

Experiment 5

Static Logic Hazards and Simulation

One of the difficulties of logic design is that a circuit may be logically correct, yet when implemented, it may produce transient conditions which correspond to incorrect behavior. There are both static and dynamic hazards. This experiment demonstrates a static 1-hazard and shows you the difference between a simulation comprised of gates with their physical properties incorporated in them and a simulation which does not take these properties into account.

A theorem of digital design states that a logic 1-hazard exists in a logic circuit *iff*:

- There exists a pair of input vectors for a logic circuit, both of which produce a logic 1 on the output.
- These input vectors differ only in that a variable in one vector is complemented in the other vector.

This can be visually seen on a Karnaugh map as two adjacent cells (minterms) which are both one in the static case, but are not covered by a common term. This can easily be demonstrated by implementing the following logic function in a two-level (SOP) form.

$$f1(A,B,C) = (\bar{A} \bullet \bar{C}) + (B \bullet C)$$

This function can easily be seen to be comprised of minterms 0, 2, 3, 7, however $f1$ does not cover minterms 2 & 3 simultaneously. Since these two minterms differ only in C and $NOT C$, a static 1-hazard exists. That is, when C changes from 0 to 1 or vice versa with $NOT A$ and B , there is a potential for the output to momentarily change to a zero when it should stay at a value of 1. To eliminate the hazard, one can add a redundant term to cover the static 1-hazard.

With respect to the above problem do the following:

1. Write a **package** containing behavioral VHDL implementations of *AND*, *INV*, and *OR* gates. Create two versions of each gate (delay and non-delay) using `ece332_gates.vhd` as a guide. Set the propagation delay for all non-ideal gates to 20 ns.
2. Write a structural VHDL implementation of $f1$ above utilizing ideal gates with no propagation delay.
3. Write a structural VHDL implementation of $f1$ above utilizing gates with a propagation delay of 20 ns.
4. Write a test-bench which steps through the following sequence of inputs:
 $ABC = 011, 010, 110, 111$
5. Analyze and compare the results of the two simulations and draw timing diagrams to show how the outputs differ when C transitions from a 1 to a zero.
6. Write a structural VHDL implementation which includes the redundant term to cover the static 1-hazard.

7. Exercise the redundant terms with the test bench and verify that the static 1-hazard has been removed.
8. Implement f_I with physical gates and capture the transient 0 which occurs when C switches from a 1 to a zero.