ENTRIES

```
 4 000000
 5 000000              ************************************************************
 5 000000              *
 6 000000              *     ENTRIES                                              *
 6 000000              *                                                          *
 6 000000              ************************************************************
 7 000000                          ENT   COMFLT
 8 000000                          ENT   FRE.
 9 000000                          ENT   GCLR++
10 000000                          ENT   GRAFIN
11 000000                          ENT   GRINI+
12 000000                          ENT   GRINIT
13 000000                          ENT   GSCALF
14 000000                          ENT   INGRAF
15 000000                          ENT   ISCOM
16 000000                          ENT   MAPXY
17 000000                          ENT   MAPABS
18 000000                          ENT   MAPCR1
19 000000                          ENT   MOV40
20 000000                          ENT   ORDER4
21 000000                          ENT   PENUP.
22 000000                          ENT   PRADDR
23 000000                          ENT   PRNTR.
24 000000                          ENT   R1.
25 000000                          ENT   RELSHK
26 000000                          ENT   RESTHK
27 000000                          ENT   TAKEHK
28 000000                          ENT   XGA1
29 000000
```

```
>>>>>          EXTERNALS                                                        <<<<<<<
 81 000000                       EXT   ONEB
 82 000000                       EXT   ONER
 83 000000                       EXT   ONEX
 84 000000                       EXT   PAGES.
 85 000000                       EXT   PRDRV
 86 000000                       EXT   PRINT
 87 000000                       EXT   REFNUM
 88 000000                       EXT   REM10
 89 000000                       EXT   RESMEM
 90 000000                       EXT   RETRHI
 91 000000                       EXT   ROMJSB
 92 000000                       EXT   ROMRTN
 93 000000                       EXT   ROUND
 94 000000                       EXT   RSTART
 95 000000                       EXT   RSUM&K
 96 000000                       EXT   RTOIN
 97 000000                       EXT   SCAN
 98 000000                       EXT   SCANLX
 99 000000                       EXT   SEP10
100 000000                       EXT   SEP15
101 000000                       EXT   SETCC-
102 000000                       EXT   SIN10
103 000000                       EXT   STACKX
104 000000                       EXT   STBEEP
105 000000                       EXT   STOSV
106 000000                       EXT   STRREF
107 000000                       EXT   SUB10
108 000000                       EXT   SUBROI
109 000000                       EXT   TAB.
110 000000                       EXT   TAN10
111 000000                       EXT   TRAFIC
112 000000                       EXT   TWOB
113 000000                       EXT   TWOR
114 000000                       EXT   TRYIN
115 000000                       EXT   WARN
116 000000
```

```
             SYSTEM ROM #1
  8 000000
    000000                 ROM#    EQU  1
120 000000                 ROM#-   EQU  377
121 000000
122 000000                         LOC  60000
123 060000 001                     VAL  ROM#
124 060001 377                     VAL  ROM#-
125 060002 012 140                 DEF  RUNTAB
126 060004 350 140                 DEF  ASCTAB
127 060006 170 140                 DEF  PARTAB
128 060010 342 173                 DEF  ERMESS
129 060012                 RUNTAB  BSS  0
130 060012 163 144                 DEF  INIT
131 060014
```

```
LOC    OBJECT CODE  SRC=LGGRAF OBJ=GEX.RA    9/11/1981 12:?? PM  PG  5
==========================================================================
       RUNTIME TABLE                                              <<<<
060014
060014 127 150      DEF   PLOTR.             1  PLOTTER IS
060016 231 173      DEF   PRNTR.             2  PRINTER IS
060020 377 377      DEF   CRT.               3  CRT IS
060022 114 147      DEF   LIMIT.             4  LIMIT
060024 214 144      DEF   GCLR.              5  GCLEAR
060026 070 146      DEF   LOCAT.             6  LOCATE
060030 203 170      DEF   BPLOT.             7  BPLOT
060032 061 147      DEF   SCALE.             10 SCALE
060034 321 147      DEF   SHOW.              11 SHOW
060036 230 147      DEF   MSCAL.             12 MSCALE
060040 371 145      DEF   CLIP.              13 CLIP
060042 272 146      DEF   UNCLI.             14 UNCLIP
060044 105 147      DEF   SETGU.             15 SETGU
060046 074 147      DEF   SETUU.             16 SETUU
060050 357 150      DEF   PENUP.             17 PENUP
060052 067 171      DEF   GREAD.             20 GREAD
060054 002 151      DEF   PEN.               21 PEN
060056 736 154      DEF   LINET.             22 LINETYPE
060060 252 151      DEF   PLOT.              23 PLOT
060062 260 151      DEF   IPLOT.             24 IPLOT
060064 234 151      DEF   MOVE.              25 MOVE
060066 243 151      DEF   IMOVE.             26 IMOVE
060070 327 151      DEF   DRAW.              27 DRAW
060072 306 151      DEF   IDRAW.             30 IDRAW
060074 266 151      DEF   RPLOT.             31 RPLOT
060076 056 172      DEF   PDIR.              32 PDIR
060100 201 170      DEF   BLOFF.             33 NOBLINK
060102 233 164      DEF   AXES.              34 AXES
060104 241 164      DEF   LAXES.             35 LAXES
060106 217 164      DEF   GRID.              36 GRID
060110 165 154      DEF   FRAME.             37 FRAME
060112 262 156      DEF   LABEL.             40 LABEL
060114 201 170      DEF   BLINK.             41 BLINK
060116 067 150      DEF   LORG.              42 LORG
060120 052 156      DEF   LDIR.              43 LDIR
060122 170 155      DEF   CSIZE.             44 CSIZE
060124 161 172      DEF   WHERE.             45 WHERE
060126 220 143      DEF   CONTR.             46 CONTROL
060130 143 172      DEF   CURSR.             47 CURSOR
060132 134 172      DEF   DIGIT.             50 DIGITIZE
060134 333 142      DEF   DUMMY              51 TRANSLATE
060136 225 164      DEF   LGRID.             52 LGRID
060140 377 377      DEF   GRAPH.             53 GRAPHICS
060142 371 163      DEF   XAXIS.             54 XAXIS
060144 001 164      DEF   YAXIS.             55 YAXIS
060146 325 172      DEF   FXD.               56 FXD
060150 033 173      DEF   ERRSC.             57 ERRSC
060152 015 173      DEF   ERROM.             60 ERROM
060154 362 172      DEF   RATIO.             61 RATIO
060156 377 377      DEF   TAB.               62 TAB
060160 051 157      DEF   LABEOL             63 LABEL EOLINE
060162 377 377      DEF   PAGES.             64 PAGE SIZE
```

```
                RUNTIME TABLE
 6 060164 377 377                DEF   ALFAL.                  65 ALPHA ALL
   060166 377 377                DEF   GRAFA.                  66 GRAPH ALL
183 060170              PARTAB   BSS   0
189 060170 211 173               DEF   FRE.                    67 FREE MEMORY
190 060172
```

```
            PARSING TABLS                                                   <<<<<<<<
 060172
 060172 061 142          DEF   G1N                 1  PLOTTER IS
 060174 254 142          DEF   G1-2N               2  PRINTER IS
 060176 254 142          DEF   G1-2N               3  CRT IS
 060200 142 142          DEF   G0-4N               4  LIMIT
 060202 164 142          DEF   G0-1N               5  GCLEAR
 060204 142 142          DEF   G0-4N               6  LOCATE
 060206 021 142          DEF   GPRNT               7  BPLOT
 060210 200 142          DEF   G4N                 10 SCALE
 060212 200 142          DEF   G4N                 11 SHOW
 060214 210 142          DEF   G2N                 12 MSCALE
 060216 142 142          DEF   G0-4N               13 CLIP
 060220 135 142          DEF   PUSHEX              14 UNCLIP
 060222 135 142          DEF   PUSHEX              15 SETGU
 060224 135 142          DEF   PUSHEX              16 SETUU
 060226 135 142          DEF   PUSHEX              17 PENUP
 060230 035 142          DEF   GREAD               20 BREAD
 060232 061 142          DEF   G1N                 21 PEN
 060234 254 142          DEF   G1-2N               22 LINETYPE
 060236 220 142          DEF   G2-3N               23 PLOT
 060240 220 142          DEF   G2-3N               24 IPLOT
 060242 210 142          DEF   G2N                 25 MOVE
 060244 210 142          DEF   G2N                 26 IMOVE
 060246 210 142          DEF   G2N                 27 DRAW
 060250 210 142          DEF   G2N                 30 IDRAW
 060252 220 142          DEF   G2-3N               31 RPLOT
 060254 254 142          DEF   G1-2N               32 PDIR
 060256 135 142          DEF   PUSHEX              33 NOBLINK
 060260 301 142          DEF   G2467               34 AXES
 060262 301 142          DEF   G2467               35 LAXES
 060264 301 142          DEF   G2467               36 GRIDS
 060266 135 142          DEF   PUSHEX              37 FRAME
 060270 015 143          DEF   LABEL               40 LABEL
 060272 135 142          DEF   PUSHEX              41 BLINK
 060274 061 142          DEF   G1N                 42 LORG
 060276 254 142          DEF   G1-2N               43 LDIR
 060300 244 142          DEF   G1-3N               44 CSIZE
 060302 334 142          DEF   WHERE               45 WHERE
 060304 235 142          DEF   G3N                 46 CONTROL
 060306 334 142          DEF   CURSOR              47 CURSOR
 060310 334 142          DEF   DIGIT               50 DIGITIZE
 060312 377 377          DEF   ERR88U              51 TRANSLATE
 060314 301 142          DEF   G2467               52 LGRID
 060316 135 142          DEF   PUSHEX              53 GRAPHICS
 060320 270 142          DEF   G124                54 XAXIS
 060322 270 142          DEF   G124                55 YAXIS
 060324 254 142          DEF   G1-2N               56 FXD
 060326 333 142          DEF   DUMMY               57 ERRSC
 060330 333 142          DEF   DUMMY               60 ERROM
 060332 333 142          DEF   DUMMY               61 RATIO
 060334 333 142          DEF   DUMMY               62 TAB
 060336 333 142          DEF   DUMMY               63 LABEL EOF
 060340 061 142          DEF   G1N                 64 PAGE SIZE
```

>>>> PARSING TABLE <<<<

```
  5 060342 135 142              DEF  PUSHEX              65 ALPHA ALL
  7 060344 135 142              DEF  PUSHEX              66 GRAPH ALL
247 060346 133 142              DEF  DUMMY               67 FREE MEMORY
249 060350
```

ASCII TABLE

```
060350
060350                  ASCTAB  BSS  0
060350 120 114 117              ASP  "PLOTTER IS"          1  PLOTTER IS
060353 124 124 105
060356 122 040
060360 111 323
060362 120 122 111              ASP  "PRINTER IS"          2  PRINTER IS
060365 116 124 105
060370 122 040
060372 111 323
060374 103 122 124              ASP  "CRT IS"              3  CRT IS
060377 040 111 323
060402 114 111 115              ASP  "LIMIT"               4  LIMIT
060405 111 324
060407 107 103 114              ASP  "GCLEAR"              5  GCLEAR
060412 105 101 322
060415 114 117 103              ASP  "LOCATE"              6  LOCATE
060420 101 124 305
060423 102 120 114              ASP  "BPLOT"               7  BPLOT
060426 117 324
060430 123 103 101              ASP  "SCALE"               10 SCALE
060433 114 305
060435 123 110 117              ASP  "SHOW"                11 SHOW
060440 327
060441 115 123 103              ASP  "MSCALE"              12 MSCALE
060444 101 114 305
060447 103 114 111              ASP  "CLIP"                13 CLIP
060452 320
060453 125 116 103              ASP  "UNCLIP"              14 UNCLIP
060456 114 111 320
060461 123 105 124              ASP  "SETGU"               15 SETGU
060464 107 325
060466 123 105 124              ASP  "SETUU"               16 SETUU
060471 125 325
060473 120 105 116              ASP  "PEN UP"              17 PENUP
060476 040 125 320
060501 102 122 105              ASP  "BREAD"               20 BREAD
060504 101 304
060506 120 105 316              ASP  "PEN"                 21 PEN
060511 114 111 116              ASP  "LINE TYPE"           22 LINE TYPE
060514 105 040 124
060517 131 120
060521 305
060522 120 114 117              ASP  "PLOT"                23 PLOT
060525 324
060526 111 120 114              ASP  "IPLOT"               24 IPLOT
060531 117 324
060533 115 117 126              ASP  "MOVE"                25 MOVE
060536 305
060537 111 115 117              ASP  "IMOVE"               26 IMOVE
060542 126 305
060544 104 122 101              ASP  "DRAW"                27 DRAW
060547 327
```

ASCII TABLE

```
  0 060550 111 104 122          ASP  "IDRAW"              30 IDRAW
  3 060553 101 327
279 060555 122 120 114          ASP  "RPLOT"              31 RPLOT
279 060560 117 324
280 060562 120 104 111          ASP  "PDIR"               32 PDIR
280 060565 322
281 060566 116 117 102          ASP  "NOBLINK"            33 NOBLINK
281 060571 114 111 116
281 060574 313
282 060575 101 130 105          ASP  "AXES"               34 AXES
282 060600 323
283 060601 114 101 130          ASP  "LAXES"              35 LAXES
283 060604 105 323
284 060606 107 122 111          ASP  "GRID"               36 GRID
284 060611 304
285 060612 106 122 101          ASP  "FRAME"              37 FRAME
285 060615 115 305
286 060617 114 101 102          ASP  "LABEL"              40 LABEL
286 060622 105 314
287 060624 102 114 111          ASP  "BLINK"              41 BLINK
287 060627 116 313
288 060631 114 117 122          ASP  "LORG"               42 LORG
288 060634 307
289 060635 114 104 111          ASP  "LDIR"               43 LDIR
289 060640 322
290 060641 103 123 111          ASP  "CSIZE"              44 CSIZE
290 060644 132 305
291 060646 127 110 105          ASP  "WHERE"              45 WHERE
291 060651 122 305
292 060653 123 105 124          ASP  "SET I/O"            46 CONTROL
292 060656 040 111 057
292 060661 317
293 060662 103 125 122          ASP  "CURSOR"             47 CURSOR
293 060665 123 117 322
294 060670 104 111 107          ASP  "DIGITIZE"           50 DIGITIZE
294 060673 111 124 111
294 060676 132 305
295 060700 124 122 101          ASP  "TRANSLATE"          51 TRANSLATE
295 060703 116 123 114
295 060706 101 124
296 060710 305
297 060711 114 107 122          ASP  "LGRID"              52 LGRID
297 060714 111 304
298 060716 107 122 101          ASP  "GRAPHICS"           53 GRAPHICS
298 060721 120 110 111
298 060724 103 323
299 060726 130 101 130          ASP  "XAXIS"              54 XAXIS
299 060731 111 323
300 060733 131 101 130          ASP  "YAXIS"              55 YAXIS
300 060736 111 323
301 060740 106 130 304          ASP  "FXD"                56 FXD
302 060743 105 122 122          ASP  "ERRSC"              57 ERRSC
302 060746 123 303
```

```
GRA 09/11/81 ==================================================================================
ITEM    LOC    OBJECT CODE   SRC=CCGRAF OBJ=GEXGRA     9/11/1981 12:38 PM   PG
==================================================================================
                  ASCII TABLE
303 060750 105 122 122          ASP   "ERROM"                   60 ERROM
303 060753 117 315
304 060755 122 101 124          ASP   "RATIO"                   61 RATIO
304 060760 111 317
305 060762 124 101 302          ASP   "TAB"                     62 TAB
306 060765 200                  OCT   200                       63 LABEL EOL
307 060766 120 101 107          ASP   "PAGESIZE"                64 PAGE SIZE
307 060771 105 123 111
307 060774 132 305
308 060776 101 114 120          ASP   "ALPHALL"                 65 ALPHA ALL
308 061001 110 101 114
308 061004 314
309 061005 107 122 101          ASP   "GRAPHALL"                66 GRAPH ALL
309 061010 120 110 101
309 061013 114 314
310 061015 106 122 305          ASP   "FRE"                     67 FREE MEMORY
311 061020 377                  OCT   377
312 061021
```

PARSING ROUTINES

```
061021
061021                 GIS      BSS  0
061021 316 377 377 GPRNT    JSB  =ROMJSB
061024 377 377              DEF  G#N            PARSE LIKE BPLOT
061026 000                  OCT  0
061027 153 270 316          LDBI R53,=PTR2+     RESTORE INPUT TOK
061032 377
061033 760 042              JMP  PTOKEX
061035
061035 143 006 344 GREAD    PUBD R43,+R6        SAVE INPUT TOKEN
061040 316 126 142          JSB  =RMSCAN
061043 316 377 377          JSB  =ROMJSB
061046 377 377              DEF  STRREF
061050 000                  OCT  0
061051 143 006 342          POBD R43,-R6        RECOVER ORIGINAL TOKEN
061054 114 310 054          CMB  R14,=COMMA
061057 766 073              JNZ  PERROR
061061                 *
061061                 **** FALL THROUGH TO GIN ***
061061                 *
061061 143 006 344 GIN      PUBD R43,+R6        SAVE INPUT TOKEN
061064 316 377 377          JSB  =ROMJSB
061067 377 377              DEF  NUMVA+         1 NUMERIC
061071 000                  OCT  0
061072 153 006 342          POBD R53,-R6        RESTORE TOKEN
061075 371 055              JEZ  PERROR         JIF PARSE ERROR
061077 316 107 142 PTOKEX   JSB  =PTOKEN        PUSH TOKENS
061102 760 054              JMP  ROMEX
061104
061104 153 043 240 PTOKN+   LDB  R53,R43        LOAD UP TOKEN
061107 156 251 001 PTOKEN   LDM  R56,=1,370     PUSH ROM TOKEN
061112 370
061113 273 315 377          STMI R56,=PTR2-
061116 153 272 315          STBI R53,=PTR2-     PUSH GRAPH TOKEN
061121 377
061122 236                  RTN
061123
061123 316 104 142 PUSH+S   JSB  =PTOKN+        PUSH TOKENS
061126 316 377 377 RMSCAN   JSB  =ROMJSB
061131 377 377              DEF  SCAN
061133 000                  OCT  0
061134 236                  RTN
061135
061135 316 123 142 PUSHEX   JSB =PUSH+S         PUSH TOKS,SCAN,ROMRTN
061140 760 016              JMP  ROMEX
061142
061142 316 316 142 G0-4N    JSB  =GET???        GET SOME NUMERICS
061145 220                  TSB  R#             0 IS OK
061146 367 327              JZR  PTOKEX         JIF 0
061150 310 004              CMB  R#,=4          4 IS OK
061152 367 323              JZR  PTOKEX         JIF 4
061154 316 377 377 PERROR   JSB  =ERROR         BAD PARSE
061157 131                  OCT  89D            BAD PARAMS
```

```
          PARSING ROUTINES
061160 104 251 377 ROMEX    GTO  ROMRTN
061163 377
061164
061164 316 377 377 G0-1N    JSB  =ROMJSB
061167 377 377              DEF  TRY1N            GET 0 OR 1 NUMERIC
061171 000                  OCT  0
061172 153 270 316 POPUSH   LDBI R53,=PTR2+       RETRIEVE TOKEN
061175 377
061176 760 277     PTJP     JMP  PTOKEX           TRY1N WILL ERROR IF NEED BE
061200
061200 316 377 377 G4N      JSB  =ROMJSB
061203 377 377              DEF  GET4N            SYSTEM ROUTINE
061205 000                  OCT  0
061206 760 362              JMP  POPUSH           POP R6 AND PUSH
061210
061210 316 377 377 G2N      JSB  =ROMJSB
061213 377 377              DEF  GET2N            SYSTEM ROUTINE
061215 000                  OCT  0
061216 760 352              JMP  POPUSH           POP R6 AND PUSH
061220
061220 316 316 142 G2-3N    JSB  =GET???          GET SOME NUMERICS
061223 310 002              CMB  R#,=2            <2 IS BAD
061225 372 325     G3N+     JNC  PERROR           JIF 0,1
061227 310 004     G<4      CMB  R#,=4            >3 IS BAD
061231 373 321              JCY  PERROR           JIF >3
061233 760 242              JMP  PTOKEX
061235
061235 316 316 142 G3N      JSB  =GET???
061240 310 003              CMB  R#,=3
061242 760 361              JMP  G3N+
061244
061244 316 316 142 G1-3N    JSB  =GET???
061247 220                  TSB  R#
061250 367 302              JZR  PERROR
061252 760 353              JMP  G<4
061254
061254 316 316 142 G1-2N    JSB  =GET???          GET SOME NUMERICS
061257 220         G1-2N+   TSB  R#
061260 367 272              JZR  PERROR
061262 310 003              CMB  R#,=3            >2 IS BAD
061264 373 266              JCY  PERROR
061266 760 207              JMP  PTOKEX
061270
061270 316 316 142 G124     JSB  =GET???
061273 310 004              CMB  R#,=4
061275 367 200     PTX      JZR  PTOKEX
061277 760 356              JMP  G1-2N+
061301
061301 316 316 142 G2467    JSB  =GET???          GET SOME NUMERICS
061304 310 007              CMB  R#,=7            7 IS OK
061306 367 365              JZR  PTX
061310 373 242              JCY  PERROR           >7 IS BAD
061312 763 240              JEV  PERROR           ODD-7 IS EVEN
```

```
>>>>>          PARSING ROUTINES                                              <<<<<<<<
061314 260 260      PUSHJM  JMP  PTJP            JIF 0,2,4,6
061316
061316 316 377 377 GET???   JSB  =ROMJSB
061321 377 377              DEF  GETPA?          GET NUMERICS
061323 000                  OCT  0
061324 153 270 316          LDBI R53,=PTR2+      RESTORE INPUT TOKEN
061327 377
061330 134                  ORP  34
061331 236                  RTN
061332
061332 241                  OCT  241
061333 236         DUMMY    RTN                  IGNORED
061334
061334             WHERE    BSS  0
061334             DIGIT    BSS  0
061334 114 036 242 CURSOR   STB  R14,R36         SAVE INPUT TOKEN
061337 316 374 142          JSB  =VITEM          GET VARIABLE
061342 371 210     PJ       JEZ  PERROR          JIF NOT 1
061344 114 310 054          CMB  R14,=COMMA      DEMAND COMMA
061347 366 203              JNZ  PERROR
061351 316 374 142          JSB  =VITEM
061354 371 364              JEZ  PJ              JIF NOT 2
061356 053 242              STB  R#,R53          IN CASE OF 2
061360 114 310 054          CMB  R14,=COMMA      COMMA?
061363 366 327              JNZ  PUSHJM          EXIT IF NO
061365 316 374 142          JSB  =VITEM
061370 053 242              STB  R#,R53
061372 360 320              JMP  PUSHJM          2 OR 3 OK
061374
061374 316 126 142 VITEM    JSB  =RMSCAN         SCAN
061377 235                  CLE
061400 136 006 344          PUBD R36,+R6         SAVE INPUT TOKEN
061403 316 377 377          JSB  =ROMJSB
061406 377 377              DEF  REFNUM
061410 000                  OCT  0
061411 136 006 342          POBD R36,-R6         POP INPUT TOKEN
061414 236                  RTN
061415
061415             *************************************************************
061415             *                                                           *
061415             *                                                           *
061415             *************************************************************
061415 316 104 142 LABEL    JSB  =PTOKN+         PUSH LABEL TOKEN ,SCAN
061420 114 270 316          LDBI R14,=PTR2+      POP LABEL TOKEN TO R14
061423 377
061424 316 377 377          JSB  =ROMJSB
061427 377 377              DEF  PRINT           PRINT PARSING
061431 000                  OCT  0
061432 153 270 316          LDBI R53,=PTR2+      CHECK LAST TOKEN
061435 377
061436 310 354              CMB  R53,=CRTOK      EOL?
061440 367 003              JZR  REMOVT          JIF YES
061442 272 315 377          STBI R53,=PTR2-      PUSH THIS TOK BACK
```

```
>>>>>>        PARSING ROUTINES                                                      <<<<<<<<
462 061445 250 063      REMOVT  LDB  R#,=LRTOK        LABEL RETURN TOKEN
463 061447 260 243              JMP  PUSHJM           PUSH TOKEN
464 061451
465 061451              LRTOK   EQU  63
466 061451              CRTOK   EQU  354
467 061451
```

```
>>>>>        SET I/O
061451
061451                 PRADD-   BSS   0
061451 316 377 377              JSB   =ONER          GET DEVICE ADDR
061454                 PRADDR   BSS   0
061454 316 377 377              JSB   =RTOIN         CONVERT TO INTEGER
061457 164 250 377              LDB   R64,=377
061462 054 242                  STB   R64,R54        FOR STACK
061464 155 223                  CLM   R55
061466 235                      CLE                  RESET ADDRESSING FLAG
061467 166 221                  TSM   R66            ANY DEVICE ID ?
061471 766 002                  JNZ   ADRNZ          JIF YES
061473 234                      ICE                  SET NO ADDRESSING
061474 165                      DRP   65
061475                 ADRNZ    BSS   0
061475 055 243                  STM   R#,R55         R55=SC
061477 223                      CLM   R#
061500 150 012 345              PUMD  R50,+R12       PUSH SC
061503 160 345                  PUMD  R60,+R12       PUSH DEVICE
061505 100 200                  ELB   R0             SAVE E
061507 316 377 377              JSB   =TUOB          CONVERT TO BINARY
061512 231                      BCD
061513 100 202                  ERB   R0             RESTORE E
061515 230                      BIN
061516 146                      DRP   46
061517 371 004                  JEZ   ADRNZ+         JIF ADDRESSING
061521 250 177                  LDB   R46,=177       FLAG FOR RS232L
061523 760 005                  JMP   R46177
061525                 ADRNZ+   BSS   0
061525 311 040 000              CMM   R#,=32D,0      DEVICE OK ?
061530 373 021                  JCY   SCERR          JIF NO
061532                 R46177   BSS   0
061532 156 311 013              CMM   R56,=11D,0     SC OK ?
061535 000
061536 373 013                  JCY   SCERR          JIF NOT
061540 310 003                  CMB   R56,=3         SC>2 ?
061542 373 016                  JCY   PREX+          EXIT WITH E SET
061544 371 005                  JEZ   SCERR          0,1,2 CAN'T HAVE ADDRESSING
061546 310 001                  CMB   R56,=1
061550 766 001                  JNZ   SCERR          AND IT MUST BE 1
061552 236             PREXIT   RTN
061553
061553                 SCERR    BSS   0
061553 316 327 173              JSB   =ROMERR
061556 316 377 377              JSB   =ERROR+
061561 012                      OCT   10D
061562
061562                 PREX+    BSS   0
061562 115 250 001              LDB   R15,=1         ONLY TALK TO LOGGED IN CARDS
061565 114 056 240              LDB   R14,R56
061570 314 003                  SBB   R14,=3
061572 767 006                  JZR   DOAND
061574                 PRL      BSS   0
061574 115 204                  LLB   R15            SHIFT MASK
```

```
[illegible]       SET I/O
521 061576 114 212                    DCB   R14
522 061600 266 372                    JNZ   PRL
523 061602                 DOAND      BSS   0
524 061602 115 327 140                ANMD  R15,=IOBITS
524 061605 202
525 061606 766 342                    JNZ   PREXIT          JIF OK
526 061610 316 327 173                JSB   =ROMER?
527 061613 316 377 377                JSB   =ERROR+
528 061616 013                        OCT   11D             BAD SELECT CODE
529 061617
530 061617 241                        OCT   241
531 061620
532 061620                 CONTR.     BSS   0
533 061620 316 377 377                JSB   =TUOB           GET REGISTER, CONTROL BYTE
534 061623 156 311 030                CMM   R56,=24D,0      REG > 23 IS ERROR
534 061626 000
535 061627 373 057                    JCY   ERROR1
536 061631 312 200                    ADB   R56,=CTLCMD     ADD IN CONTROL
537 061633 006 344                    PUBD  R56,+R6         SAVE THIS STUFF
538 061635 146 262 135                STBD  R46,=DATAB      STORE BYTE IN BUFFER
538 061640 207
539 061641 316 051 143                JSB   =PRADD-         PROCESS DEV ADDR
540 061644 117 310 300                CMB   R17,=300        ANY ERROR ?
541 061647 373 034                    JCY   ERROR2          JIF YES
542 061651 371 032                    JEZ   ERROR2          JIF ADDRESS
543 061653 156 310 001                CMB   R56,=1          1 NO GOOD EITHER
544 061656 367 025                    JZR   ERROR2
545 061660 314 003                    SBB   R56,=3
546 061662 204                        LLB   R56             GET SELECT CODE
547 061663 313 120 377                ADM   R56,=120,377    BASE
548 061666 026 243                    STM   R56,R26         CCR
549 061670 211                        ICM   R56
550 061671 076 243                    STM   R56,R76         DATA BUFFER FOR CCR
551 061673 146 006 342                POBD  R46,-R6         POP BACK COMMAND
552 061676 316 377 377                JSB   =ROMJSB
553 061701 377 377                    DEF   CNTOUT          SET CONTROL
554 061703 320                        OCT   320
555 061704 236                        RTN
556 061705
557 061705 146 006 342 ERROR2         POBD  R46,-R6         CLEAN STACK
558 061710 316 377 377 ERROR1         JSB   =ERROR
559 061713 131                        OCT   89D
560 061714 236                        RTN
561 061715
562 061715                 CTLCMD     EQU   200
563 061715
```

```
>>>>>>>>         ALPHA ALL / GRAPH ALL
 5  061715
  ) 061715                     *     SECTION REMOVED
567 061715
568 061715                     ************************************************************
568 061715                     *
569 061715                     *                                                          *
569 061715                     *     WAIT TILL CRT CONTROLLER NOT BUSY                     *
569 061715                     *                                                          *
570 061715                     ************************************************************
571 061715
572 061715                     *CHKSTS  BSS  0
573 061715                     *        BIN
574 061715                     *        DRP  30
575 061715                     *L90     BSS  0
576 061715                     *        LDBD R#,=CRTSTS
577 061715                     *        JOD  L90
578 061715                     *        RTN
579 061715
580 061715                     *        HED  INVERSE
 1  061715                     *
 2  061715                     *        BOX
583 061715                     *    INVERSE DISPLAY OF CRT
584 061715
585 061715                     *        OCT  241              PRIMARY ATTRIBUTE
586 061715                     *
587 061715                     *INVER.  BSS  0
588 061715                     *        JSB  =RETRHI          WAIT TILL RETRACE
    061715                     *        LDB  R31,=40          FORM INVERSE MASK
    061715                     *        LDBD R30,=CRTSTS      GET CRT STATUS
591 061715                     *        XRB  R30,R31          INVERSE DISPLAY
592 061715                     *        STBD R30,=CRTSTS      TELL CONTROLLER
593 061715                     *        RTN
594 061715
```

```
>>>>>         CRT GRAPHICS SECTION DEFAULTS
596 061715
597 061715                    ***********************************************************
597 061715                    *                                                         *
598 061715                    *     GRAPHICS SCALING FACTORS                            *
599 061715                    *                                                         *
600 061715                    *       SCALED-X = UNSCALED-X * XA + XB                   *
600 061715                    *                                                         *
600 061715                    ***********************************************************
601 061715
```

```
>>>>>>         GRAPHICS LIMITS                                              <<<<<<
603 061715
604 061715                    *    CRT MACHINE LIMITS
605 061715
606 061715 000 000 000 XG1      OCT  0,0,0,0,0,0,0,0
606 061720 000 000 000
606 061723 000 000
607 061725 002 000 000 XG2      OCT  2,0,0,0,0,0,90C,39C
607 061730 000 000 000
607 061733 220 071
608 061735 000 000 000 YG1      OCT  0,0,0,0,0,0,0,0
608 061740 000 000 000
608 061743 000 000
609 061745 002 000 000 YG2      OCT  2,0,0,0,0,0,90C,23C
609 061750 000 000 000
609 061753 220 043
610 061755
611 061755                    *    MACHINE LIMIT FOR GRAPH ALL
612 061755
613 061755 000 000 000 XGA1     OCT  0,0,0,0,0,0,0,0
613 061760 000 000 000
613 061763 000 000
614 061765 002 000 000 XGA2     OCT  2,0,0,0,0,0,30C,54C
614 061770 000 000 000
614 061773 060 124
615 061775 000 000 000 YGA1     OCT  0,0,0,0,0,0,0,0
615 062000 000 000 000
615 062003 000 000
616 062005 002 000 000 YGA2     OCT  2,0,0,0,0,0,90C,23C
616 062010 000 000 000
616 062013 220 043
617 062015
```

LINE TYPE DEFINITIONS

```
062015
062015                           ******************************************************
062015                           *
062015                           *     LINE TYPE DEFINITIONS, ETC.
062015                           *
062015                           ******************************************************
062015
062015                           LNTYP+  BSZ   0
062015 035 144                           DEF   TYPE1
062017 035 144                           DEF   TYPE2
062021 042 144                           DEF   TYPE3
062023 055 144                           DEF   TYPE4
062025 037 144                           DEF   TYPE5
062027 044 144                           DEF   TYPE6
062031 050 144                           DEF   TYPE7
062033 057 144                           DEF   TYPE8
062035
062035                           TYPE1   BSZ   0
062035                           TYPE2   BSS   0
062035 001                       T1#SEG  OCT   1
062036 377                       T1SEG1  OCT   377
062037
062037                           TYPE5   BSZ   0
062037 002                       T5#SEG  OCT   2
062040 012                       T5SEG1  OCT   12
062041 005                       T5SEG2  OCT   5
062042
062042                           TYPE3   BSZ   0
062042 002                       T3#SEG  OCT   2
062043 001                       T3SEG1  OCT   1
062044                           T3SEG2  BSZ   0            4 (OVERLAP TO SAVE SPACE)
062044
062044                           TYPE6   BSZ   0
062044 004                       T6#SEG  OCT   4
062045 010                       T6SEG1  OCT   10
062046 004                       T6SEG2  OCT   4
062047 001                       T6SEG3  OCT   1
062050                           T6SEG4  BSZ   0            4 (OVERLAP TO SAVE SPACE)
062050
062050                           TYPE7   BSZ   0
062050 004                       T7#SEG  OCT   4
062051 010                       T7SEG1  OCT   10
062052 003                       T7SEG2  OCT   3
062053 003                       T7SEG3  OCT   3
062054 003                       T7SEG4  OCT   3
062055
062055                           TYPE4   BSZ   0
062055 002                       T4#SEG  OCT   2
062056 006                       T4SEG1  OCT   6
062057                           T4SEG2  BSZ   0            6 (OVERLAP TO SAVE SPACE)
062057
062057                           TYPE8   BSZ   0
062057 006                       T8#SEG  OCT   6
```

```
>>>>          LINE TYPE DEFINITIONS
    062060 010            T8SEG1  OCT  10
    062061 004            T8SEG2  OCT  4
671 062062 001            T8SEG3  OCT  1
672 062063 004            T8SEG4  OCT  4
673 062064 001            T8SEG5  OCT  1
674 062065 004            T8SEG6  OCT  4
675 062066
```

```
                CHARACTER TABLE
 7 062066
 8 062066
678 062066                 ************************************************************
679 062066                 *
680 062066                 *     CHARACTER TABLE                                      *
681 062066                 *                                                          *
682 062066                 *       EACH ENTRY IS 2-BYTE.                              *
683 062066                 *       THE MOST SIGNIFICANT 4-BIT IS # OF STROKES.        *
684 062066                 *       THE REMAINING 12-BIT IS THE RELATIVE ADDR          *
684 062066                 *       TO THE STROKE TABLE ENTRY.                         *
684 062066                 *                                                          *
685 062066                 ************************************************************
686 062066
687 062066                         EXT   CHRTAB
```

STROKE TABLE

```
?   062066
?   062066
690 062066              ***************************************************************
691 062066              *                                                             *
692 062066              *    STROKE TABLE                                             *
693 062066              *                                                             *
694 062066              *       EACH ENTRY IS 1-BYTE.                                 *
695 062066              *       EACH CHAR IS A 12*8 MATRIX.                           *
696 062066              *       THE HIGH NIBBLE IS X-COOR.                            *
696 062066              *       THE LOWER NIBBLE IS Y-COOR.                           *
696 062066              *                                                             *
697 062066              ***************************************************************
698 062066
699 062066                      EXT  STROK
```

```
062066
062066            ***************************************************************
062066            *
062066            *     ROM INITIALIZATION                                          *
062066            *                                                                 *
062066            *       ROMFL= 0    POWER ON                                      *
062066            *              1    RESET                                         *
062066            *              2    SCRATCH                                       *
062066            *              3    LOADBIN                                       *
062066            *              4    RUN, INIT                                     *
062066            *              5    LOAD                                          *
062066            *              6    STOP                                          *
062066            *              7    CHAIN                                         *
062066            *             10    ALLOCATE CLASS > 56                           *
062066            *             11    DEALLOCATE CLASS > 56                         *
062066            *             12    DECOMPILE CLASS > 56                          *
062066            *             13    ERROR HALT                                    *
062066            *                                                                 *
062066            ***************************************************************
062066
062066            *INIT    BSS  0
062066            *        RTN
062066
```

