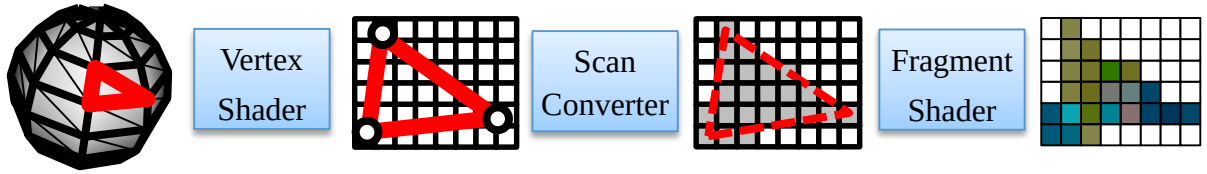
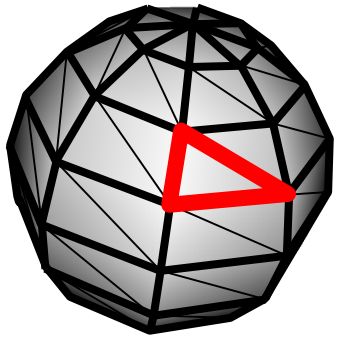
Clipping

CS418 Computer Graphics

John C. Hart

Vertex Pipeline





Model Coords

Model Xform

World Coords

Viewing Xform

Viewing Coords

Projection

Clip Coords.

Clipping

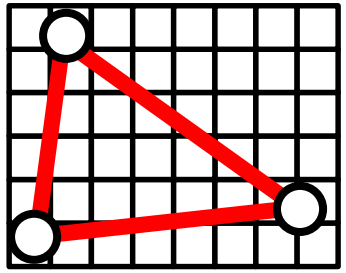
Still Clip Coords.

Homogeneous Divide

Window Coordinates

Window to Viewport

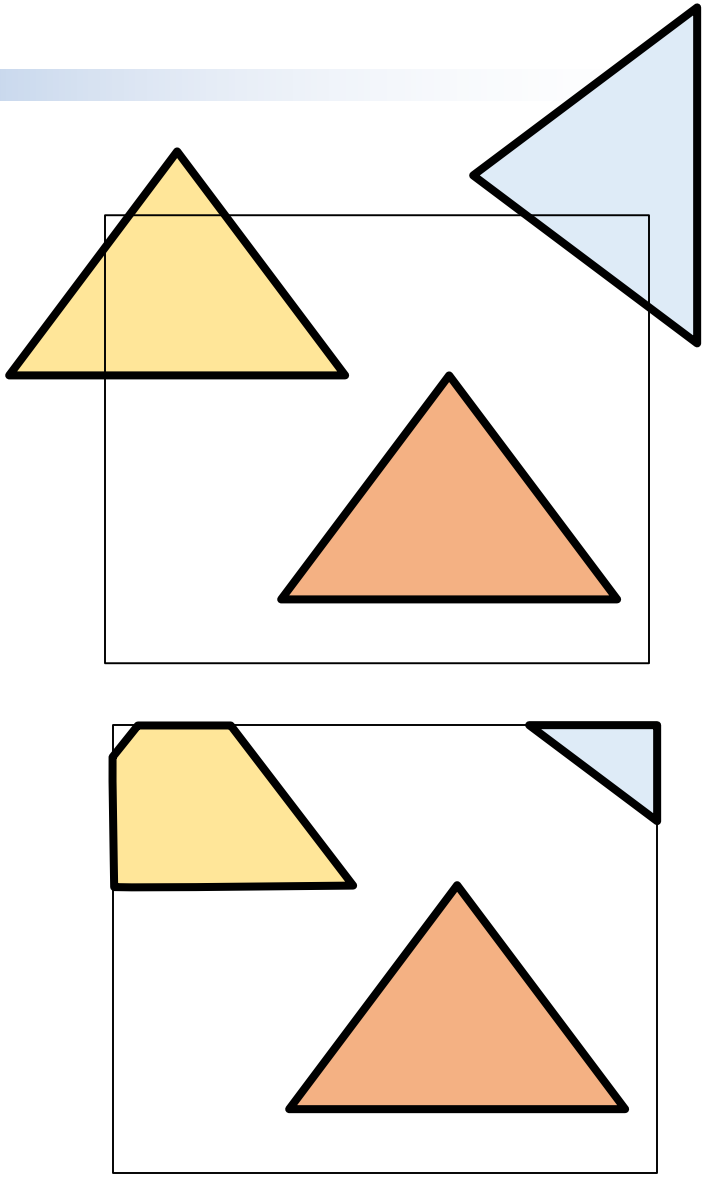
Viewport Coordinates



Why Clip?

Why not just transform all triangles to the screen and just ignore pixels off the screen?

