

**California Institute of Technology
Department of Computer Science
Electronic Design Automation**

CS286.5b, Winter 2000

Final

Monday, March 13

Due: Wednesday, March 15, 11:59PM.

Instructions

- There is no limitation on notes, literature, or books you may use for reference. If your solution is based heavily upon or inspired by some existing work, provide an appropriate reference.
- Work independently. You may not consult other people as references.
- There is no time limit, other than that you start after 12:01AM Monday and finish and turn it in before 11:59PM Wed.
- Please writeup in some electronic form and turn in.

Problems

1. Explain the freedom exploited in each of the following; Tell how this freedom can be exploited to minimize Area, Delay, and Power. (I expect about a sentence for each piece (freedom, exploit area, exploit delay, exploit power); However, if you feel you need to write more to make your answer clear, that's fine.)
 - retiming
 - logic optimization
 - sequential optimization
 - covering
 - placement
 - routing
 - scheduling

2. Consider the case in which your basic building block is an asymmetric LUT. The LUT has k inputs. Input i has delay i . So, an input arriving on input 1 can affect the output in 1 unit delay, one on input 2 in 2 unit delays, etc. Since it's a LUT, all inputs are *logically* equivalent, giving you freedom to assign logical inputs to any position.
- Assume covering and placement have already been performed and routing delay may be significant.
 - (a) How would you exploit this freedom to minimize delay?
 - (b) Sketch the algorithm to assign physical, asymmetric LUT inputs to logical LUT inputs.
 - Assume covering and placement have already been performed and you are concentrating on channel routing and detail input assignment. For this, you may assume routing delay in the channel is negligible.
 - (a) How can you identify optimization targets (delay or channel width) which are clearly not achievable?
 - (b) How tight are your bounds above? *i.e.* what can you guarantee to achieve?
 - (c) Explain the effects which might cause you to achieve non-minimal channel width.

3. Consider a logic discipline where every gate is registered. We want to take our original logic circuit (sequential netlist) and map it into this discipline, preserving the original semantics.
- Assume sequence of registers has unit cost
 - Assume we can tolerate an arbitrary number of clock delays through the circuit.
 - Assume we can also look at the output every n -th cycle, if necessary.

Questions:

- (a) How do we make the transformation preserving semantics?
- (b) What freedom do we have in implementation to reduce the number of register sequences?
- (c) How do we minimize cost if the circuit is a DAG (no cycles)?
- (d) What can you say about the case where there are cycles? (*N.B.* I think the cyclic case may be harder and require some tools we did not discuss in class. So, I'm just asking you to identify what you can about this, including why it's different from the DAG case; don't spend long answering this part.)

Where possible, you may refer to algorithms discussed in class or in the reading.

4. Consider a time-multiplexed routing network. Placement has already been performed. A graph represents the channels in the physical network and the connectivity among channels. Each channel can carry one signal from one end to the other in unit time. The channel is pipelined, so can carry a different signal across the channel in each unit of time. In the worst-case, all edges in the logical graph are routed through the same physical channel, and the delay for evaluating the logical circuit is $D_{logic} + D_{net_delay_longest_edge} + |E| - 1$, but the design is always routable if we assume enough clock cycles for routing. Assume there is no buffering between channels in the network, such that a route from a source through channel 2 to channel 4 to a destination node which occupies channel 2 on cycle 3, must use channel 4 on cycle 4.
- (a) Given that a node p has been computed at time t , formulate the search that will find the conflict-free route through the network which delivers the result to its consumer q at the earliest possible time after t given any existing routes which have already been performed on the network.
 - (b) Sketch a greedy algorithm to route all edges minimizing evaluation time for the circuit. Be clear about any assumptions you make concerning the representation of the problem during routing.
 - (c) What weaknesses will the greedy algorithm have?
 - (d) Adapt the ideas of congestion negotiation (*e.g.* pathfinder) to this problem and completely describe a better algorithm for minimizing the evaluation time of the circuit.