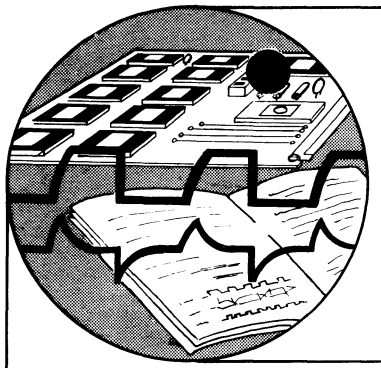


APPLICATION NOTE

NOVEMBER 1982

NA-028A

Basic routines for EF9365 and EF9366 graphic display processors

Philippe LAMBINET
Laboratoire d'Applications

THOMSON-EFCIS

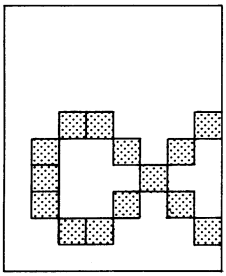
THOMSON-EFCIS integrated circuits

INTRODUCTION

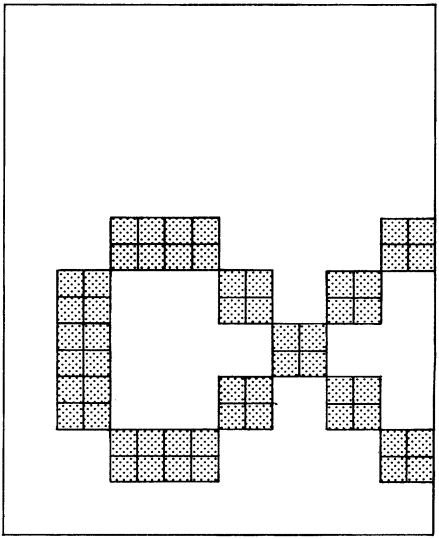
Programmers, using Assembly Language, will find in this Application Note, some software tools, allowing them to build more powerful programs.
Routines developped thereafter have been choosen because they are very commonly used.
All software has been written in EF6800 Assembly language.
Fully optimized routines are often very hard to understand and always hard to modify.
We tried to follow algorithms closely so as to make programs as clear as possible.

SOFTWARE CHARACTER GENERATOR

EF9365 users often think they have a restricted character set : the 96 on-chip character set.
This routine shows that any character set is possible, in any type of matrix.
A scaling factor, in CSIZE register, can be applied to the 8 x 10 matrix choosen in our program (cf fig. 1) on both X and Y sizes independently.
Program reads the matrix and writes corresponding dots on the screen as shown in the algorithm (cf fig. 2), space between characters is included in the 8 x 10 matrix.

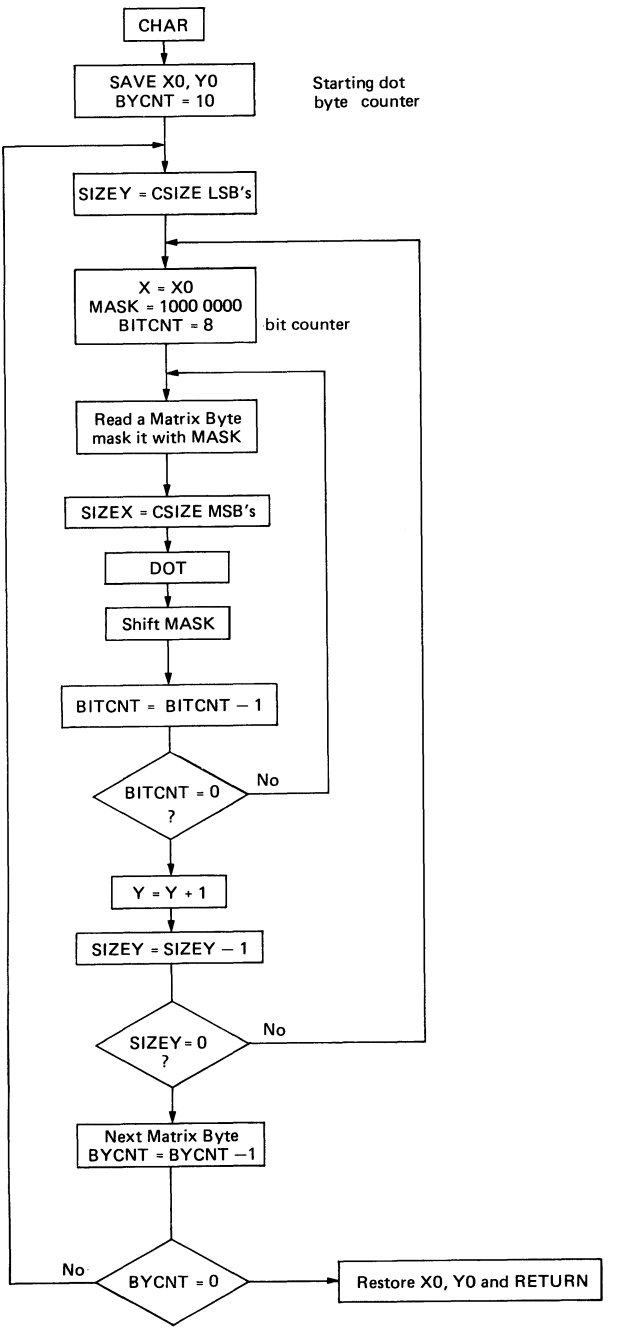


8 x 10 Matrix
Scaling factor on X = 1
on Y = 1



Scaling factor on X = 2
on Y = 2

FIGURE 1 – SCALING FACTOR EFFECT ON 8 x 10 MATRIX



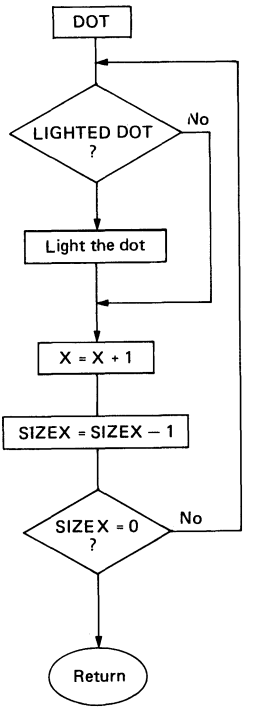


FIGURE 2 – SOFTWARE CHARACTER GENERATOR ALGORITHM

```
PAGE  001  CHARGEN .SA:0

*************************************************************************
*    ------CHARACTER     GENERATION      SUBROUTINE-------              *
*************************************************************************
*
*
*       This subroutine can be used to write any character defined
*     by software.
*       Each character is defined by 10 bytes,allowing a 8 x 10 matrix.
*     First byte is always the image of the character
*     bottom.Last byte is always the image of the character top.
*       Left side of the character is described by MSB's.Right side by
*     LSB's.
*       This subroutine is called on address CHAR by the mean of a
*                 JSR CHAR   instruction
*       Before you call this subroutine,some actions are to be taken
*                   1- The character matrix is defined somewhere in
*     memory by 10 byte table.
*                   2- EF9365 or EF9366 registers are correctly
*     initialized.
*                   3- X index register of EF6800 processor points to
*     the character matrix.
*************************************************************************
*
* this subroutine is non-reentrant.
* all registers are modified by this subroutine.
*
*************************************************************************
*
*    EXTERNAL CONDITIONS :
*    ********************
*
*        INPUT: A and B - no condition
*               X contains the address of the matrix description table
*        OUTPUT:All registers changed
*
*************************************************************************
*
*   Subroutine features:
*   ********************
*
*      this subroutine executes any character writing .
*      this subroutine takes care of GDP register CSIZE to write.
*      this means that all sizes are available between 1 and 16,on
*     both X and Y directions.
*      X register is automatically incremented and updated so as to
*     point to the next character when subroutine has ended.
*
*      the following features are not implemented:
*      writing along Y axis
*      writing slanted characters.
*
*************************************************************************
```

```
PAGE  002  CHARGEN .SA:O

                    *
                           OPT    REL
                    *
                           XREF   CCOLOR,CTRL1,CTRL2,CSIZE,MSBX,MSBY,CMD
                    *
                           XDEF   CHAR
                    *
                    ********
                    *MACRO*
                    ********
                    * THIS MACRO IS USED TO TEST IF GDP IS READY FOR A NEW COMMAND
                    *
                    TEST   MACR
                        LDAB CMD
                        BITB #$04
                        BEQ *-5
                        ENDM
                    *
                    *************
                    *
0000                       DSCT
                    *
0000  0002  A MSAVX  RMB    2          X POSITION SAFEGUARD
0002  0002  A MSAVY  RMB    2          Y POSITION SAFEGUARD
0004  0001  A LITDOT RMB    1          DOT IMAGE
0005  0001  A SIZEX  RMB    1          X SCALING FACTOR
0006  0001  A SIZEY  RMB    1          Y SCALING FACTOR
0007  0001  A MASK   RMB    1          MASK
0008  0001  A BITCNT RMB    1          BIT COUNTER
0009  0001  A BYTCNT RMB    1          BYTE COUNTER
                      *
```

```
   PAGE  003  CHARGEN .SA:0

0000                    PSCT
                        *
                        ************************************************************
                        * SUBROUTINE USED TO LIGHT OR SWITCH OFF A DOT             *
                        ************************************************************
                        *
                        *  THIS SUBROUTINE WRITES A DOT (LIGHTED OR SWITCHED OFF)
                        *  ACCORDING TO THE X SCALING FACTOR.
                        *  EACH DOT IS REPEATED AS MANY TIMES AS SPECIFIED BY
                        * X SCALING FACTOR. THIS REPRODUCES THE GDP INTERNAL FUNCTION.
                        ************************************************************************
                        *
0000 B6 0004   D DOT    LDAA   LITDOT   IN LITDOT: 0 IF DOT TO BE SWITCHED OFF
                        *                           1 IF DOT TO BE LIGHTED
0003 81 00     A        CMPA   #$00     0?
0005 27 0C 0013         BEQ    SWIOFF   IF 0 DOT UNCHANGED
0007 86 80     A        LDAA   #$80     IF 1,LIGHT ONE DOT COMMAND (SHORT VECTOR CMD)
0009 B7 0000   A        STAA   CMD
000C                    TEST
0013 7C 0001   A SWIOFF INC    MSBX+1   X  INCREMENTATION
0016 26 03 001B         BNE    DECSIZ   IF #0 NO MSB UPDATING
0018 7C 0000   A        INC    MSBX     IF =0 BORROW ON MSB
001B 7A 0005   D DECSIZ DEC    SIZEX    X SCALING FACTOR DECREMENTATION
001E 26 E0 0000         BNE    DOT      IF #0  SAME DOT IS WRITTEN AGAIN
0020 39                 RTS
                        *
                        ************************************************************
                        * SUBROUTINE USED TO READ X SCALING FACTOR IN CSIZE REGISTER *
                        ************************************************************
                        *
0021 F6 0000   A READSX LDAB   CSIZE    4 MSB's OF CSIZE
0024 0C                 CLC             GIVE THE X SCALING FACTOR
0025 56                 RORB            FOUR RIGHT SHIFTS GIVE IT
0026 57                 ASRB
0027 57                 ASRB
0028 57                 ASRB
0029 F7 0005   D        STAB   SIZEX    AND WE CAN SAVE IT IN SIZEX VARIABLE
002C 39                 RTS
                        *
                        ************************************************************
                        * SUBROUTINE USED TO READ Y SCALING FACTOR IN CSIZE REGISTER  *
                        ************************************************************
                        *
002D F6 0000   A READSY LDAB   CSIZE
0030 C4 0F     A        ANDB   #$0F     CLEAR 4 MSB's OF CSIZE
0032 F7 0006   D        STAB   SIZEY    BEFORE SAVING IT IN SIZEY VARIABLE
0035 39                 RTS
```

```
PAGE  004  CHARGEN .SA:O


                    ********************************************************************
                    *   ------MAIN  SUBROUTINE-------                                   *
                    ********************************************************************
                    *
                    *  THIS IS THE ENTRY POINT OF THE CHARACTER GENERATOR
                    *
                    ********************************************************************
                    *
          0036   P CHAR    EQU     *
0036 B6 0001   A          LDAA    MSBY+1    Y POSITION SAFEGUARD IN MSAVY
0039 B7 0003   D          STAA    MSAVY+1
003C B6 0000   A          LDAA    MSBY
003F B7 0002   D          STAA    MSAVY
0042 B6 0000   A          LDAA    MSBX      X POSITION SAFEGUARD
0045 B7 0000   D          STAA    MSAVX
0048 B6 0001   A          LDAA    MSBX+1
004B B7 0001   D          STAA    MSAVX+1
004E 86 OA     A          LDAA    #$OA      10 BYTES COUNTER
0050 B7 0009   D          STAA    BYTCNT    INITIALISATION
          0053   P LOOP1   EQU     *         ALL MATRIX BYTES WILL BE WRITTEN
                    *                       AS SPECIFIED BY CSIZE REGISTER
0053 BD 002D   P          JSR     READSY    Y SCALING FACTOR INITIALISATION
          0056   P LOOP2   EQU     *         EACH BYTE IS WRITTEN AS MANY TIMES
                    *                       AS SPECIFIED IN SIZEY
0056 B6 0000   D          LDAA    MSAVX     X POSITION INITIALISATION
0059 B7 0000   A          STAA    MSBX
005C B6 0001   D          LDAA    MSAVX+1
005F B7 0001   A          STAA    MSBX+1
0062 86 80     A          LDAA    #$80      MASK:10000000 THIS BYTE IS USED
0064 B7 0007   D          STAA    MASK      TO TEST EACH BIT OF A MATRIX BYTE
0067 86 08     A          LDAA    #$08      8 BIT COUNTER INITIALISATION
0069 B7 0008   D          STAA    BITCNT
          006C   P LOOP3   EQU     *         EACH BIT IS WRITTEN AS MANY TIMES
                    *                       AS SPECIFIED IN SIZEX
006C A6 00     A          LDAA    O,X       MATRIX BYTE
006E B4 0007   D          ANDA    MASK      BIT TEST
0071 B7 0004   D          STAA    LITDOT    COMMAND BIT SETTING
0074 BD 0021   P          JSR     READSX    X SCALING FACTOR READING
0077 BD 0000   P          JSR     DOT       DOT ON OR OFF ACCORDING TO LITDOT
007A OC                   CLC
007B 76 0007   D          ROR     MASK      MASK BYTE UPDATING TO ACCESS NEXT DOT
007E 7A 0008   D          DEC     BITCNT    IF #O NEXT BIT
0081 26 E9 006C           BNE     LOOP3     IF =O EXIT FROM LOOP3
0083 7C 0001   A          INC     MSBY+1    UPDATES Y POSITION
0086 7A 0006   D          DEC     SIZEY
0089 26 CB 0056           BNE     LOOP2     IF #O SAME BYTE IS WRITTEN AGAIN
008B 08                   INX               IF =O EXIT FROM LOOP2 AND TAKE NEXT BYTE
008C 7A 0009   D          DEC     BYTCNT    UPDATES BYTE COUNTER
008F 26 C2 0053           BNE     LOOP1     IF #O CONTINUE
0091 B6 0002   D          LDAA    MSAVY     IF =O CHARACTER GENERATION IS DONE,EXIT
0094 B7 0000   A          STAA    MSBY      RESTORE Y POSITION
0097 B6 0003   D          LDAA    MSAVY+1
009A B7 0001   A          STAA    MSBY+1
009D 39                   RTS
```

```
*****************************************
*     EXAMPLE FOR MATRIX PROGRAMMING      *
*****************************************
* WE GIVE HERE ALL GREEK LETTERS DEFINED AS NECESSARY
* TO SUIT WITH CHARGEN SUBROUTINE.
* NB:OMEGA IS GIVEN IN BOTH LOWER AND UPPER CASE
*****************************************
ALPHA FDB $0031,$4A44,$4A31,$0000,$0000
*
BETA FDB $0040,$2020,$3C22,$223C,$223C
*
GAMA FDB $0030,$3030,$1018,$1412,$6100
*
DELTA FDB $0030,$4848,$3020,$4040,$4830
*
EPSILO FDB $0018,$2040,$7840,$2018,$0000
*
DZETA FDB $0018,$0438,$4040,$2010,$0E16
*
TETA FDB $0018,$2442,$427E,$4242,$2418
*
ETA FDB $0002,$0202,$1212,$1252,$2C00
*
IOTA FDB $0030,$4840,$4040,$4000,$0000
*
KAPPA FDB $0044,$4A50,$6050,$4840,$0000
*
LANDA FDB $0042,$2418,$1010,$1010,$2040
*
MU FDB $0040,$4040,$7448,$4848,$4800
*
NU FDB $0020,$3028,$2422,$6200,$0000
*
KSI FDB $000C,$023C,$4020,$1820,$1C08
*
OMICRO FDB $0018,$2442,$4224,$1800,$0000
*
PI FDB $0014,$1414,$1454,$3F00,$0000
*
RO FDB $0040,$4040,$5864,$4224,$1800
*
SIGMA FDB $0000,$1824,$4224,$1F00,$0000
*
THO FDB $0008,$0808,$0848,$3F00,$0000
*
UMICRO FDB $0018,$2424,$2424,$6200,$0000
*
PHI FDB $0010,$1038,$5454,$5438,$1010
*
KI FDB $0023,$1408,$1462,$0000,$0000
*
PSI FDB $0008,$0808,$1C2A,$2A2A,$4908
*
OMEGA FDB $0036,$4949,$4941,$2200,$0000
*
OMEGAM FDB $0063,$2222,$4141,$4122,$1C00
*
```

