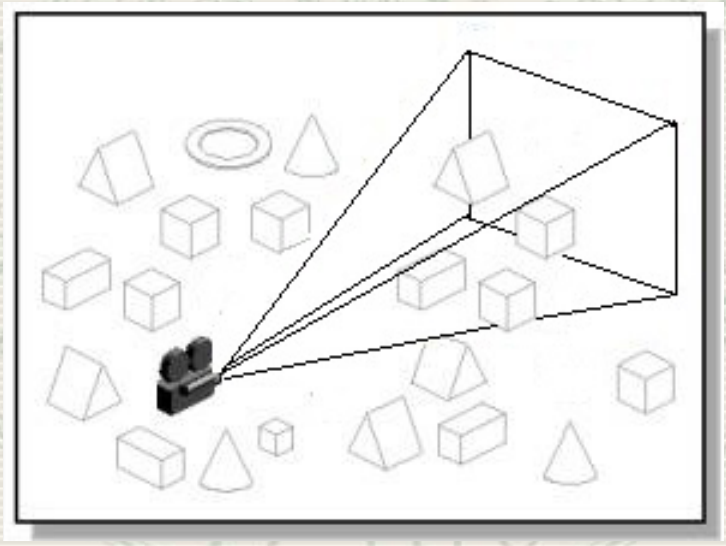
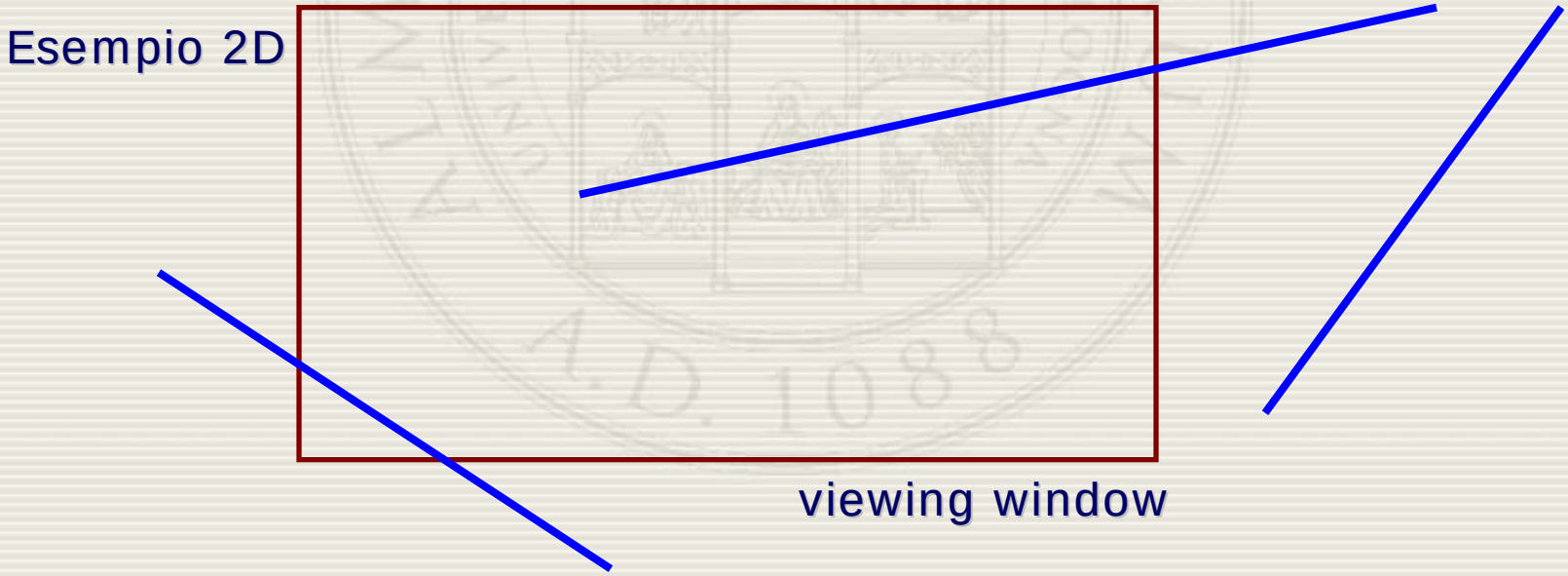
Clipping di punti e linee



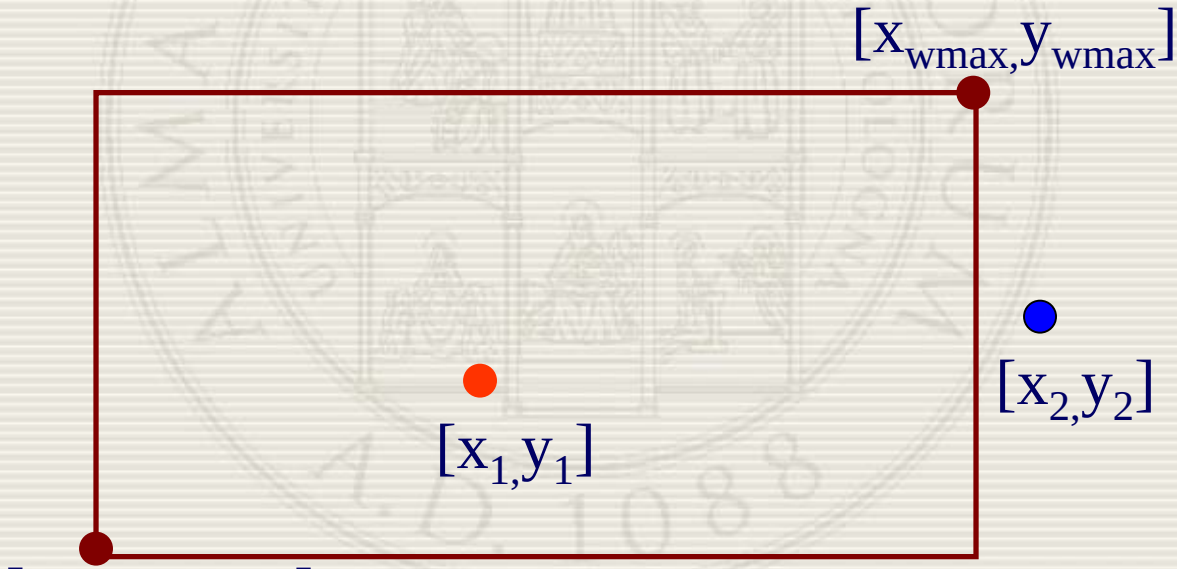
A che serve il clippping?

Serve a non perdere tempo nel disegnare oggetti che sono fuori dalla viewing window 2D (o 3D) e che quindi saranno fuori anche dalla viewport.



Clipping di punti in 2D

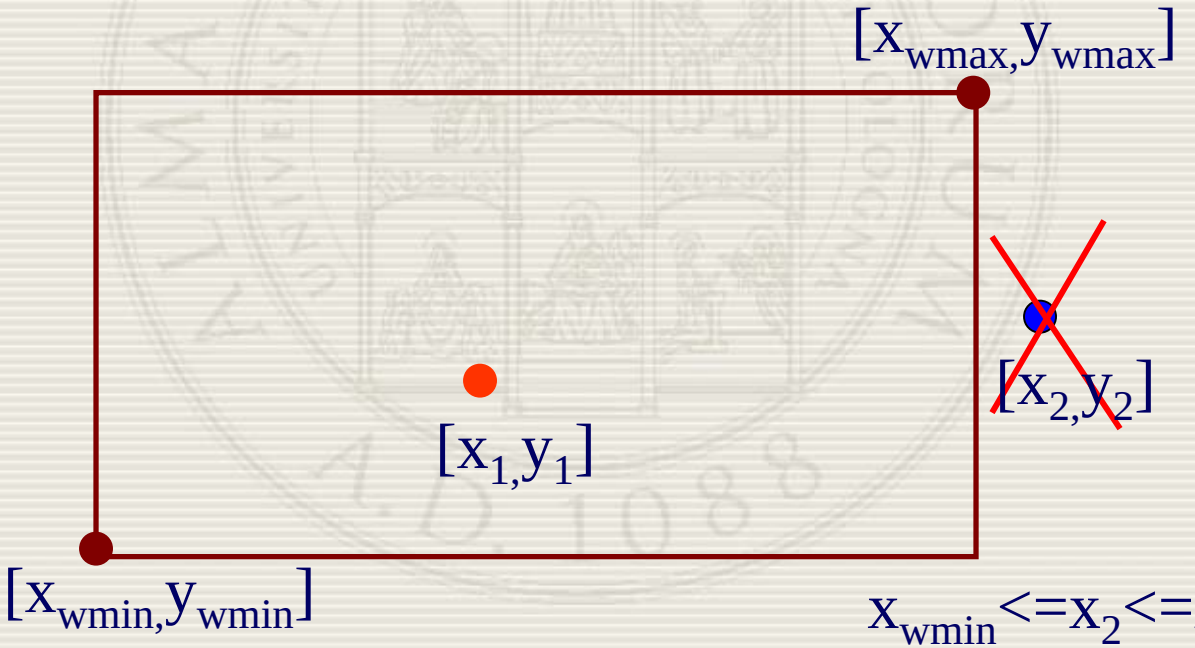
- Dato un punto $[x, y]$ e la window
  $[x_{wmin}, y_{wmin}] \times [x_{wmax}, y_{wmax}]$,
  si determina se il punto deve essere disegnato.



$x_{wmin} <= x_1 <= x_{wmax}$ **Si**
$y_{wmin} <= y_1 <= y_{wmax}$ **Si**

Clipping di punti in 2D

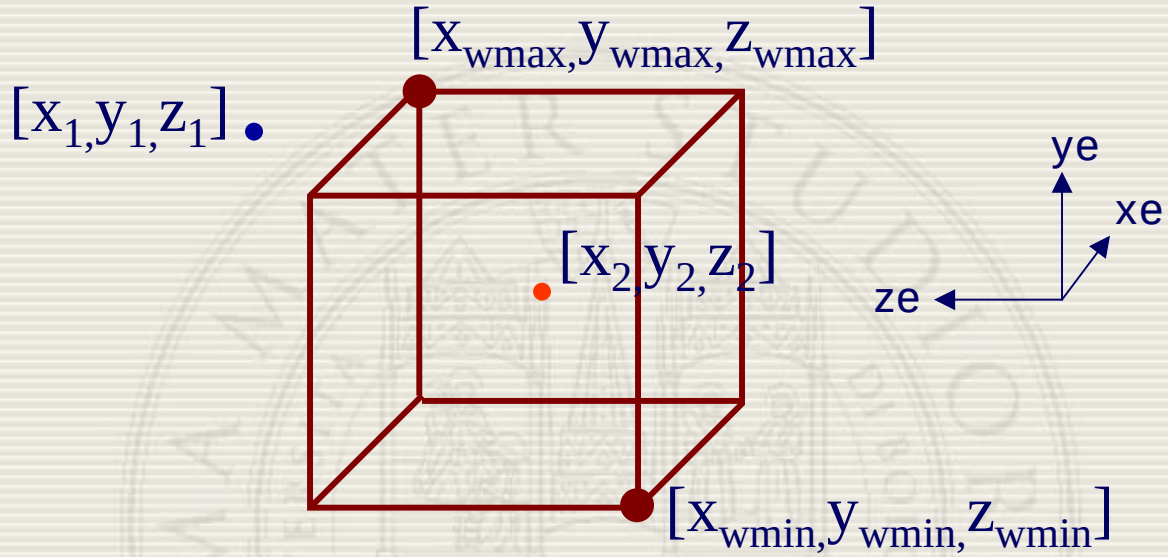
- Dato un punto $[x, y]$ e la window
  $[x_{wmin}, y_{wmin}] \times [x_{wmax}, y_{wmax}]$,
  si determina se il punto deve essere disegnato.