- Takes time to rasterize a triangle
- Very small number of triangles fall within the viewing frustum
- Output may not go directly to screen



Outcodes

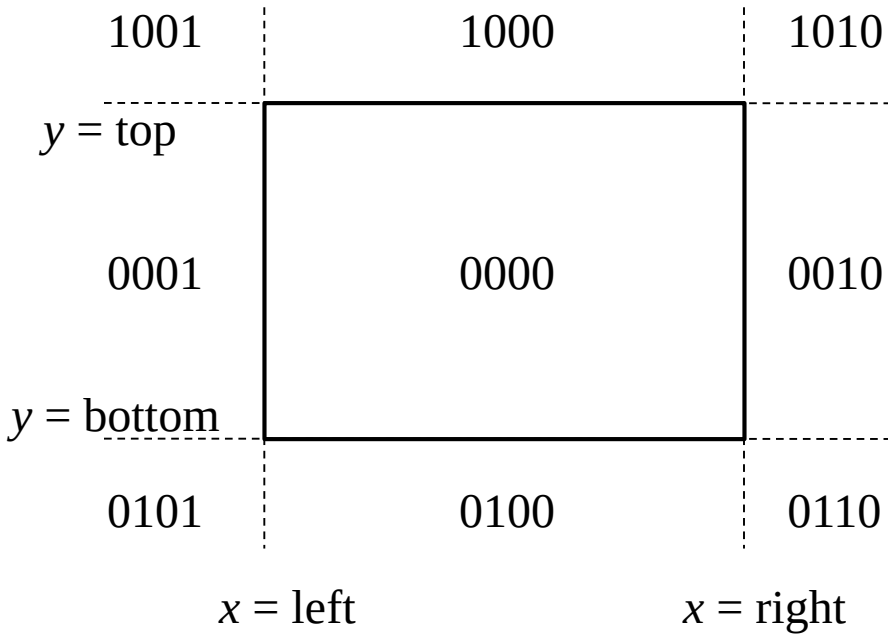
- Cohen-Sutherland algorithm
- Assign segment endpoints a bitcode: $b_3b_2b_1b_0$

  $b_0 = x <$ left

  $b_1 = x >$ right

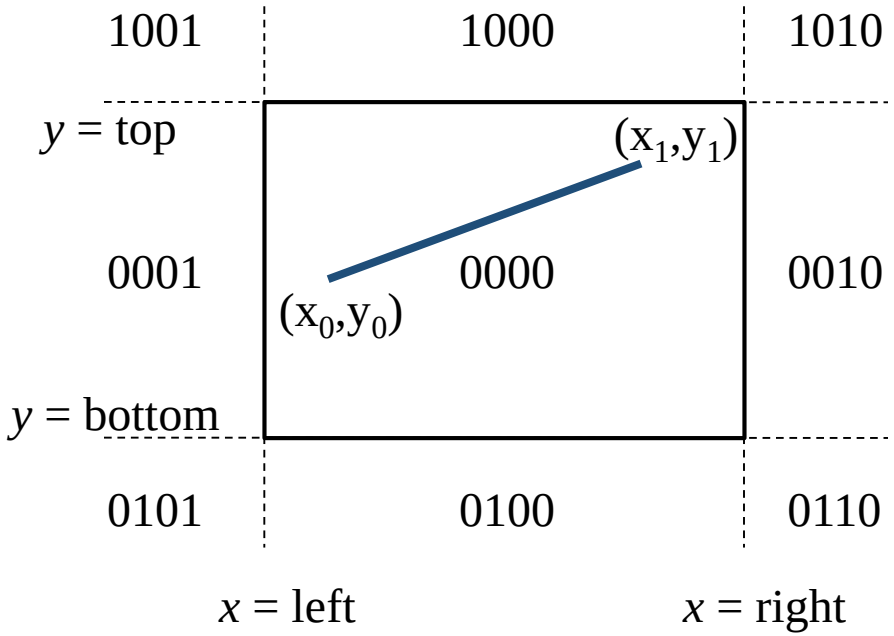
  $b_2 = y <$ bottom

  $b_3 = y >$ top



Outcodes

- Cohen-Sutherland algorithm
- Assign segment endpoints a bitcode: $b_3b_2b_1b_0$
  - $b_0 = x <$ left
  - $b_1 = x >$ right
  - $b_2 = y <$ bottom
  - $b_3 = y >$ top
- Let o_0 = outcode(x_0,y_0), o_1 = outcode(x_1,y_1)



Outcodes

- Cohen-Sutherland algorithm
- Assign segment endpoints a bitcode: $b_3b_2b_1b_0$

