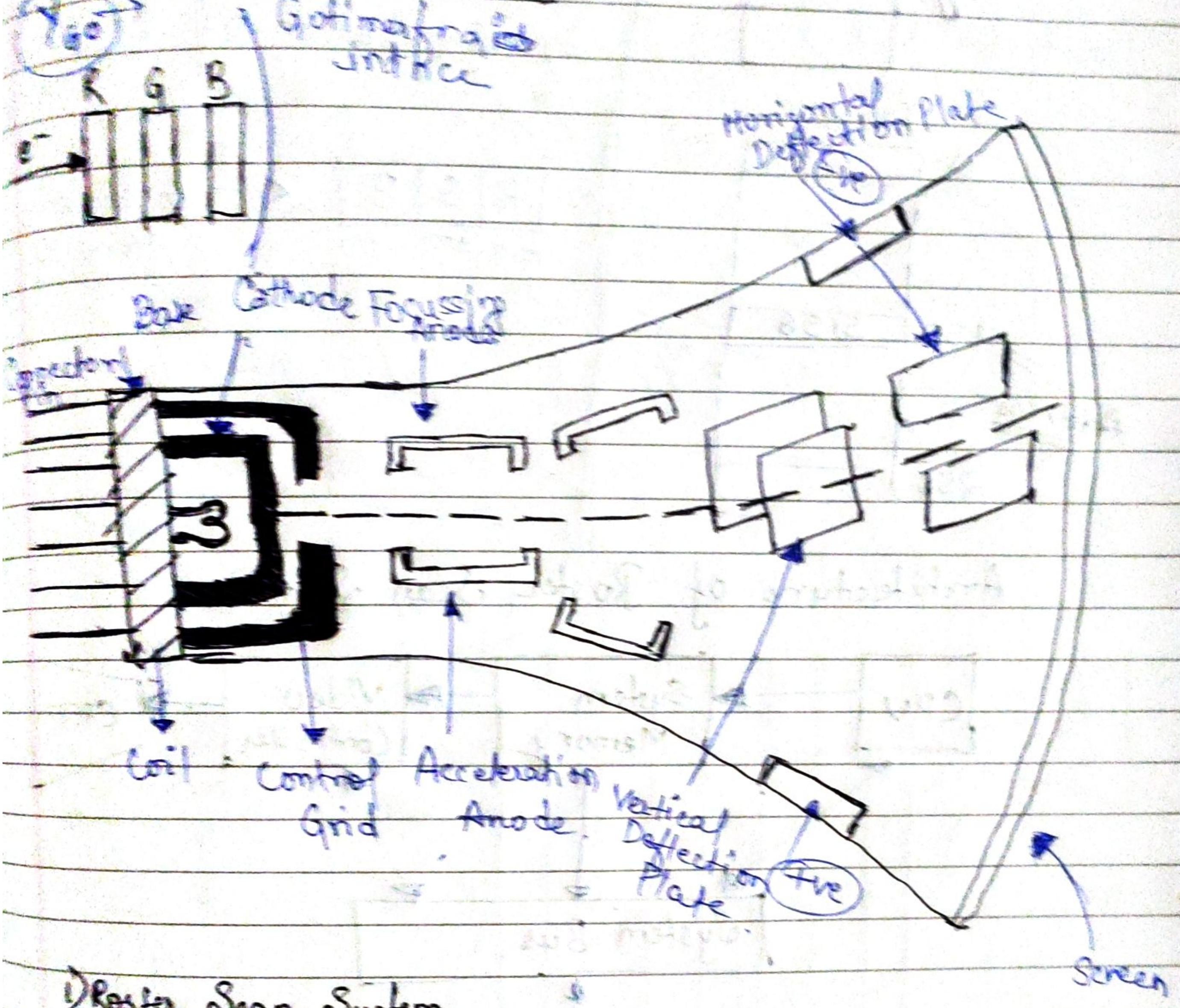
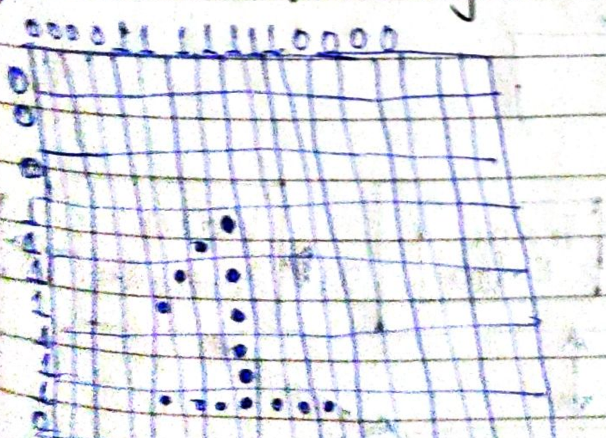
COMPUTER GRAPHICS

CHAPTER-1



1) Raster Scan System
2) Random Scan System.



