Clipping

Developed by
Dr. Arindam Roy
Assistant Professor, Dept. of Comp Sc & Appl., P. K. College, Contai

Window & Viewport: A world coordinate area selected for display is called a window. An area on a display device to which a window is mapped is called a viewport. The window defines what is to be viewed, but the viewport defines where it is to be displayed. Often, windows and viewports rectangles, with edges are parallel to the coordinate axes. Also, in general, polygon shapes and circles are used in some applications of window and viewport, but these shapes take longer to process. Generally, the mapping of a world coordinate scene to device coordinates is referred to as a viewing transformation. This two-dimensional transformation is called as window-to-viewport transformation. Figure-1 illustrates the mapping of a picture section that falls within a rectangular window onto a designated rectangular viewport.

In Computer Graphics terminology, the term window originally represent an area of a picture that is selected for viewing.

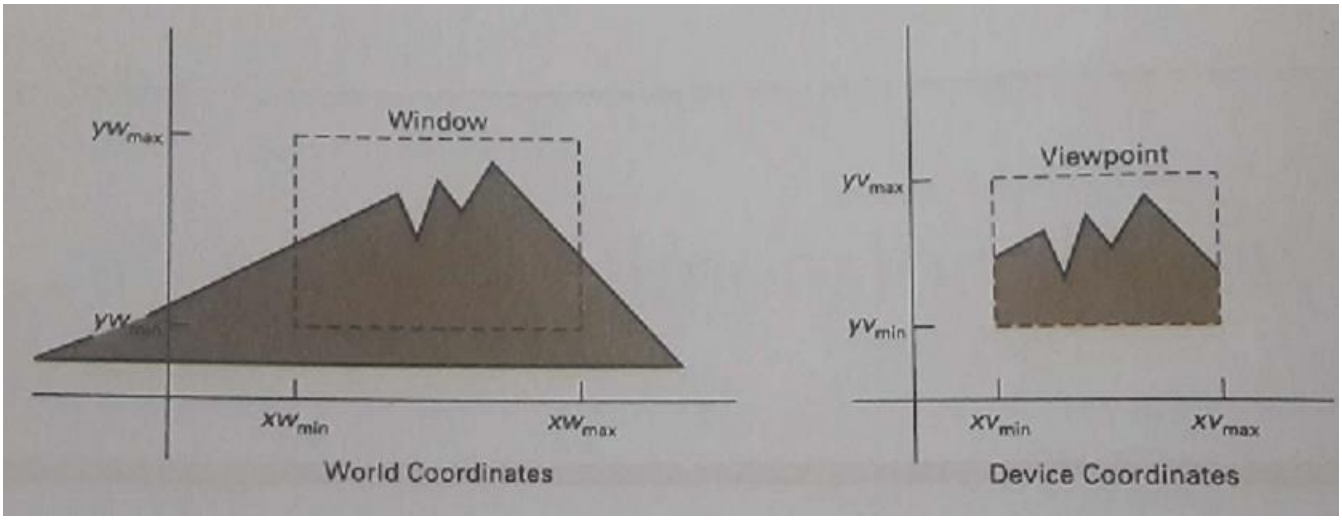

Figure-1: A viewing transformation using standard rectangles for the window and viewport

We can observe the objects at different positions on the display area of an output device. Also, by varying the size of the viewports, we can change the size and properties of the displayed objects. We achieve zooming effects by successively mapping different-sized windows on a fixed-point viewport.

Figure-2 illustrates a rotated viewing-coordinate reference frame and the mapping to normalized coordinates. Clipping procedures are the fundamental importance in computer graphics. They are used not only in viewing transformations, but also in window-manager systems, in painting and drawing packages to eliminate parts of a picture inside or outside of a designated screen area, and in many other applications.

