      .ASCII /13 = L = EDGE FLAG TEST/
      006574  036440  042440  043504
```

```
            306602  020105  046106  043501
            406610  052040  051505     124
      1358  306615     015     012                    .BYTE   15,12
      1359  306617     061  020064  020075            .ASCII  /14 = M = SHORT VECTORS AND RELATIVE POINT/
            306624  020115  020075  044123
            306632  051117  020124  042526
            306640  052103  051117  020123
            306646  047101  020104  042522
            306654  040514  044524  042526
            306662  050040  044517  052116
      1360  306670     015     012                    .BYTE   15,12
      1361  306672  032461  036440  047040            .ASCII  /15 = N = GRAPHPLOT TEST/
            306700  036440  043440  040522
            306706  044120  046120  052117
            306714  052040  051505     124
      1362  306721     015     012                    .BYTE   15,12
      1363  306723     061  020066  020075            .ASCII  /16 = O = LIGHT PEN FOLLOW/
            306730  020117  020075  044514
            306736  044107  020124  042520
            306744  020116  047506  046114
            306752  053517
      1364  306754     015     012                    .BYTE   15,12
      1365  306756  033461  036440  050040            .ASCII  /17 = P = KEYBOARD ECHO TEST/
            306764  036440  045440  054505
            306772  047502  051101  020104
            307000  041505  047510  052040
            307006  051505     124
      1366  307011     015     012     012            .BYTE   15,12,12
      1367  307014  020040  052522  047502            .ASCII  / RUBOUT TO REMAIN ON THE PATTERN/
            307022  052125  052040  020117
            307030  042522  040515  047111
            307036  047440  020116  044124
            307044  020105  040520  052124
            307052  051105     116
      1368  307055     015     012                    .BYTE   15,12
      1369  307057     040  041440  020122            .ASCIZ  / CR TO SELECT SUB-PICTURE OR STOP MOTION /
            307064  047524  051440  046105
            307072  041505  020124  052523
            307100  026502  044520  052103
            307106  051125  020105  051117
            307114  051440  047524  020120
            307122  047515  044524  047117
            307130  000040
      1370                                            .EVEN
      1371  307132  173400                            DSTOP
      1372  307134  160000                            DJMP
      1373  307136  005560                            FRME0
      1374
      1375  307140                          FRME1:
       (1)  307140  170052                            STATSA!ITAL0!SYNOFF!GREEN
      1376  307142  116124                            POINT!INT0!LPOFF!BLKOFF!LINE0
      1377  307144  041000                            INTX+1000
      1378  307146  001000                            MAXY+1/2
      1379  307150  040000                            INTX+0
```

```
      1380  307152  000000                            0
      1381  307154  041000                            INTX+1000
      1382  307156  001000                            MAXY+1/2
      1383  307160  041777                            INTX+1777
      1384  307162  000000                            0
      1385  307164  041000                            INTX+1000
      1386  307166  001000                            MAXY+1/2
      1387  307170  041777                            INTX+1777
      1388  307172  001777                            MAXY
      1389  307174  041000                            INTX+1000
      1390  307176  001000                            MAXY+1/2
      1391  307200  040000                            INTX
      1392  307202  001777                            MAXY
      1393  307204  164000                            DNOP
      1394  307206  164000                            DNOP
      1395  307210  164000                            DNOP
      1396  307212  164000                            DNOP
      1397  307214  164000                            DNOP
      1398  307216  164000                            DNOP
      1399  307220  164000                            DNOP
      1400  307222  173400                            DSTOP
      1401  307224  160000                            DJMP
      1402  307226  007140                            FRME1
      1403
      1404                             ;FILE 2 <ANALOG TUNE-UP TEST >
      1405
      1406  307230  116524            FRME2:  POINT!INT2!LPOFF!BLKOFF!LINE0
      1407  307232  000000                            0
      1408  307234  000000                            0
