Computer Graphics and Multimedia Systems SCS1302

Unit 2

Syllabus

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY FACULTY OF COMPUTING

SCS1302	COMPUTER GRAPHICS AND MULTIMEDIA SYSTEMS	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To gain knowledge to develop, design and implement two and three dimensional graphical structures.
- To enable students to acquire knowledge of Multimedia compression and animations.
- To learn creation, Management and Transmission of Multimedia objects.

UNIT 1 BASICS OF COMPUTER GRAPHICS **9 Hrs.**

Output Primitives: Survey of computer graphics - Overview of graphics systems - Line drawing algorithm - Circle drawing algorithm - Curve drawing algorithm - Attributes of output primitives - Anti-aliasing.

UNIT 2 2D TRANSFORMATIONS AND VIEWING **8 Hrs.**

Basic two dimensional transformations - Other transformations - 2D and 3D viewing - Line clipping - Polygon clipping - Logical classification - Input functions - Interactive picture construction techniques.

UNIT 3 3D CONCEPTS AND CURVES **10 Hrs.**

3D object representation methods - B-REP , sweep representations, Three dimensional transformations. Curve generation - cubic splines, Beziers, blending of curves- other interpolation techniques, Displaying Curves and Surfaces, Shape description requirement, parametric function. Three dimensional concepts – Introduction - Fractals and self similarity- Successive refinement of curves, Koch curve and peano curves.

Syllabus

UNIT 4 METHODS AND MODELS **8 Hrs.**

Visible surface detection methods - Illumination models - Halftone patterns - Dithering techniques - Polygon rendering methods - Ray tracing methods - Color models and color applications.

UNIT 5 MULTIMEDIA BASICS AND TOOLS **10 Hrs.**

Introduction to multimedia - Compression & Decompression - Data & File Format standards - Digital voice and audio - Video image and animation. Introduction to Photoshop - Workplace - Tools - Navigating window - Importing and exporting images - Operations on Images - resize, crop, and rotate - Introduction to Flash - Elements of flash document - Drawing tools - Flash animations - Importing and exporting - Adding sounds - Publishing flash movies - Basic action scripts - GoTo, Play, Stop, Tell Target

Max. 45 Hours

TEXT / REFERENCE BOOKS

1. Donald Hearn, Pauline Baker M., "Computer Graphics", 2nd Edition, Prentice Hall, 1994.
2. Tay Vaughan ,"Multimedia", 5th Edition, Tata McGraw Hill, 2001.
3. Ze-Nian Li, Mark S. Drew ,"Fundamentals of Multimedia", Prentice Hall of India, 2004.
4. D. McClelland, L.U.Fuller ,"Photoshop CS2 Bible", Wiley Publishing, 2005.
5. James D. Foley, Andries van Dam, Steven K Feiner, John F. Hughes, "Computer Graphics Principles and Practice, 2nd
6. Edition in C, Audison Wesley, ISBN - 981 -235-974-5
7. William M. Newman, Roberet F. Sproull, " Principles of Interactive Computer Graphics", Second Edition, Tata McGraw-Hill Edition.

Course Objective(CO)

CO1: Construct lines and circles for the given input.

CO2: Apply 2D transformation techniques to transform the shapes to fit them as per the picture definition.

CO3: Construct splines, curves and perform 3D transformations

CO4: Apply colour and transformation techniques for various applications.

CO5: Analyse the fundamentals of animation, virtual reality, and underlying technologies.

CO6: Develop photo shop applications

2D VIEWING

- The mapping of a 2D world coordinate system to device coordinates is called a two-dimensional viewing transformation.
- The clipping window is the section of the 2D scene that is selected for viewing.
- The display window is where the scene will be viewed.
- The viewport controls the placement of the scene within the display window

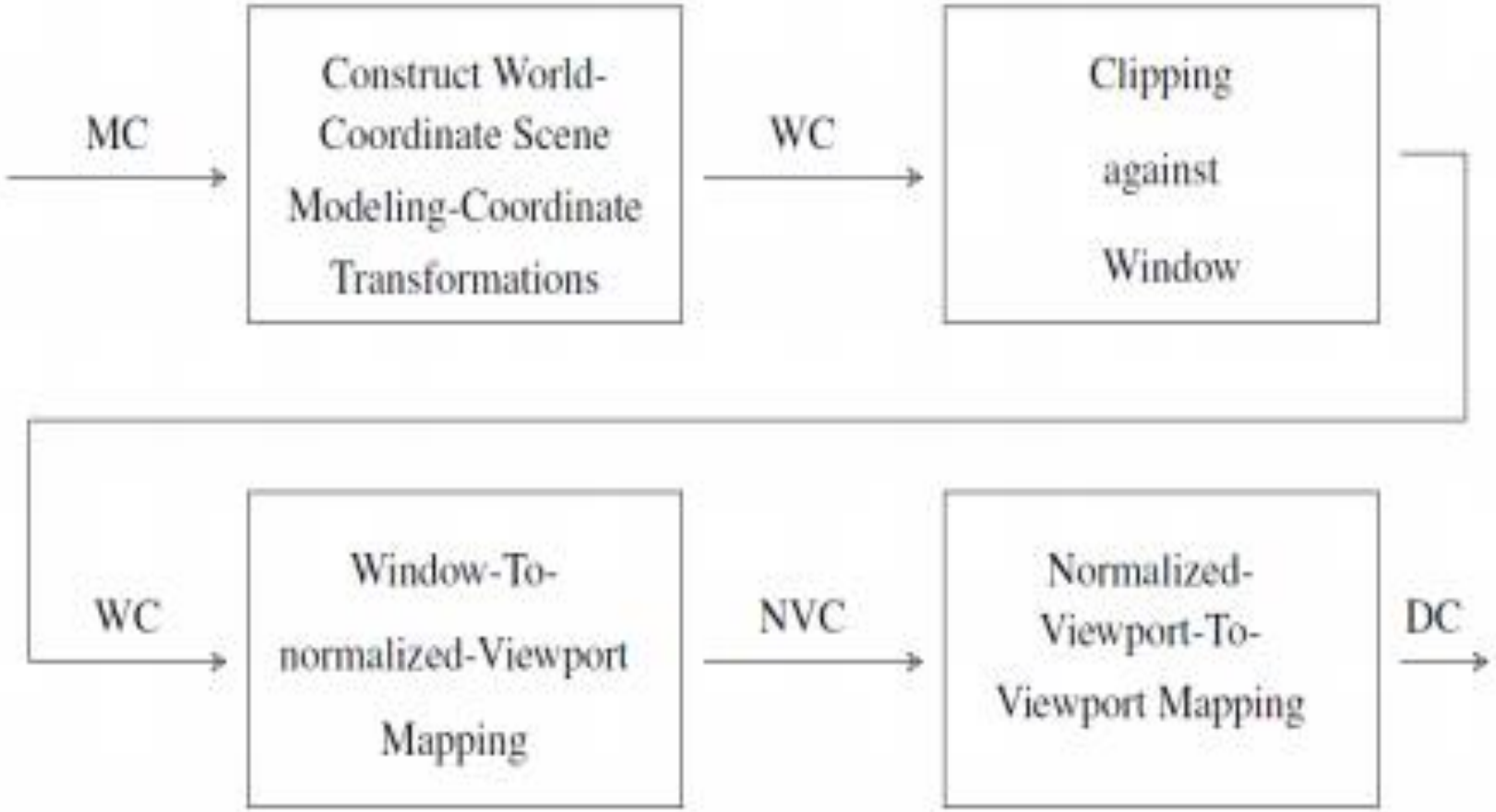
2D VIEWING

- The process of mapping a part of the world co-ordinate scene to device co-ordinate system is known as viewing transformation.
- A world co-ordinate area selected for display is called as a window and an area on display device to which a window is mapped is called as a view point.
- Window defines ‘What’ to be viewed and the viewport defines ‘Where’ it is to be displayed

2D VIEWING

- A window-viewport transformation describes the mapping of a (rectangular) window in one coordinate system into another (rectangular) window in another coordinate system.
- This transformation is defined by the section of the original image that is transformed (clipping window), the location of the resulting window (viewport), and how the window is translated, scaled or rotated

2D VIEWING


MC
Construct World-Coordinate Scene Modeling-Coordinate Transformations
WC
Clipping against Window
WC
Window-To-normalized-Viewport Mapping
NVC
Normalized-Viewport-To-Viewport Mapping
DC

A point at position (xw, yw) in window mapped into position (xv, yv) in the associated viewport.

