U.C. Berkeley — CS270: Graduate Algorithms
Professor Satish Rao
Lecturer: Satish Rao

Lecture 17
October 30, 2006
Last revised Scribe so far: Satish Rao

Lecture 17

1 Overview

Today we will discuss the VLSI chip design. In particular, we will describe a simple model for circuits, and show lower bounds for various circuits. This material as does the previous lecture directly borrows from material in a long ago course by Ranade.

2 The Or operation.

How do we build an or circuit?

- A simple circuit is a binary tree, and we could implement the tree where all the inputs are in a row at the bottom, and we build the internal gates on top, in $\log N$ levels. This yields a circuit of width N and height $\log N$. That is, its area, A, is $N \log N$. The time, T, that it takes to compute the result is $O(\log N)$. The product AT, and perhaps a decent measure of the total resource usage, is $O(N \log^N)$.
- We could do better in terms of area by using the H tree layout or divide and conquer since this has a separator family of size $O(1)$. Here, the area is $O(N)$. Thus, AT is $O(N \log N)$.
- We can do still better as follows. We build an OR circuit on $N/\log N$ nodes. Then, we group $logN$ inputs and feed them serially into an input OR gate that computes and stores the OR of its internal state and the input. The outputs of this gate are connected to the OR circuit. At time step $\log N$, the computation proceeds through the tree portion of the circuit. The area of this circuit is $O(N/\log N)$ and the time is $O(\log N)$. That is, AT is linear. This must be optimal since one should probably at least examine all the bits.

3 The Circuit Model.

3.1 The basic model.

- Each processor can store only one word , ω bits.
- Each processor can read/write one word per time step.
- Each processor can send one word on each link per time step.
- Each input word is read only once.
- The I/O is where/when oblivious.

DEFINITION 1 *INPUT:* $x_1, x_2, \ldots, x_m$
OUTPUT:$y_1, y_2, \ldots, y_n$.
Where oblivous *means that the processor that reads* x_i *(or generates* y_j*) is fixed before execution begins.* When oblivous *means that the time at which* x_i *is read (or* y_i *is generated) is fixed before execution begins.*

4 Simple Lower Bounds.

- **Memory:** $A \geq m$ where there are at last m words that must be stored in the computation.

- **I/O:** $AT \geq f$, where there are at least f words that must be input.
  If the I/O consists of f words, $AT \geq f$ since a chip of area A can read/write at most A words per time step.

- bf Bisection: $AT^2 \geq b^2$, where b is bisection width of circuit.

  Assume that a cut of length $\sqrt{A}$ had divided the chip so that b words need to be communicated the cut in T. Since $\sqrt{A}$ words can be communicated per time step, then $b < T\sqrt{A}$ and the bound follows.

4.1 A useful argument on chip area.

Chip Area: $A \geq \log N/\omega$, where N is the number of different executions after a time t given same input.

DEFINITION 2 *A* chip state *corresponds to a particulare set of values stored in the processors. A* chip behavior *corresponds to a particular set of output values.*

A chip with P processors can have $2^{P\omega}$ chip states, since each processor can store one word (ω bits). Similarly, a chip that outputs n words can have $2^{n\omega}$ behavious. Note that the chip behavior is a function of both the original chip state and the inputs.

Suppose the chip goes through a number of executions after time t, $E_1, \ldots, E_N$ with the same input, e.g., it has already read all the inputs. Then the $N > 2^{Pw}$, and $P \geq \log N/\omega$.

5 Cyclic Shift Lower Bound.

We consider the cyclic shift problem, which is as follows.
INPUT 1: $x_{n-1}, \ldots, x_0$: the data.
INPUT 2: s: shift amount.

OUTPUT: $y_{n-1}, \ldots, y_0$: $y_i = x_{(i+s) \bmod n}$.

LEMMA 1
Cyclic Shift requires $A = \Omega(n)$.

Claim 2
All the inputs must be read before any output word is produced.

Proof: (Proof of claim.) Suppose y_i is generated before x_i is read. By when obliviousness this is true in all executions. If we set $s = i - j \mod n$, then chip must ouptut $y_j = x_{(j+s) \mod n} = x_i$. We can feed different values for x_i after seeing what the chip outputs as y_j. Oops. □

Let t be time when all the data words are read and no output is produced. Consider the outputs with shift value $s = 0$. There are $2^{n\omega}$ different possible outputs, and no input is read. Thus, the circuit must have area $\Omega(\frac{\log 2^{n\omega}}{\omega}) = \Omega(n)$.

Basically, all the input words must be read, and stored before producing an output word.

5.1 Dependence

The above argument can be formalized using the notion of dependence.

Input: $x_1, \ldots, x_n$
Output: $y_1, \ldots, y_n$

Definition 3 *We say y_i depends on x_j if and only if there exists two executions in which distinct values are assigned to input x_j, all the other inputs have the same value in both executions and y_i takes on a different value in the two executions.*

For example, in the cyclic shift example, evey output depends on every input. Dependence of y_i on x_j requires that x_j is read before producing y_i.

Definition 4 *A subset of the outputs Y' f-depends on X' if and only if*

1. *y_i depends on x_j for all $i \in Y'$ and all $j \in X'$.*
2. *There exists f executions in which:*
   (a) *The values of all the inputs not in X' are the same.*
   (b) *The outputs Y' take on distinct values in each execution.*

Theorem 3
If Y' f-depends on X' then the area of any chip must be at least $\log f/\omega$.

Proof:
All X' must be read before any of Y' is generated. After X' is read but before any of Y' is generated the chip will be in at most f possible states. The theorem follows.
□

6 AT^2 lower bounds.

We first make the following claim.

We refer the reader to Ranade's notes from previous years for much of this but do prove the following result for cyclic shift in these notes.

We define the problem of cyclic shift as having inputs $x_0, \ldots, x_{n-1}$ and $s \in [0, \ldots n]$ and outputs $y_1, \ldots, y_{n-1}$ such that $y_i = x_{i+s \mod n}$.

LEMMA 4
The cyclic shift problem requires $AT^2 = \Omega(n^2)$.

The proof proceeds by first noting that if any processor reads or outputs more than $n/3$ values the lemma holds.

Otherwise, we can use the "sliding cut" argument to find a vertical cut (with a jog of length 1) where the number of inputs are balanced, that is no more than $2n/3$ of them are on either side.

Then, we consider the graph between the $\geq n/3$ inputs on one side (say left hand side) and the other side which contains $\geq n/2$ outputs.

Question 1: Without loss of generality, this holds. Why?

The number of edges in this graph is at least $n^2/6$. We color each edge of form (x_i, y_j) with the shift which shifts i onto j. There exists one shift value which has at least $\Omega(n)$ edes by average, i.e., $n^2/6$ edges n shifts, one must color at least $n/6$ edges. There are at least $2^{n/6}$ different behaviors on the right hand side fixing all the input values on the right side. Thus, at least $\Omega(n/6)$ values must cross the sliding cut.

Question 2: Complete the proof. (Pretty easy, hopefully.)