Miguel Alcón Doganoc
Algorithms for VLSI
October 25, 2019

Exercises on physical design

1 Quadratic placement

Show the two linear system of equations for the x and y coordinates.

Our system of equations is

$$AX - b_x = 0, \text{ and } AY - b_y = 0$$

where

$$A = \begin{bmatrix} 3 & 0 & -1 & -1 \\ 0 & 5 & -1 & 1 \\ -1 & -1 & 4 & 0 \\ -1 & -1 & 0 & 4 \end{bmatrix} \quad X = \begin{bmatrix} X_a \\ X_b \\ X_c \\ X_d \end{bmatrix} \quad b_x = \begin{bmatrix} 5 & 6 & 5 & 10 \end{bmatrix} \quad Y = \begin{bmatrix} Y_a \\ Y_b \\ Y_c \\ Y_d \end{bmatrix} \quad b_y = \begin{bmatrix} 0 & 12 & 2 & 5 \end{bmatrix}$$

So, the final systems are

$$\begin{cases} 3X_a - X_c - X_d - 5 = 0 \\ 5X_b - X_c - X_d - 6 = 0 \\ -X_a - X_b + 4X_c - 5 = 0 \\ -X_a - X_b + 4X_d - 10 = 0 \end{cases} \qquad \begin{cases} 3Y_a - Y_c - Y_d = 0 \\ 5Y_b - Y_c - Y_d - 12 = 0 \\ -Y_a - Y_b + 4Y_c - 2 = 0 \\ -Y_a - Y_b + 4Y_d - 5 = 0 \end{cases}$$

Solve the systems of equations and draw the final solution in a 2D plot.

The final solution is

$$a = (\frac{177}{44}, \frac{59}{44}) \qquad b = (\frac{115}{44}, \frac{141}{44}) \qquad c = (\frac{32}{11}, \frac{18}{11}) \qquad d = (\frac{183}{41}, \frac{105}{44})$$

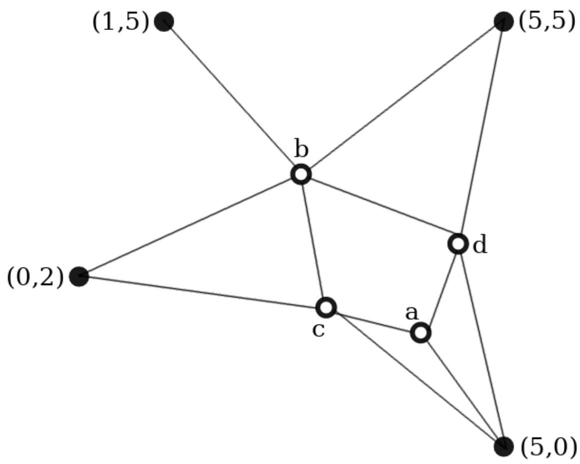
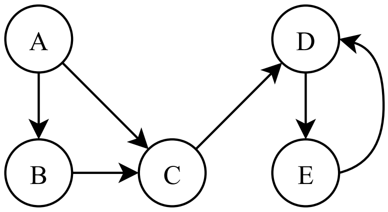


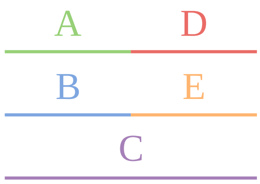
Figure 1: 2D plot of the solution

2 Channel Routing

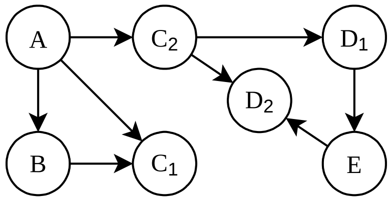
Draw the vertical constraint graph (VCG) without splitting the nets. Determine the zone representation for the nets. Draw the vertical constraint graph with net splitting.



(a) VCG without splitting



(b) Zone representation



(c) VCG without splitting

Find the minimum number of required tracks with net splitting and without net splitting.

Since without net splitting nodes D and E creates a loop in the VCG, it is impossible to route this channel. With net splitting, I found the number of tracks after applying the Dogleg Left-Edge algorithm, which is 5.

Use the Dogleg Left-Edge algorithm to route this channel. For each track, state which nets are assigned. Draw the final routed channel.

The assignment of nets to tracks is specified at the right of the route diagram.