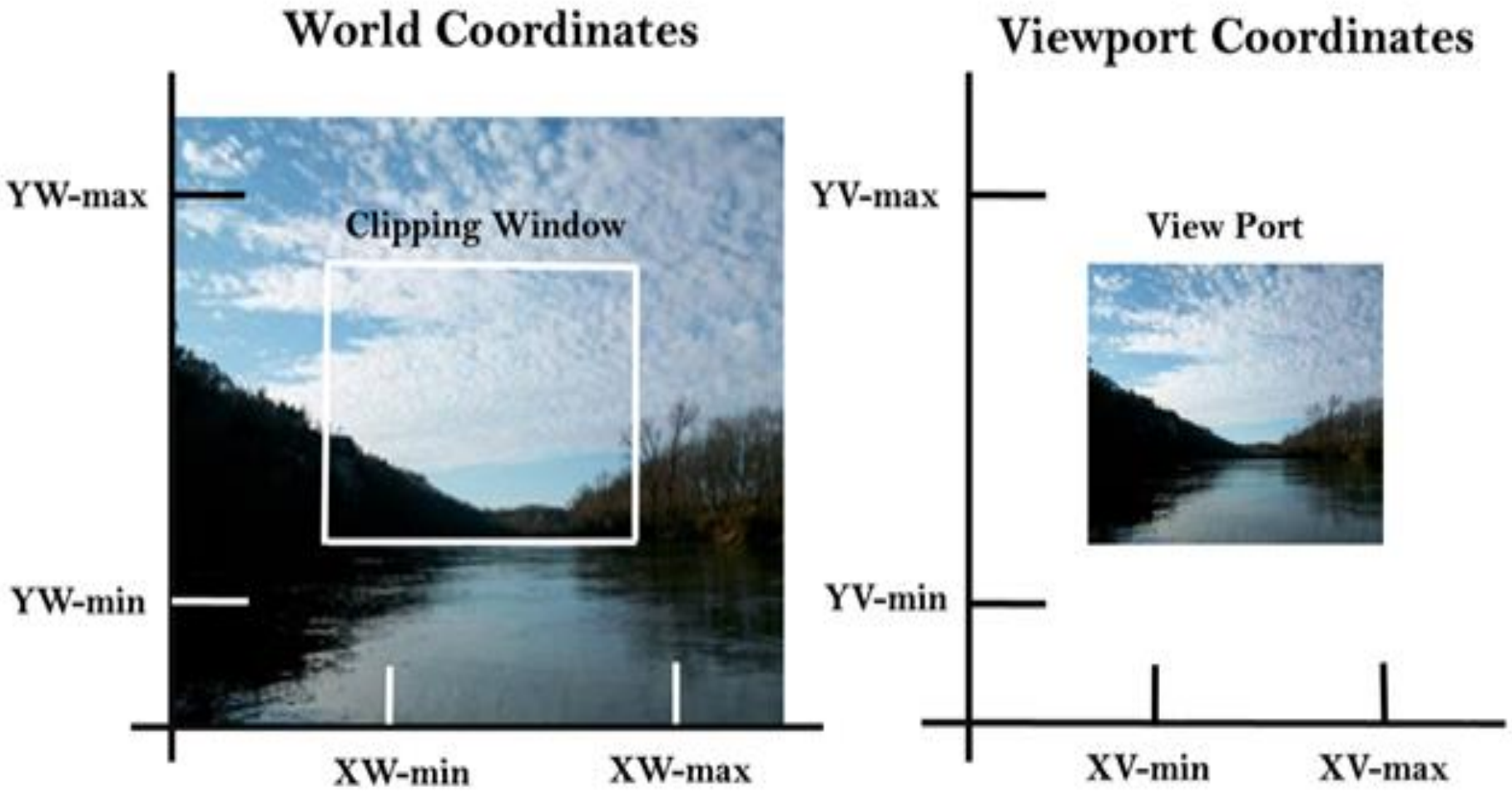
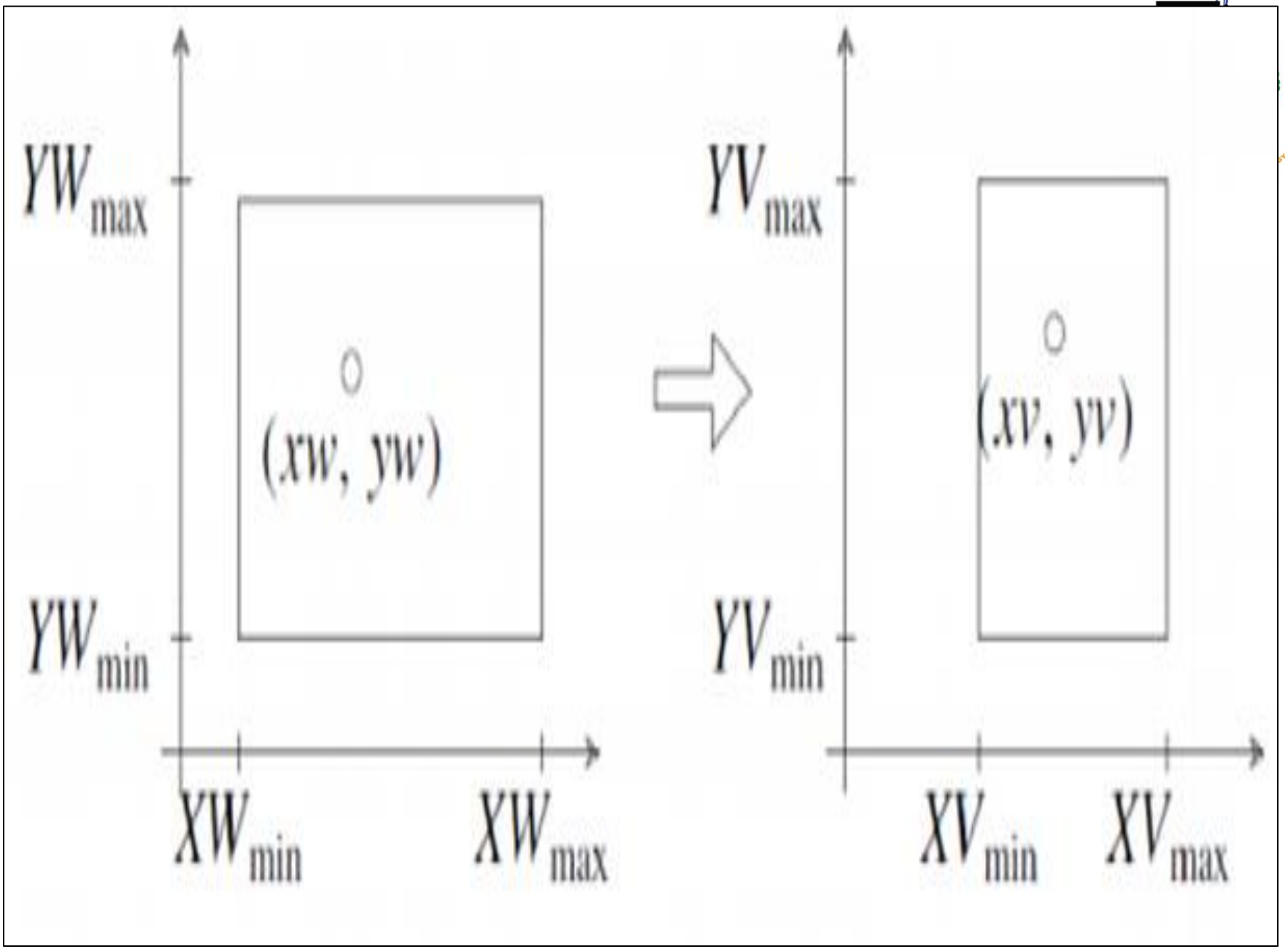


Fig: Window to viewport mapping


YW_{max}
(xw, yw)
YW_{min}
XW_{min}
XW_{max}
YV_{max}
(xv, yv)
YV_{min}
XV_{min}
XV_{max}

In order to maintain the same relative placement of the point in the viewport as in the window, we require:

$$\frac{xv - xv_{min}}{xv_{max} - xv_{min}} = \frac{xw - xw_{min}}{xw_{max} - xw_{min}}$$

$$\frac{yv - yv_{min}}{yv_{max} - yv_{min}} = \frac{yw - yw_{min}}{yw_{max} - yw_{min}}$$

Solving these impressions for the viewport position (xv, yv), we have

$$xv = xv_{min} + \frac{xw - xw_{min}}{xw_{max} - xw_{min}} (xv_{max} - xv_{min})$$

$$yv = yv_{min} + \frac{yw - yw_{min}}{yw_{max} - yw_{min}} (yv_{max} - yv_{min})$$

$$xv = xv_{min} + (xw - xw_{min})sx$$

$$yv = yv_{min} + (yw - yw_{min})sy$$

Where scaling factors are

$$sx = \frac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}}$$

$$sy = \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}}$$

Clipping

- When we have to display a large portion of the picture, then not only scaling & translation is necessary, the visible part of picture is also identified. This process is not easy. Certain parts of the image are inside, while others are partially inside. The lines or elements which are partially visible will be omitted.
- For deciding the visible and invisible portion, a particular process called clipping is used. Clipping determines each element into the visible and invisible portion. Visible portion is selected. An invisible portion is discarded.

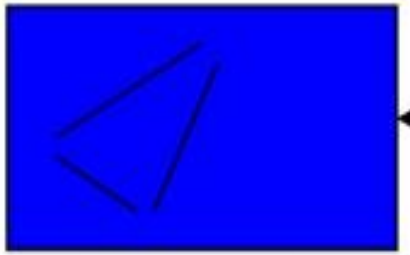
Types of Lines

Lines are of three types:

- **Visible:** A line or lines entirely inside the window is considered visible
- **Invisible:** A line entirely outside the window is considered invisible
- **Clipped:** A line partially inside the window and partially outside is clipped. For clipping point of intersection of a line with the window is determined.

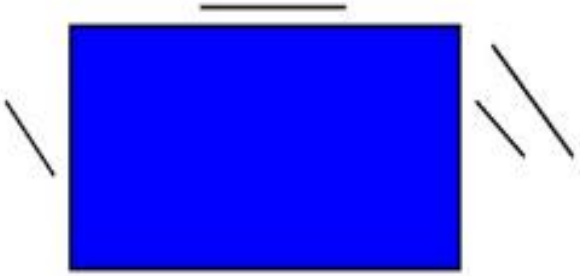
Example: Types of Lines

Case1:



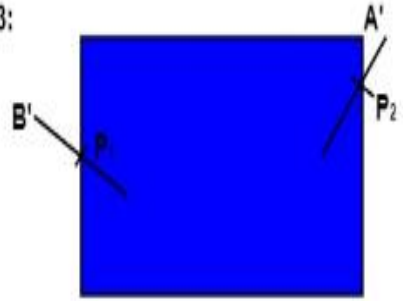
Visible Lines

Case2:



Invisible Lines

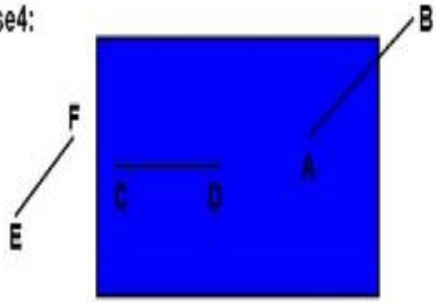
Case3:



Clipped Lines

In this figure P_1 and P_2 are point of intersection. The line P_2 to A' and P_1 to B' is discarded or clipped.

Case4:

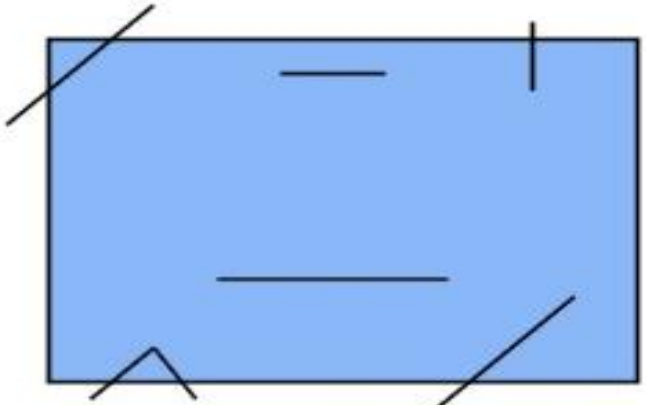


Mixed Lines

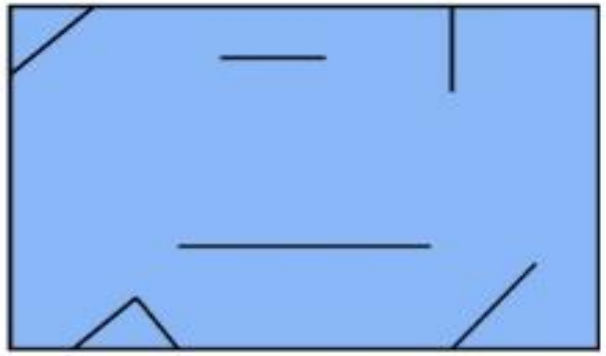
In this figure AB is clipped case.
CD is visible line.
EF is invisible line.

Before and after clipping

- The window against which object is clipped called a clip window. It can be curved or rectangle in shape.