GRAPHICS INITIALIZATION

```
062066
062066                 **********************************************************************
062066                 *                                                                    *
062066                 *     GRAPHICS INITIALIZATION                                        *
062066                 *                                                                    *
062066                 **********************************************************************
062066
062066                 GRINIT  BSS  0
062066                 ***HOOKS ARE INITIALIZED BY SYS NOW********
062066                 ********LDMD R71,=DEFHOK   INITIALIZE HOOK
062066                 ********STMD R71,=PLHOOK
062066                 ********LDB  R71,=236      INIT DGHOOK TO RTN
062066                 ********STBD R71,=DGHOOK
062066 124 223                 CLM  R24
062070 263 321 212             STMD R24,=LAXNY    DEFAULT FXD 0
062073 211                     ICM  R24           DEFAULT PLOTTER# = 1
062074 263 133 211             STMD R24,=PLOTR#
062077 251 300 077             LDM  R24,=300,77   DEFAULT GRAPH SIZE
062102 263 342 211             STMD R24,=GSIZE
062105 251 062 000             LDM  R24,=50D,0    DEFAULT #-BYTES/LINE
062110 263 340 211             STMD R24,=GLINE
062113 251 377 377             LDM  R24,=CHRTAB   DEFAULT CHAR TABLE
062116 263 346 211             STMD R24,=CHTABL
062121 251 015 144             LDM  R24,=LNTYP+   DEFAULT LINE TYPE DEFINITIONS
062124 126 251 350             LDM  R26,=LNTYPE
062127 211
062130 316 343 144             JSB  =MOV20
062133 316 072 173             JSB  =C0           INIT LIMITS
062136 112 263 344             STMD R12,=TOS      INIT TOS
062141 203
062142
062142                 GRINI+  BSS  0
062142
062142                 GPON    BSS  0             POWER ON FOR GRAPHICS
062142 316 152 173             JSB  =XMTOXV       SET DEFAULT VIEW PORT
062145 316 142 150             JSB  =GPON3        DO GPON LEVEL 3
062150 140 261 330             LDMD R40,=YM2      DEFAULT TO MACHINE LIMIT
062153 211
062154 263 072 211             STMD R40,=TMP2     FOR GCLEAR
062157 316 275 144             JSB  =GCLR!
062162 236             GRIEND  RTN
062163
062163                 INIT    BSS  0
062163 132 260 065             LDBD R32,=ROMFL    IS IT POWER ON ?
062166 210
062167 366 371                 JNZ  GRIEND        JIF NO
062171 262 116 212             STBD R32,=LBLFLG   RESET LABEL FLAG
062174 251 000 140             LDM  R32,=0,140    1ST BYTE ADR OF ROM
062177 316 377 377             JSB  =RSUM8K       DO CHECKSUM
062202 367 356                 JZR  GRIEND        JIF OK
062204 316 327 173             JSB  =ROMERR       ELSE REPORT ERROR
062207 316 377 377             JSB  =ERROR+
062212 014                     OCT  12D           P/P ROM CHECK SUM ERR
```

```
>>>>>>          GRAPHICS INITIALIZATION
3 062213
```

```
>>>>>>      GCLEAR                                                       <<<<<<
    062213
    062213                 *****************************************************
    062213                 *                                                   *
    062213                 *     GCLEAR RUN TIME ROUTINE                       *
    062213                 *                                                   *
    062213                 *****************************************************
    062213
    062213 241                      OCT   241           PRIMARY ATTRIBUTE
    062214
    062214                 GCLR.    BSS   0
    062214 130 251 002              LDM   R30,=2,0
    062217 000
    062220 316 125 145              JSB   =UPLTDR
    062223 236                      RTN
    062224
    062224                 C2       BSS   0
    062224 316 377 377              JSB   =GRAPH        SWITCH TO GRAPHICS
    062227 112 321 344              CMMD  R12,=TOS      ANY PARAMETER?
    062232 203
    062233 367 031                  JZR   GCLR1         JIF NO
    062235 316 377 377              JSB   =ONER         CONVERT TO REAL
    062240 124 012 241              LDM   R24,R12       SET UP POINTER
    062243 140 345                  PUMD  R40,+R12      SAVE IT
    062245 126 251 072              LDM   R26,=TMP2     SET DESTINATION
    062250 211
    062251 316 357 145              JSB   =GSCALF       GET SCALE FACTOR
    062254 313 020 000              ADM   R#,=20,0      POINT TO SCALE OF Y
    062257 316 217 145              JSB   =MAP1MU       MAP TO MU
    062262 012 343                  POMD  R#,-R12       THROW AWAY THE PAR
    062264 360 007                  JMP   GCLR2
    062266                 GCLR1    BSS   0
    062266 140 261 330              LDMD  R40,=YM2      DEFAULT CLEAR ALL
    062271 211
    062272 263 072 211              STMD  R40,=TMP2
    062275                 GCLR2    BSS   0             FALL THROUGH
    062275
    062275                 GCLR!    BSS   0
    062275 223                      CLM   R#
    062276 263 062 211              STMD  R#,=TMP1
    062301 126 251 062              LDM   R26,=TMP1
    062304 211
    062305 316 053 154              JSB   =MAPCR1       MAP TO CRT BINARY ADDR
    062310                          ARP   !12
    062310 122 343                  POMD  R22,-R12      ***REPLACEMENT***
    062312 120 343                  POMD  R20,-R12      ***REPLACEMENT***
    062314 316 113 154              JSB   =MAPABS       MAP TO ABS BYTE ADDR
    062317 132 260 151              LDBD  R32,=PLOTSY   GET PEN
    062322 200
    062323                 GCLR++   BSS   0
    062323 136 261 342              LDMD  R36,=GSIZE    GET LWA+1
    062326 211
   *062327 076 305                  SBM   R36,R76       GET LENGTH
    062331                 GCLR+    BSS   0
```

```
    [illegible]          GCLEAR                                                       [illegible]
 [illegible]3 062331 316 032 154          JSB  =BYTEG          TELL CRT
 [illegible]14 062334 316 377 377          JSB  =L7             CLEAR GRAPH MEMORY
 315 062337 236                          RTN
 816 062340
```

MOVE BLOCK OF DATA

```
062340                 ************************************************************
062340                 *                                                          *
062340                 *     MOVE A BLOCK OF DATA                                 *
062340                 *                                                          *
062340                 *       ENTRY:                                             *
062340                 *         R24-25=POINTER TO SOURCE                         *
062340                 *         R26-27=POINTER TO DESTINATION                    *
062340                 *                                                          *
062340                 *       WORKING REGISTERS:                                 *
062340                 *         R70-77=DATA REGISTERS                            *
062340                 *                                                          *
062340                 ************************************************************
062340                 MOV40   BSZ  0               ;MOVE 40 BYTES
062340 316 343 144             JSB  =MOV20          ;MOVE 1ST 20 BYTES
062343
062343                 MOV20   BSZ  0               ;MOVE 20 BYTES
062343 316 346 144             JSB  =MOV10          ;MOVE 1ST 10 BYTES
062346
062346                 MOV10   BSZ  0               ;MOVE 10 BYTES
062346 170 024 341             POMD R70,+R24        ;POP SOURCE
062351 026 345                 PUMD R70,+R26        ;STORE IN DEST
062353 236                     RTN
062354                 MOVXY   BSZ  0               ;MOVE THE COOR OF A PT
062354 316 346 144             JSB  =MOV10          ;MOVE X-COOR
062357 024 341                 POMD R#,+R24         ;INCR SOURCE PTR
062361 316 346 144             JSB  =MOV10          ;MOVE Y-COOR
062364 236                     RTN
062365
062365                 MOV**   BSS  0
062365 316 346 144             JSB  =MOV10          MOVE ONE ELEMENT
062370 341                     POMD R#,+R#          INCR DEST PTR
062371 316 346 144             JSB  =MOV10          MOVE 2ND ELEMENT
062374 236                     RTN
062375
```

COMPARE FLOATING POINT

```
062375                    ************************************************************
062375                    *
062375                    *     COMPARE FLOATING POINT                                 *
062375                    *                                                            *
062375                    *       ENTRY:                                               *
062375                    *         R50-57=A                                           *
062375                    *         R40-47=B                                           *
062375                    *                                                            *
062375                    *       RETURN:                                              *
062375                    *         R40-47 = R50-57 = A-B                              *
062375                    *         E=0 IF A < B                                       *
062375                    *         E=1 IF A >= B                                      *
062375                    *                                                            *
062375                    ************************************************************
062375                    COMPLT   BSZ   0
062375 231                         BCD                ;CHANGE MODE FOR MATH
062376 316 377 377                 JSB   =SUB10       ;R40=R50-R40
062401                             ARP   !12
062401 235                         CLE                ;RESET FLAG
062402 150 343                     POMD  R50,-R12     ;POP RESULT
062404 151 220                     TSB   R51          ;CHECK SIGN
062406 377 001                     JRN   CRTN         ;JIF NEGATIVE
062410 234                         ICE                ;ELSE SET .GE.
062411                    CRTN     BSZ   0
062411 140                         ARP   40
062412 236                         RTN
```

```
              GET PARMETERS
874 062413                    ************************************************************
    062413                    *                                                          *
875 062413                    *      GET TWO OR FOUR PARMETERS                           *
876 062413                    *                                                          *
877 062413                    *        OUT:                                              *
878 062413                    *          E=0 FOR PARS FROM STATEMENT                     *
879 062413                    *          E=1 FOR PARS FROM DIGITIZE                      *
880 062413                    *          E=-1 FOR STOP OR ERROR                          *
880 062413                    *                                                          *
880 062413                    ************************************************************
881 062413
882 062413                    GFOUR   BSZ   0
883 062413 316 377 377               JSB   =STACKX         ANYTHING ON STACK
884 062416 367 026                   JZR   DIGIT2          JIF NO
885 062420                           DRP   !26
886 062420 251 122 211               LDM   R26,=TMP4-      ;SETUP POINTER
887 062423 316 026 145    G4         JSB   =G2             ;GET 1ST TWO PARS
888 062426                G2         BSZ   0               ;GET NEXT TWO PARS
889 062426 316 377 377               JSB   =TUOR           ;GET 2 PARS FROM STACK
890 062431 140 026 347               PUMD  R40,-R26        ;STORE IN PLACE
891 062434 150 347                   PUMD  R50,-R26
892 062436 235                       CLE
893 062437 236                       RTN
894 062440                GTWO       BSZ   0
895 062440 126 251 102               LDM   R26,=TMP2-      ;SETUP POINTER
895 062443 211
896 062444 360 360                   JMP   G2              ;GO GET PARS
897 062446
898 062446                DIGIT2     BSS   0               DIGITIZE 2 PTS
899 062446                           DRP   !26
900 062446 251 062 211               LDM   R26,=TMP1       SET PTR TO DESTINATION
901 062451 316 063 145               JSB   =DIGIT1         DIGITIZE 1 PT
902 062454 234                       ICE
903 062455 371 024                   JEZ   NOVARX          JIF ERROR
904 062457 126 251 072               LDM   R26,=TMP2       PT TO 2ND PT
904 062462 211
905 062463                DIGIT1     BSS   0
906 062463 316 105 145               JSB   =DONEP+         DIGITIZE 1 PT
907 062466 371 013                   JEZ   NOVARX          JIF STOP OR ERROR
908 062470 316 377 377               JSB   =STBEEP         BEEP TO SIGNAL
909 062473 124 251 042               LDM   R24,=XCURS      PT IS IN CURSOR
909 062476 211
910 062477 316 365 144               JSB   =MOV**
911 062502 236             DIGEND     RTN
912 062503
913 062503                NOVARX     BSS   0
914 062503 233                       DCE
915 062504 236             NOVRTN     RTN
916 062505
917 062505                *    DIGITIZER PUT THE COORDINATES OF THE
918 062505                *    CURSOR INTO XCURS,YCURS AND SET
919 062505                *    E=1 IF SUCCESS, OR E=0 IF FAIL
920 062505                *    SHOULD LEAVE R26 UNCHANGED
```

```
>>>>>.          GET PARMETERS                                                <<<<<<<<
921 062505
922 062505                      DONEP+  BSS   0
923 062505 130 251 014                  LDM   R36,=14,0      DIGITIZE 1 PT
923 062510 000
924 062511 316 125 145                  JSB   =UPLTDR        CALL DRIVER
925 062514 370 366                      JEN   NOVRTN         JIF SUCCESS
926 062516 316 327 173                  JSB   =ROMERR        ELSE ERROR
927 062521 316 377 377                  JSB   =ERROR+
928 062524 011                          OCT   9D
929 062525
```

```
>>>>>>           UNIVERSAL PLOTTER DRIVER
062525
062525                     *     ENTRY:  R30= INDEX TO DRIVER ROUTINE
062525
062525                     JFLTDR  BSS   0
062525 114 261 133                 LDMD  R14,=PLOTR#   GET PLOTTER #
062530 211
062531 311 003 000                 CMM   R14,=3,0      EXTERNAL PLOTTER ?
062534 365 011                     JPS   PLOTER        JIF YES
062536 130 030 265                 LDMD  R30,X30,CRTDRV  GET CRT ROUTINE
062541 042 173
062543 306 000 000                 JSB   X30,ZRO       GO DO IT
062546 236                         RTN
062547                     PLOTER  BSS   0
062547 316 261 207                 JSB   =PLHOOK       JSB TO HOOK
062552 236                         RTN
062553
```

```
<<<<<       MAPPING TO MACHING UNIT
062553
062553                  ************************************************************
062553                  *
062553                  *     MAPPING TO MACHINE UNIT                               *
062553                  *                                                           *
062553                  *       ENTRY:                                              *
062553                  *         R22-23 = PTR TO SCALING FACTORS                   *
062553                  *         R24-25 = PTR TO SOURCE COORDINATES                *
062553                  *         R26-27 = PTR TO DESTINATION                       *
062553                  *                                                           *
062553                  ************************************************************
062553                  MAP4MU  BSZ   0
062553 316 156 145              JSB   =MAP2MU          ;MAP X-COOR
062556                  MAP2MU  BSZ   0                ;MAP Y-COOR
062556 140 022 341              POMD  R40,+R22         ;GET XA
062561 150 341                  POMD  R50,+R22         ;GET XB
062563 012 345                  PUMD  R50,+R12         ;SAVE XB ON STACK
062565 140 345                  PUMD  R40,+R12         ;SAVE XA
062567 150 345                  PUMD  R50,+R12         ;SAVE XB AGAIN
062571 316 176 145              JSB   =MAP+            ;SCALE 1ST PAR
062574 012 343                  POMD  R#,-R12          ;POP XA
062576                  MAP+    BSZ   0                ;SCALE NEXT PAR
062576 150 024 341              POMD  R50,+R24         ;GET UNSCALED COOR
062601 231                      BCD
062602 316 377 377              JSB   =MPY10           ;X*XA
062605 316 377 377              JSB   =ADDROI          ;X*XA+XB
062610 343                      POMD  R#,-R#           ;R40,-R12
062611 026 345                  PUMD  R#,+R26          ;STORE SCALED COOR
062613 236                      RTN
062614                  MAPXY   BSZ   0                ;MAP A LOCATION
062614 316 217 145              JSB   =MAP1MU          ;MAP X-COOR
062617                  MAP1MU  BSZ   0                ;MAP Y-COOR
062617 140 022 341              POMD  R40,+R22         ;GET XA
062622 150 341                  POMD  R50,+R22         ;GET XB
062624 012 345                  PUMD  R50,+R12         ;SAVE A COPY
062626 316 176 145              JSB   =MAP+            ;MAP IT
062631 236                      RTN
```

```
>>>>>>          CHECK BOUNDARY                                                              <<<<<<
062632                          ****************************************************************
062632                          *                                                              *
062632                          *    CHECK IF SPECIFIED LIMITS EXCEED HARD BOUNDARY            *
062632                          *                                                              *
062632                          *      ENTRY:                                                  *
062632                          *        R24-25 = PTR TO SOURCE COORDINATES                    *
062632                          *        R26-27 = PTR TO BOUNDARY                              *
062632                          *                                                              *
062632                          *      EXIT:                                                   *
062632                          *        R70    = 0 IF WITHIN BOUNDARY                         *
062632                          *               # 0 IF EXCEED BOUNDARY &                       *
062632                          *                      DEFAULT TO BOUNDARY                     *
062632                          *                                                              *
062632                          ****************************************************************
062632                  CHKBND  BSZ  0
062632 120 223                  CLM  R20           ;INDEX TO X-LIMITS
062634 970 242                  STB  R20,R70       RESET FLAG
062636 316 244 145              JSB  =CHK1         ;CHECK AGAINST HARD BND
062641 120 250 020              LDB  R20,=20       ;INDEX TO Y-LIMITS
062644
062644                  CHK1    BSZ  0
062644 316 247 145              JSB  =CHK2         ;CHECK FOR XMIN
062647
062647                  CHK2    BSZ  0             ;CHECK FOR XMAX
062647 230                      BIN
062650 122 020 241              LDM  R22,R20       COPY INDEX
062653 026 303                  ADM  R22,R26       POINT TO ONE SIDE
062655 140 022 341              POMD R40,+R22      GET LOWER BOUND
062660 150 024 245              LDMD R50,R24       ;GET USER SPEC
062663 316 375 144              JSB  =COMFLT       ;COMPARE
062666 370 007                  JEN  CHK3          ;JIF > LOWER BOUND
062670 170 210                  ICB  R70           FLAG ON
062672 140 022 343              POMD R40,-R22      USE THE BOUND
062675 760 021                  JMP  CHKRTN
062677                  CHK3    BSZ  0
062677                          DRP  !40
062677 024 245                  LDMD R40,R24       ;GET USER SPEC
062701 150 022 341              POMD R50,+R22      GET UPPER BOUND
062704 316 375 144              JSB  =COMFLT       ;COMPARE
062707 024 245                  LDMD R#,R24        ;GET USER SPEC
062711 370 005                  JEN  CHKRTN        ;JIF IN BOUND
062713 170 210                  ICB  R70           FLAG ON
062715 140 022 343              POMD R40,-R22      USE UPPER BOUND
062720                  CHKRTN  BSZ  0
062720 024 345                  PUMD R#,+R24       ;STORE BACK & INC PTR
062722 236                      RTN
062723
```

```
>>>>>>        ORDER
1027 062723
1028 062723          **************************************************************
1028 062723          *
1029 062723          *     ORDER A SET OF USER SPECIFIED LIMITS                      *
1030 062723          *                                                               *
1031 062723          *       ENTRY:                                                  *
1032 062723          *         R26-27=POINTER TO USER SPEC                           *
1032 062723          *                                                               *
1032 062723          **************************************************************
1033 062723
1034 062723                   ORDER4  BSZ   0
1035 062723 316 326 145              JSB   =ORDER2
1036 062726                   ORDER2  BSZ   0
1037 062726 120 026 241              LDM   R20,R26        ;POINT TO 1ST PAR
1038 062731 140 341                  POMD  R40,+R26       ;POP 1ST PAR
1039 062733 122 241                  LDM   R22,R26        ;POINT TO 2ND PAR
1040 062735 150 341                  POMD  R50,+R26       ;POP 2ND PAR
1041 062737 316 375 144              JSB   =COMFLT        ;COMPARE
1042 062742 370 012                  JEN   ORDRTN         ;JIF ALREADY IN ORDER
1043 062744 020 245                  LDMD  R#,R20         ;ELSE SWAP THEM
1044 062746 150 022 245              LDMD  R50,R22
1045 062751 140 247                  STMD  R40,R22
1046 062753 150 020 247              STMD  R50,R20
1047 062756 236               ORDRTN  RTN
1048 062757
```

```
             GET SCALE FACTORS
062757
062757
062757                     *************************************************************
062757                     *
062757                     *     GET SCALE FACTORS FOR CURRENT UNIT                    *
062757                     *                                                           *
062757                     *       RETURN:                                             *
062757                     *         R22-23=POINTS TO THE SCALE FACTORS                *
062757                     *                                                           *
062757                     *************************************************************
062757
062757                     GSCALF  BSZ  0
062757 230                         BIN
062760 122 251 242                 LDM  R22,=XAGU     SET TO GU
062763 210
062764 323 127 212                 ADMD R22,=USER     ADJUST
062767 236                         RTN
062770
```

SOFT CLIP

```
062770
062770                  ***************************************************************
062770                  *
062770                  *      RUN TIME ROUTINE FOR SOFT CLIP                          *
062770                  *                                                              *
062770                  *                                                              *
062770                  ***************************************************************
062770 241                              OCT   241           ;PRIMARY ATTRIBUTE
062771
062771                          CLIP.   BSZ   0
062771 316 013 145                      JSB   =GFOUR        ;GET FOUR PARS TO TEMPS
062774 370 041                          JEN   NOVARC        JIF PARS NOT IN STMT
062776 124 251 062                      LDM   R24,=TMP1     ;SOURCE
063001 211
063002 126 251 200                      LDM   R26,=XS1      ;DESTINATION
063005 211
063006 316 357 145                      JSB   =GSCALF       ;GET SCALE FACTORS
063011 316 153 145                      JSB   =MAP4MU       ;MAP TO MU
063014                          CLIP1   BSZ   0
063014 124 251 200                      LDM   R24,=XS1      ;SOURCE
063017 211
063020 126 251 140                      LDM   R26,=XH1
063023 211
063024 316 232 145                      JSB   =CHKBND       ;CHECK BOUNDARY
063027                          CLIP2   BSZ   0
063027 126 251 200                      LDM   R26,=XS1      ;ORDER IT
063032 211
063033 316 323 145                      JSB   =ORDER4
063036 236                      CLIRTN  RTN
063037
063037                          NOVARC  BSS   0
063037 126 251 200                      LDM   R26,=XS1
063042 211
063043 316 050 146                      JSB   =NOMAP
063046 760 344                          JMP   CLIP1
063050
063050                          NOMAP   BSS   0
063050 233                              DCE
063051 370 010                          JEN   NOMAPX
063053 124 251 062                      LDM   R24,=TMP1
063056 211
063057 316 340 144                      JSB   =MOV40
063062 236                              RTN
063063                          NOMAPX  BSS   0
063063 106 213                          DCM   R6
063065 213                              DCM   R6
063066 236                              RTN
063067
```

```
            LOCATE
1103 063067
1103 063067                     ******************************************************************
1104 063067                     *
1105 063067                     *      RUN TIME ROUTINE FOR LOCATE                               *
1105 063067                     *                                                                *
1105 063067                     *                                                                *
1106 063067                     ******************************************************************
1107 063067 241                             OCT   241             ;PRIMARY ATTRIBUTE
1108 063070
1109 063070                     LOCAT.      BSZ   0
1110 063070 316 013 145                     JSB   =GFOUR          ;GET PARS
1111 063073 370 040                         JEN   NOVARL          JIF PARS NOT IN STMT
1112 063075 122 251 242                     LDM   R22,=XAGU       ;GET GU SCALE
1112 063100 210
1113 063101                     LOCAT0      BSS   0
1114 063101 124 251 062                     LDM   R24,=TMP1       ;SOURCE
1114 063104 211
1115 063105 126 251 240                     LDM   R26,=XV1        ;DESTINATION
1115 063110 211
1116 063111 316 153 145                     JSB   =MAP4MU         ;MAP TO MU
1117 063114                     LOCAT1      BSS   0
1118 063114 316 121 146                     JSB   =XVTOXS         DEFAULT SOFT CLIP
1119 063117 760 273                         JMP   CLIP1           CHECK BOUND & ORDER
1120 063121
1121 063121                     XVTOXS      BSS   0
1122 063121 124 251 240                     LDM   R24,=XV1        ;SOURCE
1122 063124 211
1123 063125                     TOXS1       BSS   0
1123 063125 126 251 200                     LDM   R26,=XS1        ;DESTINATION
1124 063130 211
1125 063131 316 340 144                     JSB   =MOV40          ;DEFAULT SOFT CLIP
1126 063134 236                 LOCRTN      RTN
1127 063135
1128 063135                     NOVARL      BSS   0
1129 063135 126 251 240                     LDM   R26,=XV1
1129 063140 211
1130 063141 316 050 146                     JSB   =NOMAP          MOVE TO PLACE
1131 063144 760 346                         JMP   LOCAT1
1132 063146
```

CALCULATE SCALING FACTORS

```
1134 063146
1135 063146
1136 063146
1137 063146
1138 063146
1139 063146
1140 063146
1141 063146
1141 063146
1141 063146
1142 063146
1143 063146 122 251 062
1143 063151 211
1144 063152 124 251 240
1144 063155 211
1145 063156 126 251 302
1145 063161 210
1146 063162 316 165 146
1147 063165
1148 063165
1149 063165 140 024 341
1150 063170 012 345
1151 063172 150 024 341
1152 063175 231
1153 063176 316 377 377
1154 063201
1155 063201 150 343
1156 063203 140 022 341
1157 063206 012 345
1158 063210 150 345
1159 063212 022 341
1160 063214 316 377 377
1161 063217 343
1162 063220 150 343
1163 063222 316 254 146
1164 063225
1165 063225
1166 063225 343
1167 063226 142 222
1168 063230 143 222
1169 063232 144 222
1170 063234 140 026 345
1171 063237 150 012 343
1172 063242 316 377 377
1173 063245 316 377 377
1174 063250 343
1175 063251 026 345
1176 063253 236
1177 063254
1178 063254
1179 063254 316 377 377
1180 063257 316 377 377
```

```
*************************************************************
*                                                           *
*     CALCULATE SCALING FACTORS                             *
*                                                           *
*       ENTRY:                                              *
*         R22-23 = PTR TO SOURCE COORDINATES                *
*         R24-25 = PTR TO TARGET COORDINATES                *
*         R26-27 = PTR TO WHERE FACTORS ARE TO BE STORED    *
*                                                           *
*************************************************************
SCALF   BSZ   0
        LDM   R22,=TMP1     ;SOURCE COORDINATES
        LDM   R24,=XY1      ;TARGET COORDINATES
        LDM   R26,=XAUU     ;DEST OF SCALE FACTOR
        JSB   =SCALF1       ;SCALE X-COOR
SCALF1  BSZ   0             ;SCALE Y-COOR
        POMD  R40,+R24      ;XMIN IN MU
        PUMD  R40,+R12      SAVE A COPY
        POMD  R50,+R24      ;XMAX IN MU
        BCD
        JSB   =SUB10        ;DXMU
        ARP   !12
        POMD  R50,-R12      POP OFF DXMU
        POMD  R40,+R22      ;XMIN IN OTHER UNIT
        PUMD  R40,+R12      SAVE A COPY
        PUMD  R50,+R12      PUSH DXMU BACK
        POMD  R50,+R22      ;XMAX IN OTHER UNIT
        JSB   =SUB10        ;DXOU
        POMD  R#,-R#        POP DXOU
        POMD  R50,-R#       POP DXMU
        JSB   =DIVIDE       DXMU / DXOU
        ORP   !40
        ARP   !12
        POMD  R40,-R12      POP OFF STACK
        CLB   R42           THROW AWAY LAST 6 DIGITS
        CLB   R43
        CLB   R44
        PUMD  R40,+R26      STORE XA
        POMD  R50,-R12      ;XMIN IN OTHER UNIT
        JSB   =MPY10        ;XMIN * DXMU / DXOU
        JSB   =SUBR01       ;= XB
        POMD  R#,-R#        ;POP OFF STACK
        PUMD  R#,+R26       ;SAVE IN PLACE
        RTN
DIVIDE  BSS   0
        JSB   =SEP15        UNPACK OPERANDS
        JSB   =DIV88        DO DIVIDE
```

```
>>>>>>           CALCULATE SCALING FACTORS                                       <<<<<<
 1 063262 316 377 377          JSB   =NFR5          PACK WITHOUT ROUNDING
   063265 140 012 345          PUMD,R40,+R12        PUSH ON STACK
1183 063270 236                RTN
1184 063271
```

```
>>>>>>>>          UNCLIP                                                        <<<<<<<<
1186 063271
1187 063271                     ***********************************************************
1187 063271                     *
1188 063271                     *      UNCLIP RUN TIME ROUTINE                              *
1188 063271                     *                                                           *
1188 063271                     *                                                           *
1189 063271                     ***********************************************************
1190 063271 241                          OCT  241            PRIMARY ATTRIBUTE
1191 063272
1192 063272                     UNCLI.   BSZ  0
1193 063272 124 251 140                 LDM  R24,=XH1       SET SOFT LIMIT
1193 063275 211
1194 063276 316 125 146                 JSB  =TOXS1         EQUAL TO HARD LIMIT
1195 063301 236                          RTN
1196 063302
```

```
                  CALCULATE GDU
     063302
     063302
     063302                   *****************************************************
     063302                   *
     063302                   *     CALCULATE GDU BY USING VIEW LIMITS            *
     063302                   *                                                   *
     063302                   *****************************************************
     063302                   CALGU   BSZ  0
     063302 124 251 240               LDM  R24,=XV1        ;POINT TO VIEW LIMITS
     063305 211
     063306 316 003 147               JSB  =ABSDXY         ;ABS(DELTA X/Y)
     063311 120 223                   CLM  R20             ;INITIALIZE INDEX
     063313 150 261 072               LDMD R50,=TMP2       ;FETCH DX
     063316 211
     063317 140 261 112               LDMD R40,=TMP4       ;FETCH DY
     063322 211
     063323 316 375 144               JSB  =COMFLT         ;COMPARE
     063326 370 003                   JEN  CALGU1          ;JIF DX>DY
     063330 120 250 020               LDB  R20,=20         ;IDX TO MAX(DX,DY)
     063333                   CALGU1  BSZ  0
     063333 150 020 265               LDMD R50,X20,TMP2 ;FETCH MAX(DX,DY)
     063336 072 211
     063340 122 251 020               LDM  R22,=20,0
     063343 000
     063344 227                       SRM  R22,R20         ;IDX TO MIN(DX,DY)
     063345 140 022 265               LDMD R40,X22,TMP2 ;FETCH MIN(DX,DY)
     063350 072 211
     063352 316 377 377               JSB  =DIV10
     063355                           DRP  !40
     063355                           ARP  !12
     063355 343                       POMD R40,-R12        ;POP OFF RESULT
     063356 312 002                   ADB  R#,=2           ;MULTIPLY BY 100
     063360 020 267 072               STMD R#,X20,TMP2     ;GDU OF LONG SIDE
     063363 211
     063364 316 034 147               JSB  =R100.          ;GET A 100.
     063367 022 267 072               STMD R#,X22,TMP2     ;GDU OF SHORT SIDE
     063372 211
     063373 223                       CLM  R#              ;GET 0.
     063374 263 062 211               STMD R#,=TMP1        ;GDU ORIGIN
     063377 263 102 211               STMD R#,=TMP3
     063402 236                       RTN
     063403
```

```
 ***                  ABS (DELTA X/Y)
1229 063403
1230 063403                     **************************************************
1230 063403                     *                                                *
1231 063403                     *     ABSOLUTE DIFFERENCE OF X OR Y              *
1232 063403                     *                                                *
1233 063403                     *       ENTRY:                                   *
1234 063403                     *         R24-25=POINTER TO LIMITS               *
1234 063403                     *                                                *
1234 063403                     **************************************************
1235 063403                     ABSDXY  BSZ  0
1236 063403 126 223                     CLM  R26            ;INDEX FOR ABS(DX)
1237 063405 316 013 147                 JSB  =ABSD          ;
1238 063410 126 250 020                 LDB  R26,=20        ;INDEX FOR ABS(DY)
1239 063413
1240 063413                     ABSD    BSZ  0
1241 063413 140 024 341                 POMD R40,+R24       ;GET MIN
1242 063416 150 341                     POMD R50,+R24       ;GET MAX
1243 063420 316 375 144                 JSB  =COMFLT        ;CALC DIFF
1244 063423 151 206                     LRB  R51            ;ABSOLUTE IT
1245 063425 204                         LLB  R51
1246 063426 150 026 267                 STMD R50,X26,TMP2 ;STORE IN TEMPS
1246 063431 072 211
1247 063433 236                         RTN
1248 063434
```

```
1250 063434
1251 063434                      ***********************************************
1251 063434                      *                                             *
1252 063434                      *     GENERATE A 100. IN R40-47               *
1252 063434                      *                                             *
1252 063434                      ***********************************************
1253 063434
1254 063434                      R100.    BSZ   0
1255 063434 316 042 147                   JSB   =R1.           ;GET A 1. IN R40
1256 063437 210                           ICB   R#             ;MULT BY 100
1257 063440 210                           ICB   R#
1258 063441 236                           RTN
1259 063442
1260 063442                      ***********************************************
1260 063442                      *                                             *
1261 063442                      *     GENERATE A 1. IN R40-47                 *
1261 063442                      *                                             *
1261 063442                      ***********************************************
1262 063442
1263 063442                      R1.      BSZ   0
1264 063442 140 223                       CLM   R40            ;SET TO 0
1265 063444 147 250 020                   LDB   R47,=10C       ;SET TO 1.
1266 063447 140                           DRP   40
1267 063450 236                           RTN
1268 063451
1269 063451                      ***********************************************
1269 063451                      *                                             *
1270 063451                      *     GENERATE A 2. IN R40-47                 *
1270 063451                      *                                             *
1270 063451                      ***********************************************
1271 063451
1272 063451                      R2.      BSZ   0
1273 063451 140 223                       CLM   R40            ;SET TO 0
1274 063453 147 250 040                   LDB   R47,=20C       ;SET TO 2
1275 063456 140                           DRP   40
1276 063457 236                           RTN
1277 063460
```

```
>>>>>>>          SCALE                                                                   <<<<<<<
1279 063460
1280 063460                           ***********************************************************
1280 063460                           *                                                         *
1281 063460                           *      RUN TIME ROUTINE FOR SCALE                         *
1281 063460                           *                                                         *
1281 063460                           ***********************************************************
1282 063460
1283 063460 241                              OCT  241           ;PRIMARY ATTRIBUTE
1284 063461
1285 063461                       SCALE.     BSZ  0
1286 063461 316 013 145                      JSB  =GFOUR        ;GET PARS
1287 063464
1288 063464                       SCALE1     BSZ  0
1289 063464 316 146 146                      JSB  =SCALF        ;CALC SCALE FACTORS
1290 063467 316 074 147                      JSB  =SETUU.       ;DEFAULT TO UU
1291 063472 236                              RTN
1292 063473
```

```
063473
063473                        ****************************************************************
063473                        *                                                              *
063473                        *     RUN TIME ROUTINE FOR SETUU                               *
063473                        *                                                              *
063473                        ****************************************************************
063473
063473 241                             OCT   241            ;PRIMARY ATTRIBUTE
063474
063474                        SETUU.   BSZ   0
063474 124 251 040                     LDM   R24,=40,0      SET USER FLAG
063477 000
063500 263 127 212                     STMD  R24,=USER
063503 236                             RTN
063504
063504                        ****************************************************************
063504                        *                                                              *
063504                        *     RUN TIME ROUTINE FOR SETGU                               *
063504                        *                                                              *
063504                        ****************************************************************
063504
063504 241                             OCT   241            ;PRIMARY ATTRIBUTE
063505
063505                        SETGU.   BSZ   0
063505 124 223                         CLM   R24
063507 263 127 212                     STMD  R24,=USER      RESET USER FLAG
063512 236                             RTN
063513
```

```
ASMA 03/14/81 =====================================================================
ITEM     LOC    OBJECT CODE   SRC=CCGRAF OBJ=GENGRA    9/11/1981 12:33 PM  PG
=====================================================================================
>>>>>>>        MSCALE
     063627
     063627                    **************************************************************
1358 063627                    *                                                            *
1359 063627                    *     RUN TIME ROUTINE FOR MSCALE                            *
1359 063627                    *                                                            *
1359 063627                    **************************************************************
1360 063627
1361 063627 241                         OCT  241            ;PRIMARY ATTRIBUTE
1362 063630
1363 063630                    MSCAL.   BSZ  0
1364 063630 316 040 145                JSB  =GTWO          ;GET TWO PARS
1365 063633 122 251 342                LDM  R22,=XAMM      ;GET SCALE FACTOR
1365 063636 210
1366 063637 124 251 062                LDM  R24,=TMP1      ;SOURCE
1366 063642 211
1367 063643 026 243                    STM  R24,R26        ;DESTINATION
1368 063645 316 214 145                JSB  =MAPXY         ;MAP TO MU
1369 063650 124 251 240                LDM  R24,=XY1       ;SOURCE 1
1369 063653 211
1370 063654 126 251 062                LDM  R26,=TMP1      ;SOURCE & DEST
1370 063657 211
1371 063660 316 375 161                JSB  =ADDCR1        ;CALC XBUU
1372 063663 024 341                    POMD R#,+R24        ;INCREMENT R24
1373 063665 316 375 161                JSB  =ADDCR1        ;CALC YBUU
1374 063670 124 251 062                LDM  R24,=TMP1      UPDATE XBUU,YBUU
1374 063673 211
1375 063674 126 251 312                LDM  R26,=XBUU
1375 063677 210
1376 063700 316 365 144                JSB  =MOV**         MOVE TO DESTINATION
1377 063703 261 342 210                LDMD R#,=XAMM       ;GET MU/MM
1378 063706 263 302 210                STMD R#,=XAUU       ;SET XAUU
1379 063711 263 322 210                STMD R#,=YAUU       ;SET YAUU
1380 063714 316 074 147                JSB  =SETUU.        ;SET TO USER
1381 063717 236                        RTN
1382 063720
```

```
<<<<>>>   LIMIT                                                     <<<<>>>
063513
063513                          ***********************************************
063513                          *                                             *
063513                          *     RUN TIME ROUTINE FOR LIMIT              *
063513                          *                                             *
063513                          ***********************************************
063513
063513 241                              OCT  241            ;PRIMARY ATTRIBUTE
063514
063514                          LIMIT   BSZ  0
063514 316 013 145                      JSB  =GFOUR          ;GET PARS
063517 370 046                          JEN  NOVARS          PARS IN STATEMENT ?
063521 122 251 342                      LDM  R22,=XAMM       ;GET SCALE FACTOR
063524 210
063525 124 251 062                      LDM  R24,=TMP1       ;SOURCE
063530 211
063531 026 243                          STM  R24,R26         ;DESTINATION
063533 316 153 145                      JSB  =MAP4MU         ;MAP TO MU
063536                          LIMIT1  BSS  0
063536 124 251 062                      LDM  R24,=TMP1
063541 211
063542 262 002 207                      STBD R24,=NONSTD     FLAG NON-STD LIMIT
063545 130 251 026                      LDM  R30,=26,0
063550 000
063551 316 125 145                      JSB  =UPLTDR
063554 124 251 062                      LDM  R24,=TMP1
063557 211
063560 316 156 173                      JSB  =TOXV1
063563 316 142 150                      JSB  =GPON3          ;DO PWR ON LVL 3
063566 236                      LIMRTN  RTN
063567
063567                          NOVARS  BSS  0
063567 126 251 240                      LDM  R26,=XV1
063572 211
063573 316 050 146                      JSB  =NOMAP
063576 360 336                          JMP  LIMIT1
063600
063600                          C26     BSS  0
063600 124 251 062                      LDM  R24,=TMP1       PT TO SOURCE COOR
063603 211
063604 126 251 300                      LDM  R26,=XM1        PT TO MACHINE BNDY
063607 211
063610 316 232 145                      JSB  =CHKBND         CHECK IF IN BNDY
063613 170 220                          TSB  R70
063615 367 007                          JZR  CHRTN           JIF YES
063617 316 327 173                      JSB  =ROMERR         ELSE GIVE A WARNING
063622 316 377 377                      JSB  =WARN
063625 001                              OCT  1D
063626                          CHRTN   BSS  0
063626 236                              RTN
063627
```

```
                      SHOW
063720
063720                   ****************************************************
063720                   *                                                  *
063720                   *     RUN TIME ROUTINE FOR SHOW                    *
063720                   *                                                  *
063720                   ****************************************************
063720
063720 241                          OCT   241            ;PRIMARY ATTRIBUTE
063721
063721                   SHOW.      BSZ   0
063721 316 061 147                  JSB   =SCALE.
063724 120 223                      CLM   R20            ;RESET INDEXES
063726 122 223                      CLM   R22
063730 140 261 302                  LDMD  R40,=XAUU      ;GET X RATIO
063733 210
063734 141 206                      LRB   R41            ;ABSOLUTE IT
063736 204                          LLB   R41
063737 150 261 322                  LDMD  R50,=YAUU      ;GET Y RATIO
063742 210
063743 151 206                      LRB   R51            ;ABSOLUTE IT
063745 204                          LLB   R51
063746 316 375 144                  JSB   =COMFLT        ;COMPARE
063751 371 014                      JEZ   ADJ-X          ;JIF YAUU<XAUU
063753 150 221                      TSM   R50            ;
063755 367 106                      JZR   SHORTN         ;JIF ALREADY EQUAL
063757 120 251 020                  LDM   R20,=20,0      ;SET INDEXES TO Y
063762 000
063763 122 251 360                  LDM   R22,=360,377
063766 377
063767                   ADJ-X      BSZ   0
063767 140 020 265                  LDMD  R40,X20,TMP1 ;X(Y) MIN
063772 062 211
063774 150 265 072                  LDMD  R50,X20,TMP2 ;X(Y) MAX
063777 211
064000 316 377 377                  JSB   =ADD10
064003 150 343                      POMD  R50,-R#        ;POP OFF STACK
064005 316 051 147                  JSB   =R2.           ;GET A 2.
064010 316 377 377                  JSB   =DIV10         ;COOR OF MIDDLE PT
064013                              DRP   !40
064013 022 265 322                  LDMD  R40,X22,YAUU ;NEW XAUU
064016 210
064017 141 206                      LRB   R41            ;THROW AWAY SIGN
064021 150 020 265                  LDMD  R50,X20,XAUU ;OLD XAUU
064024 302 210
064026 137 051 240                  LDB   R37,R51        ;COPY SIGN
064031 206                          LRB   R37            ;SHIFT OUT SIGN
064032 141 200                      ELB   R41            ;SHIFT INTO NEW XAUU
064034 140 020 267                  STMD  R40,X20,XAUU ;STORE IN PLACE
064037 302 210
064041 316 377 377                  JSB   =SUB10         ;DIFFERENCE
064044 316 377 377                  JSB   =MPYROI        ;ADJUSTMENT
064047 343                          POMD  R#,-R#         ;POP OFF STACK
064050 150 020 265                  LDMD  R50,X20,XBUU ;OLD XB
```

```
                SHOW                                                    <<<<<<<<
     064053 312 210
     064055 316 377 377          JSB  =4DD10        ;NEW X8
1426 064060 343                  POMD R#,-R#        ;POP OFF STACK
1427 064061 020 267 312          STMD R#,X20,XBUU   ;STORE BACK
1427 064064 210
1428 064065 236          SHORTN  RTN
1429 064066
```

```
            LORG
     064066
1432 064066                      ***********************************************************
1432 064066                      *                                                         *
1433 064066                      *     RUN TIME ROUTINE FOR  LORG                          *
1433 064066                      *                                                         *
1433 064066                      ***********************************************************
1434 064066
1435 064066 241                          OCT   241            ;PRIMARY ATTRIBUTE
1436 064067
1437 064067                     LORG.    BSZ   0
1438 064067 316 377 377                 JSB   =ONEB          GET PARAMETER
1439 064072                              DRP   !76
1440 064072 367 007                     JZR   LORG1          ;JIF ZERO
1441 064074 364 005                     JNG   LORG1          JIF NEGATIVE
1442 064076 311 012 000                 CMM   R76,=12,0      ;CMPARE WITH 10
1443 064101 372 002                     JNC   LORG2          ;JIF PAR<10
1444 064103                     LORG1    BSS   0
1445 064103                              DRP   !76
1446 064103 223                         CLM   R76            ;DEFAULT TO 1
1447 064104 210                         ICB   R76
1448 064105                     LORG2    BSZ   0
1449 064105 176 212                     DCB   R76            BIAS BY 1
1450 064107                     LORG3    BSS   0              MOD 3
1451 064107 176 314 003                 SBB   R76,=3         SUB BY 3
1452 064112 373 006                     JCY   LORG4          JIF >= 0
1453 064114 312 003                     ADB   R76,=3         RESTORE REMAINDER
1454 064116 263 024 212                 STMD  R76,=YLORG     SAVE LORG INDEXES
1455 064121 236                         RTN
1456 064122
1457 064122                     LORG4    BSS   0
1458 064122 177 210                     ICB   R77            INC QUOTIENT
1459 064124 760 361                     JMP   LORG3          LOOP BACK
1460 064126
```

```
1462 064126
1463 064126 241                     OCT  241
1464 064127
1465 064127             PLOTR.      BSS  0              PLOTTER IS
1466 064127 316 377 377             JSB  =ONEB          GET PLOTTER #
1467 064132                         DRP  !76            IN BINARY MODE
1468 064132 263 133 211             STMD R76,=PLOTR#    SAVE IT
1469 064135 130 223                 CLM  R30            PLOTTER IS
1470 064137 316 125 145             JSB  =UPLTDR
1471 064142
1472 064142            ***************************************************************
1472 064142            *                                                             *
1473 064142            *     POWER ON LEVEL 3 ROUTINE                                *
1473 064142            *                                                             *
1473 064142            ***************************************************************
1474 064142
1475 064142            GPON3       BSZ  0
1476 064142 316 302 146             JSB  =CALGU         CALCULATE GDU
1477 064145            ******* LDM  R24,=XV1       ! SET HARD LIMIT
1478 064145            ******* LDM  R26,=XH1       !
1479 064145            ******* JSB  =MOV40         !
1480 064145            ******* LDM  R26,=XH1       ! ORDER IT
1481 064145            ******* JSB  =ORDER4        !
1482 064145 316 064 147             JSB  =SCALE1        SCALE BY GDU
1483 064150 124 251 302             LDM  R24,=XAUU      DEFAULT UU=GU
1483 064153 210
1484 064154 126 251 242             LDM  R26,=XAGU
1484 064157 210
1485 064160 316 340 144             JSB  =MOV40
1486 064163 316 121 146             JSB  =XYTOXS        DEFAULT SOFT LIMIT
1487 064166 316 027 146             JSB  =CLIP2
1488 064171 124 251 200             LDM  R24,=XS1       HARD LIMIT=SOFT LIMIT
1488 064174 211
1489 064175 126 251 140             LDM  R26,=XH1
1489 064200 211
1490 064201 316 340 144             JSB  =MOV40
1491 064204 126 251 002             LDM  R26,=XPEN      DEFAULT PEN LOCATION IS
1491 064207 211
1492 064210 316 346 150             JSB  =FROMXH        LOWER LEFT CORNER
1493 064213 316 346 150             JSB  =FROMXH        ALSO DEFAULT ORGIN
1494 064216 316 346 150             JSB  =FROMXH        ALSO DEFAULT CURSOR
1495 064221 316 357 150             JSB  =PENUP.        DEFAULT PEN POS IS UP
1496 064224 316 042 147             JSB  =R1.           DEFAULT LINE TYPE=1
1497 064227 012 345                 PUMD R#,+R12
1498 064231 316 336 154             JSB  =LINET.
1499 064234 316 042 147             JSB  =R1.           DEFAULT PEN IS 1
1500 064237 012 345                 PUMD R#,+R12
1501 064241 316 002 151             JSB  =PEN.
1502 064244 316 042 147             JSB  =R1.           DEFAULT LORG IS 1
1503 064247 012 345                 PUMD R#,+R12
1504 064251 316 067 150             JSB  =LORG.
1505 064254 140 223                 CLM  R40
1506 064256 263 036 212             STMD R40,=CHITE     FIX
```

```
                              PLOTTER IS / GPON3
064261 263 326 212                   STMD R40,=CWIDE
064264 262 125 211                   STBD R40,=CLIPBA  RESET CLIPBA FLAG
064267 012 345                       PUMD R40,+R12     DEFAULT LDIR 0
064271 316 052 156                   JSB  =LDIR.
064274 140 223                       CLM  R40          DEFAULT PDIR 0
064276 012 345                       PUMD R40,+R12
064300 316 056 172                   JSB  =PDIR.
064303                       DEFCSZ  BSS  0            DEFAULT CSIZE
064303 140 223                       CLM  R40
064305 147 250 120                   LDB  R47,=50C     DEF 5.0 FOR CRT
064310 102 261 133                   LDMD R2,=PLOTR#   GET PLOTTER #
064313 211
064314 311 003 000                   CMM  R2,=3,0      CRT ?
064317 372 003                       JNC  DEFCS1       JIF YES
064321 147 250 060                   LDB  R47,=30C     DEF 3.0 FOR PLOTTER
064324                       DEFCS1  BSS  0
064324 140 012 345                   PUMD R40,+R12
064327 316 170 155                   JSB  =CSIZE.
064332                       PNTOMP  BSS  0
064332 124 251 002                   LDM  R24,=XPEN    SET FOR BPLOT
064335 211
064336                       TOXMAP  BSS  0
064336 126 251 323                   LDM  R26,=XMAP
064341 212
064342 316 343 144                   JSB  =MOV20
064345 236                           RTN
064346
064346                       FROMXH  BSS  0            UTILITY
064346 124 251 140                   LDM  R24,=XH1
064351 211
064352 316 354 144                   JSB  =MOVXY
064355 236                           RTN
064356
```