SOFTWARE CIRCLE, RING, AND DISK GENERATOR

The following program is based on Horn's algorithm (see figure 3). It allows radius values from 0 to FF (hexadecimal). The center of the circles, disks or rings can be placed anywhere within the 12 bits computing space of the GDP.

Two versions are given : the first is working with the EF9365 (square pixel), the second is working with the EF9366 (Pixel Height is twice Pixel width). Only one subroutine changes : PL0365 becomes PL0366 subroutine.

In the program, the three algorithms, which are very similar, are mixed and a test is done each time an operation is particular to one type of drawing.
The circle, ring and disk generator needs a list of parameters given in a 12 bytes RAM table : pointed by X register (of EF6800) for calling sequence.

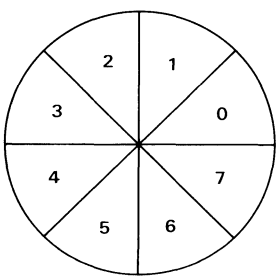
X →	X0 MSB	Center coordinates
	X0 LSB	
	Y0 MSB	
	Y0 LSB	
	Radius	
	Shape	00 if circle, 01 if ring, 02 if disk
	Thickness for the ring	
	Octant	
	Reserved	x
	Reserved	y
	Reserved	s MSB
X+ 11	Reserved	s LSB

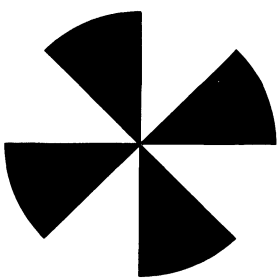
Horn's algorithm Internal variables

In byte "octant" are given octants to be lighted :
b0 = 0 octant 0 switched off, b0 = 1 lighted
b1 = 0 octant 1 switched off, b1 = 1 lighted.
⋮ ⋮

See figure 4 for octant localization.

Octant configuration





Example : Octant = 55 hexadecimal
Shape = 02 (disk).

FIGURE 4

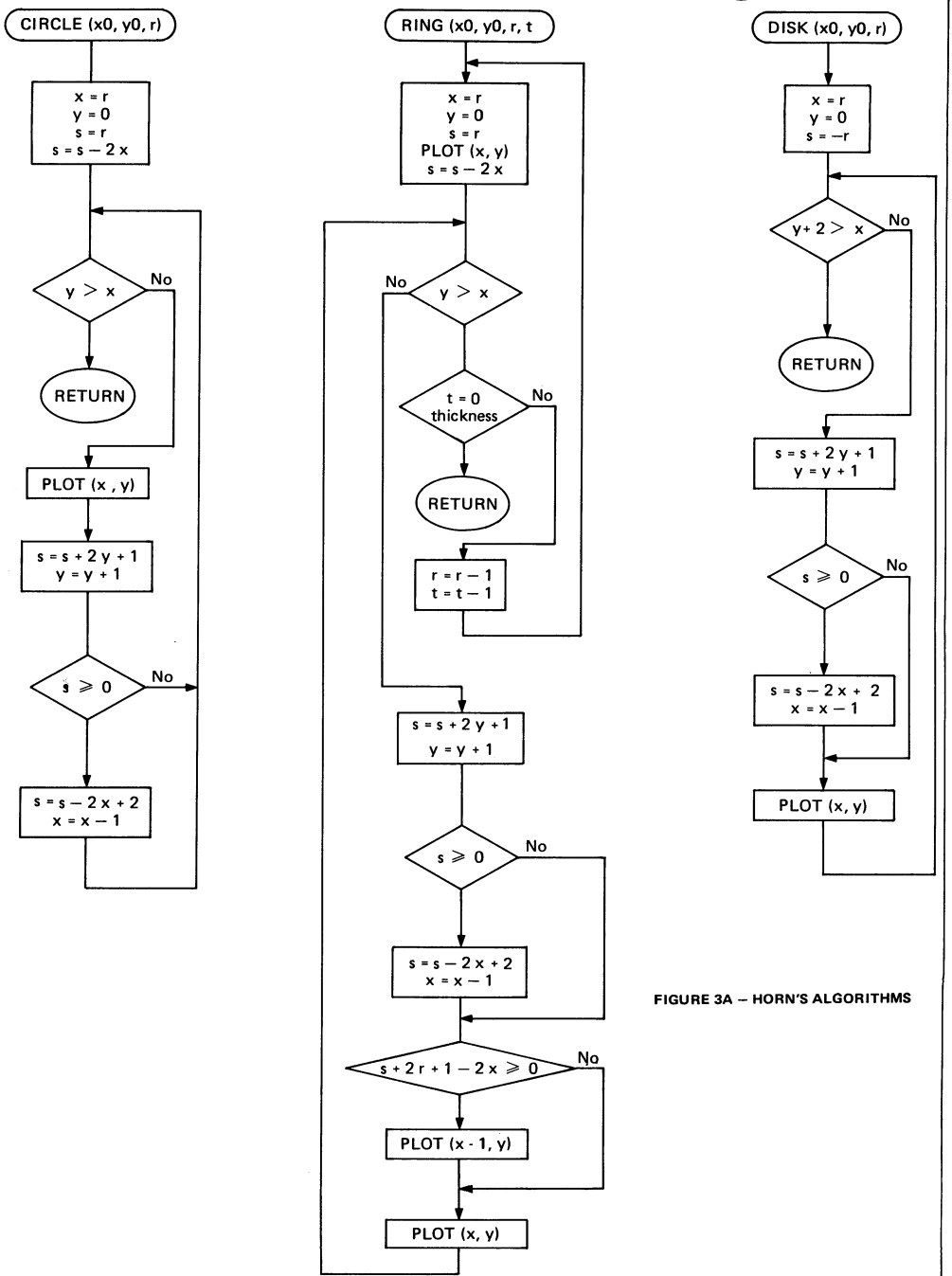
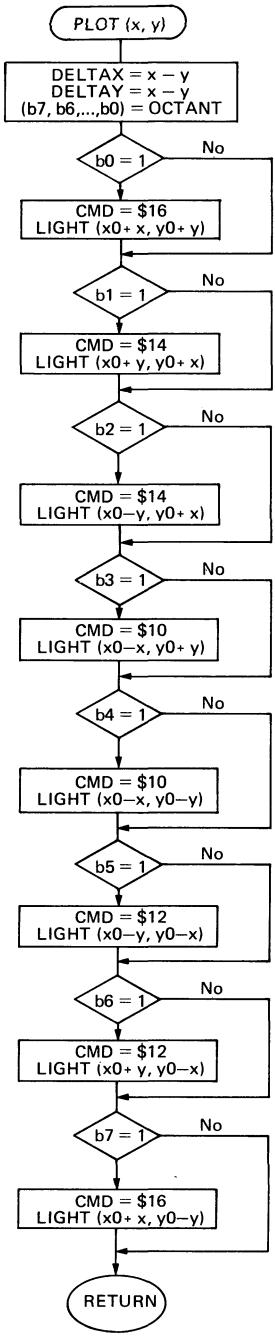