Original Picture
or
Before Clipping



After Clipping

Applications of clipping

Clipping can be applied to world co-ordinates. The contents inside the window will be mapped to device co-ordinates. Another alternative is a complete world co-ordinates picture is assigned to device co-ordinates, and then clipping of viewport boundaries is done.

- It will extract part we desire.
- For identifying the visible and invisible area in the 3D object.
- For creating objects using solid modeling.
- For drawing operations.
- Operations related to the pointing of an object.
- For deleting, copying, moving part of an object.

Types of Clipping

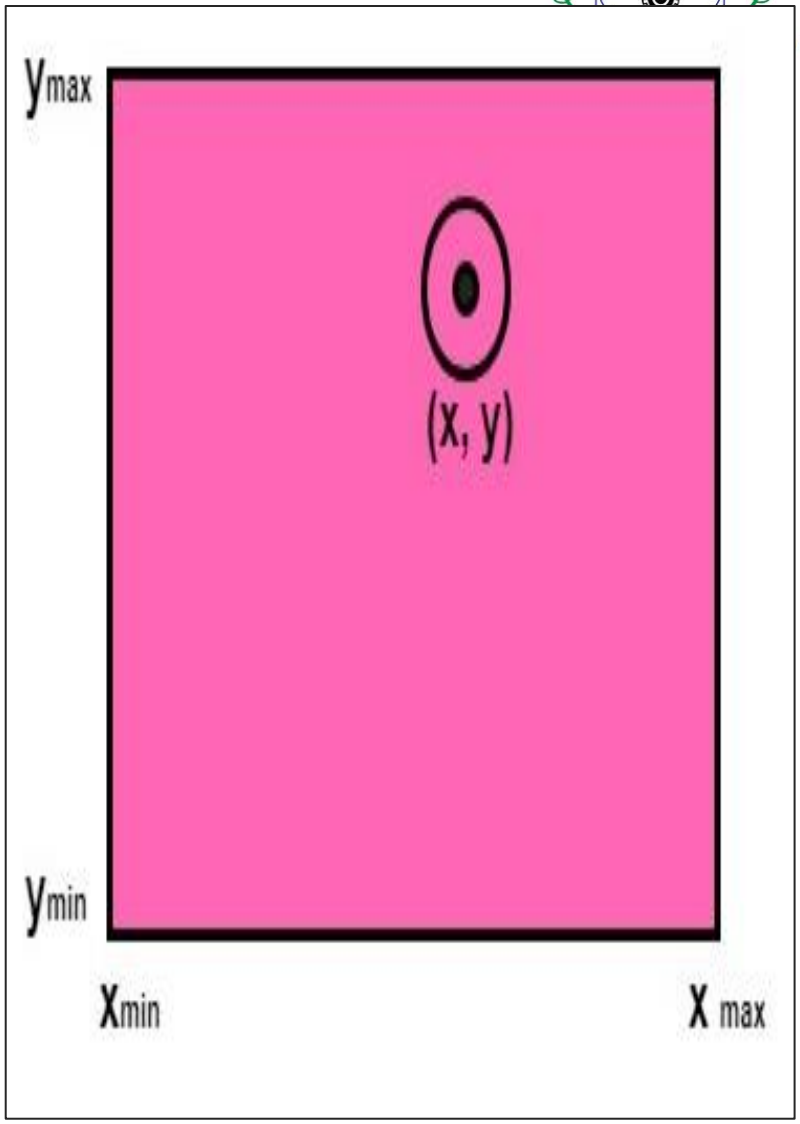
- Point Clipping
- Line Clipping
- Area Clipping (Polygon)
- Curve Clipping
- Text Clipping
- Exterior Clipping

Point Clipping

- Point Clipping is used to determining, whether the point is inside the window or not. For this following conditions are checked.

$$x_{wmin} \leq x \leq x_{wmax}$$
$$Y_{wmin} \leq Y \leq Y_{wmax}$$

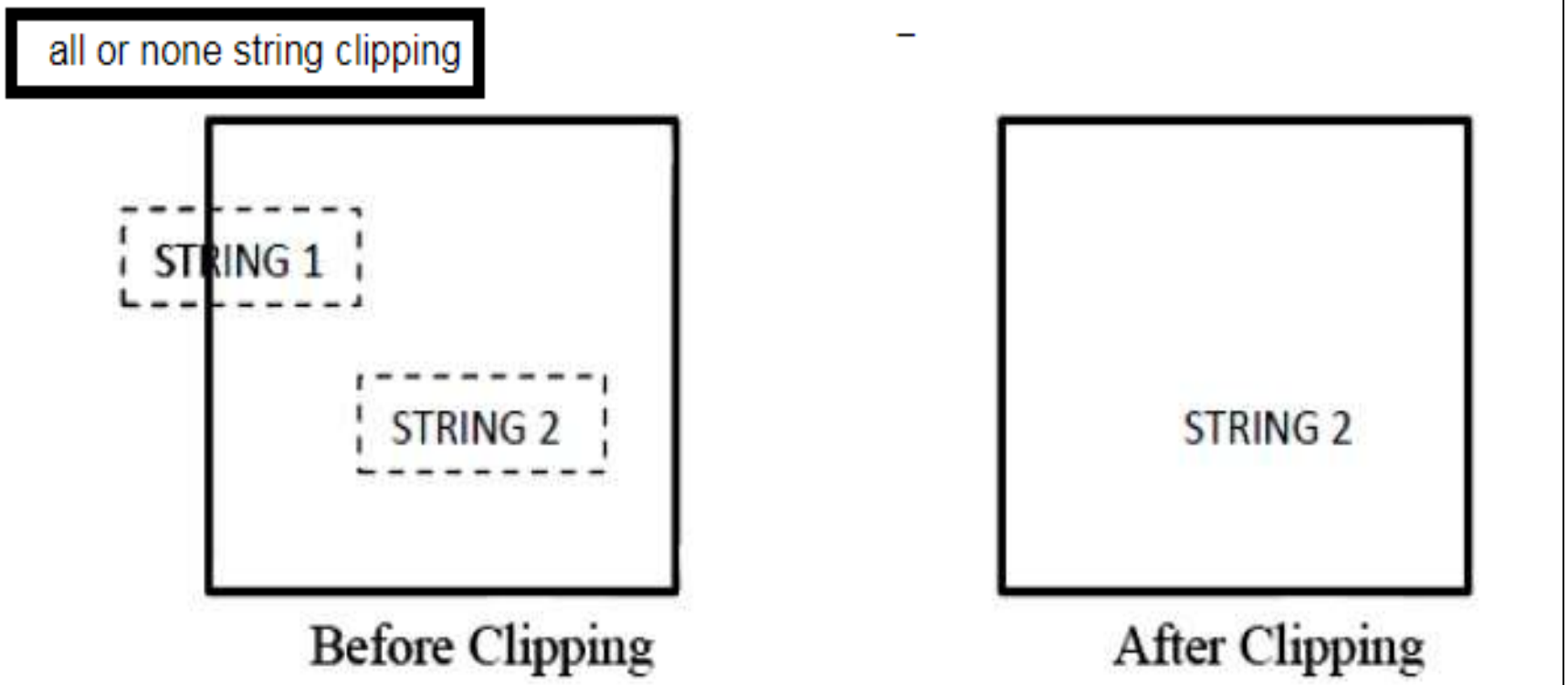
- The (x, y) is coordinate of the point. If anyone from the above inequalities is false, then the point will fall outside the window and will not be considered to be visible.



Text Clipping

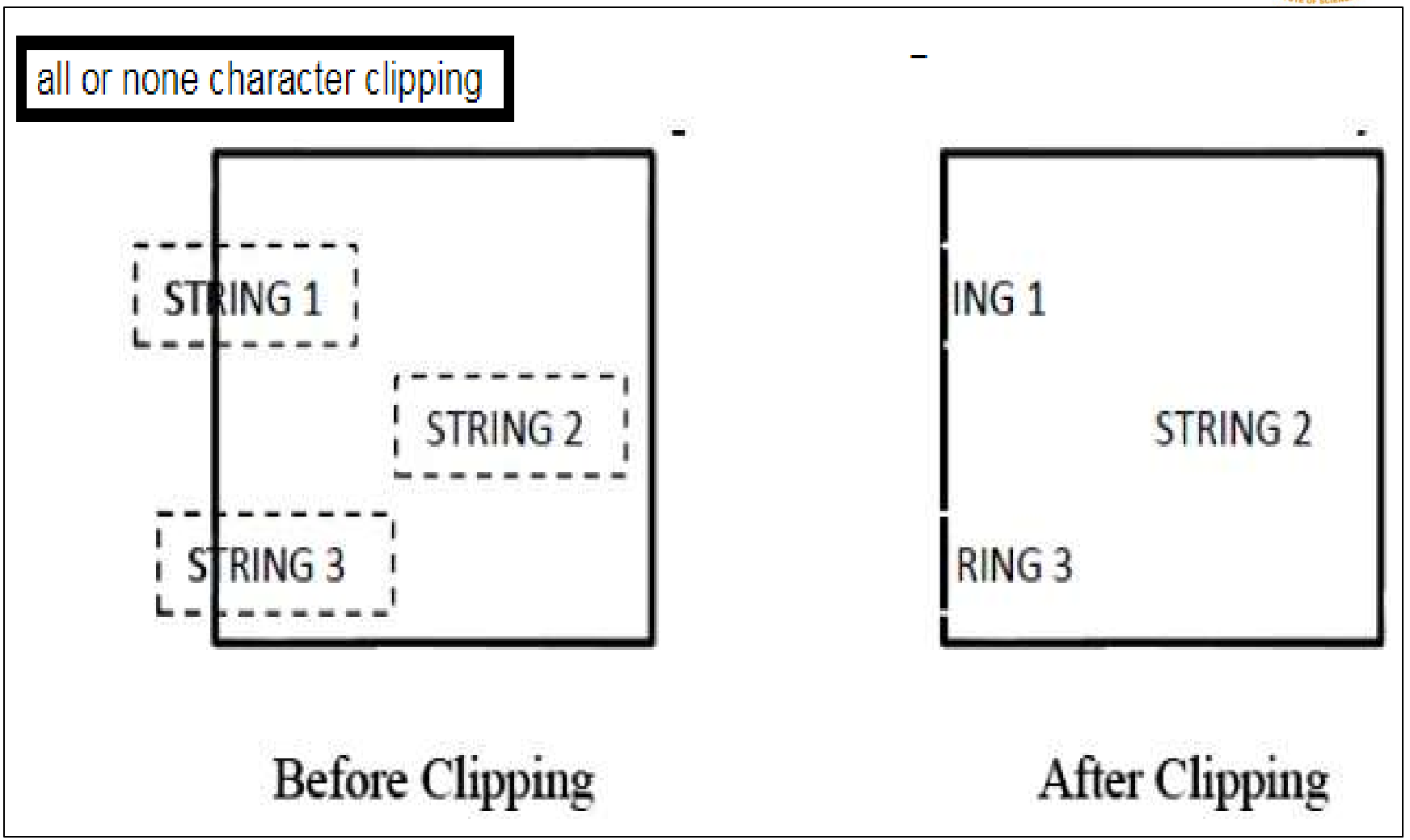
- Various techniques are used to provide text clipping in a computer graphics. It depends on the methods used to generate characters and the requirements of a particular application. There are three methods for text clipping which are listed below −
  - All or none string clipping
  - All or none character clipping
  - Text clipping