$x_{wmin} <= x_2 <= x_{wmax}$ **No**

$y_{wmin} <= y_2 <= y_{wmax}$ **Si**

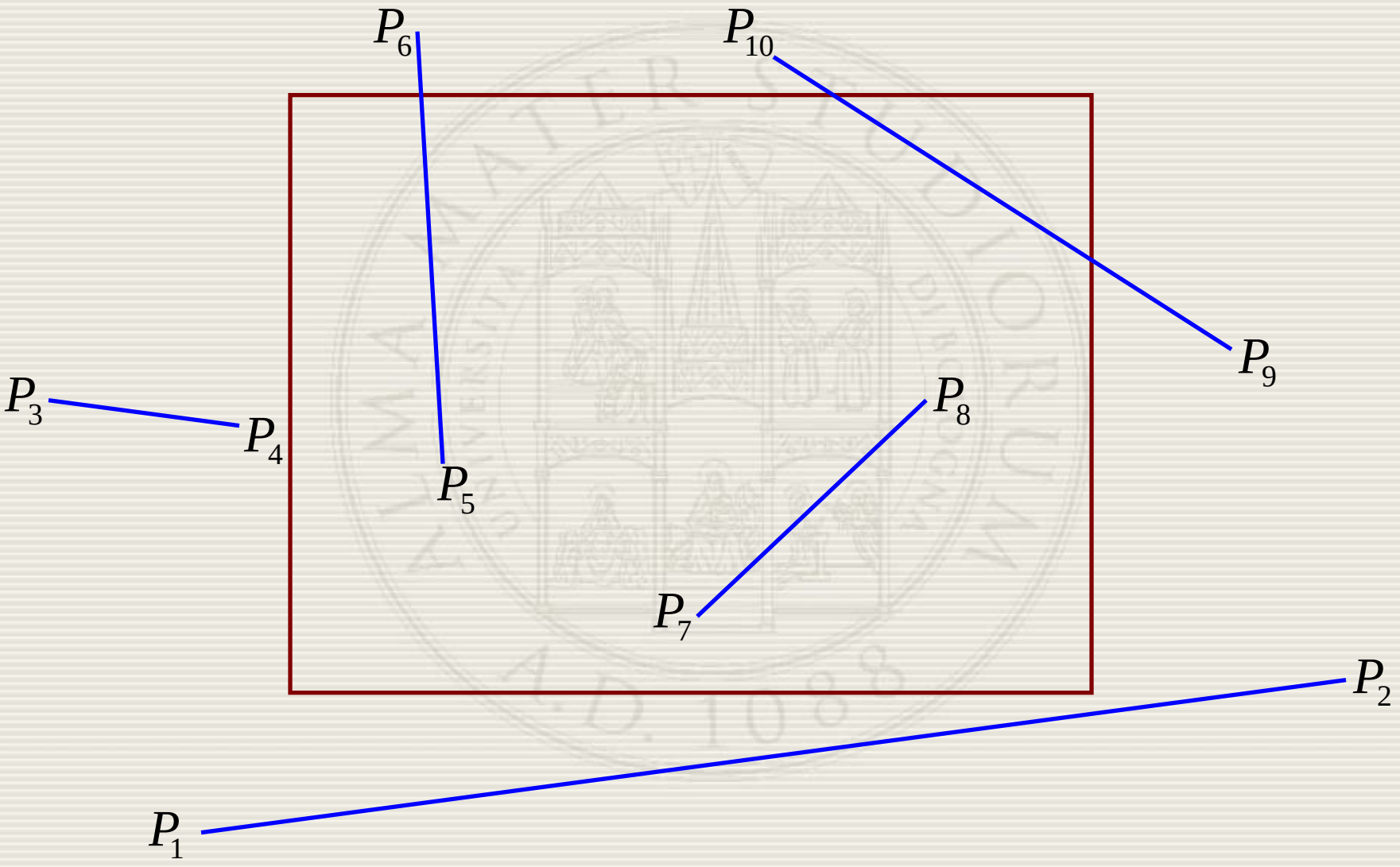
Clipping di punti in 3D



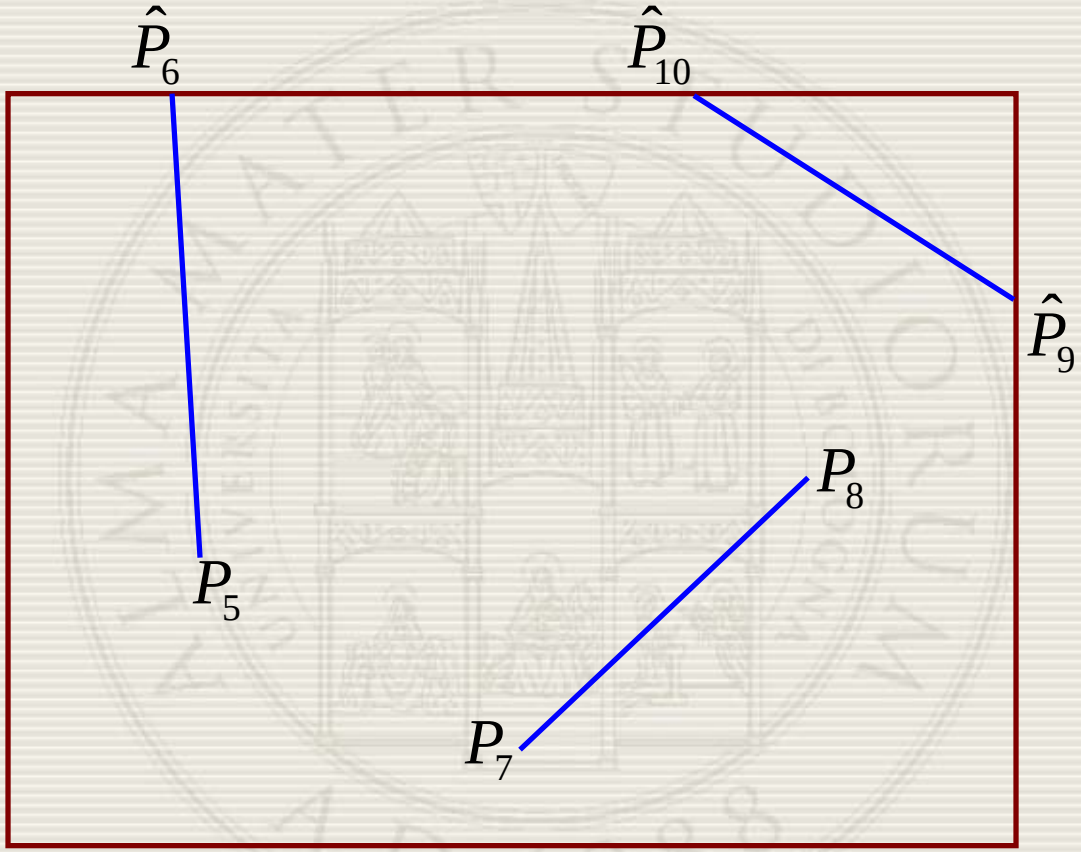
La generalizzazione al 3D è banale; si avranno 3 controlli anziché 2.

$$x_{wmin} <= x_i <= x_{wmax}$$
$$y_{wmin} <= y_i <= y_{wmax}$$
$$z_{wmin} <= z_i <= z_{wmax}$$

Clipping di linee in 2D

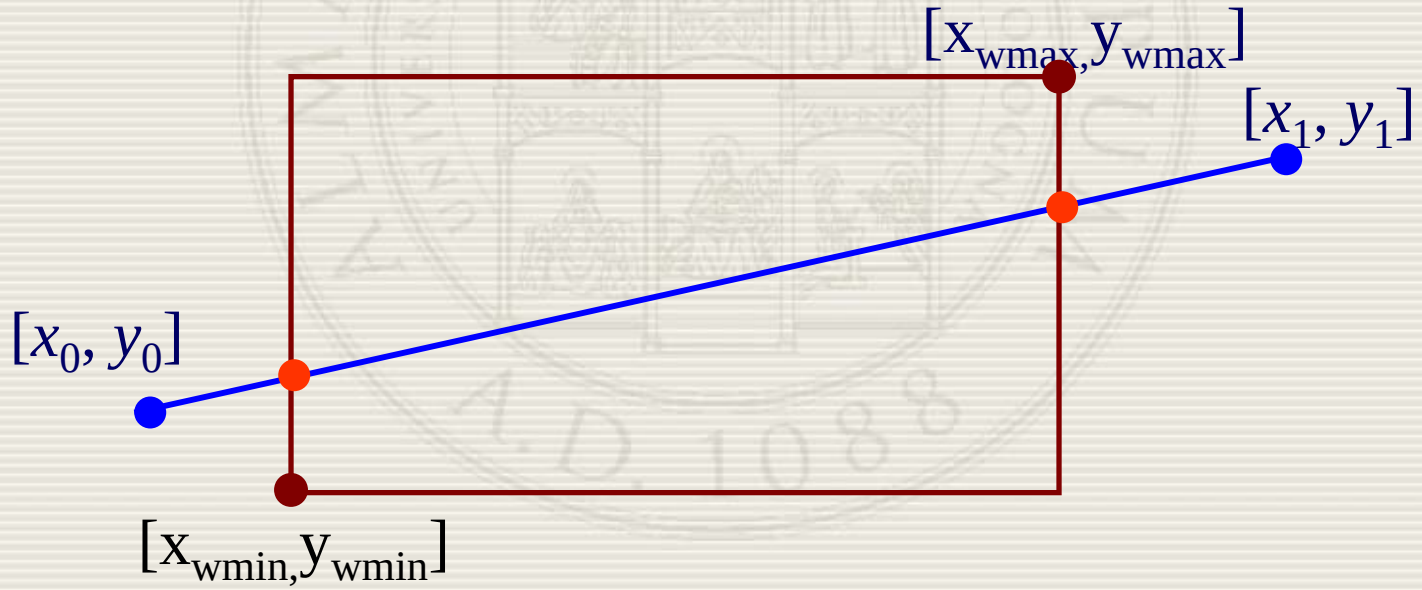


Clipping di linee in 2D



Clipping di linee in 2D

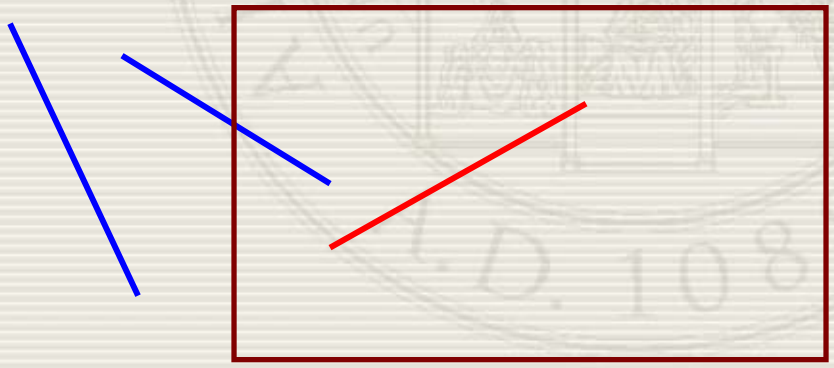
Data un linea di estremi $[x_0, y_0]$, $[x_1, y_1]$ e la window $[x_{wmin}, y_{wmin}]$ x $[x_{wmax}, y_{wmax}]$, si determina se deve essere disegnata o quale parte deve essere disegnata.



Banalmente accettate

- Ottimizzazione: accetta/scarta banalmente

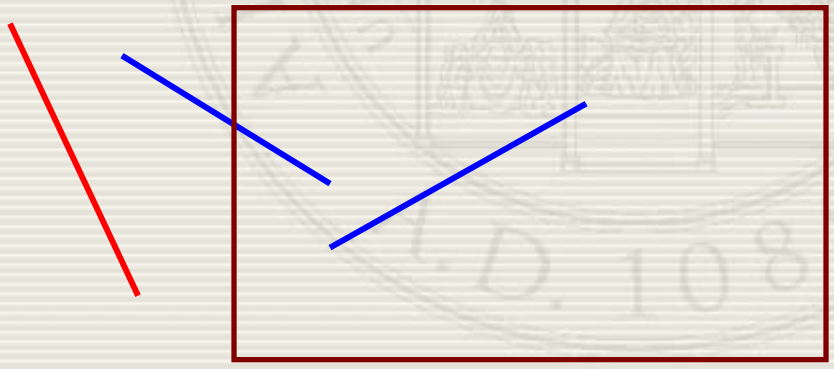
  Come si può decidere velocemente se la linea è completamente interna alla window?
- Risposta: testiamo entrambe gli estremi



Banalmente scartate

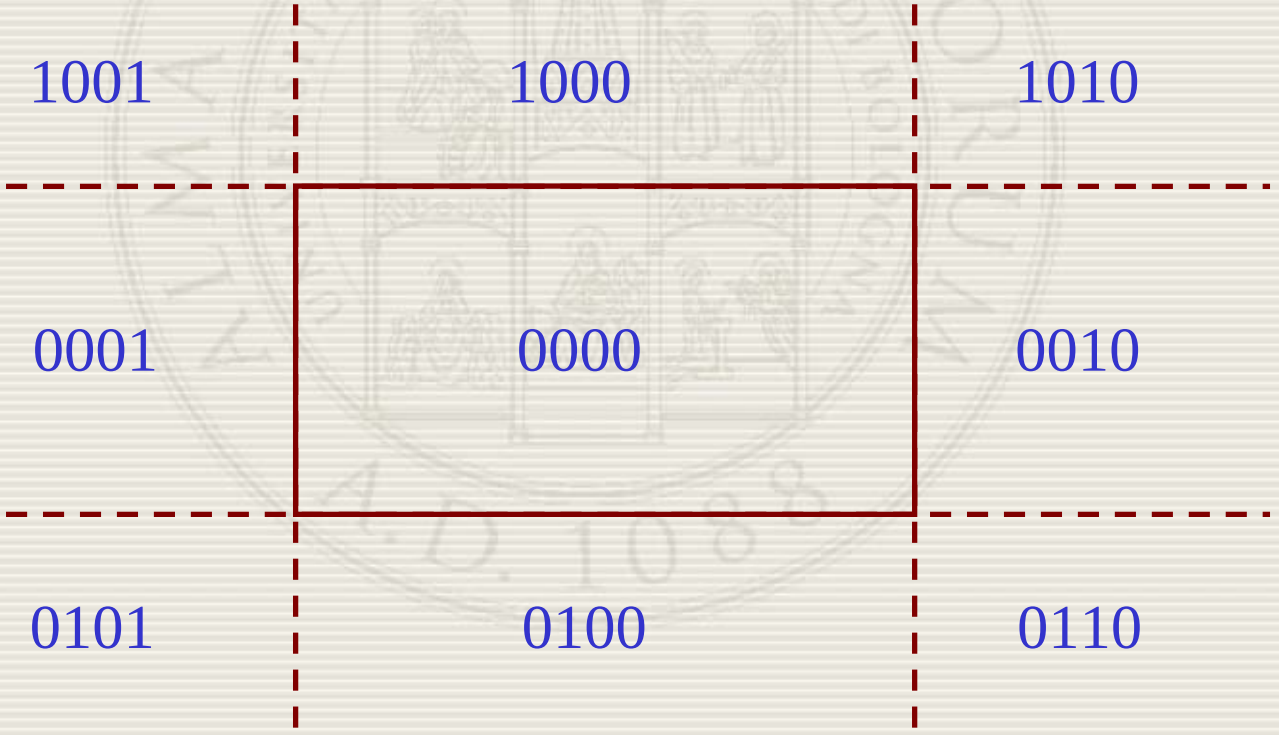
Come facciamo a sapere se una linea è completamente esterna alla window?

- Risposta: se gli estremi sono dalla stessa parte rispetto ad un lato della window, allora la linea può essere scartata.



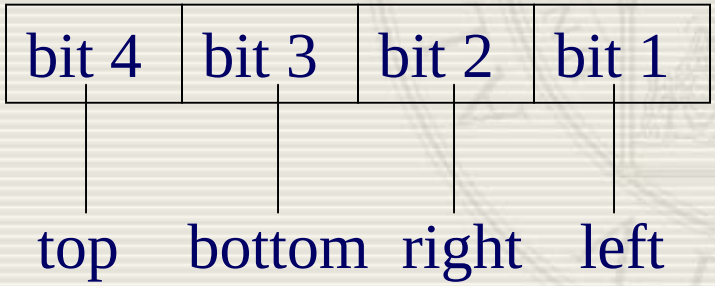
Algoritmo di Cohen-Sutherland

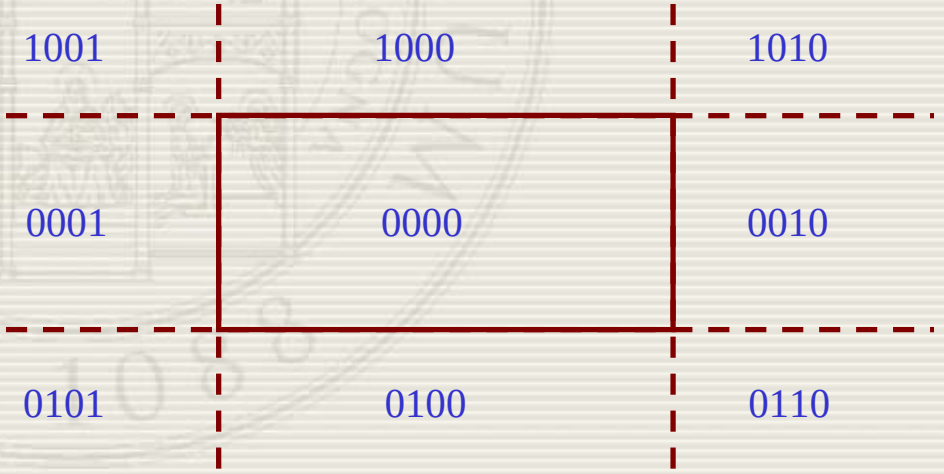
- Prolunghiamo i lati della window in modo da dividere il piano in nove zone.
- Diamo ad ogni zona un codice.