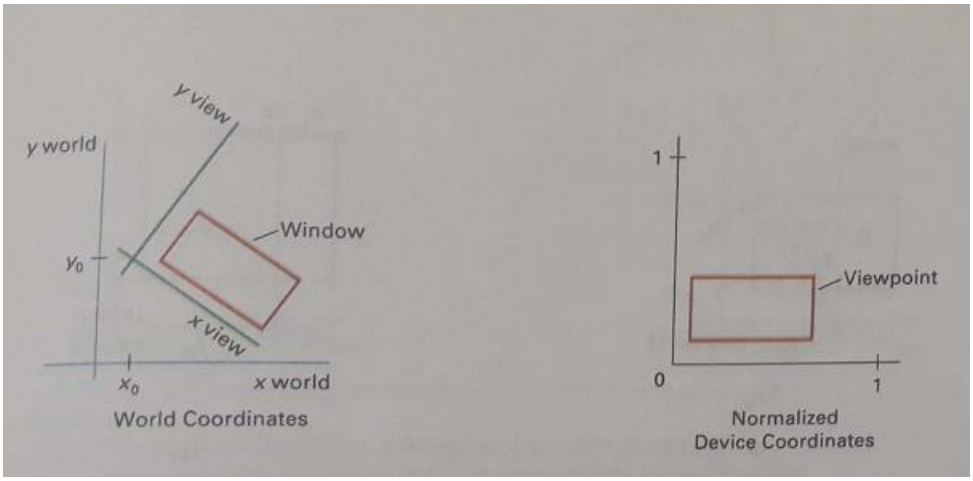

Figure-2: Setting up a rotated world window in viewing coordinates and the corresponding normalized-coordinate viewport

Window-to-Viewport coordinate transformation

Once object description have been transformed to the viewing reference frame, we choose the window extents in viewing coordinates and select the viewport limits in normalized coordinates (Figure-2). Object descriptions are then transferred to normalized device coordinates.

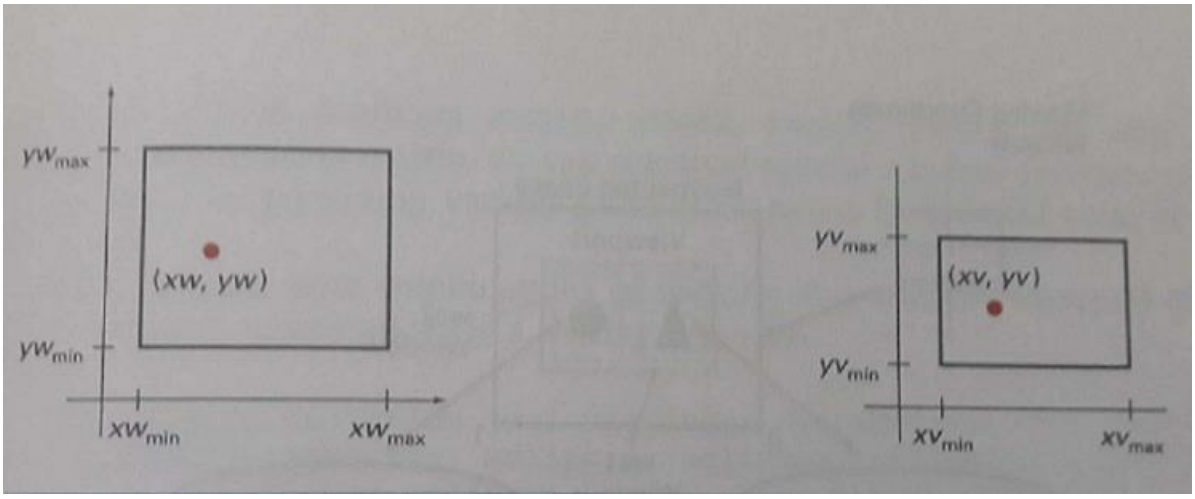

Figure-3: A point at position (xw, yw) in a designated window is mapped to viewport coordinates (xv, yv) so that relative positions in the two areas are the same.

