Area-Efficient Graph Layouts (for VLSI)

Charles E. Leiserson

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

***Abstract*—Minimizing the area of a circuit is an important problem in the domain of *Very Large Scale Integration*. We use a theoretical VLSI model to reduce this problem to one of laying out a graph, where the transistors and wires of the circuit are identified with the vertices and edges of the graph. We give an algorithm that produces VLSI layouts for classes of graphs that have good *separator theorems*. We show in particular that any planar graph of n vertices has an $O(n \lg^2 n)$ area layout and that any tree of n vertices can be laid out in linear area. The algorithm maintains a sparse representation for layouts that is based on the well-known UNION-FIND data structure, and as a result, the running time devoted to bookkeeping is nearly linear.**

1. Introduction

The remarkable advance of very large scale integrated (VLSI) circuitry has sparked research into the design of algorithms suitable for direct hardware implementation. To the computer theorist, VLSI provides an attractive model of parallel computation for three reasons. First of all, the number of components that can fit on a single chip is large, and beyond that has been doubling every one to two years. It is currently possible to place 10^5 components on a single chip, and it is projected that this number will very likely grow to 10^7 or even 10^8. These large numbers make asymptotic analysis and other theoretical tools applicable to this engineering discipline. Secondly, VLSI hardware expense can be related directly to the very mathematical and geometric cost function of *area*. Unlike older technologies, the components and interconnections between components are made out of the same "stuff" in VLSI, and hence area is a uniform cost measure for both. Finally, VLSI provides a model of parallel computation that includes communication costs as well as operation counts. The cost of communication is represented explicitly as the area of a fixed-width wire between two processors. In fact, interconnections can consume most of the area of an integrated circuit chip. A major goal, therefore, is to minimize the area required by particular interconnection schemes. This paper examines the problem in an abstract setting: *"Given a graph, produce an area-efficient layout."*

To illustrate the subtleties inherent in this problem, consider laying out a complete binary tree of $n = 2^k - 1$ vertices. Figure 1 shows an

This research was supported in part by the Defense Advanced Research Projects Agency under Contract F33615-78-C-1551 (monitored by the Air Force Office of Scientific Research), the National Science Foundation under Grant MCS 78-236-76, the Office of Naval Research under Contract N00014-76-C-0370, and a Fannie and John Hertz Foundation fellowship.

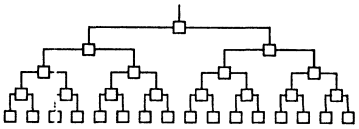

Figure 1: An $O(n \lg n)$ layout of a complete binary tree.

obvious solution that requires $O(n \lg n)$ area—$O(n)$ across the bottom times $O(\lg n)$ height. Observe that as we ascend the tree from the leaves to the root, the number of wires is halved from one level to the next, but the length of the wires doubles. This means that the amount of wire devoted to each level of the tree is the same. The recurrence that describes the area required by this layout is $A(n) = 1$ for $n = 1$, and

$$A(n) = 2A(\lfloor n/2 \rfloor) + n/2$$

for $n = 2^k - 1$ where $k > 1$.

There is a more efficient solution to this embedding problem. The so-called *H-tree* layout [17] shown in Figure 2 requires only $O(n)$ area

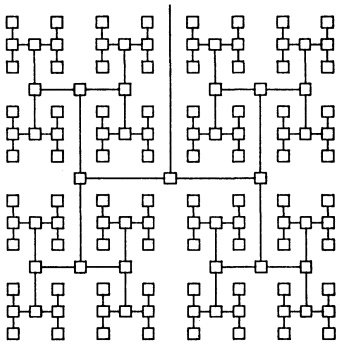

Figure 2: The linear area *H-tree* layout of a complete binary tree.

in spite of the fact that relatively long wires are used towards the root of the tree. In this layout, the number of wires is halved from level to level as we ascend to the root, but the length of the wires doubles only every two levels. Whereas, the standard $O(n \lg n)$ layout uses just one dimension for routing most of the wires, the H-tree makes better use of both spacial dimensions. The recurrence describing the area required by the H-tree is more complex than the previous one because of its nonlinear form: $A(n) = 1$ for $n = 1$, and

$$A(n) = 4A(\lfloor n/4 \rfloor) + 4\sqrt{A(\lfloor n/4 \rfloor)} + 1$$

for $n = 2 \cdot 4^k - 1$ where $k \geq 1$.