      1409  307236  170052                            STATSA!ITAL0!SYNOFF!GREEN
      1410  307240  110000                            LONGV
      1411  307242  041777                            INTX!MAXX                  ; +X, +Y
      1412  307244  000000                            0
      1413  307246  040000                            INTX                       ; +X, +Y
      1414  307250  001777                            MAXY
      1415  307252  061777                            INTX!MINUSX!MAXX           ; -X, +Y
      1416  307254  000000                            0
      1417  307256  040000                            INTX                       ; +X, -Y
      1418  307260  021777                            MINUSY!MAXY
      1419  307262  041777                            INTX!MAXX                  ; +X, -Y
      1420  307264  020000                            MINUSY
      1421  307266  060000                            INTX!MINUSX                ; -X, +Y
      1422  307270  001777                            MAXY
      1423  307272  061777                            INTX!MINUSX!MAXX           ; -X, +Y
      1424  307274  020000                            MINUSY
      1425  307276  060000                            INTX!MINUSX                ; -X, -Y
      1426  307300  021777                            MINUSY!MAXY
      1427  307302  041777                            INTX!MAXX
      1428  307304  001777                            MAXY
      1429  307306  061777                            INTX!MINUSX!MAXX
      1430  307310  021777                            MINUSX!MAXY
      1431  307312  001777                            MAXX
      1432  307314  000000                            0
      1433  307316  061777                            INTX!MINUSX!MAXX
```

```
1434  007320  001777                    MAXY
1435  007322  041777                    INTX!MAXX
1436  007324  021777                    MINUSX!MAXY
1437  007326  173400                    DSTOP
1438  007330  160000                    DJMP
1439  007332  007230                    FRME2
1440
1441                         ;OCTAGONS
1442
1443  007334  117124        FRME3;      POINT;INT4;LPOFF;BLKOFF;LINE0
1444  007336  000774                    774
1445  007340  000564                    564
1446  007342  170052                    STATSA!ITALE!SYNOFF!GREEN
1466  007344  110000                    LONGV                        ;OCTOGON 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                    INTX!MINUSX+7
(1)   007364  000007                    7
(1)   007366  060007                    INTX!MINUSX+7
(1)   007370  000000                    0
(1)   007372  060007                    INTX!MINUSX+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                        ;OCTOGON 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                    INTX!MINUSX+17
(1)   007434  000017                    17
(1)   007436  060017                    INTX!MINUSX+17
(1)   007440  000000                    0
(1)   007442  060017                    INTX!MINUSX+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                        ;OCTOGON 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                    INTX!MINUSX+37
(1)   007504  000037                    37
(1)   007506  060037                    INTX!MINUSX+37
(1)   007510  000000                    0
(1)   007512  060037                    INTX!MINUSX+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                        ;OCTOGON 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                    INTX!MINUSX+77
(1)   007554  000077                    77
(1)   007556  060077                    INTX!MINUSX+77
(1)   007560  000000                    0
(1)   007562  060077                    INTX!MINUSX+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                        ;OCTOGON 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                    INTX!MINUSX+177
(1)   007624  000177                    177
(1)   007626  060177                    INTX!MINUSX+177
(1)   007630  000000                    0
(1)   007632  060177                    INTX!MINUSX+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                    600
  1485  007652  000000                    0
  1486  007654  110000                    LONGV                      ;OCTOGON 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  060377                    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  007334                    FRME3
  1490                            ;SQUARES   7,17,37,77,177,377,777 WIDE
  1491
  1492  007724  117124            FRME3A: POINT!INT4!LPOFF!BLKOFF!LINE0          ; BY 7
  1493  007726  001000                    1000
