VLSI lower bounds

Lemma: A network with bisection width \( B \) has area \( \Omega(B^2) \).

**Proof (Pf):**

\[ B \leq W + 1 \implies A = LW \geq W^2 \geq (B - 1)^2 = \Omega(B^2) \]

Good recursive bisection \( \Rightarrow \) small area.

Bisection width lower bounds on computation

**Shifting:**

**Input:** \( x_0, x_1, \ldots, x_{n-1} \) data
\[ \text{Control: } S \in \{0, \ldots, n-1\} \]

**Output:** \( y_0, y_1, \ldots, y_{n-1} \) & \( y_0 = (x_{S}\text{mod}n) \)

<table>
<thead>
<tr>
<th>Network</th>
<th>( T(n) )</th>
<th>( B(n) )</th>
<th>( A(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear array</td>
<td>( O(n) )</td>
<td>( \Theta(1) )</td>
<td>( \Theta(n) )</td>
</tr>
<tr>
<td>Tree</td>
<td>( O(n) )</td>
<td>( \Theta(1) )</td>
<td>( \Theta(n) )</td>
</tr>
<tr>
<td>( \sqrt{n} \times \sqrt{n} ) mesh</td>
<td>( O(\sqrt{n}) )</td>
<td>( \Theta(\sqrt{n}) )</td>
<td>( \Theta(n) )</td>
</tr>
<tr>
<td>( n )-input butterfly</td>
<td>( O(\log n) )</td>
<td>( G(n) )</td>
<td>( \Theta(n^2) )</td>
</tr>
</tbody>
</table>

Let \( B(n) \) = \# edges cut to bisection outputs.

If \( T(n) \) = worst-case \# bits to cross this bisection,

then \( B(n)T(n) \geq \Sigma(n) \).

**Ex:**

\( S = 0 \) \( \implies \) 1 bit crosses bisection (L to R)
\( S = 3 \) \( \implies \) 3 bits cross - worst case \( \Rightarrow \Sigma = 3, T = 6/I = 3/2 \).
How do we know 3 bits must cross?  
Might there be a clever encoding?  

"Fooling argument"  
Ex. Suppose we have following bijection:  

\[
\begin{array}{c|c|c|c}
X_0 & Y_0 & X_{n/2} & Y_{n/2} \\
X_1 & Y_1 & \vdots & \vdots \\
X_{n/2-1} & Y_{n/2-1} & X_n & Y_n \\
\end{array}
\]

\[s = n/2 \Rightarrow \text{intuitively } n/2 \text{ bits must cross } (L \rightarrow R)\]

Claim: \(B(n) \cdot T(n) \geq n/2\).

PF. (fooling arg.) Suppose \(B(n) \cdot T(n) < n/2\).

\[\text{# communication patterns on } B(n) \text{ wires } (L \rightarrow R)\]
\[\text{over time } T(n) = 2^{B(n) \cdot T(n)} < 2^{n/2}\]

\[\text{# values for } x_0, \ldots, x_{n/2-1} = 2^{n/2}\]

\[\therefore 2 \text{ distinct } x_0', \ldots, x_{n/2-1} \text{ and } x_0'', \ldots, x_{n/2-1}'' \]

that produce identical communication patterns.

RTS of circuit can't distinguish \(\Rightarrow\) produces
same values for \(X_{n/2}, \ldots, X_n\) for both \(\Rightarrow\) must
operate wrong for one, contradiction. \(\Box\)
**Thm.** For any bisection of outputs, $B(n)T(n) = n/2$.

**Pf.** Consider arb. bisection

<table>
<thead>
<tr>
<th>Y_0</th>
<th>x_0</th>
<th>Y_1</th>
<th>x_1</th>
<th>Y_2</th>
<th>x_2</th>
<th>Y_3</th>
<th>x_3</th>
<th>Y_4</th>
<th>x_4</th>
<th>Y_5</th>
<th>x_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>2</td>
<td></td>
<td>3</td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

Make an $n \times n$ table:

<table>
<thead>
<tr>
<th></th>
<th>x_0</th>
<th>x_1</th>
<th>x_2</th>
<th>x_3</th>
<th>x_4</th>
<th>x_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

X if shift of s causes $x_i$ to cross bisection

Every column contains $n/2$ X's.

*: Average # X's per row = n/2.

=> some row contains $\geq n/2$ X's.

(some shift causes n/2 bits to cross.) **

<table>
<thead>
<tr>
<th>Network</th>
<th>B(n)</th>
<th>T(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear array</td>
<td>$\Theta(1)$</td>
<td>$\Omega(n)$</td>
</tr>
<tr>
<td>Tree</td>
<td>$\Theta(1)$</td>
<td>$\Omega(n)$</td>
</tr>
<tr>
<td>$\sqrt{n} \times \sqrt{n}$ mesh</td>
<td>$\Theta(\sqrt{n})$</td>
<td>$\Omega(\sqrt{n})$</td>
</tr>
<tr>
<td>n-input butterfly</td>
<td>$\Theta(n)$</td>
<td>$\Omega(1)$</td>
</tr>
</tbody>
</table>
Theorem: Any circuit for shifting $n$ bits has $AT^2 = \Omega(n^2)$.