All or None String Clipping

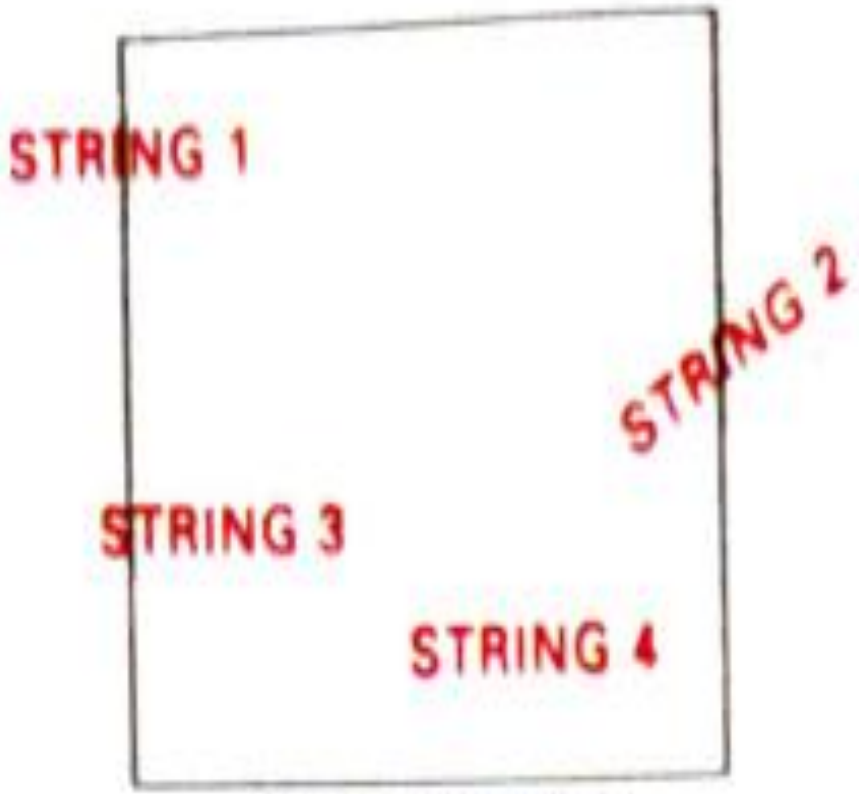


In all or none string clipping method, either we keep the entire string or we reject entire string based on the clipping window. As shown in the above figure, STRING2 is entirely inside the clipping window so we keep it and STRING1 being only partially inside the window, we reject.

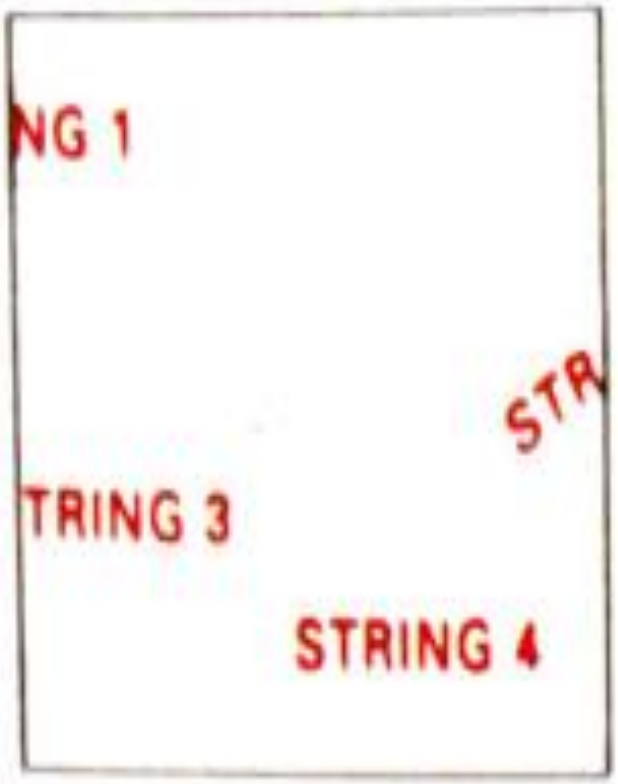
All or None Character Clipping



Text clipping (using a bounding rectangle about individual characters)

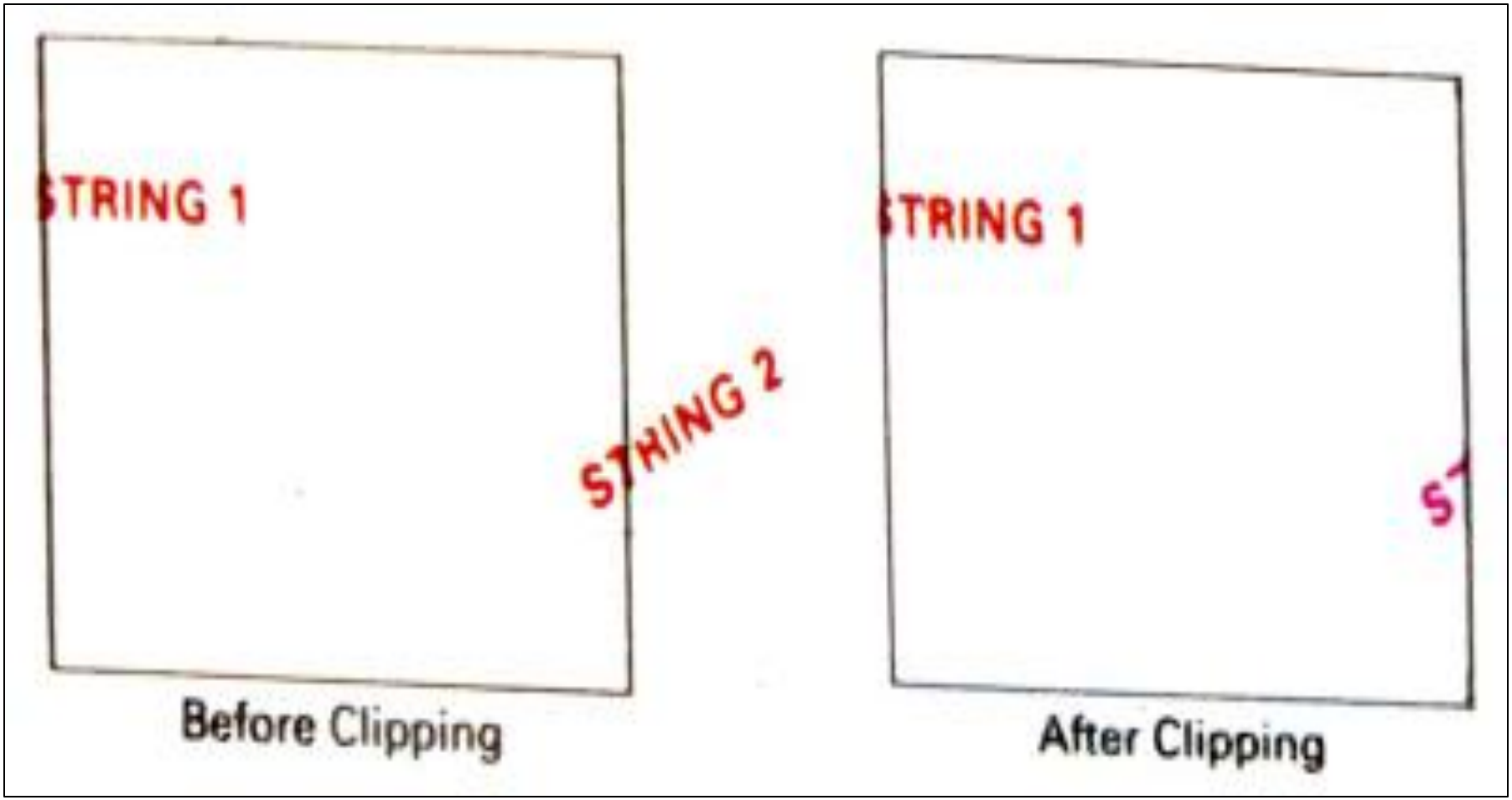


Before Clipping



After Clipping

Text clipping (performed on the components of individual characters)



Curve Clipping, Polygon and Exterior Clipping

Curve Clipping:

- Curve Clipping involves complex procedures as compared to line clipping. Curve clipping requires more processing than for object with linear boundaries. Consider window which is rectangular in shape. The circle is to consider against rectangle window. If circle is completely inside boundary of the window, it is considered visible. So save the circle. If a circle is in outside window, discard it. If circle cut the boundary then consider it to be clipping case

Polygon Clipping:

- Polygon clipping is applied to the polygons. The term polygon is used to define objects having outline of solid. These objects should maintain property and shape of polygon after clipping..

Exterior Clipping:

- It is opposite to previous clipping. Here picture which is outside the window is considered. The picture inside the rectangle window is discarded. So part of the picture outside the window is saved.

Uses of Exterior Clipping:

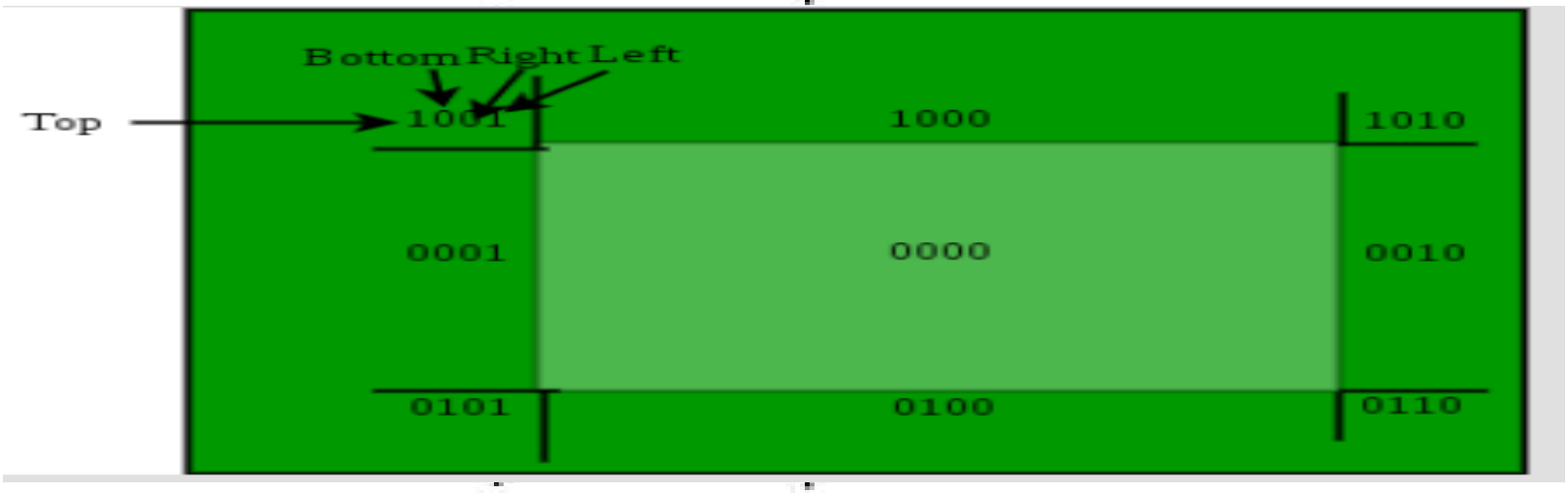
- It is used for displaying properly the pictures which overlap each other.
- It is used in the concept of overlapping windows.
- It is used for designing various patterns of pictures.
- It is used for advertising purposes.
- It is suitable for publishing.
- For designing and displaying of the number of maps and charts, it is also used.

