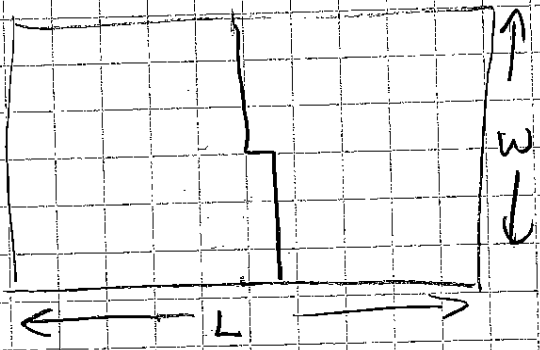
6.896
4/26/04
L19.1

VLSI lower bounds

<u>Lemma</u> A network with bisection width B has area $\Omega(B^2)$.

Pf.



$$B \le W+1 \Rightarrow A = LW \ge W^2 \ge (B-1)^2 = \Omega(B^2) \quad \boxtimes$$

Good recursive bisection ⇒ small area.

Bisection width lower bounds on computation

<u>Shifting</u>

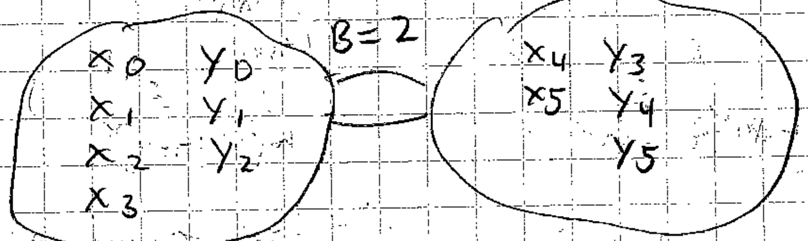
Input: $x_0, x_1, \ldots, x_{n-1}$ data
$s \in \{0, \ldots, n-1\}$ control

Output: $y_0, y_1, \ldots, y_{n-1}$ \$ $y_i = x_{(i-s) \bmod n}$

Network	T(n)	B(n)	A(n)
Linear array	$O(n)$	$\Theta(1)$	$\Theta(n)$
Tree	$O(n)$	$\Theta(1)$	$\Theta(n)$
$\sqrt{n} \times \sqrt{n}$ mesh	$O(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(n)$
n-input butterfly	$O(\lg n)$	$\Theta(n)$	$\Theta(n^2)$

Let $B(n)$ = # edges cut to bisect outputs
If $I(n)$ = worst-case # bits to cross this bisection, then $B(n)T(n) \ge I(n)$.

<u>Ex.</u>



$s=0 \Rightarrow$ 1 bit crosses bisection (L to R)
$s=3 \Rightarrow$ 3 bits cross — worst case $\Rightarrow I=3, T \ge B/I = 3/2$.

How do we know 3 bits must cross?
Might there be a clever encoding?

"Fooling argument"

Ex. Sup. we have following bisection:

$$\begin{array}{cc|cc} x_0 & y_0 & x_{n/2} & y_{n/2} \\ x_1 & y_1 & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ x_{n/2-1} & y_{n/2-1} & x_n & y_n \end{array}$$

$s = n/2 \Rightarrow$ intuitively $n/2$ bits must cross (L to R)

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

Pf. (fooling arg.) Sup. $B(n)T(n) < n/2$

\# communication patterns on $B(n)$ wires (L to R) over time $T(n) = 2^{B(n)T(n)} < 2^{n/2}$

\# values for $x_0, \ldots, x_{n/2-1} = 2^{n/2}$.

$\therefore \exists$ 2 distinct $x_0', \ldots, x_{n/2-1}'$ and $x_0'', \ldots, x_{n/2-1}''$ that produce identical comm patterns.