Randomwise Pixel checky.

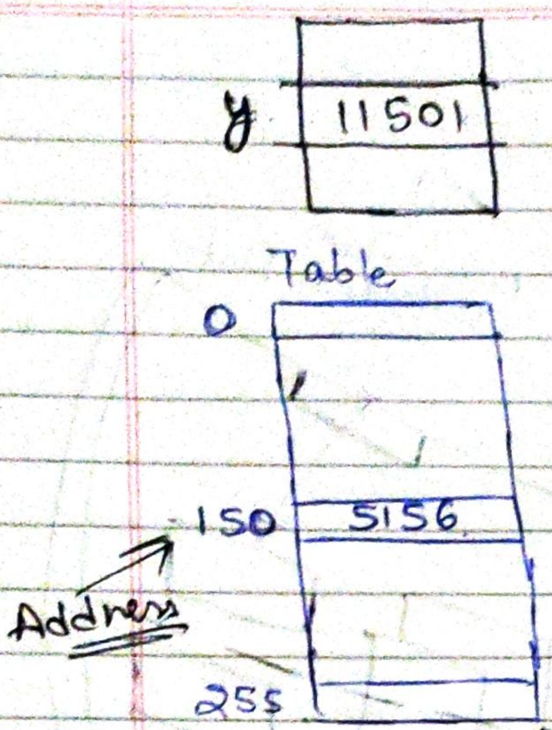
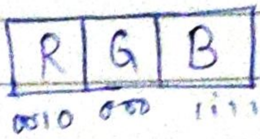
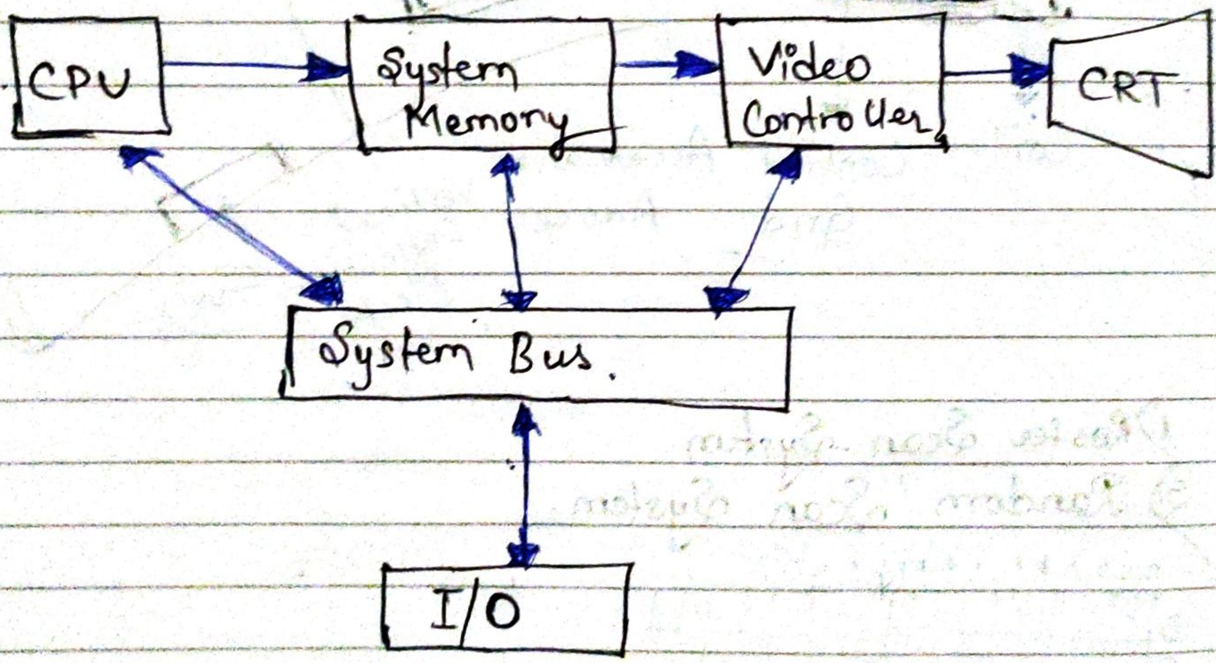


