Solution Set 10

Due: In class on Wednesday, May 5. Starred problems are optional.

Problem 10-1. Use any drawing program[1] you please to draw 2-layer VLSI layouts of the following networks:

(a) complete binary tree on 256 leaves,

(b) butterfly on 64 inputs,

(c) 16×16 mesh of trees,

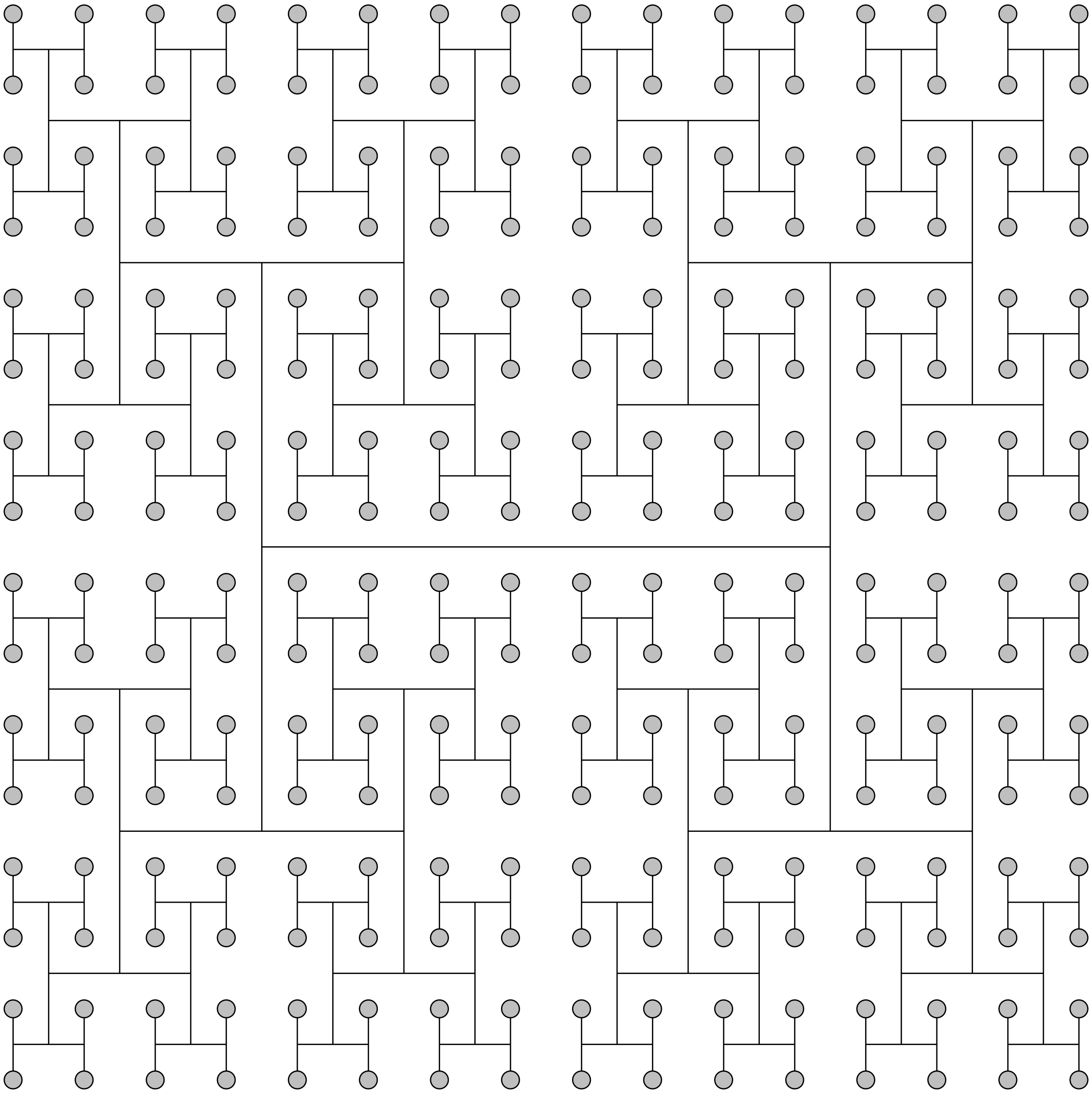
(d) 8×8 tree of meshes using divide-and-conquer,

(e) 8×8 tree of meshes using fold-and-squash.

