Lab 2 – post-lab:

Write one paragraph and attach to this sheet in hardcopy describing the most difficult portion of this lab and also give two different things that you learned from doing this lab. Describe why increasing the simulation time value in the settings (xsim.simulate.runtime) without changing the testbench will yield the exact same output.

One paragraph and attach to this sheet in hardcopy describing the most difficult portion of this lab and also give two different things that you learned from doing this lab:

The most difficult portion of this lab involved fully understanding which “signal” was being referred to in the instruction “Note: Right click on the signal name, click on Radix. You can set the value display format. Above figure, I change the seg value from hex to binary” under the portion of the lab entitled: “Part II Post-Synthesis Timing Simulation”. I was able to change seg values for each of signals “sw”, “led”, “seg” and “an” but wasn’t completely clear on which signal was being referred to. I will make sure to follow up with the TA to comfirm my understanding was correct regarding what was being asked and whether I correctly met the requirements for this portion of lab2.

Two different things I learned from doing this lab included:

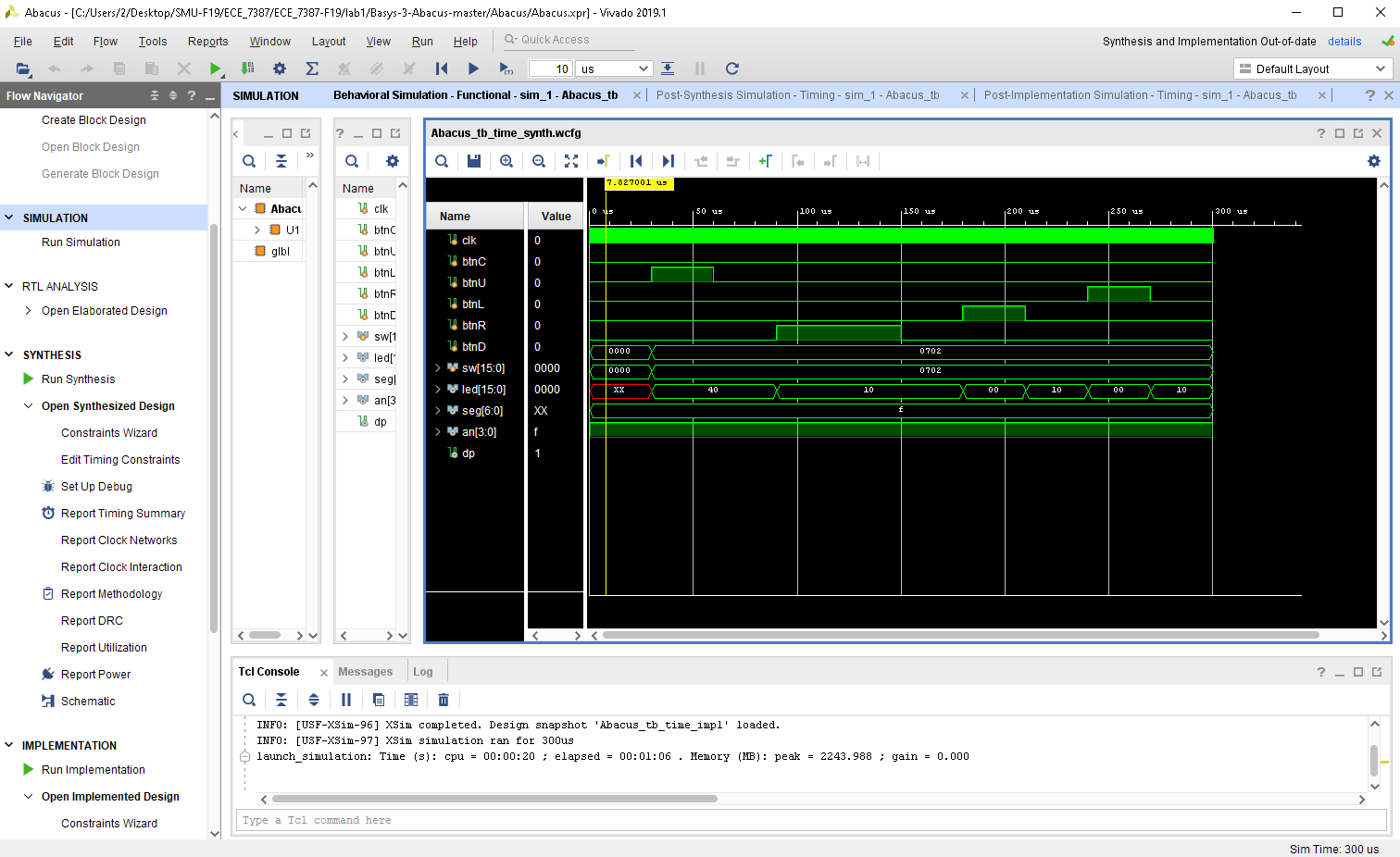
1. Changing views in the tool to display different resolutions of the waveform configuration (wcfg) file outputs.
2. Altering the format of the output values (e.g., from default of hex to binary) by navigating to the radix sub-menu for a signal.

