U.C. Berkeley — CS270: Graduate Algorithms | Lecture 16
Professor Satish Rao | October 25, 2006
Lecturer: Satish Rao Last revised Scribe so far: Satish Rao cribbing from previous years lectures

---

Lecture 16

1 Overview

Today, we will discuss models and algorithms for VLSI (very large scale integration). This, like PRAM algorithms, were a much studied topic in the past and the fundamentals of this area should inform us even in modern times.

2 VLSI Area

These days we build very large chips. Perhaps some of the basic features of the technology are

- The chip is on a thin silicon surface. It is basically 2 dimensional. That is, though there are a few layers of a chip, in the third dimension it does not scale beyond some fixed number.
- Same material is used for wires and devices.
- The resolution is limited to some size based on the physical process that is used in the manufacturing. For example, at some point, the process was a photographic process and the resolution was limited to the wavelength of the light. That is, basic device size, wirewidth, and separation between components were all larger than the wavelength of the light used in the process.

3 Thompson grid model

The Thompson grid model is a model of VLSI design.

1. The chip is a rectangular area with evenly space horizontal and vertical tracks.
2. Devices (processors?) are placed at track intersection.
3. Wires occupy tracks.
4. Via used to connect horizontal and vertical tracks.
5. Systolic operation. Signals propagate in one clock tick across wires independent of length.

Consider an example layout of the "circuit" that consists of K_4. That is, there are nodes A, B, C, D where they are all connected. In class, we gave a layout for this circuit. There are several measures of quality on the layout. For example,

- **Bounding box area** for the example layout in class was $3 * 5 = 15$. This models the size on a chip of the smallest rectangle that contains the layout.
- **Total wire length.** The sum of the wirelengths is another measure. This is certainly upper bounded by the bounding box area and usually we just use that measure as the relevant one.
- **Maximum wire length.** The maximum length of a wire may decrease the quality of the operation of the circuit. Thus, it is often separately optimized. In some cases, there may be a tradeoff between the wirelength and the bounding box area. Indeed, there are fancy tricks, where circuit designers duplicate computation (or algorithm designers) to minimize the effect of long wires.
- **Number of vias.** The example above had four. It is basically the number of turns of the wires.
- **Number of crossings.** This measure is trying to control for crosstalk among the wires. Though, these days, depending on this is probably dangerous.

We will primarily be concerned with bounding box area, and perhaps with wire length. Still, this model is simplified. For example, some of the implicit assumptions are as follows. The wires are unit delay. In fact, the wires may be ruled by their capacitance in which case the delay is proportional to $\log L$ wire L is the length of the wire. For capacitive and resistive wires the delay is linear in L. Another assumption is that there are no buses, the wires are point to point. One can, however, implement a k-terminal bus using a k leaf tree consisting of 2 terminal wires. The routing is rectilinear, other directions are dissallowed. This assumption has been holding in current technology, though perhaps will change in the future. Each node has degree at most 4. For higher degree graphs, a node is allowed to occupy more grid squares. For our discussion, we deal with only degree 4 circuits.

Another assumption is the two layer assumption, real chips may have many layers. This assumption has bounded cost, as suggested by the following lemma.

Lemma 1
If A is the area of a circuit using a k layer model, then the area in a 2-layer grid mode has area at most $O(k^2 A)$.

The idea is to stagger and project. That is, for each layer we grow each dimension by a factor of $2k$. Then horizontal track j in layer i is mapped to the horizontal track numbered $2kj + 2i$. The odd numbered tracks are avaiable for routing vias.

4 Example layouts.

We will give a example layouts for a few graphs.

For the 2-dimensional $n \times n$ area, the layout is clear. Here the area is $O(N)$ where N is the number of nodes in the grid, and the maximum wire length for this layout is 1.

For a 3-dimensional $n \times n \times n$ grid, we can use the stagger and project layout for the n-layer model. Here, the layout area is $O(n^4) = O(N^{4/3})$. The maximum wire length is $O(n) = O(N^{1/3})$.

For a complete binary tree, we can do the H-tree layout, which conists of recursive layouts of 4 binary trees, where the roots are in the middle of the layout, plus placing the root of this tree in the middle of the H and its children at the intersection points of the H. The four points of the tree are connected to the roots of the sublayouts. Note that for use to connect to the roots of the recursive layouts, we "cut" them and add this wire to the root. We assume we orient the layouts to increase the maximum sidelength of the box in a mimimal manner.

Here, the layout area is determined by the following recurrence on the sidelength.

$$S(N) \leq 2 + 2S(N/4).$$

By inspection, we can conclude that $S(N) = O(\sqrt{N})$. Thus, the bounding box area is at most $O(N)$. This layout has wirelength of $O(\sqrt{N})$, which is in some sense as bad as possible (i.e., one wire traverses the whole chip.) On the other hand, one cannot do too much better as we show in the next section.