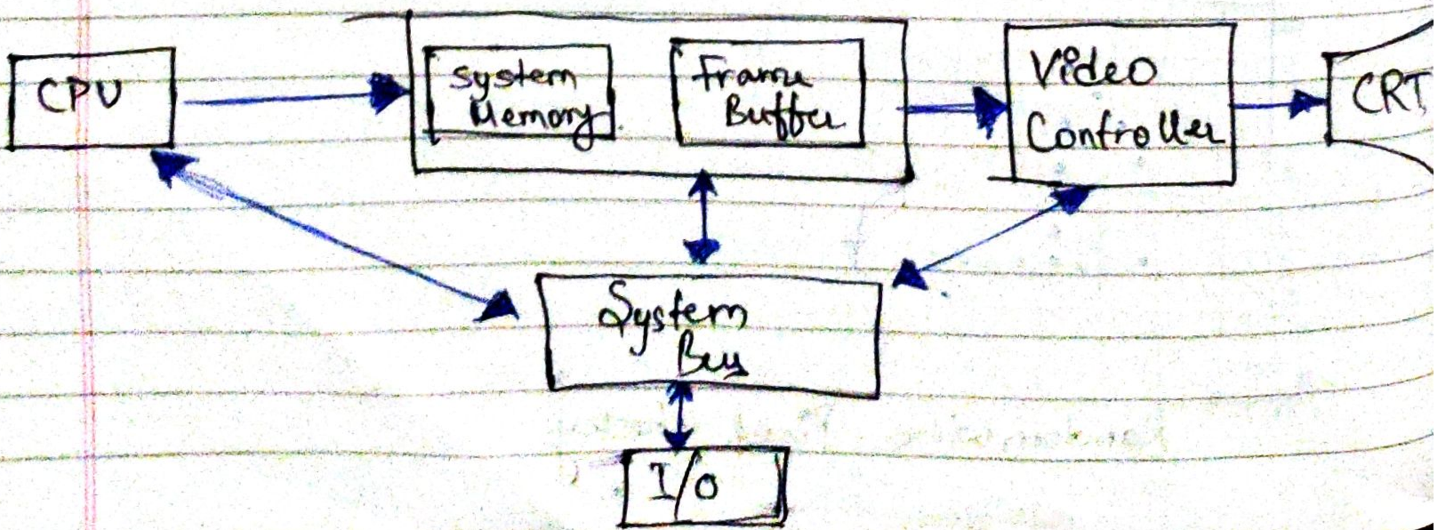
Image will generate in pixel form by summation of (R+G+B) intensity