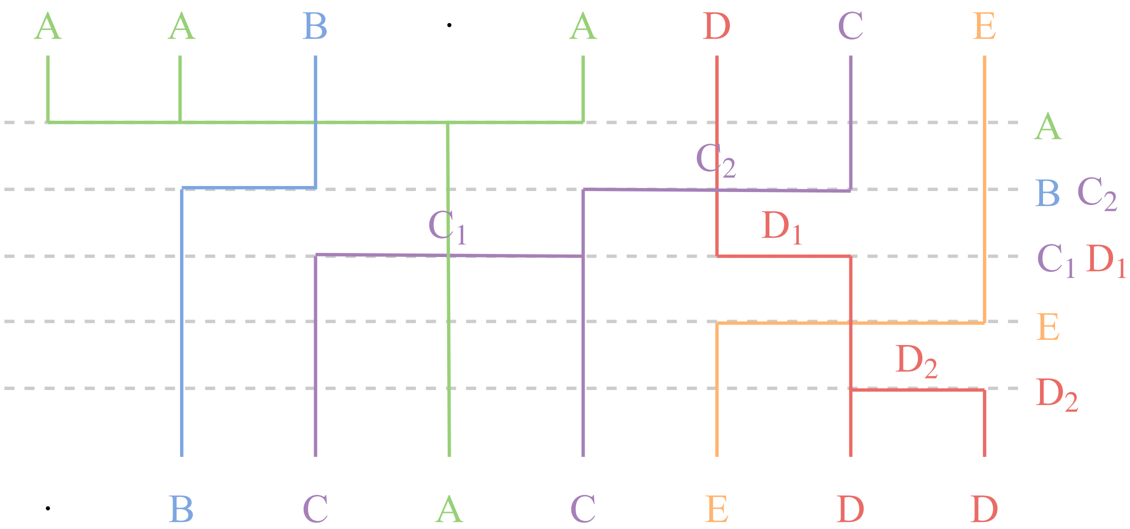
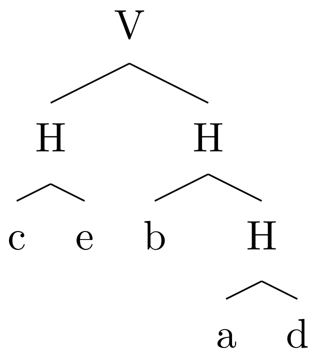


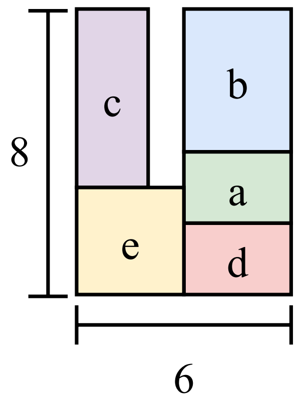
Figure 3: Final routed channel

3 Slicing Floorplan

Find the smallest bounding box that can accommodate the slicing floorplan assuming that blocks can be rotated by 90°. Show the shape functions at each node of the tree and the final floorplan.



