CAD of Digital Circuits — Physical Design

Spring 2014

Homework # 3 - Solutions

1) Problem 1

a) Can such a placement problem be modeled as a *mixed-integer linear program* (MILP)? If yes, then describe your program, and generate a placement for the given netlist. If you believe that such a problem cannot be modeled as a MILP, then give the reason as to why the program is not linear.

Solution:

Theoretically, the program suggested would require a non-linear model to determine Manhattan distances between two cells. However, there are tricks to making this problem linear. They mostly lie in the decision on positive or negative value for wire lengths. Normally we would follow the formula for Manhattan distance using the following formula:

$$x_{length} + y_{length} = \sqrt{(x_A - x_B)^2} + \sqrt{(y_A - y_B)^2}$$

Now using tricks for linear solvers, such as LPSolve, we can create a Zero-One formulation that will only return positive values for the x and the y component. A possible approach to overcome this is to constrain your components for net length the following way:

```
//For all Nets N_i
x_i = xp_i - xn_i;
abs_x_i = xp_i + xn_i;
int xp_i, xn_i;
free x_i, xp_i, xn_i; //LPSolve constrains all variables to be > 0 otherwise
//The same goes for the values of y_i
```

Now for the rest of the problem formulation, here is a way to do it:

The main purpose of a placement problem is to find a routable solution and minimize the wirelength. To achieve this the objective function is total wirelength and the constraints are as follows:

$$\underset{cost}{\text{minimize}} \quad \sum_{i=1}^{3}\sum_{j=1}^{3}(i*|x_{mi}-x_{ni1}|+j*|y_{mj}-y_{nj}|) \quad \forall m,n\in\{1,\ldots,9\}, n\neq n$$

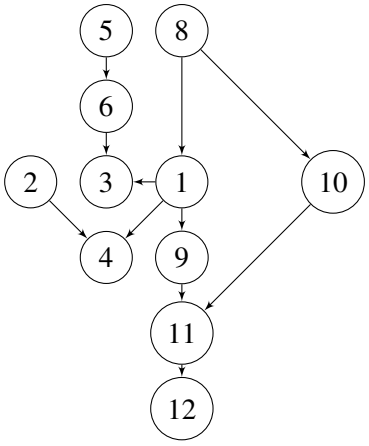
$$\text{subject to} \quad \sum_{k=1}^{3} x_{mi}=1 \qquad \forall m\in\{1,\ldots,9\}$$

$$\sum_{i=1}^{3} y_{mj}=1 \qquad \forall m\in\{1,\ldots,9\}$$

2) Problem 4

Solution:

The vertical constraint graph (VCG) for the given channel:

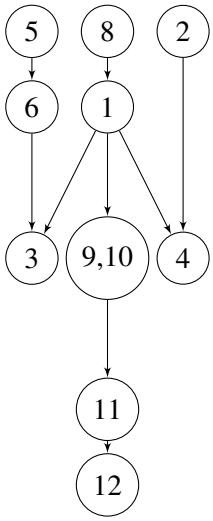


Yoshimura and Kun (Y-K) Algorithm tries to minimize the number of horizontal tracks for routing within the channel by merging nodes of the VCG without violating the horizontal constraint graph (HCG) and maintaining the max VCG (*i.e.* the nets are merged in such a way that they don't increase the max tracks given by the VCG and no nets overlap).

Merging of Two Nets: Two nets can be merged if:

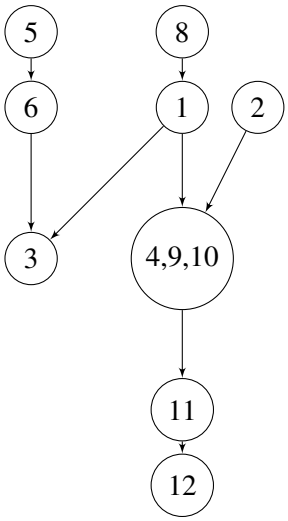
- No overlap exists between nets in the zone representation (*i.e.* they belong ot different zones).
- No directed path between nets (*i.e.* no cycles are introducted in VCG by merging nets).

a) Net 9 and Net 10 can be merged as they belong to different zones, Net 9 to Zones 3 and 4 while Net 10 to Zone 5.

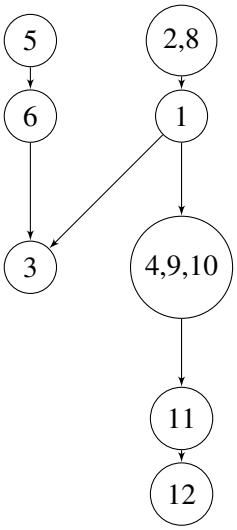


b) Net 6, cannot be merged with Net 1, as it will violate the HCG; Net 1 and 6 share Zone 2. Also, Net 1 cannot be merged with Net 2, as they share Zone 1.

c) Net 4 can be merged with Nets 9 & 10, as there is no zone overlapping or VCG violation.

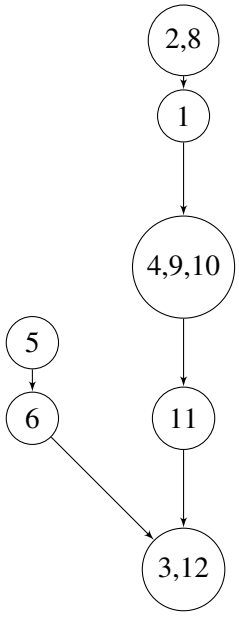


d) Net 2 can be merged with Net 8:

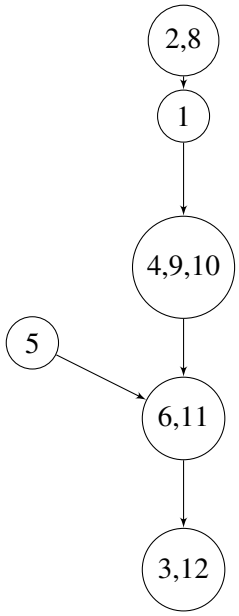


Now, Net 3 cannot be merged with already merged tracks for Net 4, 9, and 10. This is because Net 3 belongs to the same zone as Net 4.

e) We can merge Net 3 with either Net 11 or Net 12, as they both belong to different zones and no VCG Violation. I chose to merge Net 3 and Net 12:



f) Net 6 can be merged with either Net (4,9,10) or Net 11. No HCG or VCG violation exists. I will berge Net 6 with Net 11:



g) Net 5 can now be merged with either Nets (4,9,10) or (2,8) without violating any constraints or increasing the length due to the VCG. I will merge Net 5 with with Net (2,8):

Here we see that we have reduced to **5 tracks**.