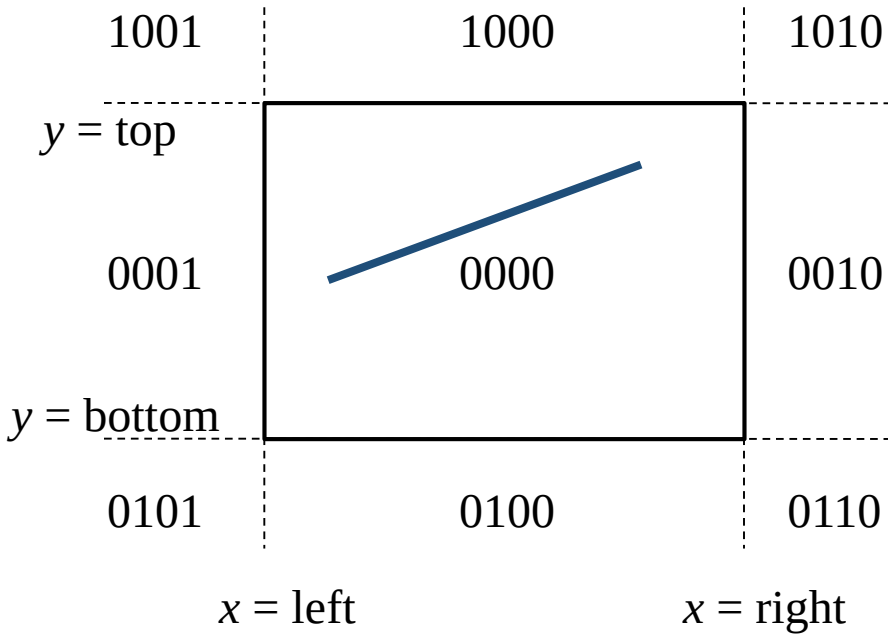
  $b_0 = x <$ left

  $b_1 = x >$ right

  $b_2 = y <$ bottom

  $b_3 = y >$ top
- Let o_0 = outcode(x_0,y_0),
  o_1 = outcode(x_1,y_1)

  $o_0 = o_1 = 0$: segment visible



Outcodes

- Cohen-Sutherland algorithm
- Assign segment endpoints a bitcode: $b_3b_2b_1b_0$

  $b_0 = x <$ left

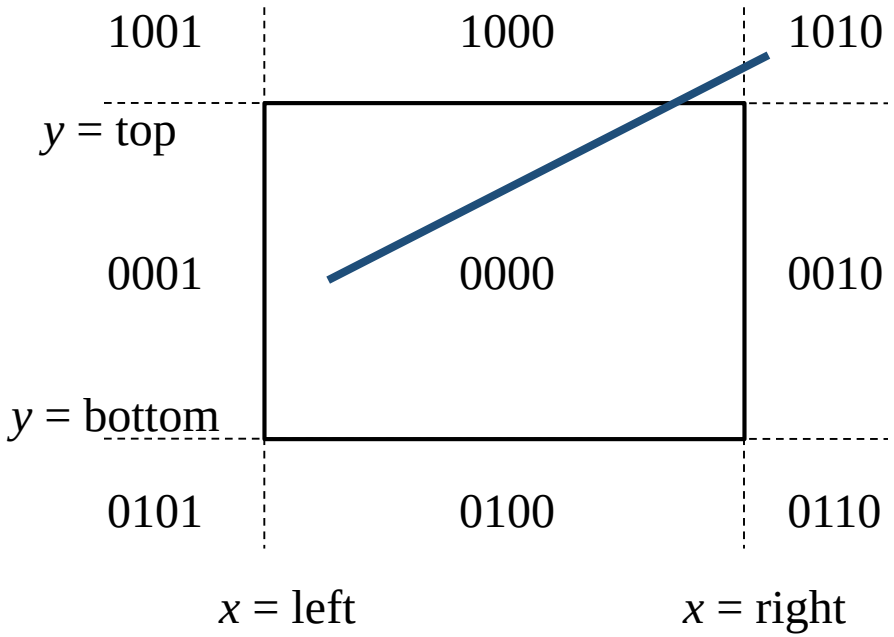
  $b_1 = x >$ right

  $b_2 = y <$ bottom

  $b_3 = y >$ top
- Let o_0 = outcode(x_0,y_0),
  o_1 = outcode(x_1,y_1)

  $o_0 = o_1 = 0$: segment visible

  $o_0 = 0$, $o_1 \neq 0$: segment must be clipped



Outcodes

- Cohen-Sutherland algorithm
- Assign segment endpoints a bitcode: $b_3b_2b_1b_0$