  1494  007730  000600                    600
  1495  007732  170052                    STATSA!ITAL&!SYNOFF!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)  010100  000000                    0
   (2)  010102  040000                    INTX
   (2)  010104  020177                    MINUSX+177
   (2)  010106  020077                    MINUSX+77
   (2)  010110  020077                    MINUSX+77
   (1)                                    .LIST
   (2)  010112  110000                    LONGV                      ;BY 377 AND 177
   (2)  010114  040377                    INTX+377
   (2)  010116  000000                    0
   (2)  010120  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)   010146   040000                     INTX
       (2)   010150   000777                     777
       (2)   010152   060777                     INTX!MINUSX*777
       (2)   010154   000000                     0
       (2)   010156   040000                     INTX
       (2)   010160   020777                     MINUSX*777
       (2)   010162   020377                     MINUSX*377
       (2)   010164   020377                     MINUSX*377
       (1)                                        .LIST
      1518   010166   173400                     DSTOP
      1519   010170   160000                     DJMP
      1520   010172   007724                     FRME3A
      1521
      1522                             ;DASH LINE TEST
      1523
      1524   010174   117000           FRME5:    POINT;INT4
      1525   010176   000000                     0
      1526   010200   001000                     1000
      1527   010202   174400                     STATSB!SIZE8
      1528   010204   170052                     STATSA!ITAL&!SYNOFF!GREEN
      1529   010206   100004                     CHAR!LINE0
      1530   010210      017      017            .BYTE   17,17
      1531   010212   047523   044514   020104   .ASCII /SOLID    /
             010220   020040   020040
      1532   010224   110004                     LONGV!LINE0
      1533   010226   040400                     40400
      1534   010230   000000                     0
      1535   010232   000400                     400
      1536   010234   000000                     0
      1537   010236   110030                     LONGV!BLKON
      1538   010240   040400                     40400
      1539   010242   000000                     0
      1540   010244   100020                     CHAR!BLKOFF
      1541   010246      015      012      012   .BYTE   15,12,12,12,12,12
             010251      012      012      012
      1542   010254   040504   044123   044440   .ASCII /DASH I    /
             010262   020040   020040
      1543   010266   110005                     LONGV!LINE1
      1544   010270   040400                     40400
      1545   010272   000000                     0
      1546   010274   000400                     400
      1547   010276   000000                     0
      1548   010300   110030                     LONGV!BLKON
      1549   010302   040400                     40400
      1550   010304   000000                     0
      1551   010306   100020                     CHAR!BLKOFF
      1552   010310      015      012      012   .BYTE   15,12,12,12,12,12
             010313      012      012      012
      1553   010316   040504   044123   044440   .ASCII /DASH II   /
             010324   020111   020040
      1554   010330   110006                     LONGV!LINE2
      1555   010332   040400                     40400
      1556   010334   000000                     0
      1557   010336   000400                     400
```

```
      1558   010340   000000                     0
      1559   010342   110030                     LONGV!BLKON
      1560   010344   040400                     40400
      1561   010346   000000                     0
      1562   010350   100020                     CHAR!BLKOFF
      1563   010352      015      012      012   .BYTE   15,12,12,12,12,12
             010355      012      012      012
      1564   010360   040504   044123   044440   .ASCII /DASH III  /
             010366   044511   020040
      1565   010372   110007                     LONGV!LINE3
      1566   010374   040400                     40400
      1567   010376   000000                     0
      1568   010400   000400                     400
      1569   010402   000000                     0
      1570   010404   110030                     LONGV!BLKON
      1571   010406   040400                     40400
      1572   010410   000000                     0
      1573   010412   110024                     LONGV!BLKOFF!LINE0
      1574   010414   000000                     0