PEN UP/DOWN

```
064356
064356                *****************************************************
064356                *                                                   *
064356                *    PEN UP/DOWN RUN TIME ROUTINE                   *
064356                *                                                   *
064356                *     PENDN = PEN DOWN FLAG                         *
064356                *           = EVEN IF PEN UP                        *
064356                *           = ODD  IF PEN DOWN                      *
064356                *                                                   *
064356                *****************************************************
064356
064356 241                    OCT  241
064357
064357                PENUP.  BSZ  0
064357 130 223                CLM  R30           LIFT UP PEN
064361 262 126 211            STBD R30,=PENDN    STORE BACK
064364 250 012                LDB  R30,=12
064366 316 125 145            JSB  =UPLTDR
064371 236                    RTN
064372
064372                PENDN.  BSZ  0
064372 100 223                CLM  R0
064374 210                    ICB  R0            SET PEN DOWN
064375 262 126 211            STBD R0,=PENDN
064400 236                    RTN
064401
064401
```

```
1561 064401
1562 064401                      *****************************************************************
1563 064401                      *                                                               *
1564 064401                      *     PEN SELECTION RUN TIME ROUTINE                            *
1564 064401                      *                                                               *
1564 064401                      *****************************************************************
1565 064401
1566 064401 241                          OCT  241
1567 064402
1568 064402                      PEN.    BSZ  0
1569 064402 316 377 377                 JSB  =ONEB           GET PARAMETER
1570 064405 130 251 004                 LDM  R30,=4,0        CALL DRIVER
1570 064410 000
1571 064411 316 125 145                 JSB  =UPLTDR
1572 064414 236                         RTN
1573 064415
1574 064415                      C4      BSS  0
1575 064415 146 221                     TSM  R46            CHECK PEN #
1576 064417 367 027                     JZR  PEN0           JIF PEN 0
1577 064421 365 015                     JPS  PEN+           JIF POSITIVE PEN
1578 064423 311 376 377                 CMM  R46,=376,377 PEN < -3 ?
1579 064426 373 003                     JCY  C10.1          JIF NO
1580 064430 251 377 377                 LDM  R46,=377,377 ELSE SET TO -1
1581 064433                      C10.1   BSS  0
1582 064433 176 223                     CLM  R76            PLOTSY FOR NEG PEN
1583 064435 216                         NCB  R#
1584 064436 360 005                     JMP  SETPEN
1585 064440                      PEN+    BSZ  0
1586 064440                              DRP  !46
1587 064440 251 001 000                 LDM  R46,=1,0       SET TO PEN 1
1588 064443 176 223                     CLM  R76            PLOTSY FOR POS PEN
1589 064445                      SETPEN  BSZ  0
1590 064445                              DRP  !76
1591 064445 262 151 200                 STBD R#,=PLOTSY     STORE IN PLOTSY
1592 064450                      PEN0    BSZ  0
1593 064450 046 241                     LDM  R#,R46         COPY PEN #
1594 064452 205                         LLM  R#             * 3 FOR INDEX
1595 064453 303                         ADM  R#,R46
1596 064454 263 135 211                 STMD R#,=PEN#
1597 064457 236                  PENRTN  RTN
1598 064460
```

```
064460
064460                     ******************************************************
064460                     *                                                    *
064460                     *     MAPPING ABS/INCR PARAMETERS TO MU              *
064460                     *                                                    *
064460                     ******************************************************
064460
064460                     MAPAR   BSZ  0
064460 316 040 145                 JSB  =GTWO         GET TWO PARS
064463 124 251 062                 LDM  R24,=TMP1     SOURCE
064466 211
064467 126 251 062                 LDM  R26,=TMP1     DESTINATION
064472 211
064473 316 357 145                 JSB  =GSCALF       GET SCALE FACTOR
064476 316 214 145                 JSB  =MAPXY        MAP TO MU
064501 236                         RTN
064502
064502
064502                     MAPIPA  BSZ  0
064502 316 114 151                 JSB  =MAPAR2       MAP TWO INCREMENTS
064505 124 014 241                 LDM  R24,R14       GET REFERENCE PT
064510 316 366 161                 JSB  =ADDCR+       ADD PEN ADDR
064513 236                         RTN
064514
064514                     MAPAR2  BSS  0             MAP INCREMENT
064514 316 127 151                 JSB  =MAPIC        GET 2 PAR & SCALE THEM
064517 124 251 066                 LDM  R24,=SINPD
064522 212
064523 316 050 162                 JSB  =ROTAXY       ROTATE BY PDIR
064526 236                         RTN
064527
064527                     MAPIC   BSS  0
064527 316 040 145                 JSB  =GTWO         GET TWO PARS
064532 124 251 062                 LDM  R24,=TMP1     SOURCE
064535 211
064536 026 243                     STM  R24,R26       DEST
064540 316 357 145                 JSB  =GSCALF       GET SCALE FACTOR
064543                     *****   FALL THROUGH       ****************
064543
064543                     MAPIC2  BSS  0
064543 316 150 151                 JSB  =MAPIC1
064546 022 341                     POMD R#,+R22       SKIP XA
064550
064550                     MAPIC1  BSS  0
064550 150 024 341                 POMD R50,+R24      GET SOURCE
064553 140 022 341                 POMD R40,+R22      GET RATIO OF SCALE
064556 316 377 377                 JSB  =MPY10
064561 343                         POMD R#,-R#        POP OFF STACK
064562 026 345                     PUMD R#,+R26       STORE IN DEST
064564 236                         RTN
064565
```

```
                        UPDATE PEN LOCATION
064565
064565                  ********************************************************
064565                  *                                                      *
064565                  *     UPDATE PEN LOCATION                              *
064565                  *                                                      *
064565                  ********************************************************
064565
064565                  UPDPEN  BSZ   0
064565 124 251 002              LDM   R24,=XPEN     GET PEN ADDR
064570 211
064571 316 042 152              JSB   =TOTMP3       MOVE TO TEMP LOC
064574 126 251 002              LDM   R26,=XPEN     GET PEN ADDR
064577 211
064600                  FRTMP1  BSS   0
064600 124 251 062              LDM   R24,=TMP1     GET NEW LOC
064603 211
064604 316 343 144              JSB   =MOV20        UPDATE PEN LOC
064607 236                      RTN
064610
064610                  UPDORG  BSS   0
064610 126 251 022              LDM   R26,=XORG     PT TO ORG
064613 211
064614 760 362                  JMP   FRTMP1
064616
064616                  UPDPT   BSS   0
064616 124 251 150              LDM   R24,=XLPEN    XLPEN TO TMP3
064621 215
064622 316 042 152              JSB   =TOTMP3
064625 126 251 150              LDM   R26,=XLPEN    TMP1 TO XLPEN
064630 215
064631 760 345                  JMP   FRTMP1
064633
```

```
>>>>>>>            MOVE / IMOVE                                                         <<<<<<<<
1669 064633
1670 064633                     ******************************************************************
1670 064633                     *                                                                *
1671 064633                     *      MOVE RUN TIME ROUTINE                                     *
1671 064633                     *                                                                *
1671 064633                     ******************************************************************
1672 064633
1673 064633 241                         OCT   241             PRIMARY ATTRIBUTE
1674 064634
1675 064634                     MOVE:   BSZ   0
1676 064634 126 251 376                 LDM   R26,=376,377 PEN CONTROL = -2
1676 064637 377
1677 064640 760 071                     JMP   DRAW.1          GOTO DRAW
1678 064642
1679 064642                     ******************************************************************
1679 064642                     *                                                                *
1680 064642                     *      IMOVE RUN TIME ROUTINE                                    *
1680 064642                     *                                                                *
1680 064642                     ******************************************************************
1681 064642
1682 064642 241                         OCT   241
1683 064643
1684 064643                     IMOVE.  BSZ   0
1685 064643 126 251 376                 LDM   R26,=376,377 PEN CONTROL = -2
1685 064646 377
1686 064647 760 041                     JMP   IDRAW1          GOTO COMMON ROUTINE
1687 064651
```

```
064651
064651                  ****************************************************
064651                  *                                                  *
064651                  *    PLOT RUN TIME ROUTINE                         *
064651                  *                                                  *
064651                  ****************************************************
064651
064651 241                     OCT   241           ;PRIMARY ATTRIBUTE
064652
064652                  PLOT.  BSZ   0
064652 316 153 152             JSB   =GETPC         GET PEN CONTROL
064655 760 054                 JMP   DRAW.1         GOTO COMMON ROUTINE
064657
064657                  ****************************************************
064657                  *                                                  *
064657                  *    IPLOT RUN TIME ROUTINE                        *
064657                  *                                                  *
064657                  ****************************************************
064657
064657 241                     OCT   241
064660
064660                  IPLOT. BSZ   0
064660 316 153 152             JSB   =GETPC         GET PEN CONTROL
064663 360 025                 JMP   IDRAW1         GOTO COMMON ROUTINE
064665
064665                  ****************************************************
064665                  *                                                  *
064665                  *    RPLOT RUN TIME ROUTINE                        *
064665                  *                                                  *
064665                  ****************************************************
064665
064665 241                     OCT 241              PRIMARY ATTRIBUTE
064666
064666                  RPLOT. BSZ   0
064666 316 153 152             JSB   =GETPC         GET PEN CONTROL
064671 263 131 211             STMD  R#,=PCNTRL     SAVE IT
064674 114 251 022             LDM   R14,=XORG      RELATIVE TO ORGIN
064677 211
064700 316 102 151             JSB   =MAPIPA        MAP PARS
064703 360 037                 JMP   DRW.25
064705                  *      LDM   R24,=XPEN      GET OLD PEN ADDR
064705                  *      LDM   R26,=TMP3      BUT DON'T UPDATE PEN
064705                  *      JSB   =MOV20
064705                  *      JMP   DRAW.3
064705
```

```
>>>>>          DRAW / IDRAW                                                  <<<<<<<
1725 064705
     064705                       ************************************************
1726 064705                       *                                              *
1727 064705                       *      IDRAW RUN TIME ROUTINE                  *
1727 064705                       *                                              *
1727 064705                       ************************************************
1728 064705
1729 064705 241                           OCT   241
1730 064706
1731 064706               IDRAW.   BSZ   0
1732 064706 126 251 377            LDM   R26,=377,377 PEN CONTROL = -1
1732 064711 377
1733 064712               IDRAW1   BSS   0
1734 064712 263 131 211            STMD  R#,=PCNTRL     SET PEN CONTROL
1735 064715 114 251 002            LDM   R14,=XPEN      RELATIVE TO PEN
1735 064720 211
1736 064721 316 102 151            JSB   =MAPIPA
1737 064724 360 013                JMP   DRAW.2
1738 064726
1739 064726                       ************************************************
1739 064726                       *                                              *
1740 064726                       *     DRAW RUN TIME ROUTINE                    *
1740 064726                       *                                              *
1740 064726                       ************************************************
1741 064726
1742 064726 241                           OCT   241
1743 064727
17   064727               DRAW.    BSZ   0
17   064727 126 251 377            LDM   R26,=377,377 PEN CONTROL = -1
1745 064732 377
1746 064733               DRAW.1   BSS   0
1747 064733 263 131 211            STMD  R#,=PCNTRL     SAVE PEN CONTROL
1748 064736 316 060 151            JSB   =MAPAR         ;MAP PARS TO MU
1749 064741               DRAW.2   BSS   0
1750 064741 316 210 151            JSB   =UPDORG        UPDATE ORGIN
1751 064744               DRW.25   BSS   0
1752 064744 316 165 151            JSB   =UPDPEN        UPDATE PEN LOCATION
1753 064747               DRAW.3   BSS   0
1754 064747 136 250 001            LDB   R36,=1         SET CLIP FLAG
1755 064752 262 127 211            STBD  R36,=CLPFLG
1756 064755               DRAW.+   BSS   0
1757 064755 316 170 152            JSB   =CLIPLN        ;CLIP THE LINE
1758 064760 370 070                JEN   DRWRTN         ;EXIT IF OUT OF WINDOW
1759 064762 124 251 062            LDM   R24,=TMP1      SET FOR SPLOT
1759 064765 211
1760 064766 316 336 150            JSB   =TOXMAP
1761 064771 136 260 126            LDBD  R36,=PENDN     GET PEN DOWN FLAG
1761 064774 211
1762 064775 362 007                JOD   DRAW.4         JIF PEN DOWN
1763 064777 261 131 211            LDMD  R36,=PCNTRL    GET PEN CONTROL
1764 065002 364 002                JNG   DRAW.4         JIF CONTROL BEFORE DRAW
1765 065004 362 007                JOD   DRAW.5         JIF DROP PEN AFTER
1766 065006               DRAW.4   BSS   0
```

```
065006                        ORP   136
065006 260 137 211            LDBD  R36,=LTYPE#    GET LINE TYPE
065011 310 002                CMB   R36,=2         TYPE 2 ?
065013 366 011                JNZ   DRAW.6         JIF NOT
065015                DRAW.5  BSS   0
065015 102 260 125            LDBD  R2,=CLIPBA     GET CLIP FLAG
065020 211
065021 362 027                JOD   DRWRTN         JIF CLIP A FOR LNTYPE 2
065023 316 036 152            JSB   =T1TOT3        DRAW END POINT
065026                DRAW.6  BSS   0
065026 130 251 010            LDM   R30,=10,0      CALL UNIVERSAL DRIVER
065031 000
065032 316 125 145            JSB   =UPLTDR
065035 236                    RTN
065036
065036                T1TOT3  BSS   0
065036 124 251 062            LDM   24,=TMP1       SET PT B = PT A
065041 211
065042                TOTMP3  BSS   0
065042 126 251 102            LDM   R26,=TMP3
065045 211
065046 316 343 144            JSB   =MOV20
065051 236                    RTN
065052
065052                DRWRTN  BSS   0
065052 260 131 211            LDBD  R#,=PCNTRL     GET PEN CONTROL
065055 262 126 211            STBD  R#,=PENDN      SET PEN DOWN FLAG
065060 236                    RTN
065061
065061                C10     BSS   0              DRAW LINE ON CRT
065061 134 261 131            LDMD  R34,=PCNTRL    GET PEN CONTROL
065064 211
065065 365 012                JPS   C14.2          JIF CONTROL AFTER DRAW
065067 363 005                JEV   C14.1          ELSE DO PEN CONTROL
065071 316 372 150            JSB   =PENDN.        PEN DOWN
065074 360 003                JMP   C14.2
065076 316 357 150    C14.1   JSB   =PENUP.        PEN UP
065101                C14.2   BSS   0
065101 100 260 126            LDBD  R0,=PENDN      GET PEN DN FLAG
065104 211
065105 362 011                JOD   C14.3          JIF PEN DN
065107 134 221                TSM   R34            CHECK PEN CONTROL
065111 364 037                JNG   C14.6          JIF CNTRL BEFORE DRAW
065113 363 035                JEV   C14.6          JIF LIFT PEN AFTER
065115 316 036 152            JSB   =T1TOT3        DRAW END POINT
065120                C14.3   BSS   0
065120 260 135 211            LDBD  R#,=PEN#       GET PEN #
065123 367 006                JZR   C14.4          JIF PEN 0
065125 316 377 377            JSB   =GRAPH
065130 316 106 153            JSB   =DRAWC         DRAW IT
065133                C14.4   BSS   0
065133 134 261 131            LDMD  R34,=PCNTRL    GET PEN CONTROL
065136 211
```

```
<<<<>>          DRAW / IDRAW                                                   <<<< <<
065137 764 011                 JNG   C14.6          JIF CONTROL BEFORE DRAW
065141 763 004                 JEV   C14.5          JIF PEN UP
065143 316 372 150             JSB   =PENDN.        PEN DOWN
065146 236                     RTN
065147 316 357 150 C14.5       JSB   =PENUP.
065152 236         C14.6       RTN
065153
065153             GETPC       BSS   0
065153 316 377 377             JSB   =STACKX
065156 376 004                 JRZ   GETPC1         JIF TWO PARS
065160 316 377 377             JSB   =ONEB          GET PEN CONTROL
065163 236                     RTN
065164             GETPC1      BSS   0
065164 251 001 000             LDM   R#,=1,0        DEFAULT PEN CONTROL
065167 236                     RTN
065170
```

```
>>>>>>>          CLIP THE LINE                                                        <<<<<<<
065170
065170                          *****************************************************************
065170                          *
065170                          *     CLIP THE LINE FROM PEN TO CURRENT POINT IN TMP1-2 *
065170                          *
065170                          *       STORAGE ALLOCATION:                                     *
065170                          *         TMP1,2=COORDINATE OF CURRENT POINT (PT A)             *
065170                          *         TMP3,4=COORDINATE OF LAST PEN LOCATION (PT B) *
065170                          *         XS1    =XMIN OF CLIP AREA                             *
065170                          *         XS2    =XMAX OF CLIP AREA                             *
065170                          *         XS3    =YMIN OF CLIP AREA                             *
065170                          *         XS4    =YMAX OF CLIP AREA                             *
065170                          *                                                               *
065170                          *       REGISTER USAGE:                                         *
065170                          *         R20=INDEX TO ONE SIDE OF THE CLIP AREA                *
065170                          *         R22=INDEX TO X- (Y-) COMPONENT                        *
065170                          *         R24=WORKING POINTER                                   *
065170                          *         R26=WORKING POINTER                                   *
065170                          *         R14=INDEX TO STORAGE SPACE                            *
065170                          *         CLPFLG=0 IF NO CLIPPING IS NEEDED                     *
065170                          *               #0 IF DO NEED CLIPPING                          *
065170                          *                                                               *
065170                          *       RETURN:                                                 *
065170                          *         E=0 IF THE LINE IS AT LEAST PARTIALLY IN CLIP A
065170                          *         E=1 IF THE LINE IS COMPLETELY OUT OF CLIP AREA*
065170                          *                                                               *
065170                          *****************************************************************
065170
065170                          CLIPLN  BSZ  0
065170                          *******CLB  R20            RESET CLIP BA FLAG
065170                          *******STBD R20,=CLIPBA
065170
065170 230                              BIN
065171 120 260 125                      LDBD R20,=CLIPBA  GET PREVIOUS CLIBA
065174 211
065175 206                              LRB  R20            SHIFT A-->B
065176 202                              ERB  R20
065177 262 125 211                      STBD R20,=CLIPBA  STORE BACK
065202
065202 260 127 211                      LDBD R20,=CLPFLG  GET CLIP FLAG
065205 367 135                          JZR  DONE           JIF NO NEED TO CLIP
065207                          S1      BSZ  0              ;INITIALIZE
065207 120 223                          CLM  R20            ;INDEX TO 1ST SIDE
065211 122 223                          CLM  R22            ;INDEX TO X-COMPONENT
065213                          S2      BSZ  0
065213 124 022 241                      LDM  R24,R22        ;INDEX TO COMP OF PT A
065216 316 346 152                      JSB  =IN?           ;CHECK IF IN BOUND
065221 230                              BIN
065222 124 312 020                      ADB  R24,=20        ;INDEX TO COMP OF PT B
065225 370 016                          JEN  S5             ;JIF A IN BOUND
065227                          S3      BSZ  0
065227 316 346 152                      JSB  =IN?           ;CHECK IF IN BOUND
065232 370 002                          JEN  S4             ;JIF B IN BOUND
```

```
                        CLIP THE LINE
065234 234                            ICE                  ;BOTH OUT OF BOUND
065235 236                            RTN
065236
065236                  S4            BSZ  0
065236 114 223                        CLM  R14             ;IDX OF INTRCPT OF PT A
065240 102 250 001                    LDB  R2,=1           FLAG INTRCPT A
065243 760 014                        JMP  CLIP+           ;CALC INTERCEPT PT
065245
065245                  S5            BSZ  0
065245 316 346 152                    JSB  =IN?            ;CHECK IF IN BOUND
065250 370 051                        JEN  NEXT            ;BOTH IN BOUND
065252 114 251 020                    LDM  R14,=20,0       ;IDX OF INTRCPT OF PT B
065255 000
065256 102 250 200                    LDB  R2,=200         FLAG INTRCPT PT B
065261
065261                  CLIP+         BSZ  0
065261 103 260 125                    LDBD R3,=CLIPBA      GET FLAG
065264 211
065265 002 224                        ORB  R3,R2           SET FLAG
065267 262 125 211                    STBD R3,=CLIPBA      UPDATE FLAG
065272 114 022 203                    ADM  R14,R22         ;ADD IDX OF COMPONENT
065275 316 014 153                    JSB  =INTRC          ;CALC INTERCEPT POINT
065300 243                            POMD R#,-R#          ;GET THE SIDE
065301 014 267 062                    STMD R#,X14,TMP1     ;ADJUST THE POINT
065304 211
065305 124 251 010                    LDM  R24,=10,0
065310 000
065311 114 024 227                    XRM  R14,R24         ;MODIFY INDEX
065314 140 012 343                    POMD R40,-R12
065317 014 267 062                    STMD R40,X14,TMP1
065322 211
065323
065323                  NEXT          BSZ  0
065323 230                            BIN
065324 120 312 010                    ADB  R20,=10         ;INC IDX TO NXT SIDE
065327 310 031                        CMB  R20,=31         ;LAST SIDE
065331 373 011                        JCY  DONE            ;JIF YES
065333 310 020                        CMB  R20,=20         ;SWITCH TO Y-BOUND
065335 766 254                        JNZ  S2              ;JIF NO
065337 122 312 010                    ADB  R22,=10         ;ELSE IDX TO Y-COMP
065342 760 247                        JMP  S2              ;GO BACK TO LOOP
065344
065344 235              DONE          CLE
065345 236                            RTN
065346
```