Increasing the simulation time value in the settings (xsim.simulate.runtime) without changing the testbench yields the exact same output because the simulation time value in the settings (xsim.simulate.runtime) indicates how long (in units of simulated time epochs) the design should be instantiated. Instantiation (what is involved here) refers to the process of creating objects (called instances) from a module template and since changing the simulation time value only changes the amount of time the objects are created for (for purposes of this simulation), this change would have no impact in this case on the behavior of the instances themselves, and would therefore not result in a change in the output as compared to the previous simulation time value.

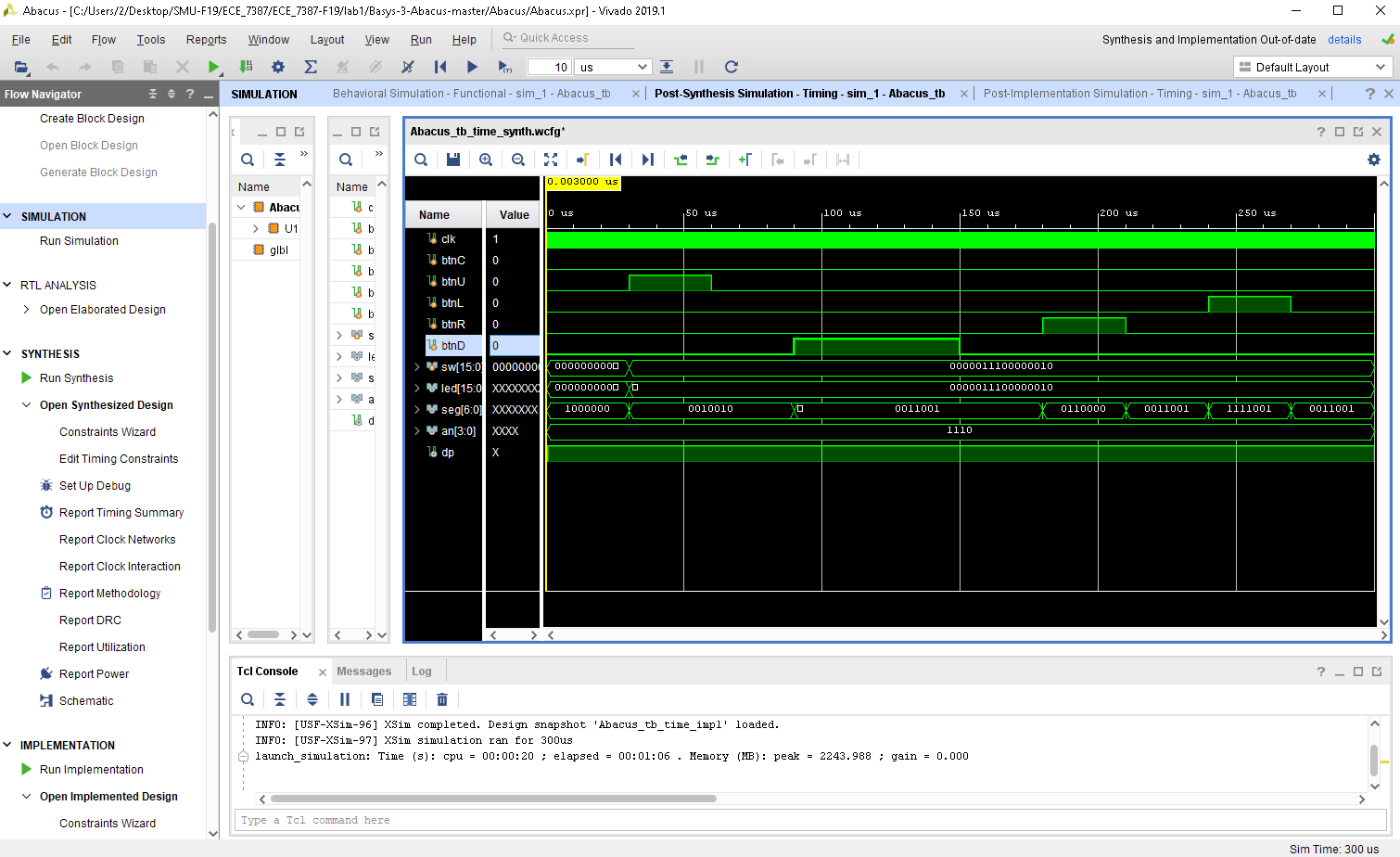
Lab 2 – relevant screenshots:

The full view should look like this following figure. It shows some different test cases and the corresponding simulation results. An important thing to note here is that output signal changes