      1575   010416   000000                     0
      1576   010420   173400                     DSTOP
      1577   010422   160000                     DJMP
      1578   010424   010174                     FRME5
      1579
      1580                             ;VECTOR LENGTH TEST <FILE 6 AND 7>
      1581
      1582   010426   114000           FRME6:    POINT
      1583   010430   001777                     MAXX
      1584   010432   000000                     0
      1585   010434   170052                     STATSA!ITAL&!SYNOFF!GREEN
      1586   010436   113724                     LONGV!INT7!LPOFF!BLKOFF!LINE0
      1587   010440   040000                     INTX
      1588   010442   001777                     MAXY
      1589   010444   114000                     POINT
      1590   010446   000000                     0
      1591   010450   001777                     MAXY
      1592   010452   110000                     LONGV
      1593   010454   041777                     INTX!MAXX
      1594   010456   000000                     0
      1595   010460   173400                     DSTOP
      1596   010462   114000           FRME6A:   POINT
      1597   010464   000000                     0
      1598   010466   000000                     0
      1599   010470   110000                     LONGV
      1600   010472   000000           DELTX6:   0
      1601   010474   000000           DELTY6:   0
      1602   010476   173400                     DSTOP
      1603   010500   160000                     DJMP
      1604   010502   010462                     FRME6A
      1605
      1606
      1607                             ;PHOSPHOR TEST
      1608
      1609   010504   114000           FRME10:   POINT
```

```
1610  010506  000000              DELTX7: 0
1611  010510  000000                      0
1612  010512  170052                      STATSA:ITAL0: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                              ;PHOSPHOR TEST
1627
1628  010544  114000              FRME11: POINT
1629  010546  000000                      0
1630  010550  000000              DELTY7: 0
1631  010552  170052                      STATSA:ITAL0:SYNOFF:GREEN
1632  010554  113724              DF111C: LONGV:INT7:LPOFF:BLKOFF:LINE0
1633  010556  041777                      INTX:MAXX
1634  010560  000000                      0
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                              ;INTENSITY TEST
1662
1663  010642  114164              FRME12: POINT:LINE0:LPON:BLKOFF
```

```
1664  010644  000000                      0
1665  010646  001200                      1200
1666  010650  170252              SYN12: STATSA:LPLI:E:SYNOFF::ITAL0:GREEN
1667  010652  103600                      CHAR:INT7
1668  010654    017    017               .BYTE  17,17
1669  010656  047111  042524  051516      .ASCII /INTENSITY 7   /
      010664  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        .BYTE 15,12,12,12
      010713    012
1677  010714  047111  042524  051516      .ASCII /INTENSITY 6   /
      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        .BYTE 15,12,12,12
      010751    012
1685  010752  047111  042524  051516      .ASCII /INTENSITY 5   /
      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  103000                      CHAR:INT4
1692  011004    015    012    012        .BYTE 15,12,12,12
      011007    012
1693  011010  047111  042524  051516      .ASCII /INTENSITY 4   /
      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        .BYTE 15,12,12,12
      011045    012
1701  011046  047111  042524  051516      .ASCII /INTENSITY 3   /
      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                          CHARIINT2
1708  011100     015     012     012          .BYTE 15,12,12,12
      011103     012
1709  011104  047111  042524  051516          .ASCII /INTENSITY 2   /
      011112  052111  020131  020062
      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                          CHARIINT1
1716  011136     015     012     012          .BYTE 15,12,12,12
      011141     012
1717  011142  047111  042524  051516          .ASCII /INTENSITY 1   /
      011150  052111  020131  020061
      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                          CHARIINT0
1724  011174     015     012     012          .BYTE 15,12,12,12
      011177     012
1725  011200  047111  042524  051516          .ASCII /INTENSITY 0   /
      011206  052111  020131  020060
      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            RAILPA: DSTOP
1740  011252  164000                          DNOP
1741  011254  164000                          DNOP
1742
1743  011256  117100            DF112A: POINTIINT4ILPOFF