Algoritmo di Cohen-Sutherland

- Ad ogni punto del piano può essere associato un codice identificativo della zona in cui è; si tratta di un codice a 4 bit.
- Ogni bit del codice indica se il punto è interno o esterno ad uno specifico lato della window



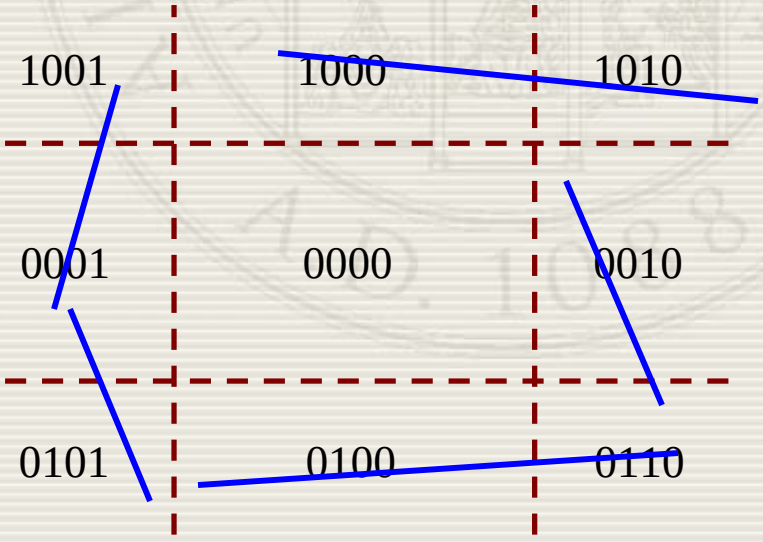


Algoritmo di Cohen-Sutherland

- Classifichiamo p_0 e p_1 usando i codici c_0 e c_1
- Se $c_0 \wedge c_1 \neq 0$, banalmente scartata
- Se $c_0 \vee c_1 = 0$, banalmente accettata
- Altrimenti cerchiamo di ridurci a casi banali mediante la suddivisione in due segmenti.

Algoritmo di Cohen-Sutherland

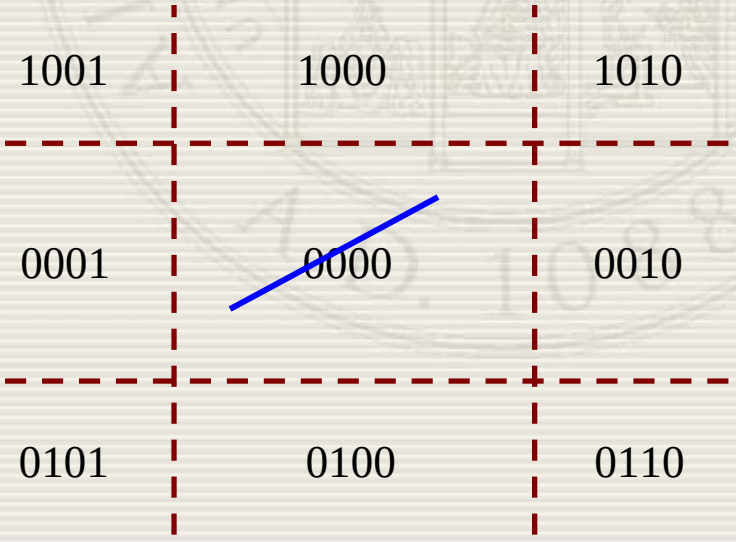
- Classifichiamo p_0 e p_1 usando i codici c_0 e c_1
- Se $c_0 \wedge c_1 \neq 0$, banalmente scartata
- Se $c_0 \vee c_1 = 0$, banalmente accettata
- Altrimenti cerchiamo di ridurci a casi banali mediante la suddivisione in due segmenti.



La linea è esterna alla window! Scartata

Algoritmo di Cohen-Sutherland

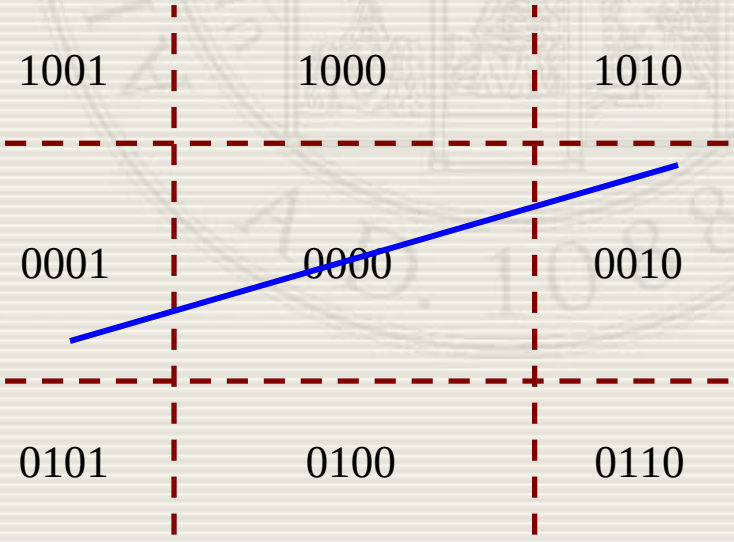
- Classifichiamo p_0 e p_1 usando i codici c_0 e c_1
- Se $c_0 \wedge c_1 \neq 0$, banalmente scartata
- Se $c_0 \vee c_1 = 0$, banalmente accettata
- Altrimenti cerchiamo di ridurci a casi banali mediante la suddivisione in due segmenti



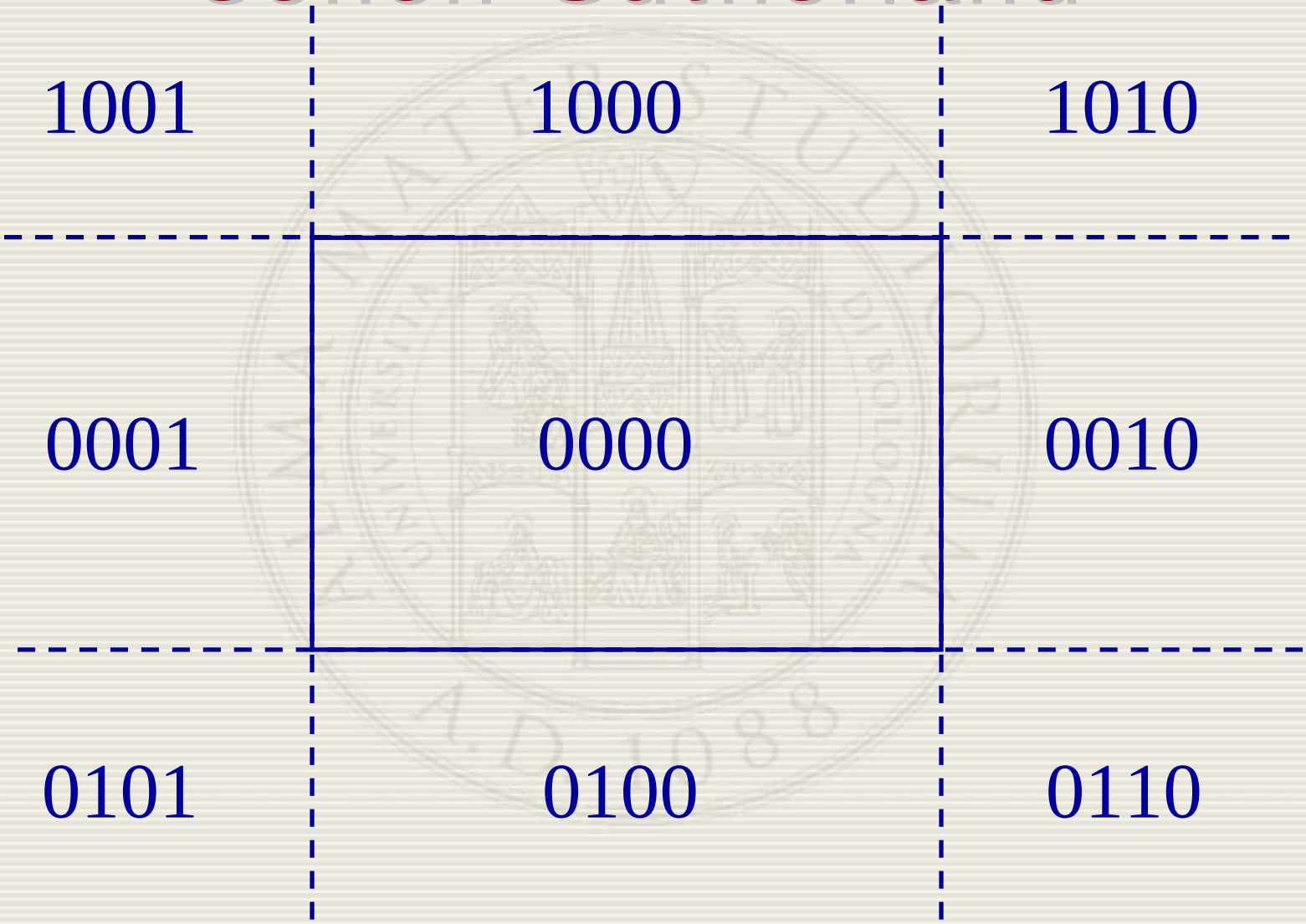
La linea è interna alla window! Accettata

Algoritmo di Cohen-Sutherland

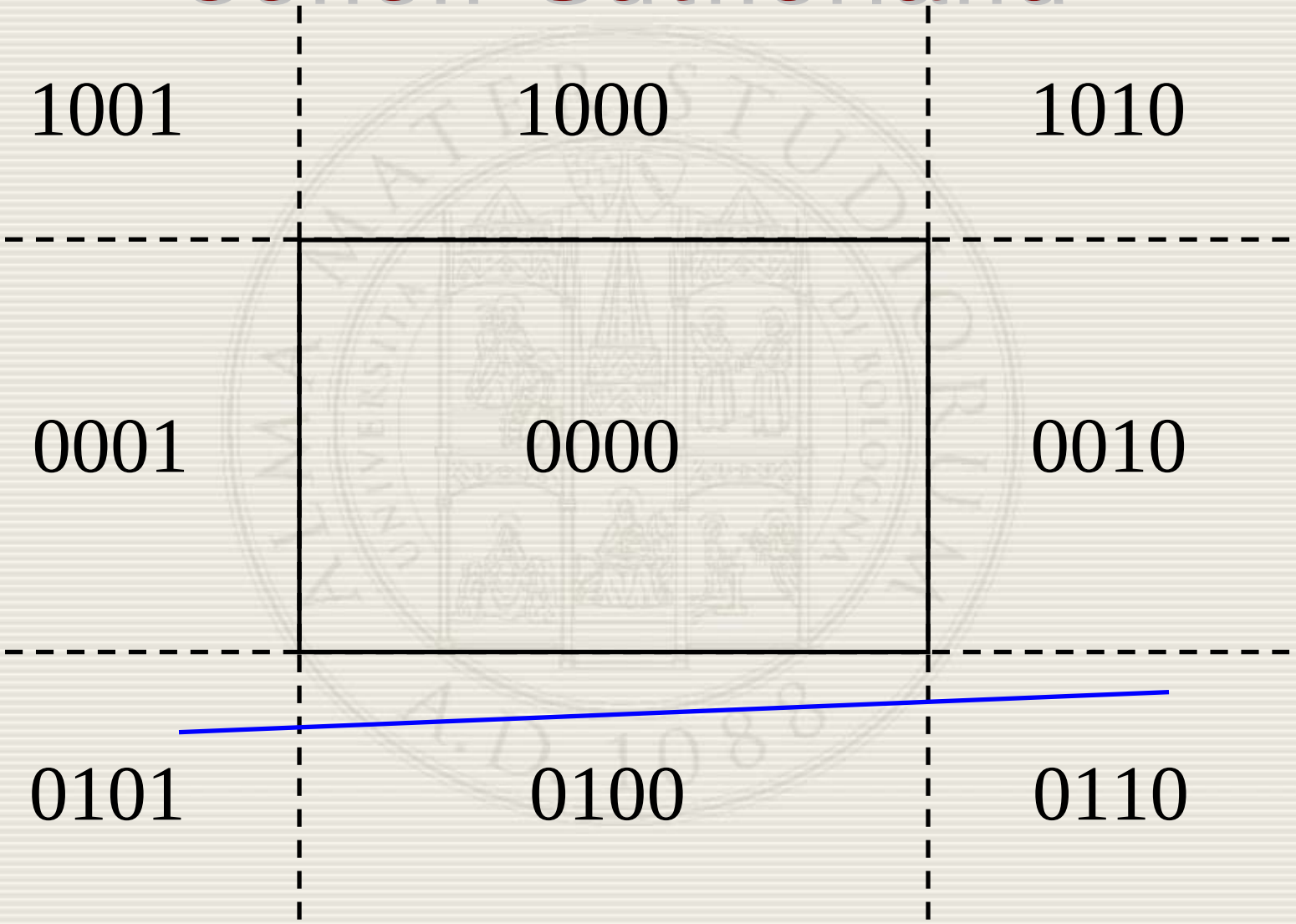
- Classifichiamo p_0 e p_1 usando i codici c_0 e c_1
- Se $c_0 \wedge c_1 \neq 0$, banalmente scartata
- Se $c_0 \vee c_1 = 0$, banalmente accettata
- Altrimenti cerchiamo di ridurci a casi banali mediante la suddivisione in due segmenti