LINE CLIPPING & POLYGON CLIPPING ALGORITHMS

- **Line clipping algorithms**:
  - Cohen–Sutherland
  - Liang–Barsky
  - Nicholl–Lee–Nicholl

- **Polygon clipping algorithms**:
  - Sutherland–Hodgman
  - Weiler–Atherton

COHEN-SUTHERLAND LINE CLIPPING ALGORITHM



- First bit set 1 : Point lies to **left** of window $x < x_{min}$
- Second bit set 1 : Point lies to **right** of window $x > x_{max}$
- Third bit set 1 : Point lies below(**bottom**) window $y < y_{min}$
- fourth bit set 1 : Point lies above(**top**) window $y > y_{max}$

The sequence for reading the codes' bits is Top, Bottom, Right, Left

Bit 1 is the sign bit of $x - xw_{min}$;
bit 2 is the sign bit of $xw_{max} - x$;
bit 3 is the sign bit of $y - yw_{min}$;
bit 4 is the sign bit of $yw_{max} - y$.

Bit 4, Bit 3, Bit 2, Bit 1 =

Top, Bottom, Right, Left.

1 0 0 1

Pseudocode


```
Step 1 : Assign a region code for two endpoints of given line.
Step 2 : If both endpoints have a region code 0000
         then given line is completely inside.
Step 3 : Else, perform the logical AND operation for both region codes.
    Step 3.1 : If the result is not 0000, then given line is completely
               outside.
    Step 3.2 : Else line is partially inside.
        Step 3.2.1 : Choose an endpoint of the line
                     that is outside the given rectangle.
        Step 3.2.2 : Find the intersection point of the
                     rectangular boundary (based on region code).
        Step 3.2.3 : Replace endpoint with the intersection point
                     and update the region code.
        Step 3.2.4 : Repeat step 2 until we find a clipped line either
                     trivially accepted or trivially rejected.
Step 4 : Repeat step 1 for other lines
```

Pseudocode

If a line is clipped case, find an intersection with boundaries of the window

$m=(y_2-y_1)(x_2-x_1)$

(a) If bit 1 is "1" line intersects with left boundary of rectangle window

$y_3=y_1+m(x-x_1)$

where $X = X_{wmin}$

where X_{wmin} is the minimum value of X co-ordinate of window

(b) If bit 2 is "1" line intersect with right boundary

$y_3=y_1+m(X-x_1)$

where $X = X_{wmax}$

where X more is maximum value of X co-ordinate of the window

(c) If bit 3 is "1" line intersects with bottom boundary

$x_3=x_1+(y-y_1)/m$

where $y = y_{wmin}$

y_{wmin} is the minimum value of Y co-ordinate of the window

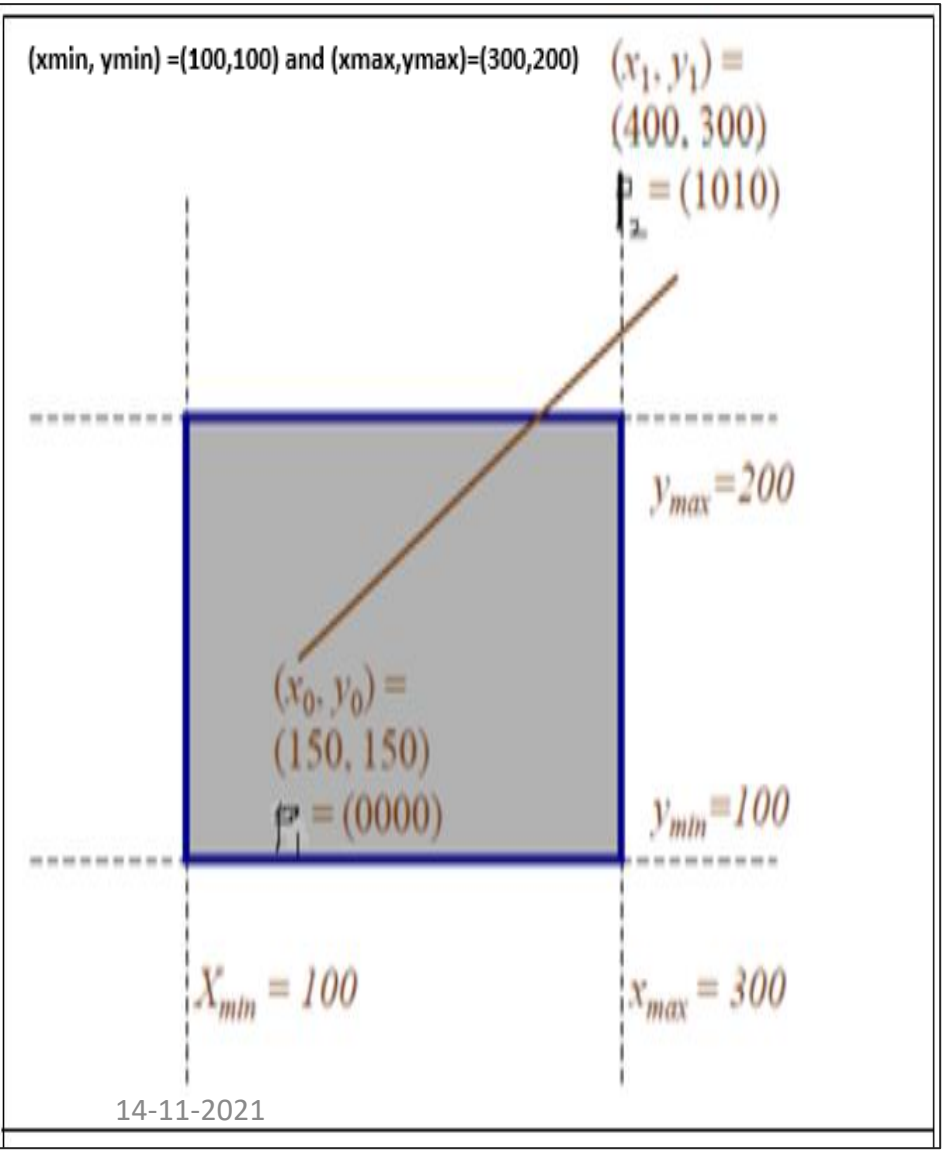
(d) If bit 4 is "1" line intersects with the top boundary

$x_{3=x}1+(y-y_1)/m$

where $y = y_{wmax}$

y_{wmax} is the maximum value of Y co-ordinate of the window

COHEN-SUTHERLAND LINE CLIPPING ALGORITHM



For P1(150,150):

(From Right to Left)

Bit 1 : x - xmin = 150 - 100 = 50
=>set the bit to 0 **(zero - as sign of 50 is +)**

Bit 2: xmax – x = 300 – 150 = 150
=> set the bit to 0

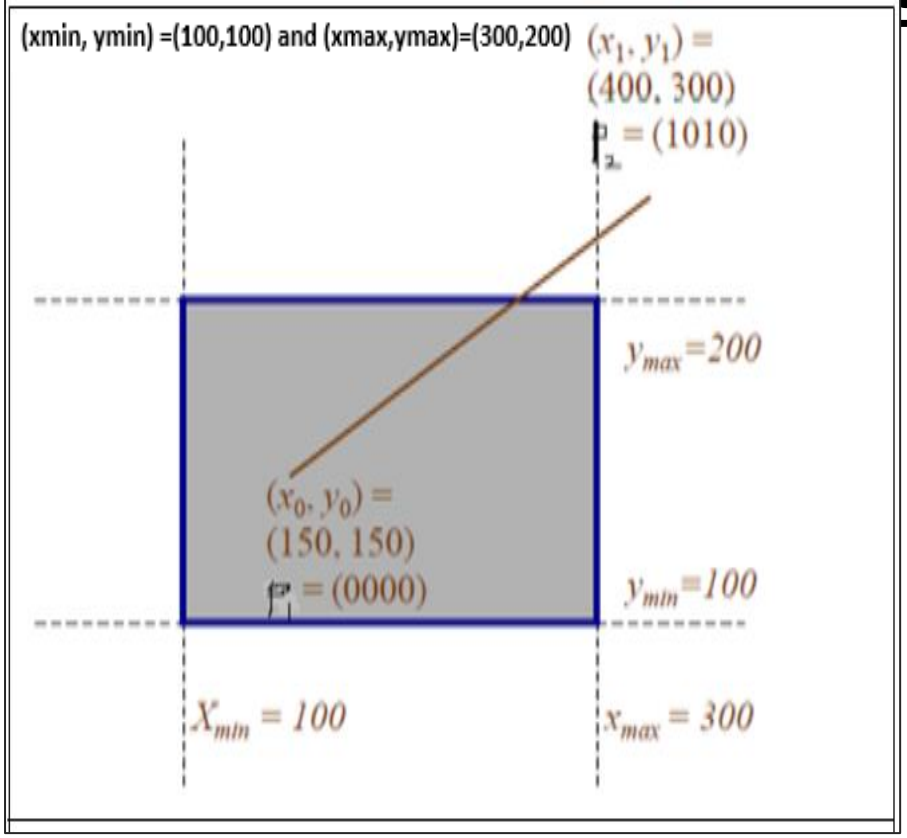
Bit 3: y - ymin = 150 - 100 = 50
=> set the bit to 0

Bit 4: ymax – y = 200 – 150 = 50
=> set the bit to 0

So, the region code for P1 is 0000.

That implies P1 is inside the Clip Window.

COHEN-SUTHERLAND LINE CLIPPING ALGORITHM



For P2(400,300):

(From Right to Left)

Bit 1 : x - xmin = 400 - 100 = 300

=>set the bit to 0 (zero, as sign of 300 is +)

Bit 2: xmax – x = 300 – 400 = - 100

=> set the bit to 1 (one, as sign of 100 is -)

Bit 3: y - ymin = 300 - 100 = 0

=> set the bit to 0

Bit 4: ymax – y = 200 – 300 = - 100

=> set the bit to 1

So, the region code for P2 is 1010.