  $b_0 = x <$ left

  $b_1 = x >$ right

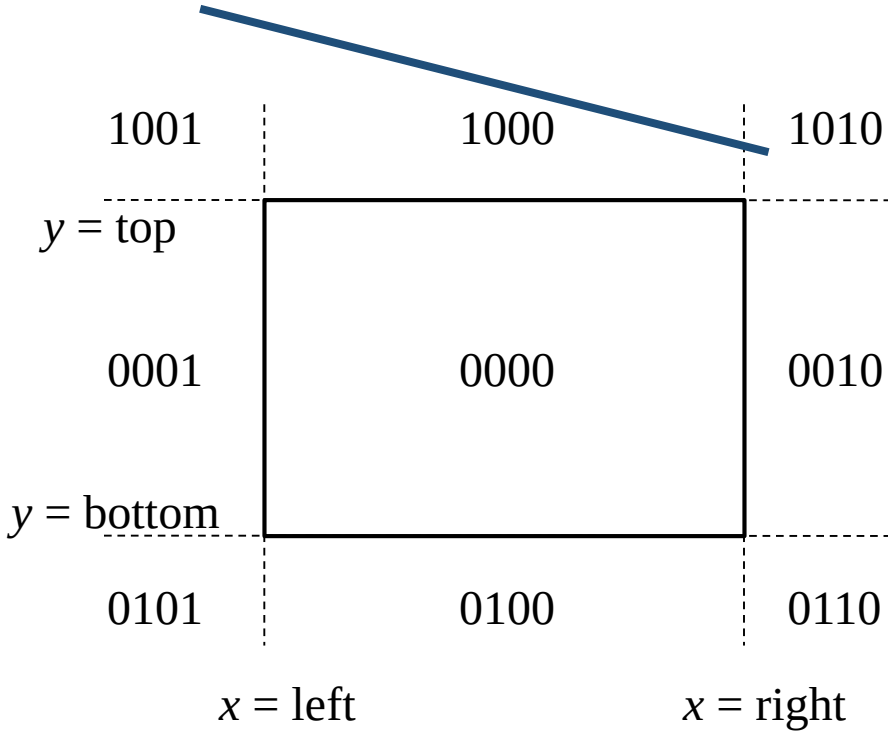
  $b_2 = y <$ bottom

  $b_3 = y >$ top
- Let o_0 = outcode(x_0,y_0),
  o_1 = outcode(x_1,y_1)

  $o_0 = o_1 = 0$: segment visible

  $o_0 = 0$, $o_1 \neq 0$: segment must be clipped

  o_0 & $o_1 \neq 0$: segment can be ignored



Outcodes

- Cohen-Sutherland algorithm
- Assign segment endpoints a bitcode: $b_3b_2b_1b_0$

  $b_0 = x <$ left

  $b_1 = x >$ right

  $b_2 = y <$ bottom

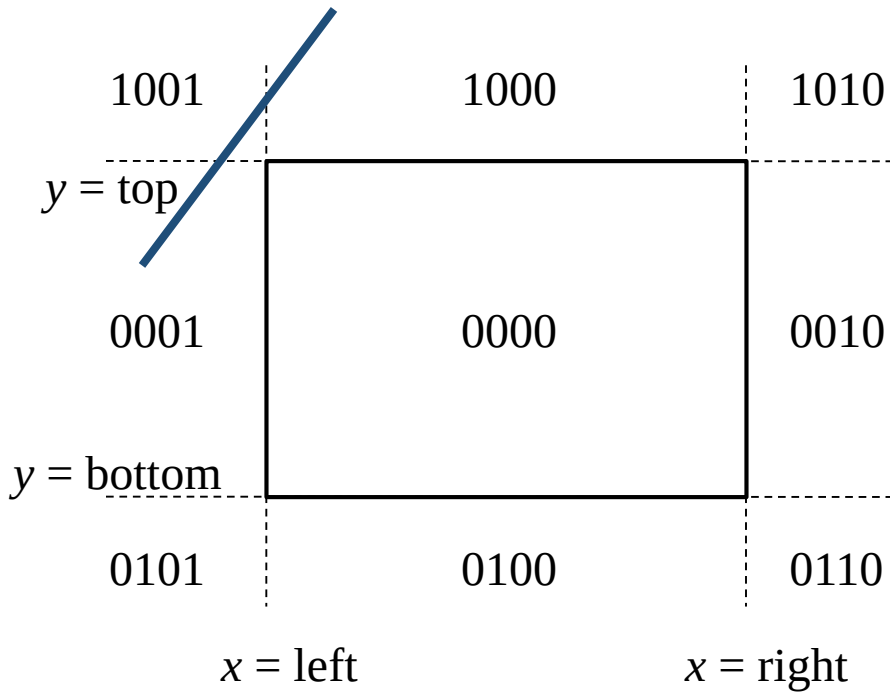
  $b_3 = y >$ top
- Let o_0 = outcode(x_0,y_0),
  o_1 = outcode(x_1,y_1)

  $o_0 = o_1 = 0$: segment visible

  $o_0 = 0$, $o_1 \neq 0$: segment must be clipped

  o_0 & $o_1 \neq 0$: segment can be ignored

  o_0 & $o_1 = 0$: segment might need clipping



Outcodes

- Cohen-Sutherland algorithm
- Assign segment endpoints a bitcode: $b_3b_2b_1b_0$

