Projects for the course *Algorithms for VLSI*

Jordi Cortadella

September 12, 2020

Abstract

Here is a very short description of potential projects that could be done for the course. The students are also encouraged to propose other projects that could have some special interest for them. It is essential that the projects contain a significant algorithmic component.

1 Graph drawing

The project consists in implementing an algorithm for graph drawing (see https://en.wikipedia.org/wiki/Graph_drawing). The algorithm will have to read some simple specification of a graph (vertices and edges) and depict a good-quality pictorial representation of the graph after applying an algorithm that tries to create an understandable visualization of the graph.

Two strategis may be considered for graph drawing:

- Spectral methods using eigenvectors [Kor05]
- Multilevel force-directed methods [Hu06]

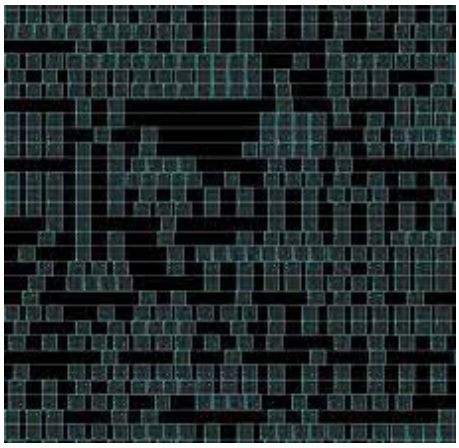
The benchmarks for graph drawing can be obtained from this library.

2 Cell placement

This project consists in placing the cells of a circuit in a 2D surface trying to minimize the total wire length (see https://en.wikipedia.org/wiki/Placement_(EDA)).

Conceptually, this project has some similarity to the project on graph drawing. The algorithm will have to read a netlist (a graph of logic cells) and position the nodes in a grid (an $n \times m$ matrix) in such a way that the total wire length is minimized. As a simplification, all cells will be assumed to occupy one slot in the matrix.

The project will have to use some meta-heuristics to solve the problem (e.g., Simulated Annealing [KGV83] or Extremal Optimization [BP03]).

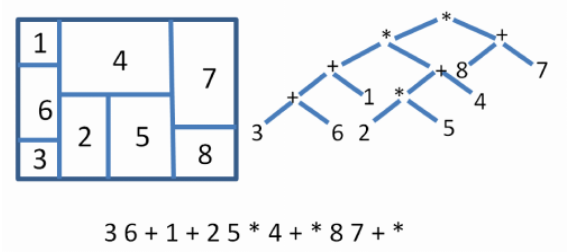


3 Floorplanning

The project consists in placing a set of blocks in a 2D surface. The quality of the layout will be determined by the area of the bounding box and the total wire length. See https://en.wikipedia.org/wiki/Floorplan_(microelectronics).

The final layout will have to be a sliceable floorplan (representable as a binary tree). The floorplanning algorithm will allow the blocks to be rotated according to some shape functions defined a priori.

A possible strategy for the implementation of the algorithm is described in [WL86].



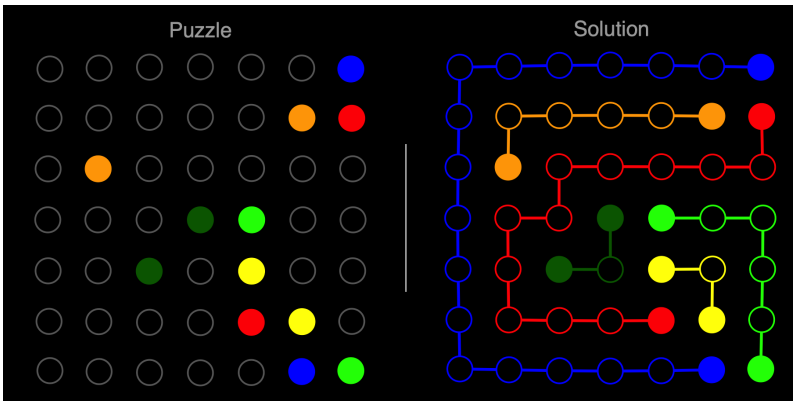
4 Routing

The project consists in designing an algorithm (or a mathematical model for a SAT or ILP solver), that generates routes in a 2D grid to connect pairs of dots. The input of the problem will be the dimensions of a grid (e.g., 4×5), and a set of pairs of nodes, e.g., $\{(1,1),(3,2)\}$, $\{(2,4),(3,1)\}$, ... The algorithm will have to find routes along the edges of the grid that connect the pairs of nodes without creating any short circuit among them.