RHS of circuit can't distinguish $\Rightarrow$ produces same values for $y_{n/2}, \ldots, y_n$ for both $\Rightarrow$ must operate wrong for one. Contradiction. $\boxtimes$

<u>Thm.</u> For <u>any</u> bisection of outputs, $B(n)T(n) \geq n/2$

Pf. Consider arb. bisection

<u>Ex.</u>

y_0 y_2 y_4 x_0 x_1 x_2 x_4 x_5	y_1 y_3 y_5 x_3

Make an $n \times n$ table:

s	x_0	x_1	x_2	x_3	x_4	x_5
0		X				X
1	X		X	X	X	
2		X				X
3	X		X	X	X	
4		X				X
5	X		X	X	X	

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) ☒

Network	$B(n)$	$T(n)$
Linear array	$\Theta(1)$	$\Omega(n)$
Tree	$\Theta(1)$	$\Omega(n)$
$\sqrt{n} \times \sqrt{n}$ mesh	$\Theta(\sqrt{n})$	$\Omega(\sqrt{n})$
n-input butterfly	$\Theta(n)$	$\Omega(1)$

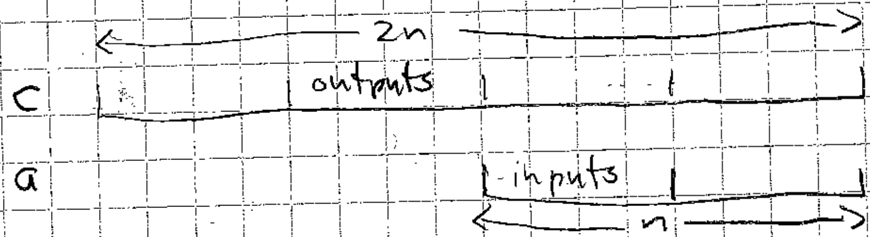
6.896
4/26/04
L19.4

Theorem Any circuit for shifting n bits has $AT^2 = \Omega(n^2)$

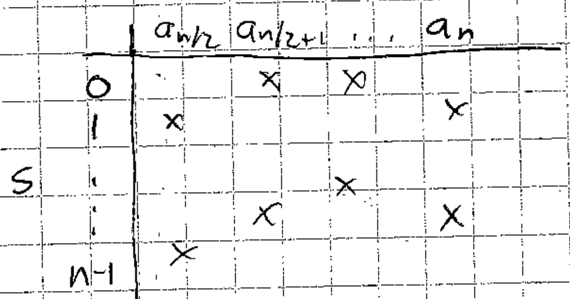
Pf. $AT^2 = \Omega(B^2T^2)$
$= \Omega(n^2)$ $\boxtimes$

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

Pf. $c = ab$. Let b take on powers of 2.
Essentially corresp. to shift problem.



Consider bisection of outputs.
Build shift/input matrix as before: $n \times n/2$



$n/4$ x's per column

$n^2/8$ x's in matrix
$\Rightarrow$ some row has $n/8$ x's.
$\therefore BT \geq \Omega(n)$ $\boxtimes$