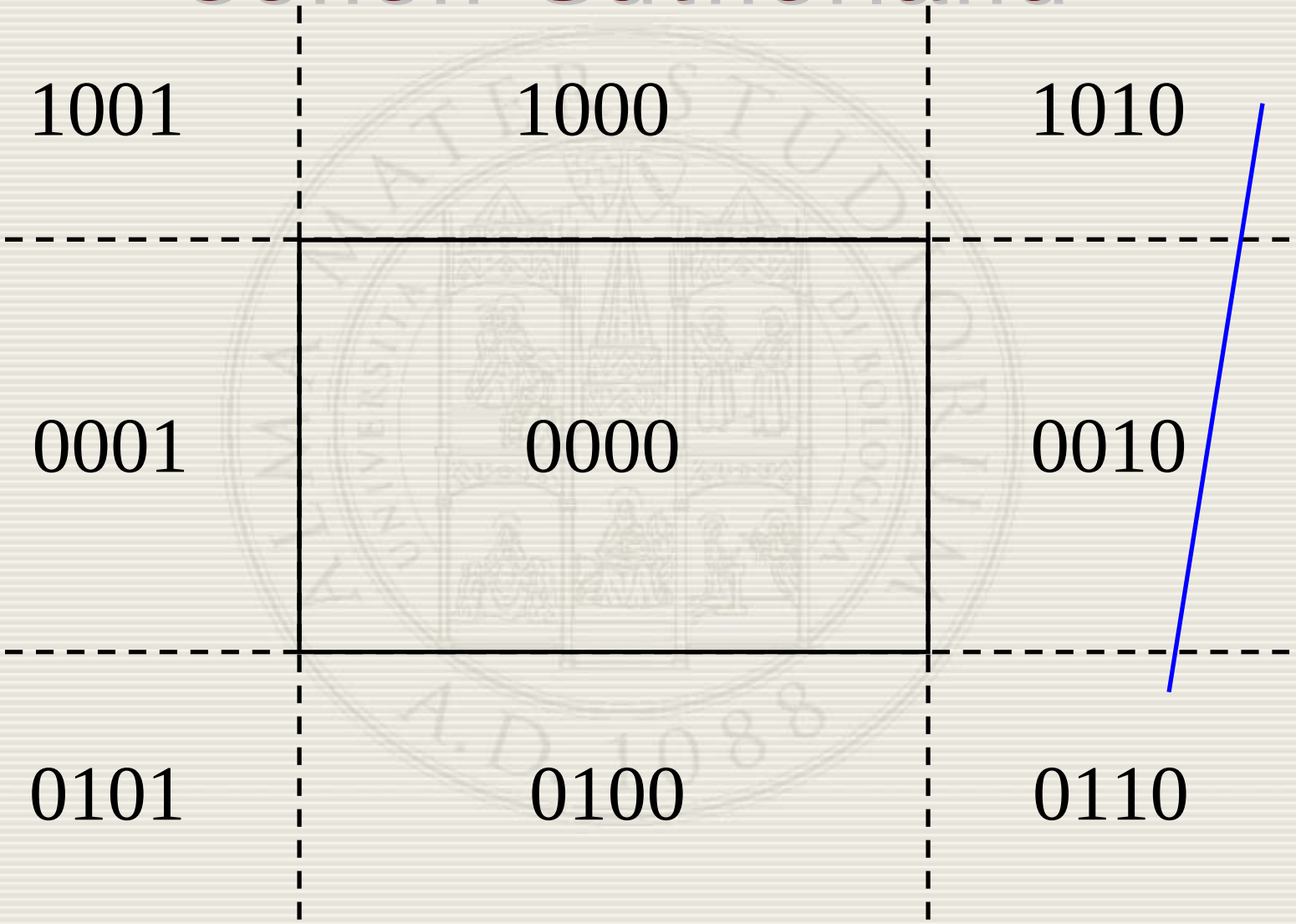
Algoritmo di Cohen-Sutherland



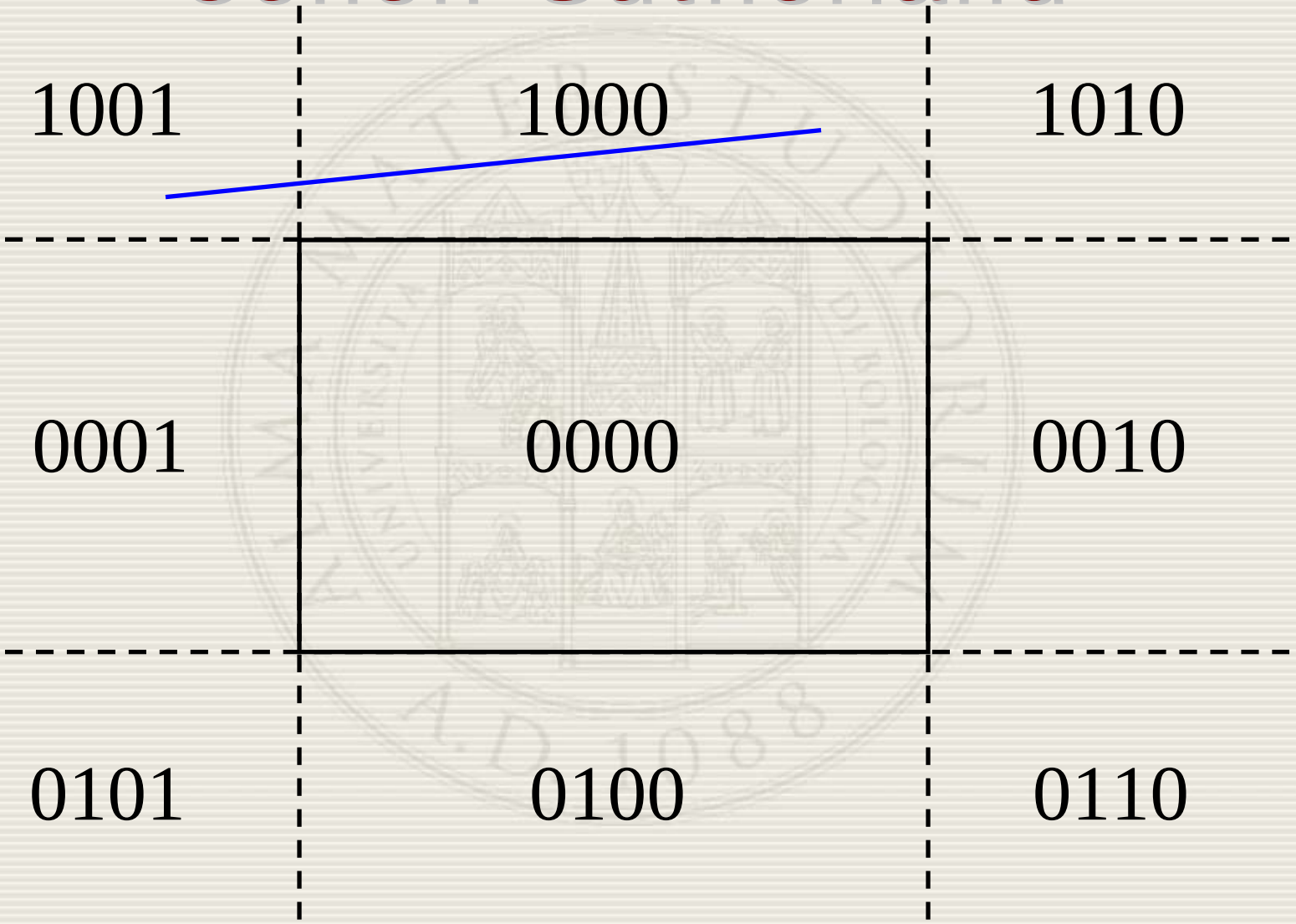
Algoritmo di Cohen-Sutherland



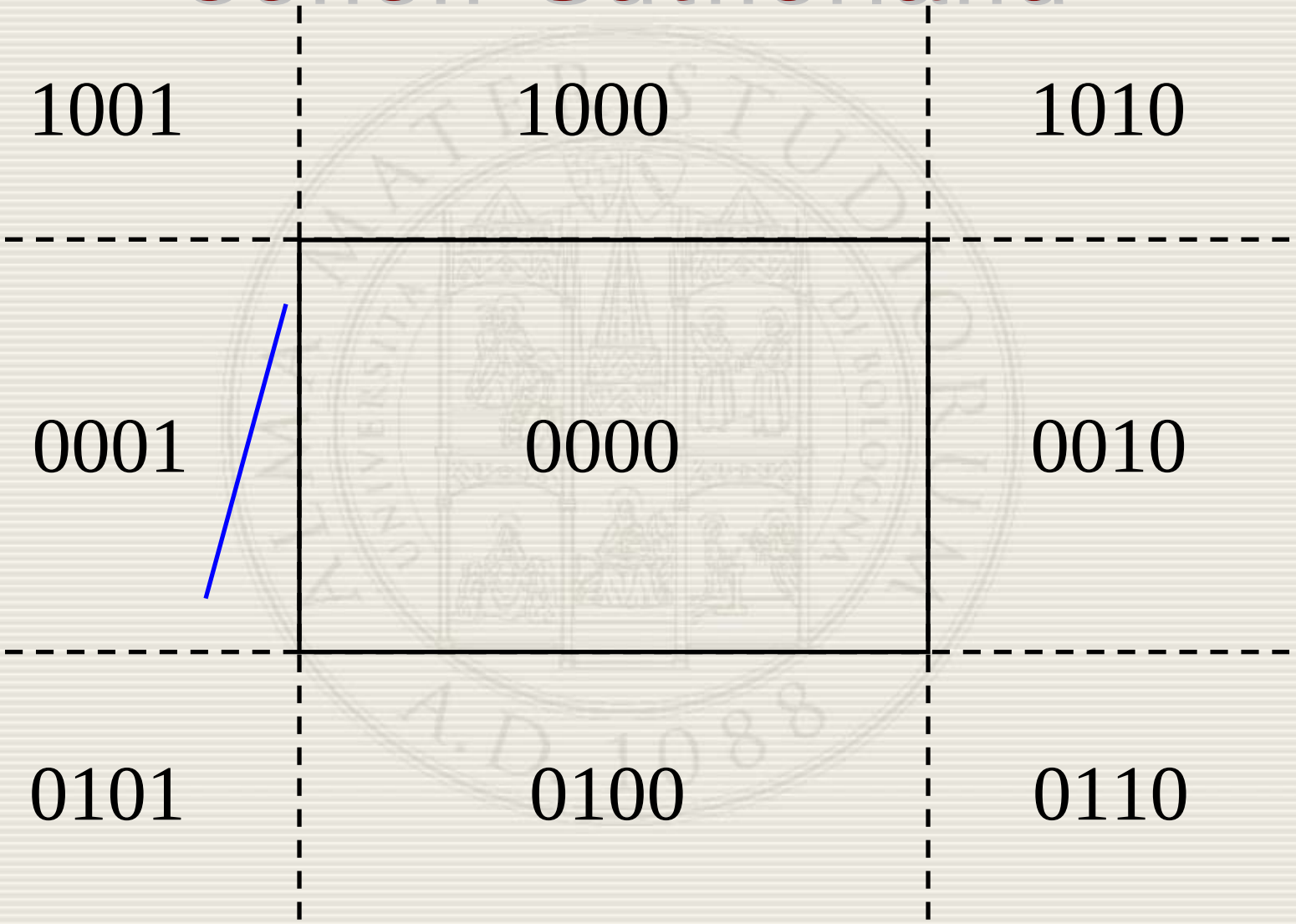
Algoritmo di Cohen-Sutherland



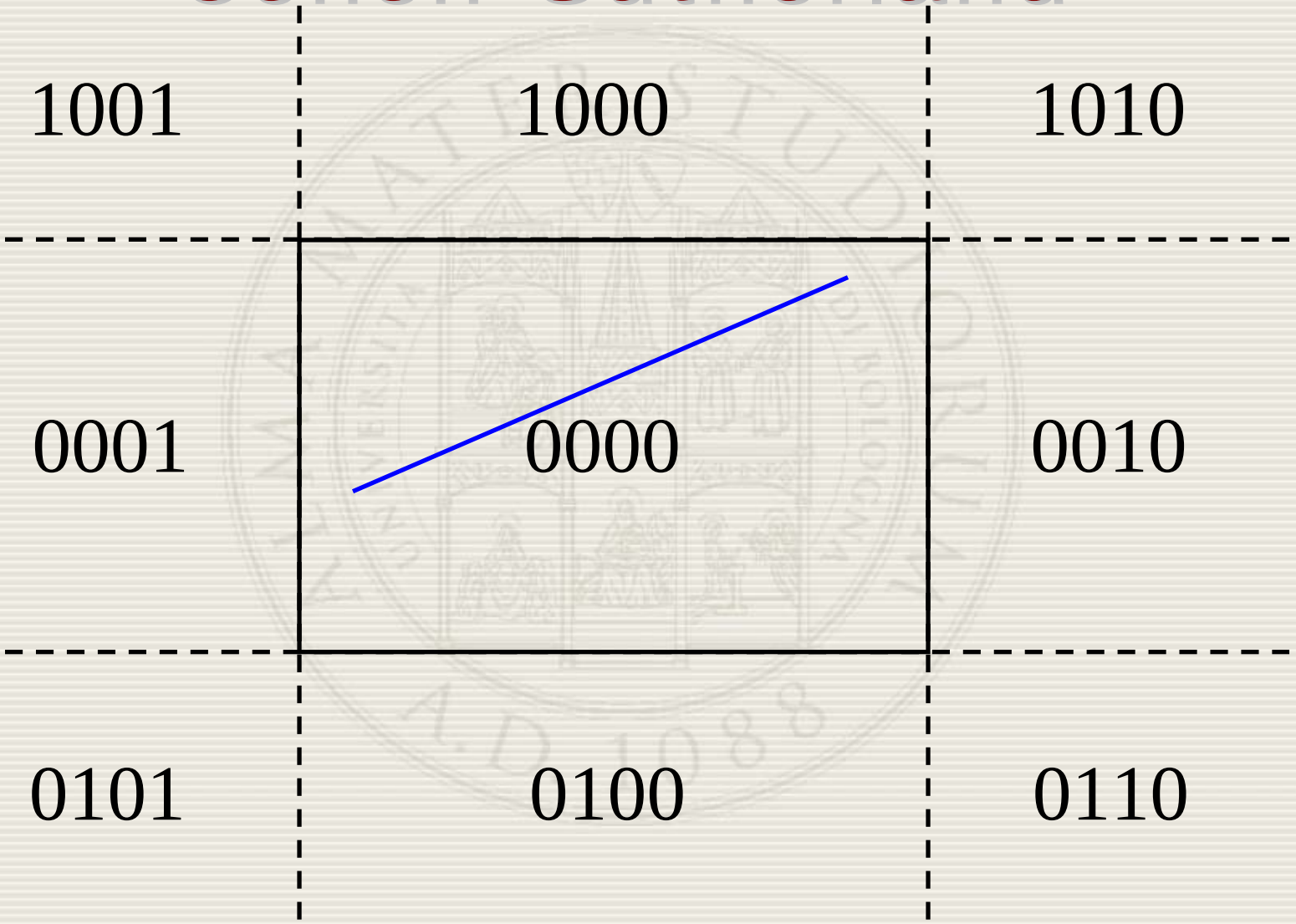
Algoritmo di Cohen-Sutherland



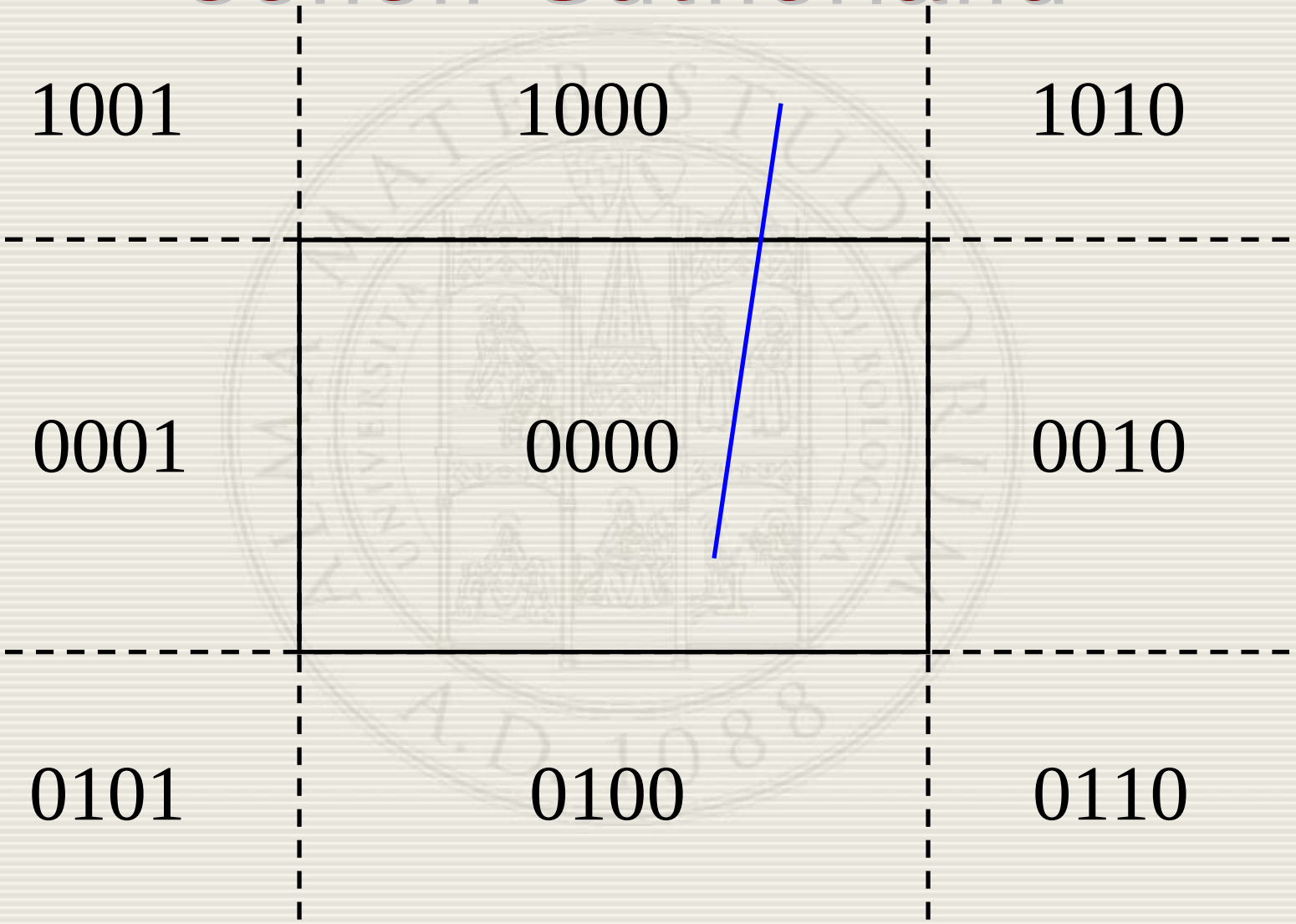
Algoritmo di Cohen-Sutherland



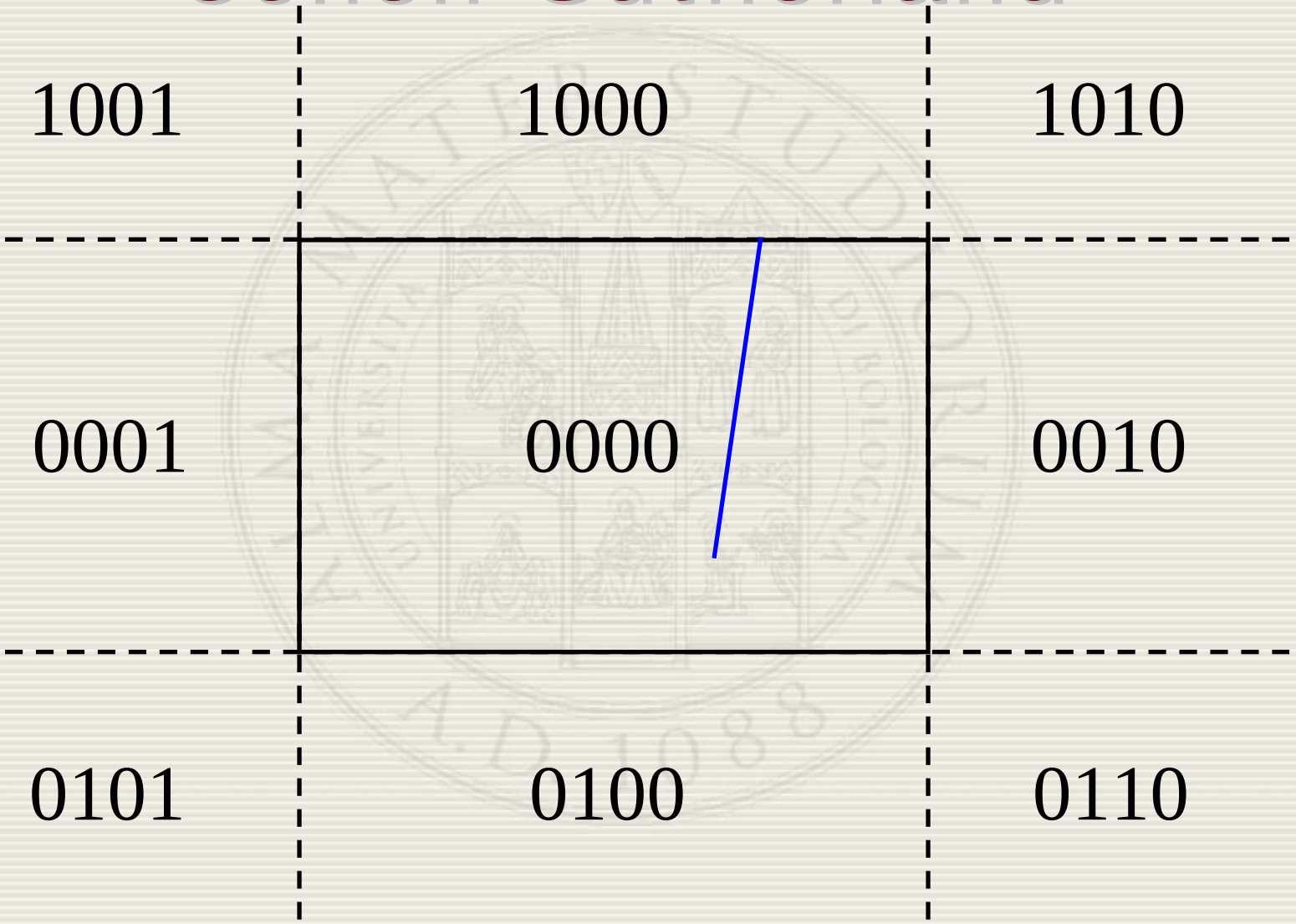
Algoritmo di Cohen-Sutherland