IN CLIP AREA CHECK

```
065346
065346
065346           ******************************************************
065346           *                                                    *
065346           *     CHECK IF THE COMPONENT IS INSIDE THE CLIP BOUNDARY*
065346           *                                                    *
065346           *       ENTRY:                                       *
065346           *         R20-21=INDEX TO ONE BOUND                  *
065346           *         R24-25=INDEX TO THE COMPONENT              *
065346           *                                                    *
065346           *       RETURN:                                      *
065346           *         E=0 IF OUT OF BOUND                        *
065346           *         E=1 IF IN BOUND                            *
065346           *                                                    *
065346           ******************************************************
065346                   IN?     BSZ  0
065346 316 377 152               JSB  =GETBND        ;FETCH BOUND
065351 140 024 265               LDMD R40,X24,TMP1 ;FETCH COMPONENT
065354 062 211
065356 316 375 144               JSB  =COMFLT        ;COMPARE
065361 120 220                   TSB  R20            ;IS IT AN UPPER BOUND
065363 377 011                   JRN  INRTN          ;JIF YES
065365 371 006                   JEZ  IN+            ;ELSE COMPLEMENT FLAG
065367 140 221                   TSM  R40            ;EXCEPT EQUAL CASE
065371 367 003                   JZR  INRTN
065373 235                       CLE
065374 236                       RTN
065375 234               IN+     ICE
065376 236               INRTN   RTN
065377
```

```
>>>>>>        LIMIT SELECTION                                              <<<<<<<
065377
065377                  *************************************************************
065377                  *                                                           *
065377                  *     GET A SIDE ACCORDING TO CURRENT UNIT                  *
065377                  *                                                           *
065377                  *       ENTRY:                                              *
065377                  *         R20-21=INDEX OF THE SIDE                          *
065377                  *                                                           *
065377                  *       RETURN:                                             *
065377                  *         R50-57=THE SIDE                                   *
065377                  *                                                           *
065377                  *************************************************************
065377
065377                  GETBND  BSZ  0
065377 230                      BIN
065400 130 261 127              LDMD R30,=USER    GET IDX TO LIMIT
065403 212
065404 020 303                  ADM  R30,R20      ADD IDX TO SIDE
065406 150 030 265              LDMD R50,X30,XH1  FETCH BOUND
065411 140 211
065413 236                      RTN
065414
```

```
                        CALCULATE INTERCEPT POINT
1965 065414
1966 065414             *************************************************
1967 065414             *                                               *
1968 065414             *     CALCULATE INTERCEPT POINT                 *
1969 065414             *                                               *
1970 065414             *       INTERCEPTION POINT ON XBOUND:           *
1971 065414             *                                               *
1972 065414             *         X'=XBOUND                             *
1973 065414             *         Y'=(XBOUND-XA)/(XB-XA)*(YB-YA)+YA     *
1974 065414             *                                               *
1975 065414             *       INTERCEPTION POINT ON YBOUND            *
1976 065414             *                                               *
1977 065414             *         X'=(YBOUND-YA)/(YB-YA)*(XB-XA)+XA     *
1978 065414             *         Y'=YBOUND                             *
1979 065414             *                                               *
1980 065414             *       REMARK: NO NEED TO CHECK FOR DIVIDED BY 0*
1981 065414             *               FOR IT IS IMPOSSIBLE IN OUR CASE.*
1982 065414             *                                               *
1983 065414             *     STORAGE ALLOCATION:                       *
1984 065414             *                                               *
1985 065414             *       TMP1=XA             ;NEW PEN LOCATION   *
1986 065414             *       TMP2=YA                                 *
1987 065414             *       TMP3=XB             ;LAST PEN LOCATION  *
1988 065414             *       TMP4=YB                                 *
1989 065414             *                                               *
1990 065414             *     REGISTER USAGE:                           *
1991 065414             *                                               *
1992 065414             *       R22-23=0 : INDEX TO X-COMPONENT         *
1993 065414             *                10: INDEX TO Y-COMPONENT       *
1994 065414             *                                               *
1995 065414             *       R20-21=INDEX TO THE BOUNDARY            *
1996 065414             *                                               *
1997 065414             *       R24-25=POINTER TO YA (XA)               *
1998 065414             *                                               *
1999 065414             *       R26-27=POINTER TO XB (YB)               *
2000 065414             *                                               *
2001 065414             *     RETURN:                                   *
2001 065414             *       RESULT ON R12 STACK.                    *
2002 065414             *                                               *
2003 065414             *************************************************
2003 065414
2004 065414             INTRC   BSZ  0
2005 065414 230                 BIN
2005 065415 124 251 072         LDM  R24,=TMP2      ;SET UP POINTERS
2006 065420 211
2006 065421 022 305             SBM  R24,R22        POINT TO YA (XA)
2007 065423 126 251 062         LDM  R26,=TMP1
2007 065426 211
2008 065427 303                 ADM  R26,R22        POINT TO XA (YA)
2009 065430 231                 BCD
2010 065431 140 024 341         POMD R40,+R24       ;POP YA (XA)
2011 065434 012 345             PUMD R40,+R12       ;SAVE A COPY
2012 065436 150 024 341         POMD R50,+R24       ;SKIP XB (YA)
```

```
              >>>>         CALCULATE INTERCEPT POINT                                   <<<<
065441 341                          POMD R50,+R24      ;POP YB (XB)
065442 316 377 377                  JSB  =SUB10        ;YB-YA (XB-XA)
065445 316 377 152                  JSB  =GETBND       ;GET THE BOUND
065450 006 345                      PUMD R#,+R6        ;SAVE A COPY ON STACK
065452 140 026 245                  LDMD R40,R26       ;GET XA (YA)
065455 316 377 377                  JSB  =SUB10        ;XBOUND-XA (YBOUND-YA)
065460 316 377 377                  JSB  =MPYROI       ;(XBOUND-XA)*(YB-YA), OR
065463                   *                             ;(YBOUND-YA)*(XB-XA)
065463 026 341                      POMD R#,+R26       ;POP XA (YA)
065465 150 341                      POMD R50,+R26      ;SKIP YA (XB)
065467 341                          POMD R50,+R26      ;POP XB (YB)
065470 316 377 377                  JSB  =SUB10        ;XB-XA (YB-YA)
065473 316 377 377                  JSB  =DIV2         ;(PREVIOUS)/(XB-XA), OR
065476                   *                             ;(PREVIOUSA)/(YB-YA)
065476 316 377 377                  JSB  =ADDROI       ;RESULT+YA (XA)
065501 006 343                      POMD R#,-R6        ;POP THE BOUND
065503 012 345                      PUMD R#,+R12       ;RETURN IT ON STACK
065505 236                          RTN
065506
```

```
065506
065506                 ****************************************************************
065506                 *                                                              *
065506                 *    PREPARATION ROUTINE FOR DRAW LINE FROM B TO A             *
065506                 *                                                              *
065506                 *      REGISTER USAGE:                                         *
065506                 *                                                              *
065506                 *        R20-21=XB                                             *
065506                 *        R22-23=YB                                             *
065506                 *        R34-35=END CHECK POINT                                *
065506                 *        R46-47=XINCREMENT                                     *
065506                 *        R56-57=YINCREMENT                                     *
065506                 *        R36-37=D                                              *
065506                 *        R52-53=DX'                                            *
065506                 *        R54-55=DY'                                            *
065506                 *        R50=SLOPE FLAG=0 IF SLOPE < 45, ELSE =1               *
065506                 *                                                              *
065506                 ****************************************************************
065506
065506                 *  REGISTER USAGE FOR MODIFIED ALGORITHM TO SPEED UP
065506                 *        R20-21=XB  THEN  CONTROL VARIABLE
065506                 *        R22-23=YB  THEN  DISCARDED
065506                 *        R34-35=XA  THEN  DISCARDED
065506                 *        R36-37=YA  THEN  D
065506                 *        R45-47=X-INCREMENT
065506                 *        R55-57=Y-INCREMENT
065506                 *        R75-77=CURRENT DOT ADDR
065506                 *        R51-52=DX'=ABS(XA-XB)
065506                 *        R53-54=DY'=ABS(YA-YB)
065506                 *        R65-67=WORKING REGISTERS
065506                 *
065506                 *        FORMAT OF 3-BYTE ADDRESS
065506                 *
065506                 *          7        0 7        0 765     0
065506                 *          ________________________________
065506                 *         I        I         I   00000I
065506                 *         I________I_________I_________I
065506                 *             BYTE ADDRESS   ; BIT ADDRESS
065506
065506                 DRAWC    BSZ  0
065506 316 044 154              JSB  =MAPCR       MAP TO CRT LOGIC ADDR
065511                          ARP  !12
065511 122 343                  POMD R22,-R12     YB
065513 120 343                  POMD R20,-R12     XB
065515 136 343                  POMD R36,-R12     YA
065517 134 343                  POMD R34,-R12     XA
065521                 *        CLM  R46
065521                 *        STB  R46,R50      INIT SLOPE<45
065521                 *        ICB  R46          XINC=1
065521                 *        STM  R46,R56      YINC=1
065521
065521 145 223                  CLM  R45          INIT X INCREMENT =
065523 250 040                  LDB  R45,=40      ONE DOT
```

```
 >>>>>>      DRAW LINE SET UP                                                <<<<<<<
2083 065525 155 223                    CLM   R55           INIT Y INCREMENT =
     065527 156 261 340                LDMD  R56,=GLINE    CHAR/LINE
2084 065532 211
2085 065533
2086 065533              *             LDM   R32,R36       COPY YA
2087 065533              *             SBM   R32,R22       DY=YA-YB
2088 065533              *             JPS   YOK
2089 065533              *             TCM   R32           ABSOLUTE DY
2090 065533              *             TCM   R56           YINC=-1
2091 065533
2092 065533 136 022 305                SBM   R36,R22       ***REPLACEMENT***
2093 065536 765 003                    JPS   YOK           ***REPLACEMENT***
2094 065540 215                        TCM   R36           ***REPLACEMENT***
2095 065541 155 215                    TCM   R55           ***REPLACEMENT***
2096 065543
2097 065543              YOK           BSZ   0
2098 065543              *             LDM   R30,R34       COPY XA
2099 065543              *             SBM   R30,R20       DX=XA-XB
2100 065543              *             JPS   XOK
2101 065543              *             TCM   R30           ABSOLUTE DX
2102 065543              *             TCM   R45           XINC=-1
2103 065543
2104 065543 134 020 305                SBM   R34,R20       ***REPLACEMENT***
2105 065546 765 003                    JPS   XOK           ***REPLACEMENT***
2106 065550 215                        TCM   R34           ***REPLACEMENT***
2107 065551 145 215                    TCM   R45           ***REPLACEMENT***
2108 065553
2'   065553              XOK           BSZ   0
2.   065553              *             CMM   R30,R32       COMPARE DX WITH DY
2111 065553              *             JCY   S<45          JIF SLOPE<45
2112 065553              *             BSZ   0
2113 065553              *             ICB   R50           FLAG SLOPE>45
2114 065553              *             LDM   R34,R36       END CHECK=YA
2115 065553              *             LDM   R36,R46       SWAP INCREMENTS
2116 065553              *             STM   R56,R46
2117 065553              *             LDM   R56,R36
2118 065553              *             STM   R32,R52       SWAP DX, DY
2119 065553              *             STM   R30,R54
2120 065553              *             LDM   R36,R22       SWAP XB, YB
2121 065553              *             STM   R20,R22
2122 065553              *             LDM   R20,R36
2123 065553
2124 065553 316 113 154                JSB   =MAPABS       MAP STARTING POINT
2125 065556 134 036 301                CMM   R34,R36       COMP DX' WITH DY'
2126 065561 373 021                    JCY   S<45          JIF SLOPE <= 45
2127 065563 165 045 241                LDM   R65,R45       SWAP INCREMENTS
2128 065566 155 243                    STM   R55,R45
2129 065570 065 241                    LDM   R55,R65
2130 065572 134 053 243                STM   R34,R53       SWAP DX', DY'
2131 065575 136 051 243                STM   R36,R51
2132 065600 020 243                    STM   R36,R20       SET CONTROL VARIABLE
2133 065602
2134 065602 360 010                    JMP   READY         READY TO GO
```

```
>>>>>         DRAW LINE SET UP                                                   <<<<<<
  065604                    SK45     BSZ   0
  065604                    *        STM   R30,R52        SETUP DX',DY'
  065604                    *        STM   R32,R54
  065604
  065604 134 051 243                 STM   R34,R51        SET DX
  065607 020 243                     STM   R34,R20        SET CONTROL VARIABLE
  065611 136 053 243                 STM   R36,R53        SET DY
  065614
  065614                    READY    BSZ   0
  065614                    *        LDM   R36,R52        INIT D=DX'/2
  065614
  065614                             DRP   !36
  065614 051 241                     LDM   R36,R51        INIT D=DX'/2
  065616
  065616 137 207                     LRM   R37
  065620 372 002                     JNC   READY2         JIF EVEN
  065622 136 211                     ICM   R36            ELSE INCR BY 1
  065624 316 230 153 READY2          JSB   =DRAWGO        DRAW THE LINE
  065627 236                         RTN
  065630
```

```
>>>>>          DRAW LINE SET UP                                          <<<<<<<<
     065604                  S<45     BSZ  0
     065604                  *        STM  R30,R52         SETUP DX',DY'
     065604                  *        STM  R32,R54
     065604
     065604 134 051 243               STM  R34,R51         SET DX
     065607 020 243                   STM  R34,R20         SET CONTROL VARIABLE
     065611 136 053 243               STM  R36,R53         SET DY
     065614
     065614                  READY    BSZ  0
     065614                  *        LDM  R36,R52         INIT D=DX'/2
     065614
     065614                           DRP  !36
     065614 051 241                   LDM  R36,R51         INIT D=DX'/2
     065616
     065616 137 207                   LRM  R37
     065620 372 002                   JNC  READY2          JIF EVEN
     065622 136 211                   ICM  R36             ELSE INCR BY 1
     065624 316 230 153 READY2        JSB  =DRAWGO         DRAW THE LINE
     065627 236                       RTN
     065630
```

```
                        DRAW LINE WITH LINE TYPE
2156 065630
2157 065630             ********************************************************
2158 065630             *                                                      *
2159 065630             *     DRAW LINE WITH LINE TYPE                         *
2160 065630             *                                                      *
2161 065630             *          R76-77=GRAPHICS BYTE ADDRESS                *
2162 065630             *          R75=GRAPHICS BIT ADDRESS                    *
2163 065630             *          R30-33=WORKING REGISTERS                    *
2164 065630             *                                                      *
2164 065630             *     THE LINE IS DRAWN FROM POINT B TO A.             *
2164 065630             *                                                      *
2165 065630             ********************************************************
2166 065630             DRAWGO  BSZ   0
2167 065630 170 260 013         LDBD  R70,=DSKPF    SKIP FLAG
2167 065633 212
2168 065634 110 223             CLM   R10
2169 065636 260 011 212         LDBD  R10,=DSEG#    SEGMENT #
2170 065641 171 260 012         LDBD  R71,=DTCNT    DOT COUNT OF THE SEG
2171 065644 212
2172 065645             DRAWG2  BSS   0
2172 065645                     ORP   !71
2173 065645 212                 DCB   R71           DOT REMAINED IN SEG
2174 065646 373 026             JCY   GODRAW        JMP TO DRW THE REMAINING DOTS
2175 065650 170 216             NCB   R70           FLIP SKIP FLAG
2176 065652 110 210             ICB   R10           INC TO NEXT SEGMENT
2177 065654 320 370 211         CMBD  R10,=D#SEG    IS IT LAST SEGMENT
2178 065657 372 006             JNC   SETUP         JIF NOT
2179 065661 367 004             JZR   SETUP
2180 065663 122                 CLB   R10           ELSE RESET SKIP FLAG
2181 065664 970 242             STB   R10,R70
2182 065666 210                 ICB   R10           RESET SEG # TO 1
2183 065667             SETUP   BSZ   0
2184 065667 171 010 264         LDBD  R71,X10,D#SEG GET DOT COUNT IN SEG
2184 065672 370 211
2185 065674 360 347             JMP   DRAWG2        LOOP BACK
2186 065676
2187 065676             GODRAW  BSZ   0
2188 065676 170 220             TSB   R70           CHECK SKIP FLAG
2189 065700 364 003             JNG   ENDRW?        JMP TO SKIP PT
2190 065702
2191 065702             *        TSB   R50           CHECK SLOPE
2192 065702             *        JZR   D0<45         JIF < 45
2193 065702             *        LDM   R30,R22       GET X-COOR
2194 065702             *        LDM   R32,R20       GET Y-COOR
2195 065702             *        JMP   DRAW+
2196 065702             +D0<45   BSZ   0
2197 065702             *        LDM   R30,R20       GET X-COOR
2198 065702             *        LDM   R32,R22       GET Y-COOR
2199 065702             +DRAW+   BSZ   0
2200 065702             *        JSB   =MAPABS       MAP TO ABSOLUTE ADDR
2201 065702
2202 065702 316 351 153         JSB   =DRAWPT       DRAW A POINT
```

```
2202         DRAW LINE WITH LINE TYPE
2203 065705                    ENDRW2  BSZ  0
2204 065705                    *       CMM  R20,R34        END OF DRAW LINE?
2205 065705
2206 065705 120 213                    DCM  R20            ***REPLACEMENT***
2207 065707
2208 065707 372 023                    JNC  DREXIT         JIF YES
2209 065711                    *       SBM  R36,R54        D=D-DY'
2210 065711
2211 065711 136 053 305                SBM  R36,R53        ***REPLACEMENT***
2212 065714
2213 065714 367 002                    JZR  INCY           JIF D=0
2214 065716 365 006                    JPS  INCX           JIF D>0
2215 065720                    INCY    BSZ  0              FOR D<=0
2216 065720                    *       ADM  R36,R52        D=D+DX'
2217 065720                    *       ADM  R22,R56        Y=Y+YINC
2218 065720
2219 065720 136 051 303                ADM  R36,R51        ***REPLACEMENT***
2220 065723 175 055 303                ADM  R75,R55        ***REPLACEMENT***
2221 065726
2222 065726                    INCX    BSZ  0
2223 065726                    *       ADM  R20,R46        X=X+XINC
2224 065726
2225 065726 175 045 303                ADM  R75,R45        ***REPLACEMENT***
2226 065731
2227 065731 171                        DRP  71
2228 065732 360 311                    JMP  DRAWG2         LOOP BACK
2229 065734                    DREXIT  BSZ  0
2230 065734 110 262 011                STBD R10,=DSEG#     SAVE SEG #
2230 065737 212
2231 065740 171 262 012                STBD R71,=DTCNT     SAVE REMAIN DOT COUNT
2231 065743 212
2232 065744 170 262 013                STBD R70,=DSKPF     SAVE SKIP FLAG
2232 065747 212
2233 065750 236                        RTN
2234 065751
```

```
065751
065751                       ****************************************************
065751                       *                                                  *
065751                       *      DRAW A POINT                                *
065751                       *                                                  *
065751                       *         ENTRY:                                   *
065751                       *           R76-77=BYTE ADDR                       *
065751                       *           R75=BIT ADDR                           *
065751                       *                                                  *
065751                       ****************************************************
065751                       DRAWPT  BSZ   0
065751 316 032 154                   JSB   =BYTEG         SET GRAPHICS BYTE ADDR
065754 316 377 377                   JSB   =INCHR         FETCH BYTE
065757 133 250 200                   LDB   R33,=200       SET BIT
065762                       *       TSB   R75            CHECK BIT POSITION
065762
065762 131 075 240                   LDB   R31,R75        ***REPLACEMENT***
065765
065765 367 007                       JZR   GETPSY
065767                       SHFTIT  BSZ   0
065767 133 206                       LRB   R33            SHIFT TO BIT POSI
065771                       *       SBB   R75,=40        DEC BIT COUNT
065771
065771 131 314 040                   SBB   R31,=40        ***REPLACEMENT***
065774
065774 366 371                       JNZ   SHFTIT
065776                       GETPSY  BSZ   0
065776 316 032 154                   JSB   =BYTEG         ADDRESS CRT
066001 130 261 135                   LDMD  R30,=PEN#      GET PEN #
066004 211
066005 132                           DRP   32
066006 030 306 024                   JSB   X30,PEN00      DO IT
066011 154
066012 316 037 154                   JSB   =OUTBYT        OUT TO CRT
066015 236                           RTN
066016
066016                       *
066016                       *    PEN# = PEN SELECTION CODE * 3
066016                       *
066016
066016                       PEN-2   BSS   0              FOR PEN -2
066016                               DRP   !32
066016 033 226                       XRB   R32,R33        XOR THE BIT
066020 236                           RTN
066021
066021                       PEN-1   BSS   0              FOR PEN -1
066021 133 216                       NCB   R33
066023 132                           DRP   32
066024                       PEN00   BSS   0              PSEUDO ENTRY
066024                               DRP   !32            FOR INDEXING
066024 033 307                       ANM   R32,R33        AND THE BIT
066026 236                           RTN
066027
```

```
                DRAW A POINT
     066027              PENI     CSS  0
22   066027                       SRP  !32
2286 066027 F33 224               ORB  R32,R33       OR THE BIT
2287 066031 236                   RTN
2288 066032
```

```
              ADDRESS CRT AND OUT A BYTE                                [illegible]
066032
066032                       ***********************************************
066032                       *
066032                       *    ADDRESS CRT AND OUTPUT A BYTE TO CRT     *
066032                       *
066032                       *      ENTRY:                                 *
066032                       *        R76-77=BYTE ADDR OF GRAPHICS         *
066032                       *        R32=DATA BYTE                        *
066032                       *                                             *
066032                       ***********************************************
066032
066032                       BYTEG    BSZ  0
066032                       *        JSB  =CHKSTS       WAIT TILL CRT READY
066032 176 263 301                    STMD R76,=CRTBAD   TELL CRT
066035 377
066036 236                            RTN
066037
066037
066037                       OUTBYT   BSZ  0
066037                       *        JSB  =CHKSTS       WAIT TILL CRT READY
066037 132 262 303                    STBD R32,=CRTDAT   OUTPUT A BYTE
066042 377
066043 236                            RTN
066044
066044
```

MAP TO CRT DOT ADDRESS

```
066044
066044                        ********************************************************************
066044                        *
066044                        *     MAP POINT LOCATION FROM MU TO CRT DOT ADDR                    *
066044                        *
066044                        ********************************************************************
066044
066044                        MAPCR   BSZ  0
066044 126 251 062                    LDM  R26,=TMP1     GET ADDR OF POINTS
066047 211
066050 316 053 154                    JSB  =MAPCR1       MAP THEM
066053
066053                        MAPCR1  BSZ  0
066053 160 026 341                    POMD R60,+R26      POP ONE COMP
066056 316 377 377                    JSB  =CONINT       CONVERT TO INTEGER
066061 176 012 345                    PUMD R76,+R12      SAVE ON STACK
066064 160 026 341                    POMD R60,+R26      POP Y COMPONENT
066067 316 377 377                    JSB  =CONINT       CONVERT TO INTEGER
066072 230                            BIN
066073 132 220                        TSB  R32           NEGATIVE ?
066075 377 006                        JRN  MAP1.5        JIF YES
066077 176 315 357                    SBM  R76,=G#LN-1   CONV TO LINE ADDR
066102 100
066103 372 002                        JNC  MAPCR2        JIF IN THE WINDOW
066105                        MAP1.5  BSS   0
066105 176 223                        CLM  R76           ELSE DEFAULT TO 0
066107                        MAPCR2  BSS  0
066107                                DRP  !76
066107 215                            TCM  R76
066110 012 345                        PUMD R76,+R12       SAVE ON STACK
066112 236                            RTN
066113
```

```
 066113
 066113
 066113                 ****************************************************************
 066113                 *                                                              *
 066113                 *     MAP CRT DOT ADDRESS TO ABSOLUTE ADDR                     *
 066113                 *                                                              *
 066113                 *       ENTRY:                                                 *
 066113                 *         R30-31=X-COOR OF THE DOT                             *
 066113                 *         R32-33=Y-COOR OF THE DOT                             *
 066113                 *                                                              *
 066113                 *         NOW USE R20-21 INSTEAD OF R30-R31, AND               *
 066113                 *                 R22-23 INSTEAD OF R32-33.                    *
 066113                 *                                                              *
 066113                 *                                                              *
 066113                 *       RETURN:                                                *
 066113                 *         R76-77=BYTE ADDR OF THE DOT                          *
 066113                 *         R75=BIT ADDR OF THE DOT                              *
 066113                 *                                                              *
 066113                 ****************************************************************
 066113
 066113                 MAPABS  BSZ  0
 066113 230                     BIN
 066114 175 223                 CLM  R75
 066116 250 006                 LDB  R75,=6         BIT COUNT
 066120 177 260 340             LDBD R77,=GLINE     # OF BYTES / LINE
 066123 211
 066124 204                     LLB  R77            THROW AWAY LEADING 0
 066125 022                     ARP  22             INITIALIZE ARP
 066126                 MULPY   BSS  0              Y-LINE * #-BYTE / LINE
 066126 176 205                 LLM  R76            SHIFT ONE BIT
 066130 372 001                 JNC  MULPY1         JIF IT'S 0
 066132 303                     ADM  R76,R#         ADD MULTIPLICAND
 066133                 MULPY1  BSS  0
 066133 175 212                 DCB  R75            DECREMENT COUNT
 066135 366 367                 JNZ  MULPY          JIF NOT DONE
 066137 176 205                 LLM  R76            RESULT * 2
 066141 121 207                 LRM  R21            DIVIDE X BY 8
 066143 175 202                 ERB  R75
 066145 121 207                 LRM  R21
 066147 175 202                 ERB  R75
 066151 121 207                 LRM  R21            R20=INT(X/8)
 066153 175 202                 ERB  R75            R75=BIT ADDR=RMD(X/8)
 066155 176 020 303             ADM  R76,R20        ADD TO GIVE BYTE ADDR
 066160 313 340 020             ADM  R76,=GBASE     +BASE=BYTE ADDR
 066163 236                     RTN
 066164
```

```
                    FRAME
066164
066164
066164                          ****************************************************************
066164                          *
066164                          *     FRAME RUN TIME ROUTINE                                   *
066164                          *                                                              *
066164                          ****************************************************************
066164 241                               OCT  241              PRIMARY ATTRIBUTE
066165
066165                          FRAME.   BSZ  0
066165 230                               BIN
066166 124 223                           CLM  R24              RESET INDEX
066170 012 344                           PUBD R24,+R12         SAVE FLAG
066172 323 127 212                       ADMD R24,=USER        COUNT IN MODE
066175 345                               PUMD R24,+R12         SAVE INDEX TO Y-COMP
066176 345                               PUMD R24,+R12         SAVE INDEX TO X-COMP
066177 313 140 211                       ADM  R24,=XH1         POINT TO BOUNDARY
066202 126 251 062                       LDM  R26,=TMP1        SETUP FOR MOVE
066205 211
066206 316 354 144                       JSB  =MOVXY
066211 316 130 167                       JSB  =MOVENU          MOVE PEN TO LOWER LEFT
066214 124 223                           CLM  R24              SET PEN CONTROL = -1
066216 213                               DCM  R24              WHICH IS DROP PEN FIRST
066217 263 131 211                       STMD R24,=PCNTRL
066222 020 241                           LDM  R24,R20          INIT INDEX TO X-COMP
066224 126 241                           LDM  R26,R20          INDEX TO Y-COMP
066226 100 222                           CLB  R0               RESET FLIP FLAG
066230 316 244 154                       JSB  =FRASUB          GO DRAW FRAME
066233 124 012 343                       POMD R24,-R12         CLEAN STACK
066236 343                               POMD R24,-R12
066237 342                               POBD R24,-R12
066240 316 357 150                       JSB  =PENUP.          LIFT PEN UP
066243 236                               RTN
066244
066244                          *
066244                          FRASUB   BSZ  0
066244 316 247 154                       JSB  =FRASU1
066247                          FRASU1   BSZ  0
066247 316 252 154                       JSB  =FRASU2
066252                          FRASU2   BSZ  0
066252 120 251 010                       LDM  R20,=10,0
066255 000
066256 124 012 343                       POMD R24,-R12         POP X INDEX
066261 126 343                           POMD R26,-R12         POP Y INDEX
066263 100 342                           POBD R0,-R12          POP FLAG
066265 364 005                           JNG  FRASU3
066267 126 020 227                       XRM  R26,R20          FLIP Y INDEX
066272 360 003                           JMP  FRASU4
066274                          FRASU3   BSZ  0
066274 124 020 227                       XRM  R24,R20          FLIP X INDEX
066277                          FRASU4   BSZ  0
066277 100 216                           NCB  R0               FLIP FLAG
066301 012 344                           PUBD R0,+R12          SAVE FOR NEXT
```

```
            FRAME
  29 066303 126 245            PUMD R26,+R12
   ? 066305 124 245            PUMD R24,+R12
2431 066307 140 024 265        LDMD R40,X24,XH1  FETCH X COMPONENT
2431 066312 140 211
2432 066314 263 062 211        STMD R40,=TMP1    STORE IN PLACE
2433 066317 026 265 160        LDMD R40,X26,YH1  FETCH Y COMPONENT
2433 066322 211
2434 066323 263 072 211        STMD R40,=TMP2    STORE IN PLACE
2435 066326 316 165 151        JSB  =UPDPEN      UPDATE PEN LOCATION
2436 066331 316 026 152        JSB  =DRAW.6      DRAW ONE SIDE
2437 066334 236                RTN
2438 066335
```

LINE TYPE

```
066335
066335              ****************************************************
066335              *
066335              *      LINE TYPE RUN TIME ROUTINE                   *
066335              *                                                   *
066335              *                                                   *
066335              ****************************************************
066335
066335 241                      OCT   241          PRIMARY ATTRIBUTE
066336
066336                  LINET.  BSZ   0
066336 130 251 006              LDM   R30,=6,0
066341 000
066342 316 125 145              JSB   =UPLTDR
066345 236                      RTN
066346
066346                  C6      BSS   0
066346 316 377 377              JSB   =STACKX      HOW MANY PARS ON STACK ?
066351 374 014                  JLZ   LINET1       JIF ONE PAR
066353 316 377 377              JSB   =ONER        FETCH 1ST PAR
066356 316 031 156              JSB   =PARGK?      LEGAL PAR?
066361 316 015 156              JSB   =GUTOMU      CONVRT TO MU
066364 243                      POMD  R#,-R#       POP OFF STACK
066365 006 345                  PUMD  R#,+R6       SAVE ON SYS STK
066367                  LINET1  BSZ   0
066367 316 377 377              JSB   =ONEX        GET ABS OF TYPE#
066372                          DRP   !76
066372 066 243                  STM   R76,R66      SAVE IT
066374 251 011 000              LDM   R76,=9D,0    GET A 9
066377 166                      DRP   66           FOR MOD
066400 316 377 377              JSB   =MOD         MOD 9
066403                          DRP   !66
066403 366 004                  JNZ   TYPEOK
066405                  DEFTYP  BSZ   0            DEFAULT TO TYPE 1
066405 166 251 001              LDM   R66,=1,0
066410 000
066411                  TYPEOK  BSZ   0
066411                          DRP   !66
066411 262 137 211              STBD  R66,=LTYPE#  SAVE LINE TYPE
066414 212                      DCB   R66          DECR TYPE CODE
066415 204                      LLB   R66          (CODE-1)*2
066416 124 066 265              LDMD  R24,X66,LNTYPE GET LINE TYPE DEF
066421 350 211
066423 022 243                  STM   R24,R22      GET A COPY
066425 126 220                  TSB   R26          HOW MANY PARS?
066427 374 046                  JLZ   LINET3       JIF ONE PAR
066431 127 340                  POBD  R27,+R22     GET # OF SEGMENTS
066433 140 223                  CLM   R40          RESET ACCUMULATOR
066435 012 345                  PUMD  R40,+R12     PUSH ON STACK
066437                  LINET2  BSZ   0
066437 136 223                  CLM   R36          RESET BUFFER
066441 022 340                  POBD  R36,+R22     GT DOT CNT OF SEG
066443 316 377 377              JSB   =CONBIN      CONVRT TO FLOAT PT
```

LINE TYPE

```
066446 140 012 345              PUMD R40,+R12       SAVE ON STK
066451 231                      BCD
066452 316 377 377              JSB  =ADDR01        SUM IT
066455 230                      BIN
066456 127 212                  DCB  R27            DEC SEG COUNT
066460 766 355                  JNZ  LINET2         LOOP BACK
066462 150 006 343              POMD R50,-R6        GET PATTERN LENGTH
066465 140 012 343              POMD R40,-R12       GET TOTAL DOTS/PATRN
066470 316 377 377              JSB  =DIV10         GET RATIO
066473 343                      POMD R#,-R#         POP OFF STK
066474 263 014 212              STMD R#,=PRATIO     SAVE THE RATIO
066477                 LINET3   BSS  0
066477 122 251 370              LDM  R22,=D#SEG     GET DEST ADDR
066502 211
066503 127 024 340              POBD R27,+R24       GET # OF SEGMENTS
066506 022 344                  PUBD R27,+R22       STORE IN PLACE
066510                 LINET5   BSZ  0
066510 136 223                  CLM  R36
066512 024 340                  POBD R36,+R24       POP A SEGMENT
066514 126 220                  TSB  R26            HOW MANY PAR ?
066516 136                      ARP  36
066517 374 021                  JLZ  LINET6         JIF ONE
066521 316 377 377              JSB  =CONBIN        CONVRT TO REAL
066524 150 261 014              LDMD R50,=PRATIO    GET RATIO
066527 212
066530 231                      BCD
066531 316 377 377              JSB  =MPY10         ADJUST DOT COUNT
066534 316 377 377              JSB  =CEIL10
066537 316 377 377              JSB  =ONEB          CONVRT TO BINARY
066542                 LINET6   BSS  0
066542 022 344                  PUBD R#,+R22        STORE THE SEGMENT
066544 230                      BIN
066545 127 212                  DCB  R27            DEC SEG COUNT
066547 766 337                  JNZ  LINET5         DO NEXT SEGMENT
066551 262 013 212              STBD R#,=DSKPF      RESET SKIP FLAG
066554 210                      ICB  R#             INIT SEG = 1
066555 262 011 212              STBD R#,=DSEG#
066560 260 371 211              LDBD R#,=DSEG       DOT COUNT IN SEG #1
066563 262 012 212              STBD R#,=DTCNT
066566 236                      RTN                 DONE
066567
066567
```

LOC OBJECT CODE

CSIZE

```
066567
066567                     ************************************************************
066567                     *
066567                     *     CSIZE RUN TIME ROUTINE                               *
066567                     *                                                          *
066567                     ************************************************************
066567
066567 241                        OCT   241
066570
066570                     CSIZE.  BSS   0
066570 122                        CLB   R#             FLAG DEFAULT SLANT
066571 262 122 211                STBD  R#,=DEFSLA
066574 316 377 377                JSB   =STACK*        HOW MANY PARS ?
066577 374 024                    JLZ   CSIZE2         JIF ONE PAR
066601 376 015                    JRZ   CSIZE1         JIF TWO PARS
066603 316 377 377                JSB   =TAN10         CALC TAN(SLANT)
066606 140 012 243                POMD  R40,-R12       POP OFF STACK
066611 263 106 212                STMD  R#,=TANSL      STORE IN PLACE
066614 147 262 122                STBD  R47,=DEFSLA    FLAG NOT DEFAULT
066617 211
066620                     CSIZE1  BSS   0
066620 316 377 377                JSB   =ONER          GET ASPECT RATIO
066623 760 031                    JMP   CSIZE3         GO ON
066625                     CSIZE2  BSZ   0
066625 114 261 133                LDMD  R14,=PLOTR#
066630 211
066631 311 003 000                CMM   R14,=3,0       IS IT A PLOTR
066634 765 006                    JPS   LDPLTD         JIF YES
066636 140 261 005                LDMD  R40,=R2/3      DEFAULT ASP RATIO=2/3
066641 156
066642 760 012                    JMP   CSIZE3
066644 140 251 231         LDPLTD  LDM   R40,=990,900,0,0,0,0,0,600     FOR PLOTTER
066647 220 000 000
066652 000 000 000
066655 140
066656                     ***************************  DEFAULT ASP RATIO=6/10
066656                     CSIZE3  BSS   0
066656                            ORP   140
066656 070 243                    STM   R40,R70        SAVE A COPY
066660 100 250 001                LDB   R0,=1          SET NON DEFAULT CSIZE
066663 316 377 377                JSB   =ONER          GET HIGHT FACTOR
066666 114 311 003                CMM   R14,=3,0       IS IT A PLOTR
066671 000
066672 765 026                    JPS   CSIZE4         JIF YES
066674 140 311 000                CMM   R40,=0,0,0,0,0,0,0,500
066677 000 000 000
066702 000 000 000
066705 120
066706                     ***************************  DEFAULT HEIGHT ?
066706 766 012                    JNZ   CSIZE4         JIF NOT
066710 170 321 005                CMMD  R70,=R2/3      DEFAULT ASPECT RATIO
066713 156
066714 766 004                    JNZ   CSIZE4         JIF NOT
```

```
[illegible]           CSIZE
[illegible] 066716 100 260 002                 LDBD R0,=NONSTD   FLAG DEFAULT CSIZE
[illegible] 066721 207
2568 066722                    CSIZE4   BSS  0
2569 066722 100 262 123                 STBD R0,=DEFSZE
2569 066725 211
2570 066726 170 012 345                 PUMD R70,+R12      SAVE ASP RATIO
2571 066731 316 015 156                 JSB  =GUTOMU       CONVRT HIGHT TO MU
2572 066734 250 243                     STM  R#,R50        SAVE A COPY
2573 066736 223                         CLM  R#            REG HITE=12
2574 066737 210                         ICB  R#
2575 066740 147 250 022                 LDB  R47,=12C
2576 066743 316 377 377                 JSB  =DIV10        GET HITE RATIO
2577 066746 343                         POMD R#,-R#        POP OFF STK
2578 066747 263 036 212                 STMD R#,=CHITE     SAVE IN PLACE
2579 066752 316 377 377                 JSB  =MPYROI       WIDTH=HIGHT*ASP RATIO
2580 066755                             DRP  !40
2581 066755 223                         CLM  R40           REG WIDTH=8
2582 066756 147 250 200                 LDB  R47,=80C
2583 066761 150 343                     POMD R50,-R#       GET SPEC WIDTH
2584 066763 316 377 377                 JSB  =DIV10        GET WIDTH RATIO
2585 066766 343                         POMD R#,-R#        POP OFF STK
2586 066767 263 026 212                 STMD R#,=CWIDE     SAVE IN PLACE
2587 066772 130 251 020                 LDM  R30,=20,0
2587 066775 000
2588 066776 316 125 145                 JSB  =UFLTDR
2589 067001 316 113 156                 JSB  =CDISP        SET DISP FOR CHAR
2590 067004 236               C20       RTN
2591 067005
2592 067005 231 220 147 R2/3            OCT  99C,90C,67C,66C,66C,66C,66C,66C
2592 067010 146 146 146
2592 067013 146 146
2593 067015
```

```
067015
067015
067015                 ****************************************************
067015                 *
067015                 *     GU TO  MU CONVERSION                           *
067015                 *                                                    *
067015                 *       ENTRY:                                       *
067015                 *         R40-47=DATA IN GU                          *
067015                 *                                                    *
067015                 *       RETURN:                                      *
067015                 *         DATA IN MU ON STACK                        *
067015                 *                                                    *
067015                 ****************************************************
067015
067015                 GUTOMU  BSZ  0
067015 150 261 242             LDMD R50,=XAGU      GET MU/GU
067020 210
067021 231                     BCD
067022 151 206                 LRB  R51            ABSOLUTE IT
067024 204                     LLB  R51
067025 316 377 377             JSB  =MPY10         CONVERT
067030 236                     RTN
067031
067031                 ****************************************************
067031                 *
067031                 *     LEGAL PARAMETER CHECK                          *
067031                 *                                                    *
067031                 *       ENTRY:                                       *
067031                 *         R40-47=PARAMETER                           *
067031                 *                                                    *
067031                 *       RETURN:                                      *
067031                 *         TO CALLING ROUTINE IF PAR > 0 .(OK)        *
067031                 *         TO SYSTEM IF PAR <= 0  (NO GOOD)           *
067031                 *                                                    *
067031                 ****************************************************
067031
067031                 PAROK?  BSZ  0
067031 140 221                 TSM  R40            TEST PAR
067033 367 004                 JZR  ERRRTN         JIF ZERO
067035 141 220                 TSB  R41            CHECK SIGN
067037 376 007                 JRZ  PAROK          JIF POSITIVE
067041                 ERRRTN  BSZ  0
067041 316 377 377             JSB  =ERROR
067044 131                     OCT  89D            ILLEGAL PAR
067045 146 006 343             POMD R46,-R6        POP RTN ADDR
067050 236             PAROK   RTN                 RTN TO SYSTEM
067051
```

```
     067051
     067051
2636 067051                    ********************************************************
2637 067051                    *                                                      *
2637 067051                    *     LDIR RUN TIME ROUTINE                            *
2637 067051                    *                                                      *
2638 067051                    ********************************************************
2639 067051 241                          OCT  241            PRIMARY ATTRIBUTE
2640 067052
2641 067052                    LDIR.     BSZ  0
2642 067052 114 251 046                  LDM  R14,=SINLD     SET DEST ADDR
2642 067055 212
2643 067056 316 062 172                  JSB  =PDIR1         CALC SIN, COS
2644 067061 130 251 022                  LDM  R30,=22,0      DO THE REST
2644 067064 100
2645 067065 316 125 145                  JSB  =UPLTDR
2646 067070 236                          RTN
2647 067071
2648 067071                    C22       BSS  0
2649 067071 316 113 156                  JSB  =CDISP         SET DISP FOR CHAR
2650 067074 316 042 147                  JSB  =R1.           GET A REAL NUM 1.0
2651 067077 321 056 212                  CMMD R#,=COSLD      IS IT 0 DEGREE ?
2652 067102 366 002                      JNZ  C30.1          JIF NOT
2653 067104 147 222                      CLB  R47            ELSE FLAG DEFAULT
2654 067106                    C30.1     BSS  0
2655 067106 147 262 124                  STBD R47,=DEFLDR    FLG DEFAULT LDIR
2655 067111 211
2656 067112 236                          RTN                 DONE
2657 067113
```

```
067113
067113                    *   HORIZONTALLY
067113                    *       X=8 , Y=0
067113                    *       X'=X * CUIDE * COSLD
067113                    *       Y'=X * CUIDE * SINLD
067113                    *
067113                    *   VERTICALLY
067113                    *       X=0 , Y=12
067113                    *       X'= - Y * CHITE * SINLD
067113                    *       Y'= Y * CHITE * COSLD
067113
067113                    CDISP   BSS   0
067113 231                        BCD
067114 140 251 000                LDM   R40,=0,0,0,0,0,0,0,80C
067117 000 000 000
067122 000 000 000
067125 200
067126                    ************************  DEF HORI DISP
067126 150 260 123                LDBD R50,=DEFSZE   DEFAULT CSIZE ?
067131 211
067132 367 007                    JZR   CDISP1       JIF YES
067134 261 026 212                LDMD R50,=CWIDE    ADJ BY WIDTH RATIO
067137 316 377 377                JSB   =MPY10
067142 343                        POMD R#,-R#        GET A COPY FROM STACK
067143                    CDISP1  BSS   0
067143 140 012 345                PUMD R40,+R12      SAVE A COPY
067146 150 261 056                LDMD R50,=COSLD    ADJ BY LDIR
067151 212
067152 316 377 377                JSB   =MPY10
067155 343                        POMD R#,-R#        ADJ'ED HORI X
067156 263 131 212                STMD R#,=HDISPX
067161 343                        POMD R#,-R#
067162 150 261 046                LDMD R50,=SINLD
067165 212
067166 316 377 377                JSB   =MPY10
067171 343                        POMD R#,-R#        ADJ'ED HORI Y
067172 263 141 212                STMD R#,=HDISPY
067175 251 001 000                LDM   R#,=1,0,0,0,0,0,0,12C
067200 000 000 000
067203 000 000 022
067206                    ************************  GET DEF Y DISP
067206 150 260 123                LDBD R50,=DEFSZE   DEFAULT CSIZE ?
067211 211
067212 367 007                    JZR   CDISP2       JIF YES
067214 261 036 212                LDMD R50,=CHITE    GET HIGHT RATIO
067217 316 377 377                JSB   =MPY10       ADJ BY HIGHT
067222 343                        POMD R#,-R#        GET A COPY
067223                    CDISP2  BSS   0
067223                            ARP   !12
067223 140 345                    PUMD R40,+R#       SAVE A COPY
067225 316 377 377                JSB   =CHS10       NEGATE
067230 343                        POMD R#,-R#        GET IT BACK
067231 150 261 046                LDMD R50,=SINLD    ADJ BY LDIR
```

```
                        DISP FOR HORIZ/VERT CHAR SPACE
    1 067234 212
2703 067235 316 377 377             JSB   =MPY10
2704 067240 343                     POMD  R#,-R#          POP OFF STACK
2705 067241 263 151 212             STMD  R#,=VDISPX      ADJ'ED VERT X
2706 067244 343                     POMD  R#,-R#
2707 067245 150 261 056             LDMD  R50,=COSLD
2707 067250 212
2708 067251 316 377 377             JSB   =MPY10
2709 067254 343                     POMD  R#,-R#
2710 067255 263 161 212             STMD  R#,=VDISPY      ADJ'ED VERT Y
2711 067260 236                     RTN
2712 067261
```