Figure-3 illustrates the window-to-viewport mapping. A point at position (xw, yw) in the window is mapped into the position (xv, yv) in the corresponding viewport. To maintain the same relative placement in the viewport as in the window, we require the following transformations given below.

$$\frac{xv - xv_{min}}{xv_{max} - xv_{min}} = \frac{xw - xw_{min}}{xw_{max} - xw_{min}} \qquad (1)$$

$$\frac{yv - yv_{min}}{yv_{max} - yv_{min}} = \frac{yw - yw_{min}}{yw_{max} - yw_{min}} \quad (2)$$

Solving these above two equations for the point (xv, yv) lying in the viewport, we get

$$xv = xv_{min} + (xw - xw_{min})s_x \quad (3)$$

$$yv = yv_{min} + (yw - yw_{min})s_y \quad (4)$$

Where the scaling factors are

$$s_x = \frac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}} \quad (5)$$

$$s_y = \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}} \quad (6)$$

Equations (3) and (4) can also be derived with a set of transformations that converts the window area into the viewport area.

Clipping: A Procedure that identifies those portions of a picture that are either inside or outside of a specified region of a space is referred to as a **Clipping** algorithm or simply **Clipping** . The region against which an object is to **clipped** is called a Clip Window.

In this process, we consider how to cut off the lines which are outside the window so that only the lines within the window are displayed. In clipping we examine each line of the display to determine whether or not it is completely inside the window, lies completely outside the window, or crosses the window boundary.

- If it is inside, the line will display.
- If it is outside, nothing is drawn.
- If it crosses the boundary, we must determine the point of intersection and draw only those portions which lie inside.

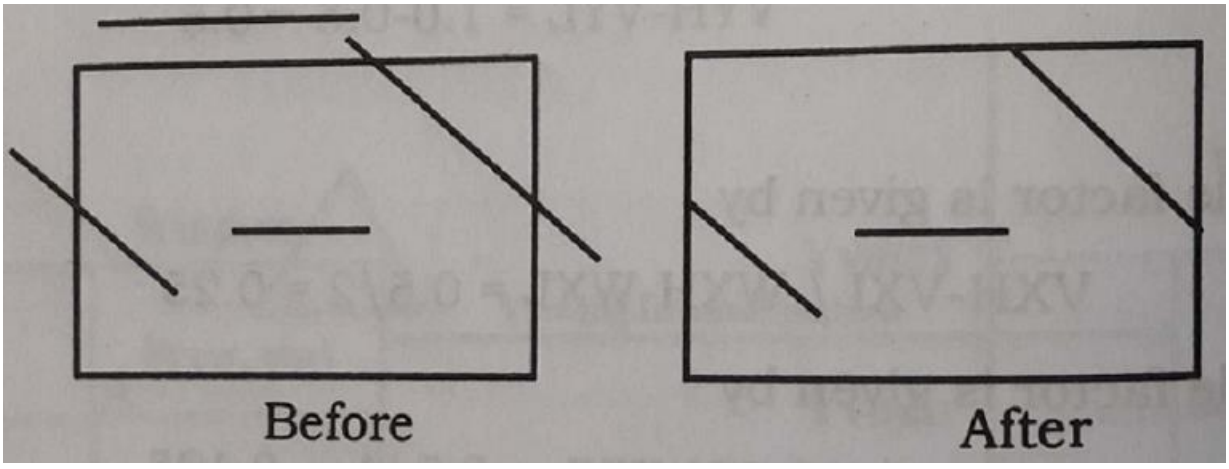

Figure-4: Examples of Clipping

There are many types of Clipping, and different types of clipping algorithm require different graphics operation. Such types of Clippings are

(i) Point Clipping
(ii) Line Clipping (Straight line segments)
(iii) Area Clipping (Polygons)
(iv) Curve Clipping
(v) Text Clipping

Here, we discuss two types of Clipping i.e, Point Clipping & Line Clipping.

Point Clipping: A point P(x, y) can be saved for display of clipping window, if the following inequalities are satisfied.

$$xw_{min} \leq x \leq xw_{max}$$

$$yw_{min} \leq y \leq yw_{max}$$

Where xw_{min} and xw_{max} are edges of the boundaries of clip window parallel to Y axis and yw_{min} and yw_{max} are edges of the boundaries of clip window parallel to the X axis.