FIGURE 3A – HORN'S ALGORITHMS


PLOT (x, y)
DELTAX = x – y
DELTAY = x – y
(b7, b6,...,b0) = OCTANT
b0 = 1
No
CMD = $16
LIGHT (x0+ x, y0+ y)
b1 = 1
No
CMD = $14
LIGHT (x0+ y, y0+ x)
b2 = 1
No
CMD = $14
LIGHT (x0– y, y0+ x)
b3 = 1
No
CMD = $10
LIGHT (x0– x, y0+ y)
b4 = 1
No
CMD = $10
LIGHT (x0– x, y0– y)
b5 = 1
No
CMD = $12
LIGHT (x0– y, y0– x)
b6 = 1
No
CMD = $12
LIGHT (x0+ y, y0– x)
b7 = 1
No
CMD = $16
LIGHT (x0+ x, y0– y)
RETURN

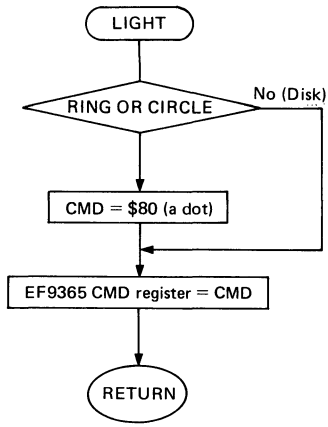

LIGHT
RING OR CIRCLE
No (Disk)
CMD = $80 (a dot)
EF9365 CMD register = CMD
RETURN