  $b_0 = x <$ left

  $b_1 = x >$ right

  $b_2 = y <$ bottom

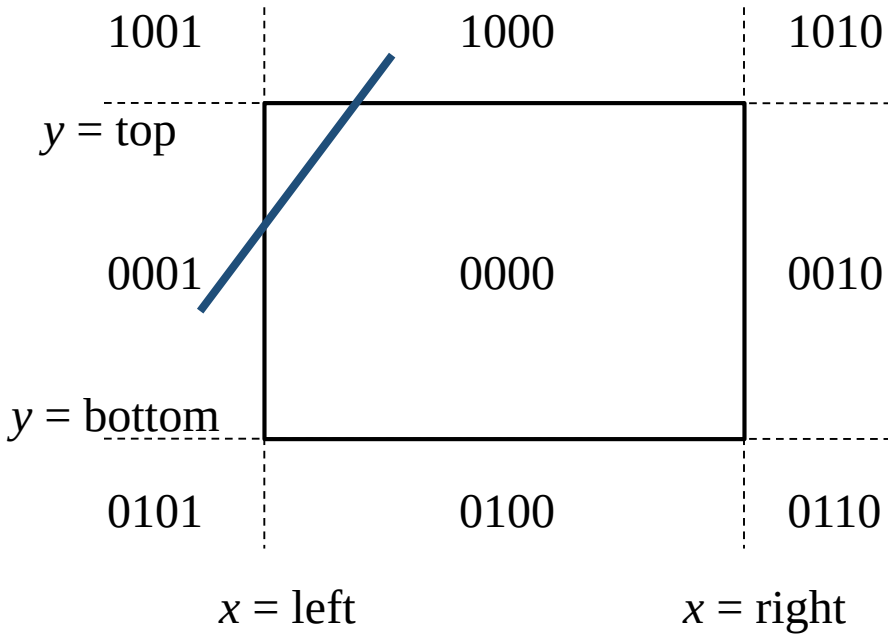
  $b_3 = y >$ top
- Let o_0 = outcode(x_0,y_0),
  o_1 = outcode(x_1,y_1)

  $o_0 = o_1 = 0$: segment visible

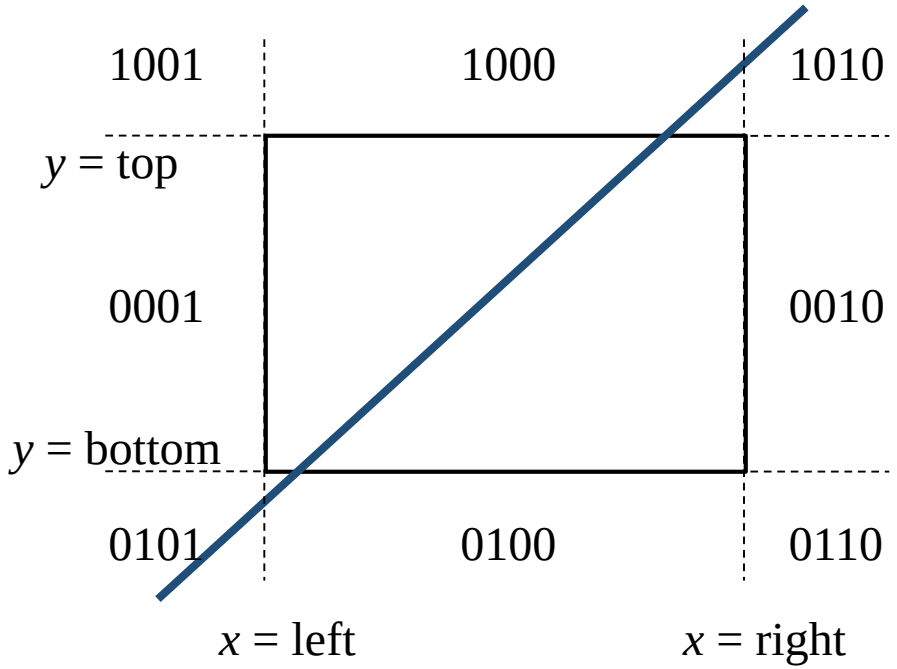
  $o_0 = 0$, $o_1 \neq 0$: segment must be clipped

