Exam Feedback – VLSI Design (EEE310/6036), May 2011

General

In general, the paper was straightforward and 'reasonably-well' attempted. However, there were particular problems.

Q1.
In this question, the problem was part b) where (for reasonable assumptions) the voltages at 3 points were to be determined. Most people could see that, assuming that all the transistors were on, that X would be V_{DD}-V_T. However, what seemed to confuse people (despite the point being made in lectures) was that because X is already one V_T below the gate voltage, Y will also be approximately V_{DD}-V_T as will Z. The most common mistake was to say that Y= V_{DD}-2V_T and Z= V_{DD}-3V_T!

Q2.
This question was reasonably answered. Clearly, there were the usual batch of mistakes in translating the circuit into transistors. People seem to be confused about what gets inverted and where. To recap, for the pull-down network, the expression for Y is inverted and a true term on the RHS means a n-type transistor driven directly by the input. For the pull-up network, Y is not inverted and a barred term on the RHS means a p-type transistor driven directly by an input. If a term is barred in the pull-down expression then the input must be inverted before driving the transistor. If a term is no barred in the pull-up expression then, again the input must be inverted.

Q3.
Part a) was reasonably well answered although sub-parts iv) and v) were less well answered. Part b) was less well answered (I was quite generous in part b.i) for example in what constituted a connection to test.

Q4.
This seemed to be the least well approached question – possibly because it is more 'wordy'. Section iii) was particularly badly dealt with – despite me going over this particular point in a lecture. To meet the demands of causality, a simulator will add an infinitesimal delay, Δ, to the time at which an output changes (relative to the input change) if no delay is specified. So if a changes as T then it will change at T+Δ. The simulator will pursue all the changes arising from a change at time T and any multiple of Δ thereafter until no more changes occur and assume that all these changes take place at T before moving on to a subsequent time. However, in this case a change at T results in a change at T+Δ and this results in a change at t+2Δ and so on. Thus, because this process is never ending the simulator can never escape from time T and the simulation stalls. Conversely, is the statement had been a <=not a after 1 ns; then because the initial change in a at time T takes place at a time later than T i.e. T+1e-9 then this change will not be considered (or pursued) until pursuing all the changes at T are completed and the simulator moves on to consider T+1e-9. In synthesis (and this is not asked for), the statement is fine because it will just implement a signal connected between the input and output of an inverter. It will, of course be useless because it will probably result in a being stuck at some indeterminate level.