FIGURE 3B

```
PAGE  001  CERGDP  .SA:0


                        OPT    REL       SUBROUTINE GIVEN IN RELOCATABLE FORMAT
                      ************************************************************
                        XDEF   PCIRCL    ENTRY POINT
                      ************************************************************
                        XREF   CMD,MSBX,MSBY,DELTAX,DELTAY
                      ************************************************************
                      * EF9365 OR EF9366 REGISTERS ADDRESSES ARE DEFINED IN THE  *
                      * CALLING PROGRAM                                          *
                      ************************************************************
0000                    DSCT
                      ************************************************************
                      * 2 RAM BYTES ARE RESERVED IN SUBROUTINE AREA SO AS TO SAVE X
                      * X INITIAL VALUE IS PRESERVED USING THIS RAM LOCATION.
                      * BECAUSE OF THAT,CIRCLE SUBROUTINE IS NOT REENTRANT.
                      ************************************************************
0000    0002  A SAVX    RMB    2
                      ******************************************************
0000                    PSCT
                      *
                      *
                      *USEFUL SUBROUTINES*
                      ********************
                      *
        0000  P ADD16   EQU    *
                      ***********
                      *THIS SUBROUTINE ADDS P1 AND P2.
                      *16 BITS SIGNED PARAMETERS.
                      *INPUT STACK STATE (BEFORE ADD16 CALLING)
                      *       SP-(     )
                      *          ( P1H )
                      *          ( P1L )
                      *          ( P2H )
                      *          ( P2L )
                      *
                      *OUTPUT STACK STATE (AFTER RTS)
                      *          (     )
                      *          ( P1H )
                      *          ( P1L )
                      *          ( P2H )
                      *       SP-( P2L )
                      *WITH RESULT (P1+P2) LSB IN A,AND MSB IN B
                      *
                      ********************************************************************
                      * NO INPUT CONDITION ON A,B,X REGISTERS.X REGISTER MODIFIED *
                      * OUTPUT CONDITION:A CONTAINS P1+P2 LSB.B CONTAINS P1+P2 MSB*
                      ************************************************************
0000 30                 TSX
0001 A6 03      A       LDAA   3,X       P1L IN A
0003 AB 05      A       ADDA   5,X       P1L+P2L IN A
0005 E6 04      A       LDAB   4,X       P2H IN B
0007 C9 00      A       ADCB   £$0       P2H+C IN B
0009 EB 02      A       ADDB   2,X       P1H+P2H
```

```
PAGE  002  CLRGDP  .SA:O
000B EE 00    A         LDX    0,X       RETURN ADDRESS
000D 31                 INS
000E 31                 INS
000F 31                 INS
0010 31                 INS
0011 31                 INS
0012 31                 INS              STACK RESTORED
0013 6E 00    A         JMP    0,X       RETURN FROM SUBROUTINE
                        ***********
         0015 P SUB16   EQU    *
                        ***********
                        *THIS SUBROUTINES SUBSTRACTS P2 AND P1
                        *16 BITS SIGNED PARAMETERS.
                        *INPUT:STACK STATE IS THE SAME AS
                        *      INPUT STACK STATE OF ADD16.
                        *OUTPUT:STACK STATE IS THE SAME AS
                        *       OUTPUT STACK STATE OF ADD16
                        *       WITH RESULT=P2-P1.
                        ************************************************************
                        * NO INPUT CONDITIONS ON A,B,X REGISTERS. X REGISTER MODIFIED*
                        * OUTPUT CONDITION:A CONTAINS P2-P1  LSB.B CONTAINS P2-P1 MSB*
                        ************************************************************
0015 30                 TSX
0016 A6 05    A         LDAA   5,X       P2L IN A
0018 A0 03    A         SUBA   3,X       P2L-P1L IN A
001A E6 04    A         LDAB   4,X       P2H IN B
001C C2 00    A         SBCB   #$0       P2H-C=P2H
001E E0 02    A         SUBB   2,X       P2H-P1H IN B
0020 EE 00    A         LDX    0,X
0022 31                 INS
0023 31                 INS
0024 31                 INS
0025 31                 INS
0026 31                 INS
0027 31                 INS              STACK RESTORED
0028 6E 00    A         JMP    0,X       RETURN FROM SUBROUTINE
         002A P ADDIT   EQU    *         THIS SUBROUTINE ADDS 1 TO s PARAMETER
                        ***********
002A 6C 0B    A         INC    11,X      sL
002C 26 02 0030         BNE    STOP
002E 6C 0A    A         INC    10,X
0030 39           STOP  RTS
         0031 P ADXCOO  EQU    *
                        ***********
                        *THIS SUBROUTINE CALCULATES G.D.P X COORDINATE
                        *REGISTER IN ADDING x OR y WITH x0.
                        *INPUT PARAMETERS: IN A x OR y
0031 5F                 CLRB
0032 36                 PSHA             x or y IN STACK
0033 37                 PSHB             0 IN STACK
0034 A6 01    A         LDAA   1,X       x0L
0036 E6 00    A         LDAB   0,X       x0H
0038 36                 PSHA
0039 37                 PSHB
003A BD 0000  P         JSR    ADD16     x0+x LSB IN A,x0+x MSB IN B
003D FE 0000  D         LDX    SAVX
0040 F7 0000  A         STAB   MSBX
0043 B7 0001  A         STAA   MSBX+1
0046 39                 RTS
```

```
PAGE  003  CERGDP  .SA:0

                          ************
                0047   P ADYCOO EQU    *
                          ************
                          *THIS SUBROUTINE CALCULATES G.D.P COORDINATE
                          *REGISTER IN ADDING x OR y WITH yOL.
                          *INPUT PARAMETERS:IN A x OR y.
          0047 5F                 CLRB
          0048 36                 PSHA            x or y IN STACK
          0049 37                 PSHB
          004A A6 03     A        LDAA   3,X      yOL
          004C E6 02     A        LDAB   2,X      yOH
          004E 36                 PSHA
          004F 37                 PSHB
          0050 BD 0000   P        JSR    ADD16
          0053 FE 0000   D        LDX    SAVX
          0056 F7 0000   A        STAB   MSBY
          0059 B7 0001   A        STAA   MSBY+1
          005C 39                 RTS
                          ************
                005D   P SUXCOO EQU    *
                          ************
                          *THIS SUBROUTINE CALCULATES G.D.P X COORDINATE
                          *REGISTER IN SUBSTRACTING xO WITH x OR y.
                          *INPUT PARAMETERS:IN A x OR y.
          005D E6 01     A        LDAB   1,X      xOL
          005F 37                 PSHB
          0060 E6 00     A        LDAB   0,X      xOH
          0062 37                 PSHB
          0063 5F                 CLRB
          0064 36                 PSHA            x or y in STACK
          0065 37                 PSHB
          0066 BD 0015   P        JSR    SUB16
          0069 FE 0000   D        LDX    SAVX
          006C F7 0000   A        STAB   MSBX
          006F B7 0001   A        STAA   MSBX+1
          0072 39                 RTS
                          ************
                0073   P SUYCOO EQU    *
                          ************
                          *THIS SUBROUTINE CALCULATES G.D.P Y COORDINATE
                          *REGISTER IN SUBSTRACTING yO WITH x OR y.
                          *INPUT PARAMETERS:x OR y.
                          *                X IS PRESERVED.
          0073 E6 03     A        LDAB   3,X      yOL
          0075 37                 PSHB
          0076 E6 02     A        LDAB   2,X      yOH
          0078 37                 PSHB
          0079 5F                 CLRB
          007A 36                 PSHA            x or y
          007B 37                 PSHB
          007C BD 0015   P        JSR    SUB16
          007F FE 0000   D        LDX    SAVX
          0082 F7 0000   A        STAB   MSBY
          0085 B7 0001   A        STAA   MSBY+1
          0088 39                 RTS
```

```
PAGE  004  CERGDP  .SA:0


                    ************
          0089  P PL0365 EQU    *
                    ************
                    * THIS SUBROUTINE IS USED BY ALL CIRCLE,RING AND DISK MODES
                    * ONCE A DOT IS COMPUTED(x AND y),THIS SUBROUTINE LIGHTS
                    * DOTS ACCORDING TO THE COORDINATES OF THE CIRCLE CENTER.
                    * 8 DOTS ARE PLOTTED EACH TIME ONE DOT IS COMPUTED.
                    * DEDUCTION IS DONE BY SYMETRY.
                    * THIS SUBROUTINE TAKES CARE OF OCT INDICATOR TO KNOW WHICH
                    * OCTANTS ARE TO BE LIGHTED.
                    * NO INPUT CONDITION ON A,B REGISTERS
                    * X REGISTERS MUST CONTAIN THE ADDRESS OF THE PARAMETERS TABLE.
                    * X IS NOT MODIFIED ON OUTPUT,ALL OTHERS REGISTERS LOST.
                    ****************************************************************
0089 A6 08     A            LDAA   8,X        x
008B A0 09     A            SUBA   9,X        y x-y
008D B7 0000   A            STAA   DELTAX
0090 B7 0000   A            STAA   DELTAY
0093 A6 07     A            LDAA   7,X        OCTANT
0095 85 01     A            BITA   £$1
0097 27 0F 00A8             BEQ    OCT2
                    ******FIRST OCTANT******
0099 A6 08     A            LDAA   8,X        x
009B BD 0031   P            JSR    ADXC00     x+x0
009E A6 09     A            LDAA   9,X        y
00A0 BD 0047   P            JSR    ADYC00     y+y0
00A3 86 16     A            LDAA   £$16
00A5 BD 013C   P            JSR    LIGHT
00A8 A6 07     A OCT2       LDAA   7,X
00AA 85 02     A            BITA   £$2
00AC 27 0F 00BD             BEQ    OCT3
                    ******SECOND OCTANT******
00AE A6 09     A            LDAA   9,X        y
00B0 BD 0031   P            JSR    ADXC00     y+x0
00B3 A6 08     A            LDAA   8,X        x
00B5 BD 0047   P            JSR    ADYC00     x+y0
00B8 86 14     A            LDAA   £$14
00BA BD 013C   P            JSR    LIGHT
00BD A6 07     A OCT3       LDAA   7,X
00BF 85 04     A            BITA   £$4
00C1 27 0F 00D2             BEQ    OCT4
                    *******THIRD OCTANT*******
00C3 A6 09     A            LDAA   9,X        y
00C5 BD 005D   P            JSR    SUXC00     x0-y
00C8 A6 08     A            LDAA   8,X        x
00CA BD 0047   P            JSR    ADYC00     x+y0
00CD 86 14     A            LDAA   £$14
00CF BD 013C   P            JSR    LIGHT
```

```
PAGE  005  CERGDP  .SA:0


00D2 A6 07    A OCT4   LDAA   7,X
00D4 85 08    A        BITA   £$8
00D6 27 0F 00E7        BEQ    OCT5
                    *******FOURTH OCTANT*******
00D8 A6 08    A        LDAA   8,X       x
00DA BD 005D  P        JSR    SUXCOO    x0-x
00DD A6 09    A        LDAA   9,X       y
00DF BD 0047  P        JSR    ADYCOO    y+y0
00E2 86 10    A        LDAA   £$10
00E4 BD 013C  P        JSR    LIGHT
00E7 A6 07    A OCT5   LDAA   7,X
00E9 85 10    A        BITA   £$10
00EB 27 0F 00FC        BEQ    OCT6
                    *******FIFTH OCTANT*******
00ED A6 08    A        LDAA   8,X       x
00EF BD 005D  P        JSR    SUXCOO    x0-x
00F2 A6 09    A        LDAA   9,X       y
00F4 BD 0073  P        JSR    SUYCOO    y0-y
00F7 86 10    A        LDAA   £$10
00F9 BD 013C  P        JSR    LIGHT
00FC A6 07    A OCT6   LDAA   7,X
00FE 85 20    A        BITA   £$20
0100 27 0F 0111        BEQ    OCT7
                    ********SIXTH OCTANT********
0102 A6 09    A        LDAA   9,X       y
0104 BD 005D  P        JSR    SUXCOO    x0-y
0107 A6 08    A        LDAA   8,X       x
0109 BD 0073  P        JSR    SUYCOO    y0-x
010C 86 12    A        LDAA   £$12
010E BD 013C  P        JSR    LIGHT
0111 A6 07    A OCT7   LDAA   7,X
0113 85 40    A        BITA   £$40
0115 27 0F 0126        BEQ    OCT8
                    ********SEVENTH OCTANT********
0117 A6 09    A        LDAA   9,X       y
0119 BD 0031  P        JSR    ADXCOO    x0+y
011C A6 08    A        LDAA   8,X       x
011E BD 0073  P        JSR    SUYCOO    y0-x
0121 86 12    A        LDAA   £$12
0123 BD 013C  P        JSR    LIGHT
0126 A6 07    A OCT8   LDAA   7,X
0128 85 80    A        BITA   £$80
012A 27 0F 013B        BEQ    ENDPLO
                    *******EIGHTH OCTANT*******
012C A6 08    A        LDAA   8,X
012E BD 0031  P        JSR    ADXCOO    x0+x
0131 A6 09    A        LDAA   9,X       y
0133 BD 0073  P        JSR    SUYCOO    y0-y
0136 86 16    A        LDAA   £$16
0138 BD 013C  P        JSR    LIGHT
013B 39          ENDPLO RTS
```

```
PAGE  006  CERGDP  .SA:0


       013C  P LIGHT  EQU    *
                ***********
                * LIGHT SUBROUTINE LIGHTS A DOT IF IN CERCLE OR RING MODES,
                * AND WRITES A SEGMENT WHEN IN DISK MODE.
                * INPUT PARAMETERS: A REGISTER CONTAINS VECTOR CODE FOR DISK SEGMENT
                *           NO CONDITION ON B AND X REGISTERS
                *          DELTAX AND DELTAY REGISTERS MUST BE POSITONNED FOR DISKS
                *          X AND Y GDP REGISTERS MUST BE CORRECTLY POSITIONNED.
                * OUTPUT CONDITIONS:A,B REGISTERS MODIFIED,X PRESERVED
                ********************************************************************
013C E6 05    A         LDAB   5,X
013E C5 02    A         BITB   £$02      DISK?
0140 26 02 0144         BNE    DISK
0142 86 80    A         LDAA   £$80      COMMAND TO LIGHT A POINT
0144 B7 0000  A DISK    STAA   CMD       COMMAND REGISTER
0147 B6 0000  A TESTSR  LDAA   CMD       STATUS REGISTER
014A 85 04    A         BITA   £$4
014C 27 F9 0147         BEQ    TESTSR
014E 39                 RTS
                ********************************************************************
       014F  P PCIRCL EQU    *         CIRCLE ROUTINE ENTRY POINT
                ********************************************************************
                *THIS SUBROUTINE ALLOWS TO DRAW CIRCLE,RING,DISK OR
                *JUST SOME OCTANT OF THOSE ONE.
                *IT CALLS PL0365,LIGHT,ADDIT,SUXCOO,SUYCOO,ADXCOO,ADYCOO,
                *ADD16 AND SUB16 SUBROUTINES.
                *CIRCLE RING AND DISK ARE GENERATED WITH THE PRINCIPLE
                *OF HORN'S ALGORITHM.
                *INPUT PARAMETERS:NO CONDITION ON A,B REGISTERS
                *                 X HAS TO CONTAIN THE ADDRESS OF A 12
                *BYTES MEMORY LOCATION (TABLE):
                *TABLE-0,X ( xOH )
                *         1,X ( xOL )
                *         2,X ( yOH ) (xO,yO)CIRCLE,RING OR DISK
                *         3,X ( yOL ) CENTER COORDINATE.
                *         4,X (  r  ) CIRCLE,RING OR DISK RADIUS.
                *         5,X (SHAPE) OO:CIRCLE,O1:RING,O2:DISK.
                *         6,X (     ) RING'S THICKNESS.
                *         7,X ( OCT )
                *         8,X (  x  )
                *         9,X (  y  ) x,y,s ARE REQUISITE VARIABLES
                *        10,X ( sH  ) FOR HORN 'S ALGORITHM.
                *        11,X ( sL  )
                *IN BYTE ( OCT ) bO=O OCTANTO LIGHT OFF,bO=1 OCTANTO LIGHT ON.
                *                bl=O       1          bl=1       1
                *                  .                     .
                *                  .                     .
                ********************************************************************
```

```
PAGE  007  CERGDP  .SA:0


                    *VARIABLES INITIALISATION:
014F FF 0000  D          STX    SAVX
0152 A6 04    A          LDAA   4,X        r
0154 A7 08    A          STAA   8,X        x IS INITIALIZED WITH r.
0156 6F 09    A          CLR    9,X        y IS INITIALIZED WITH 0.
0158 6F 0A    A          CLR    10,X       sH
015A A7 0B    A          STAA   11,X       s IS INITIALISED WITH r.
015C A6 05    A          LDAA   5,X        FORME
015E 27 10 0170          BEQ    BOTH       IF CIRCLE
0160 85 02    A          BITA   £$2
0162 27 09 016D          BEQ    RING       IF RING
0164 60 0B    A          NEG    11,X
0166 86 FF    A          LDAA   £$FF
0168 A7 0A    A          STAA   10,X
016A 7E 022C  P          JMP    OUTDEF     IF DISK
016D BD 0089  P RING     JSR    PL0365     PLOT(x,y) WITH (x,y)=(r,0)
                    *****s=s-2x*****
0170 E6 0B    A BOTH     LDAB   11,X       sL
0172 37                  PSHB
0173 E6 0A    A          LDAB   10,X       sH
0175 37                  PSHB
0176 0C                  CLC
0177 5F                  CLRB
0178 A6 08    A          LDAA   8,X        x
017A 48                  ASLA              2xL IN A
017B C9 00    A          ADCB   £0         2xH IN B
017D 36                  PSHA
017E 37                  PSHB
017F BD 0015  P          JSR    SUB16
0182 FE 0000  D          LDX    SAVX
0185 E7 0A    A          STAB   10,X       s-2x MSB IN sH
0187 A7 0B    A          STAA   11,X       s-2x LSB IN sL
0189 A6 09    A TESTYX   LDAA   9,X        y IN A
018B E6 05    A          LDAB   5,X
018D C5 02    A          BITB   £$02
018F 27 02 0193          BEQ    SUITST
0191 8B 02    A          ADDA   £$2
0193 A1 08    A SUITST   CMPA   8,X        x
0195 23 03 019A          BLS    AFTER      IF y.LE.x AFTER
0197 7E 0232  P          JMP    ENDP       IF y.GT.x LAHAUT
019A A6 05    A AFTER    LDAA   5,X        FORME
019C 26 03 01A1          BNE    NOCIRC     RING
019E BD 0089  P          JSR    PL0365
```

```
PAGE  008  CERGDP  .SA:0


                          *****s=s+2y+1*****
01A1 0C                   NOCIRC CLC
01A2 5F                          CLRB
01A3 A6 09     A                 LDAA   9,X       y
01A5 48                          ASLA              2yL IN A
01A6 C9 00     A                 ADCB   #0        2yH IN B
01A8 36                          PSHA
01A9 37                          PSHB
01AA A6 0B     A                 LDAA   11,X      sL
01AC 36                          PSHA
01AD A6 0A     A                 LDAA   10,X      sH
01AF 36                          PSHA
01B0 BD 0000   P                 JSR    ADD16
01B3 FE 0000   D                 LDX    SAVX
01B6 E7 0A     A                 STAB   10,X      s+2y MSB IN sH
01B8 A7 0B     A                 STAA   11,X      s+2y LSB IN sL
01BA BD 002A   P                 JSR    ADDIT     s=s+1
                          *****y=y+1*****
01BD 6C 09     A                 INC    9,X
01BF A6 0A     A                 LDAA   10,X      sH
01C1 2B 27 01EA                  BMI    TSFORM    IF s.LT.0 TESTYX
                          *****s=s-2x+2*****
01C3 A6 0B     A                 LDAA   11,X      sL
01C5 36                          PSHA
01C6 A6 0A     A                 LDAA   10,X      sH
01C8 36                          PSHA
01C9 0C                          CLC
01CA 5F                          CLRB
01CB A6 08     A                 LDAA   8,X       x
01CD 48                          ASLA              2xL IN A
01CE C9 00     A                 ADCB   #0        2xH IN B
01D0 36                          PSHA
01D1 37                          PSHB
01D2 BD 0015   P                 JSR    SUB16
01D5 FE 0000   D                 LDX    SAVX
01D8 E7 0A     A                 STAB   10,X      s-2x MSB in sH
01DA A7 0B     A                 STAA   11,X      s-2x LSB IN sL
01DC BD 002A   P                 JSR    ADDIT     s=s+1
01DF BD 002A   P                 JSR    ADDIT     s=s+1
                          *****x=x-1*****
01E2 A6 08     A                 LDAA   8,X
01E4 80 01     A                 SUBA   #1
01E6 A7 08     A                 STAA   8,X
01E8 25 48 0232                  BCS    ENDP
01EA A6 05     A TSFORM          LDAA   5,X
01EC 27 41 022F                  BEQ    JUMP
```

```
PAGE  009  CERGDP  .SA:0


                          *****s-2x+2r+1*****
          01EE 85 01     A CIRCNO BITA   £$01
          01F0 27 3A 022C         BEQ    OUTDEF
          01F2 E6 0B     A        LDAB   11,X      sL
          01F4 37                 PSHB
          01F5 E6 0A     A        LDAB   10,X      sH
          01F7 37                 PSHB
          01F8 0C                 CLC
          01F9 5F                 CLRB
          01FA A6 08     A        LDAA   8,X       xL
          01FC 48                 ASLA             2xL IN A
          01FD C9 00     A        ADCB   £0        2xH IN B
          01FF 36                 PSHA
          0200 37                 PSHB
          0201 BD 0015   P        JSR    SUB16     s-2x
          0204 FE 0000   D        LDX    SAVX
          0207 36                 PSHA
          0208 37                 PSHB
          0209 5F                 CLRB
          020A A6 04     A        LDAA   4,X       r
          020C 48                 ASLA             2rL IN A
          020D C9 00     A        ADCB   £0        2rH IN B
          020F 36                 PSHA
          0210 37                 PSHB
          0211 BD 0000   P        JSR    ADD16     s-2x+2r
          0214 FE 0000   D        LDX    SAVX
          0217 8B 01     A        ADDA   £$1       s-2x+2r+1  LSB IN A
          0219 C9 00     A        ADCB   £0        s-2x+2r+1   MSB IN B
          021B C5 80     A        BITB   £$80
          021D 26 0D 022C         BNE    OUTDEF
          021F E6 08     A        LDAB   8,X
          0221 C0 01     A        SUBB   £1
          0223 E7 08     A        STAB   8,X
          0225 25 03 022A         BCS    IFNEG
          0227 BD 0089   P        JSR    PL0365    (x-1,y)
          022A 6C 08     A IFNEG  INC    8,X
          022C BD 0089   P OUTDEF JSR    PL0365    (x,y)
          022F 7E 0189   P JUMP   JMP    TESTYX
          0232 A6 06     A ENDP   LDAA   6,X       EPAISSEUR
          0234 27 11 0247         BEQ    ENDRIN
          0236 6A 06     A        DEC    6,X
          0238 6A 04     A        DEC    4,X       r-1
          023A A6 04     A        LDAA   4,X
          023C A7 08     A        STAA   8,X
          023E A7 0B     A        STAA   11,X
          0240 6F 09     A        CLR    9,X
          0242 6F 0A     A        CLR    10,X
          0244 7E 014F   P        JMP    PCIRCL
          0247 39          ENDRIN RTS
                                  END
    TOTAL ERRORS 00000--00000
```

```
PAGE  005  CER366  .SA:0


                    ************
          00E5   P PL0366 EQU    *
                    ************
                    * THIS SUBROUTINE IS EQUIVALENT TO PL0365 SUBROUTINE BUT
                    * WORKS WITH EF9366 CIRCUIT.ALL Y COORDINATES ARE DIVIDED BY TWO
                    * BEFORE PLOTTING THE VECTOR SO AS TO GET A CIRCLE.
00E5 A6 08      A           LDAA   8,X        x
00E7 A0 09      A           SUBA   9,X        y x-y
00E9 B7 F825    A           STAA   DELTAX
00EC 0C                     CLC                DIVIDE BY 2
00ED 44                     LSRA
00EE B7 F827    A           STAA   DELTAY
00F1 A6 07      A           LDAA   7,X         OCTANT
00F3 85 01      A           BITA   £$1
00F5 27 10 0107             BEQ    OCT2
                    ******FIRST OCTANT******
00F7 A6 08      A           LDAA   8,X         x
00F9 BD 0081    P           JSR    ADXCOO      x+x0
00FC A6 09      A           LDAA   9,X         Y
00FE 44                     LSRA               y/2
00FF BD 009A    P           JSR    ADYCOO      y+y0
0102 86 16      A           LDAA   £$16
0104 BD 01A3    P           JSR    LIGHT
0107 A6 07      A OCT2      LDAA   7,X
0109 85 02      A           BITA   £$2
010B 27 10 011D             BEQ    OCT3
                    ******SECOND OCTANT******
010D A6 09      A           LDAA   9,X         y
010F BD 0081    P           JSR    ADXCOO      y+x0
0112 A6 08      A           LDAA   8,X         x
0114 44                     LSRA               x/2
0115 BD 009A    P           JSR    ADYCOO      x+y0
0118 86 14      A           LDAA   £$14
011A BD 01A3    P           JSR    LIGHT
011D A6 07      A OCT3      LDAA   7,X
011F 85 04      A           BITA   £$4
0121 27 10 0133             BEQ    OCT4
                    ******THIRD OCTANT******
0123 A6 09      A           LDAA   9,X         y
0125 BD 00B3    P           JSR    SUXCOO      x0-y
0128 A6 08      A           LDAA   8,X         x
012A 44                     LSRA               x/2
012B BD 009A    P           JSR    ADYCOO      x+y0
012E 86 14      A           LDAA   £$14
0130 BD 01A3    P           JSR    LIGHT
```