  o_0 & $o_1 \neq 0$: segment can be ignored

  o_0 & $o_1 = 0$: segment might need clipping

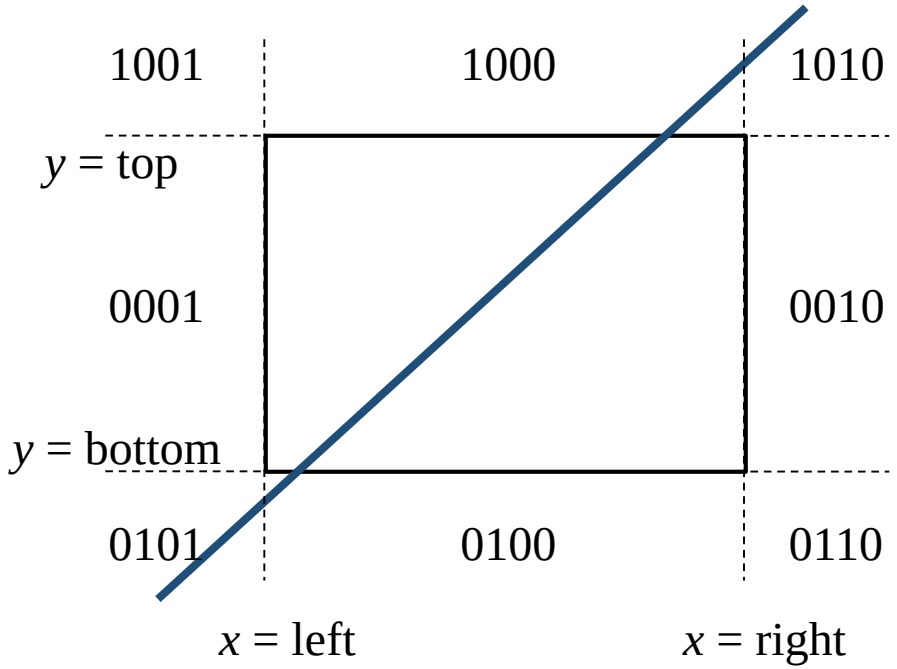


Serial Clipping



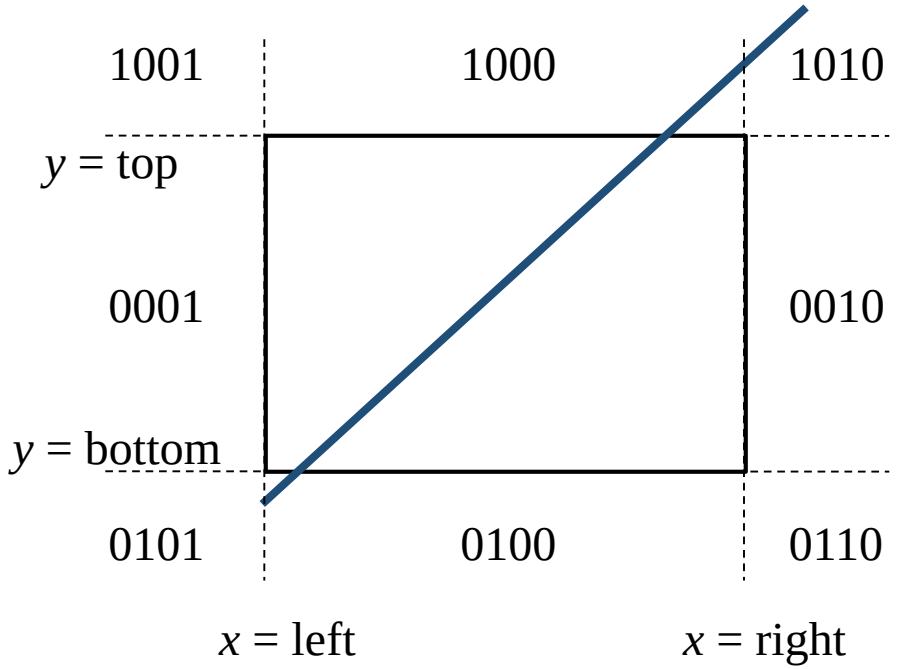
Serial Clipping

- First clip 0001



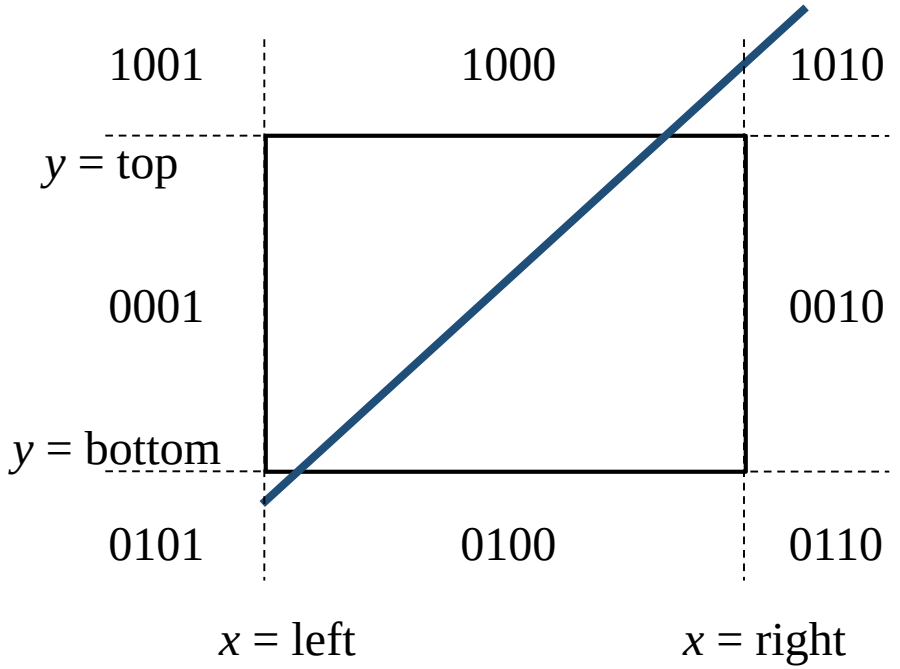
Serial Clipping

- First clip 0001