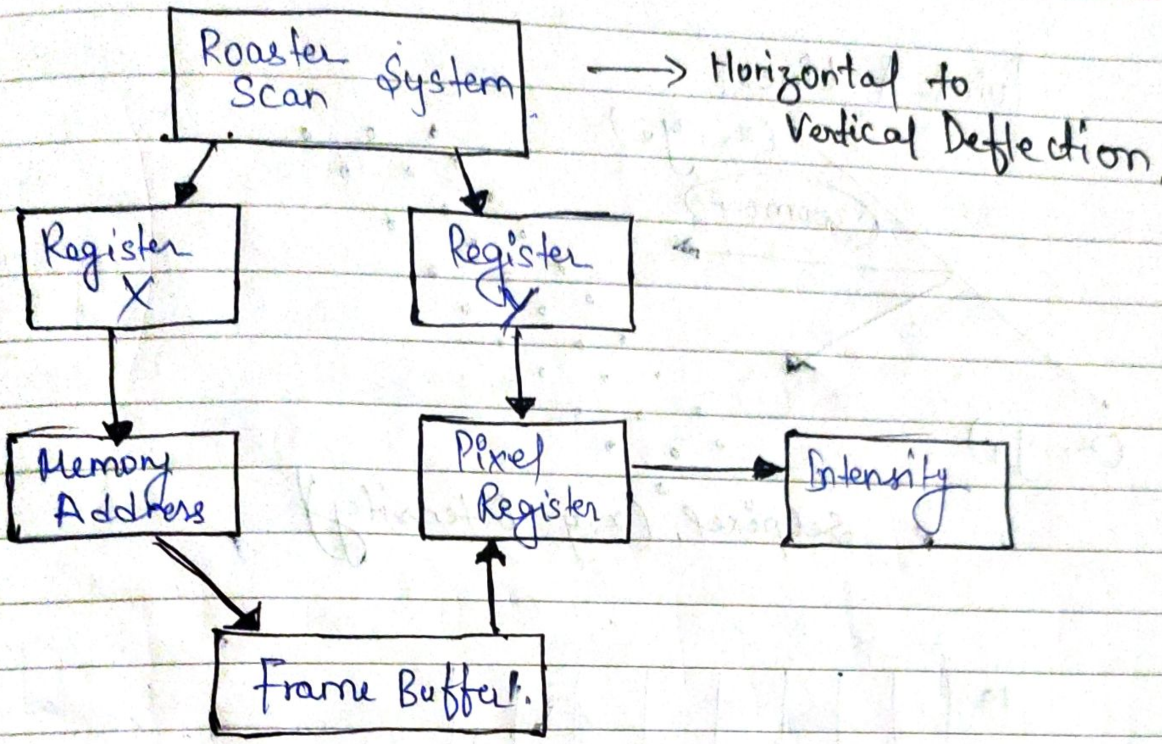


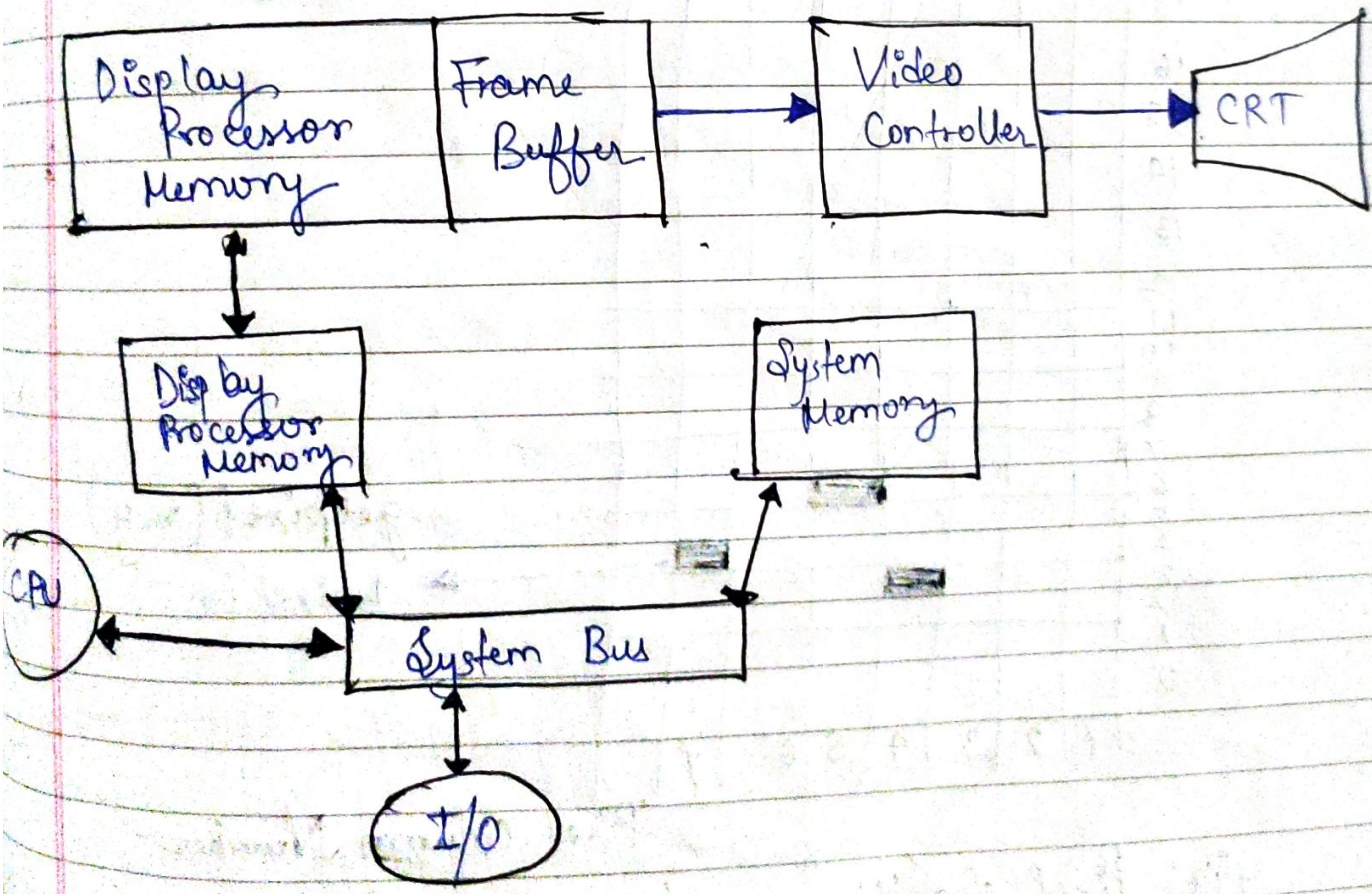
In binary form.

