Solution Set 9

Due: In class on Wednesday, April 28. Starred problems are optional.

Problem 9-1. Show that any VLSI layout can be transformed into a nearly square layout. That is, even a long skinny layout can be folded to be close to square.

Solution:

Lemma 1*Any VLSI layout can be transformed into a (nearly) square layout with an increase of no more than a constant factor to area.*

Proof. Suppose that the bounding box of the original circuit is $m \times n$, where $m \leq n$. Then we can fold over the original circuit (like an accordion) at intervals of $\sqrt{mn}$ on the long edge. Thus, we fold over the circuit into $\lceil n/\sqrt{mn} \rceil$ sections to fit inside a bounding box that is $\sqrt{mn} \times (m \cdot \lceil n/\sqrt{mn} \rceil)$, which is no larger than a box that is $\sqrt{mn} \times (\sqrt{mn} + m)$.

Next, we must take into account the extra wires added due to the folds. Since the width of the circuit at the folds is only m, there can only be m wires crossing the fold. Thus, we can fit the circuit with the new wires inside a box that is $(\sqrt{mn} + 2m) \times (\sqrt{mn} + m)$[1], which fits inside a square box with dimensions $(\sqrt{mn} + 2m) \times (\sqrt{mn} + 2m)$. The total area is thus less than $(\sqrt{mn} + 2m)^2 = mn + 4m\sqrt{mn} + 4m^2$. Since $m < n$, we have that this area is less than $mn + 4mn + 4mn = 9mn$, and the total area of the circuit did not increase by more than the constant factor of 9.

Problem 9-2. Show that for any VLSI layout of a complete binary tree with all the leaves in a straight line, that the total wire area is $\Omega(n \log n)$.

Solution:

Lemma 2*For any VLSI layout of a complete binary tree with n-leaves in a straight line, the wire area of the longest*[2] *root-to-leaf path is at least $n/2 - 1$.*

Proof. This lemma is fairly obvious. Since the leaves are in a straight line, the distance between the leftmost and rightmost leaves must be at least $n - 1$. By the triangle inequality, one of the root-to-leaf paths (just considering the leftmost and rightmost leaves) is at least $(n-1)/2 > n/2 - 1$.

Theorem 3*For any VLSI layout of a complete binary tree with n-leaves in a straight line, the total wire area is $\Omega(n \log n)$.*

Proof. We can prove this lemma with a simple recurrence. Consider any VLSI layout of a complete binary tree with n-leaves in a straight line. We let $A(n)$ be the minimum total wire area, not counting the longest root-to-leaf path in the tree, of any such layout. Then we will show that the recurrence is $A(n) \geq 2A(n/2) + n/4 - 1 = 2A(n/2) + O(n)$, which can be solved with the master method to get $A(n) = \Omega(n \log n)$.

Consider any complete subtree T with k leaves in the VLSI layout. The tree T is composed of two $k/2$-leaf subtrees, T_1 and T_2, with a root node connecting to the roots of T_1 and T_2. Without loss of generality, the longest root-to-leaf path in T includes the longest root-to-leaf path in T_1.[3] Let l_2 be the length of the longest root-to-leaf path in T_2. Let

[1] in the most unbalanced case, where $\lceil n/\sqrt{mn} \rceil = n/\sqrt{mn}$, we have a rectangle that is $\left(\sqrt{mn} + 2m\right) \times \sqrt{mn}$. This is "nearly square" because $m < n$, and hence $m < \sqrt{mn}$, so the longer dimension is at most 3 times the shorter one.

[2] ties are broken arbitrarily

[3] The longest root-to-leaf path has to pass through the roots of one of the subtrees, so it must include the root-to-leaf path of one of the subtrees

A_1 and A_2 be the total wire area of each of the subtrees T_1 and T_2, respectively, not counting the longest root-to-leaf paths. Then the total wire area A of T not counting the longest root-to-leaf path is given by $A \geq A_1 + A_2 + l_2$. Since T_1 and T_2 have $k/2$ leaves, $T_1 \geq A(k/2)$, and $T_2 \geq A(k/2)$. Thus, $A \geq 2A(k/2) + l_2$. Applying Lemma 2, we get $A(k) \geq 2A(k/2) + k/4 - 1$.

Note that the above proof does not depend on the ordering of leaves in the VLSI layout. That is to say, the leaves of the two $n/2$-leaf subtrees of an n-leaf VLSI layout can be interleaved and reordered arbitrarily without affecting correctness of the proof.

Problem 9-3. Show that any binary tree with an even number of nodes can be cut exactly in half by cutting $O(\log n)$ edges. What is the constant?

Solution: That a binary tree with an even number of nodes can be cut exactly in half by cutting $O(\log n)$ edges follows directly from lecture on April 21. We have the following two lemmas from class:

Lemma 4*Binary trees are* 1*-separable.* □

Lemma 5*If a graph* G *is* S*-separable, then* G *is strongly* Γ_s*-separable.* □

In class, we defined

$$\Gamma_s(n) = \sum_{i=0}^{\lceil \log_{3/2} n \rceil} S\left(\left(\frac{2}{3}\right)^i n\right) .$$

By Lemma 4, we have that $S(n) = 1$ for a binary tree and any n with $1 \leq n$. Thus, $\Gamma_s(n) = O(\log n)$, and a binary tree is strongly $O(\log n)$-separable, meaning that it can be cut exactly in half by cutting $O(\log n)$ edges.