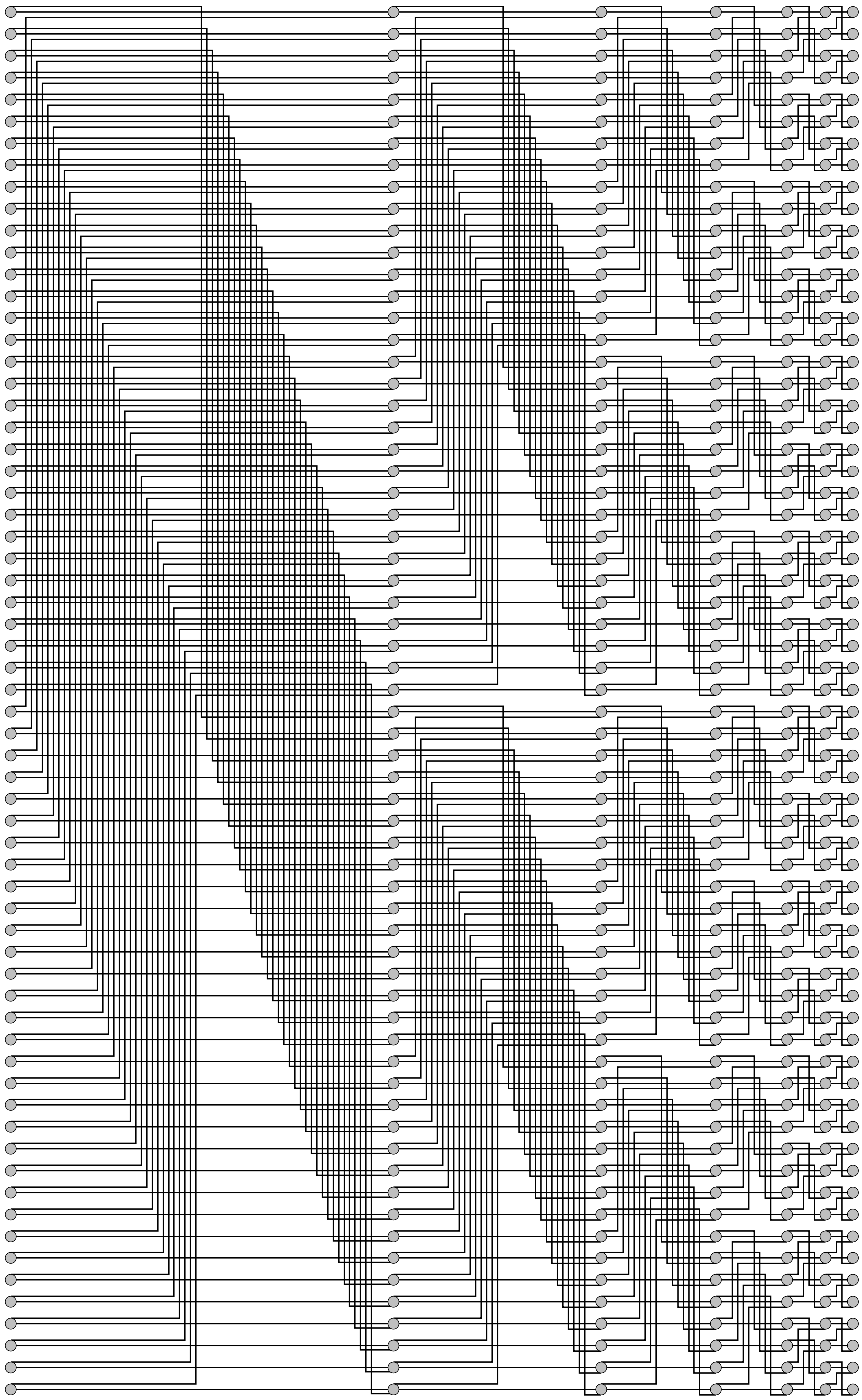
Solution: See figures.

[1] *Suggested drawing programs:* xfig, dia (Linux); PowerPoint, Illustrator, SmartDraw, Visio (Windows); ClarisDraw (Macintosh). If you're more ambitious, write programs outputting PostScript, for example, to draw the graphs.