That implies P2 is in the top and above the Clip Window.

Intersection Point

- **Region code of P1 is 0000.**
- **Region code of P2 is 1010.**

If a line is clipped case, find an intersection with boundaries of the window

$$m=(y_2-y_1)(x_2-x_1)$$

(b) If bit 2 is "1" line intersect with right boundary

$$y_3=y_1+m(x-x_1)$$

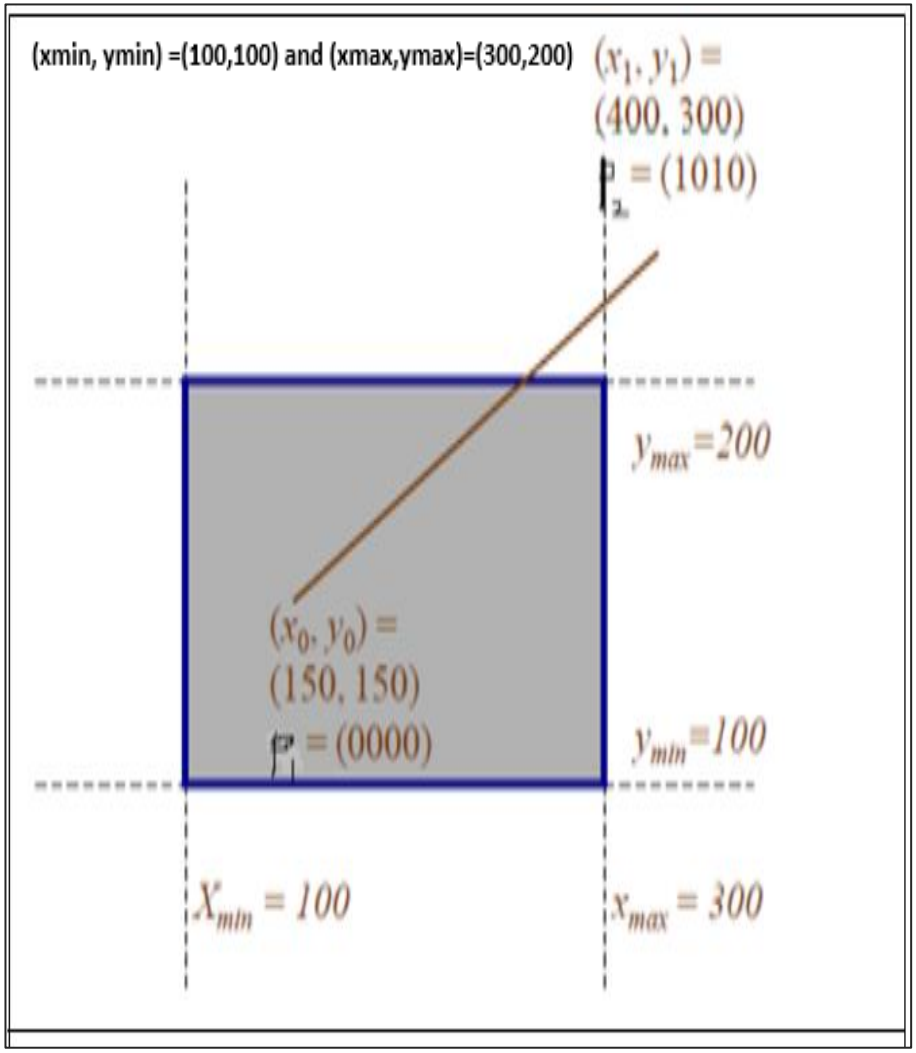
where $x = x_{wmax}$

(d) If bit 4 is "1" line intersects with the top boundary

$$x_{3=x}1+(y-y_1)/m$$

where $y = y_{wmax}$

COHEN-SUTHERLAND LINE CLIPPING ALGORITHM



- **To find the intersection point:**

$y = y1 + m (x_{boundary} - x1)$

where $x_{boundary}$ is set either to xmin or xmax

$x = x1 + (y_{boundary} - y1) / m$

Where $y_{boundary}$ is set either to ymin or ymax

$m= (y1-y0) / (x1-x0)$

Therefore,

$m = (300-150) / (400-150) = 0.6$

$x = x1 + (y_{boundary} - y1) / m$

Therefore,

$x = 400 + (200 – 300) / 0.6 = 233$

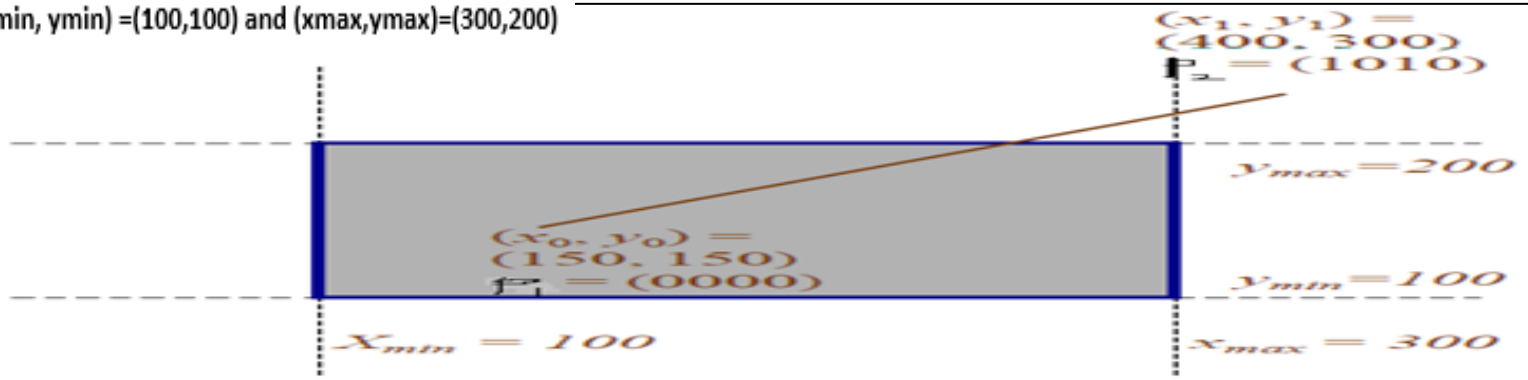
Here, $y_{boundary}$ is the Ymax,

Therefore, the intersection point is,

(x, ymax) =(233, 200)

COHEN-SUTHERLAND LINE CLIPPING ALGORITHM

(xmin, ymin) =(100,100) and (xmax,ymax)=(300,200)



For p1(150,150),

(From Right to left)
First bit: x – xmin = 150 – 100 = 50 = set the bit to 0
Second bit: xmax – x = 300 – 150 = 150 =set the bit to 0
Third bit: y – ymin = 150 – 100 =50 = set the bit to 0
Fourth bit: ymax – y = 200 – 150 = 50 = set the bit to 0

So the region code for P1 is 0000

P1 is inside and p2 is in the top and above region code

To find the intersection point:

$$y = y_1 + m\,(x_{boundary} - x_1)$$

Where $x_{boundary}$ is set either to xmin or xmax

For p2(400,300),

(From Right to left)
First bit: x – xmin = 400 -100 = 300 = set the bit to 0
Second bit: xmax – x = 300 – 400 = - 100 = set the it to 1
Third bit: y – ymin = 300 – 300=0= set the bit to 0
Fourth bit: ymax – y = 200 – 300 = -100 = set the bit to 1.

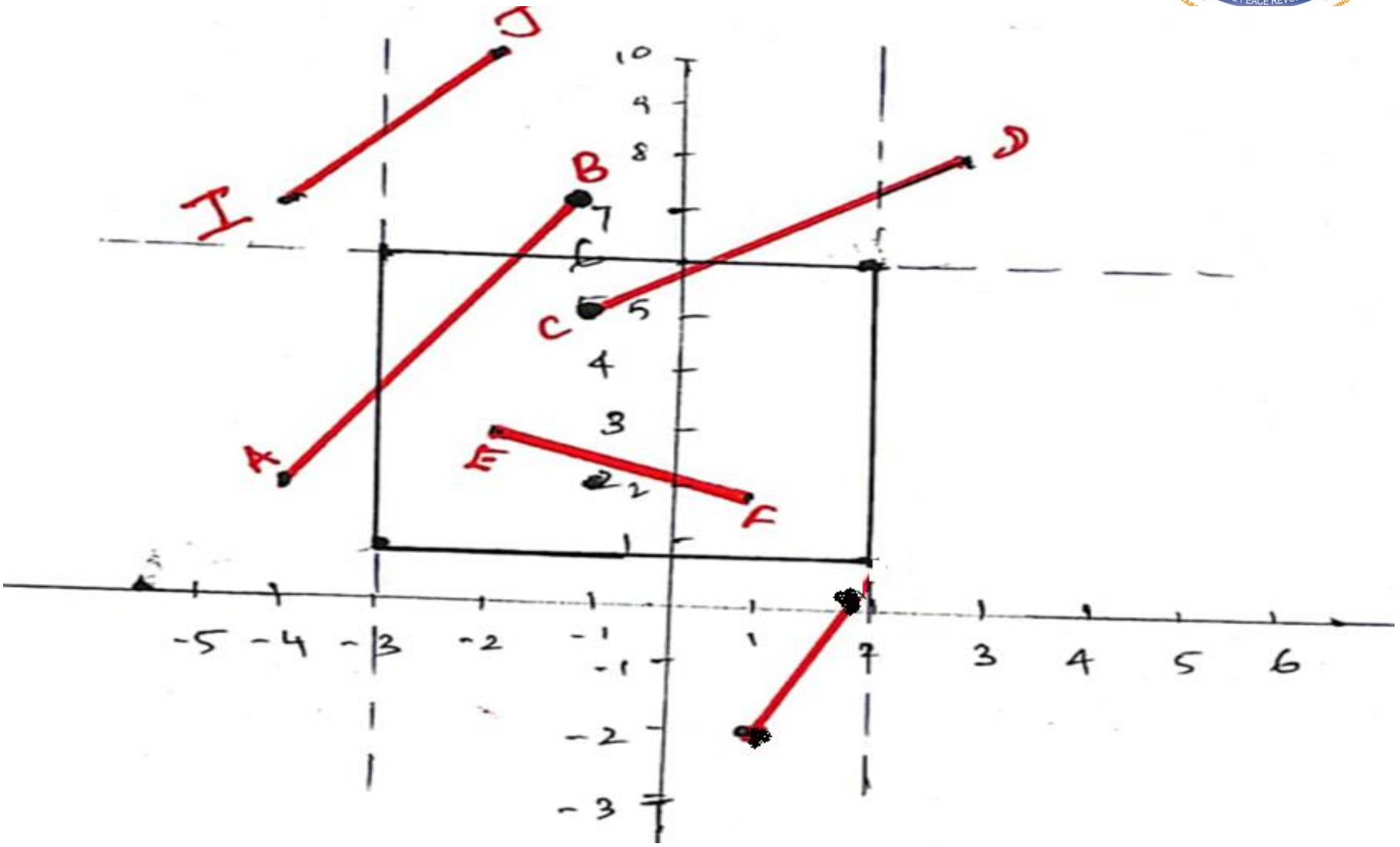
So the region code for P2 is 1010

Bit 1 is the sign bit of $x - xw_{min}$;
bit 2 is the sign bit of $xw_{max} - x$;
bit 3 is the sign bit of $y - yw_{min}$;
bit 4 is the sign bit of $yw_{max} - y$.