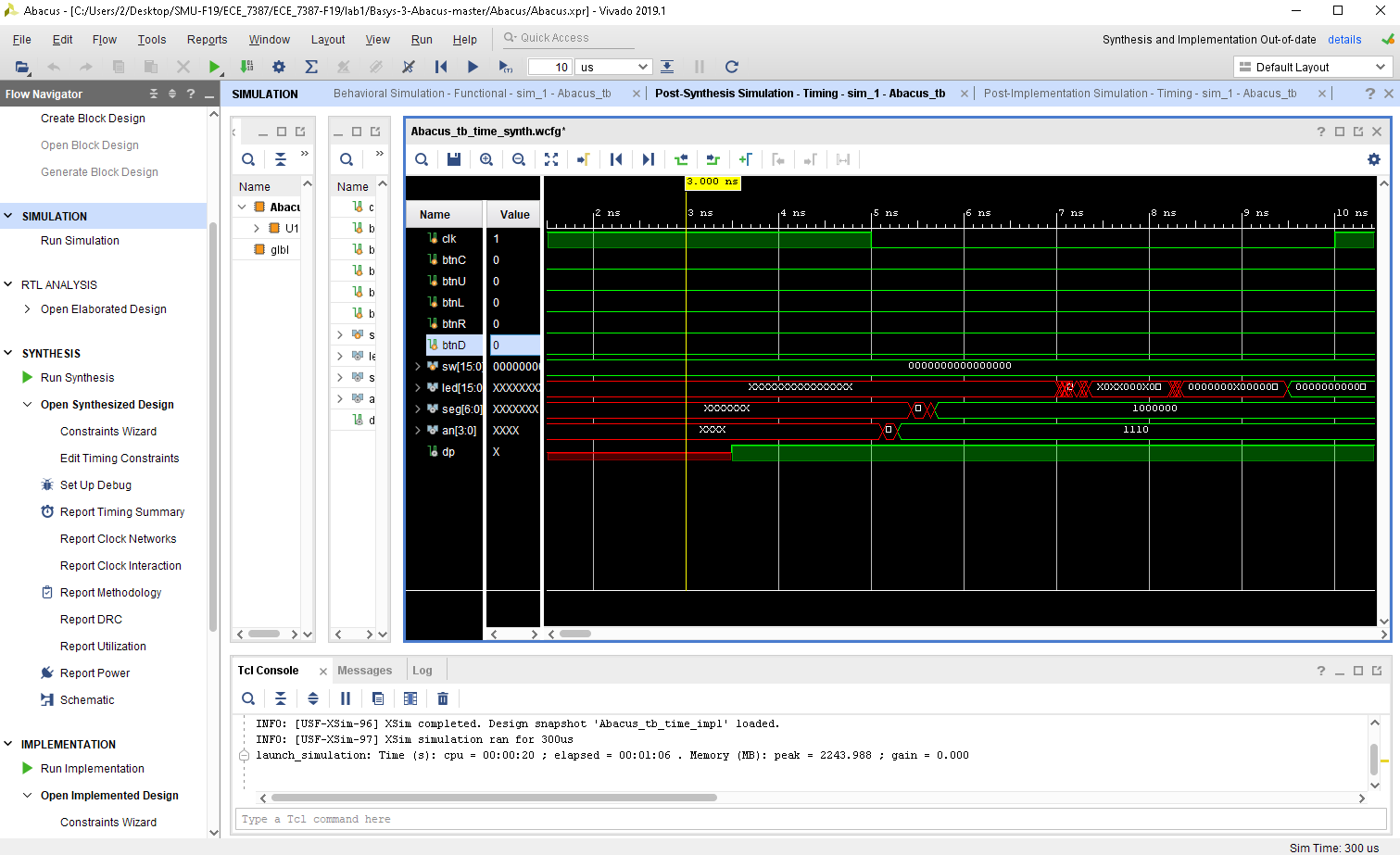
occur with ZERO delay after the inputs change and internal state of the design change. This is because there is no timing delay information included.



Part II Post-Synthesis Timing Simulation After running the behavioral synthesis, the “Run Post-synthesis Timing Simulation” option will be available (in black, if it is in grey, it is not available). The post-synthesis timing simulation process is almost the same as the behavioral synthesis procedure, except we will click on “Run Post-Synthesis Timing simulation.” This type of synthesis takes longer to run (in terms of “wall clock” time) than behavioral simulation. The following figure shows the post-synthesis timing simulation.



When one zooms at the beginning of the simulation time epochs to the leftmost portion of the waveform, we can observe delay shown on the outputs with respect to input events.



Part III Post-Implementation Timing Simulation When the implementation step is finished, we are able to run a post-implementation simulation (both post-implementation timing simulation and post-implementation functional simulation are available). Again, the only difference to run post-implementation timing simulation as compared with behavioral and post-synthesis timing simulation is to click on “Run PostImplementation Timing Simulation”