5 Simple lower bounds.

To provide a lower bound on the area of the layout of a graph, we define the notion of bisection width.

DEFINITION 1 *The bisection width of a graph $G = (V, E)$ is the minimum over partitions $(S, \overline{S})$ of V where $|S| = |\overline{S}|$ of the number of edges between S and $\overline{S}$.*

For example, the bisection width of a 2-d $n \times n$ grid is at least $n = \Omega(\sqrt{N})$.

Actually, this requires a proof since it appears difficult to look at all bisections of the grid. Though, surely they must be worse than the natural bisection in terms of the number of crossing edges.

Before discussing proving lower bounds on bisections, we state the following lemma regarding its usefulness for reasoning about VLSI.

LEMMA 2
The VLSI layout of a graph is $\Omega(B^2)$ where B is the bisection width of the graph.

PROOF:
We consider a layout of the graph. Then we slide a vertical line (assuming without loss of generality that the vertical direction is the thinner one of the layout) across the layout until exactly half the nodes of the graph are on one side of the cut and half are on the other. We may need to have the cut jog by one to do so.

The number of wires crossing the cut between two sides of this bisection is at most $\sqrt{A} + 1$ since this is a legal layout of the graph. Thus,

$$B \leq \sqrt{A} + 1.$$

Squaring both sides, gives us the lemma.
□

We can apply this lemma to 3-d grids (assuming the bisection width is at least $N^{2/3}$ to conclude that the layout of the previous section was optimal.

We can also prove a lower bound on the minimum wirelength of a graph based on its diameter.

LEMMA 3
The maximum wirelength of any layout of a diameter d graph G must be $\Omega(\sqrt{A}/d)$. Diameter is defined to be the largest possible length of a shortest path in G.

PROOF:
Let S and T be active elements on the left and right boundaries of the layout (where the layout is wider than it is high.) These correspond to 2 edges in the graph. There is a path connecting the endpoints of the edges that consists of d edges. That path and the two current edges travel in the layout from the left to right edge. Thus, the average length of the edges in the path is at least $\sqrt{A}/(d+2)$. At least one edge must have the average length. This proved the lemma.
□

For a binary tree, the minimum wirelength must be $\Omega(\sqrt{N}/\log N)$ since the length of the path is at most $O(\log N)$. Thus, the H-tree layout is close to optimal, though one can indeed do better.

6 Lower bounds on bisection width.

Question 1: What is the bisection width of the complete graph on N nodes, K_N.

Question 2: Show that the bisection width of the 3-dimensional $n \times n \times n$ grid is $\Omega(n^2)$.

7 Divide and conquer layouts.

Recall, the H-tree layout for the complete binary tree. We sliced the recusive layout to add the root node (and its children.)

This is a generic technique. That is, one can add an edge to any layout, by simply slicing the current layout vertically and horizontally next two the two endpoints of the edge.

Similiarly, one can add a node by making two slices, and then adding each of the four edges.

We can use this technique to give a recursive method for finding layouts.

We use the notion of *vertex separators* for a graph $G = (V, E)$ which are defined as a partition of V into A, B and C such that

- $|A|, |B| \leq |V|/2$.
- there is no edge between A and B.

A graph family (a set of graphs that is closed under subgraphs) is said to have $f(N)$-bisectors if any N-node graph in the family has vertex separator where $|S| \leq f(N)$.

For example, the family considting of subgraphs of grid graphs have an $O(\sqrt{N})$-bisectors. Furthermore, planar graphs have $O(\sqrt{N})$-bisectors.

We can now state the following lemma that follows from slicing.

LEMMA 4
Any graph in a graph family with $f(N)$-bisectors has a VLSI sidelength that is determined by the following recurrence,

$$S(n) \leq 2S(n/4) + f(N),$$

with $S(4) = O(1)$.

Thus, for planar graphs, the recurrence is
$S(n) \leq 2S(n/4) + O(\sqrt{N})$.

This solves to $O(\sqrt{N} \log N)$. Thus, the area is at most $O(N \log^2 N)$. In fact, the area is required to be $\Omega(N \log N)$ by a very sophisticated argument of Leighton. Closing this gap remains a difficult and perhaps interesting question.

For butterfly networks, this technique gives optimal $O(N^2/\log^2 N)$ area layouts. For meshes, it gives non-optimal $O(N \log^2 N)$ layouts.

One could similiarly use edge bisectors to provide a recurrence as well.

8 Remark

Finally, we note that for all degree d graphs, one need not do slicing, one can simply layout the nodes on a horizontal line(using d by d grids for each), and connect them by allocating a two vertical tracks and one horizontal track for each edge. This gives an area $O(N^2 d^2)$ layout for these graphs. (Asymptotically, the same as slicing.)