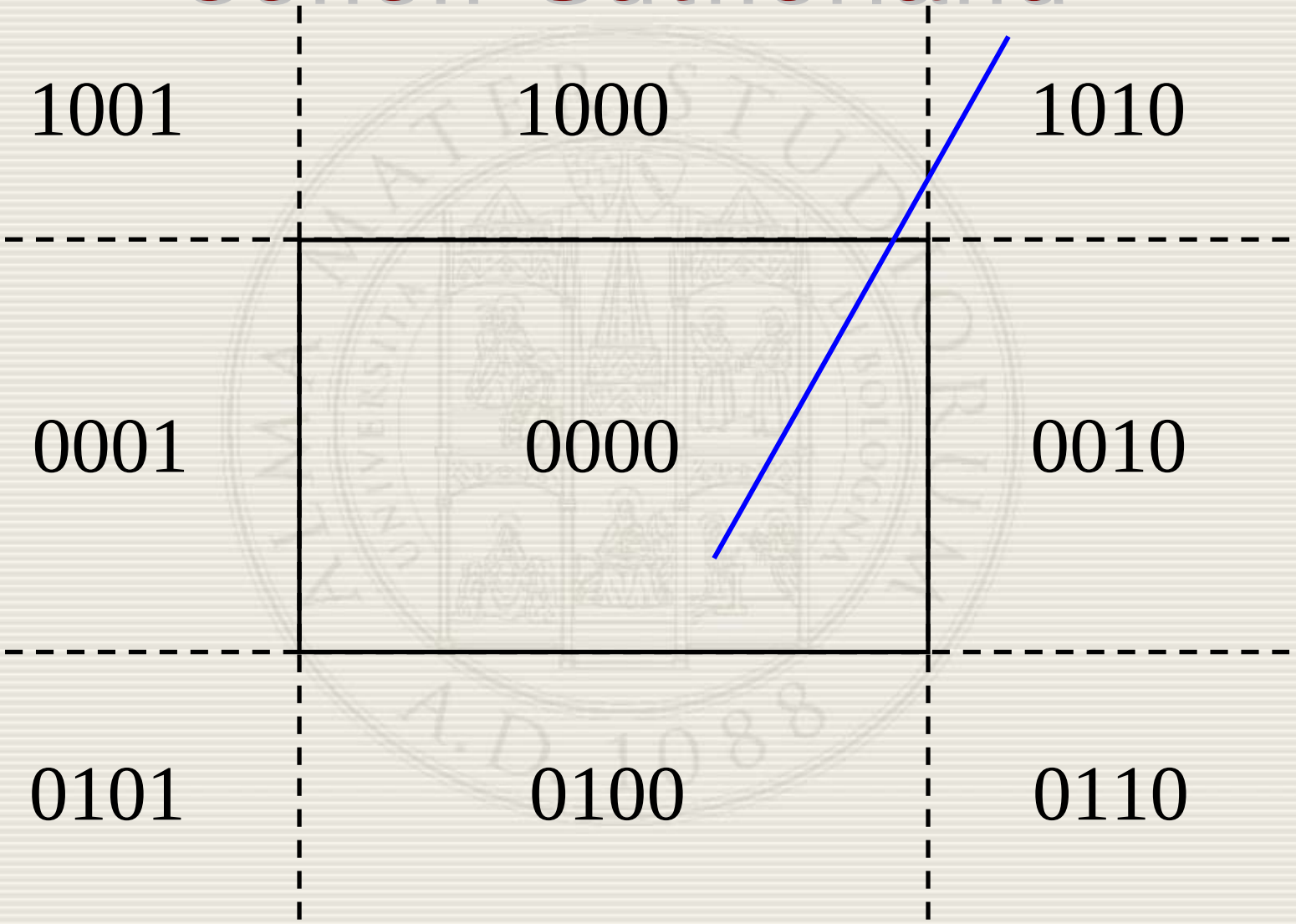
Algoritmo di Cohen-Sutherland



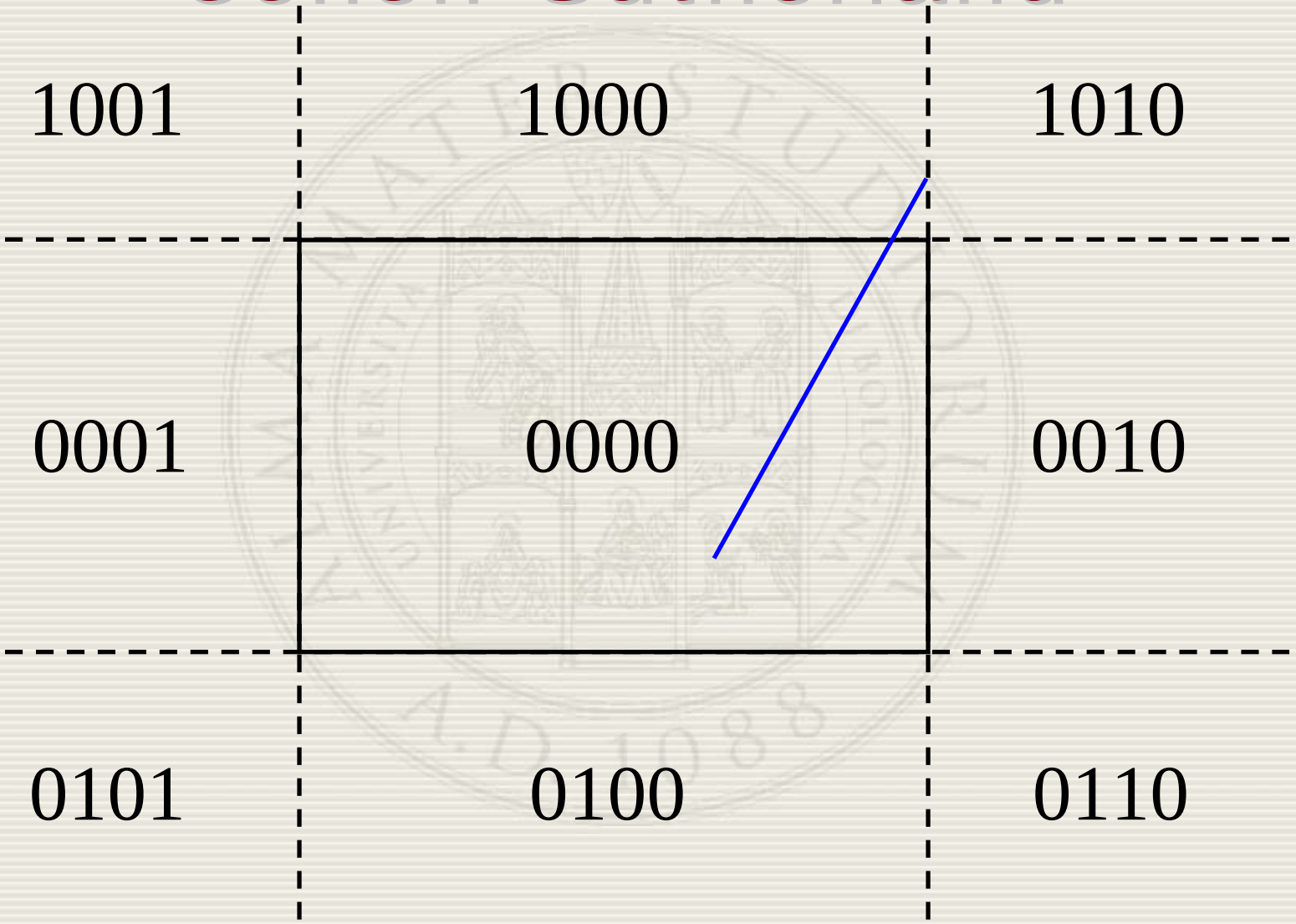
Algoritmo di Cohen-Sutherland



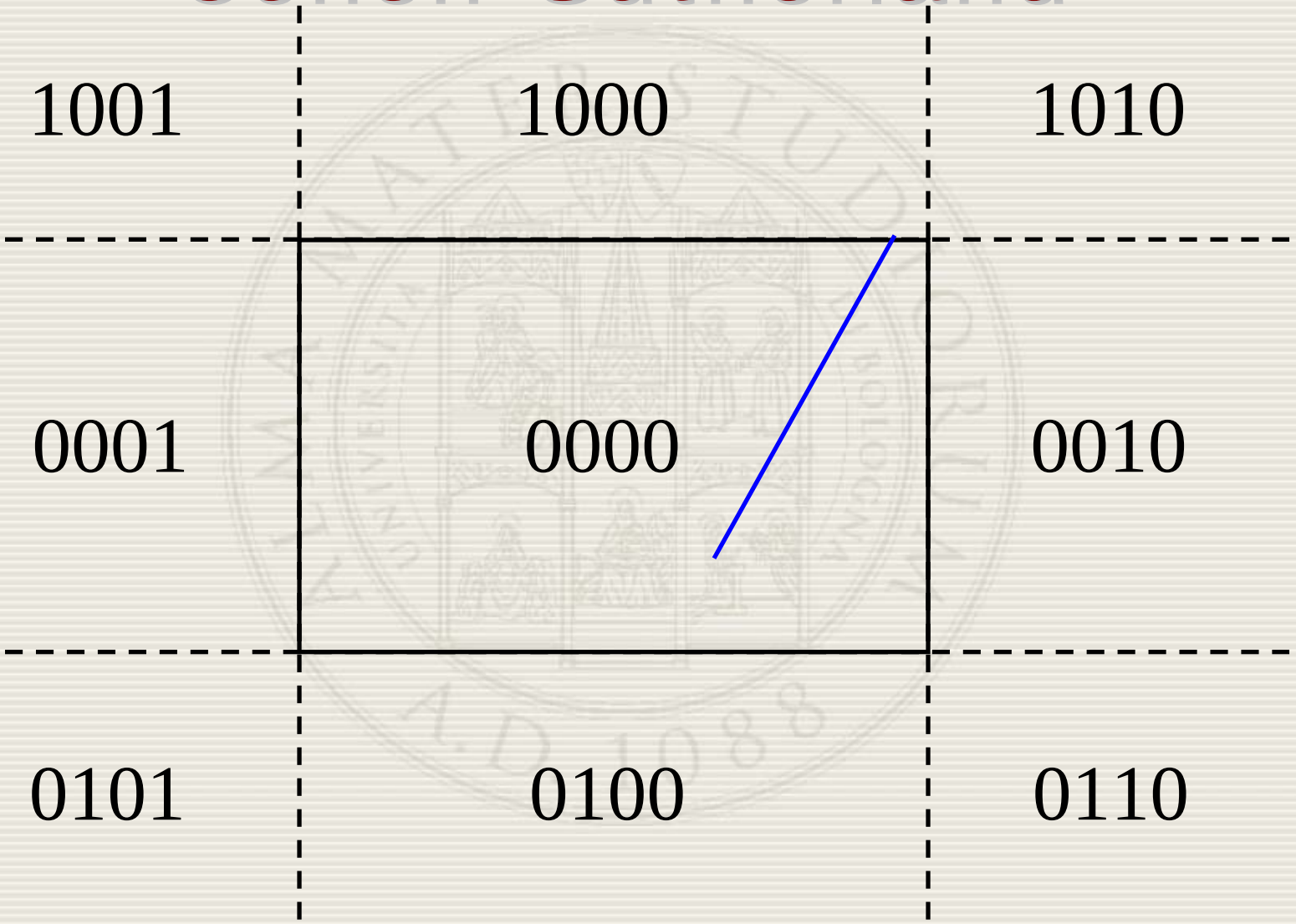
Algoritmo di Cohen-Sutherland



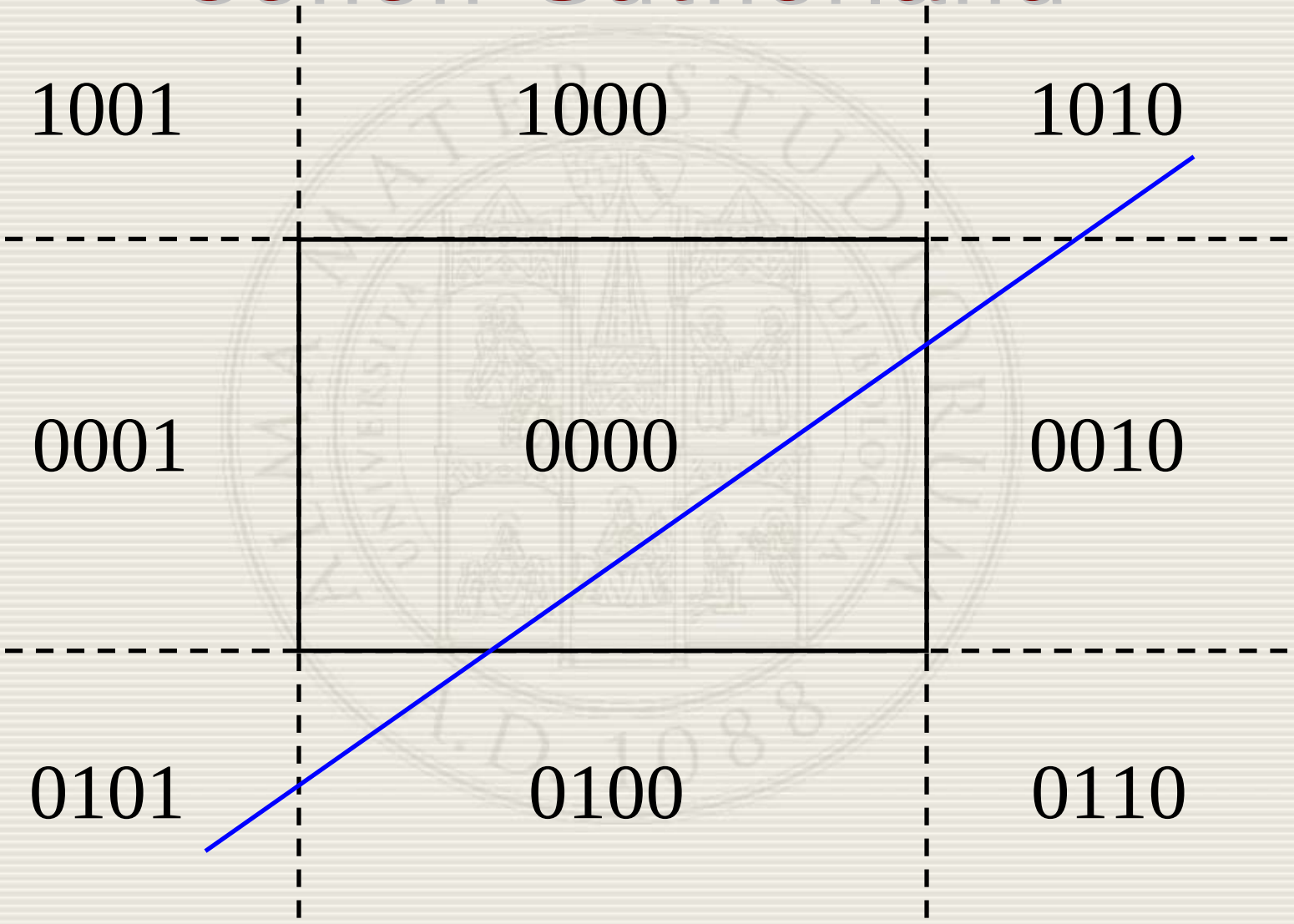
Algoritmo di Cohen-Sutherland



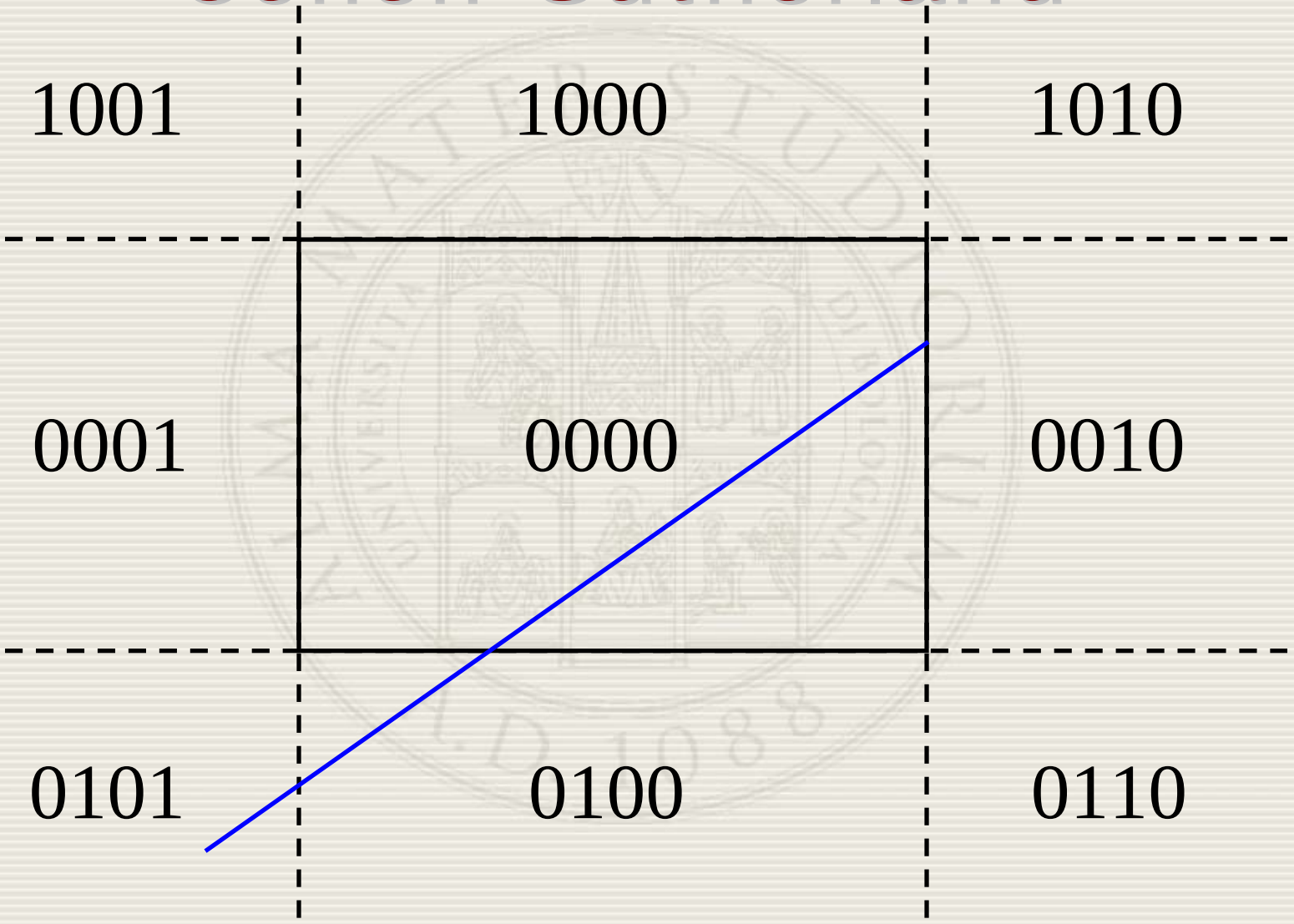
Algoritmo di Cohen-Sutherland



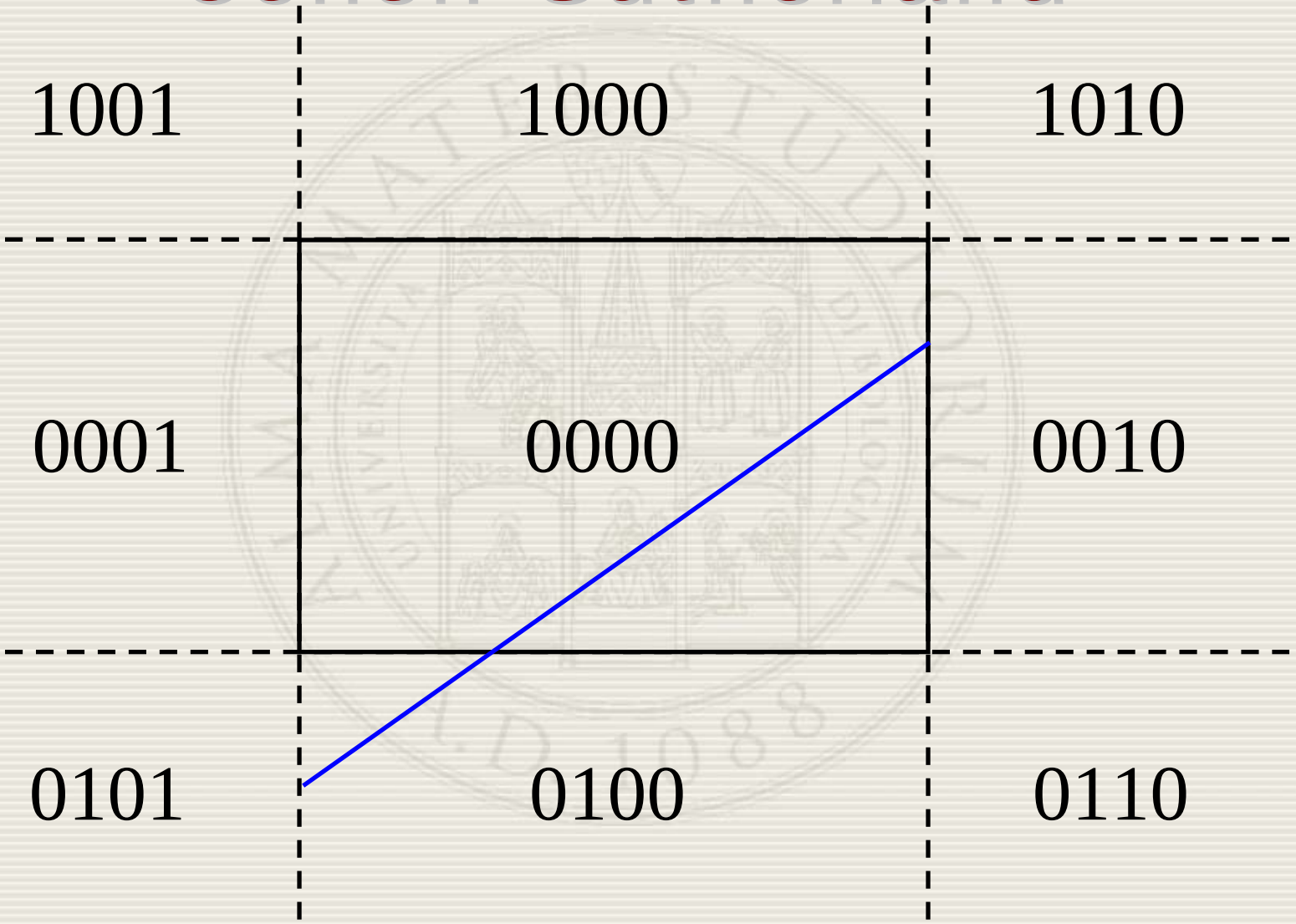
Algoritmo di Cohen-Sutherland



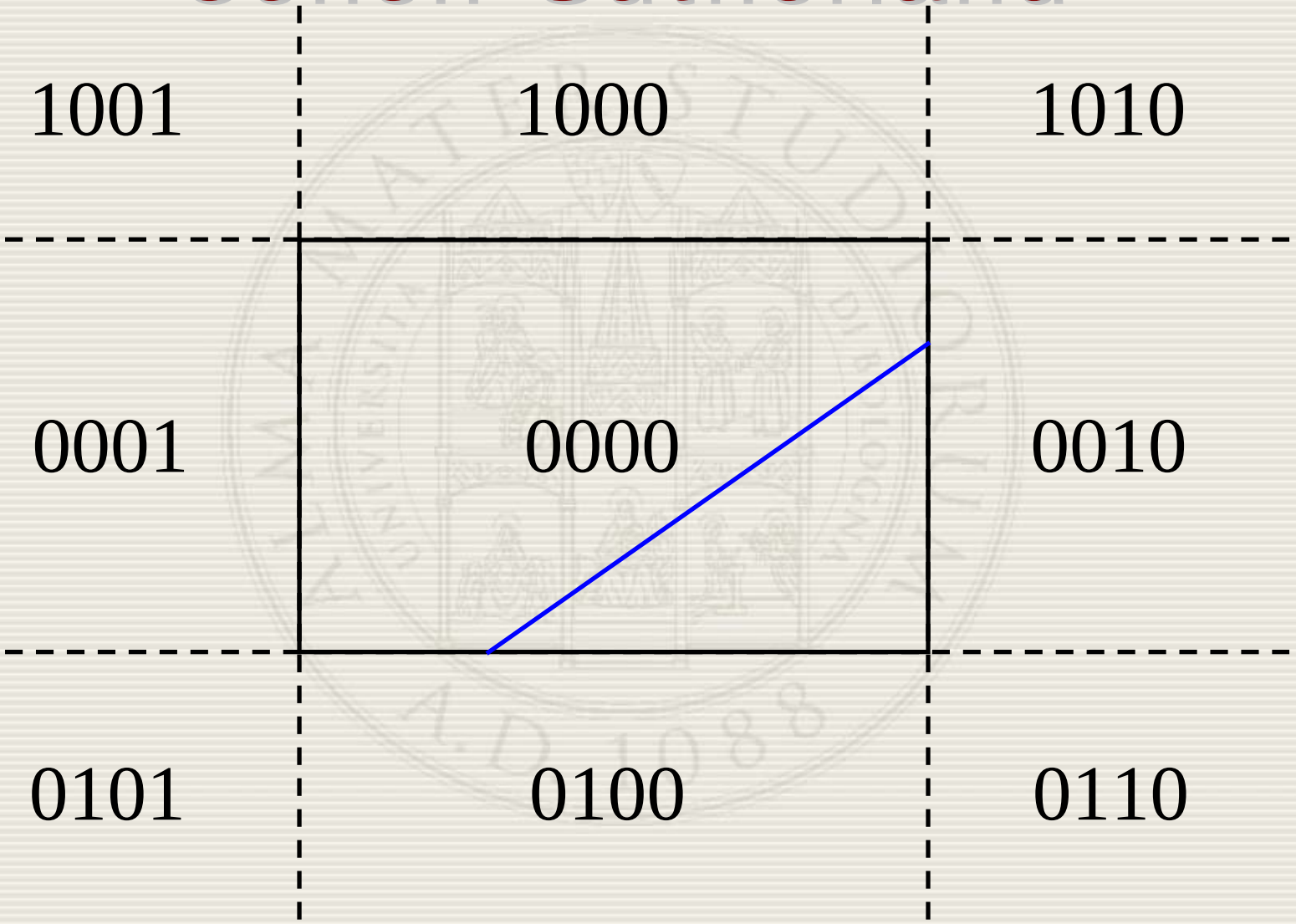