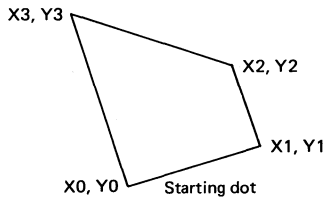
```
PAGE  006  CER366  .SA:0

0133 A6 07    A OCT4   LDAA   7,X
0135 85 08    A        BITA   £$8
0137 27 10 0149        BEQ    OCT5
                     ******FOURTH OCTANT******
0139 A6 08    A        LDAA   8,X      x
013B BD 00B3  P        JSR    SUXCOO   x0-x
013E A6 09    A        LDAA   9,X      y
0140 44                LSRA            y/2
0141 BD 009A  P        JSR    ADYCOO   y+y0
0144 86 10    A        LDAA   £$10
0146 BD 01A3  P        JSR    LIGHT
0149 A6 07    A OCT5   LDAA   7,X
014B 85 10    A        BITA   £$10
014D 27 11 0160        BEQ    OCT6
                     ******FIFTH OCTANT******
014F A6 08    A        LDAA   8,X      x
0151 BD 00B3  P        JSR    SUXCOO   x0-x
0154 A6 09    A        LDAA   9,X      y
0156 0C                CLC
0157 44                LSRA            x/2
0158 BD 00CC  P        JSR    SUYCOO   y0-y
015B 86 10    A        LDAA   £$10
015D BD 01A3  P        JSR    LIGHT
0160 A6 07    A OCT6   LDAA   7,X
0162 85 20    A        BITA   £$20
0164 27 10 0176        BEQ    OCT7
                     ******SIXTH OCTANT******
0166 A6 09    A        LDAA   9,X      y
0168 BD 00B3  P        JSR    SUXCOO   x0-y
016B A6 08    A        LDAA   8,X      x
016D 44                LSRA            x/2
016E BD 00CC  P        JSR    SUYCOO   y0-x
0171 86 12    A        LDAA   £$12
0173 BD 01A3  P        JSR    LIGHT
0176 A6 07    A OCT7   LDAA   7,X
0178 85 40    A        BITA   £$40
017A 27 10 018C        BEQ    OCT8
                     ******SEVENTH OCTANT******
017C A6 09    A        LDAA   9,X      y
017E BD 0081  P        JSR    ADXCOO   x0+y
0181 A6 08    A        LDAA   8,X      x
0183 44                LSRA            x/2
0184 BD 00CC  P        JSR    SUYCOO   y0-x
0187 86 12    A        LDAA   £$12
0189 BD 01A3  P        JSR    LIGHT
018C A6 07    A OCT8   LDAA   7,X
018E 85 80    A        BITA   £$80
0190 27 10 01A2        BEQ    ENDPLO
                     ******EIGHT OCTANT******
0192 A6 08    A        LDAA   8,X      x
0194 BD 0081  P        JSR    ADXCOO   x0+x
0197 A6 09    A        LDAA   9,X      y
0199 44                LSRA            y/2
019A BD 00CC  P        JSR    SUYCOO   y0-y
019D 86 16    A        LDAA   £$16
019F BD 01A3  P        JSR    LIGHT
01A2 39          ENDPLO RTS
```

QUADRILATERAL GENERATOR

The following program may be used to generate any convex quadrilateral. This quadrilateral can be filled or not. The quadrilateral is described by giving its four apexes beginning from the lowest dot of the quadrilateral and going anti-clockwise(see figure 5).

FIGURE 5



Because internal vector generator is able to draw vectors with only a 256 dots length, we choosed to draw the quadrilateral using software Bresenham Algorithm allowing X, Y, ΔX and ΔY to be 12 bits signed numbers.
Principle of the filling algorithm is as simple as possible.
The program draws from the starting dot in two direction from (X0, Y0) to (X1, Y1) (right side) and from (X0, Y0) to (X3, Y3) (left side). Each time a dot with a new Y coordinate is found on the right side, the dot with the same Y coordinate is computed on the left side.
Both dots are then bound together using internal vector generator. This principle is repeated until the highest dot is reached.

Figure 6 describes Bresenham algorithm
Figure 7 describes the detailed quadrilateral generator algorithm.

Calling sequence :

Before calling the QUADRI subroutine a 17 byte table must be positionned (figure 8).

	INDIC	
X →	X0	MSB
	X0	LSB
	Y0	MSB
	Y0	LSB
	X1	MSB
	X1	LSB
	Y1	MSB
	Y1	LSB
	X2	MSB
	X2	LSB
	Y2	MSB
	Y2	LSB
	X3	MSB
	X3	LSB
	Y3	MSB
X + 16 →	Y3	LSB

FIGURE 8 – PARAMETER TABLE

X register must point to the top of the table.

BRESENHAM ALGORITHM

The segment is defined by its origin X0, Y0 and its projections on x and y axis, ΔX, ΔY.
In order to make it clearer, we suppose $X0 = Y0 = 0$ and $\Delta X \geqslant \Delta Y \geqslant 0$.

```
BEGIN      X = 0 ; Y = 0 ; S = -ΔX/2

      WHILE      X < ΔX   DO

          BEGIN      BLACK (X, Y) ;
                     X = X + 1 ;
                     S = S + ΔY ;
                     if S ⩾ 0 THEN
                           BEGIN  Y = Y + 1 ;
                                  S = S – ΔX
                           END
          END
END.
```

BLACK (X, Y) plots the dot with X and Y coordinates.