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

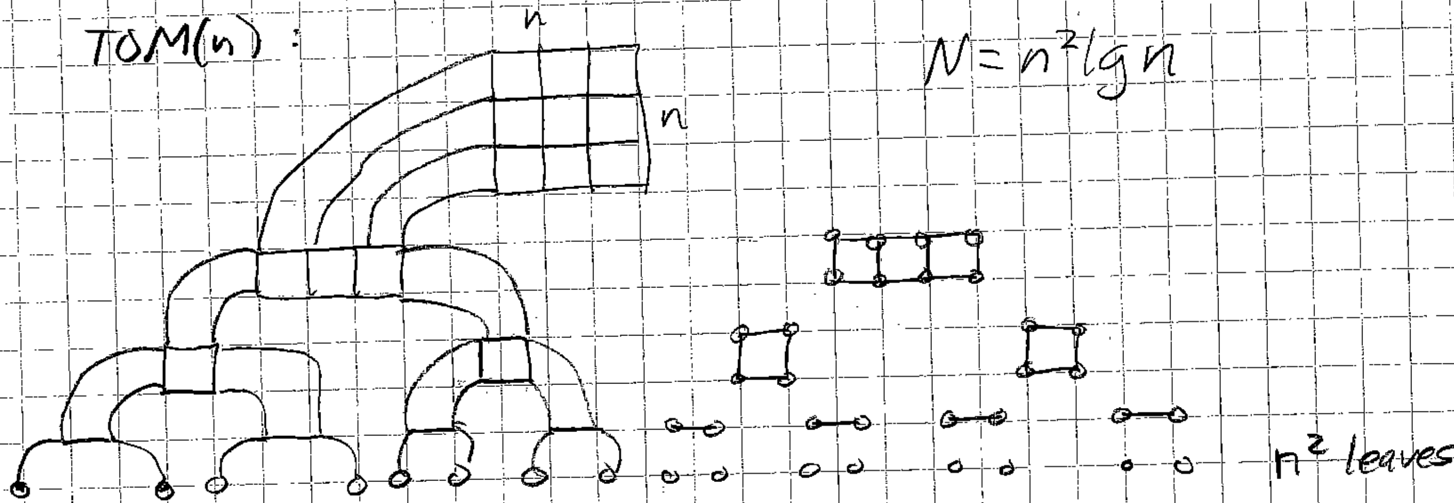
Exam: In class on May 12. Prob. session May 10.

6.896
4/28/04
L20.1

General Layout Strategy

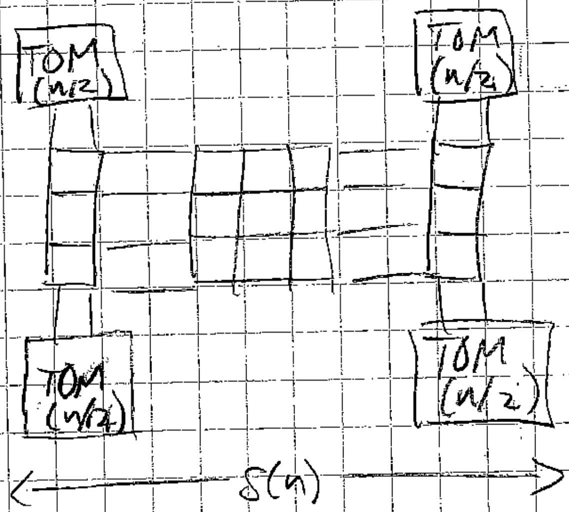
Tree of meshes (not mesh of trees)

TOM(n) :



$N = n^2 \lg n$

Area:

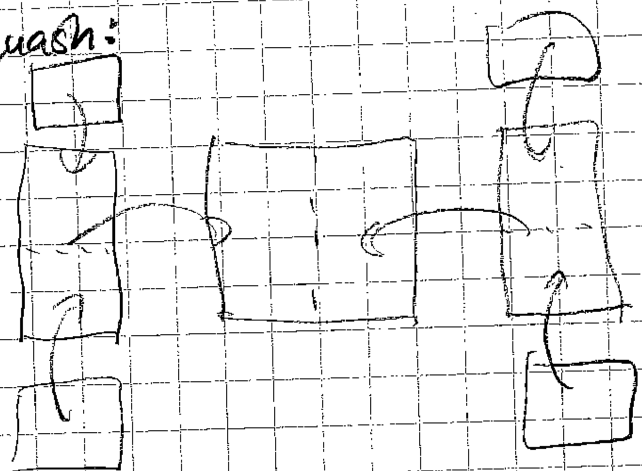


$$S(n) = 2S(n/2) + n$$
$$= \Theta(n \lg n)$$

$$A(n) = \Theta(n^2 \lg^2 n)$$

6.896
4/28/04
L20.2

Fold and squash:



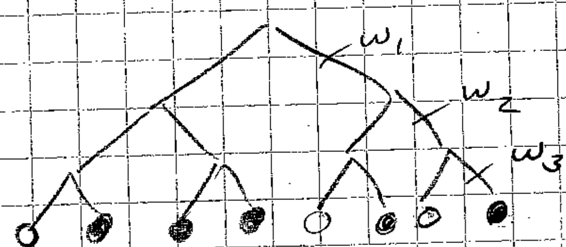
$n^2 \times \lg n$ layers $\xrightarrow{\text{squash}} \Theta(n^2 \lg^2 n)$ area.