Algoritmo di Cohen-Sutherland



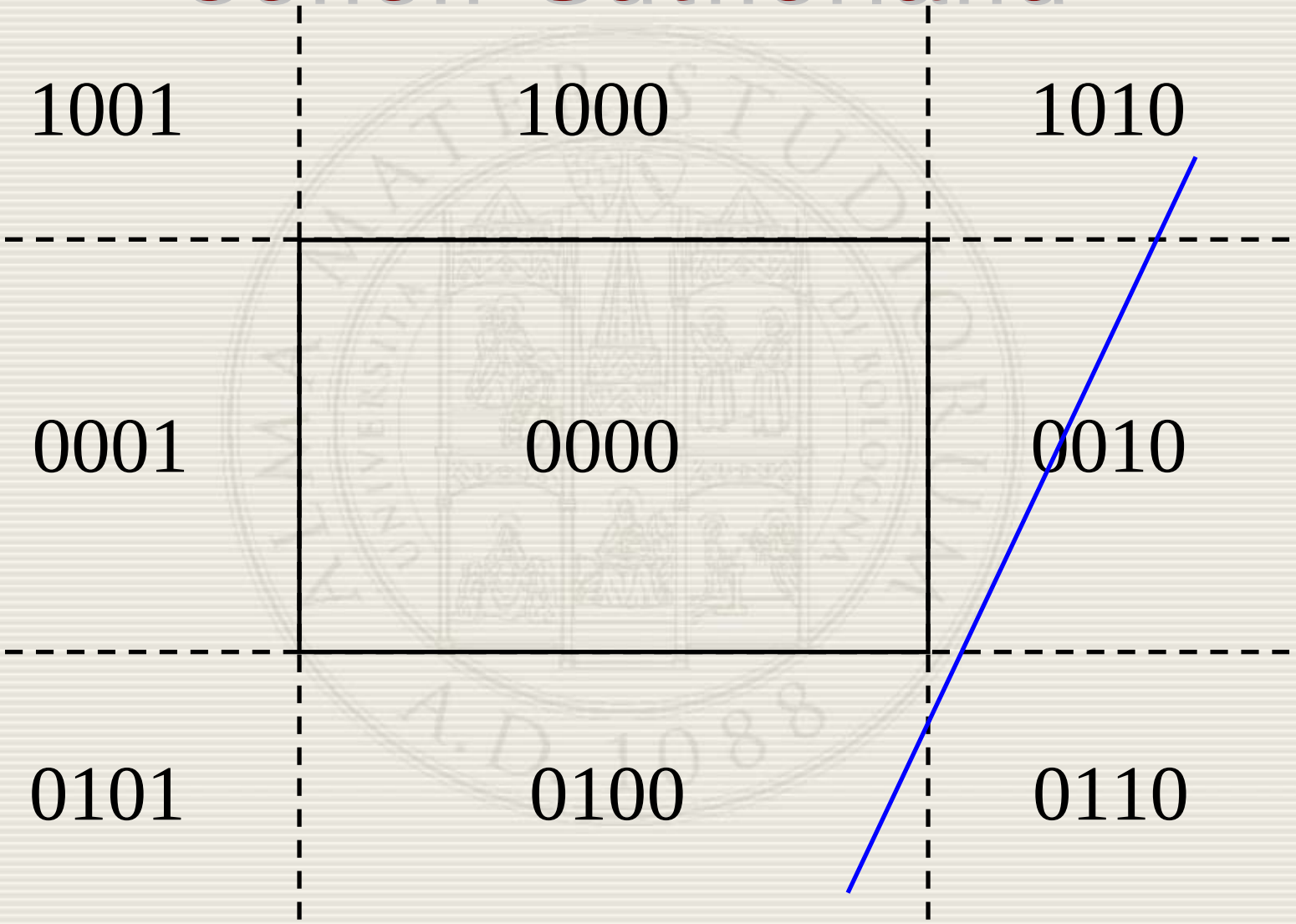
Algoritmo di Cohen-Sutherland



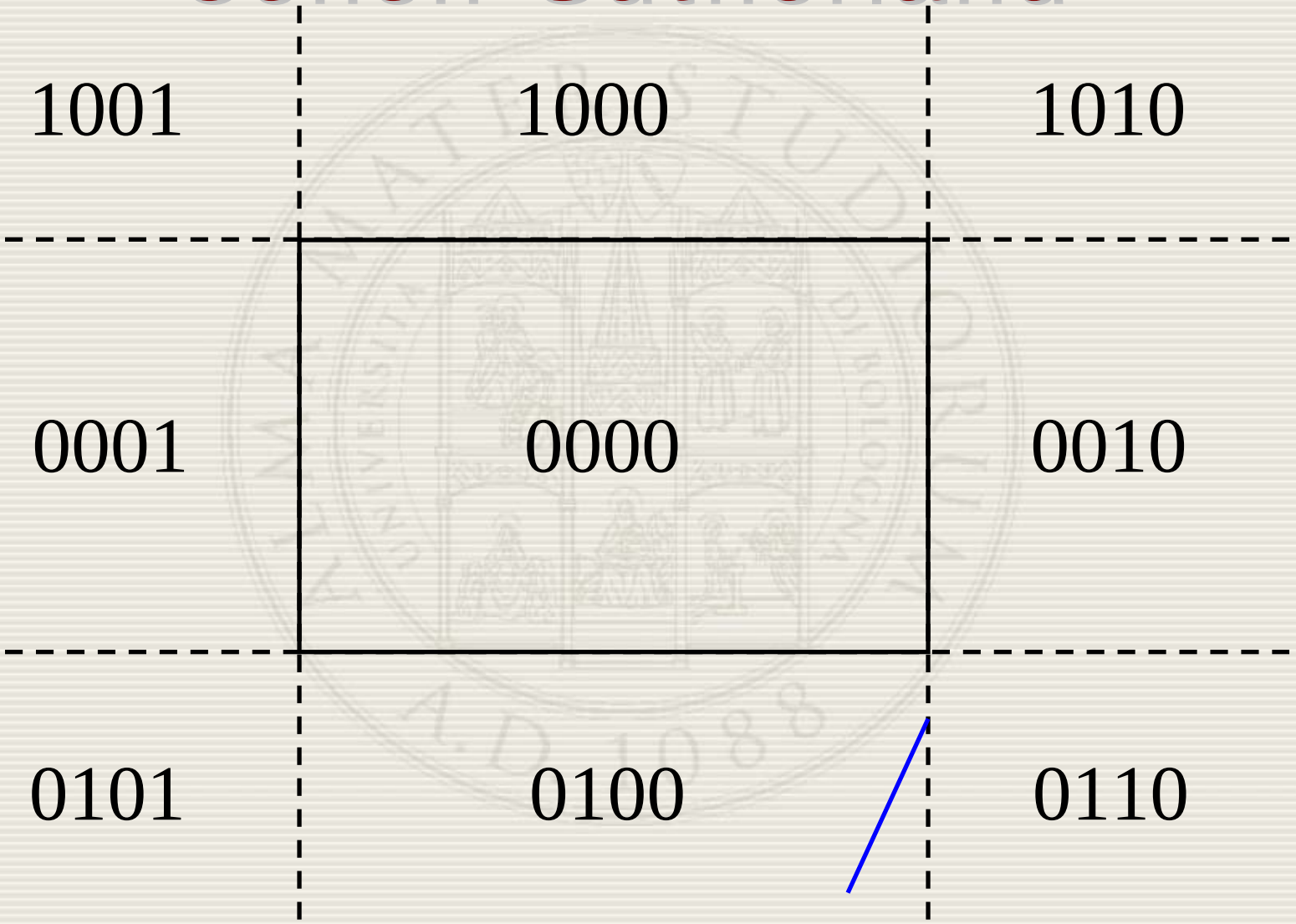
Algoritmo di Cohen-Sutherland



Algoritmo di Cohen-Sutherland



Algoritmo di Cohen-Sutherland

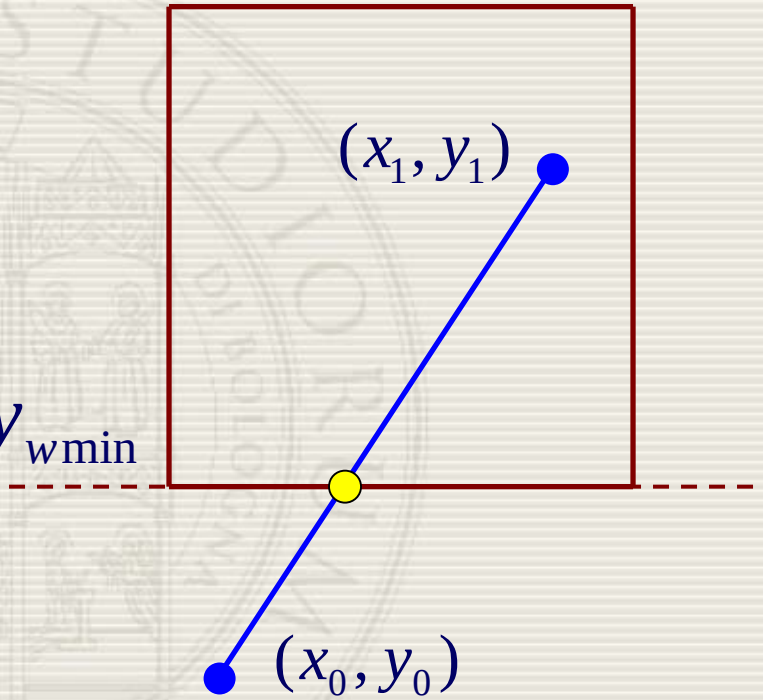


Intersezione fra Linee in forma Parametrica e lati Window

$$\begin{cases} x(t) = x_0 + (x_1 - x_0)t \\ y(t) = y_0 + (y_1 - y_0)t \end{cases}$$

$$0 \le t \le 1$$