Architecture of Roster Scan System

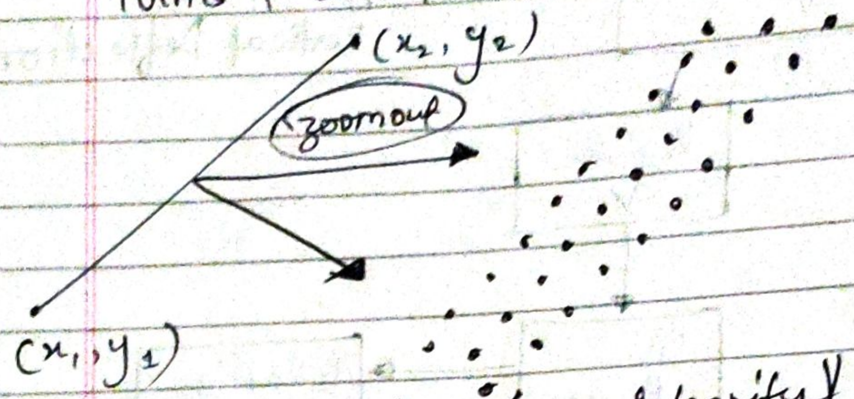





Roaster Scan System
Horizontal to Vertical Deflection.
Register X
Register Y
Memory Address
Pixel Register
Intensity
Frame Buffer.


Display Processor Memory
Frame Buffer
Video Controller
CRT
Display Processor Memory
System Memory
CPU
System Bus
I/O

Points & Lines.



setpixel (x, y, intensity)

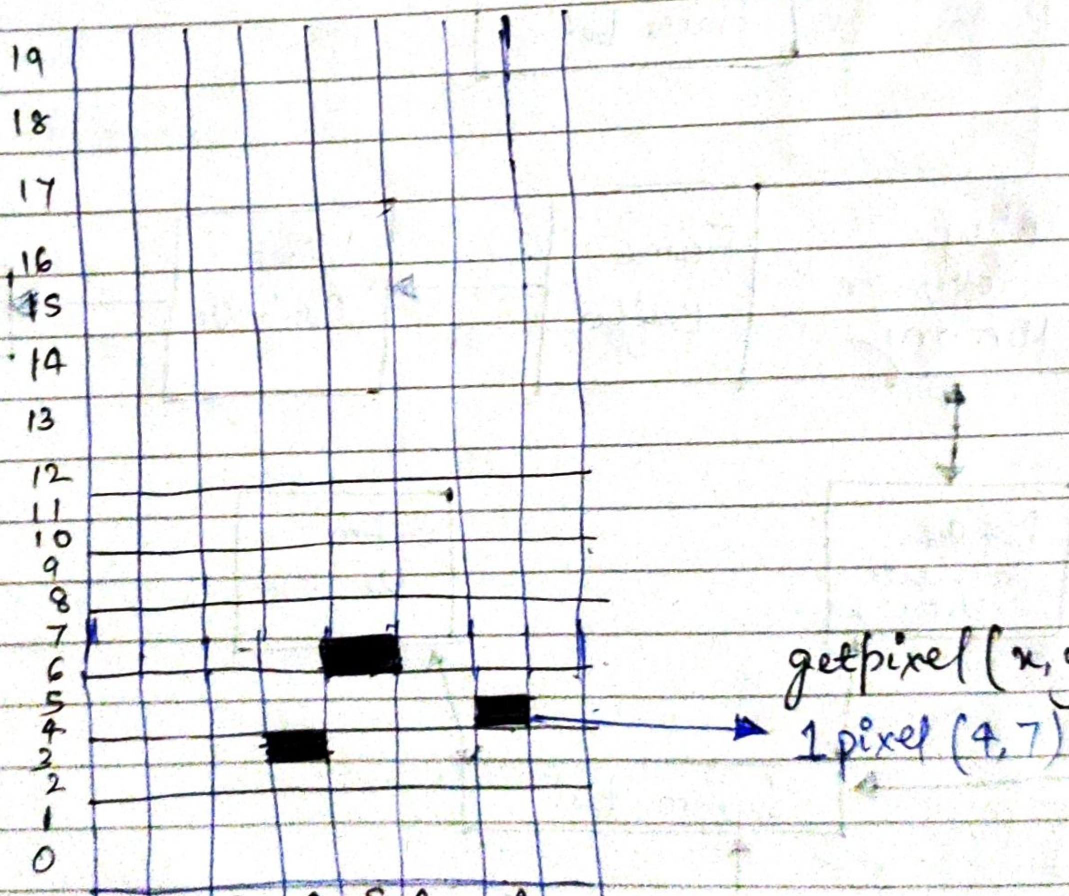


Fig: Pixel position represented by scan line number

* LINE DRAWING ALGORITHM

If line point is (x_1, y_1) & (x_2, y_2). Cartesian slope intercept equation for straight line is

$$y = mx + b \quad \text{— (I)}$$

Since slope of the line $m = \frac{y_2 - y_1}{x_2 - x_1}$ — (II) and b is y-intercept.