LABEL

```
067261
067261                  ****************************************************************
067261                  *
067261                  *     LABEL RUN TIME ROUTINE
067261                  *
067261                  ****************************************************************
067261 241                        OCT  241            PRIMARY ATTRIBUTE
067262
067262                  LABEL.    BSZ  0
067262 316 301 156                JSB  =TAKEHK        TAKE OVER HOOK
067265                            ORP  !70            R70 = 0
067265 262 064 200                STBD R70,=USING?    CLEAR USING
067270 173 263 117                STMD R73,=LBLEN     CLEAR LABEL ADDR/LENGTH
067273 212
067274 212                        DCB  R73            MAKE IT NEGATIVE
067275 262 116 212                STBD R73,=LBLFLG    SET LABEL FLAG
067300 236              LABEX     RTN
067301
067301                  TAKEHK    BSS  0
067301 171 261 243                LDMD R71,=IOTRFC    SAVE IO HOOK
067304 207
067305 263 014 212                STMD R71,=PRATIO    IN A TEMPORARY
067310 261 324 156                LDMD R71,=LBHOOK    TAKE OVER HOOK
067313 263 243 207                STMD R71,=IOTRFC
067316 170 223                    CLM  R70            SET SCTEMP
067320 263 321 203                STMD R70,=SCTEMP
067323 236                        RTN
067324
067324                  LBHOOK    BSS  0
067324 316 377 377                JSB  =ROMJSB        HOOK TO TRAP ROUTINE
067327 333 156                    DEF  LBTRAP
067331 001                        VAL  ROM#
067332 236                        RTN
067333
067333                  LBTRAP    BSS  0
067333 230                        BIN
067334 124 223                    CLM  R24
067336 210                        ICB  R24
067337 262 116 212                STBD R24,=LBLFLG    SET POSITIVE FLAG
067342 261 117 212                LDMD R24,=LBLEN     GET LABEL LENGTH
067345 366 014                    JNZ  LABOL2         JIF ALREADY SOME
067347 136 263 117                STMD R36,=LBLEN     SET LENGTH
067352 212
067353 175 261 314                LDMD R75,=PTR2
067356 377
067357 263 121 212                STMD R75,=LBADDR    SET ADDR
067362 236                        RTN
067363
067363                  LABOL2    BSS  0
067363 136 221                    TSM  R36            ZERO LENGTH
067365 367 311                    JZR  LABEX          JIF YES. DON'T DO ANYTHING.
067367                  ******    CLB  R40
```

```
                                    LABEL
067367                        ******* LDMD R75,=PTR2    CHECK IF ENOUGH ROOM
067367                        ******* STM  R75,R65      SAVE NEW ADDR
067367                        ******* ADM  R75,R36      R40 MUST BE 0
067367                        ******* CMMD R75,=SAVPT2  ENOUTGH ROOM
067367                        ******* JCY  LABEX        JIF NO
067367                        ******* STMD R75,=PTR2    SET DEST FOR MOVE
067367                        ******* LDMD R55,=PTR1    SAVE PC
067367                        ******* PUMD R55,+R6
067367                        ******* LDM  R55,R36      SAVE CURR LENGTH
067367                        ******* LDMD R36,=LBLEN   GET OLD LENGTH
067367                        ******* LDM  R45,R36      COPY OLD LENGTH
067367                        ******* ADM  R45,R65      + CURR ADR
067367                        ******* SBMD R45,=LBADDR  - OLD ADR = BYTE COUNT
067367                        ******* STMD R65,=PTR1
067367                        ******* LDM  R65,R55      GET NEW STRING LEN
067367                        ******* ADMD R65,=LBADDR  ADD TO OLD ADDR
067367                        ******* STMD R65,=LBADDR  SET NEW ADDR
067367                        ******* ADM  R36,R55      OLD LEN + NEW LEN
067367                        ******* STMD R36,=LBLEN   = CURR LEN
067367                        ******* JSB  =EMOVUP      MOVE WHOLE THING UP
067367                        ******* STMD R75,=PTR1    MOVE CURR STR DOWN
067367                        ******* LDM  R45,R55
067367                        ******* JSB  =EMOVUP
067367                        ******* POMD R55,-R6      RESTORE PC
067367                        ******* STMD R55,=PTR1
067367                        ******* RTN
067367
067367 155 223                        CLM  R55          RESET ACCU
067371 102 261 117                    LDMD R2,=LBLEN    GET OLD LENGTH
067374 212
067375 055 243                        STM  R2,R55       MAKE IT 3 BYTES
067377 316 377 377                    JSB  =RESMEM      RESERVE SPACE
067402 370 274                        JEN  LABEX        JIF NOT ENOUGH SPACE
067404                                DRP  !75
067404 261 310 377                    LDMD R75,=PTR1    SAVE PTR1
067407 006 345                        PUMD R75,+R6
067411 261 121 212                    LDMD R75,=LBADDR  GET OLD ADDR
067414 263 310 377                    STMD R75,=PTR1    SOURCE FOR MOVE
067417 165 263 121                    STMD R65,=LBADDR  UPDATE LABEL ADDR
067422 212
067423 263 314 377                    STMD R65,=PTR2    DESTINATION
067426 136 055 303                    ADM  R36,R55      ADD NEW LENGTH
067431 263 117 212                    STMD R36,=LBLEN   UPDATE LENGTH
067434 145 241                        LDM  R45,R55      BYTE COUNT
067436 316 377 377                    JSB  =EMOVUP      MOVE TO POSITION
067441 155 006 343                    POMD R55,-R6      RESTORE PTR1
067444 263 310 377                    STMD R55,=PTR1
067447 236                            RTN
067450
067450 035                            OCT  35           PRIMARY ATTRIBUTE
067451
067451                        LABEOL  BSS  0
067451 230                            BIN
```

```
                          LABEL
067452 122 260 064                LDBD R22,=USING?  LABEL USING ?
067455 200
067456 367 055                    JZR  SNDETX       JIF NO
067460 130 261 266                LDMD R30,=IMCLEN  GET CURR IMAGE LEN
067463 203
067464 175 261 310                LDMD R75,=PTR1    SAVE PC
067467 377
067470 006 345                    PUMD R75,+R6
067472 261 363 203                LDMD R75,=IMSADR  GET START IMAGE ADDR
067475 165 261 370                LDMD R65,=IMCADR  GET CURRT IMAGE ADDR
067500 203
067501 263 310 377                STMD R65,=PTR1    SET FOR FORMATTER
067504 075 301                    CMM  R65,R75      SAME ?
067506 766 005                    JNZ  CLSCN        JIF NO
067510 260 116 212                LDBD R65,=LBLFLG  1ST TIME
067513 765 006                    JPS  RST6         JIF NO
067515                   CLSCN    BSS  0
067515 316 377 377                JSB  =ROMJSB
067520 377 377                    DEF  SCANLX       FORMAT SCANNER
067522 100                        OCT  0
067523                   RST6     BSS  0
067523 175 006 343                POMD R75,-R6      RESTORE PC
067526 263 310 377                STMD R75,=PTR1
067531 112 261 344                LDMD R12,=TOS     CLEAN R12 STACK
067534 203
067535                   SNDETX   BSS  0
067535 132 261 117                LDMD R32,=LBLEN   GET LABEL LENGTH
067540 212
067541 030 243                    STM  R32,R30      SAVE A COPY
067543 367 042                    JZR  SEND+        JIF ZERO LENGTH
067545 175 261 121                LDMD R75,=LBADDR  GET LABEL ADDR
067550 212
067551                   SNDLX1   BSS  0
067551 130 223                    CLM  R30          RESET CURR COUNT
067553 175 263 314                STMD R75,=PTR2
067556 377
067557                   SNDLX2   BSS  0
067557 132 213                    DCM  R32          DEC TOTAL BYTE COUNT
067561 102 270 315                LDBI R2,=PTR2-    POP ONE CHAR
067564 377
067565 310 015                    CMB  R2,=CR       IS IT CR?
067567 367 006                    JZR  SENDET       JIF YES
067571 130 211                    ICM  R30          INC CURR COUNT
067573 132 221                    TSM  R32          TOTAL EXAUSTED ?
067575 766 360                    JNZ  SNDLX2       JIF NO
067577                   SENDET   BSS  0
067577 165 261 314                LDMD R65,=PTR2    SAVE PTR2
067602 377
067603 175 263 314                STMD R75,=PTR2    SET STRING ADDR
067606 377
067607
067607                   SEND+    BSS  0
067607 132 006 245                PUMD R32,+R6      SAVE REMAIN LEN
```

```
               LABEL..
067612 165 345                  PUMD R65,+R6      SAVE NEXT ADDR
067614 130 263 117              STMD R30,=LABLNG  SET LABEL LENGTH
067617 212
067620 251 024 000              LDM  R30,=24,0    CALL DRIVER
067623 316 125 145              JSB  =UFLTDR
067626 316 161 162              JSB  =DNPNAD      LOGICALLY DOWN PEN ONE LINE
067631 316 115 167              JSB  =BACKAX      PHYSICALLY MOVE THE PEN
067634 230                      BIN               BACK TO BINARY MODE
067635 175 006 343              POMD R75,-R6      RETRIEVE ADDR
067640 132 343                  POMD R32,-R6      RETRIEVE LEN
067642 366 305                  JNZ  SNDLX1       JIF MORE
067644                   RELSHK BSS  0            RELEASE HOOK
067644 171 222                  CLB  R71
067646 262 116 212              STBD R71,=LBLFLG
067651                   RESTHK BSS  0            RESTORE HOOK
067651 171 261 014              LDMD R71,=PRATIO
067654 212
067655 263 243 207              STMD R71,=IOTRFC
067660 236               C24X   RTN
067661
067661                   C24    BSS  0
067661                   *****  LDMD R2,=PEN#     RTN IF PEN 0
067661                   *****  JZR  C24X
067661                   *****  BIN
067661 316 364 157              JSB  =SAVTYP      SAVE LINE TYPE
067664 126 223                  CLM  R26          DEFAULT TYPE LNGTH
067666 316 005 155              JSB  =DEFTYP      LINE TYPE 1
067671 316 372 150              JSB  =PENDN.      PEN DOWN
067674                          DRP  !0
067674 262 131 211              STBD R0,=PCNTRL   SET PEN CNTRL = 1
067677 136 261 127              LDMD R36,=USER    SAVE USER SCALE FLAG
067702 212
067703 006 345                  PUMD R36,+R6
067705 223                      CLM  R36          FORCE TO HARD LIMIT
067706 263 127 212              STMD R36,=USER
067711 316 054 160              JSB  =SETORG      SET ORIGIN PT BY LORG
067714                   CHLOOP BSZ  0            LABEL A CHAR LOOP
067714 134 261 117              LDMD R34,=LABLNG  GET LABEL LENGTH
067717 212
067720 367 030                  JZR  LABEND       JIF ZERO LENGTH
067722 230                      BIN
067723 213                      DCM  R34          ELSE DEC LENGTH
067724 263 117 212              STMD R34,=LABLNG  UPDATE LENGTH
067727
067727 223                      CLM  R34          RESET INDEX
067730 270 315 377              LDBI R34,=PTR2-   GET A CHAR
067733 204                      LLB  R34          2 BYTE/ENTRY
067734 323 346 211              ADMD R34,=CHTABL  ADD BASE ADDR
067737 136 034 245              LDMD R36,R34      FETCH ENTRY
067742 316 051 161              JSB  =LABCHR      LABEL ONE CHAR
067745 316 145 162              JSB  =RTCHAD      RIGHT 1 CHAR POS
067750 360 342                  JMP  CHLOOP       LOOP BACK
067752                   LABEND BSZ  0
```

```
ITEM         LABEL
2902 067752 136 006 243            POMD R36,-R6       RESTORE USER SCALE FLAG
2903 067755 263 127 212            STMD R36,=USER
2904 067760                 LABEN2 ESS  0
2905 067760 316 020 160            JSB  =RESTYP       RESTORE LINE TYPE
2906 067763 236                    RTN
2907 067764
2908 067764
```

SAVE/RESTORE LINE TYPE

```
067764
067764                      ************************************************************
067764                      *                                                          *
067764                      *     SAVE LINE TYPE                                       *
067764                      *                                                          *
067764                      ************************************************************
067764
067764                      SAVTYP  BSZ  0
067764 130 006 343                  POMD R30,-R6        SAVE RETURN ADDR
067767 136 260 137                  LDBD R36,=LTYPE#    SAVE LINETYPE FLG
067772 211
067773 344                          PUBD R36,+R6
067774 251 370 211                  LDM  R36,=D#SEG     GET TYPE BFR ADDR
067777 140 036 341                  POMD R40,+R36       SAVE 20 BYTES
070002 006 345                      PUMD R40,+R6
070004 036 341                      POMD R40,+R36
070006 006 345                      PUMD R40,+R6
070010 144 036 341                  POMD R44,+R36
070013 006 345                      PUMD R44,+R6
070015 130 345                      PUMD R30,+R6        RESTORE RETURN ADDR
070017 236                          RTN
070020
070020                      ************************************************************
070020                      *                                                          *
070020                      *     RESTORE LINE TYPE AND PEN UP/DOWN FLAG               *
070020                      *                                                          *
070020                      ************************************************************
070020
070020                      RESTYP  BSZ  0
070020 130 006 343                  POMD R30,-R6        SAVE RETURN ADDR
070023 136 251 014                  LDM  R36,=DSEG-     PT TO BOTTOM OF BFR
070026 212
070027 144 343                      POMD R44,-R6        RESTORE 20 BYTES
070031 036 347                      PUMD R44,-R36
070033 140 006 343                  POMD R40,-R6
070036 036 347                      PUMD R40,-R36
070040 006 343                      POMD R40,-R6
070042 036 347                      PUMD R40,-R36
070044 006 342                      POBD R40,-R6        RESTORE LINETYPE
070046 262 137 211                  STBD R40,=LTYPE#
070051 130 345                      PUMD R30,+R6        RESTORE RET ADDR
070053 236                          RTN
070054
```

SET STARTING ADDR FOR LABEL

```
070054
070054                         ************************************************************
070054                         *
070054                         *      SET STARTING ADDR RELATIVE TO LORG                  *
070054                         *                                                          *
070054                         ************************************************************
070054
070054                         SETORG  BSZ  0
070054 230                             BIN
070055 140 223                         CLM  R40
070057 141 250 011                     LDB  R41,=9C        "-" SIGN
070062 147 250 040                     LDB  R47,=20C       FORM -2
070065 140 263 062                     STMD R40,=TMP1      INIT XORG
070070 211
070071 147 250 060                     LDB  R47,=30C       FORM -3
070074 140 263 072                     STMD R40,=TMP2      INIT YORG
070077 211
070100 136 261 117                     LDMD R36,=LABLNG    GET STRING LENGTH
070103 212
070104 140 260 025                     LDBD R40,=XLORG     GET X-IDX OF LORG
070107 212
070110 367 045                         JZR  SETOR3         JIF X-IDX=0
070112 136 205                         LLM  R36            LABEL LNGTH \* 8
070114 205                             LLM  R36
070115 205                             LLM  R36
070116 315 003 000                     SBM  R36,=3,0       SUB 3 DOTS
070121 316 377 377                     JSB  =CONBIN        CONVERT TO REAL
070124 150 260 025                     LDBD R50,=XLORG     X-IDX=1
070127 212
070130 212                             DCB  R50
070131 366 011                         JNZ  SETOR2         JIF NOT
070133 040 241                         LDM  R50,R40        COPY LENGTH OF LABEL
070135 316 051 147                     JSB  =R2.           DIVIDE IT BY 2.0
070140 316 377 377                     JSB  =DIV10
070143 343                             POMD R#,-R#         POP IT BACK
070144                         SETOR2  BSS  0
070144 150 261 062                     LDMD R50,=TMP1      MODIFY XORG
070147 211
070150 316 377 377                     JSB  =SUB10
070153 343                             POMD R#,-R#         POP OFF STACK
070154 263 062 211                     STMD R#,=TMP1       SAVE IN PLACE
070157                         SETOR3  BSZ  0
070157 260 024 212                     LDBD R#,=YLORG      GET Y-INDEX OF LORG
070162 310 001                         CMB  R#,=1
070164 366 006                         JNZ  SETOR4         JIF Y-INDEX = 0 OR 2
070166 136 251 003                     LDM  R36,=3,0       Y-DISP FOR Y-IDX = 1
070171 000
070172 360 006                         JMP  SETOR5
070174                         SETOR4  BSZ  0
070174 364 022                         JNG  SETOR6         JIF Y-INDEX = 0
070176 136 251 006                     LDM  R36,=6,0       Y-DISP FOR Y-IDX =2
070201 000
070202                         SETOR5  BSZ  0
```

```
                        SET STARTING ADDR FOR LABEL
070202 316 377 377              JSB  =CONBIN        CONVRT TO REAL
070205 150 261 072              LDMD R50,=TMP2      GET YORG
070210 211
070211 316 377 377              JSB  =SUB10         MODIFY YORG
070214 343                      POND R#,-R#         POP OFF STK
070215 263 072 211              STMD R#,=TMP2       SAVE IN PLACE
070220                  SETOR6  BSS  0
070220 316 333 161              JSB  =ADJINC        ADJ BY CSIZE,LDIR
070223 316 012 162              JSB  =ADDPEN        ADD PEN LOCATION
070226 124 251 062              LDM  R24,=TMP1      GIVING ABS LABEL ORG
070231 211
070232 126 251 170              LDM  R26,=XORG
070235 215
070236 316 343 144              JSB  =MOV20         MOVE TO DEST
070241 316 245 160              JSB  =PRECLP        DO PRECLIPPING
070244 236                      RTN
070245
070245                  PRECLP  BSS  0
070245 114 250 001              LDB  R14,=1         ASSUME NEED CLIPPING
070250 102 260 122              LDBD R2,=DEFSLA     DEFAULT SLANT ?
070253 211
070254 766 120                  JNZ  PRECL1         JIF NOT
070256 316 003 161              JSB  =POK?          CHECK LOWER LEFT CORNER
070261 371 113                  JEZ  PRECL1         JIF OUT OF WINDOW
070263 124 251 151              LDM  R24,=VDISP     MOVE UP ONE BLOCK
070266 212
070267 126 251 062              LDM  R26,=TMP1
070272 211
070273 316 372 161              JSB  =ADDOR2        BY ADDING Y DISPLACE
070276 316 003 161              JSB  =POK?          CHECK UPPER LEFT CORNER
070301 371 073                  JEZ  PRECL1         JIF OUT OF WINDOW
070303 316 036 152              JSB  =T1TOT3        SAVE UPPER LEFT COOR
070306 136 261 117              LDMD R36,=LABLNG    GET BYTE COUNT
070311 212
070312 230                      BIN
070313 205                      LLM  R36            COUNT * 8 TO GIVE
070314 205                      LLM  R36            WIDTH IN # OF DOTS
070315 205                      LLM  R36
070316 316 377 377              JSB  =CONBIN        CONVERT TO REAL
070321 140 263 062              STMD R40,=TMP1      SAVE IT
070324 211
070325 223                      CLM  R40            DELTA Y = 0
070326 263 072 211              STMD R40,=TMP2
070331 316 333 161              JSB  =ADJINC        ADJUST BY CSIZE, LDIR
070334 124 251 102              LDM  R24,=TMP3      ADD UPPER LEFT COOR
070337 211
070340 126 251 062              LDM  R26,=TMP1      TO WIDTH OF THE STRING
070343 211
070344 316 372 161              JSB  =ADDOR2        TO GIVE UPPER RIGHT COOR
070347 316 003 161              JSB  =POK?          CHECK UPPER RIGHT CORNER
070352 371 022                  JEZ  PRECL1         JIF OUT OF WINDOW
070354 124 251 151              LDM  R24,=VDISP     SUBTRACT Y DISPLACEMENT
070357 212
```

```
                              SET STARTING ADDR FOR LABEL
070360 126 251 062                   LDM  R26,=TMP1      FROM UPPER RIGHT CORNER
070363 211
070364 316 174 162                   JSB  =ONPNA2        TO GIVE LOWER RIGHT COOR
070367 316 003 161                   JSB  =POK?          CHECK LOWER RIGHT CORNER
070372 371 002                       JEZ  PRECL1         JIF OUT OF WINDOW
070374 114 212                       DCB  R14            FLAG NO CLIP
070376                       PRECL1  BSS  0
070376 114 262 127                   STBD R14,=CLPFLG     SET CLIP FLAG
070401 211
070402 236                           RTN
070403
070403                       *
070403                       *     TMP1 = X COORDINATE OF THE POINT
070403                       *     TMP2 = Y COORDINATE OF THE POINT
070403                       *
070403                       *     E = 0 IF THE POINT IS IN WINDOW
070403                       *         1 IF THE POINT IS OUT OF WINDOW
070403
070403                       POK?    BSS  0
070403 120 223                       CLM  R20            INDEX TO XMIN
070405 124 223                       CLM  R24            INDEX TO X COOR
070407 316 346 152                   JSB  =IN?           X > XMIN ?
070412 371 034                       JEZ  PNOTOK         JIF NO
070414 230                           BIN
070415 120 312 010                   ADB  R20,=10        INDEX TO XMAX
070420 316 346 152                   JSB  =IN?           X < XMAX ?
070423 371 023                       JEZ  PNOTOK         JIF NO
070425 230                           BIN
070426 124 312 010                   ADB  R24,=10        INDEX TO Y COOR
070431 120 312 010                   ADB  R20,=10        INDEX TO Y MIN
070434 316 346 152                   JSB  =IN?           Y > YMIN ?
070437 371 007                       JEZ  PNOTOK         JIF NO
070441 230                           BIN
070442 120 312 010                   ADB  R20,=10        INDEX TO Y MAX
070445 316 346 152                   JSB  =IN?           Y < YMAX
070450 236                   PNOTOK  RTN                 RTN THE FLAG
070451
```

```
070451
070451                       ********************************************************
070451                       *                                                      *
070451                       *    LABEL A CHARACTER                                 *
070451                       *                                                      *
070451                       *      ENTRY:                                          *
070451                       *        R36-37=ENTRY OF CHRTAB FOR THE CHARACTER      *
070451                       *                                                      *
070451                       ********************************************************
070451
070451                       LABCHR  BSZ  0
070451 316 154 161                   JSB  =SEPC          SEPERATE STRK COUNT
070454                       *                           FROM ADDR TO STRK TABLE
070454 316 202 161                   JSB  =GTSTRK        GET STRK COOR
070457 316 234 161                   JSB  =CNVRTS        CONVERT TO REAL
070462 316 263 161                   JSB  =ADJSTR        ADJ BY LDIR & CSIZE
070465 316 216 151                   JSB  =UPDPT         INIT 1ST PT
070470 316 372 150                   JSB  =PENDN.        PEN DOWN
070473                       STLOOP  BSZ  0              STROKE LOOP
070473 316 202 161                   JSB  =GTSTRK        GET A STROKE
070476 121 310 017                   CMB  R21,=17        IS IT A CONTROL FUNC
070501 367 044                       JZR  STCNTR         JIF YES
070503 316 234 161                   JSB  =CNVRTS        ELSE CONVERT TO REAL
070506 316 263 161                   JSB  =ADJSTR        ADJ BY LDIR & CSIZE
070511 316 216 151                   JSB  =UPDPT         UPDATE STROKE POINT
070514 137 260 126                   LDBD R37,=PENDN     GET PEN FLAG
070517 211
070520 763 020                       JEV  NXTST2         JIF PEN UP
070522 316 355 151                   JSB  =DRAW.+        ELSE CLIP & DRAW
070525                       *****   JSB  =GRAPH         FORCE TO GRAPH MODE
070525                       *****   JSB  =CLIPLN        CLIP THE STROKE
070525                       *****   JEN  NXTSTR         JIF NOTHING TO DRAW
070525                       *****   JSB  =DRAWC         ELSE DRAW STOKE
070525                       NXTSTR  BSZ  0
070525 230                           BIN
070526 260 124 212                   LDBD R#,=NOSTRK     DEC STROKE COUNT
070531 212                           DCB  R#
070532 764 005                       JNG  ENDCHR         JIF END OF CHAR
070534 262 124 212                   STBD R#,=NOSTRK     UPDATE COUNT
070537 760 332                       JMP  STLOOP         LOOP BACK
070541 236                   ENDCHR  RTN
070542
070542                       NXTST2  BSS  0
070542 316 372 150                   JSB  =PENDN.        DOWN PEN
070545 760 356                       JMP  NXTSTR
070547
070547                       ********************************************************
070547                       *                                                      *
070547                       *    STROKE CONTROL FUNCTIONS                          *
070547                       *                                                      *
070547                       *      ENTRY:                                          *
070547                       *        R20=FUNCTION CODE                             *
070547                       *              =0 : PEN UP                             *
```

```
========================================================================================
3112                  LABEL A CHARACTER
3113 070547                        *                =1 : PEN DOWN
3114 070547                        *
3114 070547                        ************************************************************
3115 070547
3116 070547                        STCNTR  BSZ  0
3117 070547 316 357 150                    JSB  =PENUP.        PEN UP FOR CODE 0
3118 070552 260 351                        JMP  NXTSTR         GO ON FOR NXT STRK
3119 070554
```

LABCHR UTILITIES

```
070554
070554                  ************************************************************
070554                  *
070554                  *     SEPERATE # OF STROKES FROM POINTER TO STROKE TABLE*
070554                  *     AND STORE THEM IN PLACE                            *
070554                  *                                                        *
070554                  *       ENTRY:                                           *
070554                  *         R36-37=ENTRY OF CHRTAB FOR THE CHAR            *
070554                  *                                                        *
070554                  ************************************************************
070554
070554                  SEPC      BSZ  0
070554 231                        BCD
070555 134 222                    CLB  R34            RESET BUFFER
070557 136 201                    ELM  R36            SHIFT OUT # OF STRK -
070561 134 200                    ELB  R34            SHIFT INTO R34
070563 137 207                    LRM  R37            SHIFT PTR BACK
070565 230                        BIN
070566 134 262 124                STBD R34,=NOSTRK    SAVE # OF STROKES
070571 212
070572 136 313 377                ADM  R36,=STROK     ADD STRK TABLE BASE
070575 377
070576 263 125 212                STMD R36,=STRKAD    SAVE PTR
070601 236                        RTN                 DONE
070602
070602                  ************************************************************
070602                  *
070602                  *     GET THE COOR OF THE END PT FOR NEXT STROKE         *
070602                  *                                                        *
070602                  *       ENTRY:                                           *
070602                  *         STRKAD=PTR TO NEXT PT                          *
070602                  *                                                        *
070602                  *       RETURN:                                          *
070602                  *         R21=X-COOR OF THE PT IN THE CHAR MATRIX        *
070602                  *         R20=Y-COOR                                     *
070602                  *                                                        *
070602                  ************************************************************
070602
070602                  GTSTRK    BSZ  0
070602 136 261 125                LDMD R36,=STRKAD    GET POINTER
070605 212
070606 120 036 340                POBD R20,+R36       GET COORDINATE
070611 136 263 125                STMD R36,=STRKAD    UPDATE POINTER
070614 212
070615 316 221 161                JSB  =SEPS          SEP X FROM Y COOR
070620 236                        RTN
070621
070621                  SEPS      BSZ  0
070621 121 222                    CLB  R21            RESET BUFFER
070623 231                        BCD                 FOR FAST SHIFT
070624 120 200                    ELB  R20            SHIFT OUT X-COOR
070626 121 200                    ELB  R21            SHIFT INTO R21
070630 120 206                    LRB  R20            SHIFT Y-COOR BACK
```

```
070554
070554                   ************************************************************
070554                   *                                                          *
070554                   *    SEPERATE # OF STROKES FROM POINTER TO STROKE TABLE*
070554                   *    AND STORE THEM IN PLACE                               *
070554                   *                                                          *
070554                   *      ENTRY:                                              *
070554                   *        R36-37=ENTRY OF CHRTAB FOR THE CHAR               *
070554                   *                                                          *
070554                   ************************************************************
070554
070554                   SEPC    BSZ  0
070554 231                       BCD
070555 134 222                   CLB  R34           RESET BUFFER
070557 136 201                   ELM  R36           SHIFT OUT # OF STRK
070561 134 200                   ELB  R34           SHIFT INTO R34
070563 137 207                   LRM  R37           SHIFT PTR BACK
070565 230                       BIN
070566 134 262 124               STBD R34,=NOSTRK   SAVE # OF STROKES
070571 212
070572 136 313 377               ADM  R36,=STROK    ADD STRK TABLE BASE
070575 377
070576 263 125 212               STMD R36,=STRKAD   SAVE PTR
070601 236                       RTN                DONE
070602
070602                   ************************************************************
070602                   *                                                          *
070602                   *    GET THE COOR OF THE END PT FOR NEXT STROKE            *
070602                   *                                                          *
070602                   *      ENTRY:                                              *
070602                   *        STRKAD=PTR TO NEXT PT                             *
070602                   *                                                          *
070602                   *      RETURN:                                             *
070602                   *        R21=X-COOR OF THE PT IN THE CHAR MATRIX           *
070602                   *        R20=Y-COOR                                        *
070602                   *                                                          *
070602                   ************************************************************
070602
070602                   GTSTRK  BSZ  0
070602 136 261 125               LDMD R36,=STRKAD   GET POINTER
070605 212
070606 120 036 340               POBD R20,+R36      GET COORDINATE
070611 136 263 125               STMD R36,=STRKAD   UPDATE POINTER
070614 212
070615 316 221 161               JSB  =SEPS         SEP X FROM Y COOR
070620 236                       RTN
070621
070621                   SEPS    BSZ  0
070621 121 222                   CLB  R21           RESET BUFFER
070623 231                       BCD                FOR FAST SHIFT
070624 120 200                   ELB  R20           SHIFT OUT X-COOR
070626 121 200                   ELB  R21           SHIFT INTO R21
070630 120 206                   LRB  R20           SHIFT Y-COOR BACK
```

LABCHR UTILITIES

```
070632 230                   BIN
070633 236                   RTN
070634
070634            ****************************************************************
070634            *                                                              *
070634            *     CONVERT COOR FROM BINARY TO REAL                         *
070634            *                                                              *
070634            ****************************************************************
070634
070634            CNVRTS  BSZ   0
070634 136 223            CLM   R36              RESET BUFFER
070636 021 240            LDB   R36,R21          LOAD X-COOR
070640 316 377 377        JSB   =CONBIN          CONVERT TO REAL
070643 140 263 062        STMD  R40,=TMP1        STORE IN PLACE
070646 211
070647 136 223            CLM   R36
070651 020 240            LDB   R36,R20          LOAD Y-COOR
070653 316 377 377        JSB   =CONBIN          CONVERT TO REAL
070656 140 263 072        STMD  R40,=TMP2        STORE IN PLACE
070661 211
070662 236                RTN
070663
```

```
070663
070663          ************************************************************
070663          *                                                          *
070663          *     ADJUST STROKE COORDINATE                             *
070663          *     BY                                                   *
070663          *       1.   ADJUST BY SLANT                               *
070663          *       2.   ELONGATE BY CSIZE                             *
070663          *       3.   ROTATE BY LDIR                                *
070663          *       4.   ADD THE COOR OF THE ORIG OF CHAR              *
070663          *                                                          *
070663          ************************************************************
070663
070663                   ADJSTR  BSS  0
070663 316 275 161               JSB  =ADJSLA       ADJUST BY SLANT
070666 316 333 161               JSB  =ADJINC       ADJUST RELATIVE ADDR
070671 316 360 161               JSB  =ADXORG       ADD TO GIVE ABS ADDR
070674 236                       RTN
070675
070675                   ADJSLA  BSS  0
070675 100 260 122               LDBD R0,=DEFSLA    DEFAULT SLANT
070700 211
070701 367 027                   JZR  ADJSL1        RTN IF YES
070703 140 261 106               LDMD R40,=TANSL    GET TAN(SLANT)
070706 212
070707 150 261 072               LDMD R50,=TMP2     GET Y COOR
070712 211
070713 316 377 377               JSB  =MPY10
070716 343                       POMD R#,-R#        POP OFF STACK
070717 150 261 062               LDMD R50,=TMP1
070722 211
070723 316 377 377               JSB  =ADD10
070726 343                       POMD R#,-R#
070727 263 062 211               STMD R#,=TMP1
070732 236               ADJSL1  RTN
070733
070733                   ADJINC  BSZ  0
070733 260 123 211               LDBD R#,=DEFSZE    DEFAULT CSIZE ?
070736 367 003                   JZR  ADJST1        JIF YES
070740 316 020 162               JSB  =ADJSZE       ADJUST BY CSIZE
070743 260 124 211       ADJST1  LDBD R#,=DEFLDR    LDIR = 0 ?
070746 367 007                   JZR  ADJST2        JIF YES
070750 124 251 046               LDM  R24,=SINLD    ROTATE BY LDIR
070753 212
070754 316 050 162               JSB  =ROTAXY
070757                   ADJST2  BSS  0
070757 236                       RTN
070760
070760                   ADXORG  BSS  0
070760 124 251 170               LDM  R24,=XXORG    PT TO XORG
070763 215
070764 360 000                   JMP  ADDOR+
070766
070766 126 251 062       ADDOR+  LDM  R26,=TMP1     POINT TO COOR
```

ADJUST STROKE COORDINATE

```
070771 211
070772 316 375 161 ADDOR2   JSB   =ADDOR1
070775                 ADDOR1   BSZ   0
070775 150 024 341               POMD  R50,+R24     POP ORG
071000 140 026 245               LDMD  R40,R26      GET COOR
071003 316 377 377               JSB   =ADD10       ADD
071006 343                       POMD  R#,-R#       POP OFF STK
071007 026 345                   PUMD  R#,+R26      STORE BACK
071011 236                       RTN
071012
071012                 ADDPEN   BSS   0
071012 124 251 002               LDM   R24,=XPEN
071015 211
071016 360 346                   JMP   ADDOR+
071020
071020                 ADJSZE   BSZ   0
071020 124 251 026               LDM   R24,=CWIDE   POINT TO RATIO
071023 212
071024 126 251 062               LDM   R26,=TMP1    POINT TO COOR
071027 211
071030 316 033 162               JSB   =ADJSZ1
071033                 ADJSZ1   BSZ   0
071033 150 024 341               POMD  R50,+R24     POP RATIO
071036 140 026 245               LDMD  R40,R26      LOAD COOR
071041 316 377 377               JSB   =MPY10       ADJ BY RATIO
071044 343                       POMD  R#,-R#       POP OFF STK
071045 026 345                   PUMD  R#,+R26      PUSH BACK
071047 236                       RTN
071050
071050                 ****************************************************
071050                 *                                                  *
071050                 *    COORDINATE AFTER ROTATION :                   *
071050                 *      X'=X*COSLD-Y*SINLD                          *
071050                 *      Y'=X*SINLD+Y*COSLD                          *
071050                 *      R24=PTR TO SINLD OR SINPD                   *
071050                 *                                                  *
071050                 ****************************************************
071050
071050                 ROTAXY   BSZ   0
071050 126 251 062               LDM   R26,=TMP1    POINT TO COOR
071053 211
071054 150 026 341               POMD  R50,+R26     GET X
071057 012 345                   PUMD  R50,+R12     SAVE X
071061 140 024 341               POMD  R40,+R24     GET SIN(LDIR)
071064 012 345                   PUMD  R40,+R12     SAVE ON STK
071066 026 245                   LDMD  R40,R26      GET Y
071070 012 345                   PUMD  R40,+R12     SAVE Y
071072 024 245                   LDMD  R40,R24      GET COS(LDIR)
071074 012 345                   PUMD  R40,+R12     SAVE ON STK
071076 316 377 377               JSB   =MPY10       X*COSLD
071101                           DRP   !40
071101 024 343                   POMD  R40,-R24     GET SIN(LDIR)
071103 150 026 245               LDMD  R50,R26      GET Y
```

```
                        ADJUST STROKE COORDINATE
071106 316 377 377          JSB   =MPY10        Y*SINLD
071111 316 377 377          JSB   =SUBR01       X*COSLD-Y*SINLD
071114 343                  POMD  R#,-R#        POP OFF X'
071115 263 062 211          STMD  R#,=TMP1      SAVE IN PLACE
071120 316 377 377          JSB   =MPYR01       Y*COSLD
071123 343                  POMD  R#,-R#        POP OFF STK
071124 306 345              PUMD  R#,+R6        SAVE ON SYS STK
071126 316 377 377          JSB   =MPYR01       X*SINLD
071131 306 343              POMD  R#,-R6        POP PREVIOUS RESULT
071133 012 345              PUMD  R#,+R12
071135 316 377 377          JSB   =ADDR01       X*SINLD+Y*COSLD
071140 343                  POMD  R#,-R#        POP OFF Y'
071141 263 072 211          STMD  R#,=TMP2      SAVE IN PLACE
071144 236                  RTN
071145
```