FIGURE 6

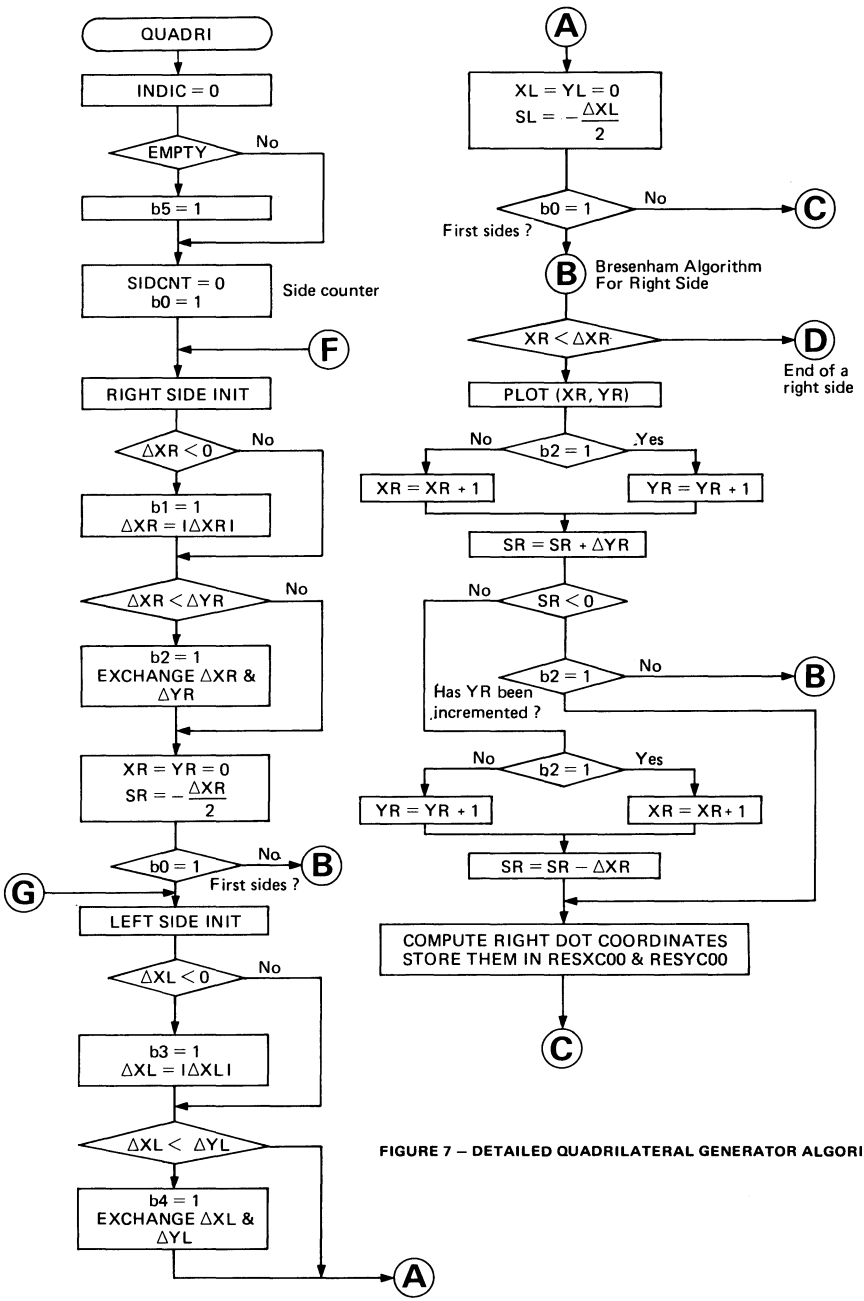


FIGURE 7 – DETAILED QUADRILATERAL GENERATOR ALGORITHM

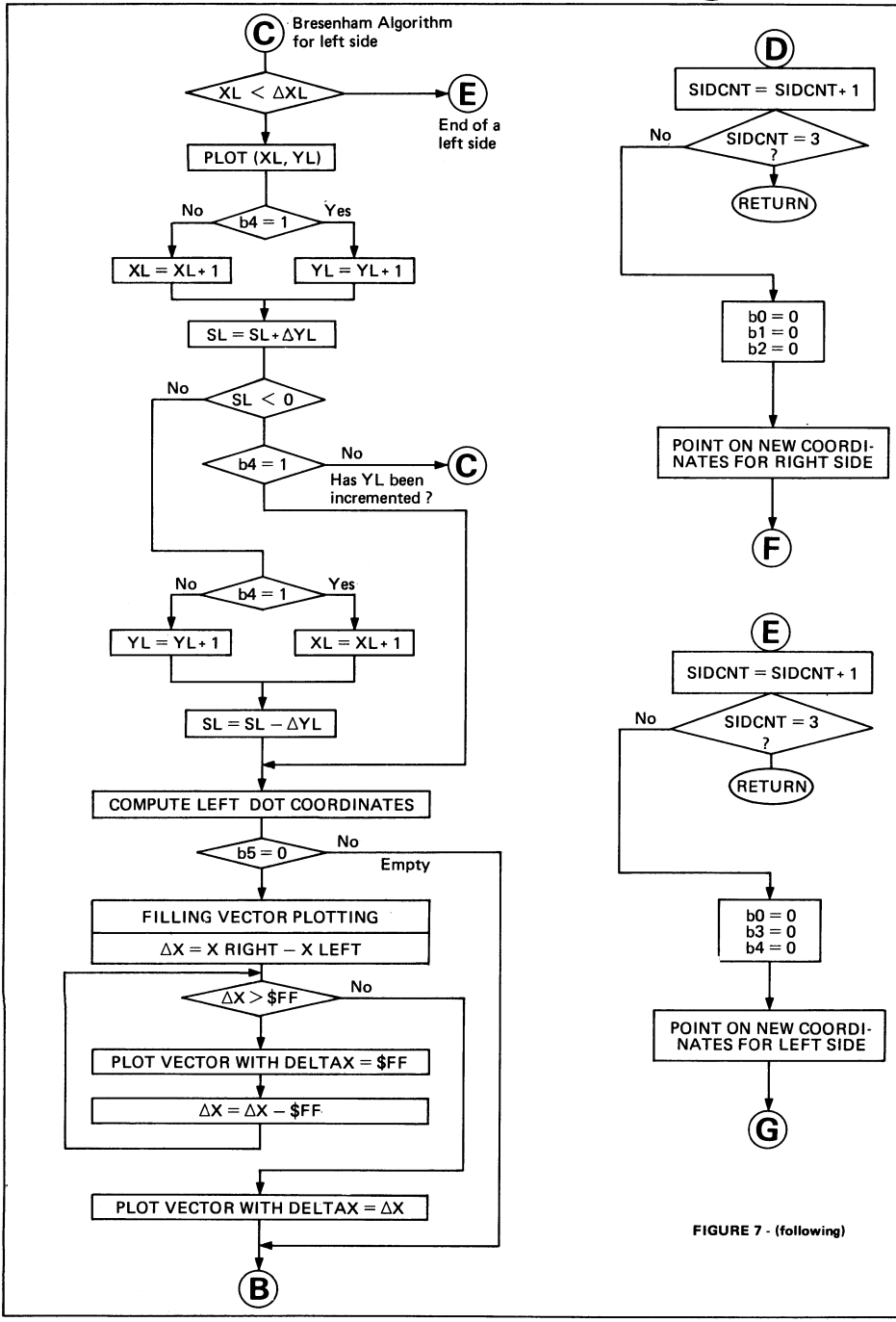


FIGURE 7 - (following)

```
001  QUADRO  .SA:0

                    OPT     REL       RELOCATABLE FORMAT
                    XREF    CMD,MSBX,MSBY,DELTAX,DELTAY EF9365 EXTERNALLY DEFINED
                    XDEF    QUADRI    ENTRY POINT
                    DSCT              THIS PROGRAM NEEDS THE FOLLOWING PARAMETERS
    0002  A SAVEX   RMB     2         X EF6800 REGISTER IS SAVED HERE
    0001  A SIDCNT  RMB     1         SIDE COUNTER
    0001  A INDIC   RMB     1         FLAGS REGISTER
    0002  A XR      RMB     2         BRESENHAM ALGORITHM PARAMETERS
    0002  A YR      RMB     2         FOR RIGHT SIDE
    0002  A SR      RMB     2
    0002  A DXR     RMB     2
    0002  A DYR     RMB     2
    0002  A XL      RMB     2         BRESENHAM ALGORITHM VARIABLES
    0002  A YL      RMB     2         FOR LEFT SIDE
    0002  A SL      RMB     2
    0002  A DXL     RMB     2
    0002  A DYL     RMB     2
    0002  A RESXCO  RMB     2         X AND Y COORDINATES FOR RSIDE DOT
    0002  A RESYCO  RMB     2         FOR TEMPORARY STORAGE
                    PSCT
              *************************
              *SOME USEFUL SUBROUTINES*
              *************************
    0000  P TRACE   EQU     *         SEND COMMAND TO EF9365 AND TEST IF READY
              *INPUT:A CONTAINS COMMAND TO BE SENT TO GDP
              *OUTPUT:A MODIFIED,ALL OTHER REGISTERS UNCHANGED
B7 0000   A         STAA    CMD
B6 0000   A TEST    LDAA    CMD
85 04     A         BITA    £$04
27 F9 0003          BEQ     TEST
39                  RTS
    000B  P ADD16   EQU     *         16 BITS ADDITION.PARAMETERS IN STACK
              *INPUT STACK STATE(BEFORE CALLING)
              *        SP-(     )
              *           ( P1H )
              *           ( P1L )
              *           ( P2H )
              *           ( P2L )
              * NO CONDITION ON A,B,X
              *OUTPUT:A CONTAINS P1+P2 MSB
              *        B CONTAINS P1+P2 LSB
              *        X MODIFIED
30                  TSX
E6 03     A         LDAB    3,X       P1L
EB 05     A         ADDB    5,X       P1L+P2L IN B
A6 04     A         LDAA    4,X       P2H
89 00     A         ADCA    £$0       P2H+C
AB 02     A         ADDA    2,X       P1H+P2H IN A
31                  INS               RESTORE STACK POINTER
31                  INS
31                  INS
31                  INS
31                  INS
31                  INS
EE 00     A         LDX     0,X       RETURN ADDRESS
6E 00     A         JMP     0,X       RETURN FROM SUBROUTINE
```

```
002  QUADRO  .SA:O


   0020  P SUB16  EQU    *         16 BITS SUBSTRACTION
         *SAME METHOD AS ADD16.RESULT IS P2-P1
30               TSX
E6 05    A       LDAB   5,X       P2L
EO 03    A       SUBB   3,X       P2L-P1L IN B
A6 04    A       LDAA   4,X       P2H
82 00    A       SBCA   £$O       P2H-C
AO 02    A       SUBA   2,X       P2H-P1H IN A
31               INS               RESTORE STACK POINTER
31               INS
31               INS
31               INS
31               INS
31               INS
EE 00    A       LDX    O,X       RETURN ADDRESS
6E 00    A       JMP    O,X       RETURN FROM SUBROUTINE
   0035  P VALABS EQU    *         ABSOLUTE VALUE OF A 12 BITS NUMBER.
         *INPUT: A CONTAINS NUMBER MSB
         *       B ---------------- LSB
         *OUTPUT:A CONTAINS RESULT MSB
         *       B ---------------- LSB
43               COMA              1 COMPLEMENT
53               COMB              1 COMPLEMENT
5C               INCB              2 COMPLEMENT FOR LSB
26 01 003B       BNE    VALEND     NO CARRY
4C               INCA              IF LSB =O
84 OF    A VALEND ANDA   £$OF       12 BITS RESULT
39               RTS
   003E  P CALCUO EQU    *         DX AND DY CALCULATION
         *INPUT:NO CONDITION ON A,B
         *      X CONTAINS ADDRESS IN TABLE FOR THE END OF THE VECTOR.
         *      (X POINTS ON XH FOR DX CALCULATION,ON YH FOR DY CALCULATION)
         *OUTPUT:A CONTAINS RESULT MSB
         *       B ---------------- LSB
A6 05    A       LDAA   5,X       X1H OR Y1H (OR X2H,OR...)
E6 06    A       LDAB   6,X       X1L OR Y1L (OR X2L,OR...)
37               PSHB
36               PSHA
A6 01    A       LDAA   1,X       XOH OR YOH (OR X1H,OR...)
E6 02    A       LDAB   2,X       XOL RO YOL (OR X1L,OR...)
37               PSHB
36               PSHA
FF 0000  D       STX    SAVEX
BD 0020  P       JSR    SUB16     X1-XO(OR X2-X1,OR X3-X2)
FE 0000  D       LDX    SAVEX
84 OF    A       ANDA   £$OF      12 BITS RESULT
39               RTS
```

```
003  QUADRO  .SA:O


   0056  P CALCU1 EQU    *         SAME AS CALCUO FOR X3-XO OR Y3-YO CALCULATION
A6 OD    A        LDAA   13,X      X3H OR Y3H
E6 OE    A        LDAB   14,X      X3L OR Y3L
37                PSHB
36                PSHA
A6 01    A        LDAA   1,X       XOH OR YOH
E6 02    A        LDAB   2,X       XOL OR YOL
37                PSHB
36                PSHA
FF 0000  D        STX    SAVEX
BD 0020  P        JSR    SUB16     X3-XO
FE 0000  D        LDX    SAVEX
84 OF    A        ANDA   £$OF      12 BITS RESULT
39                RTS
   006E  P CALCU2 EQU    *         SAME AS CALCUO FOR X2-X3 OR Y2-Y3 CALCULATION
A6 09    A        LDAA   9,X       X2H OR Y2H
E6 0A    A        LDAB   10,X      X2L OR Y2L
37                PSHB
36                PSHA
A6 0D    A        LDAA   13,X      X3H OR Y3H
E6 0E    A        LDAB   14,X      X3L OR Y3L
37                PSHB
36                PSHA
FF 0000  D        STX    SAVEX
BD 0020  P        JSR    SUB16     X2-X3
FE 0000  D        LDX    SAVEX
84 OF    A        ANDA   £$OF      12 BITS RESULT
39                RTS
   0086  P ADDXR1 EQU    *         XR=XR+1 CALCULATION
7C 0005  D        INC    XR+1
26 03 008E        BNE    NOC
7C 0004  D        INC    XR
39          NOC   RTS
   008F  P ADDYR1 EQU    *         YR=YR+1 CALCULATION
7C 0007  D        INC    YR+1
26 03 0097        BNE    NOC1
7C 0006  D        INC    YR
39          NOC1  RTS
   0098  P ADDXL1 EQU    *         XL=XL+1 CALCULATION
7C 000F  D        INC    XL+1
26 03 00A0        BNE    NOC2
7C 000E  D        INC    XL
39          NOC2  RTS
   00A1  P ADDYL1 EQU    *         YL=YL+1 CALCULATION
7C 0011  D        INC    YL+1
26 03 00A9        BNE    NOC3
7C 0010  D        INC    YL
39          NOC3  RTS
```

```
004  QUADRO  .SA:0

00AA  P PLOT   EQU    *        COMPUTES X AND Y COORDINATES AND PUT THEM IN GDP
         * INPUT STACK STATE: SP-(      )
         *                      ( YOH )
         *                      ( YOL )  LEFT SIDE ORIGIN
         *                      ( XOH )
         *                      ( XOL )
         *                      (      )
         *                      (      )  RIGHT SIDE  ORIGIN
         *                      (      )
         *                      (      )
30                TSX
B6 0003  D        LDAA   INDIC
85 80    A        BITA   £$80      IS IT A RSIDE ?
27 14 00C6        BEQ    RSIDE0
         *LSIDE COORDINATES. BEGIN XO+(OR -)XL CALCULATION*
E6 05    A        LDAB   5,X       XL LSB
37                PSHB
E6 04    A        LDAB   4,X       XL MSB
37                PSHB
F6 000F  D        LDAB   XL+1
37                PSHB
F6 000E  D        LDAB   XL
37                PSHB
85 08    A        BITA   £$08      DXL.LT.0 ?
26 19 00DD        BNE    ISSUB     IF DX.LT.0 THEN XO-XL CALCULATION
20 12 00D8        BRA    ISADD     IF DX.GE.0 THEN XO+XL CALCULATION
         *RSIDE COORDINATES. BEGIN XO+(OR -)XR CALCULATION*
E6 09    A RSIDE0 LDAB   9,X       RIGHT SIDE ORIGIN X LSB
37                PSHB
E6 08    A        LDAB   8,X       MSB
37                PSHB
F6 0005  D        LDAB   XR+1
37                PSHB
F6 0004  D        LDAB   XR
37                PSHB
85 02    A        BITA   £$02      DXR.LT.0 ?
26 05 00DD        BNE    ISSUB     IF DXR.LT.0 THEN XO-XR
BD 000B  P ISADD  JSR    ADD16     IF DXR.GT.0 THEN XO+XR
20 03 00E0        BRA    EXIT
BD 0020  P ISSUB  JSR    SUB16
B7 0000  A EXIT   STAA   MSBX
F7 0001  A        STAB   MSBX+1
30                TSX              UPDATING X
B6 0003  D        LDAA   INDIC
85 80    A        BITA   £$80      IS IT A RIGHT SIDE?
27 10 00FE        BEQ -  RSIDE2
         *LSIDE Y COORDINATES.START YO+(OR -)Y CALCULATION
E6 03    A        LDAB   3,X       LEFT SIDE ORIGIN Y COORDINATE LSB
37                PSHB
E6 02    A        LDAB   2,X       MSB
37                PSHB
F6 0011  D        LDAB   YL+1
37                PSHB
F6 0010  D        LDAB   YL
37                PSHB
20 0E 010C        BRA    GOMSBY
```

```
 005  QUADRO  .SA:0


          *RSIDE Y COORDINATES CALCULATION
E6 07    A RSIDE2 LDAB   7,X      RIGHT SIDE ORIGIN Y COORDINATE LSB
37                PSHB
E6 06    A        LDAB   6,X      MSB
37                PSHB
F6 0007  D        LDAB   YR+1
37                PSHB
F6 0006  D        LDAB   YR
37                PSHB
BD 000B  P GOMSBY JSR    ADD16    YO+Y
B7 0000  A        STAA   MSBY
F7 0001  A        STAB   MSBY+1
FE 0000  D        LDX    SAVEX    RESTORE X AFTER ADD16
39                RTS