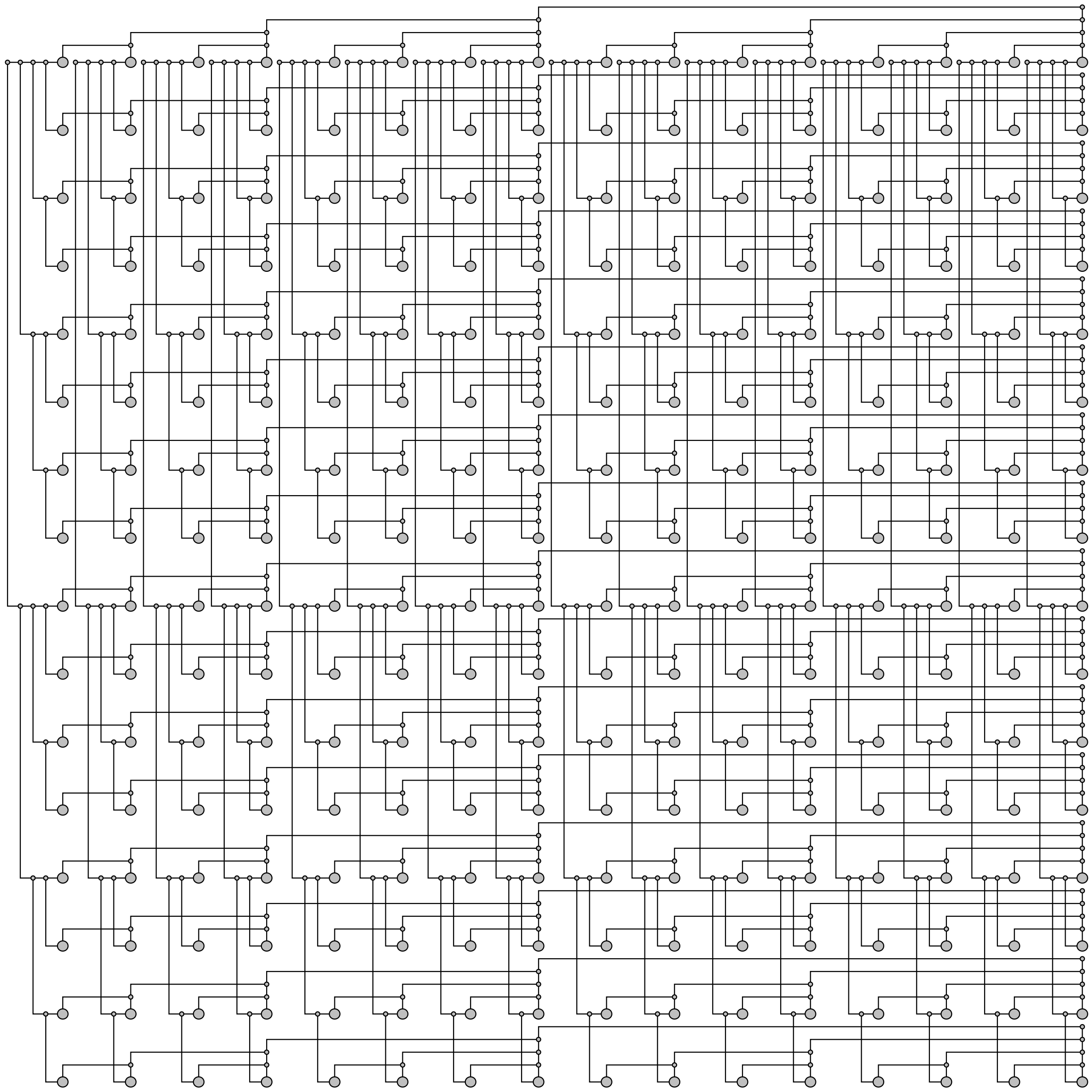
(a)