Truncated TOM: TOM(n,k) – top k levels.
Area $= \Theta(n^2 k^2)$

__Decomposition trees__

T is a $\langle w_1, w_2, \ldots, w_r \rangle$ __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 $\le w_i$



For $1 < \alpha \le 2$, G has a (w, α) decomp tree if it has a $\langle w, w/\alpha, w/\alpha^2, \ldots, O(1) \rangle$ decomp tree.

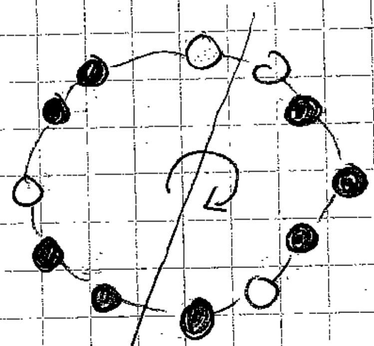
A decomp tree is __balanced__ if all subgraphs at the same depth have same # vertices to within 1.

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 $O(w^2 \lg^2 n)$ area layout.

Balancing decomp trees

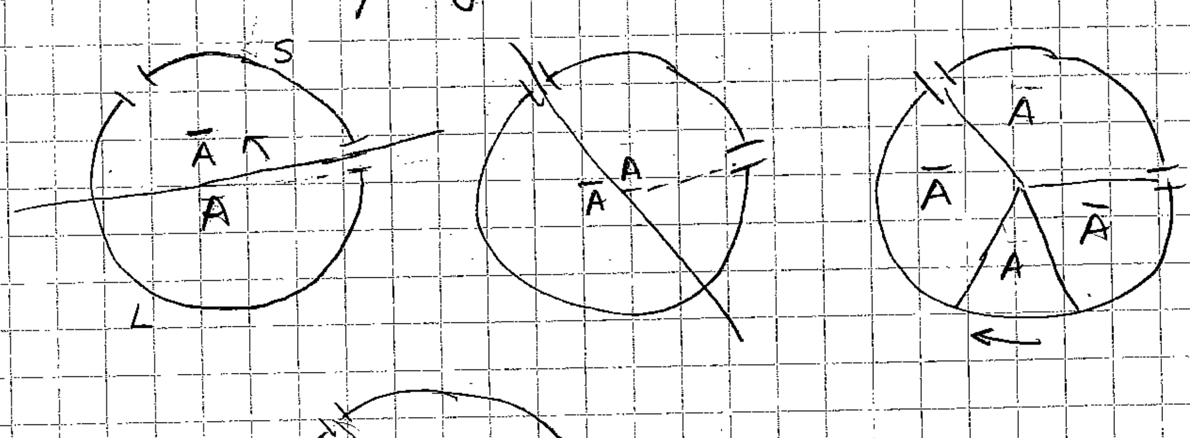
Warm-up: Necklace with black and white pearls. How many cuts to divide into 2 sets, each with half the pearls of each color?

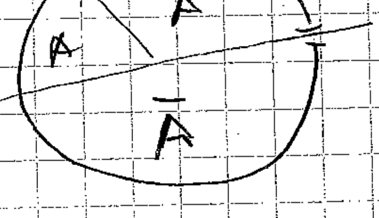


2 cuts suffice.
Continuity argument.

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.)