So, $b = y_1 - m \cdot x_1$ — (III)

x-interval = Δx along a line
and y-interval = Δy along a line. Then

$$\Delta y = m \cdot \Delta x \quad \text{— (IV)}$$

and obtained the x-interval Δx corresponding to a specified Δy as

$$\Delta x = \frac{\Delta y}{m} \quad \text{— (V)}$$

This equation from the basis for determining deflection voltages in analog devices (A) For line slope magnitudes $|m| < 1$, Δx can be set proportional to a small horizontal deflection voltage and the corresponding vertical deflection is then set proportional to Δy as calculated by eq^n (IV)

(B) When the slope magnitude $|m| > 1$, Δy can be set proportional to a small vertical deflection voltage with the corresponding horizontal deflection voltage is proportional voltage to Δx calculated from eq^n (V)

(C) For slope magnitude $|m| = 1$, it means that $\Delta x = \Delta y$ and the horizontal and vertical deflection always are equal.

In each case a smooth line with slope m is generated betw the specified end points. On ~~negt~~ raster system lines are plotted with pixels and step size is horizontal and vertical directions are constrained by pixel separation.

Digital Differential Analyser Algorithm (DDA)

The DDA is a scan conversion line algorithm method based on calculating either Δy or Δx using eqn (IV) & (V), we sample the line at unit intervals in one coordinate and determine corresponding integer values nearest the line path for the other coordinate.

Consider a first line with positive slope, if the slope is less than or equal to 1, we sample at unit x intervals. ($\Delta x = 1$) and compute each successive y value as

~~y(k+1) = y(k + m)~~

$$\boxed{y_{k+1} = y_k + m} \quad \text{— (VI)}$$

$$\boxed{\because |m| \leq 1}$$

Subscript 'k' takes integer values taking from '1' to the first point and increased by '1' until the final endpoint is reach

Since 'm' can be any real number b/w 0 & 1, the calculat y values must be rounded to nearest integer.

For lines, with a positive slope (> 1), we reverse the roles of x & y i.e. we sample at unit y intervals ($\Delta y = 1$) and calculate each succeeding x value as

$$\boxed{x_{k+1} = x_k + \frac{1}{m}} \quad \text{— (VII)}$$

$$\boxed{\because |m| > 1}$$

Algorithms ⟶ H. Baker
+
Numericals ⟶ Other books

Date 12/12/22

Eq^n (VI) & (VII) are based on assumptions that lines are to be processed from the left endpoint to the right endpoint. If the processing is reversed so that the starting endpoint is at the right, then either we have:

$$\Delta x = -1 \text{ and } y_{k+1} = y_k - m \quad \text{— (VIII)}$$

or

when the slope is (>1), we have:

$$\Delta y = -1 \text{ with } x_{k+1} = x_k - \frac{1}{m} \quad \text{— (IX)}$$

Equation (VI) through (IX) can also be used to calculate pixel positions along a line with positive slope. If the absolute value of the slope is less than 1 and the start endpoint is at the left, we set $\Delta x = 1$ and calculate y values with eq^n no. (6).

When the start endpoint is at the right (for the same slope), we set $\Delta x = -1$ and obtain y-positions from equation (VIII).

Similarly, when the absolute value of a negative slope is (>1), we use $(\Delta y = -1)$ and eq^n (IX) or we use $(\Delta y = 1)$ and eq^n (VII)

Bressenham's Line Drawing Algorithm

Steps

(1) Input the two line endpoints and store the left endpoint in (x_0, y_0)