1744  011260  001500                          1500
1745  011262  001200            LPPNT: 1200
1746  011264  100000                          CHAR
1747  011266  044514  044107  020124          .ASCIZ  /LIGHT PEN HIT/
      011274  042520  020116  044510
```

```
      011302  000124
1748                                          .EVEN
1749  011304  173400                          DSTOP
1750  011306  160000                          DJMP
1751  011310  010642                          FRME12
1752
1753                          ;EDGE FILE
1754
1755  011312  117124            FRME13: POINTIINT4ILPOFFIBLKOFFILINE0
1756  011314  000000                          0
1757  011316  000000                          0
1758  011320  170052                          STATSAIITAL0ISYN0FF IGREEN
1759  011322  100000                          CHAR
1760  011324     017     017                  .BYTE  17,17
1761  011326  110000                          LONGV
1762  011330  041777                          INTXIMAXX
1763  011332  000000                          0
1764  011334  040000                          INTX
1765  011336  001777                          MAXY
1766  011340  061777                          INTXIMINUSXIMAXX
1767  011342  000000                          0
1768  011344  040000                          INTX
1769  011346  021777                          MINUSYIMAXY
1770  011350  114000                          POINT
1771  011352  000100                          100
1772  011354  000300                          300            ;LEFT SIDE
1773  011356  110000                          LONGV
1774  011360  040000                          INTX
1775  011362  000400                          400
1776  011364  060200                          INTXIMINUSX*200
1777  011366  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                          INTXIMINUSX**400
1791  011422  000000                          0
1792  011424  040000                          INTX
1793  011426  020200                          MINUSY*200
1794  011430  114000                          POINT          ;RIGHT SIDE
1795  011432  001700                          1700
1796  011434  001500                          MAXY*1=300
1797  011436  110000                          LONGV
1798  011440  040000                          INTX
1799  011442  020400                          MINUSY*400
1800  011444  040200                          INTX*200
```

```
1801  011446  000000                    0
1802  011450  040000                    INTX
1803  011452  000400                    400
1804  011454  060200                    INTX!MINUSX+200
1805  011456  000000                    0
1806  011460  114000                    POINT
1807  011462  001600                    1600              ;BOTTOM SIDE
1808  011464  000100                    100
1809  011466  110000                    LONGV
1810  011470  060400                    INTX!MINUSX+400
1811  011472  000000                    0
1812  011474  040000                    INTX
1813  011476  020200                    MINUSY+200
1814  011500  040400                    INTX+400
1815  011502  000000                    0
1816  011504  040000                    INTX
1817  011506  000200                    200
1818  011510  114000                    POINT
1819  011512  001777                    MAXX
1820  011514  000400                    400
1821  011516  110000                    LONGV
1822  011520  000020                    20
1823  011522  000000                    0
1824  011524  100000                    CHAR
1825  011526     015     101            .BYTE   15,101      ;"CR" AND AN "A"
1826  011530  114000                    POINT
1827  011532  000000                    0
1828  011534  000500                    500
1829  011536  110000                    LONGV
1830  011540  020012                    MINUSX+12
1831  011542  000000                    0
1832  011544  100000                    CHAR
1833  011546     040     102            .BYTE   40,102      ;"SPACE" AND AN "B"
1834  011550  164000                    DNOP
1835  011552  164000                    DNOP
1836  011554  173400                    DSTOP
1837  011556  164000                    DNOP
1838  011560  164000                    DNOP
1839  011562  160000                    DJMP
1840  011564  011312                    FRME13
1841
```

```
1843
1857
1858  011566  170052          FRME14: STATSA!!TAL!!SYNOFF!GREEN
1859  011570  117124                  POINT!INT4!BLKOFF!LPOFF!LINE0
1860  011572  000000          FRM14A: 0
1861  011574  000000          FRM14B: 0
1862  011576  104000                  SHORTV
1863  011600  056200                  INTX+16200
(1)   011602  056271                  INTX+16200+71
