Unit I Continued ....

Character Generation

- Basic Concept
- Stroke Principle
- Starburst Principle
- Bit map method
- Aliasing and anti-aliasing

Basic Concept

- Letters, numbers, and other characters are often used to annotate drawing and give instructions and information to the user.
- Most of the times characters are built into the graphics display devices, as a hardware but sometimes through software.
- There are basic three methods:
  - Stroke method
  - Starbust method
  - Bitmap method

1. Stroke method

- This method uses small line segments to generate a character.
- The small series of line segments are drawn like a strokes of a pen to form a character as shown in figure.
- We can build our own stroke method.
  - By calling a line drawing algorithm.
  - Here it is necessary to decide which line segments are needed for each character and
  - Then drawing these segments using line drawing algo.
- This method supports scaling of the character.
  - It does this by changing the length of the line segments used for character drawing.

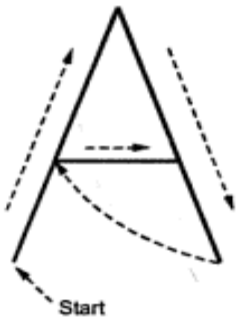
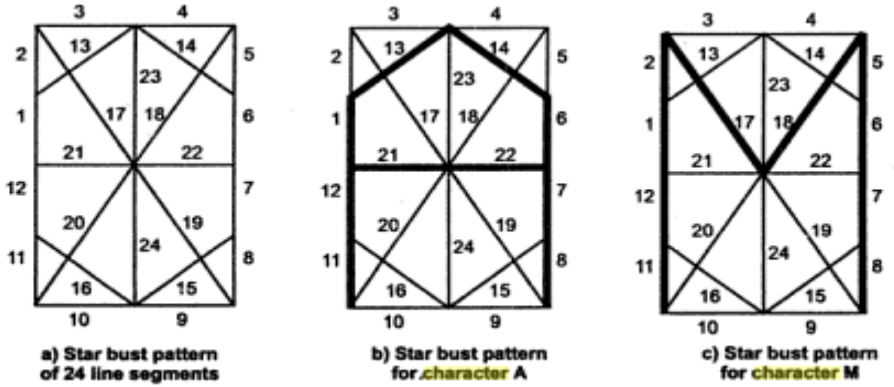


Fig. 2.31 Stroke method

2. Starburst method

- In this method a fix pattern of line segments are used to generate characters.
- As shown in figure, there are 24 line segments.
- Out of 24 line segments, segments required to display for particular character, are highlighted
- This method is called starburst method because of its characteristic appearance.

- Fig shows the starburst patterns for character A and M.
- The patterns for particular characters are stored in the form of 24 bits code.
- Each bit representing one line segment.
- The bit is set to one to highlight the line segment otherwise it is set to zero.



a) Star bust pattern of 24 line segments

b) Star bust pattern for character A

c) Star bust pattern for character M

Character A : 0011 0000 0011 1100 1110 0001
Character M:0000 0011 0000 1100 1111 0011

- This method of character generation is not used now a days because of following disadvantages:
  - The 24-bits are required to represent a character. Hence more memory is required.
  - Requires code conversion software to display character from its 24 bits code.
  - Character quality poor. Worst for curve shaped character.

3. Bitmap Method

- The third method for character generation.
- Also known as dot matrix because in this method characters are represented by an array of dots in the matrix form.
- It's a two dimensional array having columns and rows : 5 X 7 as shown in figure.
- 7 X 9 and 9 X 13 arrays are also used.
- Higher resolution devices may use character array 100 X 100.

- Each dot in the matrix is a pixel.
- The character is placed on the screen by copying pixel values from the character array into some position of the screen's frame buffer.
- Value of the pixel controls the intensity of the pixel.
- Usually the dot patterns for all characters are stored in the hardware device called a character generation chip.
- This chip accepts address for the character and gives the bit pattern for that character as an output.
- Here the size of the pixel is fixed and hence the size of the dot.
- Characters can be represented in many fonts.
- When number of fonts are more, the bit patterns for characters may also be stored in RAM.
- Anti-aliasing is possible in this method.

'C' code for Generation Text 'A'

(Softcopy of this program is available at vtubooks.com)

```
#include<stdio.h>
#include<graphics.h>
main()
{int gd,gm,i,j;

/* Save character map of letter A
------------------------------------------- */
int a[13][9] = {
        { 0, 0, 0, 0, 1, 0, 0, 0, 0},
        { 0, 0, 0, 1, 0, 1, 0, 0, 0},
        { 0, 0, 1, 0, 0, 0, 1, 0, 0},
        { 0, 1, 0, 0, 0, 0, 0, 1, 0},
        { 0, 1, 0, 0, 0, 0, 0, 1, 0},
        { 0, 1, 0, 0, 0, 0, 0, 1, 0},
        { 0, 1, 1, 1, 1, 1, 1, 1, 0},
        { 0, 1, 0, 0, 0, 0, 0, 1, 0},
        { 0, 1, 0, 0, 0, 0, 0, 1, 0},
        { 0, 1, 0, 0, 0, 0, 0, 1, 0},
        { 0, 1, 0, 0, 0, 0, 0, 1, 0},
        { 0, 1, 0, 0, 0, 0, 0, 1, 0},
        { 0, 1, 0, 0, 0, 0, 0, 1, 0},
         };
/* Initialise graphics mode
------------------------------------- */
detectgraph(&gd,&gm);
initgraph(&gd,&gm,"");
for(i=0;i<13;i++)
{
   for(j=0;j<9;j++)
   {
   putpixel(200+j,200+i,15*a[i][j]);
   }
}
getch();
closegraph();
}
```

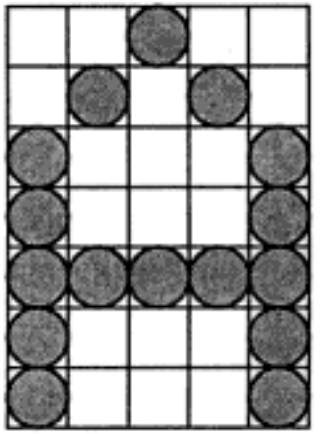


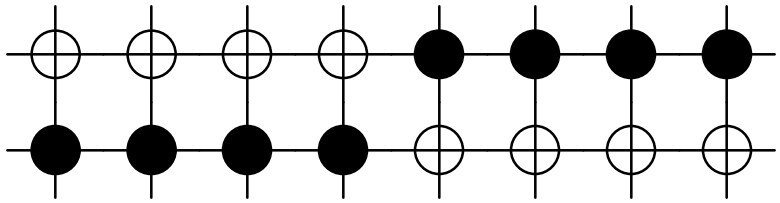
Fig. 2.33 Character A in 5 × 7 dot matrix format

Aliasing

- Aliasing is caused by the discrete nature of pixels *(Sampling Error)*.
- Approximation of lines and circles with discrete points often gives a staircase appearance or "Jaggies".
- Eg.

Desired line

Aliased rendering of the line



Anti-aliasing

- Aliasing can be smoothed out by using higher addressability.
- If addressability is fixed but intensity is variable, use the intensity to control the address of a "virtual pixel". Two adjacent pixels can be be used to give the impression of a point part way between them.
- An anti-aliased line has a series of virtual pixels each located at the proper address.