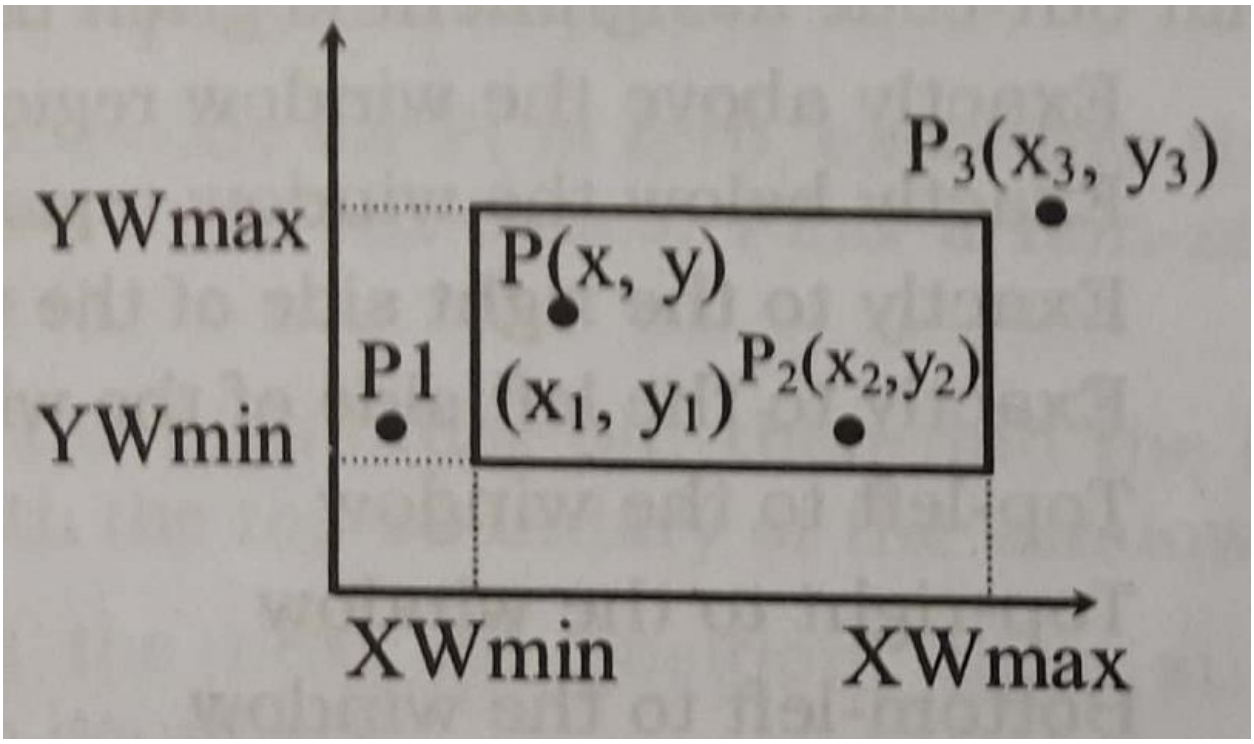

Figure-5: Point Clipping

In the above shown Figure-5, the points P(x, y) and P_2(x, y) are satisfying the above stated inequalities so that we may not clip those points. However, the points P_1(x, y) and P_3(x, y) will be clipped and will not be displayed.

Line Clipping: In line clipping, we examine each line of the display to determine whether it is completely inside the window, lies completely outside the window, or crosses the window boundary.

- If the line is inside, the line will display.
- If the line is outside, nothing is drawn.
- If the line crosses the boundary, we must determine the point of intersection and draw only those portions, which lie inside.

Cohen-Sutherland Line Clipping Algorithm: Cohen-Sutherland proposed a popular method / algorithm for clipping lines. This algorithm quickly removes lines, which lie entirely one-side of the clipping region (both endpoints above, or below, or right, or left). It makes clever use of bit operations (out codes) to perform the required task efficiently. These segment endpoints are each given4-bit binary codes.