- Move (x_0,y_0) to (left,…)



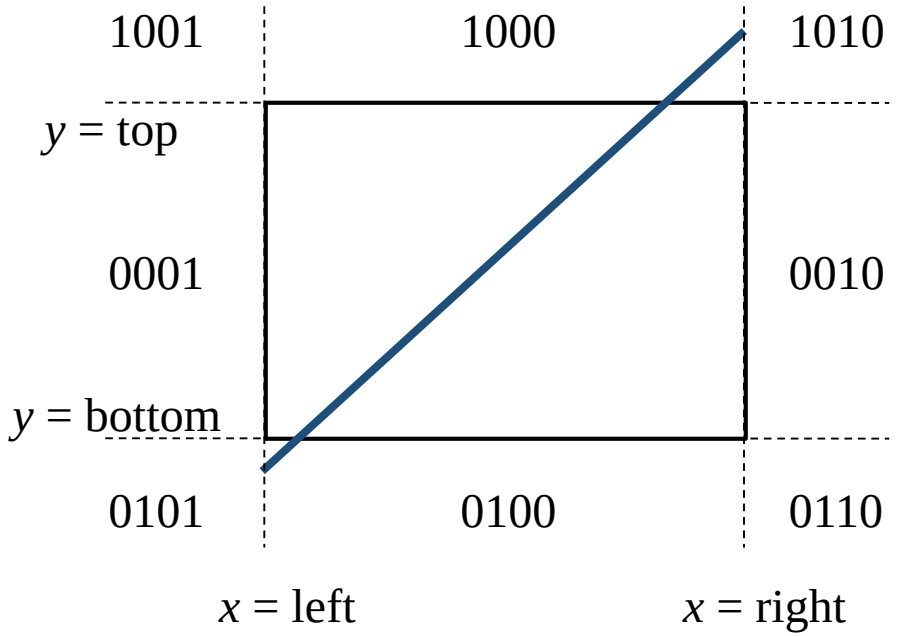
Serial Clipping

- First clip 0001
- Move (x_0,y_0) to (left,…)
- Then clip 0010



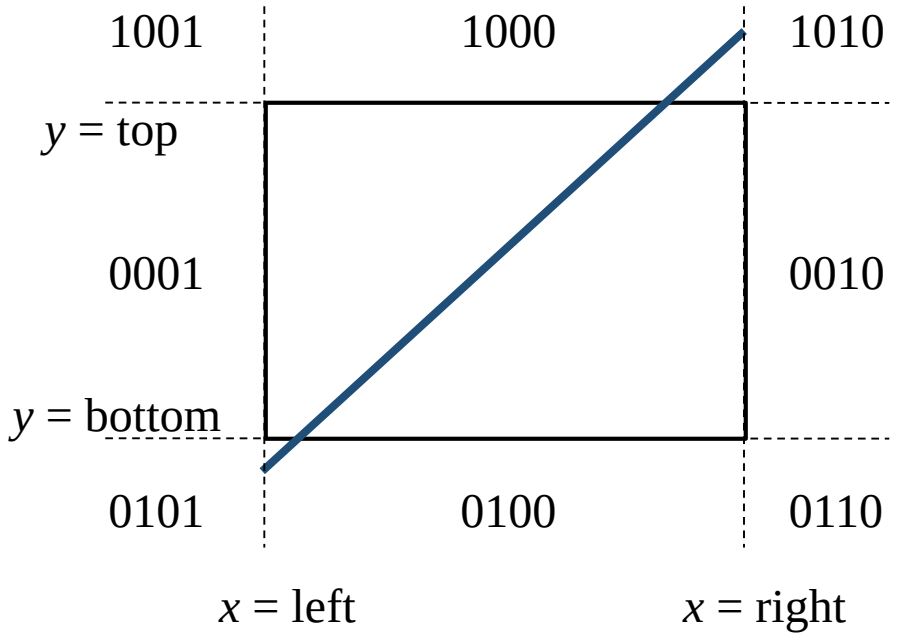
Serial Clipping

- First clip 0001
- Move (x_0,y_0) to (left,…)
- Then clip 0010
- Move (x_1,y_1) to (right,…)