          ************************************************************************
          * EMPTY OR FULL QUADRILATERAL GENERATOR *
          *****************************************
          * THIS SUBROUTINE MAY BE USED TO DRAW ANY QUADRILATERAL
          * EACH QUADRILATERAL IS DEFINED BY A 17 BYTE TABLE
          * WHICH MUST BE POINTED TO BY X EF6800 REGISTER BEFORE CALLING QUADRI
          * TABLE ORGANISATION: 0,X - ( FLAG )      9,X - ( X2H )
          *                     1,X - ( XOH )      10,X - ( X2L )
          *                     2,X - ( XOL )      11,X - ( Y2H )
          *                     3,X - ( YOH )      12,X - ( Y2L )
          *                     4,X - ( YOL )      13,X - ( X3H )
          *                     5,X - ( X1H )      14,X - ( X3L )
          *                     6,X - ( X1L )      15,X - ( Y3H )
          *                     7,X - ( Y1H )      16,X - ( Y3L )
          *                     8,X - ( Y1L )
          *
          * (XO,YO) IS THE STARTING DOT OF THE QUADRILATERAL DRAWING
          * IT MUST BE THE LOWEST DOT OF THE QUADRILATERAL.
          * THEN (X1,Y1),(X2,Y2),(X3,Y3) ARE GIVEN IN RETROGRAD CLOCK WISE.
          * QUADRILATERAL MUST BE CONVEX.
          * FLAG GIVES THE FULL/EMPTY PARAMETER: FLAG=0 FULL
          *                                      FLAG=1 EMPTY
          *
          * THIS PROGRAM USES A FLAG REGISTER FOR INTERNAL SERVICING
          * THIS BYTE IS CALLED INDIC AND IS DEFINED AS FOLLOWS:
          * B0=1 FIRST SIDE          B0=0 FOLLOWING SIDES
          * B1=0 DXR.GE.0            B1=1 DXR.LT.0
          * B2=0 /DXR/.GE.DYR        B2=1 /DXR/.LT.DYR
          * B3=0 DXL.GE.0            B3=1 DXL.LT.0
          * B4=0 /DXL/.GE.DYL        B4=1 /DXL.LT.DYL
          * B5=0 FULL QUADRILATERAL  B5=1 EMPTY QUADRILATERAL
          * B6 NOT USED
          * B7=0 RIGHT SIDE          B7=1 LEFT SIDE
          *
          ************************************************
```

```
006  QUADRO   .SA:0


          0119  P QUADRI EQU    *          ENTRY POINT
    7F 0003  D         CLR    INDIC
    A6 00    A         LDAA   0,X        TEST IF EMPTY OR FULL
    27 08 0128         BEQ    FULL
    F6 0003  D         LDAB   INDIC
    CA 20    A         ORAB   £$20       SET B5 OF INDIC
    F7 0003  D         STAB   INDIC
          0128  P FULL   EQU    *
    7F 0002  D         CLR    SIDCNT     INIT SIDE COUNTER
    7C 0003  D NEXT    INC    INDIC      B0=1 FIRST SIDES
                 * RIGHT SIDE INIT *
    A6 02    A GRIGHT LDAA   2,X        XOL
    36                 PSHA
    A6 01    A         LDAA   1,X        XOH
    36                 PSHA
    A6 04    A         LDAA   4,X        YOL
    36                 PSHA
    A6 03    A         LDAA   3,X        YOH
    36                 PSHA
    34                 DES
    34                 DES
    34                 DES                DON'T CHANGE LSIDE STARTING DOT
    34                 DES
    08                 INX
    08                 INX
    BD 003E  P         JSR    CALCUO     Y1-Y0 COMPUTATION
    B7 000C  D         STAA   DYR
    F7 000D  D         STAB   DYR+1
    09                 DEX                UPDATING X
    09                 DEX
    BD 003E  P         JSR    CALCUO     X1-X0 COMPUTATION
    F7 000B  D         STAB   DXR+1
    B7 000A  D         STAA   DXR
```

```
007  QUADRO  .SA:0


           *INDIC BYTE SETTING FOR RIGHT SIDE*
85 08    A RIGHTI BITA   £$08
27 14 016C        BEQ    POSIT    IF DXR.GE.0 NEXT TEST IN POSIT
F6 0003  D        LDAB   INDIC    IF DXR.LT.0 THEN SET B1=1
CA 02    A        ORAB   £$02
F7 0003  D        STAB   INDIC
F6 000B  D        LDAB   DXR+1
BD 0035  P        JSR    VALABS   COMPUTE /DXR/
B7 000A  D        STAA   DXR
F7 000B  D        STAB   DXR+1
B1 000C  D POSIT  CMPA   DYR
22 27 0198        BHI    XRYRSR   DXR.GT.DYR
26 05 0178        BNE    EXCH     DXR.LT.DYR
F1 000D  D        CMPB   DYR+1
22 20 0198        BHI    XRYRSR   DXR.GT.DYR
B6 0003  D EXCH   LDAA   INDIC    IF DXR.LT.DYR
8A 04    A        ORAA   £$04     THEN SET B2=1
B7 0003  D        STAA   INDIC
B6 000B  D        LDAA   DXR+1    EXCHANGE DXR AND DYR FOR BRESENHAM ALGORITHM
F6 000D  D        LDAB   DYR+1
B7 000D  D        STAA   DYR+1
F7 000B  D        STAB   DXR+1
B6 000A  D        LDAA   DXR
F6 000C  D        LDAB   DYR
B7 000C  D        STAA   DYR
F7 000A  D        STAB   DXR
7F 0004  D XRYRSR CLR    XR       XR=YR=0 INIT FOR BRESENHAM
7F 0005  D        CLR    XR+1
7F 0007  D        CLR    YR+1
7F 0006  D        CLR    YR
BD 00AA  P        JSR    PLOT     INIT GDP REGISTERS ON STARTING DOT
86 80    A        LDAA   £$80
BD 0000  P        JSR    TRACE    PLOT FIRST DOT .
B6 000B  D SRCOMP LDAA   DXR+1    COMPUTE SR=-DXR/2**************
0C                CLC
46                RORA
40                NEGA
B7 0009  D        STAA   SR+1
86 FF    A        LDAA   £$FF
B7 0008  D        STAA   SR
B6 0003  D        LDAA   INDIC
85 01    A        BITA   £$01      IS IT THE FIRST RIGHT SIDE?
26 03 01C4        BNE    FIRST     IF IT IS GOTO LEFT SIDE INIT
7E 0279  P        JMP    RSIDE     IF NOT GOTO BRESENHAM ALGORITHM FOR RIGHT SIDE
```

```
008  QUADRO  .SA:0


  08          FIRST  INX             LEFT SIDE INIT***************
  08                 INX
  BD 0056  P         JSR     CALCU1  Y3-YO
  B7 0016  D         STAA    DYL
  F7 0017  D         STAB    DYL+1
  09                 DEX             NEW X POINTER
  09                 DEX
  BD 0056  P         JSR     CALCU1  X3-XO
  B7 0014  D         STAA    DXL
  F7 0015  D         STAB    DXL+1
  31                 INS             DON'T MODIFY RIGHT SIDE ORIGIN
  31                 INS
  31                 INS
  31                 INS
  A6 02    A         LDAA    2,X     XOL    STORE LEFT SIDE ORIGIN
  36                 PSHA
  A6 01    A         LDAA    1,X     XOH
  36                 PSHA
  A6 04    A         LDAA    4,X     YOL
  36                 PSHA
  A6 03    A         LDAA    3,X     YOH
  36                 PSHA
  20 22 020E         BRA     LEFTI
  08          GOLEFT INX             LEFT SIDE INIT IF NOT THE FIRST SIDE
  08                 INX
  BD 006E  P         JSR     CALCU2  Y2-Y3 (IF 2ND SIDE),Y1-Y2 (IF THIRD)
  B7 0016  D         STAA    DYL
  F7 0017  D         STAB    DYL+1
  09                 DEX             NEW X POINTER
  09                 DEX
  BD 006E  P         JSR     CALCU2  X2-X3(IF 2ND SIDE) , X1-X2 (IF THIRD)
  B7 0014  D         STAA    DXL
  F7 0015  D         STAB    DXL+1
  A6 0E    A         LDAA    14,X    X3L (OR X2L)    STORE LEFT SIDE ORIGIN
  36                 PSHA
  A6 0D    A         LDAA    13,X    X3H (OR X2H)
  36                 PSHA
  A6 10    A         LDAA    16,X    Y3L (OR Y2L)
  36                 PSHA
  A6 0F    A         LDAA    15,X    Y3H (OR Y2H)
  36                 PSHA
  B6 0014  D LEFTI   LDAA    DXL
  85 08    A         BITA    £$08
  27 14 0229         BEQ     POSITL  IF DX LEFT.GE.0 GOTO POSITL
  F6 0003  D         LDAB    INDIC   IF DX.LT.0  THEN B3=1
  CA 08    A         ORAB    £$08
  F7 0003  D         STAB    INDIC
  F6 0015  D         LDAB    DXL+1
  BD 0035  P         JSR     VALABS  AND DX=/DX/
  B7 0014  D         STAA    DXL
  F7 0015  D         STAB    DXL+1
```

```
009  QUADRO  .SA:0
       B1 0016  D POSITL CMPA   DYL
       22 27 0255        BHI    XLYLSL    IF DX.GT.DY NO EXCHANGE
       26 05 0235        BNE    EXCHAN    IF DX.LT.DY EXCHANGE DX AND DY
       F1 0017  D        CMPB   DYL+1
       22 20 0255        BHI    XLYLSL
       B6 0003  D EXCHAN LDAA   INDIC     AND SET B4=1
       8A 10    A        ORAA   £$10
       B7 0003  D        STAA   INDIC
       B6 0015  D        LDAA   DXL+1      DX AND DY ARE EXCHANGED HERE
       F6 0017  D        LDAB   DYL+1
       B7 0017  D        STAA   DYL+1
       F7 0015  D        STAB   DXL+1
       B6 0014  D        LDAA   DXL
       F6 0016  D        LDAB   DYL
       F7 0014  D        STAB   DXL
       B7 0016  D        STAA   DYL
       7F 000E  D XLYLSL CLR    XL         XL=YL=0
       7F 000F  D        CLR    XL+1
       7F 0010  D        CLR    YL
       7F 0011  D        CLR    YL+1
       B6 0015  D SLCOMP LDAA   DXL+1      SL=-DXL/2 CALCULATION*******
       0C                CLC
       46                RORA
       40                NEGA
       B7 0013  D        STAA   SL+1
       86 FF    A        LDAA   £$FF
       B7 0012  D        STAA   SL
       B6 0003  D        LDAA   INDIC
       95 01    A        BITA   £$01       FIRST SIDES ?
       26 03 0279        BNE    RSIDE      IF B0=1 RIGHT SIDE BRESENHAM
       7E 0326  P        JMP    LSIDE      IF B0=0 LEFT SIDE BRESENHAM
       F6 0003  D RSIDE  LDAB   INDIC      RIGHT SIDE PLOTTING***********
       C5 04    A        BITB   £$04       TEST B2
       27 10 0290        BEQ    NOECH1     IF B2=0 DX.GE.DY;NO EXCHANGE
       B6 0006  D        LDAA   YR         COMPARE YR AND DYR
       B1 000A  D        CMPA   DXR
       26 18 02A0        BNE    SAMER      IF YR.LT.DXR
       B6 0007  D        LDAA   YR+1       COMPARE LSB
       20 0E 029B        BRA    COMPA
       7E 0413  P NEWR   JMP    TESTRI     END OF A RIGHT SIDE
       B6 0004  D NOECH1 LDAA   XR         COMPARE XR AND DXR