MOVE ONE CHAR SPACE IN GRAPHICS

```
071145
071145                   ************************************************************
071145                   *
071145                   *     MOVE CHAR POINTER RIGHT ONE POSITION                 *
071145                   *
071145                   ************************************************************
071145
071145                   RTCHAD  BSZ  0
071145 124 251 131               LDM  R24,=HDISP    HORIZ DISP
071150 212
071151 126 251 170               LDM  R26,=XNORG    ABS LABEL ORG ADDR
071154 215
071155 316 372 161               JSB  =ADDOR2       RIGHT ONE POSITION
071160 236                       RTN
071161
071161                   ************************************************************
071161                   *
071161                   *     MOVE PEN LOCATION DOWN ONE CHAR POSITION             *
071161                   *
071161                   ************************************************************
071161
071161                   DNPNAD  BSZ  0
071161 316 357 150               JSB  =PENUP.       MOVE PEN UP
071164 124 251 151               LDM  R24,=VDISP    GET UNIT VERT DISP
071167 212
071170 126 251 002               LDM  R26,=XPEN     GET PEN ADDR
071173 211
071174 316 177 162       DNPNA2  JSB  =DNPNA1
071177                   DNPNA1  BSS  0
071177 150 026 245               LDMD R50,R26       GET PEN COOR
071202 140 024 341               POMD R40,+R24      GET DISP
071205 316 377 377               JSB  =SUB10        ADJUST
071210 343                       POMD R#,-R#        POP OFF STACK
071211 026 345                   PUMD R#,+R26       UPDATE
071213 236                       RTN
071214
```

```
             INPUT IN GRAPHICS
071214
071214                     CLOOP0  BSS  0            INPUT LOOP
071214                             DRP  !14
071214 261 267 203                 LDMD R14,=ERRTMP  GET CIR BFR OUT PTR
071217 321 317 203                 CMMD R14,=INPR10  ANYTHING IN CIR BFR ?
071222 766 114                     JNZ  CHECK-       JIF YES
071224 316 377 377                 JSB  =COUNTK      DELAY FOR A WHILE
071227 132 260 162                 LDBD R32,=SVCWRD  ANY REQUEST ?
071232 200
071233 763 064                     JEV  CLOOP        JIF NO
071235 212                         DCB  R32          CLEAR REQUEST
071236 262 162 200                 STBD R32,=SVCWRD
071241 117 317 357                 ANM  R17,=357     RESET REQUEST
071244 114 261 143                 LDMD R14,=INPTR
071247 202
071250 132 260 142                 LDBD R32,=KEYHIT  REPEAT LAST KEY
071253 202
071254 760 133                     JMP  CHECK2
071256
071256                     GRAFIN  BSS  0            CHEDIT FOR INPUT IN GRAF
071256 171 261 277                 LDMD R71,=KYIDLE  SAVE HOOK
071261 207
071262 263 161 210                 STMD R71,=RESEND
071265 261 144 163                 LDMD R71,=KYHOOK  TAKE OVER HOOK
071270 263 277 207                 STMD R71,=KYIDLE
071273 132 250 077                 LDB  R32,=77      OUTPUT A "?"
071276 316 260 163                 JSB  =INGRAF
071301 114 251 076                 LDM  R14,=ERRBUF  INIT POINTERS
071304 201
071305 263 317 203                 STMD R14,=INPR10  INIT INP PTR FOR CIR BFR
071310 263 267 203                 STMD R14,=ERRTMP  INIT OUT PTR FOR CIR BFR
071313 251 236 200                 LDM  R14,=INPBUF
071316 263 143 202                 STMD R14,=INPTR   INIT INP PTR FOR INPUT BFR
071321                     CLOOP   BSS  0
071321 114 260 002                 LDBD R14,=KEYSTS  A KEY DOWN ?
071324 377
071325 206                         LRB  R14
071326 762 264                     JOD  CLOOP0       JIF YES
071330                     CLOOP1  BSS  0
071330                             DRP  !14
071330 261 267 203                 LDMD R14,=ERRTMP  GET CIR BFR OUT PTR
071333 321 317 203                 CMMD R14,=INPR10  ANYTHING IN CIR BFR ?
071336 367 361                     JZR  CLOOP        JIF NO
071340                     CHECK-  BSS  0
071340                             DRP  !14
071340 311 142 201                 CMM  R14,=ERBEND  END OF CIR BFR
071343 766 003                     JNZ  CHECK        JIF NOT
071345 251 076 201                 LDM  R14,=ERRBUF  ELSE WRAP AROUND
071350                     CHECK   BSS  0
071350 132 014 340                 POBD R32,+R14     FETCH ONE CHAR
071353 114 263 267                 STMD R14,=ERRTMP  UPDATE OUT PTR
071356 203
071357 260 154 200                 LDBD R14,=KRPET1  RESTORE REPEAT COUNT
```

```
071362 262 153 200               STBD  R14,=KEYCNT
071365 261 143 202               LDMD  R14,=INPTR     GET INP BFR PTR
071370 132 310 213               CMB   R32,=213       RESET ?
071373 766 004                   JNZ   CHECK1         JIF NOT
071375 104 251 377               GTO   RSTART         GO DO RESET
071400 377
071401                   CHECK1  BSS   0
071401                           DRP   !32
071401 310 216                   CMB   R32,=216       PAUSE ?
071403 367 110                   JZR   PAUSE.         JIF YES
071405 310 232                   CMB   R32,=232       CR ?
071407 367 067                   JZR   CR.            JIF YES
071411                   CHECK2  BSS   0
071411                           DRP   !32
071411 310 231                   CMB   R32,=231       BACK SPACE
071413 766 033                   JNZ   NOTBS          JIF NOT
071415                   *
071415                   ***do back space
071415                   *
071415 114 311 236               CMM   R14,=INPBUF    BEGIN OF BFR ?
071420 200
071421 367 276                   JZR   CLOOP          JIF YES
071423 316 347 163               JSB   =LTPNAD        LEFT PEN ADDR
071426 132 014 342               POBD  R32,-R14       POP BACK ONE CHAR
071431 114 263 143               STMD  R14,=INPTR     UPDATE INP PTR
071434 202
071435 316 147 163               JSB   =COMPLI        COMPLEMENT PEN
071440 316 270 163               JSB   =INGSUB        ERASE THE CHAR
071443 316 147 163               JSB   =COMPLI        RESTORE PEN
071446 760 251                   JMP   CLOOP
071450                   NOTBS   BSS   0
071450                           DRP   !32
071450 310 240                   CMB   R32,=KEYE      "E" ON NUM KEY PAD ?
071452 766 002                   JNZ   NRMLKY         JIF NOT
071454 250 105                   LDB   R32,=EKEY      ELSE MAKE IT LETTER "E"
071456                   NRMLKY  BSS   0
071456 114 311 076               CMM   R14,=LAST+1    BFR FULL ?
071461 201
071462 367 235                   JZR   CLOOP          JIF YES
071464 132 014 344               PUBD  R32,+R14       PUT IN BFR
071467 114 263 143               STMD  R14,=INPTR     UPDATE INP PTR
071472 202
071473 316 260 163               JSB   =INGRAF        ALSO LABEL IT
071476 760 221                   JMP   CLOOP
071500                   KEYE    EQU   240
071500                   EKEY    EQU   105
071500
071500                   CR.     BSS   0
071500 132 250 015               LDB   R32,=CR
071503 014 344                   PUBD  R32,+R14       PUSH A CR
071505 251 236 200               LDM   R32,=INPBUF    RESET PTR
071510 263 143 202               STMD  R32,=INPTR
071513 760 017                   JMP   PAUSE1
```

```
3408               INPUT IN GRAPHICS
3409 071515
3410 071515                        PAUSE.   BSS   0
3410 071515                                 DRP   !32
3411 071515 262 142 202                     STBD  R32,=KEYHIT   SAVE THE CHAR
3412 071520 260 162 200                     LDBD  R32,=SVCWRD   FLAG KEY
3413 071523 210                             ICB   R32
3414 071524 262 162 200                     STBD  R32,=SVCWRD
3415 071527 250 260                         LDB   R32,=260      GET OUT & POST SRVC
3416 071531 117 032 224                     ORB   R17,R32
3417 071534                        PAUSE1   BSS   0
3418 071534 171 261 161                     LDMD  R71,=RESEND   RESTORE HOOK
3418 071537 210
3419 071540 263 277 207                     STMD  R71,=KYIDLE
3420 071543 236                             RTN                 LET SYS HANDLE IT
3421 071544
3422 071544                        KYHOOK   BSS   0
3423 071544 316 166 163                     JSB   =BFRIN        DO BFR IN, NEVER COME BACK HER
3424 071547
3425 071547                        COMFLI   BSS   0
3426 071547 230                             BIN
3427 071550 100 261 135                     LDMD  R0,=PEN#      GET PEN #
3427 071553 211
3428 071554 311 372 377                     CMM   R0,=372,377   IS IT PEN -2 ?
3429 071557 367 004                         JZR   EXIT          JIF YES
3430 071561 215                             TCM   R0            COMPLEMENT PEN SENSE
3431 071562 263 135 211                     STMD  R0,=PEN#      UPDATE PEN
3432 071565 236                     EXIT     RTN
3433 071566
3434 071566                        BFRIN    BSS   0
3435 071566 262 001 377                     STBD  R#,=GINTDS      DISABLE INTERRUPTS
3436 071571 262 100 377                     STBD  R#,=INTRSC      REENABLE OTHER INTERFACES
3437 071574 144 012 345                     PUMD  R44,+R12        SAVE R44
3438 071577 146 261 317                     LDMD  R46,=INPR10     GET INPUT PTR
3438 071602 203
3439 071603 311 142 201                     CMM   R46,=ERBEND     END OF BUFFER ?
3440 071606 366 003                         JNZ   BFRIN1          JIF NOT
3441 071610 251 076 201                     LDM   R46,=ERRBUF     ELSE WRAP AROUND
3442 071613                        BFRIN1   BSS   0
3443 071613 144 260 003                     LDBD  R44,=KEYCOD     GET THE KEY
3443 071616 377
3444 071617 310 213                         CMB   R44,=213        IS IT RESET ?
3445 071621 366 004                         JNZ   BFR1.5          JIF NOT
3446 071623 104 251 377                     GTO   RSTART          ELSE RESTART
3446 071626 377
3447 071627
3448 071627                        BFR1.5   BSS   0
3449 071627                                 DRP   !44
3450 071627 046 344                         PUBD  R44,+R46        PUSH IN BUFFER
3451 071631 146 321 267                     CMMD  R46,=ERRTMP     BFR FULL ?
3451 071634 203
3452 071635 367 003                         JZR   BFRIN2          JIF YES
3453 071637 263 317 203                     STMD  R46,=INPR10     UPDATE INPUT PTR
3454 071642                        BFRIN2   BSS   0
```

```
ITEM    LOC    OBJECT CODE   SRC=GCGRAF OBJ=GEXGRA   9/11/1981 12:38 PM  PG 1
                      INPUT IN GRAPHICS
3456 071642 144 250 001          LDB   R44,=1          RESTART KEYBOARD
3457 071645 262 003 377          STBD  R44,=KEYCOD
3458 071650 006 343              POMD  R44,-R6         TRASH TWO RTN
3459 071652 012 343              POMD  R44,-R12        RESTORE R44
3460 071654 104 251 377          GTO   ENDSR           RTN TO THE INTERRUPTED ROUTI
3461 071657 377
3462 071660
3463 071660                 ********************************************************
3464 071660                 *                                                      *
3465 071660                 *     INPUT IN GRAPHICS                                *
3466 071660                 *                                                      *
3467 071660                 *       INPUT:                                         *
3468 071660                 *         R32=ASCII CODE OF THE CHARACTER              *
3469 071660                 *                                                      *
3470 071660                 *                                                      *
3471 071660                 ********************************************************
3472 071660
3473 071660                 INGRAF  BSS   0
3474 071660 316 270 163          JSB   =INGSUB         SPIT ONE CHAR
3475 071663 316 341 163          JSB   =RTPNAD         UPDATE PEN LOCATION
3476 071666 230                  BIN
3477 071667 236                  RTN
3478 071670
3479 071670                 INGSUB  BSS   0
3480 071670 140 261 002          LDMD  R40,=XPEN       SAVE PEN LOCATION
3481 071673 211
3482 071674 006 345              PUMD  R40,+R6
3483 071676 261 012 211          LDMD  R40,=YPEN
3484 071701 345                  PUMD  R40,+R6
3485 071702
3486 071702                 *        LDMD  R46,=XLORG    SAVE LORG
3487 071702                 *        PUMD  R46,+R6
3488 071702                 *        LDMD  R40,=SINLD    SAVE LDIR
3489 071702                 *        PUMD  R40,+R6
3490 071702                 *        LDMD  R40,=COSLD
3491 071702                 *        PUMD  R40,+R6
3492 071702                 *        CLM   R40
3493 071702                 *        STMD  R40,=SINLD    FORCE LDIR=0
3494 071702                 *        STMD  R46,=XLORG    FORCE LORG=1
3495 071702                 *        LDB   R47,=10C
3496 071702                 *        STMD  R40,=COSLD
3497 071702                 *        JSB   =LABEL.       SPIT THE CHAR
3498 071702                 *        POMD  R40,-R6       RESTORE LDIR
3499 071702                 *        STMD  R40,=COSLD
3500 071702                 *        POMD  R40,-R6
3501 071702                 *        STMD  R40,=SINLD
3502 071702                 *        POMD  R46,-R6       RESTORE LORG
3503 071702                 *        STMD  R46,=XLORG
3504 071702
3505 071702 145 261 025          LDMD  R45,=LAVAIL     GET LAST AVAILABLE SPACE
3506 071705 200
3507 071706 263 314 377          STMD  R45,=PTR2       SET UP POINTER
3508 071711 132 272 316          STBI  R32,=PTR2+      PUSH CHARACTER
```

INPUT IN GRAPHICS

```
071714 377
071715 251 001 000           LDM  R32,=1,0      SET LENGTH = 1
071720 263 117 212           STMD R32,=LABLNG
071723 316 261 157           JSB  =C24          GO LABEL THE CHAR
071726
071726 140 006 343           POMD R40,-R6       RESTORE PEN LOC
071731 263 012 211           STMD R40,=YPEN
071734 343                   POMD R40,-R6
071735 263 002 211           STMD R40,=XPEN
071740 236                   RTN
071741
071741
071741               RTPNAD  BSS  0
071741               ******* LDM  R26,=XPEN     MOVE PEN RIGHT ONE POSI
071741               ******* LDM  R24,=HDISP
071741               ******* JSB  =ADDCR2
071741               ******* RTN
071741
071741 102 251 372           LDM  R2,=ADDOR2    ENTRY TO JSB TO
071744 161
071745 760 004               JMP  MVPNAD        GO MOVE PEN ADDR
071747
071747
071747               LTPNAD  BSS  0
071747 102 251 174           LDM  R2,=DNPNA2    ENTRY TO JSB TO
071752 162
071753               MVPNAD  BSS  0
071753 126 251 002           LDM  R26,=XPEN     MOVE PEN LEFT ONE POSITION
071756 211
071757 124 251 131           LDM  R24,=HDISP
071762 212
071763 002 306 000           JSB  X2,ZRO
071766 000
071767 236                   RTN
071770
```

```
                        XAXIS / YAXIS
071770
071770                  *************************************************
071770                  *                                               *
071770                  *     XAXIS RUN TIME ROUTINE                    *
071770                  *                                               *
071770                  *************************************************
071770
071770 241                      OCT 241             PRIMARY ATTRIBUTE
071771
071771                  YAXIS.  BSZ  0
071771 110 223                  CLM  R10            XAXIS INDEX
071773 130 250 001              LDB  R30,=1         X AXIS FLAG
071776 260 010                  JMP  X-YAX
072000
072000                  *************************************************
072000                  *                                               *
072000                  *     YAXIS RUN TIME ROUTINE                    *
072000                  *                                               *
072000                  *************************************************
072000
072000 241                      OCT  241            PRIMARY ATTRIBUTE
072001
072001                  YAXIS.  BSZ  0
072001 110 251 042              LDM  R10,=YINDX     YAXIS INDEX
072004 000
072005 130 250 002              LDB  R30,=2         Y AXIS FLAG
072010                  X-YAX   BSS  0
072010 262 130 211              STBD R#,=AXFLAG     SET AXES FLAG
072013 110 263 275              STMD R10,=AXIDX     SET AXES INDEX
072016 212
072017 316 377 377              JSB  =STACKX        HOW MANY PARS ?
072022 375 076                  JLN  TWOFOR         JIF MORE THAN ONE
072024 140 223                  CLN  R40            PUSH A DUMMY PAR
072026 012 345                  PUMD R40,+R12
072030                  TWO     BSS  0
072030 316 143 164              JSB  =BGNEND        DEFAULT BEGIN/END
072033                  TWO+    BSS  0
072033 316 377 377              JSB  =ONER          GET TIC SPACING
072036 141 051 242              SIB  R41,R51        SAVE SIGN
072041 202                      ERB  R41            ABSOLUTE IT
072042 204                      LLB  R41
072043 140 010 267              STMD R40,X10,XSPAC  STORE IN PLACE
072046 211 212
072050 265 171 212              LDMD R40,X10,XSTART   FETCH START
072053 151 220                  TSB  R51            CHECK SIGN
072055 376 004                  JRZ  UST            JIF POSITIVE SPACING
072057 140 265 201              LDMD R40,X10,XEND
072062 212
072063                  UST     BSS  0
072063 130 251 042              LDM  R30,=YINDX
072066 000
072067 227                      XRM  R30,R#         FLIP INDEX
072070 140 030 267              STMD R40,X30,YINTR  STORE FOR TIC1
```

```
XAXIS / YAXIS
072073 221 212
072075 130 250 001              LDB   R30,=1          MAJOR TIC COUNT = 1
072100 010 267 231              STMD  R30,X10,XMAJOR
072103 212
072104 316 110 165              JSB   =DFTIC          DEFAULT TIC SIZE
072107 316 377 377              JSB   =ONER           GET Y INTERCEPT
072112 010 267 221              STMD  R#,X10,YINTR
072115 212
072116 104 251 220              GTO   AXMAIN          AXES MAIN ROUTINE
072121 165
072122
072122                  TWOFOR  BSS   0
072122 310 020                  CMB   R#,=20          TWO PARAMETERS ?
072124 367 302                  JZR   TWO             JIF YES
072126 316 377 377              JSB   =TWOR
072131 010 267 201              STMD  R#,X10,XEND
072134 212
072135 150 267 171              STMD  R50,X10,XSTART
072140 212
072141 760 270                  JMP   TWO+
072143
072143                  BGNEN-  BSS   0
072143                  BGNEND  BSS   0
072143 230                      BIN
072144 124 251 140              LDM   R24,=XHI
072147 211
072150 323 127 212              ADMD  R24,=USER       USE CURR BOUND
072153 110 220                  TSB   R10             X AXIS ?
072155 367 004                  JZR   BGNEN1          JIF YES
072157 124 313 020              ADM   R24,=20,0       ELSE PT TO Y LIMIT
072162 000
072163                  BGNEN1  BSS   0
072163 150 024 341              POMD  R50,+R24        GET XMIN (YMIN)
072166 006 345                  PUMD  R50,+R6         SAVE FOR LATER
072170 024 341                  POMD  R50,+R24        GET X MAX (YMAX)
072172 316 266 172              JSB   =USERU          MAP TO USER UNIT
072175 343                      POMD  R#,-R#          POP OFF STACK
072176 010 267 201              STMD  R#,X10,XEND         STORE IN PLACE
072201 212
072202 150 006 343              POMD  R50,-R6         GET XMIN (YMIN)
072205 316 266 172              JSB   =USERU          MAP TO USER UNIT
072210 343                      POMD  R#,-R#          POP OFF STACK
072211 010 267 171              STMD  R#,X10,XSTART       STORE IN PLACE
072214 212
072215 236                      RTN
072216
072216 241                      OCT   241
072217
072217                  GRID.   BSS   0
072217 126 250 007              LDB   R26,=7          GRID,Y & X AXES
072222 760 020                  JMP   AXES.+
072224
072224 241                      OCT   241
```

```
                        XAXIS / YAXIS
072225
072225                  LGRID.  BSS  0
072225 126 250 207              LDB  R26,=207      LABEL GRID
072230 760 012                  JMP  AXES.+
072232
072232 241                      OCT  241
072233
072233                  AXES.   BSS  0
072233 126 250 003              LDB  R26,=3        X & Y AXES
072236 760 004                  JMP  AXES.+
072240
072240 241                      OCT  241
072241
072241                  LAXES.  BSS  0
072241 126 250 203              LDB  R26,=203      LABEL AXES
072244
072244                  AXES.+  BSS  0
072244 262 130 211              STBD R#,=AXFLAG    SET FLAG
072247 316 110 165              JSB  =DFTIC        DEFAULT TIC SIZE
072252 316 377 377              JSB  =STACKX       HOW MANY PARS ?
072255 367 054                  JZR  DEFALL        JIF ZERO
072257 376 012                  JRZ  TFS           JIF NOT 7
072261 316 377 377              JSB  =ONER         ELSE MUST BE 7
072264 263 307 212              STMD R#,=TICSZE    GET TIC SIZE
072267 230                      BIN
072270 126 314 010              BBB  R26,=10       DEC PAR COUNT
072273 206              TFS     LRB  R#
072274 376 027                  JRZ  EFF           JIF 4 PARS
072276 374 062                  JLZ  TES           JIF 2 PARS
072300 316 377 377              JSB  =TWO8         MUST BE 6
072303 146 263 273 YAGN         STMD R46,=YMAJOR   Y MAJOR COUNT
072306 212
072307 766 003                  JNZ  XAGN
072311 211                      ICM
072312 760 367                  JMP  YAGN
072314 156 263 231 XAGN         STMD R56,=XMAJOR   X MAJOR COUNT
072317 212
072320 766 051                  JNZ  TICOK
072322 211                      ICM  R56
072323 760 367                  JMP  XAGN
072325                  EFF     BSS  0
072325 146 223                  CLM  R46           DEFAULT MAJOR COUNT=1
072327 156 223                  CLM  R56
072331 760 350                  JMP  YAGN
072333                  DEFALL  BSS  0
072333 110 223                  CLM  R10
072335 316 117 165              JSB  =GENSPC       DEFAULT X SPACING
072340 345                      PUMD R#,+R#        SAVE XMIN
072341 110 251 042              LDM  R10,=YINDX
072344 200
072345 316 117 165              JSB  =GENSPC
072350 150 006 343              POMD R50,-R6       POP XMIN
072353 012 345                  PUMD R50,+R12      PUSH TWO PARS
```

```
      >>        XAXIS / YAXIS
      072355 140 345                  PUMD R40,+R12
      072357 223                      CLM  R40
3669  072360 760 005                  JMP  TEE+
3670  072362                TEE       BSS  0
3671  072362 140 223                  CLM  R40
3672  072364 012 345                  PUMD R40,+R12
3673  072366 345                      PUMD R40,+R12
3674  072367                TEE+      BSS  0
3675  072367 050 243                  STM  R#,R50
3676  072371 760 310                  JMP  YAGN
3677  072373
3678  072373                TICOK     BSS  0
3679  072373 316 377 377              JSB  =TUOR           GET INTERCEPT PTS
3680  072376 263 221 212              STMD R#,=YINTR       STORE IN PLACE
3681  072401 150 263 263              STMD R50,=XINTR
3681  072404 212
3682  072405 316 377 377              JSB  =TUOR           GET TIC SPACING
3683  072410 102 260 130              LDBD R2,=AXFLAG      LOAD FLAG
3683  072413 211
3684  072414 141 202                  ERB  R41             CHECK SIGN OF Y TIC SPC
3685  072416 371 005                  JEZ  YPOS            JIF Y TIC SPC >= 0
3686  072420 230                      BIN
3687  072421 102 312 020              ADB  R2,=20          ELSE SET FLAG BIT
3688  072424 231                      BCD
3689  072425 151 202        YPOS      ERB  R51             CHECK SIGN OF X TIC SPC
3690  072427 371 005                  JEZ  XPOS            JIF X TIC SPC >= 0
3691  072431 230                      BIN
3692  072432 102 312 010              ADB  R2,=10          ELSE SET FLAG BIT
3693  072435 231                      BCD
3694  072436 102 262 130 XPOS         STBD R2,=AXFLAG      SAVE FLAG
3694  072441 211
3695  072442 151 204                  LLB  R51             ABSOLUTE X TIC SPC
3696  072444 141 204                  LLB  R41             ABSOLUTE Y TIC SPC
3697  072446 140 263 253              STMD R40,=YSPAC      STORE IN PLACE
3697  072451 212
3698  072452 150 263 211              STMD R50,=XSPAC
3698  072455 212
3699  072456 110 223                  CLM  R10             INDEX TO X
3700  072460 316 143 164              JSB  =BGNEN-         SET X BGN, END
3701  072463 110 250 042              LDB  R10,=YINDX      INDEX TO Y
3702  072466 316 143 164              JSB  =BGNEN-
3703  072471 760 126                  JMP  AXMAIN
3704  072473
3705  072473                SWITCH    BSS  0
3706  072473 110 221                  TSM  R10             TEST INDEX
3707  072475 367 010                  JZR  EXSW            JIF INDEX TO X
3708  072477                SWITC+    BSS  0
3709  072477 140 012 345              PUMD R40,+R12
3710  072502 050 241                  LDM  R40,R50
3711  072504 150 012 343              POMD R50,-R12
3712  072507 236            EXSW      RTN
3713  072510
3714  072510                DFTIC     BSS  0
```

```
072510 316 051 147          JSB   =RC,          DEFAULT TIC SIZE
072513                      DRP   !40
072513 263 307 212          STMD  R40,=TICSZE
072516 236                  RTN
072517
072517             GENSPC   BSS   0
072517 316 143 164          JSB   =BGNEN-       SET BEGIN, END
072522 006 345              PUMD  R#,+R6        PUSH MIN VALUE
072524 150 010 265          LDMD  R50,X10,XEND  FETCH MAX VALUE
072527 201 212
072531 231                  BCD
072532 316 377 377          JSB   =SUB10
072535 316 042 147          JSB   =R1,
072540 210                  ICB   R#            GENERATE A 10.0
072541 012 345              PUMD  R#,+R12
072543 316 377 377          JSB   =DIV2         TEN TICS
072546 243                  POMD  R#,-R#        POP RESULT
072547 147 200              ELB   R47
072551 142 223              CLM   R42           TRUNCATE
072553 147 202              ERB   R47           KEEP MS DIGIT
072555 140 345              PUMD  R40,+R#
072557 006 243              POMD  R40,-R6       RECOVER MIN VALUE
072561 236                  RTN
072562
072562             EDGE     BSS   0
072562 140 022 265          LDMD  R40,X22,XH1   GET MIN
072565 140 211
072567 012 345              PUMD  R40,+R12
072571 150 022 265          LDMD  R50,X22,XH2   GET MAX
072574 150 211
072576 231                  BCD
072577 316 377 377          JSB   =SUB10        (MAX-MIN)
072602 223                  CLM   R#            GENERATE A 60.0
072603 210                  ICB   R#
072604 147 250 140          LDB   R47,=140
072607 140 345              PUMD  R40,+R#
072611 316 377 377          JSB   =DIV2         (MAX-MIN)/60
072614 316 377 377          JSB   =SUBROI
072617 343                  POMD  R#,-R#        MIN-(MAX-MIN)/60
072620 236                  RTN
072621
```