$$x = x_1 + (y_{boundary} - y_1) / m$$

Where $y_{boundary}$ is set either to ymin or ymax

Example 2



Example 2

Given:

AB : A (-4,2) B(-1,7)

CD : C (-1,5) D(3,8)

EF : E (-2,3) F(1,2)

GH : G (1,-2) H(1.9,0)

IJ : I (-4,7) J(-2,10)

$(x_{min}, y_{min}) = (-3,1)$

$(x_{max}, y_{max}) = (2,6)$

Example 2

Region Codes:

A(-4,2) = 0001

B(-1,7) = 1000

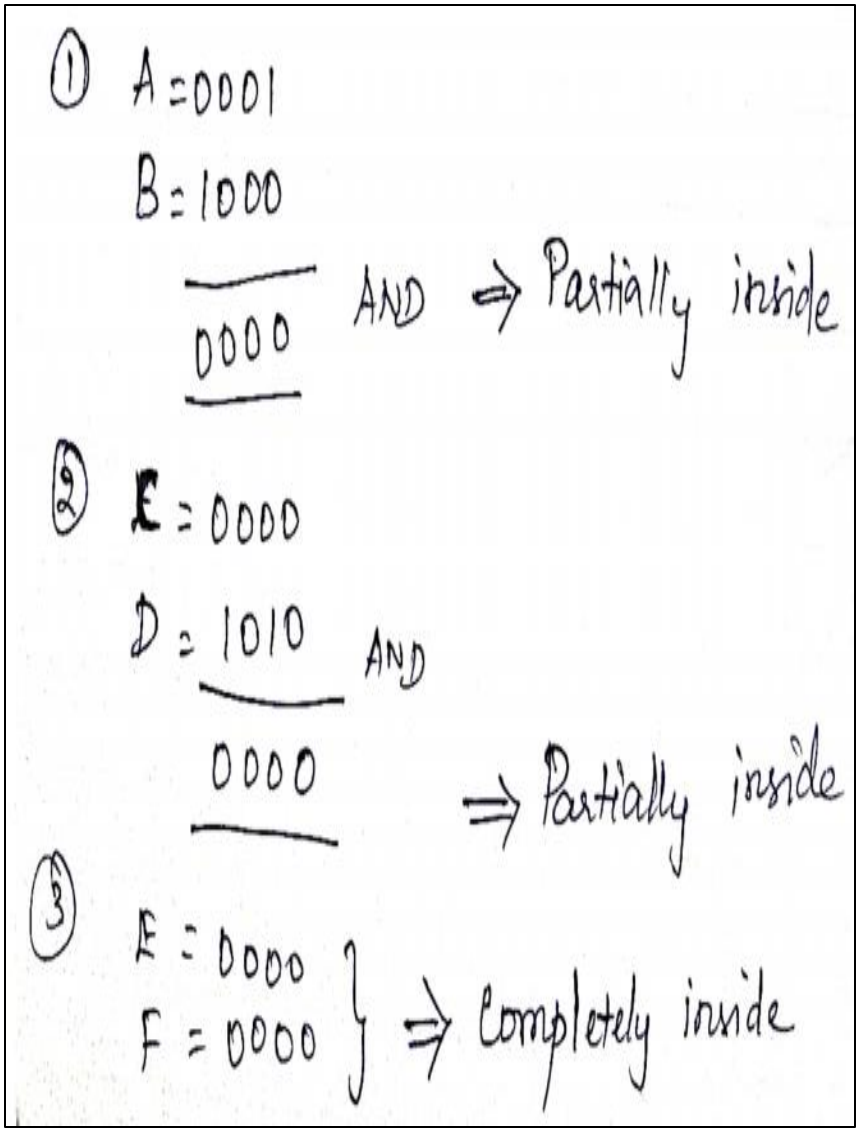
C = 0000 D = 1010

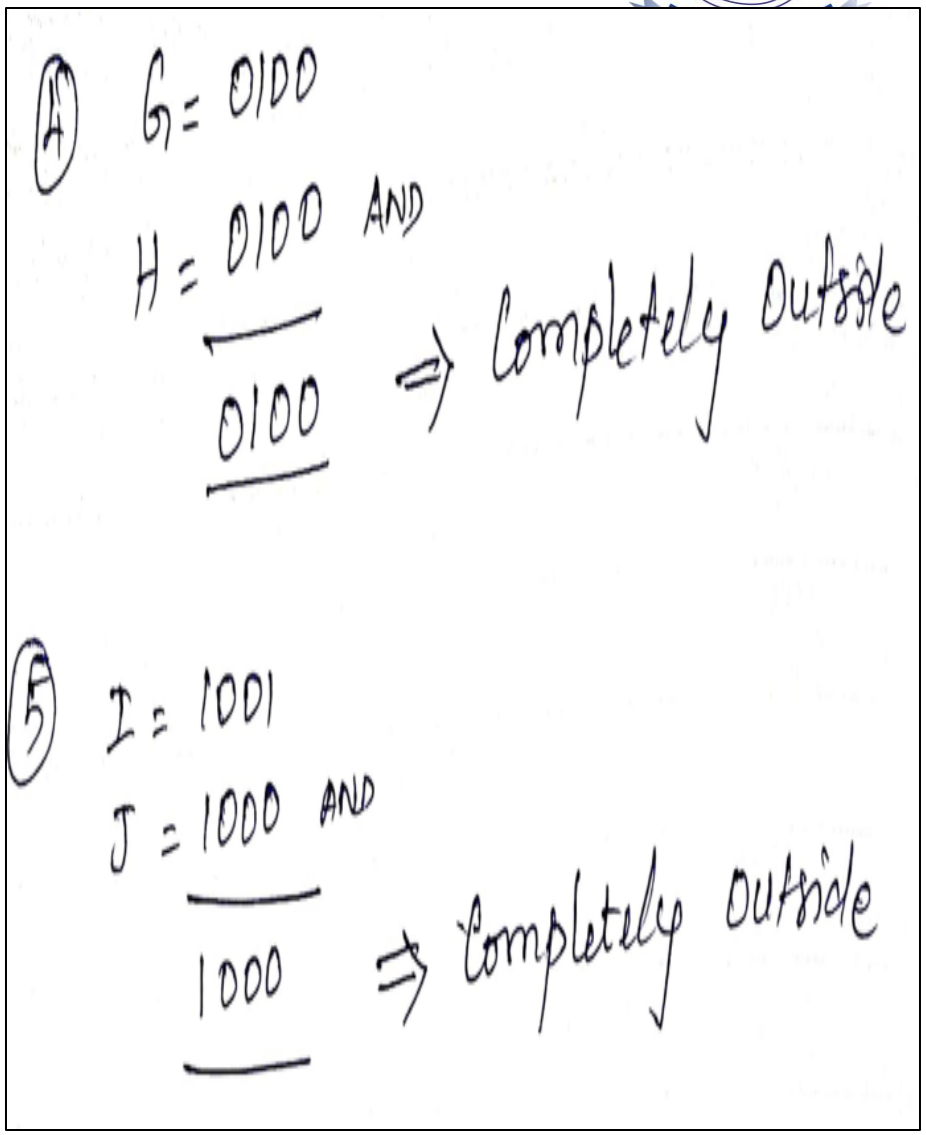
E = 0000 F = 0000

G = 0100 H = 0100

I = 1001 J = 1000

Example 2


① A = 0001
B = 1000
AND
0000
⇒ Partially inside
② E = 0000
D = 1010
AND
0000
⇒ Partially inside
③ E = 0000
F = 0000
⇒ Completely inside


④ G = 0100
H = 0100
AND
0100
⇒ Completely Outside
⑤ I = 1001
J = 1000
AND
1000
⇒ Completely outside

Example 2

∴ For AB and CD, we have to find Intersection points as it is partially inside & Outside.

∴ Preserve/Save : EF Completely

∴ Reject Completely : IJ and GH

Pseudocode

If a line is clipped case, find an intersection with boundaries of the window

$m=(y_2-y_1)(x_2-x_1)$

(a) If bit 1 is "1" line intersects with left boundary of rectangle window

$y_3=y_1+m(x-x_1)$

where $X = X_{wmin}$

where X_{wmin} is the minimum value of X co-ordinate of window

(b) If bit 2 is "1" line intersect with right boundary

$y_3=y_1+m(X-x_1)$

where $X = X_{wmax}$

where X more is maximum value of X co-ordinate of the window

(c) If bit 3 is "1" line intersects with bottom boundary

$x_3=x_1+(y-y_1)/m$

where $y = y_{wmin}$

y_{wmin} is the minimum value of Y co-ordinate of the window

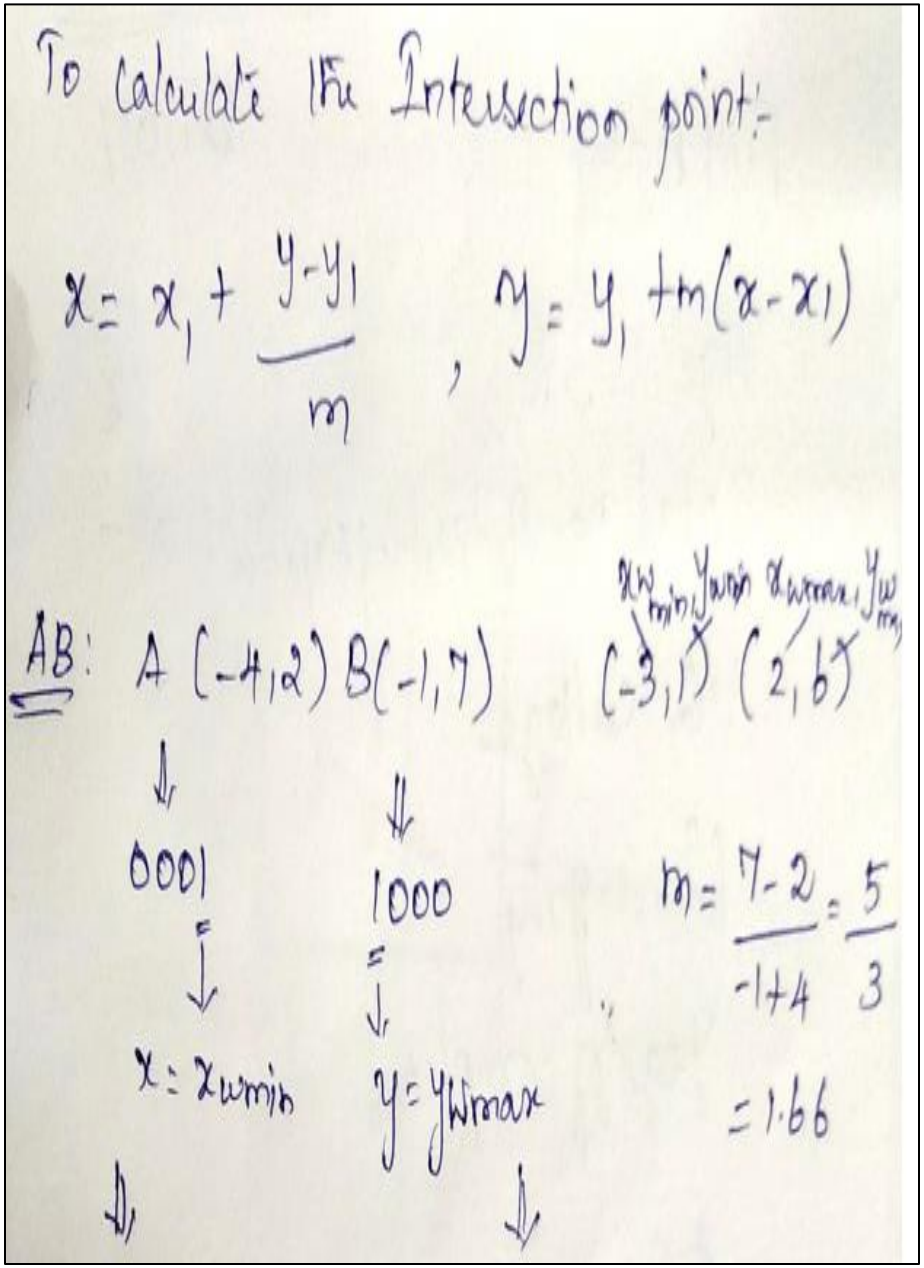
(d) If bit 4 is "1" line intersects with the top boundary