       B1 000A  D        CMPA   DXR
       26 08 02A0        BNE    SAMER      XR.LT.DXR
       B6 0005  D        LDAA   XR+1       COMPARE LSB
       B1 000B  D COMPA  CMPA   DXR+1
       27 ED 028D        BEQ    NEWR       IF XR.EQ.DXR
       C4 7F    A SAMER  ANDB   £$7F       RIGHT SIDE INDICATOR
       F7 0003  D        STAB   INDIC
       BD 00AA  P        JSR    PLOT       COMPUTE DOT COORDINATES
       86 80    A        LDAA   £$80       DOT COMMAND
       BD 0000  P        JSR    TRACE      PLOT IT
       B6 0003  D        LDAA   INDIC
       85 04    A        BITA   £$04       TEST B2
       26 05 02B9        BNE    INCYR      IF B2=1
       BD 0086  P        JSR    ADDXR1     XR=XR+1
       20 03 02BC        BRA    AFTINC
       BD 008F  P INCYR  JSR    ADDYR1     YR=YR+1
```

```
010  QUADRO  .SA:0


     B6 0008  D AFTINC LDAA   SR        SR=SR+DYR CALCULATION***********
     F6 0009  D        LDAB   SR+1
     37                PSHB
     36                PSHA
     B6 000C  D        LDAA   DYR
     F6 000D  D        LDAB   DYR+1
     37                PSHB
     36                PSHA
     BD 000B  P        JSR    ADD16
     F7 0009  D        STAB   SR+1
     F6 0003  D        LDAB   INDIC
     B7 0008  D        STAA   SR        SR IS A 16 BIT SIGNED NUMBER
     2A 0A 02E4        BPL    IFSRGE
     FE 0000  D        LDX    SAVEX
     C5 04    A        BITB   £$04      TEST B2
     26 28 0309        BNE    MEMXY     IF B2=1
     7E 0279  P        JMP    RSIDE     IF B2=0  LOOP
     C5 04    A IFSRGE BITB   £$04      IF SR.GE.0
     26 05 02ED        BNE    INCXR     IF B2=1
     BD 008F  P        JSR    ADDYR1    YR=YR+1
     20 03 02F0        BRA    AFINCR
     BD 0086  P INCXR  JSR    ADDXR1    XR=XR+1
     B6 0008  D AFINCR LDAA   SR        SR=SR-DXR CALCULATION*********
     F6 0009  D        LDAB   SR+1
     37                PSHB
     36                PSHA
     B6 000A  D        LDAA   DXR
     F6 000B  D        LDAB   DXR+1
     37                PSHB
     36                PSHA
     BD 0020  P        JSR    SUB16
     B7 0008  D        STAA   SR
     F7 0009  D        STAB   SR+1
     B6 0003  D MEMXY  LDAA   INDIC
     84 7F    A        ANDA   £$7F      RIGHT SIDE
     B7 0003  D        STAA   INDIC
     BD 00AA  P        JSR    PLOT      DOT COORDINATES
     B7 001A  D        STAA   RESYCO    STORE THEM
     F7 001B  D        STAB   RESYCO+1
     B6 0000  A        LDAA   MSBX
     F6 0001  A        LDAB   MSBX+1
     B7 0018  D        STAA   RESXCO
     F7 0019  D        STAB   RESXCO+1
```

```
011  QUADRO  .SA:0


F6 0003  D LSIDE  LDAB   INDIC    LEFT SIDE BRESENHAM******
C5 10    A        BITB   £$10     DX.LT.DY ?
27 10 033D        BEQ    NOEXCH   IF DX.GE.DY NO EXCHANGE
B6 0010  D        LDAA   YL       IF EXCHANGE COMPARE YL WITH DXL
B1 0014  D        CMPA   DXL
26 18 034D        BNE    SAMEL    YL.LT.DYL CONTINUE
B6 0011  D        LDAA   YL+1     COMPARE LSB
20 0E 0348        BRA    COMPAR
7E 0437  P NEWL   JMP    TESTLE   IF XL.EQ.DXL END OF A LEFT SIDE
B6 000E  D NOEXCH LDAA   XL       IF NO EXCHANGE COMPARE XL WITH DXL
B1 0014  D        CMPA   DXL
26 08 034D        BNE    SAMEL    XL.LT.DXL CONTINUE
B6 000F  D        LDAA   XL+1     COMPARE LSB
B1 0015  D COMPAR CMPA   DXL+1
27 ED 033A        BEQ    NEWL
CA 80    A SAMEL  ORAB   £$80     B7=1 IT IS A LEFT SIDE
F7 0003  D        STAB   INDIC
BD 00AA  P        JSR    PLOT     COMPUTE DOT COORDINATES
86 80    A        LDAA   £$80     DOT COMMAND
BD 0000  P        JSR    TRACE    PLOT IT
B6 0003  D        LDAA   INDIC
85 10    A        BITA   £$10     TEST B4
26 05 0366        BNE    INCYL    IF B4=1 GOTO INCYL
BD 0098  P        JSR    ADDXL1   IF B4=0 XL=XL+1
20 03 0369        BRA    LNEXT
BD 00A1  P INCYL  JSR    ADDYL1   IF B4=1 YL=YL+1
B6 0012  D LNEXT  LDAA   SL       SL=SL+DYL CALCULATION*******
F6 0013  D        LDAB   SL+1
37                PSHB
36                PSHA
B6 0016  D        LDAA   DYL
F6 0017  D        LDAB   DYL+1
37                PSHB
36                PSHA
BD 000B  P        JSR    ADD16
F7 0013  D        STAB   SL+1
F6 0003  D        LDAB   INDIC
B7 0012  D        STAA   SL
2A 0A 0391        BPL    IFSLGE
FE 0000  D        LDX    SAVEX    IF SL.LT.0 TEST B4
C5 10    A        BITB   £$10
26 28 03B6        BNE    VECTOR   IF B4=1 COMPUTE X,Y COORDINATES FOR LEFT SIDE
7E 0326  P        JMP    LSIDE    IF B4=1 LOOP (YL HAS NOT BEEN INCREMENTED
C5 10    A IFSLGE BITB   £$10     IF SL.GE.0 CONTINUE
26 05 039A        BNE    INCXL    IF B4=1 GOTO INCXL
BD 00A1  P        JSR    ADDYL1   YL=YL+1
20 03 039D        BRA    FOLLOW
BD 0098  P INCXL  JSR    ADDXL1   XL=XL+1
```

```
012  QUADRO  .SA:0


 B6 0012  D FOLLOW LDAA   SL        SL=SL-DXL CALCULATION******
 F6 0013  D        LDAB   SL+1
 37                PSHB
 36                PSHA
 B6 0014  D        LDAA   DXL
 F6 0015  D        LDAB   DXL+1
 37                PSHB
 36                PSHA
 BD 0020  P        JSR    SUB16
 B7 0012  D        STAA   SL
 F7 0013  D        STAB   SL+1
 B6 0003  D VECTOR LDAA   INDIC     COMPUTE LEFT DOT COORDINATES
 8A 80    A        ORAA   £$80
 B7 0003  D        STAA   INDIC
 BD 00AA  P        JSR    PLOT
 B6 0003  D        LDAA   INDIC     IS IT AN EMPTY QUADRILATERAL
 85 20    A        BITA   £$20
 27 08 03D0        BEQ    FILL      IF IT IS TO BE FILLED
 7F 0000  A        CLR    DELTAX
 7F 0000  A        CLR    DELTAY
 20 38 0408        BRA    LOOP3
 B6 0019  D FILL   LDAA   RESXCO+1
 F6 0018  D        LDAB   RESXCO
 36                PSHA
 37                PSHB
 B6 0000  A        LDAA   MSBX      LEFT SIDE DOT X COORDINATE
 F6 0001  A        LDAB   MSBX+1
 37                PSHB
 36                PSHA
 BD 0020  P        JSR    SUB16     DELTAX OF THE FILLING VECTOR
 F7 0000  A        STAB   DELTAX    LSB DIRECTLY IN DELTAX EF9365 REGISTER
 84 0F    A        ANDA   £$0F      12 BIT NUMBER
 B7 0000  A        STAA   DELTAY    DELTAY IS USED AS TEMPORARY STORAGE AREA
 27 1B 0408 LOOP4  BEQ    LOOP3     IF DELTAX MSB =0 PLOT THE FILLING VECTOR
 7A 0000  A        DEC    DELTAY    DELTAX - $FF CALCULATION
 5C                INCB
 26 03 03F6        BNE    NOCARR
 7C 0000  A        INC    DELTAY
 86 FF    A NOCARR LDAA   £$FF      A $FF LONG VECTOR IS PLOTTED
 B7 0000  A        STAA   DELTAX
 86 10    A        LDAA   £$10      HORIZONTAL VECTOR COMMAND
 BD 0000  P        JSR    TRACE
 F7 0000  A        STAB   DELTAX    STORE INTERMEDIATE RESULT
 B6 0000  A        LDAA   DELTAY    TEST IF MSB=0
 20 E3 03EB        BRA    LOOP4
 FE 0000  D LOOP3  LDX    SAVEX
 86 10    A        LDAA   £$10      PLOT THE REMAINING LENGTH
 BD 0000  P        JSR    TRACE
 7E 0279  P        JMP    RSIDE     BACK TO BRESENHAM ALGORITHM FOR RIGHT SIDE
```

```
013  QUADRO  .SA:O


             ****** END OF A RIGHT SIDE ******
 31          TESTRI INS              RESTORE STACK POINTER
 31                 INS
 31                 INS
 31                 INS
 31                 INS
 31                 INS
 31                 INS
 31                 INS
 7C 0002  D         INC     SIDCNT
 B6 0002  D         LDAA    SIDCNT
 81 03    A         CMPA    £$03      IS IT THE END ?
 26 03 0428         BNE     NEXTSI
 7E 045E  P         JMP     ENDQUA    IF Y=YMAX EXIT FROM SUBROUTINE
 B6 0003  D NEXTSI  LDAA    INDIC     IF Y.LT.YMAX NEXT SIDE
 84 F8    A         ANDA    £$F8      INDIC INITIALISATION FOR NEXT RIGHT SIDE
 B7 0003  D         STAA    INDIC
 08                 INX
 08                 INX               UPDATE X
 08                 INX
 08                 INX
 7E 012E  P         JMP     GRIGHT
             ********* END OF A LEFT SIDE ********
 7C 0002  D TESTLE  INC     SIDCNT
 B6 0002  D         LDAA    SIDCNT
 81 03    A         CMPA    £$03
 27 17 0458         BEQ     ENDLEF
 B6 0003  D NEXTSL  LDAA    INDIC
 31                 INS               UPDATING STACK POINTER
 31                 INS
 31                 INS
 31                 INS
 85 01    A         BITA    £$01      WAS IT THE FIRST LEFT SIDE?
 26 04 0450         BNE     NEWIND    IF IT WAS
 09                 DEX               IF IT WAS NOT
 09                 DEX
 09                 DEX
 09                 DEX               UPDATE X
 84 E6    A NEWIND  ANDA    £$E6      INDIC INIT
 B7 0003  D         STAA    INDIC
 7E 01EC  P         JMP     GOLEFT    START A NEW LEFT SIDE
 86 08    A ENDLEF  LDAA    £$8       RESTORE STACK POINTER
 31          STAINI INS
 4A                 DECA
 26 FC 045A         BNE     STAINI
 39          ENDQUA RTS
                    END

ERRORS 00000--00000
```

BIBLIOGRAPHY

EF9365, EF9366 Data Sheet

EF6800 Programming Manual

Algorithm for computer control of a digital plotter - IBM system journal vol 4 n° 1

La Réalisation des logiciels graphiques interactifs CEA - EDF - INRIA - EYROLLES

Principles of Interactive Computer Graphics W. NEWMAN - RF SPROULL
Mac Graw Hill New York 1979

Filling algorithms for raster graphics T. PAULIDIS
Comp. Graphics and Image proc. 10 1979.

Circles generators for display devices, Computer Graphics and Image Processing 5, 1976, 280-288.
B.K.P. - HORN.

Algorithms for Generation of discrete circles, rings and disks, Computer Graphics and Image Processing
10, 1979, 366 - 371 MAREK DOROS.