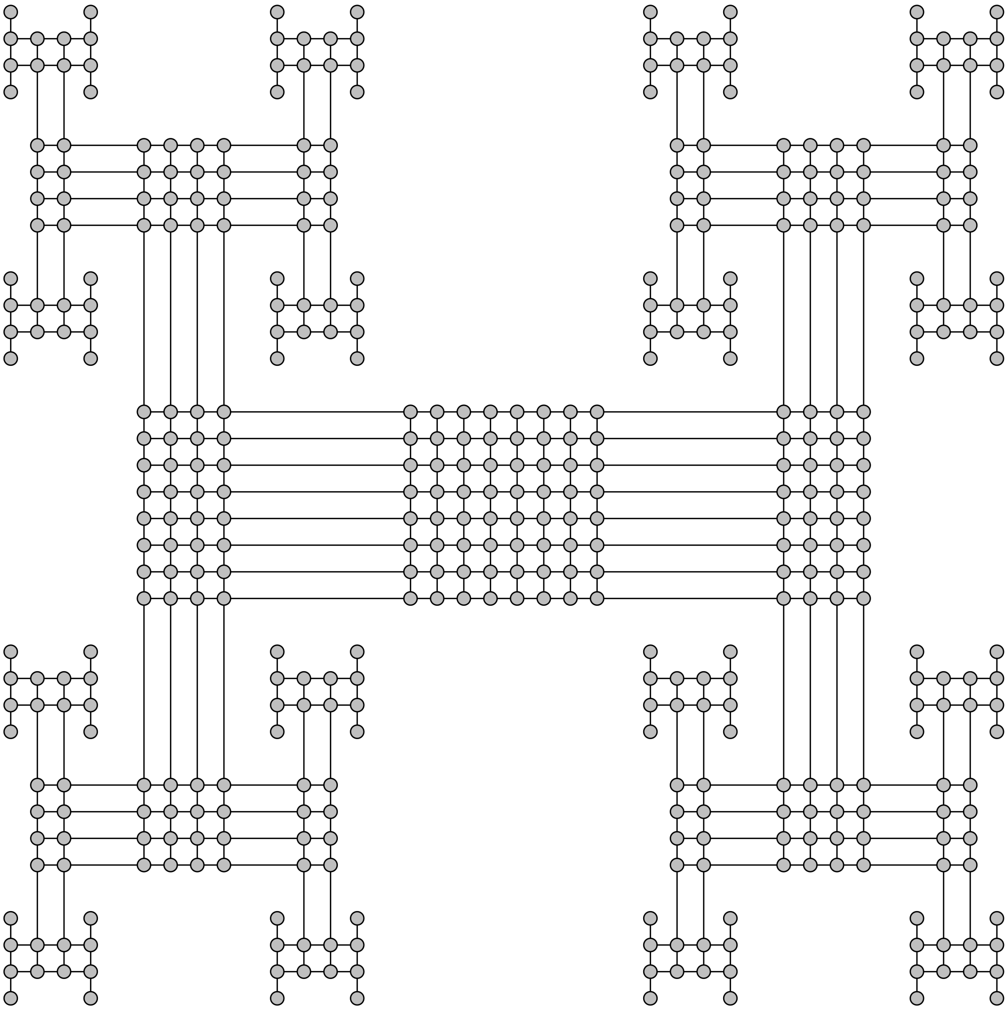
(b)



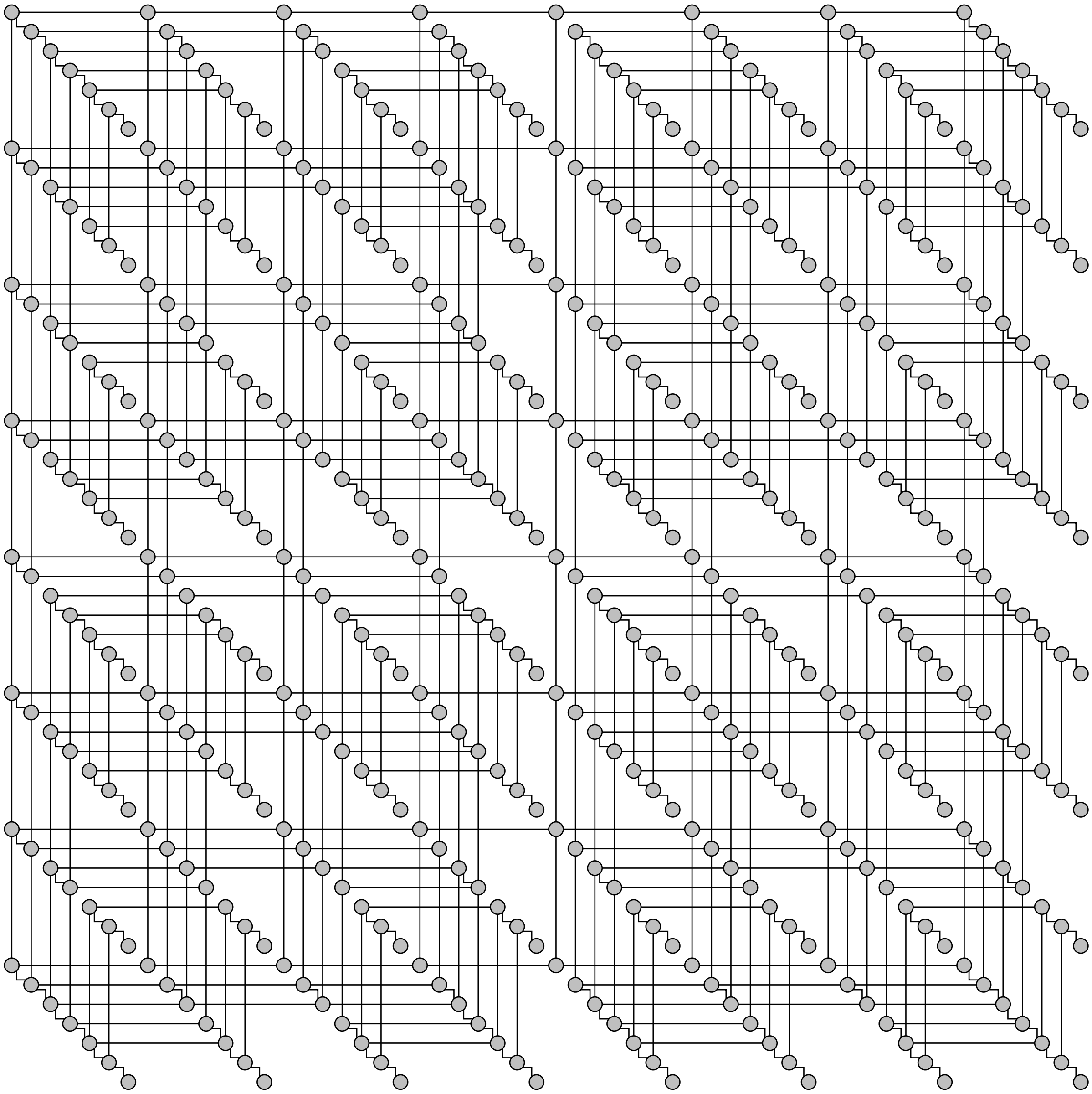
(c)