- The high-order bit is set to 1, if the point is above the window.
- Next bit is set to 1, if the point is below the window.
- The third bit 1, indicate the region is right of the window.
- The third bit 1, indicate the region is right of the window.
  So, the lines which form the window boundary divide the plane into nine regions with the codes:
- If the line is entirely within the window, then both endpoints will have the out-code as 0000. The out-codes of other regions are given below:
  1000 - Exactly above the window region.
  0100 - Exactly below the window region.
  0010 - Exactly to the right side of the window.
  0001 - Exactly to the left side of the window.
  1001 - Top-left to the window.
  1010 - Top-right to the window.
  0101 - Bottom-left to the window.
  0110 - Bottom right to the window.
- Now, we check to see if the line is entirely on one side of the window by taking the logical AND of the out-codes for the two endpoints. If the result of the AND operation is non-zero, then the line segment may be rejected.

Thus one test can be decided if the line segment is entirely above, or entirely below, or entirely to the right, or entirely to the left of the window.

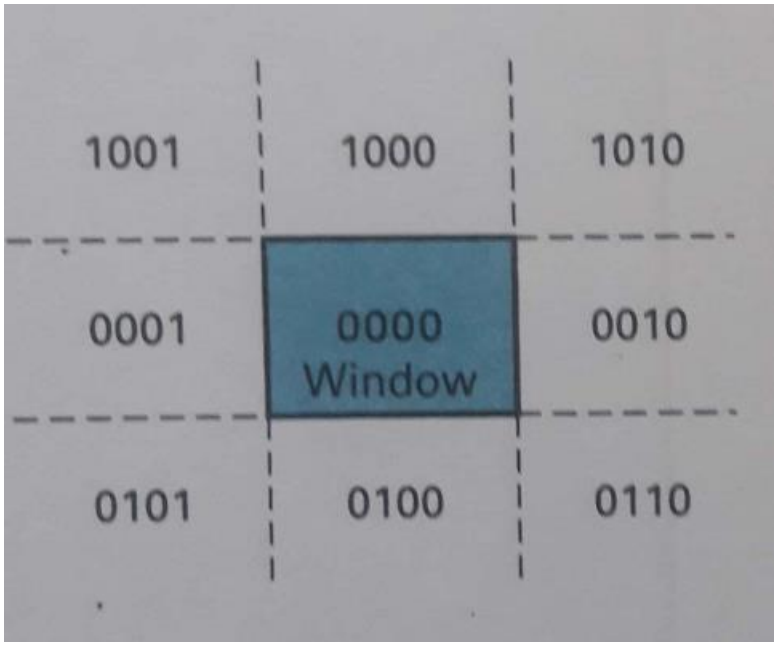

Figure-6: Out-code representation

Cohen-Sutherland Line Clipping Algorithm:

Step-1: First, we compute the out-codes for the two endpoints (P_1 and P_2) of the segment.

Step-2: Enter into the loop

- Within the loop, we check to see if both out-codes are zero: if so, enter the segment into the display file, exit the loop, and return.
- If the out-codes are not both zero, then we perform the logical, function, and check a non-zero result: if this test is non-zero, then we reject the line, exit the loop, and return.

Step-3: If neither of these tests is satisfied, we must subdivide the line segment and repeat the loop.

- If the out-code for P_1 is zero, exchange the points P_1 and P_2 and also their out-codes. Find a non-zero bit in the out-code of P_1.
- If it is the high-order bit, then find the intersection of the line with the top boundary of the window.
- If it is the next bit position, then subdivide along the bottom boundary.
- The other two bits indicate that the right and left boundaries should be used.

Step-4: Replace the point P_1 with the intersection point and calculate it's out-code. Repeat the loop.