AXIS GENERATION ROUTINE

```
     072621
     072621                 AXMAIN  BSS  0
     072621 230                     BIN
     072622 110 223                 CLM  R10            INDEX TO X
     072624 130 260 130             LDBD R30,=AXFLAG    CHECK FLAG
     072627 211
     072630 763 003                 JEV  TRYYAX         JIF NO X AXIS
     072632 316 251 165             JSB  =AXISG-        ELSE DRAW X AXIS
     072635 230             TRYYAX  BIN
     072636 130 260 130             LDBD R30,=AXFLAG    CHECK FLAG
     072641 211
     072642 206                     LRB  R30
     072643 763 242                 JEV  EXSW           JIF NO Y AXIS
     072645 110 251 042             LDM  R10,=YINDX
     072650 000
     072651 110 263 275 AXISG-      STMD R10,=AXIDX     SAVE INDEX TO AXIS
     072654 212
     072655 126 010 241             LDM  R26,R10        COPY INDEX
     072660 313 171 212             ADM  R26,=XSTART    PT TO START
     072663 316 326 145             JSB  =ORDER2        ORDER START END
     072666 140 010 265             LDMD R40,X10,XSPAC     GET TIC SPACE
     072671 211 212
     072673 766 010                 JNZ  SET1ST         JIF TIC TO DRAW
     072675 265 201 212             LDMD R40,X10,XEND      ELSE SET TO END
     072700 263 277 212             STMD R40,=AXTIC1
     072703 760 043                 JMP  GOGEN
     072705 136 010 265 SET1ST      LDMD R36,X10,XMAJOR    GET MAJOR TIC COUNT
     072710 231 212
     072712 316 171 166             JSB  =CMPAX1        COMPUTE AXTIC1
     072715 136 223                 CLM  R36            MINOR TIC COUNT=1
     072717 210                     ICB  R36
     072720 316 171 166             JSB  =CMPAX1        COMPUTE AXTIC1
     072723 263 277 212             STMD R#,=AXTIC1     STORE IT
     072726 316 377 377             JSB  =SUBROI        DIFFERENCE
     072731 010 265 211             LDMD R#,X10,XSPAC GET SPACING
     072734 212
     072735 012 345                 PUMD R#,+R12
     072737 316 377 377             JSB  =DIV2          DIVIDE TO GET COUNT
     072742 316 377 377             JSB  =ONEB
     072745 263 317 212             STMD R#,=TICNT      SET TIC COUNT
     072750 140 010 265 GOGEN       LDMD R40,X10,XSTART    GET START
     072753 171 212
     072755 150 265 221             LDMD R50,X10,YINTR     GET Y INTERCEPT
     072760 212
     072761 316 073 165             JSB  =SWITCH
     072764 140 012 345             PUMD R40,+R12       PUSH PAR FOR MOVE
     072767 150 345                 PUMD R50,+R12
     072771 316 234 151             JSB  =MOVE.         MOVE TO 1ST PT
     072774 150 223                 CLM  R50
     072776 760 011                 JMP  CKOUTB
     073000 110 261 275 AXLP        LDMD R10,=AXIDX
     073003 212
     073004 150 010 265             LDMD R50,X10,XSPAC     GET SPACING
```

```
073007 211 212
073011 140 261 277 CKOUTB  LDMD R40,=AXTIC1   FETCH CURRENT PT
073014 212
073015 316 377 377         JSB  =4DD10        CALC NEXT PT
073020 343                 POMD R#,-R#        POP OFF STACK
073021 345                 PUMD R#,+R#        SAVE A COPY
073022 150 010 265         LDMD R50,X10,XEND  GET XEND
073025 261 212
073027 316 375 144         JSB  =COMFLT       DO FLEXIBLE COMPARE
073032                     DRP  !40
073032                     ARP  !12
073032 343                 POMD R40,-R12      POP NXT TIC FROM STACK
073033 263 277 212         STMD R40,=AXTIC1   STORE IN PLACE
073036 371 075             JEZ  OFFAX         JIF XEND < NEXT TIC
073040 316 150 166         JSB  =AXDSUB       DRAW AXIS
073043 110 261 275         LDMD R10,=AXIDX    RESTORE INDEX
073046 212
073047 230                 BIN
073050 140 010 265         LDMD R40,X10,XSPAC    GET SPACING
073053 211 212
073055 267 065             JZR  EXMAIN        JIF NO TICS
073057 260 130 211         LDBD R40,=AXFLAG   GET FLAG
073062 206                 LRB  R40
073063 206                 LRB  R40           SHIFT TO GRID
073064 130 261 317         LDMD R30,=TICNT    GET TIC COUNT
073067 212
073070 267 011             JZR  MAJORT        JIF MAJOR TIC
073072 213                 DCM  R30
073073 263 317 212         STMD R30,=TICNT    DEC TIC COUNT
073076 316 376 166         JSB  =TICMIN       DRAW MINOR TIC
073101 260 275             JMP  AXLP          DO AXIS LOOP
073103 010 265 231 MAJORT  LDMD R#,X10,XMAJOR
073106 212
073107 213                 DCM  R#
073110 263 317 212         STMD R#,=TICNT     REINIT TIC COUNT
073113 316 351 166         JSB  =TICMAJ       DO MAJOR TIC
073116 110 261 275         LDMD R10,=AXIDX
073121 212
073122 130 260 130         LDBD R30,=AXFLAG   CHECK FOR LABELS
073125 211
073126 265 250             JPS  AXLP          JIF NO LABEL
073130 316 143 167         JSB  =LAX          ELSE DO LABEL
073133 260 243             JMP  AXLP
073135
073135             OFFAX   BSS  0
073135                     DRP  !40
073135 010 265 201         LDMD R#,X10,XEND   DRAW TO END
073140 212
073141 316 150 166         JSB  =AXDSUB       DRAW AXIS
073144 316 357 150 EXMAIN  JSB  =PENUP.       LIST PEN UP
073147 236                 RTN
073150
073150             AXDSUB  BSS  0
```

```
                         AXIS GENERATION ROUTINE
073150 150 010 265            LDMD R50,X10,YINTP    GET Y COOR
073153 221 212
073155 316 073 165            JSB  =SWITCH          SWITCH IF Y AXIS
073160 140 012 345            PUMD R40,+R12         PUSH X COOR
073163 150 345                PUMD R50,+R12         PUSH Y COOR
073165 316 327 151            JSB  =DRAW.           DRAW THE LINE
073170 236                    RTN
073171
073171
```

```
073171
073171                   YINDX   EQU  42
073171
073171
073171                   *   J = O-RMD(O-S,T)+T*(O>S)
073171                   *   AXTIC1 = J+T*(O>J)-T*(J=O+T)
073171
073171                   CMPAX1  BSS  0
073171 316 377 377               JSB  =CONBIN         CONVERT TO REAL
073174 150 010 265               LDMD R50,X10,XSPAC      GET TIC SPACING
073177 211 212
073201 316 377 377               JSB  =MPY10          SPACONG BETWEEN MAJOR TICS
073204 343                       POMD R#,-R#          POP T OFF STACK
073205 263 072 211               STMD R#,=TMP2
073210 130 251 042               LDM  R30,=YINDX    FLIP R10 IN R30
073213 000
073214 010 227                   XRM  R30,R10
073216 140 030 265               LDMD R40,X30,YINTR      GET S
073221 221 212
073223 263 062 211               STMD R40,=TMP1          STORE IN PLACE
073226 010 265 171               LDMD R40,X10,XSTART     GET O
073231 212
073232 263 102 211               STMD R40,=TMP3          STORE IN PLACE
073235 012 345                   PUMD R40,+R12           ALSO PUSH A COPY
073237 316 331 166               JSB  =O-S^T^            PUSH O-S, PUSH T
073242 316 377 377               JSB  =REM10             RMD(O-S,T)
073245 316 377 377               JSB  =SUBR01            O-RMD(O-S,T)
073250 316 300 166               JSB  =O>S+T             O-RMD(O-S,T)+(O>S)*T
073253 343                       POMD R#,-R#        POP J
073254 345                       PUMD R#,+R#        KEEP A COPY
073255 263 062 211               STMD R#,=TMP1      S BECOMES J
073260 316 331 166               JSB  =O-S^T^
073263 316 377 377               JSB  =ADDR01
073266 343                       POMD R#,-R#
073267 366 007                   JNZ  NOT2FR        JIF NOT TOO FAR
073271 261 072 211               LDMD R#,=TMP2
073274 345                       PUMD R#,+R#        STACK T
073275 316 377 377               JSB  =SUBR01
073300                   O>S+T   BSS  0
073300                   NOT2FR  BSS  0
073300                           DRP  !40
073300                           ARP  !12
073300 261 102 211               LDMD R40,=TMP3     O
073303 345                       PUMD R40,+R12
073304 261 062 211               LDMD R40,=TMP1     S OR J
073307 345                       PUMD R40,+R12
073310 316 377 377               JSB  =ROMJSB
073313 377 377                   DEF  GR.
073315 000                       OCT  0
073316 261 072 211               LDMD R#,=TMP2      T
073321 345                       PUMD R#,+R#
073322 316 377 377               JSB  =MPYR01
073325 316 377 377               JSB  =ADDR01
```

```
073330 236                      RTN
073331
073331              O-S*T      BSS  0
073331 150 261 102              LDMD R50,=TMP3    GET O
073334 211
073335 140 261 062              LDMD R40,=TMP1    GET S
073340 211
073341 316 377 377              JSB  =SUB10       O-S
073344 261 072 211              LDMD R#,=TMP2     GET T
073347 345                      PUMD R#,+R#
073350 236                      RTN
073351
073351 140 220      TICMAJ      TSB  R40          GRID ?
073353 763 013                  JEV  GEN2         JIF NOT
073355 223                      CLM  R40          GENERATE A LARGE #
073356 250 231                  LDB  R40,=99C     1.0*10^99 GDU
073360 147 250 020              LDB  R47,=10C
073363 140 012 345              PUMD R40,+R12
073366 760 026                  JMP  PUSHR#
073370 223          GEN2        CLM  R#
073371 147 250 040              LDB  R47,=40      GENERATE A 2
073374 760 010                  JMP  TICCOM
073376 140 220      TICMIN      TSB  R40
073400 362 366                  JOD  GEN2         JIF GRID
073402 223                      CLM  R40
073403 147 250 100              LDB  R47,=100     GENERATE A 4
073406 150 261 307  TICCOM      LDMD R50,=TICSZE  GET TIC SIZE
073411 212
073412 231                      BCD
073413 316 377 377              JSB  =DIV10       RPLOT UP HALFWAY
073416 343          PUSHR#      POMD R#,-R#       GET HALF TIC LENGTH
073417 345                      PUMD R#,+R#       SAVE A COPY
073420 345                      PUMD R#,+R#       ONE MORE
073421 316 030 167              JSB  =TIC         DRAW TICS
073424 231                      BCD
073425 316 377 377              JSB  =CHSROI      GO OPPOSITE DIRECTION
073430 110 261 275  TIC         LDMD R10,=AXIDX
073433 212
073434 150 012 343              POMD R50,-R12     GET CURRENT Y
073437 140 223                  CLM  R40          X=0
073441 316 073 165              JSB  =SWITCH
073444 140 263 062              STMD R40,=TMP1    STORE DELTA X
073447 211
073450 150 263 072              STMD R50,=TMP2    STORE DELTA Y
073453 211
073454 122 251 242              LDM  R22,=XAGU    SET GDU SCALING
073457 210
073460 124 251 062              LDM  R24,=TMP1
073463 211
073464 026 243                  STM  R24,R26
073466 316 143 151              JSB  =MAPIC2      MAP TO NU
073471 316 012 162              JSB  =ADDPEN      RPLOT WITHOUT ROTATION
073474 124 251 002              LDM  R24,=XPEN    STARTING ADDR = PEN ADDR
```

```
073477 211
073500 316 042 152            JSB   =TOTMP3
073503 130 251 377            LDM   R30,=377,377 PEN CONTROL = -1
073506 377
073507 263 131 211            STMD  R30,=PCNTRL
073512 316 347 151            JSB   =DRAW.3      PEN CONTROL ALREADY -1
073515                 BACKAX BSS   0
073515 124 251 002            LDM   R24,=XPEN    MOVE BACK TO ON AXIS
073520 211
073521 126 251 062            LDM   R26,=TMP1
073524 211
073525 316 343 144            JSB   =MOV20
073530                 MOVENU BSS   0            MOVE WITHOUT SCALING
073530 130 251 376            LDM   R30,=376,377 PEN CONTROL = -2
073533 377
073534 263 131 211            STMD  R30,=PCNTRL
073537 316 341 151            JSB   =DRAW.2      MOVE BACK TO ON AXIS
073542 236                    RTN
073543
073543                 LAX    BSS   0
073543 140 261 002            LDMD  R40,=XPEN    SAVE PEN ADDR
073546 211
073547 366 345                PUMD  R40,+R6
073551 261 012 211            LDMD  R40,=YPEN
073554 345                    PUMD  R40,+R6
073555 110 221                TSM   R10          IS IT X AXIS ?
073557 366 025                JNZ   LAY          JIF NOT
073561 122 261 127            LDMD  R22,=USER    MOVE TO BELOW Y MIN
073564 212
073565 313 020 000            ADM   R22,=20,0
073570 316 162 165            JSB   =EDGE        MOVE TO YMIN-
073573 263 072 211            STMD  R#,=TMP2
073576 261 002 211            LDMD  R#,=XPEN     X COOR = XPEN
073601 263 062 211            STMD  R#,=TMP1
073604 360 020                JMP   LMOVE        GO MOVE PEN
073606                 LAY    BSS   0
073606 122 261 127            LDMD  R22,=USER    INDEX TO X SCALE
073611 212
073612 316 162 165            JSB   =EDGE        MOVE TO XMIN-
073615 263 062 211            STMD  R#,=TMP1
073620 261 012 211            LDMD  R#,=YPEN     Y COOR = YPEN
073623 263 072 211            STMD  R#,=TMP2
073626                 LMOVE  BSS   0
073626 230                    BIN
073627 316 130 167            JSB   =MOVENU      MOVE PEN TO POSITION
073632 140 261 277            LDMD  R40,=AXTIC1  GET TIC COORDINATE
073635 212
073636 170 223                CLM   R70
073640 174 250 010            LDB   R74,=10      0 LEFT, 8 RIGHT
073643 231                    BCD
073644 316 377 377            JSB   =SEP10       SEPERATE
073647 316 377 377            JSB   =ROMJSB      ROUND
073652 377 377                DEF   ROUND
```

```
073654  00                          OCT   0
073655  155 251 002                 LDM   R55,=2,1,56
073660  101 056
073662  070 243                     STM   R55,R70
073664  154 223                     CLM   R54
073666  250 011                     LDB   R54,=9D         9 CHARS
073670  120 261 275                 LDMD  R20,=AXIDX      CHECK INDEX
073673  212
073674  766 005                     JNZ   LDNY            JIF Y AXIS
073676  260 322 212                 LDBD  R20,=LAXN       GET N FOR X
073701  760 003                     JMP   LAXNNN
073703                     LDNY     BSS   0
073703                              DRP   !20
073703  260 321 212                 LDBD  R20,=LAXNY      GET N FOR Y
073706                     LAXNNN   BSS   0
073706                              DRP   !20
073706  074 242                     STB   R20,R74         SAVE N
073710  036 303                     ADM   R20,R36         EXP + N
073712  375 022                     JLN   EFMT            JIF TOTAL NEGATIVE
073714  311 007 000                 CMM   R20,=7,0        JIF TOO MANY DIGIT LEFT
073717  373 015                     JCY   EFMT
073721  176 250 007                 LDB   R76,=7
073724  074 304                     SBB   R76,R74         CHARS LEFT
073726  174 220                     TSB   R74             0 CHAR RIGHT ?
073730  766 020                     JNZ   FORMNC          JIF NOT
073732  171 222                     CLB   R71
073734  760 010                     JMP   CL72
073736                     EFMT     BSS   0
073736  173 210                     ICB   R73             E FORMAT
073740  174 223                     CLM   R74
073742  176 210                     ICB   R76
073744  154 212                     DCB   R54
073746                     CL72     BSS   0
073746  154 212                     DCB   R54
073750  172 222                     CLB   R72
073752                     FORMNC   BSS   0
073752  140 261 277                 LDMD  R40,=AXTIC1     GET TIC COORDINATE
073755  212
073756  316 377 377                 JSB   =ROMJSB         FORMAT IT
073761  377 377                     DEF   FORMG           PTR2 PTS TO FORMATTED STRING
073763  000                         OCT   0
073764  156 054 241                 LDM   R56,R54         GET # OF CHARS
073767  230                         BIN
073770                     POPLP    BSS   0
073770  154 270 315                 LDBI  R54,=PTR2-      POP ONE CHAR
073773  377
073774  310 040                     CMB   R54,=BLANK      IS IT A BLANK ?
073776  766 004                     JNZ   GCH             JIF NOT
074000  156 212                     DCB   R56             DEC BYTE COUNT
074002  760 364                     JMP   POPLP
074004                     GCH      BSS   0
074004  272 316 377                 STBI  R#,=PTR2+       PUSH THE CHAR BACK
074007  156 263 417                 STMD  R56,=LABLNG     SAVE LABEL LENGTH
```

```
                              AXIS UTILITIES
074012 212
074013 176 261 025         LDMD R76,=XLORG   SAVE LORG
074016 212
074017 106 345             PUMD R76,+R6
074021 140 261 046         LDMD R40,=SINLD   SAVE LDIR
074024 212
074025 345                 PUMD R40,+R6
074026 261 056 212         LDMD R40,=COSLD
074031 345                 PUMD R40,+R6
074032
074032                *****************************************************************
074032                *                                                               *
074032                *   AXIS  SPACING  LDIR   LORG     LABEL                        *
074032                *                                                               *
074032                *    X      >0      90     8    PERPENDICULAR                   *
074032                *    X      <0       0     6    PARALLEL                        *
074032                *    Y      >0       0     8    PERPENDICULAR                   *
074032                *    Y      <0      90     4    PARALLEL                        *
074032                *                                                               *
074032                *****************************************************************
074032
074032 176 251 010               LDM  R76,=8,0      SET LORG 8
074035 000
074036 316 042 147               JSB  =R1.          SET LDIR 90
074041 150 223                   CLM  R50
074043 120 260 130               LDBD R20,=AXFLAG   GET AXFLAG
074046 211
074047 206                       LRB  R20           GET NEG X SPACE FLG
074050 206                       LRB  R20
074051 206                       LRB  R20
074052 110 261 275               LDMD R10,=AXIDX    GET INDEX
074055 212
074056 367 005                   JZR  XL            JIF X AXIS
074060 316 077 165               JSB  =SWITC+       SET LDIR 0
074063 120 206                   LRB  R20           GET NEG Y SPACE FLG
074065                     XL    BSS  0
074065 120 220                   TSB  R20           CHECK FLAG
074067 363 015                   JEV  EVEN          JIF SPACE > 0
074071 316 077 165               JSB  =SWITC+       ROTATE 90
074074 176 212                   DCB  R76           LORG 6
074076 212                       DCB  R76
074077 110 221                   TSM  R10           X AXIS ?
074101 367 003                   JZR  EVEN          JIF YES
074103 176 212                   DCB  R76           LORG 4
074105 212                       DCB  R76
074106                     EVEN  BSS  0
074106 316 154 170               JSB  =ORGDIR       SET LORG, LDIR
074111 130 251 024               LDM  R30,=24,0     GO LABEL IT
074114 000
074115 316 125 145               JSB  =UFLTDR
074120 230                       BIN
074121 150 006 343               POMD R50,-R6       RESTORE LDIR
074124 140 343                   POMD R40,-R6
```

```
          AXIS UTILITIES
     074126 316 154 170          JSB  =ORGDIR      PSEUDO LORG
     074131 176 056 343          POMD R76,-R6      RESTORE LORG
4081 074134 263 025 212          STMD R76,=XLORG
4082 074137 140 343              POMD R40,-R6      RESTORE PEN ADDR
4083 074141 263 072 211          STMD R40,=TMP2
4084 074144 343                  POMD R40,-R6
4085 074145 263 062 211          STMD R40,=TMP1
4086 074150 316 130 167          JSB  =MOVEMU      MOVE PEN BACK TO AXIS
4087 074153 236                  RTN
4088 074154
4089 074154              ORGDIR  BSS  0
4090 074154 316 105 150          JSB  =LORG2       SET LORG
4091 074157 140 263 046          STMD R40,=SINLD   SET LDIR
4091 074162 212
4092 074163 150 263 056          STMD R50,=COSLD
4092 074166 212
4093 074167 130 251 022          LDM  R30,=22,0
4093 074172 100
4094 074173 316 125 145          JSB  =UFLTDR
4095 074176 230                  BIN
4096 074177 236                  RTN
4097 074200
```

```
074200
074200 241                     OCT  241
074201
074201            BLOFF.  BSS  0
074201            BLINK.  BSS  0
074201 236                RTN
074202
074202            ****************************************************************
074202            *
074202            *      BPLOT RUN TIME ROUTINE                                   *
074202            *                                                               *
074202            ****************************************************************
074202
074202 241                OCT  241              PRIMARY ATTRIBUTE
074203
074203            BPLOT.  BSS  0
074203 316 336 171        JSB  =COMN
074206 367 371            JZR  BLINK.           JIF ZERO LENGTH STRING
074210 316 377 377        JSB  =GRAPH           FORCE TO GRAPH
074213            NXTB    BSS  0                OUTER LOOP
074213 176 261 062        LDMD R76,=TMP1        INIT FOR INNER LOOP
074216 211
074217 263 072 211        STMD R76,=TMP2
074222 316 032 154        JSB  =BYTEG           TELL CRT
074225 133 222            CLB  R33              RESET FOR INIT CHAR
074227 146 044 241        LDM  R46,R44          INNER LOOP COUNT
074232 765 011            JPS  NXTB2            JIF POSITIVE COUNT
074234 316 377 377        JSB  =INCHR           INPUT CURRENT CHAR
074237 316 032 154        JSB  =BYTEG           BACK TO CURRENT ADDR
074242            NXTB1   BSS  0
074242 316 040 172        JSB  =SHIFTL          SHIFT TO R33
074245            NXTB2   BSS  0
074245 133 006 344        PUBD R33,+R6          SAVE INIT BYTE
074250            NXTBPL  BSS  0                INNER LOOP
074250 316 377 377        JSB  =INCHR           INPUT CURRENT BYTE
074253 176 261 072        LDMD R76,=TMP2        GET CURRENT PLOT ADDR
074256 211
074257 321 342 211        CMMD R76,=GSIZE       OUT OF BOUND ?
074262 373 131            JCY  UNW2             JIF YES
074264 211                ICM  R76              INC FOR NEXT
074265 263 072 211        STMD R76,=TMP2        SAVE IT
074270 213                DCM  R76              RESTORE CURRENT ADDR
074271 316 032 154        JSB  =BYTEG           TELL CRT
074274
074274 137 006 342        POBD R37,-R6          POP PREVIOUS BYTE
074277 136 270 315        LDBI R36,=PTR2-       GET CURRENT BYTE
074302 377
074303 344                PUBD R36,+R6          SAVE FOR NEXT TIME
074304 316 313 171        JSB  =GPBYTE          BPLOT ONE BYTE
074307 122 213            DCM  R22              DEC STRING COUNT
074311 767 106            JZR  ENDBPL           JIF DONE
074313            CONTU   BSS  0
074313 146 221            TSM  R46              INNER LOOP COUNT > 0
```

```
                       BPLOT
074315 365 003                  JPS   CONTU1          JIF YES
074317 211                      ICM   R46             ELSE INC INNER LOOP CNT
074320 760 001                  JMP   CONTU2
074322                 CONTU1   BSS   0
074322                          DRP   !46
074322 213                      DCM   R46             DEC INNER LOOP COUNT
074323                 CONTU2   BSS   0
074323 366 323                  JNZ   NXTBPL          JIF MORE TO PLOT
074325 316 346 170              JSB   =BPLAST         ELSE BPLOT LAST CHAR
074330 176 261 062              LDMD  R76,=TMP1       AND
074333 211
074334 323 340 211              ADMD  R76,=GLINE      MOVE DOWN ONE LINE
074337 263 062 211              STMD  R76,=TMP1
074342 126 213                  DCM   R26             DEC Y COOR
074344 760 245                  JMP   NXTB            LOOP BACK
074346
074346
074346                 BPLAST   BSS   0               BPLOT LAST CHAR
074346 316 377 377              JSB   =INCHR          INPUT CURRENT CHAR
074351 102 006 343              POMD  R2,-R6           GET RTN ADDR
074354 137 342                  POBD  R37,-R6          GET LAST CHAR
074356 176 261 072              LDMD  R76,=TMP2       GET CURRENT PLOT ADDR
074361 211
074362 321 342 211              CMMD  R76,=GSIZE      OUT OF BOUND ?
074365 373 056                  JCY   UNW1            JIF YES
074367 316 032 154              JSB   =BYTEG          RESTORE CURRENT ADDR
074372                          ARP   !6
074372 102 345                  PUMD  R2,+R6          PUSH RTN ADDR BACK
074374 136 222                  CLB   R36             RESET CURRENT CHAR
074376 146 044 241              LDM   R46,R44         REPEAT COUNT > 0
074401 365 006                  JPS   BPLAS1          JIF YES
074403                 ******   JSB   =INCHR          ELSE READ CURRENT CHAR
074403                 ******   LDM   R76,=TMP2       RESTORE CURR BYTE ADDR
074403                 ******   JSB   =BYTEG
074403 316 040 172              JSB   =SHIFTL         SHIFT TO THE LEFT
074406 136 032 240              LDB   R36,R32         LOAD IT
074411                 BPLAS1   BSS   0
074411 316 313 171              JSB   =BPBYTE         BPLOT ONE BYTE
074414 236                      RTN
074415
074415                 UNW2     BSS   0
074415 166 342                  POBD  R#,-R6          CLEAN STACK
074417 760 024                  JMP   UNW1            FINISH UP
074421
074421                 ENDBPL   BSS   0
074421 316 346 170              JSB   =BPLAST         BPLOT LAST CHAR
074424 176 261 062              LDMD  R76,=TMP1       GET LAST ADDR
074427 211
074430 323 340 211              ADMD  R76,=GLINE      MOVE DOWN ONE LINE
074433 321 342 211              CMMD  R76,=GSIZE      EXCEED GRAPH AREA
074436 373 005                  JCY   UNW1            JIF YES
074440 263 062 211              STMD  R76,=TMP1
074443 126 213                  DCM   R26
```

```
074445              UNM'    BSS  0
074445 136 026 241          LDM  R36,R26
074450 316 377 377          JSB  =CONBIN      CONVERT TO REAL
074453 140 263 333          STMD R40,=YMAP    STORE BACK
074456 212
074457 236                  RTN
074460
074460              CLRSTX  BSS  0
074460 112 261 344          LDMD R12,=TOS     CLEAN UP STACK
074463 203
074464 360 357              JMP  UN#1
074466
074466
074466              ******************************************************
074466              *                                                    *
074466              *       BREAD                                        *
074466              *                                                    *
074466              ******************************************************
074466
074466 241                  OCT  241
074467
074467              GREAD.  BSS  0
074467 316 336 171          JSB  =COMN        DO COMMON PART
074472 367 364              JZR  CLRSTX       JIF ZERO LENGTH
074474 146 221              TSM  R46          JIF REPEAT COUNT=0
074476 367 360              JZR  CLRSTX
074500 365 003              JPS  GREAD1       JIF REPEAT COUNT>0
074502 215                  TCM  R46
074503 044 243              STM  R46,R44
074505              GREAD1  BSS  0
074505 165 012 343          POMD R65,-R12     GET 1ST CHAR ADDR
074510 263 314 377          STMD R65,=PTR2    SET UP POINTER
074513 316 377 377          JSB  =GRAPH       FORCE TO GRAPH
074516                      DRP  !30
074516 271 314 377          LDMI R30,=PTR2    GET ACT LENGTH OF DEST
074521 311 377 377          CMM  R30,=377,377 NULL ?
074524 366 001              JNZ  LENSET       JIF NOT
074526 211                  ICM  R30          MAKE IT ZERO
074527              LENSET  BSS  0
074527 165 075 305          SBM  R65,R75      OFFSET FROM 1ST CHAR
074532 120 065 241          LDM  R20,R65      USE 2 BYTES ONLY
074535 215                  TCM  R20
074536 032 243              STM  R20,R32      SAVE IT
074540 022 303              ADM  R20,R22      OFFSET OF THE END OF THE SINK
074542 030 301              CMM  R20,R30      > OLD ACT LENGTH ?
074544 372 003              JNC  LENOK        JIF NOT
074546 273 314 377          STMI R20,=PTR2    UPDATE ACT LENGTH
074551              LENOK   BSS  0
074551 130 032 301          CMM  R30,R32      ACT LEN < OFFSET OF SINK
074554 373 032              JCY  OUTER        JIF NO
074556 065 243              STM  R30,R65
074560 167 222              CLB  R67
074562 165 325 314          SBMD R65,=PTR2    MOVE TO THE END OF OLD
```

```
                 BPLOT
074565 377
074566 215                        TCM  R65
074567 263 314 377                STMD R65,=PTR2
074572 130 032 705                SBM  R30,R32       BLANK FILL TO THE SINK
074575 132 250 040                LDB  R32,=BLANK
074600                   BLKFIL   BSS  0
074600 132 272 315                STBI R32,=PTR2-    BLANK ONE BYTE
074603 377
074604 130 211                    ICM  R30           INC COUNT
074606 766 370                    JNZ  BLKFIL        JIF MORE
074610                   OUTER    BSS  0             OUTER LOOP FOR BREAD
074610 176 261 062                LDMD R76,=TMP1
074613 211
074614 263 072 211                STMD R76,=TMP2     INIT FOR INNER LOOP
074617 316 032 154                JSB  =BYTEG        TELL CRT
074622 146 044 241                LDM  R46,R44       GET REPEAT COUNT
074625                   INNER    BSS  0
074625 316 377 377                JSB  =INCHR        READ ONE BYTE
074630                            DRP  !32
074630 033 242                    STB  R32,R33       SAVE IT
074632 316 377 377                JSB  =INCHR        READ NEXT BYTE
074635 176 261 072                LDMD R76,=TMP2     GET ADDR
074640 211
074641 321 342 211                CMMD R76,=GSIZE    OUT OF BOUND ?
074644 373 212           CLRST1   JCY  CLRSTX        JIF YES
074646 176 211                    ICM  R76           INCR ADDR
074650 263 072 211                STMD R76,=TMP2     SAVE FOR NEXT TIME
074653 316 032 154                JSB  =BYTEG        ALSO TELL CRT
074656 316 040 172                JSB  =SHIFTL       SHIFT TO POSITION
074661 133 272 315                STBI R33,=PTR2-    STORE IN PLACE
074664 377
074665 122 213                    DCM  R22           DEC STRING COUNT
074667 367 353                    JZR  CLRST1        JIF DONE
074671 146 213                    DCM  R46           DEC REPEAT COUNT
074673 766 730                    JNZ  INNER         JIF MORE
074675 176 261 062                LDMD R76,=TMP1
074700 211
074701 323 340 211                ADMD R76,=GLINE    MOVE DOWN ONE LINE
074704 263 062 211                STMD R76,=TMP1     SAVE FOR NEXT
074707 126 213                    DCM  R26           DEC Y-COOR
074711 760 275                    JMP  OUTER         JMP TO OUTER LOOP
074713
074713                   SPBYTE   BSS  0
074713 316 023 172                JSB  =SHIFTR       SHIFT PREV BYTE IN
074716                   SPBYT2   BSS  0
074716 146 221                    TSM  R46           REPEAT COUNT > 0
074720 765 005                    JPS  BPBYT3        JIF YES
074722 132 036 240                LDB  R32,R36       ELSE ABS BPLOT
074725 760 003                    JMP  BPBYT4
074727                   SPBYT3   BSS  0
074727 132 036 226                XRB  R32,R36       XOR IN DATA
074732                   SPBYT4   BSS  0
074732 316 037 154                JSB  =OUTBYT       OUTPUT TO CRT
```

```
074735 236                      RTN
074736
074736                 COMN     BSS  0
074736 230                      BIN
074737 160 261 323              LDMD R60,=XMAP      GET X-COOR OF PEN
074742 212
074743 012 345                  PUMD R60,+R12
074745 316 377 377              JSB  =ONEB          CONVERT TO BINARY
074750                          DRP  !76
074750 020 243                  STM  R76,R20        SAVE IT
074752 160 261 333              LDMD R60,=YMAP      GET Y-COOR OF PEN
074755 212
074756 012 345                  PUMD R60,+R12
074760 316 377 377              JSB  =ONEB          CONVERT TO BINARY
074763                          DRP  !76
074763 026 243                  STM  R76,R26        SAVE IT
074765 315 357 000              SBM  R76,=239D,0    Y'=239-Y
074770 215                      TCM  R76
074771 022 243                  STM  R76,R22        SAVE IT
074773 316 113 154              JSB  =MAPABS        MAP TO ABS CRT ADDR
074776 176 263 062              STMD R76,=TMP1      SAVE BYTE ADDR
075001 211
075002 175 070 243              STM  R75,R70        SAVE BIT ADDR
075005 316 377 377              JSB  =ONEB          GET REPEAT COUNT
075010                          DRP  !76
075010 044 243                  STM  R76,R44        SAVE IT
075012 175 012 343              POMD R75,-R12       GET STRING ADDRESS
075015 263 314 377              STMD R75,=PTR2
075020 122 343                  POMD R22,-R12       GET STRING LENGTH
075022 236                      RTN
075023
075023
075023                 *     R36=CURRENT BYTE
075023                 *     R37=PREVIOUS BYTE
075023                 *     R70=BIT ADDRESS
075023
075023                 SHIFTR   BSS  0              SHIFT TO BIT ADDR
075023 171 070 240              LDB  R71,R70        COPY BIT ADDR
075026 367 007                  JZR  SHIFT2         JIF AT BOUNDARY
075030                 SHIFT1   BSS  0
075030 137 207                  LRM  R37            SHIFT ONE BIT POS(
075032 171 314 040              SBB  R71,=40        DEC BIT ADDR
075035 366 371                  JNZ  SHIFT1         JIF MORE TO SHIFT
075037 236             SHIFT2   RTN
075040
075040                 *      R32=CURRENT BYTE
075040
075040                 SHIFTL   BSS  0
075040 171 070 240              LDB  R71,R70        COPY BIT ADDR
075043 367 372                  JZR  SHIFT2         JIF AT BOUNDARY
075045                 SHIFT3   BSS  0
075045 132 205                  LLM  R32            SHIFT ONE BIT POS(
075047 171 314 040              SBB  R71,=40        DEC BIT ADDR
```

```
              BPLOT
+   073602 266 371                JMZ  SHIFT3
+   075054 236                    RTN
```

```
075055
075055
075055                       ************************************************************
075055                       *
075055                       *      PDIR RUN TIME ROUTINE
075055                       *
075055                       ************************************************************
075055 241                              OCT  241
075056
075056                       PDIR.      BSS  0
075056 114 251 066                      LDM  R14,=SINPD     PT TO SINPD
075061 212
075062                       PDIR1      BSS  0
075062 316 377 377                      JSB  =STACK%        HOW MANY PARS ?
075065 375 022                          JLN  PDIR3          JIF 2
075067                       PDIR2      BSS  0
075067 140 012 343                      POMD R40,-R12       GET PAR
075072 345                              PUMD R40,+R12       SAVE A COPY
075073 345                              PUMD R40,+R12       SAVE ONE MORE
075074 316 377 377                      JSB  =SIN10         GET SIN(LDIR)
075077 343                              POMD R#,-R#         POP OFF STK
075100 014 345                          PUMD R#,+R14        SAVE IN PLACE
075102 316 377 377                      JSB  =COS10         GET COS(LDIR)
075105 343                              POMD R#,-R#         POP OFF STK
075106 014 345                          PUMD R#,+R14        SAVE IN PLACE
075110 236                              RTN
075111
075111                       PDIR3      BSS  0
075111 316 127 151                      JSB  =MAPIC         GET 2 PAR & SCALE THEM
075114                                  DRP  !40
075114                                  ARP  !26
075114 343                              POMD R40,-R26       SWITCH PARS
075115 012 345                          PUMD R40,+R12
075117 026 343                          POMD R40,-R26
075121 012 345                          PUMD R40,+R12
075123 316 377 377                      JSB  =ROMJSB        GET THETA
075126 377 377                          DEF  ATN2.
075130 000                              OCT  0
075131 260 334                          JMP  PDIR2          DO THE REST
075133
```

DIGITIZE / CURSOR / WHERE

```
075133
075133 241                      OCT   241
075134
075134                 DIGIT    BSS   0
075134 316 105 145              JSB   =DONEP+
075137 370 011                  JEN   CURSR1
075141 236                      RTN
075142
075142 241                      OCT   241
075143
075143                 CURSR.   BSS   0
075143 130 251 016              LDM   R30,=16,0
075146 000
075147 316 125 145              JSB   =UPLTDR
075152                 CURSR1   BSS   0
075152 124 251 042              LDM   R24,=XCURS     GET CURSOR ADDR
075155 211
075156 260 005                  JMP   SEWH1
075160
075160 241                      OCT   241
075161
075161                 WHERE.   BSS   0
075161 124 251 002              LDM   R24,=XPEN      GET PEN LOCATION
075164 211
075165                 SEWH1    BSS   0
075165 126 251 062              LDM   R26,=TMP1      MOVE TO BFR
075170 211
075171 316 343 144              JSB   =MOV20
075174                 TRANV    BSS   0
075174 316 377 377              JSB   =STACKX        HOW MANY PARS ?
075177 310 016                  CMB   R#,=14D        JIF TWO SIMPLE
075201 367 030                  JZR   NOPENT
075203 310 023                  CMB   R#,=19D
075205 367 024                  JZR   NOPENT         JIF 2 WITH 1 ARAY ELEM
075207 310 030                  CMB   R#,=24D
075211 367 020                  JZR   NOPENT         JIF 2 & BOTH ARAY ELEM
075213 140 223                  CLM   R40
075215 147 260 126              LDBD  R47,=PENDN     GET PEN DOWN FLG
075220 211
075221 367 002                  JZR   CALSTO         JIF PEN UP
075223 250 020                  LDB   R47,=10C       ELSE SET TO 1.0
075225                 CALSTO   BSS   0
075225 140 012 345              PUMD  R40,+R12       PUSH ON STACK
075230 316 377 377              JSB   =STOSV         LET SYS STORE IT
075233                 NOPENT   BSS   0
075233 110 251 010              LDM   R10,=10,0      INDEX TO Y-COOR
075236 000
075237 150 261 072              LDMD  R50,=TMP2      GET Y-COOR
075242 211
075243 316 266 172              JSB   =USERU         MAP TO USER UNIT
075246 316 377 377              JSB   =STOSV         LET SYS STORE IT
075251 110 223                  CLM   R10            INDEX TO X-COOR
075253 150 261 062              LDMD  R50,=TMP1      GET X-COOR
```

```
       LOC    OBJECT CODE   SRC=LOGRWF OBJ=GENGRA     9/11/1981 12:36 PM
===============================================================================
                DIGITIZE / CURSOR / WHERE
4    075256 211
4427 075257 316 266 172              JSB  =USERU        MAP TO USER UNIT
4428 075262 316 377 377              JSB  =STOSV        LET SYS STORE IT
4429 075265 236                      RTN
4430 075266
4431 075266                   *     UU=(MU-XB)/XA
4432 075266                   *
4433 075266                   *       IN:
4434 075266                   *         R50=COORDINATE IN MU
4435 075266                   *         R10=0 IF X AXIS
4436 075266                   *            #0 IF Y AXIS
4437 075266                   *
4438 075266                   *       OUT:
4439 075266                   *         COORDINATE IN UU ON R12 STACK
4440 075266
4441 075266                   USERU   BSS  0
4442 075266 316 357 145              JSB  =GSCALF      PT TO CURR XA
4443 075271 110 220                  TSB  R10          X AXIS ?
4444 075273 367 004                  JZR  USERU0       JIF YES
4445 075275 122 313 020              ADM  R22,=20,0    PT TO YA
4445 075300 000
4446 075301                   USERU0  BSS  0
4447 075301 122 313 020              ADM  R22,=20,0    PT TO AFTER XB
4447 075304 000
4448 075305                   USERU1  BSS  0
4449 075305 140 022 343              POMD R40,-R22     LOAD XB
4450 075310 316 377 377              JSB  =SUB10       MU-XB
4451 075313 150 343                  POMD R50,-R#      POP OFF STACK
4452 075315 140 022 343              POMD R40,-R22     LOAD XA
4453 075320 316 377 377              JSB  =DIV10       (MU-XB)/XA
4454 075323 236                      RTN
4455 075324
4456 075324 241                      OCT  241
4457 075325
4458 075325                   FXD.    BSS  0
4459 075325 100 223                  CLM  R0
4460 075327 316 341 172              JSB  =FXD1        GET ONE PAR
4461 075332 100 211                  ICM  R0
4462 075334 316 377 377              JSB  =STACKX      ANY MORE PAR ?
4463 075337 367 011                  JZR  FXD3         JIF NO
4464 075341 316 377 377 FXD1         JSB  =ONEB        GET ONE PAR
4465 075344 311 010 000              CMM  R#,=10,0     TOO LARGE ?
4466 075347 372 001                  JNC  FXD3         JIF NO
4467 075351 122                FXD2   CLB  R#
4468 075352 176 000 266 FXD3         STBD R76,X0,LAXNY
4468 075355 321 212
4469 075357 236                      RTN
4470 075360
4471 075360 000 055                  OCT  0,55
4472 075362
4473 075362                   RATIO.  BSS  0
4474 075362 140 261 140              LDMD R40,=XH1     XMIN
4474 075365 211
```

DIGITIZE / CURSOR / WHERE

```
075366 150 261 150         LDMD R50,=XH2      XMAX
075371 211
075372 316 377 377         JSB  =SUB10
075375 261 160 211         LDMD R#,=YH1       YMIN
075400 150 261 170         LDMD R50,=YH2      YMAX
075403 211
075404 316 377 377         JSB  =SUB10
075407 316 377 377         JSB  =DIV2         (XMAX-XMIN)/(YMAX-YMIN)
075412 236                 RTN
075413
075413 100 055             OCT  0,55
075415
075415             ERROM.  BSS  0
075415 136 223             CLM  R36           ZERO MSB
075417 260 121 200         LDBD R36,=ERROM#   GET ERROR ROM #
075422 316 377 377 ERCOM   JSB  =CONBIN       CONVERT TO REAL
075425 140 012 345 ERPUSH  PUMD R40,+R12
075430 236                 RTN
075431
075431 100 055             OCT  0,55
075433
075433             ERRSC.  BSS  0
075433 136 223             CLM  R36
075435 260 141 202         LDBD R36,=ERRSC    GET ERRSC
075440 760 760             JMP  ERCOM         DO COMMON PART
075442
```

```
[illegible] ========================================================================================
ITEM   LOC    OBJECT CODE   SAC[illegible]GRAF OBJ[illegible]GENGRA   [illegible]   12:[illegible] PM   [illegible]
========================================================================================
[illegible]         CRT DRIVERS TABLE
[illegible] 075442
4501 075442                       ******************************************************************
4501 075442                       *
4501 075442                       *
4501 075442                       ******************************************************************
4502 075442
4503 075442                       CRTDRV  BSS   0
4504 075442 072 173                       DEF   C0              PLOTTER IS
4505 075444 224 144                       DEF   C2              SCLEAR
4506 075446 015 151                       DEF   C4              PEN SELECT
4507 075450 346 154                       DEF   C6              LINE TYPE
4508 075452 061 152                       DEF   C10             PLOT
4509 075454 165 173                       DEF   C12             PENUP
4510 075456 166 173                       DEF   C14             DIGITIZE
4511 075460 173 173                       DEF   C16             READ CURSOR
4512 075462 004 156                       DEF   C20             CSIZE
4513 075464 071 156                       DEF   C22             LDIR
4514 075466 261 157                       DEF   C24             LABEL
4515 075470 200 147                       DEF   C26             LIMIT
45[illegible] 075472
```

```
075472
075472
075472 124 250 302          C0        BSS   0
075475 377                            LDBD  R24,=CRTSTS     GRAPH ALL ?
075476 200                            ELB   R24
075477 200                            ELB   R24
075500 373 005                        JCY   C0.1
075502 251 315 143                    LDM   R24,=XG1        JIF YES
075505 760 003                        JMP   C0.2            LIMIT FOR GRAPH MODE
075507
075507                      C0.1      BSS   0
075507 251 355 143                    DRP   !24
075512                                LDM   R24,=XGA1
075512 126 251 300          C0.2      BSS   0               LIMIT FOR GRAPH ALL
075515 211                            LDM   R26,=XM1
075516 316 340 144                                          SET MACHINE LIMITS
075521                                JSB   =MOV40
075521 251 300 300                    DRP   !70
075524 140 164 061                    LDM   R70,=0,0,60C,74C,31C,60C,74C,31C
075527 140 164 061
075532
075532 263 342 210          *         3.17460317460  MU/MM FOR CRT
075535 263 362 210                    STMD  R#,=XANM
075540 223                            STMD  R#,=YANM
075541 263 352 210                    CLM   R#
075544 263 372 210                    STMD  R#,=XBMM
075547 262 002 207                    STMD  R#,=YBMM
075552                                STBD  R#,=MONSTD
075552 124 251 300          XMTOXV    BSS   0               FLAG STD LIMIT
075555 211                            LDM   R24,=XM1
075556                                                      SET DEFAULT VIEW PORT
075556 126 251 240          TOXV1     BSS   0
075561 211                            LDM   R26,=XV1
075562 316 340 144
075565 236                            JSB   =MOV40
075566                      C12       RTN
075566
075566 235                  C14       BSS   0
075567 316 044 210                    CLE
075572 236                            JSB   =DGHOOK         FLAG FAILED
075573                                RTN                   TRY HOOK
075573
075573 124 251 302          C16       BSS   0
075576 211                            LDM   R24,=XPEN
075577 126 251 042                                          FOR CRT, CURSOR ADDR
075602 211                            LDM   R26,=XCURS
075603 316 343 144                                          IS THE SAME AS THE
075606 236                            JSB   =MOV20
075607                                RTN                   PEN ADDR
```

```
075607               ***********************************************************
075607               *
075607               *    Free memory function attributes                       *
075607               *
075607               ***********************************************************
075607
075607 100 055              OCT  0,55
075611
075611               FRE.   BSS  0
075611 230                  BIN
075612 155 251 022          LDMD R55,=NXTMEM    get bottom end
075615 200
075616 325 025 200          SBMD R55,=LAVAIL    - top
075621 316 377 377          JSB =CONBI3         convert in R40
075624 140 012 345          PUMD R40,+R12       PUSH ON STACK
075627 236                  RTN
075630
075630 241                  OCT 241
075631 120 251 222   PRNTR. LDM R20,=PS.C.      PRINTER SELECT CODE
075634 200
075635 122 251 137          LDM R22,=PRNTLN     PRINTER LENGTH
075640 202
075641 176 250 120   ISCOM  LDB R76,=120        DEFAULT BUFFER LENGTH
075644 316 377 377          JSB =STACKX
075647 374 014              JLZ SETLN           JIF ONE PARAMETER
075651 316 377 377          JSB =ONEB           GET LINE LN
075654 367 363              JZR ISCOM           DEFAULT TO 80 IF 0
075656 311 335 000          CMM R#,=221D,0      >220
075661 372 002              JNC SETLN           JIF NO
075663 250 334              LDB R#,=220D        220 IS MAX
075665               SETLN  BSS  0
075665 176 022 246          STBD R76,R22        STORE LINELN
075670 316 377 377          JSB =ONER           GET SC
075673 221                  TSM R#
075674 367 025              JZR ADERR           0 NEG
075676 020 247              STMD R#,R20         STORE SELECT CODE
075700 316 377 377          JSB =RTOIN          REAL TO INTEGER
075703 165 311 003          CMM R65,=3,0,0      3 OR MORE?
075706 000 000
075710 373 010              JCY ISOUT
075712 230                  BIN
075713 250 120             LDB R65,=120        MAX FOR SC 1,2 IS 80
075715 022 330              CMBD R65,R22        USER SPEC>80???
075717 373 001              JCY ISOUT           JIF NO
075721 246                  STBD R65,R22        STORE MAX
075722 236           ISOUT  RTN
075723
075723
075723 316 377 377   ADERR  JSB =ERROR+
075726 131                  OCT 89D
075727
075727               ROMERR BSS  0
075727 146 260 123          LDBD R46,=ERRORS    GET ERROR FLAG & CHECK IT
```

```
                 FREE MEMORY FUNCTION
075732 200
075733 766 005                     JNZ  ERNESS          JIF ALREADY ERROR
075735 250 001                     LDB  R46,=ROM#       SET ERR ROM #
075737 262 120 200                 STBD R46,=ERRROM
075742 236                 ERNESS  RTN
075743
075743                   ***WARNING MESSAGES********************
075743 114 111 115                 ASP  "LIMIT OUT OF BOUND"    1
075746 111 124 040
075751 117 125
075753 124 040 117
075756 106 040 102
075761 117 125
075763 116 304
075765                             REP  7
075765 200                         OCT  200
075766 200                         OCT  200
075767 200                         OCT  200
075770 200                         OCT  200
075771 200                         OCT  200
075772 200                         OCT  200
075773 200                         OCT  200
075774                   ***ERROR MESSAGES**********************
075774 104 111 107                 ASP  "DIGITIZE"              9
075777 111 124 111
076002 132 305
076004 101 104 104                 ASP  "ADDR"                  10
076007 322
076010 123 105 114                 ASP  "SELECT CODE"           11
076013 105 103 124
076016 040 103
076020 117 104 305
076023 120 057 120                 ASP  "P/P ROM"               12
076026 040 122 117
076031 315
076032 377                         OCT  377
076033
076033                             FIN
```