(a) Initial floorplan tree



(b) Floorplan and bounding box

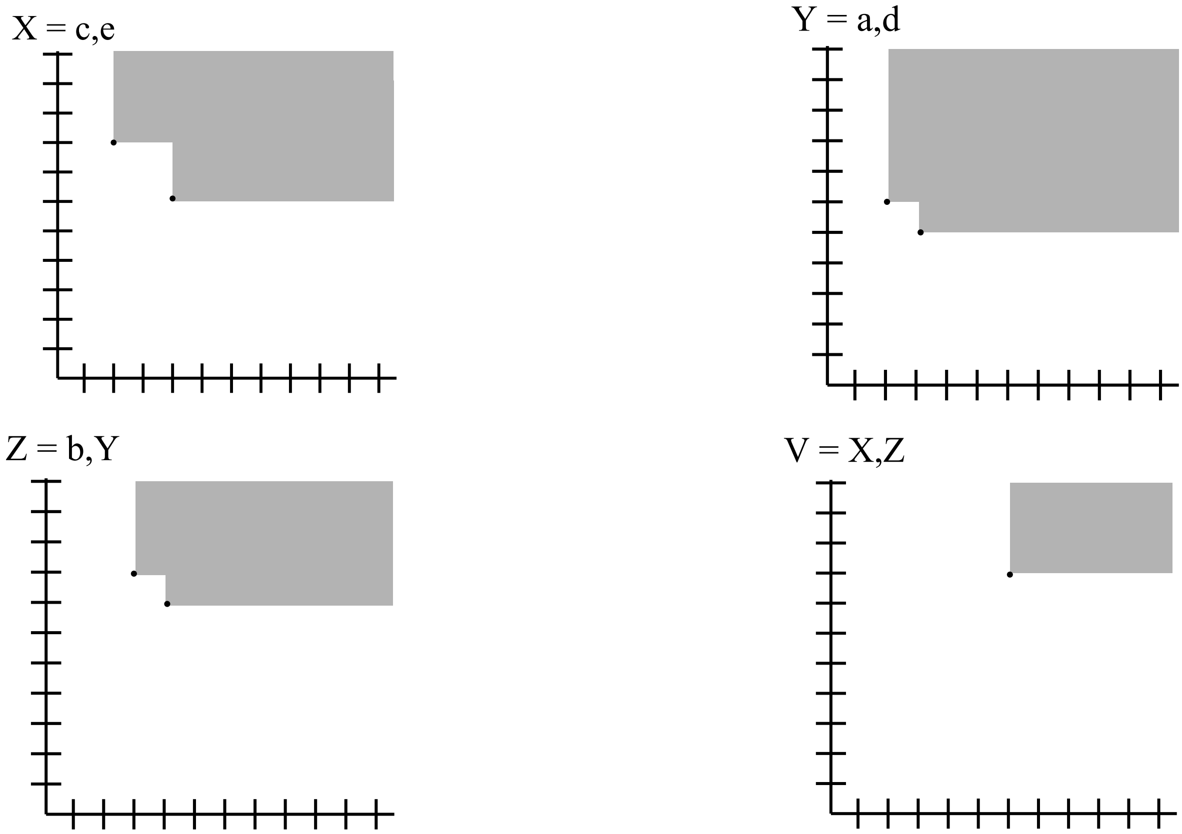
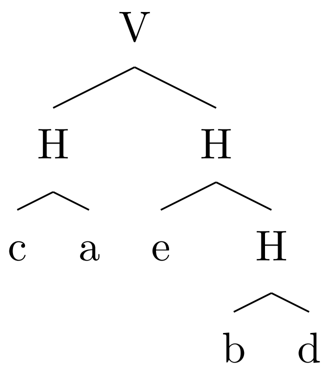
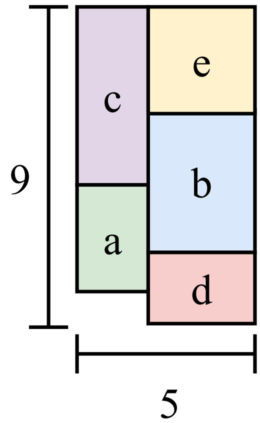


Figure 5: Shape functions

Find another slicing tree that can give a smaller floorplan. Draw the new floorplan.



(a) New floorplan tree

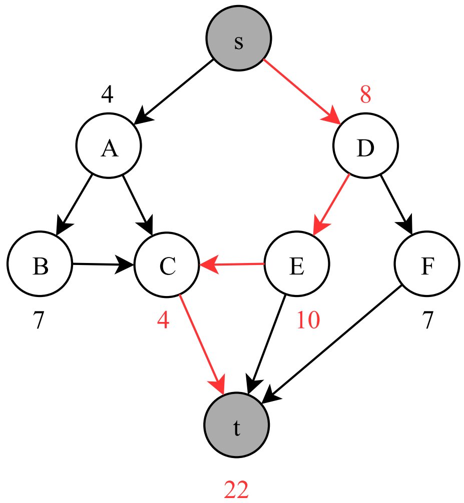


(b) New floorplan and bounding box

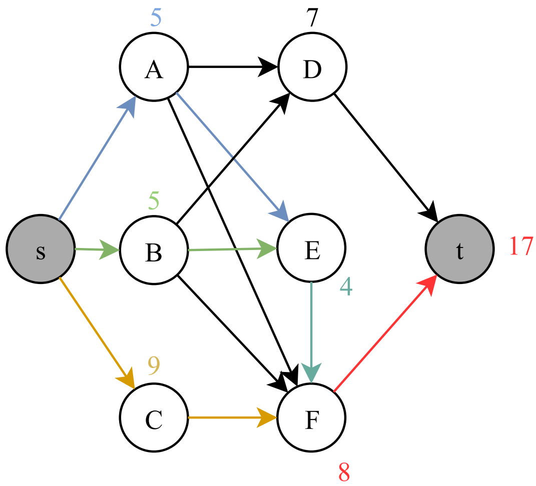
4 Sequence pairs

Calculate the coordinates of the modules in the floorplan represented by the following sequence pair: (ABDECF, CBAEFD).

First I drew the vertical and horizontal constraint graphs of the sequence pair to detect the critical (largest) paths.