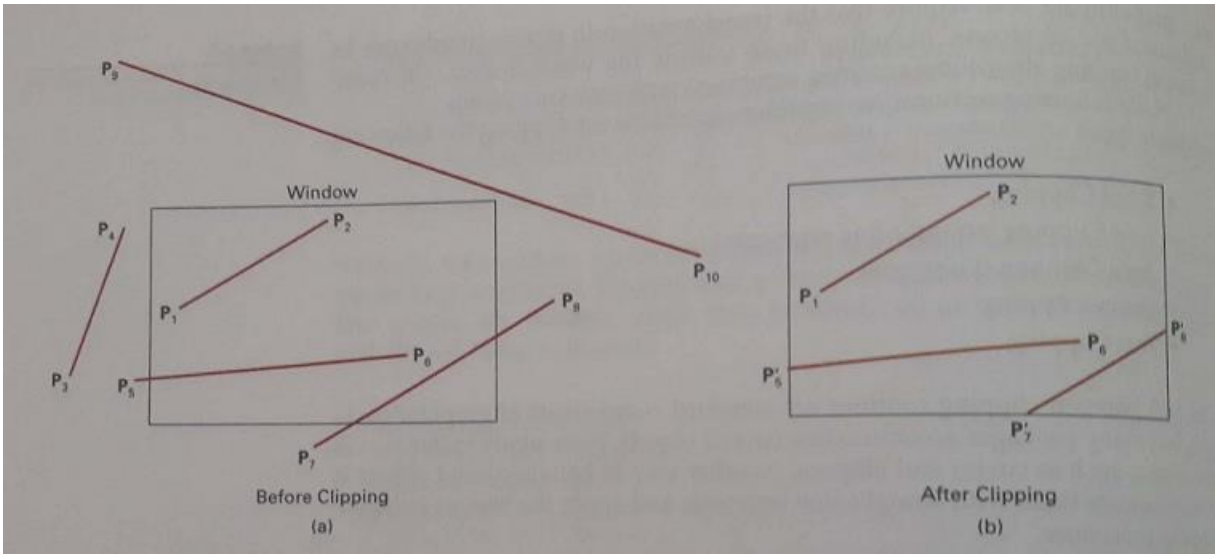

Figure-7: Before Clipping and after Clipping

Example:

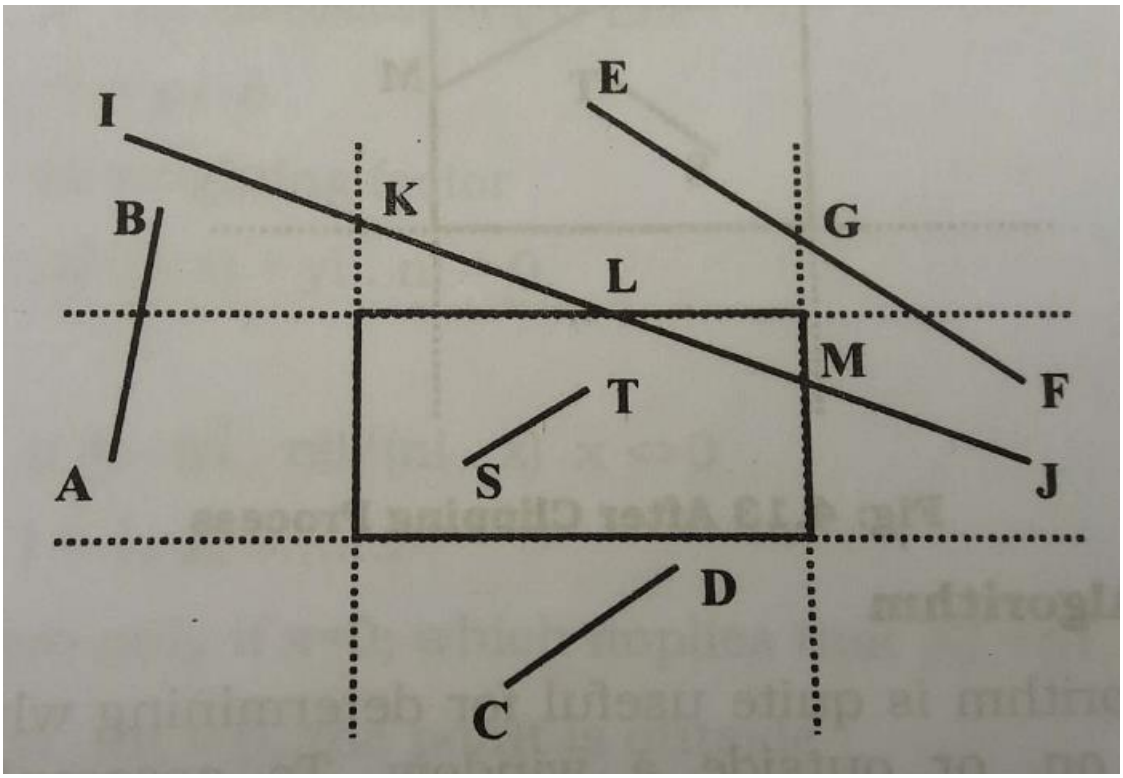

Figure-8: Cohen-Sutherland Algorithm example

In the above shown example, the considered line segments are AB, CD, EF, IJ, and ST. From the above Cohen-Sutherland line clipping algorithm we know that the line segments
AB - completely outside of the window bso it is eliminated.
CD - completely below the window so it is eliminated.

EF - it is also eliminated.
IJ - crosses the window boundary, first we subdivide the line segment into IK, KL, LM, and MJ. Here we consider the intersection points on the window boundary and perform the logical AND operation between the out-codes of the two endpoints. If we find the non-zero result, then reject the line segment otherwise consider to display into window region.
For example logical AND operation the line segments KL and MJ.
KL out-code : 1000
Logical AND
MJ out-code : 0010
We got, the result as 0000, so the LM line segment is consider for display.
ST - it is completely inside the window boundary, so consider to display.

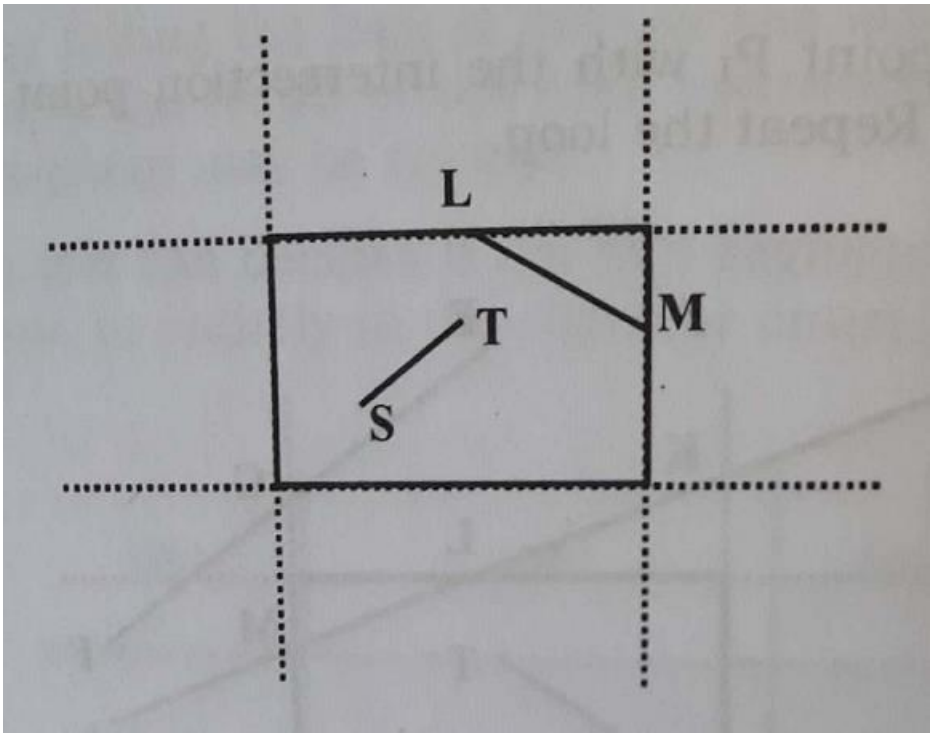

Figure-9: After Clipping process