Proof: $AT^2 = \Omega(b^2T^2)$

$= \Omega(n^2) \quad \square$

Theorem: Any circuit for multiplying two $n$-bit #s has $AT^2 = \Omega(n^2)$.

Proof: $c = ab$. Let $b$ take on powers of 2.

Essentially corresponds to shift problem.

Consider bisection of outputs.

Build shift/input matrix as before: $n \times n/2$

\[
\begin{array}{cccc}
0 & 1 & \ldots & n-1 \\
\hline
0 & x & x & \ldots & x \\
1 & x & x & \ldots & x \\
\vdots & & & & \\
\end{array}
\]

$n/4 \times 5$'s per column

$n^2/8 \times 5$'s in matrix

$\Rightarrow$ some row has $n/8 \times 5$'s

$\therefore AT \geq \Omega(n^2) \quad \square$

Also, FFT, convolution, sorting, routing, etc.

General Layout Strategy

Tree of meshes (not mesh of trees)

\[ T(n) = \Omega(n) \]

\[ N = n^2 \log n \]

\[ n^2 \text{ leaves} \]

\[ S(n) = 2S(n/2) + n \]

\[ = \Theta(n \log n) \]

\[ A(n) = \Theta(n^2 \log^2 n) \]
Fold and squash:

\[ n^2 \times \log n \text{ layers } \rightarrow \Theta(n^2 \log^2 n) \text{ area.} \]

Truncated TOM: TOM(n, k) - top k levels

Area = \( \Theta(n^2 k^2) \)

Decomposition trees

T is a \((w_1, w_2, \ldots, w_r)\) decomposition tree for \(G=(V, E)\):

1. Vertices in \(V\) mapped to leaves of \(T\).
2. Edges in \(E\) run through links of \(T\).
3. \#edges leaving subtree rooted at depth \(i\) is \(\leq w_i\)

For \(1 < \alpha \leq 2\), \(G\) has a \((w, \alpha)\) decomptree if it has a \((w, w/\alpha, w/\alpha^2, \ldots, 0)\) decomptree.

A decomptree is balanced if all subgraphs at the same depth have same \# vertices to within 1.
Layout strategy

1. Start with \((w, \sqrt{2})\) decomp tree.
2. Balance the decomp tree.
3. Embed the balanced tree in trunc TOM.
4. Use trunc TOM layout to yield \(O(w^2 \log^2 n)\) area layout.

Balancing decomp trees

Warm-up: Necklace with black and white pearls.
How many cuts to divide into 2 sets, each with half the pearls of each color?

2 cuts suffice.
Continuity argument.

Lemma: Consider any 2 strings composed of an even # of black pearls and an even # of white pearls. By making at most 2 cuts, the pearls can be partitioned into 2 sets, each containing 2 strings, such that each set has \(1/2\) the pearls of each color.

Pf. (Continuity arg.)
Lemma. Let $T$ be a CBT drawn with $n$ leaves on a straight line, and consider any set $S$ of $k$ consecutive leaves of $T$. Then, there is a forest $F$ of complete binary subtrees of $T$ such that:

1. $S = \{\text{leaves of } F\}$
2. at most 2 trees of $F$ have any given height.
3. depth of largest tree in $F$ is $\leq \log k$.

Proof. Let $F$ be a forest of maximal CBT's whose leaves lie only in $S$. (1) and (3) follow. Use induction to prove (2). □

Theorem. Let $G$ be a graph on $n$ vertices that has a $(w_1, w_2, ..., w_r)$ decomposable tree $T$. Then, $G$ has a $(w_1', w_2', ..., w_{r+1})$ balanced decomposable tree $T'$, where

$$w_i' = \frac{1}{r} \sum_{k=i}^{r} w_k.$$

Proof. Color leaves of $T$: 1 = node of $G$, 0 = empty.

Recursively split $B & W$ leaves evenly. Each stage has $\leq 2$ strings of consecutive leaves from $T$, each of which has $\leq 2$ CBT's of a given height.
Total # wires leaving a string
\leq \text{sum of wires leaving each of its cbrs}.

\[ w_i \leq 4 \sum_{k=i}^{r} w_k. \]

**Corollary.** A graph with a \((w, \alpha)\) decompc tree, \(\alpha\) const, has an \(O(w), \alpha)\) balanced decompc tree.

**Pr.** Sum is geometric:

\[ w_i' = 4 \sum_{k=i}^{r} w_k \leq 4 \sum_{k=i}^{r} \frac{w}{\alpha} \leq 4 \frac{w}{\alpha^{i-1}} \leq 4w \frac{\alpha}{\alpha^{i-1}} (\frac{\alpha}{\alpha - 1}). \]

Graph has \((4w\alpha / (\alpha - 1)), \alpha)\) decompc tree.

Next week: Embed in trunc. TM \( \Rightarrow \) layout.
Area-universal networks.

<<Exam issues>>