(2) Load (x_0, y_0) in the frame buffer i.e. plot the first point

(3) Calculate constants $\Delta x, \Delta y, 2\Delta y$ and $2\Delta y - 2\Delta x$ and obtain the starting value for the decision parameter as

$$\boxed{P_0 = 2\Delta y - \Delta x}$$

(4) Let each x_k along the line starting at $k=0$, perform the following test

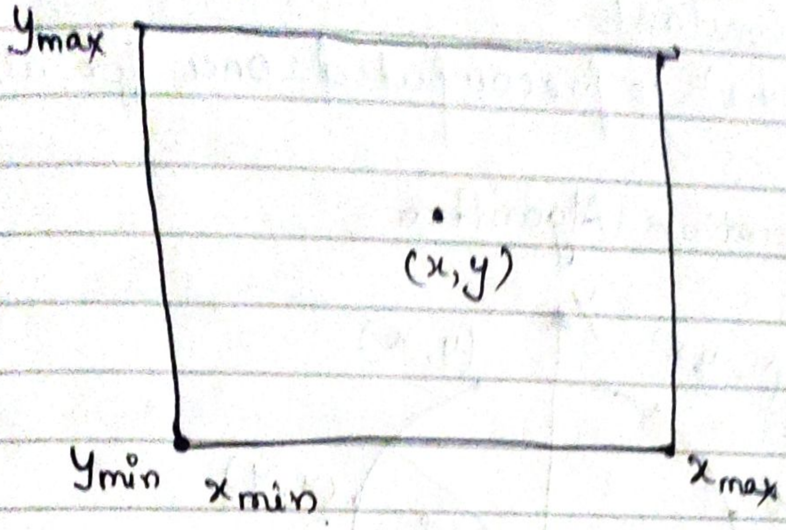
If $p_k < 0$, the next point to plot is (x_{k+1}, y_k) and $p_{k+1} = p_k + 2\Delta y$ otherwise the next point to plot is (x_{k+1}, y_{k+1}) and $(P_{k+1} = P_k + 2\Delta y - 2\Delta x)$

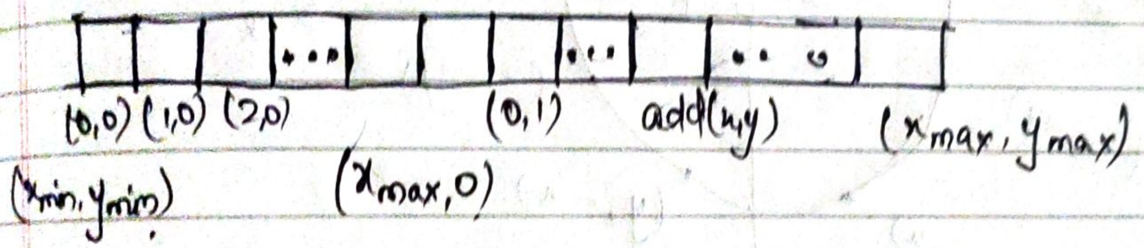
(5) Repeat step (4), Δx times.

Question> Draw the line of end-points $(20, 10)$ and $(30, 18)$, this line has a slope of 0.8.

→

* Frame Buffer Array





Suppose the frame buffer array is addressed in row - measure order and that pixel position varies from (0,0) ~~to~~ of the lower left screen corner to (x_{max}, y_{max}) at the top right corner. For a y-level system (one bit per pixel), the frame buffer bit address for pixel position (x,y) is calculated as:

$$addr(x,y) = addr(0,0) + y(x_{max}+1) + x \quad \text{—— } ①$$

Moving across a scanned line, we can calculate the frame buffer address for the pixel at (x+1, y) as the following offset from the address for the position (x,y)