(1)   011604  040071                  INTX+71
(1)   011606  076271                  INTX!MINUSX+16200+71
(1)   011610  076200                  INTX!MINUSX+16200
(1)   011612  076371                  INTX!MINUSX+16200+MINSUY+71
(1)   011614  040171                  INTX+MINSUY+71
(1)   011616  056371                  INTX+16200+MINSUY+71
(1)   011620  020504                  20504
(1)   011622  164000                  DNOP
(1)   011624  164000                  DNOP
1864  011626  130000                  RELATV
1865  011630  057000                  INTX+17000
(1)   011632  057074                  INTX+17000+74
(1)   011634  040074                  INTX+74
(1)   011636  077074                  INTX!MINUSX+17000+74
(1)   011640  077000                  INTX!MINUSX+17000
(1)   011642  077174                  INTX!MINUSX+17000+MINSUY+74
(1)   011644  040174                  INTX+MINSUY+74
(1)   011646  057174                  INTX+17000+MINSUY+74
(1)   011650  020504                  20504
(1)   011652  164000                  DNOP
(1)   011654  164000                  DNOP
1866  011656  104000                  SHORTV
1867  011660  057600                  INTX+17600
(1)   011662  057677                  INTX+17600+77
(1)   011664  040077                  INTX+77
(1)   011666  077677                  INTX!MINUSX+17600+77
(1)   011670  077600                  INTX!MINUSX+17600
(1)   011672  077777                  INTX!MINUSX+17600+MINSUY+77
(1)   011674  040177                  INTX+MINSUY+77
(1)   011676  057777                  INTX+17600+MINSUY+77
(1)   011700  020504                  20504
(1)   011702  164000                  DNOP
(1)   011704  164000                  DNOP
1868  011706  173400                  DSTOP
1869  011710  160000                  DJMP
1870  011712  011566                  FRME14
```

```
1872
1873   011714   117124                      FRME16: POINT!INT4!LPOFF!BLKOFF!LINE0
1874   011716   000000                              0
1875   011720   001600                              MAXY=177
1876   011722   170052                              STATSA!ITAL&!SYNOFF!GREEN
1877   011724   100000                              CHAR
1878   011726      017         017                  .BYTE   17,17
1879   011730   044514   044107   020124            .ASCII  /LIGHT PEN FOLLOW TEST /
       011736   042520   020116   047506
       011744   046114   053517   052040
       011752   051505   020124
1880   011756      015      012      012            .BYTE   15,12,12
1881   011761      130   020075                      .ASCII  /X= /
1882   011764   030061   030060            DLT14A: .ASCII  /1000/
1883   011770      015      012      012            .BYTE   15,12,12
1884   011773      131   020075                      .ASCII  /Y= /
1885   011776   033060   030060            DLT14B: .ASCII  /0600/
1886   012002   114140                              POINT!LPON
1887   012004   001000            RAY14A: 1000
1888   012006   000600            RAY14B: 600
1889   012010   164000                              DNOP
1890   012012   164000                              DNOP
1891   012014   130000                              RELATV
1892   012016   074000            TAB16A: INTX!MINUSX*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: INTX!MINSUY*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: INTX!MINUSX*160
1926   012246   020160                      MINUSX*160
1927   012250   060160            PNT16C: INTX!MINUSX*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  012310  170052                      STATSA;ITAL;SYNOFF;GREEN
1945  012312  100000                      CHAR
1946  012314     017     017              .BYTE   17,17
1947  012316  044514  044107  020124      .ASCII  /LIGHT PEN FIELD OF VIEW /
      012324  042520  020116  044506
      012332  046105  020104  043117
      012340  053040  042511  020127
1948  012346     015     012     012      .BYTE   15,12,12
1949  012351     116  046525  042502      .ASCII  /NUMBER OF HITS = 0000/
      012356  020122  043117  044040
      012364  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  012410  001600                      MAXY=177
1957  012412  170052                      STATSA;ITAL;SYNOFF;GREEN
1958  012414  103000                      CHAR;INT4
1959  012416     017     017              .BYTE   17,17
1960  012420  042513  041131  040517      .ASCIZ  /KEYBOARD ECHO TEST/
      012426  042122  042440  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  012461     000     000     000      .BYTE   0,0,0,0
      012464     000
1964  012465     015     012     012 KBOCT: .BYTE   15,12,12
1965
1966  012470  164000           BUFFER: DNOP
1967