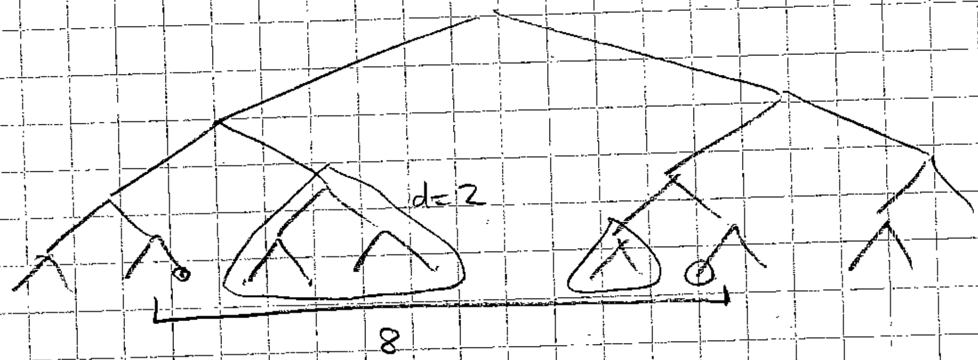




$\boxtimes$

Lemma. Let T be cbt drawn with n leaves on a straight line, and consider any set S of k consecutive leaves of T. Then, $\exists$ a forest F of complete binary subtrees of T $\ni$

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 $\le \lg k$.

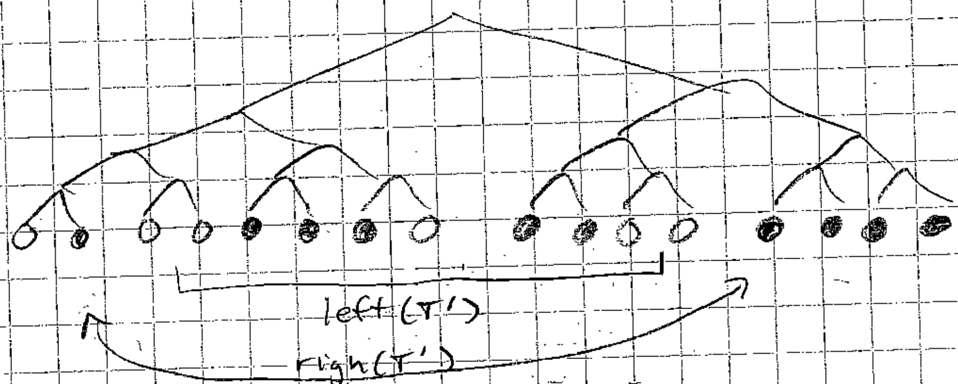


Pf. F be forest of maximal cbt's whose leaves lie only in S. (1) & (3) follow. Use induction to prove (2). ⊠

Thm. Let G be a graph on n vertices that has a $\langle w_1, w_2, \ldots, w_r\rangle$ decomp tree T. Then, G has a $\langle w_1', w_2', \ldots, w_{\lceil \lg n\rceil}'\rangle$ balanced decomp tree T', where

$$w_i' = 4 \sum_{k=i}^{r} w_k .$$

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



Recursively split B&W leaves evenly. Each stage has ≤ 2 strings of consec. leaves from T, each of which has ≤ 2 cbt's of a given height.

6.896
4/28/04
L20.5

Total # wires leaving a string
$\le$ sum of wires leaving each of its cbt's.

$$w_i' \le 4 \sum_{k=i}^{r} w_k. \quad \boxtimes$$

Corollary A graph with a (w, α) decomp tree, α const, has an $(O(w), \alpha)$ balanced decomp tree.

Pf. Sum is geometric:

$$w_i' = 4 \sum_{k=i}^{r} w_k$$

$$\le 4 \sum_{k=i}^{r} \frac{w}{\alpha^{k-1}}$$

$$\le \frac{4w}{\alpha^{i-1}} \left(\frac{\alpha}{\alpha - 1} \right).$$

Graph has $(4w\alpha/(\alpha-1), \alpha)$ decomp tree. $\boxtimes$

Next week: Embed in trunc TOM $\Rightarrow$ layout.
Area-universal networks.

«Exam issues»