(a) VCG

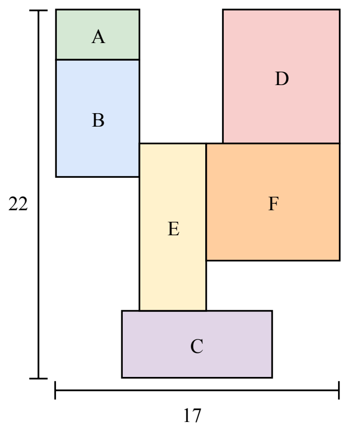


(b) HCG

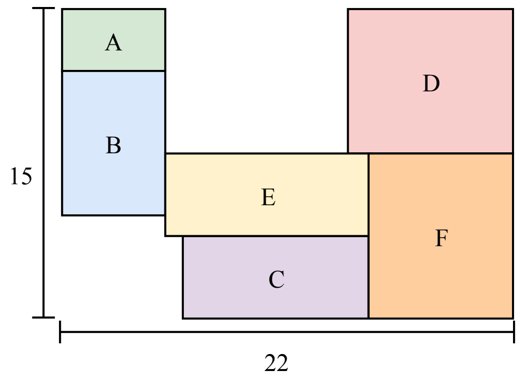
With it, I was able to compute the bounding box and draw the floorplan (figure 8a).

Given the same sequence pair, and assuming that modules can be rotated 90°, propose some rotation to reduce the area.

Rotating modules D, E and F, I obtained a smaller floorplan (figure 8b) (probably the minimum with this sequence pair). I rotated modules that were in one of the critical paths (vertical or horizontal), and that improved the floorplan.



(a) Initial



(b) With rotations

Figure 8: Floorplans

5 Partitioning

Given the graph in the figure, apply Kernighan&Lin to find a good bi-partition, starting from the one in the figure. Show all the steps performed during the execution of the algorithm.

Next you can see the two iterations that I had to do to reach the good bi-partition. Graph marked in grey is the best bi-partitioned graph of the iteration. The black border of the nodes means that they are fixed, i.e. they have already been switched.

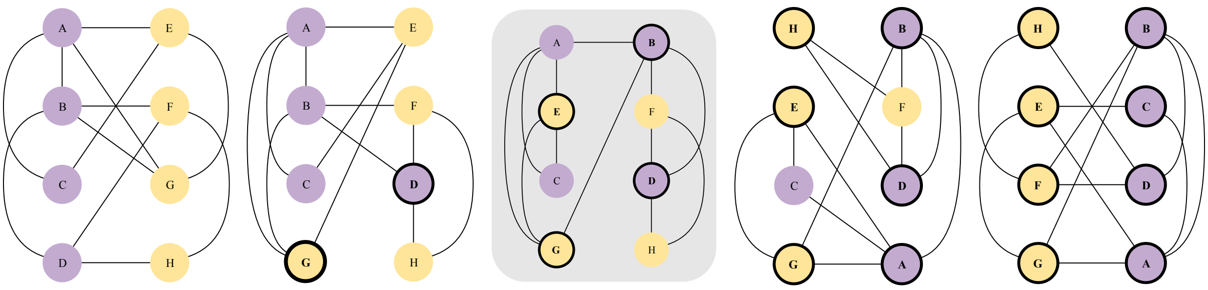


Figure 9: First iteration of the algorithm.

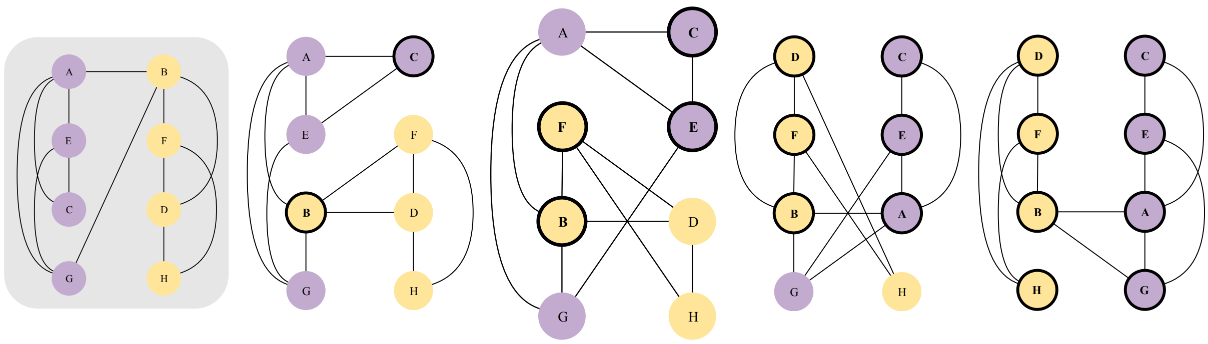


Figure 10: Second iteration of the algorithm.