$$addr(x+1, y) = addr(x,y) + 1 \quad \text{—— } ②$$

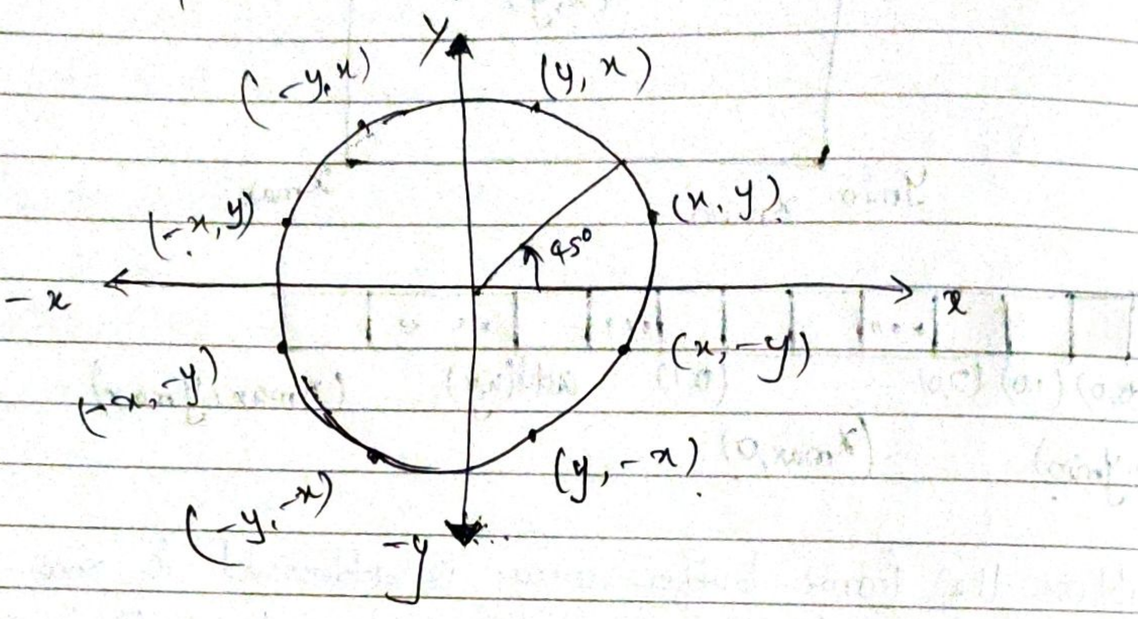
Stepping diagonally upto the next scanned line from (x,y) we get to the frame buffer address of (x+1, y+1) with the calculation :

C.CO
Monday

Date 19/12/22

$addr(x+1, y+1) = addr(x, y) + x_{max} + 2$
where the Constant
$(x_{max}+2)$ is precomputed once for all line Segmen

Circle Generation Algorithm



Mid-Point Circle Algorithm

(1) Input radius ~~are on~~ R and (x_c, y_c) and obtain the first point on the circumference of a circle centred on the origin as $(x_0, y_0) = (0, r)$

(2) Calculate the Initial value of the decision parameter as $P_0 = \frac{5}{4} - r$

or $P_0 = 1 - r$.

(3) At each x_k position, starting at $K=0$, perform the following test.

If $P_k < 0$, the next point along the circle centred on (0,0) is : (x_{k+1}, y_k) and decision parameter ~~$P_{k+1} = P_k$~~
$P_{k+1} = P_k + 2x_{k+1} + 1$

otherwise the next point along the circle is (x_{k+1}, y_{k-1}) and decision parameter is $(P_{k+1} = P_k + 2x_{k+1} + 1 - 2y_{k+1})$

$\because 2x_{k+1} = 2x_k + 2$

&

$2y_{k+1} = 2y_k - 2$

(4) Determine symmetry points in the other 7 octets.

(5) Move each calculated pixel position (x, y) on to the circle path centred on (x_c, y_c) and plot the coordinate values $(\boxed{x = x + x_c}, \boxed{y = y + y_c})$

(6) Repeat step (3) to (5) until $(x \geq y)$.

Q) If circle radius = 10, then determine the position along the ~~of~~ circle obtain in the first quadrant of $(x=0, x=y)$

→ Given: $(x_0, y_0) = (0, 10)$

$P_0 = \frac{5}{4} - 10 = -\frac{35}{4} = -8.75$

or $P_0 = 1 - 10 = -9$

$\because P_0 < 0$, then ~~$P_1 = P_1 + 2$~~ $(x_{k+1}, y_k) = (1, 10)$

$P_1 = -9 + 2 \times$ [illegible] + [illegible] $+ 1$

$= -6$

$P_1 < 0$ then $(x_{k+2}, y_{k+1}) = (2, 10)$

$P_2 = -6 + 2 \times 2 +$ [illegible] $+ 1$

$= -1$

$P_2 < 0$ then $(3, 10)$

$P_3 = -1 + 2 \times 3 +$ [illegible] $+ 1 = 6$

$P_4 = -3, P_5 = 0$

$P_3 \geq 0$ then $(4,9)$

$P_4 = \cancel{6 + 2\times 3 + 2 + 1 - 2\times 10 + 4}$

$= \cancel{6 + 6 + 2 + 1 - 20 + 4}$

$= \cancel{19 - 20}$

$= 6 + 2\times 4 + 1 - 2\times 9$

$= 6 + 8 + 1 - 18$

$= -3$

$P_4 < 0$ then $(5,9)$

$P_5 = -3 + 2\times 5 + 1$

$= 11 - 3 = 8$

$P_5 \geq 0$ then $(6,8)$

$P_6 = 8 + 2\times 6 + 1 - 2\times 8$

$= 8 + 12 + 1 - 16$

$= 5$

$P_6 \geq 0$ then $(7,7)$