1968          000001                      .END
```

```
BIT6    = 000100     BLKOFF  = 000020     BLKON   = 000030     BUFFER    012470
CHANGE    001050     CHAR    = 100000     CHRCNT    000330     CNTR      001046
COUNT     005952     CRLF      002670     DBUF      001016     DBUF1     001020
DBUF2     001022     DBUF3     001024     DBUF4     001026     DBUF5     001030
DDCNE     001064     DDONE1    001066     DELTX6    010472     DELTX7    010506
DELTY6    010472     DELTY7    010550     DFI10A    010514     DFI11C    010554
DFI12A    011256     DFL15B    004242     DFL15C    004144     DFL15D    004620
DISPLA  = 177570     DISPTC    001210     DJMP    = 160000     DLTI4A    011764
DLTI4B    011776     DNOP    = 164000     DPC       001054     DSAVE     001032
DSAVE1    001034     DSAVE2    001036     DSAVE3    001040     DSR       001056
DSTOP   = 173400     D7A       003330     D7C       003360     D7D       003404
D7F       003434     ECHOA     005060     ECHOB     005316     ECHOC     005166
FILE      005954     FILE0     002050     FILE1     002062     FILE10    003322
FILE11    003378     FILE12    003452     FILE13    003614     FILE14    003626
FILE15    004106     FILE16    004342     FILE17    005052     FILE2     002074
FILE2A    002274     FILE2B    002302     FILE2C    004322     FILE2D    002330
FILE3     002340     FILE3A    002350     FILE3B    004370     FILE4     002414
FILE4A    002746     FILE5     003024     FILE6     003036     FILE7     003170
FILLA     002716     FILLIT    002712     FIL14A    003734     FLE12A    003476
FLE12B    003514     FLE12C    003522     FLE12D    003606     FRME0     005560
FRME1     007140     FRME10    010504     FRME11    010544     FRME12    010604
FRME13    011312     FRME14    011566     FRME16    011714     FRME17    012404
FRME2     007230     FRME3     007334     FRME3A    007724     FRME5     010174
FRME6     010426     FRME6A    010462     FRM10     010604     FRM14A    011572
FRM14B    011574     FRM16A    012302     FRM16B    012376     GRAPHX  = 120000
GRAPHY  = 124000     GREEN   = 000002     GSADD     001000     GSBRL     001004
GSVCT     001002     HERE      004330     HIGPWR    001302     WITCNT    005050
HOLD      001042     ICNT      001006     INCR    = 000100     INTX    = 040000
INT0    = 002000     INT1    = 002200     INT2    = 002400     INT3    = 002600
INT4    = 003000     INT5    = 003200     INT6    = 003400     INT7    = 003600
ITAL0   = 000040     ITAL1   = 000060     KBCHR     000332     KBOCT     012465
KRBD      002046     KYT1      002014     KYT2      001774     KYT3      002024
KYT4      002036     KYT5      001736     LADVT     003714     LINE0   = 000004
LINE1   = 000005     LINE2   = 000006     LINE3   = 000007     LOADAC    004934
LOADBF    002624     LOADSP    002564     LOADUP    004956     LOADVT    003704
LOGICA    004316     LOKRB     001052     LONGV   = 110000     LOOPA     003062
LOOPA1    003076     LOOPA2    003114     LOOPA3    003140     LOOPB     003214
LOOPB1    003222     LOOPB2    003246     LOOPB3    003272     LOWPWR    001250
LOWSV     001300     LPDARK  = 000300     LPLITE  = 000200     LPOFF   = 000100
LPCN    = 000140     LPPNT     011262     LPVCT     001070     LPVCT1    001072
MAXSX   = 217000     MAXSY   = 000077     MAXX    = 001777     MAXY    = 001777
MESG      005412     MESGA     005430     MESGAA    005444     MESGAB    005514
MESGB     005464     MESGBA    005514     MINSUY  = 000100     MINUSX  = 020000
MINUSY  = 020000     PC      =%000007     PNT16A    012240     PNT16B    012244
PNT16C    012250     PNT16D    012254     POINT   = 114000     PSW       001010
Q         001777     R       = 000777     RAYLPA    011250     RAY14A    012004
RAY14B    012006     RED     = 000003     RELATV  = 130000     RETB      001700
RETLP     003350     RET117    005206     RET14     004614     RET20     005244
RET21     005230     R0      =%000000     R1      =%000001     R2      =%000002
R3      =%000003     R4      =%000004     R5      =%000005     SCOPE   = 104000
SCOPEA    001100     SCOPEB    001156     SCOPEC    001162     SCOPED    001170
SCOPEE    001126     SCOPEF    001136     SETPNT    002252     SETUP     001936
SETUPA    001622     SETUPB    001664     SHORTV  = 104000     SIZE0   = 000400
SIZE1   = 000600     SP      =%000006     SPACE     002730     START     001356
```

STATSA = 170007   STATSB = 174000   STCHAR   002666   STKPTR = 000300
STRA   001400   STRB   001424   STRC   001450   SWITCH   005556
SYNOFF = 000010   SYNON = 000014   SYN12   010650   TAB16A   012016
TAB16B   012130   TIMEVT   001074   TKB   001014   TKS   001012
TMEVT1   001076   TSAVE   001044   XPOS   003060   YPOS   001062
.   = 012472

ERRORS DETECTED: 0