(d)



(e)



Wires should be rectilinear on a grid, and your layouts should have asymptotically optimal area.

Problem 10-2. Let V be the volume of a 3-dimensional layout of a circuit to solve the circular-shifting problem on inputs of size n, and let T be the worst-case time of the circuit. Prove an analogous result to $AT^2 = \Omega(n^2)$, but with volume instead of area.

Solution: Assume the circuit fits in a 3D box of volume V. First I'll prove that the bisection width for such a circuit is $O(V^{2/3})$. The largest dimension must be $\Omega(V^{1/3})$. Sweep a plane perpendicular to this direction. At the beginning there are 0 nodes on the left, and at the end there are 0 nodes on the right, so at some point we switch from more on the left to more on the right. If we cut all edges crossing these consecutive positions of the plane, and also all edges that fit between these two, we will certainly have bisected the network. The numer of edges cut is proportional to the area, which must be $O(V^{2/3})$ because the other dimension is $\Omega(V^{1/3})$. We proved in class that the number of bits going across any cut bisecting the inputs for the cyclic shift problem is $\Omega(n)$. Hence, $O(V^{2/3}) \cdot T = \Omega(n) \Rightarrow V \cdot T^{3/2} = \Omega(n^{3/2})$.

Problem 10-3. * Consider a ***tree machine*** consisting of processors interconnected as a complete binary tree. We wish to package such a machine into VLSI chips, where each chip uses as few pins (I/O connections) as possible. Suppose that a chip can hold up to M processors. Show that a tree machine of size $N \geq M$ can be assembled from $O(N/M)$ identical VLSI chips, each having $O(1)$ pins.

Solution: Since each chip has M processors, these must expect $\Omega(M)$ input bits per cycle (because at most $M/2$ of the processors can feed only from other processors on the same chip). But if there are only $O(1)$ pins, the input takes $\Omega(M)$ cycles to come in, so when we solve the problem, an $\Omega(M)$ slowdown is inevitable.

Now, an M-processor chip can fit a complete binary tree of height $\lfloor \lg M \rfloor$. We can embed our complete binary tree with N nodes into a B-tree, with $B = \Theta(M)$ (basically, M rounded down to the highest power of two smaller than it). Then each node does the work of $O(M)$ processors, so we can simulate every node of the B-tree with one of our chips. The only trouble is that we have $\omega(1)$ indegree. But we can link the children of a node in a linear array, so each passes messages destined to the parent to the left sibiling, and the leftmost sibiling passes them up to the parent. All chips have 3 connections: the left sibiling (pass messages), the right sibiling (receive messages to relay) and the leftmost child (messages from children). With an $O(M)$ slowdown, everything can be routed without problem; the slowdown is needed because a message from the rightmost child takes $O(M)$ hops to get to the parent.