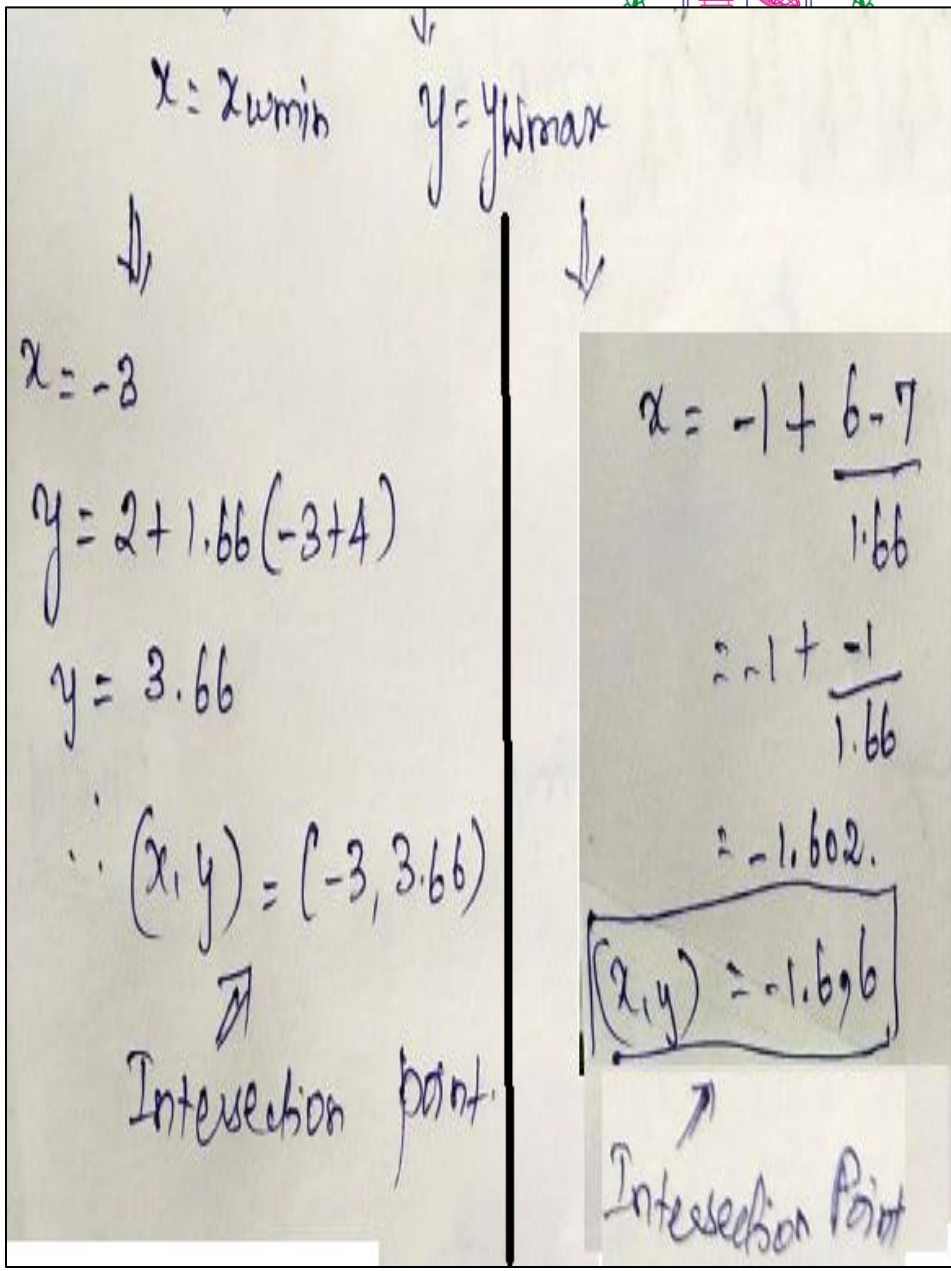
$x_{3=x}1+(y-y_1)/m$

where $y = y_{wmax}$

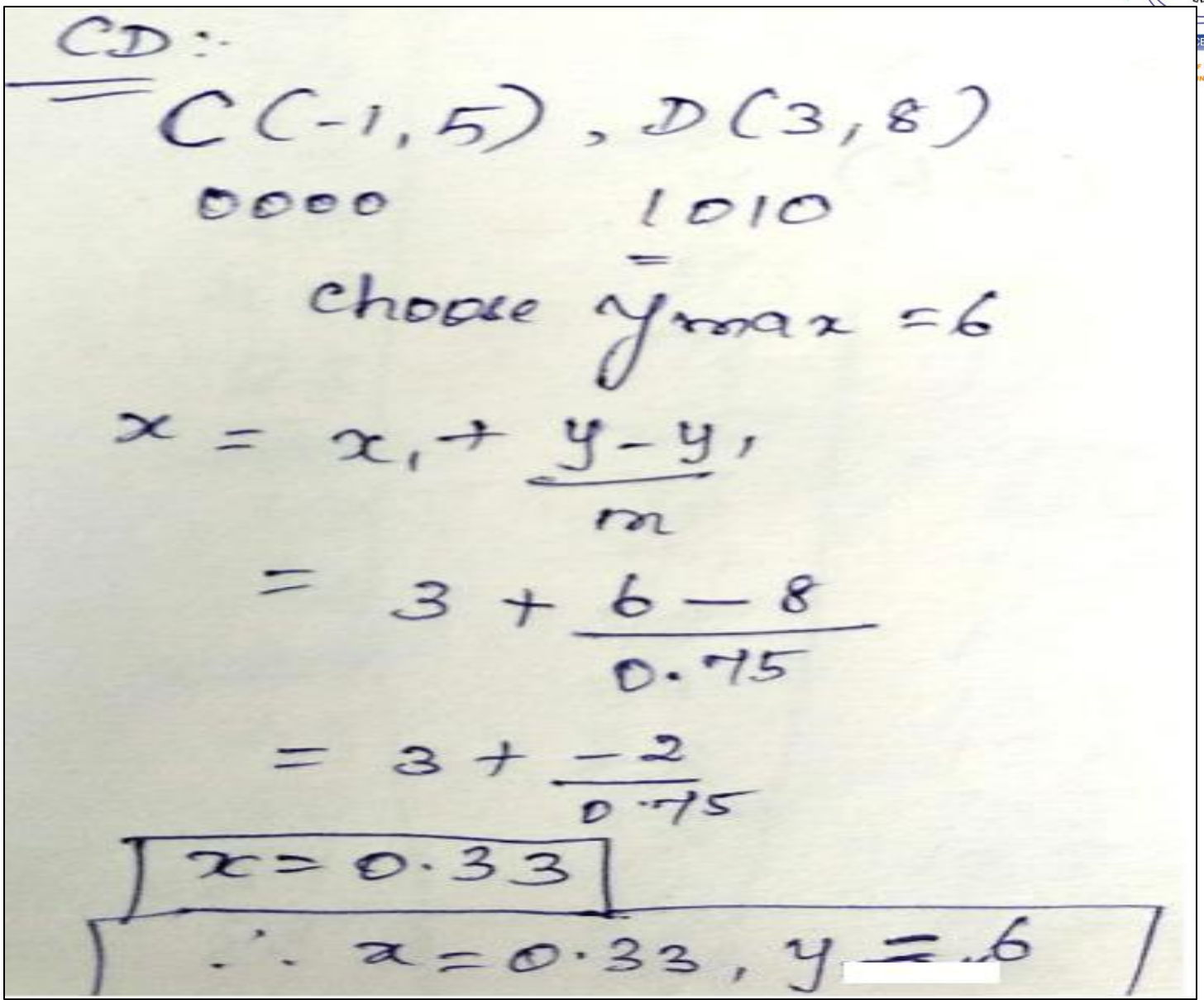
y_{wmax} is the maximum value of Y co-ordinate of the window

Example 2


To calculate the Intersection point:-
$x = x_1 + \frac{y - y_1}{m}$, $y = y_1 + m(x - x_1)$
AB: A (-4,2) B(-1,7)
x_{wmin}, y_{wmin} x_{wmax}, y_{wmax}
(-3,1) (2,6)
0001
1000
$m = \frac{7-2}{-1+4} = \frac{5}{3}$
$x = x_{wmin}$
$y = y_{Wmax}$
$= 1.66$


$x = x_{wmin}$
$y = y_{Wmax}$
$x = -3$
$y = 2 + 1.66(-3+4)$
$y = 3.66$
$\therefore (x, y) = (-3, 3.66)$
Intersection point.
$x = -1 + \frac{6-7}{1.66}$
$= -1 + \frac{-1}{1.66}$
$= -1.602.$
$(x, y) = -1.696$
Intersection Point

Example 2


CD:-
C(-1,5), D(3,8)
0000
1010
Choose ymax = 6
x = x1 + (y - y1)/m
= 3 + (6 - 8)/0.75
= 3 + (-2)/0.75
x = 0.33
∴ x = 0.33, y = 6


SIST
JUSTICE PEACE REVOLUTION
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
DEEMED UNIVERSITY

Thank You