$$y = y_{w\min}$$

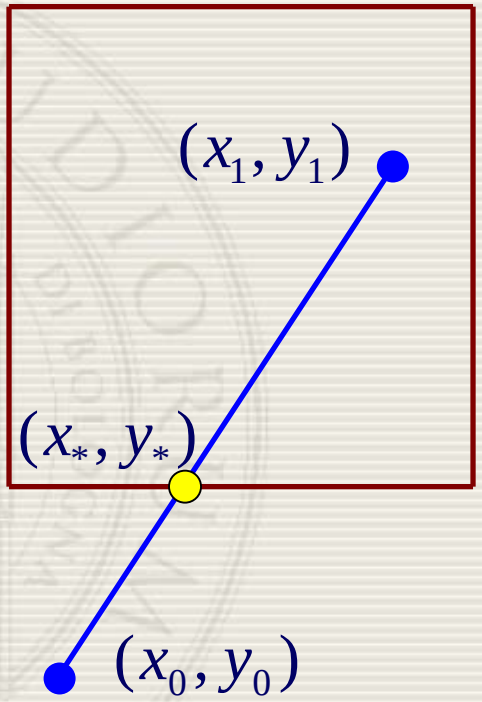


$$\begin{cases} y(t) = y_0 + (y_1 - y_0)t \\ y = y_{w\min} \end{cases}$$

Intersezione fra Linee in forma Parametrica e lati Window

$$\begin{cases} y(t) = y_0 + (y_1 - y_0)t \\ y = y_{w\min} \end{cases}$$

$$t = \frac{y_{w\min} - y_0}{y_1 - y_0}$$



$$x_* = x_0 + (x_1 - x_0)\frac{y_{w\min} - y_0}{y_1 - y_0}$$

$$y_* = y_{w\min}$$

Esempio

Archvio SDL2prg_1516.tgz, cartella SDL2prg1, codice:

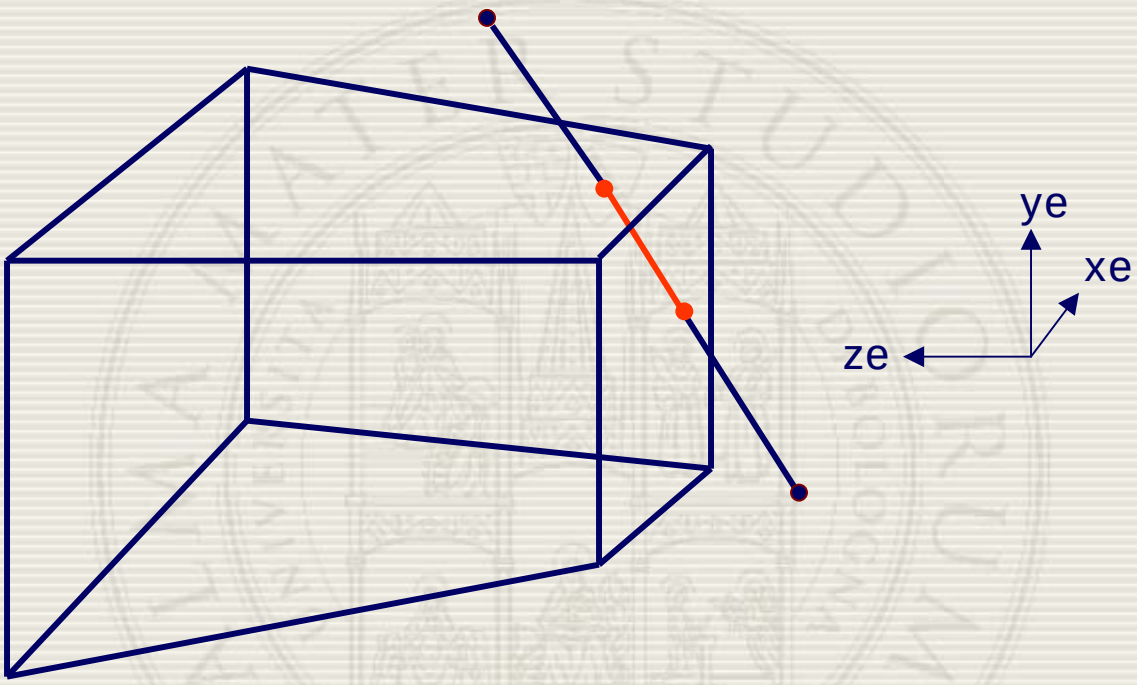
line_clip.c

Conclusioni

- Algoritmo di Cohen-Sutherland
  - Clipping ripetuti possono essere costosi
  - Le migliori prestazioni si hanno quando la maggior parte delle linee possono essere accettate o scartate banalmente.

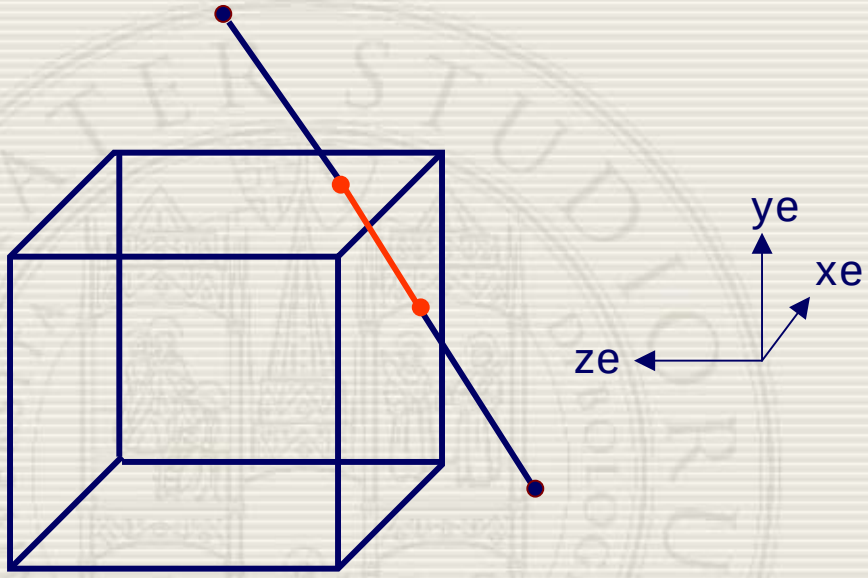
Questo algoritmo può essere esteso in 3D?

E in 3D? Piramide di Vista



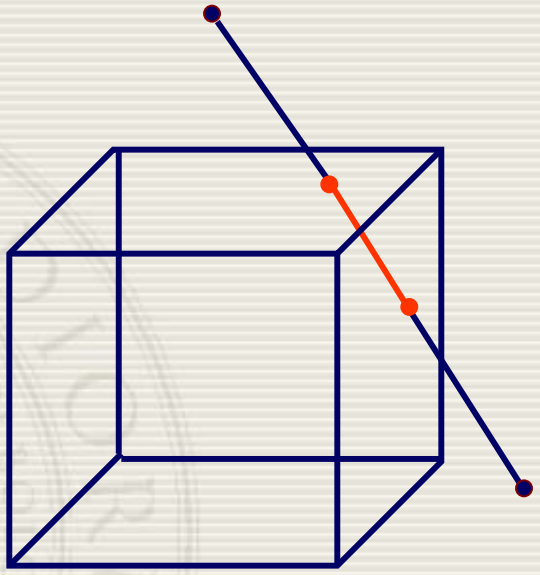
Trasformiamo il tronco di piramide ...

Piramide di Vista



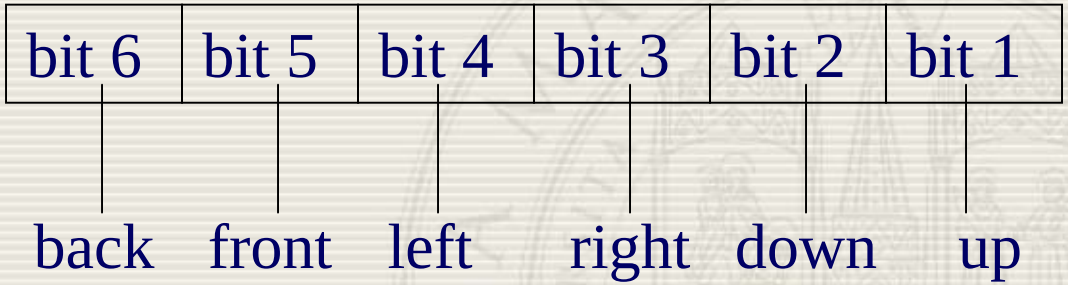
... in un cubo.

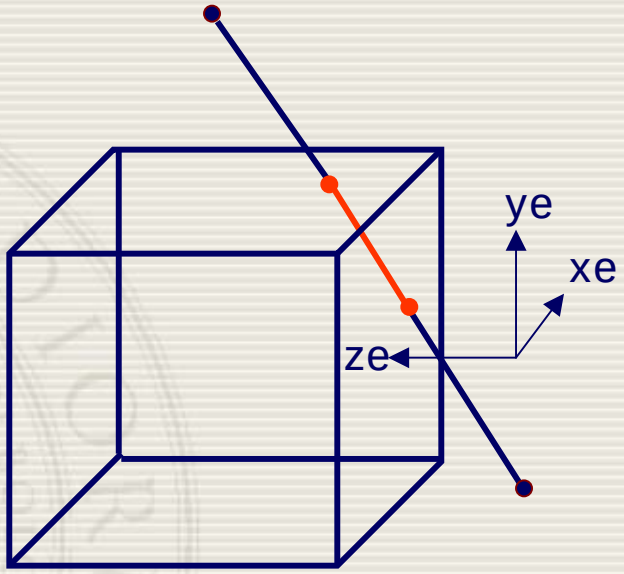
Cipping 3D di linee



- Prolunghiamo i piani della piramide/cubo 3D in modo da dividere lo spazio in 27 regioni.
- Diamo ad ogni regione un codice.
- Ad ogni punto dello spazio può essere associato un codice identificativo della zona in cui è; si tratta di un codice a 6 bit.
- Ogni bit del codice indica se il punto è interno o esterno alla piramide/cubo.

Cipping 3D di linee

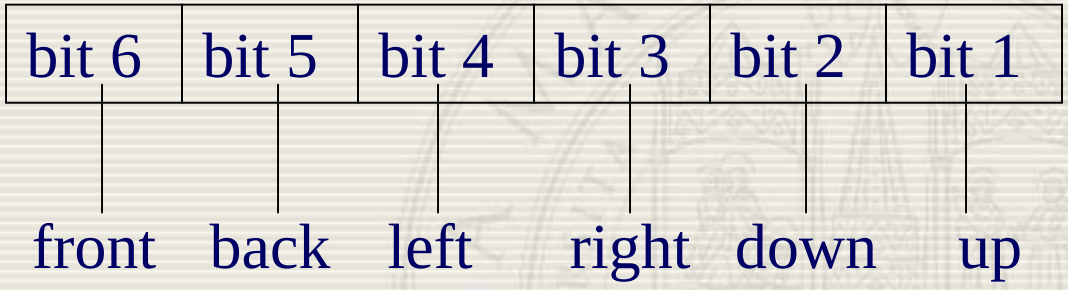


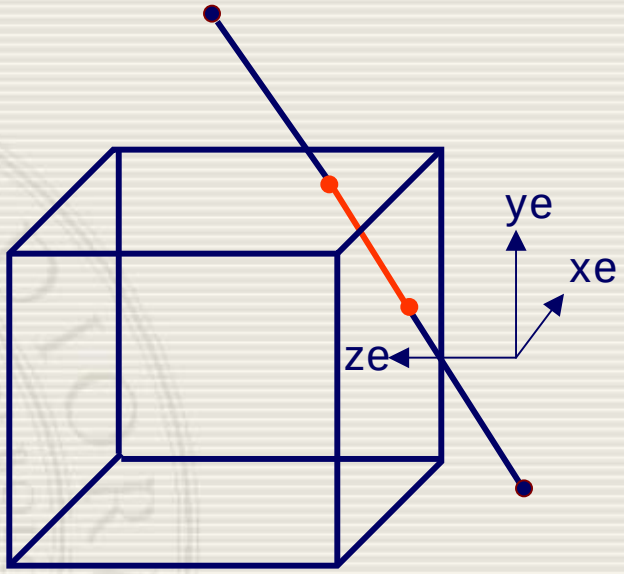


Un punto interno alla piramide/cubo avrà codice:

000000

Cipping 3D di linee





Piano up: y=ywmax
Piano down: y=ywmin
Piano right: x=xwmax
Piano left: x=xwmin
Piano back: z=zwmax
Piano front: z=zwmin

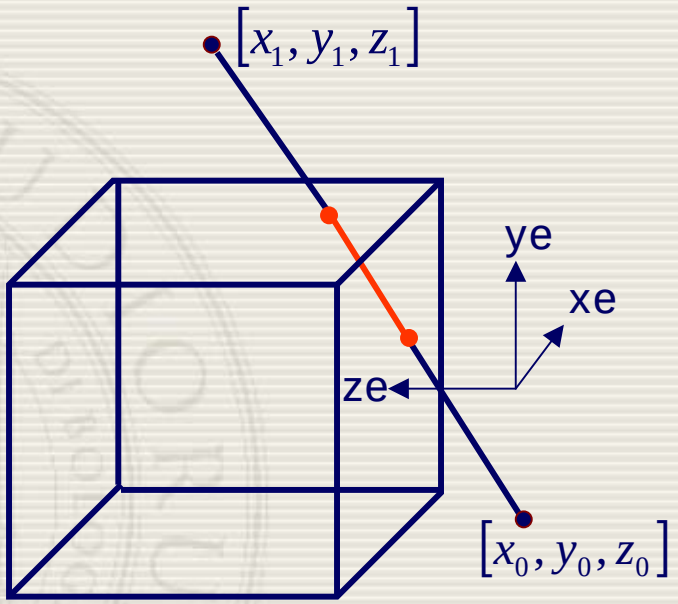
Intersezione fra Linee in forma Parametrica e piani cubo

$$\begin{cases} x(t) = x_0 + (x_1 - x_0)t \\ y(t) = y_0 + (y_1 - y_0)t \\ z(t) = z_0 + (z_1 - z_0)t \end{cases}$$

$$0 \le t \le 1$$

$$y = y_{w\max}$$

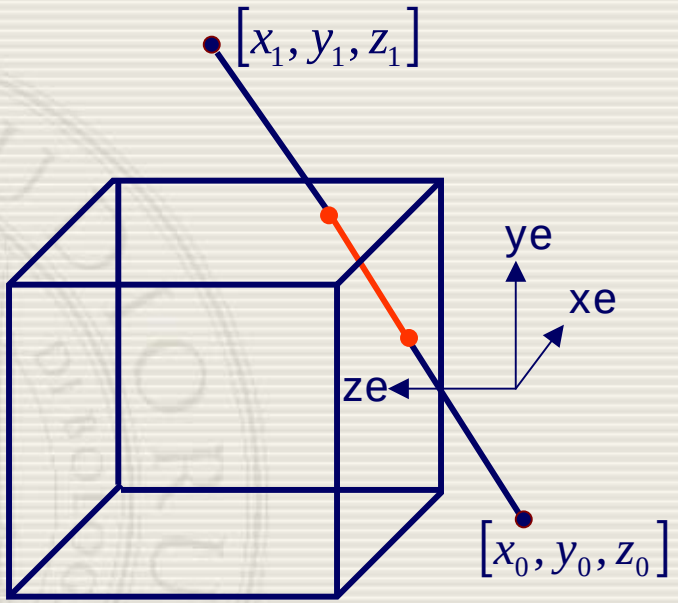
$$\begin{cases} y(t) = y_0 + (y_1 - y_0)t \\ y = y_{w\max} \end{cases}$$



Intersezione fra Linee in forma Parametrica e piani cubo

$$\begin{cases} y(t) = y_0 + (y_1 - y_0)t \\ y = y_{w\max} \end{cases}$$

$$t = \frac{y_{w\max} - y_0}{y_1 - y_0}$$



$$x_* = x_0 + (x_1 - x_0)\frac{y_{w\max} - y_0}{y_1 - y_0}$$

$$y_* = y_{w\max}$$

$$z_* = z_0 + (z_1 - z_0)\frac{y_{w\max} - y_0}{y_1 - y_0}$$