SYMBOL	VALUE	TYPE	COUNT	
[illegible]	065413	LCL	1	
[illegible]DRXY	065403	LCL	1	
A[illegible]D	177777	EXT	5	
AD[illegible]CR+	970755	LCL	3	
ADDCR1	970775	LCL	3	
ADDCR2	970772	LCL	4	
ADDPEN	971012	LCL	2	
ADDROI	177777	EXT	6	
ADERR	975723	LCL	1	
ADJ-X	063767	LCL	1	
ADJINC	970733	LCL	3	
ADJSL1	970732	LCL	1	
ADJSLA	970675	LCL	1	
ADJST1	970743	LCL	1	
ADJST2	970757	LCL	1	
ADJSTR	970667	LCL	2	
ADJSZ1	971037	LCL	1	
ADJSZE	971025	LCL	1	
ADRNZ	061475	LCL	1	
ADRNZ+	061525	LCL	1	
ADXORG	970760	LCL	1	
BL[illegible]L.	177777	EXT	1	
AS[illegible]AB	060350	LCL	1	
ATN2.	177777	EXT	1	
AXDSUB	973150	LCL	2	
AXES.	972233	LCL	1	
AXES.+	972244	LCL	3	
AXFLAG	104530	G LAD	9	
AXIDX	105275	G LAD	8	
AXISG-	972651	LCL	1	
AXLP	973001	LCL	3	
AXMAIN	972621	LCL	2	
AXTIC1	105277	G LAD	6	
BACKAX	073515	LCL	1	
BFR1.5	971627	LCL	1	
BFRIN	971564	LCL	1	
BFRIN1	971613	LCL	1	
BFRIN2	971642	LCL	1	
BGNEN-	972143	LCL	3	
BGNEN1	972167	LCL	1	
BG[illegible]ND	972143	LCL	1	
BL[illegible]K	000040	G EQU	2	
BLINK.	974201	LCL	2	
BLKFIL	974660	LCL	1	
BLOFF.	974201	LCL	1	
BPBYT2	974716	LCL	0	<--- NOT REFERENCED??
BPBYT3	974727	LCL	1	
BPBYT4	974732	LCL	1	
BPBYTE	974713	LCL	2	
BPLAS1	974411	LCL	1	
BPLAST	974346	LCL	2	
BPLOT.	974203	LCL	1	
BYTEG	066032	LCL	9	
C0	975472	LCL	2	

[illegible]	VALUE	TYPE	COUNT	Symbol Table	
[illegible]	[illegible]	LCL	1		
[illegible]	[illegible]	LCL	1		
[illegible]	065061	LCL	1		
[illegible]	064433	LCL	1		
C12	073563	LCL	1		
C14	07356[illegible]	LCL	1		
C14.1	065076	LCL	1		
C14.2	065101	LCL	2		
C14.3	065125	LCL	1		
C14.4	065173	LCL	1		
C14.5	065147	LCL	1		
C14.6	065152	LCL	3		
C16	075573	LCL	1		
C2	062224	LCL	1		
C20	067004	LCL	1		
C22	067071	LCL	1		
C24	067661	LCL	2		
C24X	067660	LCL	0	<--- NOT REFERENCED??	
C26	06360[illegible]	LCL	1		
C30.1	067106	LCL	1		
C4	064415	LCL	1		
C6	066346	LCL	1		
C[illegible]GU	063302	LCL	1		
CALGU1	063333	LCL	1		
CALSTO	073225	LCL	1		
CDISP	067113	LCL	2		
CDISP1	067143	LCL	1		
CDISP2	067223	LCL	1		
CETL10	177777	EXT	1		
C[illegible]K	071350	LCL	1		
CHECK-	071340	LCL	1		
CHECK1	071401	LCL	1		
CHECK2	071411	LCL	1		
CHITE	105036	G	CAD	3	
CHK1	062644	LCL	1		
CHK2	062647	LCL	1		
CHK3	062677	LCL	1		
CHKBND	062632	LCL	2		
CHKRTN	062725	LCL	2		
CHLOOP	067714	LCL	1		
CH[illegible]TAB	177777	EXT	1		
CH[illegible]N	063626	LCL	1		
CH[illegible]10	177777	EXT	1		
CHSROI	177777	EXT	1		
CHTABL	104746	G	CAD	2	
CKOUTB	073011	LCL	1		
CL72	073746	LCL	1		
CLIP+	065261	LCL	1		
CLIP.	062771	LCL	1		
CLIP1	063014	LCL	2		
CLIP2	063027	LCL	1		
CLIPBA	104525	G	CAD	6	
CLIPLN	065170	LCL	1		
CLIRTN	063036	LCL	0	<--- NOT REFERENCED??	

```
SYMBOL  VALUE   TYPE   COUNT     Symbol Table
=====================================================================================
.OOP    071321    LCL    6
..O...  071214    LCL    1
...  1  071330    LCL    0    <--- NOT REFERENCED??
CLPFLG  104527  G SAD    3
CLRST1  074643    LCL    1
CLRSTX  074460    LCL    3
CLSCN   067515    LCL    1
CMPAX1  073171    LCL    2
CNTOUT  177777    EXT    1
CNTRTR  177777    EXT    0    <--- NOT REFERENCED??
CNVRTS  070634    LCL    2
COMFLT  062375    ENT    9
COMMA   000054  G EQU    3
COMN    074776    LCL    2
COMFLI  071547    LCL    2
CONBI3  177777    EXT    1
CONBIN  177777    EXT   10
CONINT  177777    EXT    2
CONTR.  061625    LCL    1
CONTU   074313    LCL    0    <--- NOT REFERENCED??
CONTU1  074322    LCL    1
CON   2 074323    LCL    1
COL.0   177777    EXT    1
COSLD   105056  G SAD    5
COUNTK  177777    EXT    1
CR      000015  G EQU    2
CR.     071500    LCL    1
CRT.    177777    EXT    1
CRTBAD  177701  G SAD    1
CRTDAT  177703  G SAD    1
CRTDRV  075442    LCL    1
CRTN    062411    LCL    1
CRTOK   000354    EQU    1
CRTSTS  177702  G SAD    1
CRTUNW  177777    EXT    0    <--- NOT REFERENCED??
CRTWPD  177777    EXT    0    <--- NOT REFERENCED??
CSIZE.  066575    LCL    2
CSIZE1  066625    LCL    1
CSIZE2  066625    LCL    1
CSIZE3  066656    LCL    2
CSI  4  066722    LCL    3
CT.  MD 000200    EQU    1
CURSOR  061334    LCL    1
CURSR.  075143    LCL    1
CURSR1  075152    LCL    1
CWIDE   105026  G SAD    4
DASEG   104775  G SAD    4
DATAB   103535  G SAD    1
DEFALL  072333    LCL    1
DEFCS1  064324    LCL    1
DEFCS2  064303    LCL    0    <--- NOT REFERENCED??
DEFHOK  177777    EXT    0    <--- NOT REFERENCED??
DEFLDR  104524  G SAD    2
DEFSLA  104522  G SAD    4
```

SYMBOL	VALUE	TYPE	COUNT	Symbol Table
[illegible]SIZE	104527	G DAD	4	
[illegible]TYP	066405	LCL	1	
[illegible]C	072510	LCL	2	
[illegible]OK	104044	G DAD	1	
DIGEND	062502	LCL	0	<--- NOT REFERENCED??
DIGIT	061334	LCL	1	
DIGIT	075134	LCL	1	
DIGIT1	062447	LCL	1	
DIGIT2	062443	LCL	1	
DIV10	177777	EXT	8	
DIV2	177777	EXT	5	
DIV88	177777	EXT	1	
DIVIDE	063254	LCL	1	
DNPNA1	071177	LCL	1	
DNPNA2	071174	LCL	2	
DNPNAD	071161	LCL	1	
DOAND	061602	LCL	1	
DONE	065344	LCL	2	
DONEP+	062505	LCL	2	
DRAW.	064727	LCL	2	
DRAW.+	064755	LCL	1	
DRAW.1	064733	LCL	2	
DRAW.2	064741	LCL	2	
DRAW.3	064747	LCL	1	
DRAW.4	065006	LCL	2	
DRAW.5	065015	LCL	1	
DRAW.6	065026	LCL	2	
DRAWC	065506	LCL	1	
DRAWG2	065645	LCL	2	
D[illegible]GO	065636	LCL	1	
DRAWPT	065751	LCL	1	
DREXIT	065734	LCL	1	
DRW.25	064744	LCL	1	
DRWRTN	065052	LCL	2	
DSEG	104771	G DAD	1	
DSEG#	105011	G DAD	3	
DSEG-	105014	G DAD	1	
DSKPF	105013	G DAD	3	
DTCNT	105012	G DAD	3	
DUMMY	061333	LCL	7	
ED	072562	LCL	2	
EF	072325	LCL	1	
EFMT	073736	LCL	2	
EKEY	000105	EQU	1	
EMOVDN	177777	EXT	0	<--- NOT REFERENCED??
EMOVUP	177777	EXT	1	
ENDBPL	074421	LCL	1	
ENDCHR	070541	LCL	1	
ENDRW?	065705	LCL	1	
ENDSR	177777	EXT	1	
ERBEND	100542	G DAD	2	
ERCOM	075422	LCL	1	
ERMESS	075742	LCL	2	
ERPUSH	075425	LCL	0	<--- NOT REFERENCED??

SYMBOL	VALUE		TYPE	COUNT	
[illegible]	177777		EXT	1	
[illegible]F	100476	G	CAD	3	
[illegible]04	100121	G	CAD	1	
ERROM.	975415	.	LCL	1	
ERROR	177777		EXT	3	
ERROR+	177777		EXT	5	
ERROR1	061710		LCL	1	
ERROR2	061705		LCL	3	
ERRORS	100123	G	CAD	1	
ERRROM	100126	G	CAD	1	
ERRFTN	067041		LCL	1	
ERRSC	101141	G	CAD	1	
ERRSC.	975437		LCL	1	
ERRTMP	101667	G	CAD	5	
EVEN	974106		LCL	2	
EXIT	971565		LCL	1	
EXMAIN	973144		LCL	1	
EXSW	972507		LCL	2	
FORMG	177777		EXT	1	
FORMNC	973752		LCL	1	
FRA[illegible].	066165		LCL	1	
FR[illegible]U1	066247		LCL	1	
FRASU2	066252		LCL	1	
FRASU3	066274		LCL	1	
FRASU4	066277		LCL	1	
FRASUB	066244		LCL	1	
FRE.	975611		ENT	2	
FROMXH	064346		LCL	3	
FRTMP1	064600		LCL	2	
FXD.	075325		LCL	1	
FXD1	975341		LCL	1	
FXD2	975351		LCL	0	<--- NOT REFERENCED??
FXD3	975352		LCL	2	
G#LN-1	000357	G	EQU	1	
G$N	177777		EXT	1	
G0-1N	061164		LCL	1	
G0-4N	061142		LCL	3	
G1-2N	061254		LCL	6	
G1-2N+	061257		LCL	1	
G1-7N	061244		LCL	1	
G1[illegible]	061270		LCL	2	
G1.	061061		LCL	4	
G1S	061021		LCL	0	<--- NOT REFERENCED??
G2	062426		LCL	2	
G2-3N	061220		LCL	3	
G2467	061301		LCL	4	
G2N	061210		LCL	5	
G3N	061235		LCL	1	
G3N+	061225		LCL	1	
G4	062423		LCL	0	<--- NOT REFERENCED??
G4N	061200		LCL	2	
GX4	061227		LCL	1	
GBASE	010340	G	EQU	1	
GCH	974004		LCL	1	

SYMBOL	VALUE		TYPE	COUNT	
[illegible]CLP+	062275		LCL	1	
[illegible]+	062331		LCL	0	<--- NOT REFERENCED??
[illegible]++	062323		ENT	1	
G[illegible].	062214		LCL	1	
GCLR1	062234		LCL	1	
GCLR2	062275		LCL	1	
GENE	073775		LCL	2	
GENSPC	072517		LCL	2	
GET1$	177777		EXT	0	<--- NOT REFERENCED??
GET2N	177777		EXT	1	
GET4N	177777		EXT	1	
GET???	061316		LCL	7	
GETBND	065377		LCL	2	
GETPA?	177777		EXT	1	
GETPC	065153		LCL	3	
GETPC1	065164		LCL	1	
GETPSY	065776		LCL	1	
GFOUR	062413		LCL	4	
GINTDS	177401	G	CAD	1	
GLINE	104740	G	CAD	6	
GO[illegible]AW	065676		LCL	1	
GO[illegible]N	072750		LCL	1	
GPON	062142		LCL	0	<--- NOT REFERENCED??
GPON3	064142		LCL	2	
GPRNT	061021		LCL	1	
GR.	177777		EXT	1	
GRAFA.	177777		EXT	1	
GRAFIN	071256		ENT	1	
GRAPH	177777		EXT	4	
G[illegible]H.	177777		EXT	1	
GREAD	061035		LCL	1	
GREAD.	074467		LCL	1	
GREAD1	074505		LCL	1	
GRID.	072217		LCL	1	
GRIEND	062162		LCL	2	
GRINI+	062142		ENT	1	
GRINIT	062065		ENT	1	
GSCALF	062757		ENT	6	
GSIZE	104742	G	CAD	6	
GTSTRK	070602		LCL	2	
GT[illegible]	062440		LCL	3	
GU[illegible]MU	067015		LCL	2	
HDISP	105131	G	CAD	2	
HDISPX	105131	G	CAD	1	
HDISPY	105141	G	CAD	1	
IDRAW.	064706		LCL	1	
IDRAW1	064712		LCL	2	
IMCADR	101770	G	CAD	1	
IMCLEN	101765	G	CAD	1	
IMOVE.	064643		LCL	1	
IMSADR	101763	G	CAD	1	
IN+	065375		LCL	1	
IN?	065346		LCL	7	
INCHR	177777		EXT	6	

SYMBOL VALUE TYPE COUNT Symbol Table [illegible]

SYMBOL	VALUE		TYPE	COUNT	
[illegible]CH	065725		LCL	1	
[illegible]CY	065726		LCL	1	
[illegible]AF	071667		ENT	3	
[illegible]SUB	071679		LCL	2	
[illegible]IT	062167		LCL	1	
[illegible]NER	074625		LCL	1	
INPBUF	100236	G	SAD	3	
INPR10	101717	G	SAD	5	
INPTR	101143	G	SAD	6	
INRTN	065376		LCL	2	
INTRC	065414		LCL	1	
INTRSC	17756[illegible]	G	SAD	1	
IOBITS	101140	G	SAD	1	
IOTRFC	103643	G	SAD	3	
IPLOT.	064660		LCL	1	
ISCOM	075641		ENT	2	
ISOUT	075721		LCL	2	
KEYCNT	100153	G	SAD	1	
KEYCOD	177403	G	SAD	2	
KEYE	000240		EQU	1	
KEYHIT	101142	G	SAD	2	
KEY[illegible]S	177402	G	SAD	1	
KR[illegible]T1	100154	G	SAD	1	
KYHOOK	071544		LCL	1	
KYIDLE	103677	G	SAD	3	
L7	177777		EXT	1	
LABCHR	070451		LCL	1	
LABEL	061415		LCL	1	
LABEL.	067262		LCL	1	
LABEN2	067760		LCL	0	<--- NOT REFERENCED??
LABEND	067752		LCL	1	
LABEOL	067451		LCL	1	
LABEX	067306		LCL	2	
LABLNG	105117	G	SAD	7	
LABOL2	067367		LCL	1	
LAST+1	100476	G	SAD	1	
LAVAIL	100025	G	SAD	2	
LAX	073543		LCL	1	
LAXES.	072241		LCL	1	
LAXN	105322	G	SAD	1	
LAX[illegible]N	073706		LCL	1	
LA[illegible]Y	105321	G	SAD	3	
LAY	073606		LCL	1	
LBADDR	105121	G	SAD	4	
LBHOOK	067324		LCL	1	
LBLEN	105117	G	SAD	6	
LBLFLG	105116	G	SAD	5	
LBTRAP	067337		LCL	1	
LDIR.	067952		LCL	2	
LDNY	073703		LCL	1	
LDPLTD	066644		LCL	1	
LENOK	074551		LCL	1	
LENSET	074527		LCL	1	
LGRID.	072225		LCL	1	

[illegible] Symbol Table [illegible]

NAME	VALUE	TYPE	COUNT	
[illegible]	[illegible]	LCL	1	
[illegible]	066536	LCL	1	
[illegible]	066556	LCL	0	<--- NOT REFERENCED??
[illegible]	066336	LCL	2	
LINET1	066367	LCL	1	
LINET2	066437	LCL	1	
LINET3	066477	LCL	1	
LINET5	066510	LCL	1	
LINET6	066542	LCL	1	
LMOVE	073626	LCL	1	
LNTYP+	062015	LCL	1	
LNTYPE	104750	G LAD	2	
LOCAT.	063075	LCL	1	
LOCAT0	063101	LCL	0	<--- NOT REFERENCED??
LOCAT1	063114	LCL	1	
LOCRTN	063134	LCL	0	<--- NOT REFERENCED??
LORG.	064067	LCL	2	
LORG1	064103	LCL	2	
LORG2	064105	LCL	2	
LORG3	064107	LCL	1	
LORG4	064122	LCL	1	
LR[illegible]	000663	EQU	1	
LT[illegible]AD	071747	LCL	1	
LTYPE#	104537	G LAD	4	
MAJORT	073103	LCL	1	
MAP+	062576	LCL	2	
MAP1.5	066105	LCL	1	
MAP1MU	062617	LCL	2	
MAP2MU	062554	LCL	1	
M[illegible]MU	062557	LCL	3	
MAPABS	066113	ENT	4	
MAPAR	064460	LCL	1	
MAPAR2	064514	LCL	1	
MAPCR	066044	LCL	1	
MAPCR1	066057	ENT	3	
MAPCR2	066107	LCL	1	
MAPIC	064527	LCL	2	
MAPIC1	064550	LCL	1	
MAPIC2	064543	LCL	1	
MAPIPA	064502	LCL	2	
MAP[illegible]	062614	ENT	3	
MO[illegible]	177777	EXT	1	
MOV**	062365	LCL	2	
MOV10	062346	LCL	5	
MOV20	062343	LCL	9	
MOV40	062340	ENT	7	
MOVDN	177777	EXT	0	<--- NOT REFERENCED??
MOVE.	064634	LCL	2	
MOVEMU	073530	LCL	3	
MOVXY	062354	LCL	2	
MPY10	177777	EXT	16	
MPYROI	177777	EXT	6	
MSCAL.	063630	LCL	1	
MULPY	066126	LCL	1	

```
[illegible]
[illegible]   VALUE   [illegible]   COUNT   [illegible]
=================================================================
[illegible] [illegible]     LCL     1
[illegible] 071752          LCL     1
[illegible] [illegible]     LCL     1
FR5       177777         EXT      1
OMAF      063050         LCL      3
OMAPX     063067         LCL      1
ONSTD     103402       G CAD      3
OPENT     075233         LCL      3
OSTRK     105124       G CAD      3
OT2FR     073300         LCL      1
OTBS      071450         LCL      1
OVARC     063037         LCL      1
OVARL     063135         LCL      1
OVARS     063567         LCL      1
OVARX     062503         LCL      2
OVRTN     062504         LCL      1
RMLKY     071456         LCL      1
UMVA+     177777         EXT      1
XTB       074213         LCL      1
XTB1      074242         LCL      0    <--- NOT REFERENCED??
XT[illegible]   074245   LCL      1
XT[illegible]L  074250   LCL      1
XTMEM     100022       G CAD      1
XTST2     070542         LCL      1
XTSTR     070525         LCL      2
O-S*T     073331         LCL      2
O>S+T     073300         LCL      1
OBFCK     177777         EXT      0    <--- NOT REFERENCED??
OFFAX     073135         LCL      1
ONES      177777         EXT     11
ONER      177777         EXT      9
ONEX      177777         EXT      1
ORDER2    062726         LCL      2
ORDER4    062723         ENT      2
ORDRTN    062756         LCL      1
ORGDIR    074154         LCL      2
OUTBYT    066037         LCL      2
OUTER     074610         LCL      2
PAGES.    177777         EXT      1
PAROK     067050         LCL      1
PAP[illegible]?  067031  LCL      1
PAR.AB    060179         LCL      1
PAUSE.    071515         LCL      1
PAUSE1    071534         LCL      1
PCNTRL    104531       G CAD     11
PDIR.     075056         LCL      2
PDIR1     075062         LCL      1
PDIR2     075067         LCL      1
PDIR3     07511          LCL      1
PEN#      104535       G CAD      5
PEN+      064440         LCL      1
PEN-1     066021         LCL      0    <--- NOT REFERENCED??
PEN-2     066016         LCL      0    <--- NOT REFERENCED??
PEN.      064402         LCL      2
```

SYMBOL	VALUE	TYPE	COUNT	SYMBOL TABLE
[illegible]	[illegible]	LCL	1	
[illegible]	063024	LCL	1	
[illegible]	[illegible]	LCL	0	<--- NOT REFERENCED??
[illegible]ON	104526	G LAD	7	
PENDN.	064272	LCL	5	
PENPTN	064457	LCL	0	<--- NOT REFERENCED??
PENUP.	064357	ENT	9	
PERROR	061154	LCL	[illegible]	
PJ	061342	LCL	1	
PLHOOK	103661	G LAD	1	
PLOT.	064652	LCL	1	
PLOTER	062547	LCL	1	
PLOTP#	104531	G LAD	5	
PLOTR.	064127	LCL	1	
PLOTSY	100151	G LAD	2	
PNOTOK	070450	LCL	3	
PNTOMP	064332	LCL	0	<--- NOT REFERENCED??
POK?	070402	LCL	4	
POPLP	073775	LCL	1	
POPUSH	061172	LCL	2	
PR[illegible]D-	061451	LCL	1	
PR[illegible]DP	061454	ENT	1	
PR[illegible]FIO	105014	G LAD	4	
PRDRV	177777	EXT	0	<--- NOT REFERENCED??
PRECL1	070376	LCL	5	
PRECLP	070245	LCL	1	
PREX+	061562	LCL	1	
PREXIT	061552	LCL	1	
PRINT	177777	EXT	1	
P[illegible]	061574	LCL	1	
PRNTLN	101137	G LAD	1	
PRNTR.	075631	ENT	2	
PS.C.	100222	G LAD	1	
PTJP	061176	LCL	1	
PTOKEN	061107	LCL	1	
PTOKEX	061077	LCL	7	
PTOKN+	061104	LCL	2	
PTR1	177710	G LAD	6	
PTR2	177714	G LAD	12	
PTR2+	177716	G LAD	7	
PT[illegible]-	177715	G LAD	9	
P[illegible]	061275	LCL	1	
PUSH+S	061123	LCL	1	
PUSHEX	061135	LCL	10	
PUSHJM	061314	LCL	3	
PUSHR#	073416	LCL	1	
R1.	063442	ENT	8	
R100.	063434	LCL	1	
R2.	063451	LCL	3	
R2/3	067005	LCL	2	
R46177	061532	LCL	1	
RATIO.	075362	LCL	1	
READY	065614	LCL	1	
READY2	065624	LCL	1	

SYMBOL	VALUE		TYPE	COUNT	
[illegible]	177777		EXT	1	
[illegible]	067603		INT	1	
[illegible]	177777		EXT	1	
[illegible]	061445		LCL	1	
RESEND	104161	G	LAD	2	
RESMEM	177777		EXT	1	
RESTHK	067651		INT	1	
RESTYP	070025		LCL	1	
RETRHI	177777		EXT	0	<--- NOT REFERENCED??
RMSCAN	061126		LCL	2	
ROM#	000001		EQU	3	
ROM#-	000377		EQU	1	
ROMERR	075727		LCL	5	
ROMEX	061160		LCL	2	
ROMFL	104065	G	LAD	1	
ROMJSB	177777		EXT	17	
ROMRTN	177777		EXT	1	
ROTAXY	071050		LCL	2	
ROUND	177777		EXT	1	
RPLOT.	064665		LCL	1	
RST6	067523		LCL	1	
RSTART	177777		EXT	2	
RS[illegible]K	177777		EXT	1	
RTLHAD	071145		LCL	1	
RTOIN	177777		EXT	2	
RTRNAD	071741		LCL	1	
RUNTAB	060012		LCL	1	
S1	065207		LCL	0	<--- NOT REFERENCED??
S2	065213		LCL	2	
S3	065227		LCL	0	<--- NOT REFERENCED??
S4	065236		LCL	1	
S5	065245		LCL	1	
SX45	065604		LCL	1	
SAVTYP	067764		LCL	1	
SCALE.	063461		LCL	2	
SCALE1	063464		LCL	1	
SCALF	063146		LCL	1	
SCALF1	063165		LCL	1	
SCAN	177777		EXT	1	
SCANLX	177777		EXT	1	
SCERR	061557		LCL	4	
SC[illegible]P	101721	G	LAD	1	
SE[illegible]+	067667		LCL	1	
SENDET	067577		LCL	1	
SEP10	177777		EXT	1	
SEP15	177777		EXT	1	
SEPC	070554		LCL	1	
SEPS	070621		LCL	1	
SET1ST	072705		LCL	1	
SETCC-	177777		EXT	0	<--- NOT REFERENCED??
SETGU.	063505		LCL	1	
SETLN	075665		LCL	2	
SETOR2	070144		LCL	1	
SETOR3	070157		LCL	1	

SYMBOL	VALUE		TYPE	COUNT	Symbol Table
[illegible]OC4	070174		LCL	1	
[illegible]LTORS	070206		LCL	1	
[illegible]S	070225		LCL	1	
[illegible]RG	070054		LCL	1	
SCTPEN	064445		LCL	1	
SETUP	065667		LCL	2	
SETUU.	063474		LCL	3	
SEWH1	075165		LCL	1	
SHFTIT	065767		LCL	1	
SHIFT1	075030		LCL	1	
SHIFT2	075037		LCL	2	
SHIFT3	075045		LCL	1	
SHIFTL	075040		LCL	3	
SHIFTR	075023		LCL	1	
SHORTN	064065		LCL	1	
SHOW.	063721		LCL	1	
SINI0	177777		EXT	1	
SINLD	105046	G	SAD	6	
SINPD	105066	G	SAD	2	
SNDETX	067539		LCL	1	
SNDLX1	067551		LCL	1	
SND[illegible]X2	067557		LCL	1	
S[illegible]KX	177777		EXT	10	
STBEEP	177777		EXT	1	
STCNTR	070547		LCL	1	
STLOOP	070473		LCL	1	
STOSV	177777		EXT	3	
STRKAD	105125	G	SAD	3	
STROK	177777		EXT	1	
S[illegible]EF	177777		EXT	1	
SU[illegible]10	177777		EXT	16	
SUBROI	177777		EXT	6	
SVCWRD	100162	G	SAD	4	
SWITC+	072477		LCL	2	
SWITCH	072473		LCL	3	
T1#SEG	062035		LCL	0	<--- NOT REFERENCED??
T1SEG1	062036		LCL	0	<--- NOT REFERENCED??
T1TOTS	065036		LCL	3	
T3#SEG	062042		LCL	0	<--- NOT REFERENCED??
T3SEG1	062043		LCL	0	<--- NOT REFERENCED??
T3SEG2	062044		LCL	0	<--- NOT REFERENCED??
T4[illegible]EG	062055		LCL	0	<--- NOT REFERENCED??
T4SEG1	062056		LCL	0	<--- NOT REFERENCED??
T4SEG2	062057		LCL	0	<--- NOT REFERENCED??
T5#SEG	062037		LCL	0	<--- NOT REFERENCED??
T5SEG1	062040		LCL	0	<--- NOT REFERENCED??
T5SEG2	062041		LCL	0	<--- NOT REFERENCED??
T6#SEG	062044		LCL	0	<--- NOT REFERENCED??
T6SEG1	062045		LCL	0	<--- NOT REFERENCED??
T6SEG2	062046		LCL	0	<--- NOT REFERENCED??
T6SEG3	062047		LCL	0	<--- NOT REFERENCED??
T6SEG4	062050		LCL	0	<--- NOT REFERENCED??
T7#SEG	062050		LCL	0	<--- NOT REFERENCED??
T7SEG1	062051		LCL	0	<--- NOT REFERENCED??

SYMBOL	VALUE		TYPE	COUNT	
[illegible]	[illegible]		LCL	0	<--- NOT REFERENCED??
[illegible]	062057		LCL	0	<--- NOT REFERENCED??
[illegible]	062054		LCL	0	<--- NOT REFERENCED??
TS[illegible]EG	0620[illegible]		LCL	0	<--- NOT REFERENCED??
TSSEG1	062060		LCL	0	<--- NOT REFERENCED??
TSSEG2	062061		LCL	0	<--- NOT REFERENCED??
TSSEG3	062062		LCL	0	<--- NOT REFERENCED??
TSSEG4	062063		LCL	0	<--- NOT REFERENCED??
TSSEG5	062064		LCL	0	<--- NOT REFERENCED??
TSSEG6	062065		LCL	0	<--- NOT REFERENCED??
TAB.	177777		EXT	1	
TAKEHK	067301		ENT	2	
TANIO	177777		EXT	1	
TANSL	105106	G	CAD	2	
TEE	072362		LCL	1	
TEE+	072367		LCL	1	
TFS	072273		LCL	1	
TIC	073430		LCL	1	
TICCOM	073406		LCL	1	
TICMAJ	073351		LCL	1	
TICMIN	073376		LCL	1	
TI[illegible]T	105317	G	CAD	4	
TI[illegible]K	072373		LCL	1	
TICSZE	105307	G	CAD	3	
TMP1	104462	G	CAD	65	
TMP2	104472	G	CAD	36	
TMP2-	104502	G	CAD	1	
TMP3	104502	G	CAD	6	
TMP4	104512	G	CAD	1	
TMP4-	104522	G	CAD	1	
TOS	101744	G	CAD	4	
TOTMP3	065042		LCL	3	
TOXNAP	064336		LCL	1	
TOXS1	063125		LCL	1	
TOXV1	075554		LCL	1	
TRAFIC	177777		EXT	0	<--- NOT REFERENCED??
TRANV	075174		LCL	0	<--- NOT REFERENCED??
TRY1N	177777		EXT	1	
TRYYAX	072635		LCL	1	
TWO	072036		LCL	1	
TWO[illegible]	072037		LCL	1	
T[illegible]	177777		EXT	3	
TWOFOR	072122		LCL	1	
TWOR	177777		EXT	4	
TYPE1	062033		LCL	1	
TYPE2	062035		LCL	1	
TYPE3	062042		LCL	1	
TYPE4	062053		LCL	1	
TYPE5	062037		LCL	1	
TYPE6	062044		LCL	1	
TYPE7	062050		LCL	1	
TYPE8	062057		LCL	1	
TYPEOK	066411		LCL	1	
UNCLI.	063272		LCL	1	

SYMBOL VALUE FLG COUNT SYMBOL TABLE

SYMBOL	VALUE	FLAG	COUNT	
[illegible]	074445	LCL	4	
[illegible]	074415	LCL	1	
[illegible]RG	064010	LCL	1	
[illegible]EN	064565	LCL	2	
UPDPT	064516	LCL	2	
UPLTOR	062525	LCL	14	
USER	105127	G CAD	11	
USERU	075265	LCL	4	
USERUG	075301	LCL	1	
USERU1	075305	LCL	0	<--- NOT REFERENCED??
USING?	100064	G CAD	2	
UST	072063	LCL	1	
VDISP	105151	G CAD	3	
VDISPX	105151	G CAD	1	
VDISPY	105161	G CAD	1	
VITEM	061374	LCL	3	
WARN	177777	EXT	1	
WHERE	061334	LCL	1	
WHERE.	075161	LCL	1	
X-YAX	072010	LCL	1	
XAGN	072314	LCL	2	
X[illegible]	104242	G CAD	5	
XAMM	104342	G CAD	4	
XAUU	104302	G CAD	6	
XAXIS.	071771	LCL	1	
XBMM	104352	G CAD	1	
XBUU	104312	G CAD	3	
XCURS	104442	G CAD	3	
XEND	105201	G CAD	7	
XF[illegible]	061715	LCL	1	
XGL	061725	LCL	0	<--- NOT REFERENCED??
XGA1	061755	ENT	2	
XGA2	061765	LCL	0	<--- NOT REFERENCED??
XH1	104540	G CAD	10	
XH2	104550	G CAD	2	
XINTR	105263	G CAD	1	
XL	074065	LCL	1	
XLORG	105025	G CAD	4	
XLPEN	106550	G CAD	2	
XMI	104700	G CAD	3	
XM[illegible]OR	105231	G CAD	4	
X[illegible]P	105323	G CAD	2	
XMTOXV	075552	LCL	1	
XOK	065557	LCL	1	
XORG	104422	G CAD	2	
XPEN	104402	G CAD	16	
XPOS	072436	LCL	1	
XS1	104600	G CAD	6	
XSPAC	105211	G CAD	7	
XSTART	105171	G CAD	6	
XV1	104640	G CAD	8	
XVTOXS	063121	LCL	2	
XXORG	106575	G CAD	3	
YAGM	072303	LCL	3	

BOL	VALUE	TYPE	COUNT	
[illegible]	104362	G :AD	1	
[illegible]	10432[illegible]	G :AD	3	
[illegible]	072001	LCL	1	
[illegible]	104272	G :AD	1	
:1	061735	LCL	0	<--- NOT REFERENCED??
:2	061745	LCL	0	<--- NOT REFERENCED??
GA1	061775	LCL	0	<--- NOT REFERENCED??
GA2	062005	LCL	0	<--- NOT REFERENCED??
H1	104560	G :AD	2	
H2	104570	G :AD	1	
INOX	000042	EQU	6	
INTR	105221	G :AD	6	
LORG	105024	G :AD	2	
M2	104730	G :AD	2	
MAJOR	105273	G :AD	1	
MAP	105333	G :AD	2	
OK	065543	LCL	1	
PEN	104412	G :AD	4	
POS	072425	LCL	1	
SPAC	105253	G :AD	1	
ERO	000005	G :AD	2	

COGRAF HAD 0 ERRORS 0 WARNINGS 777 LABELS LAST ERROR AT 0