The project is a simplified version of a similar problem solved in this paper [CPGM14]:

https://www.cs.upc.edu/~jordicf/gavina/BIB/files/tcad2014_Routing.pdf

For more intuition, the problems to be solved are similar to the Flow Free game.



5 Two-level minimization

The project consists in implementing an exact algorithm for two-level minimization, i.e., finding the smallest sum-of-product form of a Boolean expression. The algorithm should be based on Quine-McCluskey method [McC56].

Given an expression representing a Boolean function, the algorithm will have to generate all prime implicants and select the smallest subset that covers all minterms of the function.

For example, given the following truth table

$abcd$	f
0000	0
0001	0
0010	-
0011	1
0100	0
0101	0
0110	0
0111	1
1000	0
1001	0
1010	0
1011	0
1100	1
1101	-
1110	1
1111	1

where '-' represents a *don't care* value, the minimizer should give the representation:

$$f = ab + \bar{a}cd$$

6 Algebraic factorization

The project consists in implementing an algorithm to find a good factorization of a Boolean expression using algebraic methods. For example, given the expression:

$$F = adf + aef + bdf + bef + cdf + cef + bfg + h$$

the factorization algorithm could find a multi-level expression such as this one:

$$F = (a + b + c)(d + e)f + bfg + h.$$

One of the popular algorithms for factorization is based on kernel extraction [Bra87] and reduces the problem to finding rectangles in a Boolean matrix. Another possible strategy is based on Graph Partitioning [GM99].

7 Solving Boolean relations using SAT

A Boolean relation is formalism to describe the behavior of a multi-output circuit. For each value at the inputs, several values at the outputs are acceptable.

Solving a Boolean relation means deriving Boolean functions that implement one of the acceptable behaviors. Some work has been done in the past using Binary Decision Diagrams [BCK09]. However, this approach can only be used for small Boolean relations.

A later approach [JLH09] proposed the use of SAT to solve large Boolean relations. The project will consist of implementing this approach.

References

[BCK09] D. Baneres, J. Cortadella, and M. Kishinevsky. A recursive paradigm to solve boolean relations. *IEEE Transactions on Computers*, 58(4):512–527, 2009.

[BP03] Stefan Boettcher and Allon G. Percus. *Extremal Optimization: An Evolutionary Local-Search Algorithm*, pages 61–77. Springer US, Boston, MA, 2003.

[Bra87] R. K. Brayton. Factoring logic functions. *IBM Journal of Research and Development*, 31(2):187–198, 1987.

[CPGM14] Jordi Cortadella, Jordi Petit, Sergio Gómez, and Francesc Moll. A Boolean Rule-Based Approach for Manufacturability-Aware Cell Routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(3):409–422, March 2014.

[GM99] M. C. Golumbic and A. Mintz. Factoring logic functions using graph partitioning. In *IEEE/ACM International Conference on Computer-Aided Design*, pages 195–198, 1999.

[Hu06] Yifan Hu. Efficient, high-quality force-directed graph drawing. *The Mathematica journal*, 10:37–71, 2006.

[JLH09] J. R. Jiang, H. Lin, and W. Hung. Interpolating functions from large boolean relations. In *IEEE/ACM International Conference on Computer-Aided Design*, pages 779–784, 2009.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[Kor05] Y. Koren. Drawing graphs by eigenvectors: theory and practice. *Computers & Mathematics With Applications*, 49:1867–1888, 2005.

[McC56] E. J. McCluskey. Minimization of boolean functions. *The Bell System Technical Journal*, 35(6):1417–1444, 1956.

[WL86] D. F. Wong and C. L. Liu. A new algorithm for floorplan design. In *23rd ACM/IEEE Design Automation Conference*, pages 101–107, 1986.