Layout Algorithms for VLSI Design

ANDREA S. LAPAUGH

Department of Computer Science, Princeton University

VLSI (very large scale integrated) circuits contain hundreds of thousands to millions of transistors. To design a VLSI circuit, one must determine the placement of semiconductor materials for each transistor on the chip. However, it is not feasible to consider the positioning of each transistor separately. Transistors are organized into subcircuits called components; often a hierarchy of components is created between the individual transistors and the complete VLSI circuit. Wires, running in two or more layers on a planar substrate, interconnect the components. The layout problem for VLSI circuits becomes one of positioning components and their interconnecting wires on the plane, following design rules, and optimizing some measure such as area or wire length. Within this basic problem structure are a multitude of variations arising from changes in design rules, flexibilities within components as to size, shape, and regions where wires may connect to components, and a range of layout styles. Graph models are used heavily, both to model the components and interconnections themselves and to capture constraint among objects. Geometric aspects of the layout problem must also be modeled. Most layout problems are optimization problems and most are NP-complete. Therefore heuristics are also employed heavily.

The layout problem is often decomposed into two stages: placement of components and routing of wires. Placement algorithms can be divided into two types: constructive initial placement algorithms and iterative improvement algorithms. For a given set of components and nets,[1] both types of algorithms try to find a legal placement that optimizes some cost function. Common cost functions measure area used by the configuration of components, estimated routing area, estimated total wire length, estimated maximum wire length of a net, or a combination of these. The problem of wire routing is actually first encountered as one tries to estimate costs for placement.

For VLSI layout, routing is often done using the *channel model*. In this model, the regions of the chip that are not covered by components are partitioned into nonoverlapping rectangles called *channels*. The routing problem is then decomposed into two subproblems: global routing and local or detailed routing. Global routing chooses which channels will be used to make the interconnections for each net. Detailed routing, called *channel routing*, determines separately for each channel the paths to be used for each wire. Channel routing itself is a very well-developed area of study, with many models and algorithms for different layout situations. In most models, the primary goal is to minimize the size of the channel needed for successful routing. Almost all models yield NP-complete routing problems. One exception is the single-layer case, which has a rich and elegant theory based on a constraint model that admits polynomial-time algorithms, even under generalizations.

Iterative improvement is commonly

[1]A net is a sets of points that must be interconnected. These points usually lie on the boundaries of components and are called terminals.

This work was supported in part by the National Science Foundation, FAW award MIP-9023542.

used to try heuristically to improve candidate solutions to NP-complete problems. The domain of layout problems is no exception. One of the most sophisticated and successful paradigms for general iterative improvement is simulated annealing; it has been applied successfully to many versions of the placement problem.

Graph algorithms are heavily used as foundations for algorithms in all aspects of layout. One important technique is graph partitioning. Graph partitioning underlies the partitioning of circuits into chip-sized subcircuits and forms the core of a recursive partitioning method for constructive initial placement. Two graph-partitioning techniques are often used as the basis of algorithms for these applications: the Kernighan-Lin algorithm and simulated annealing. Spectral methods, based on eigenvectors of the Laplacian, have been used increasingly in recent years.

Not surprisingly, path-finding algorithms also play an important role in solving layout problems. The most efficient algorithms for these problems find direct applications in placement and routing. Minimum spanning trees are used in estimating wire lengths; minimum Steiner trees are needed to generate wire paths. Depending on the application, Euclidean distance, rectilinear distance, or general edge costs are used to define tree costs. To find minimum Steiner trees (NP-complete under all the preceding metrics), a variety of heuristic algorithms are used, including breaking up the tree into point-to-point connections and finding shortest paths. Basic breadth-first search and more general single-source shortest-path algorithms are used. Single-source shortest-path algorithms are also important in solving a special class of linear constraint problems in which all constraints are of the form $x - y \leq d$. This class of constraints, and various subclasses, arise in one-dimensional compaction of layout area and in single-layer routing.

The first concern of designers of VLSI layout algorithms has been reducing layout area. However, as the field has matured, researchers have added circuit performance as a primary criterion. In recent work on performance-driven layout, one finds use of the techniques of linear programming, integer programming, and even specialized higher-order programming. Concerns in performance-driven layout include minimizing skew in clock net routing and minimizing crosstalk in routing, as well as minimizing circuit delay during placement and routing.

Historically and currently, developments in technology influence layout models and algorithms. The first layout algorithms were developed for printed circuit boards, beginning in the late 1950s and early 1960s. Foundations developed for printed circuit boards were also relevant to layout techniques for VLSI circuits. Recent developments in technology that are influencing work in layout include the increase of routing layers, the continuing shrinking of feature sizes, and the introduction of new hardware implementation media. The foundational work in routing has used two routing layers. An increase in the number of layers available for routing[2] has motivated work on multilayer models, though most of the multilayer routing techniques are derived from single-layer or two-layer techniques. The continuing shrinking of the minimum feature-size of integrated circuits has increased the role of wire delay in determining total circuit speed. This trend accelerates the pursuit of performance-based layout methods. As field-programmable gate arrays have gained in popularity, layout techniques for the particular constraints of that very structured design style have been developed. The use of multichip modules for high-performance packaging has also presented a new set of problems in layout. Thus, founded on the techniques of combinatorial algorithms and a core of

[2]At this time, state of the art for microprocessor chip technology is four routing layers.

mature layout algorithms, layout is expanding to include new layout problems for new technologies.

Layout is not the only area of computer-aided digital design that has made effective use of algorithmic techniques. Many aspects of digital design yield to combinatorial models and algorithmic techniques, for example, scheduling functional units in high-level design, logic optimization, simulation (at varying levels of abstraction), and artwork analysis. Using such techniques, the field of computer-aided design has had great success in achieving its goal of developing tools to assist designers in producing digital systems correctly and efficiently.

REFERENCES

CORMEN, T. H., LEISERSON, C. E., AND RIVEST, R. L. 1990. *Introduction to Algorithms*. MIT Press, Cambridge, MA.

LENGAUER, T. 1990. *Combinatorial Algorithms for Integrated Circuit Layout*. Wiley, West Sussex, England.

SHERWANI, N. 1995. *Algorithms for VLSI Physical Design Automation*, 2nd ed. Kluwer Academic, Norwell, MA.

WOLF, W. H. 1994. *Modern VLSI Design: A Systems Approach*. Prentice-Hall, Englewood Cliffs, NJ.