Now, we consider the constants. We have that $\Gamma_s(n) = \sum_{i=0}^{\lceil \log_{3/2} n \rceil} 1 = \left\lceil \log_{3/2} n \right\rceil$. So the binary tree can be cut in half by cutting at most $\left\lceil \log_{3/2} n \right\rceil$ edges. We can compute the appropriate constants by changing the base of the log.

For a binary tree, there is also a simple more direct[4] approach. Basically, i claim that for any binary tree with n nodes, and a given number k, with $k \leq n$, I can cut off a subtree with between $k/2$ and k nodes with a single cut.

Lemma 6*Given a binary tree with* n *nodes and an input* k*, with* $1 \leq k < n$*, there is a cut of a single edge that cuts off between* $k/2$ *and* k *nodes.*

Proof. Consider the depth (distance from root) of every node in the tree. Iterate through all the nodes in order of depth starting from the deepest node, calculating the size of the full subtree rooted at each node.[5] Stop once a subtree with size $k/2$ to k is found. Cut the single edge connecting this subtree to the rest of the tree.

The only point to prove is that we will always find a subtree of the correct size. Assume for the sake of contradiction that the smallest subtree with a size at least $k/2$ that we can find is rooted at a node v and has a size of $l > k$. If v is a leaf, then we are done, because the size of the tree rooted at v is 1 (and $1 \not> k$). If v has 1 child v_1, then the subtree rooted at v_1 has size $l - 1$, which is better. Suppose instead that v has two children v_1 and v_2. Then the size of v is one more than the sum of the sizes of v_1 and v_2. Thus, the sum of the sizes of the two subtrees is at most k, and it follows that one of the subtrees must be at least $k/2$, which generates a contradiction.

[4] not relying on what we did in class, but similar
[5] For example, a leaf is a subtree of size 1. A node with two leaves as children is the root of subtree with size 3.

Using Lemma 6, we can cut a binary tree exactly in half by cutting at most $\lg n$ edges. For example, in the first cut, we ask for a subtree of size $n/2$. In the worst case, we only get a subtree of size $n/4$, and we still need to cut off $n/4$ nodes. Thus, we ask for a subtree of size $n/4$ on the next iteration and may be unlucky again, getting a subtree of size $n/8$. In the worst case, our cuts produce subtrees of size $\lceil n/4 \rceil + \lceil n/8 \rceil + \lceil n/16 \rceil + \ldots$, requiring $\lg n$ cuts to cut the tree exactly in half.

Problem 9-4. Show that there is a layout for cube-connected-cycles network (or a butterfly) with $n = k2^k$ vertices with only $O(n^2/\log^2 n)$ area.

Solution: We proved the following lemma in class on April 14.

Lemma 7 *There is a layout for an n-input butterfy with only $O(n^2)$ area.* □

Now, we show the following corollary, which is just a rewrite of the above lemma.

Corollary 8 *There is a layout for a cube-connected-cycles network (or butterfly) with $n = k2^k$ vertices with only $O(n^2/\log^2 n)$ area.*

Proof. A butterfly with $n = k2^k$ vertices is a butterfly with $m = 2^k$ inputs. So, applying Lemma 7, there is a layout that takes $O(m^2) = O((2^k)^2)$ space. We also have that $\lg n = \lg k + k$, or $k = \lg n - \lg k$. Since $n > k$, we have $k = \Theta(\lg n - \lg \lg n)$. Thus, we have that the area is $O((2^{\lg n - \lg \lg n})^2) = O((2^{\lg n}/2^{\lg \lg n})^2)$, which reduces to $O(n^2/\log^2 n)$.

Problem 9-5. * Show that the minimum dimension of any layout of a complete binary tree is $\Omega(\log n)$.

Solution: Assume n is a power of two; this is enough because we are arguing a lower bound (so we must show an infinite family of counterexamples). Assume the minimum dimension is $w = o(\lg n)$. Pick a subset S of the nodes, and a number k. The a k-subset of S can be isolated from the remaining nodes in S by cutting at most $w + 1$ edges — this was done in class. Now let S be the leaves of the tree, and $k = 222\ldots$ in base 4, having $\lg n/2$ digits. To isolate k leaves from the rest requires $\Omega(\lg n)$ cuts. This is a bit tricky to prove.

So, consider some cuts, ordered by level; consider first the cuts at higher levels (closer to the root). This means that every cut isolates a subtree that has a power-of-two number of leaves. The cuts so far split the tree into some parts. Take an arbitrary subset of the parts, and sum up their number of leaves. I claim that after k cuts, no such sum can have more than $2k + 1$ alternations between 0 and 1 when read in binary. Initially, $k = 0$, and there is just one size, a power of two, so it works. In the end, some sum yeilds $222\ldots$ in base 4, so it has $\Omega(\lg n)$ alternations, so we must have made $\Omega(\lg n)$ cuts. It remains to prove the inductive step. The new cut generates two new sizes a and b (say $a \geq b$). Since cuts are considered from higher levels to lower levels, b is a power of two (it cuts an untouched subtree). For subsets which include both a, b, or none, nothing changes because these subsets existed previously. Consider now a set S including b. The sum for $S \setminus b$ has at most $2(k - 1) + 1$ alternations, and we add a power of two, which makes the number of alternations go up by at most 2, which is good. Now consider a set S including a. Previously, $S \setminus a \cup \{a + b\}$ has at most $2(k - 1) + 1$ alternations, and we subtract a power of two (i.e. b), so again the number of alternations goes up by at most 2, so we're done.