Rodrigo Becerril Ferreyra

Student ID 017584071

CECS 440 Section 02

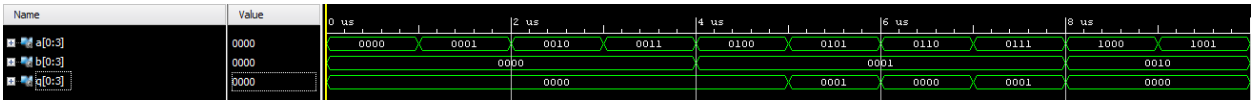
13 October 2021

Lab 3

Before I start showing screenshots and code, I want to talk about my experience with this lab. As far as HDLs go, I am proficient in Verilog, so I was taken by surprise when I saw that this lab was to be completed in VHDL. I did have some trouble translating concepts and code from one language to the other, but thanks to a few websites which provide helpful tables and equivalent statements, I was able to complete the Lab. Also, I noticed that none of these circuits are sequential. I am grateful, and would love to be eased in to sequential circuits in VHDL.

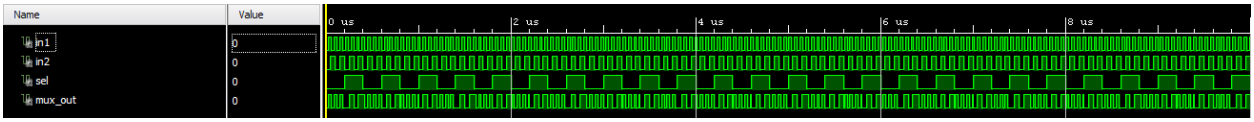
All source code is given in the drop box.

Below is the test for the four-bit AND gate (feel free to zoom in).

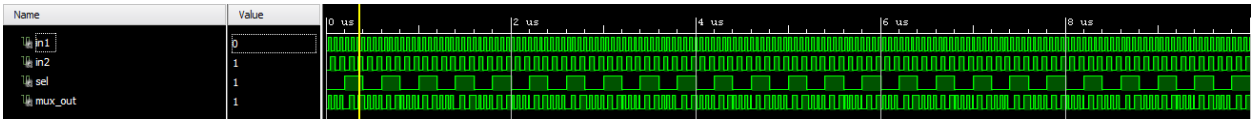


As you can see, each individual bit in the result `q` is set only if both corresponding bits in `a` and `b` are also set. This simulation does not show all the combinations, but the amount of data is sufficient to see that this design does in fact work.

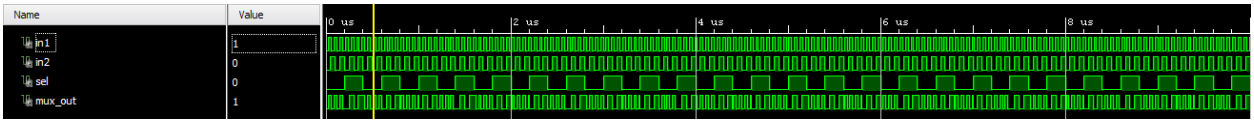
Below is the test for the one-bit two-to-one mux.



This data is harder to test, so I will explain three test cases. The first (above) has `in1` cleared, `in2` cleared, and `sel` cleared. The mux picks `in1`, which has a value of 0, so `mux_out` has a value of 0.

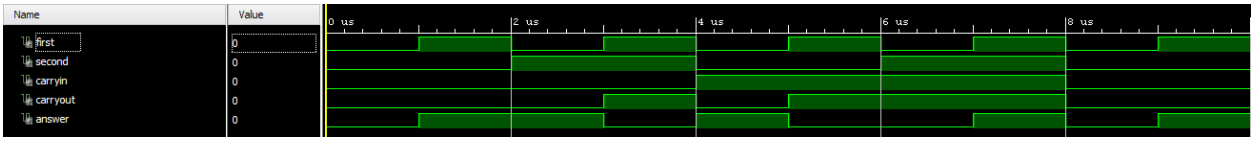


At this point in time, `in1` has a value of 0, and `in2` and `sel` both have a value of 1. `sel` means that the mux chooses `in2`, so `mux_out` has a value of 1 as well.



In this last example, `in1` has a value of 1, and both `in2` and `sel` have a value of 0. The mux chooses `in1` due to the value of `sel`, so `mux_out` has a value of 1.

Below is the simulation for the one-bit full adder.



Note that all eight test cases are covered in this simulation. The results can be verified by hand that they are correct (they are).