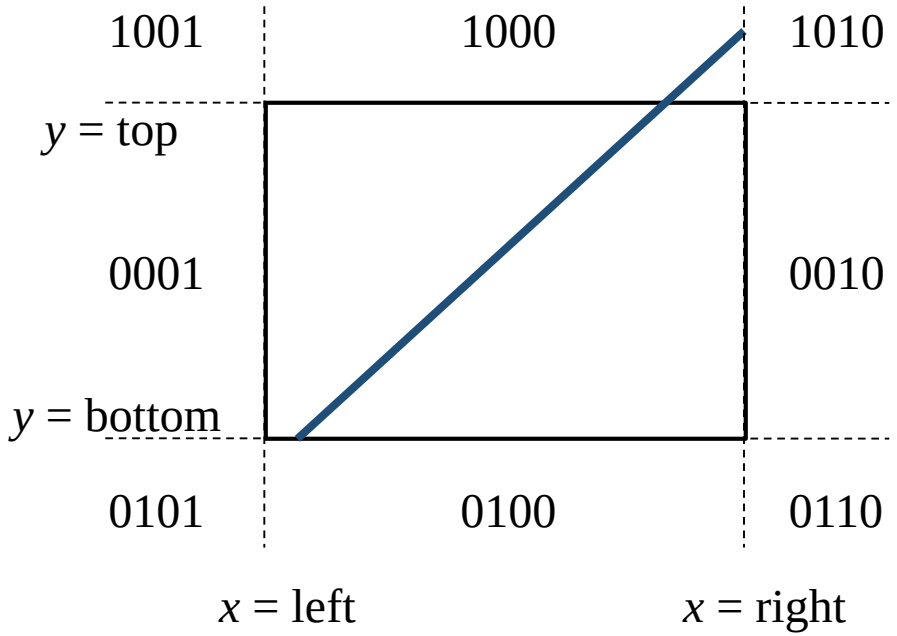
Serial Clipping

- First clip 0001
- Move (x_0,y_0) to (left,…)
- Then clip 0010
- Move (x_1,y_1) to (right,…)
- Then clip 0100



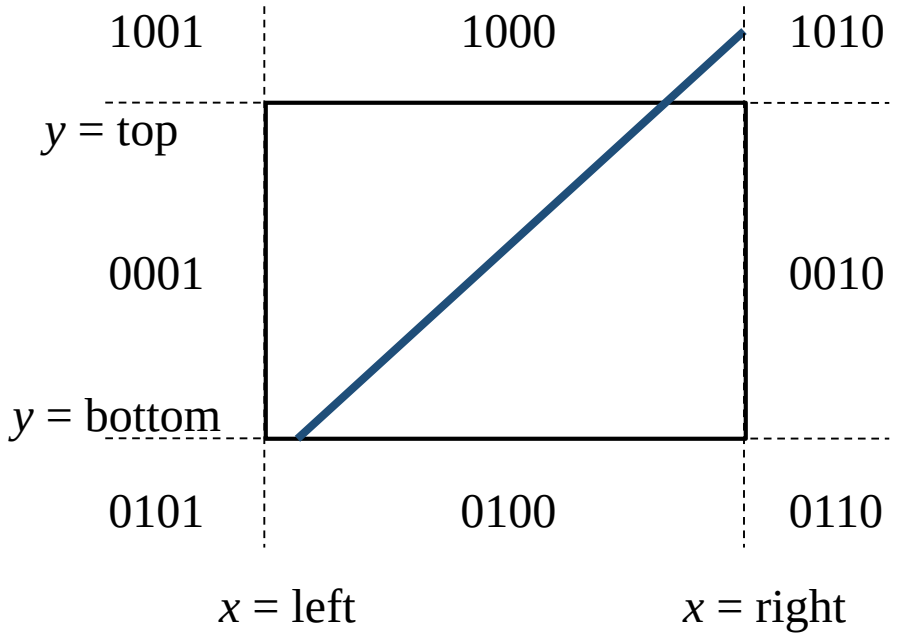
Serial Clipping

- First clip 0001
- Move (x_0,y_0) to (left,…)
- Then clip 0010
- Move (x_1,y_1) to (right,…)
- Then clip 0100
- Move (x_0,y_0) again, now to (…,bottom)



Serial Clipping

- First clip 0001
- Move (x_0,y_0) to (left,…)
- Then clip 0010
- Move (x_1,y_1) to (right,…)
- Then clip 0100
- Move (x_0,y_0) again, now to (…,bottom)
- Finally clip 1000



Serial Clipping

- First clip 0001
- Move (x_0,y_0) to (left,…)
- Then clip 0010
- Move (x_1,y_1) to (right,…)
- Then clip 0100
- Move (x_0,y_0) again, now to (…,bottom)
- Finally clip 1000
- Move (x_1,y_1) again, now to (…,top)