

CECS 360 Fall 2016 Project 4

Simulating the Netlist

By

Shinesunny Som

11/29/2016

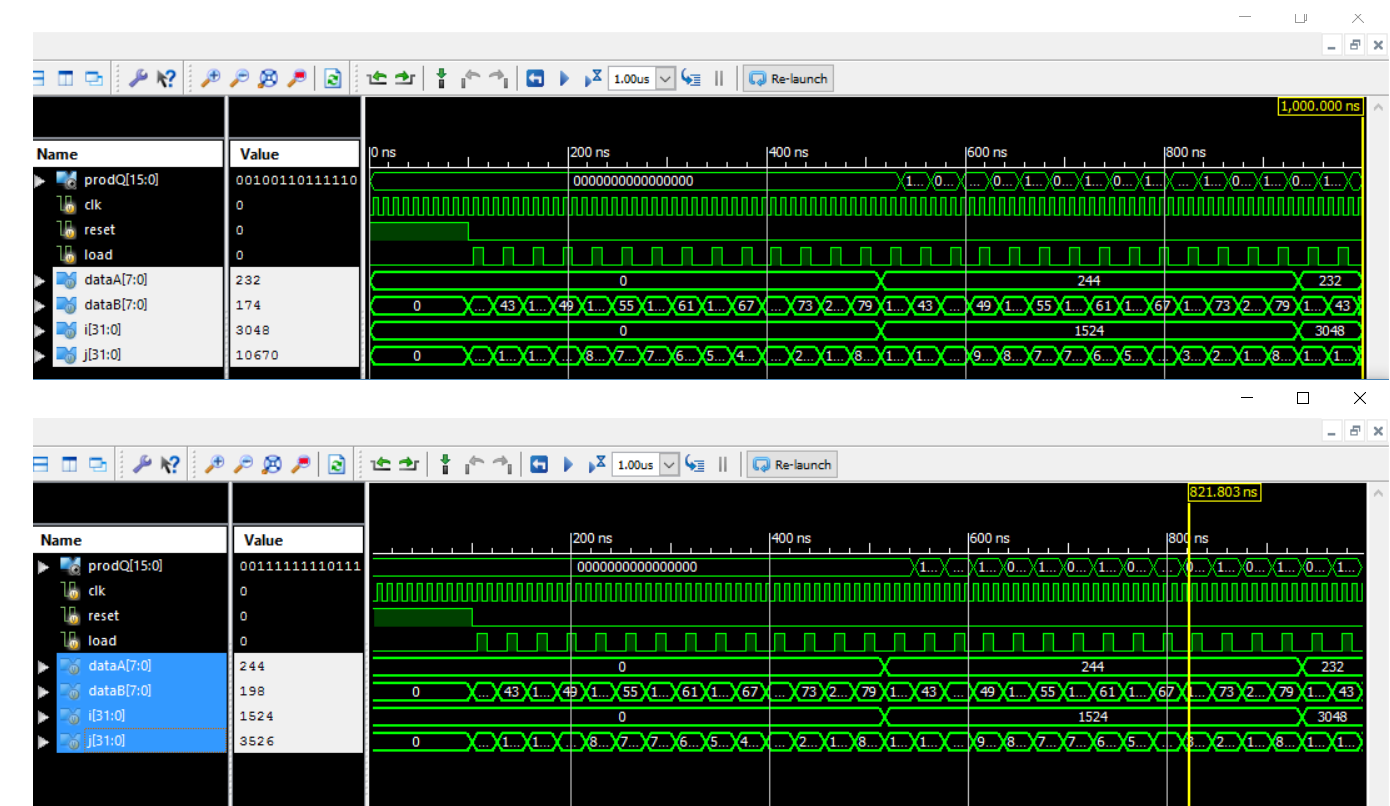
Instructor: John Tramel

**Lab Description:** We will use ISE (Xilinx Project Manager ver. 14.7 x64) for analysis of our design configured the design in the default Nexys2 format. Testing the “Synchronous Multiplier” design, we are to synthesize the design with the given verilog modules, implementation constraints file, and testbench stimulus. The entire project is said to be completed but it is seemingly so with careful inspections of simulation tests. This project is designed to identify propagation delays, and verify the design with inspections of simulation tests.

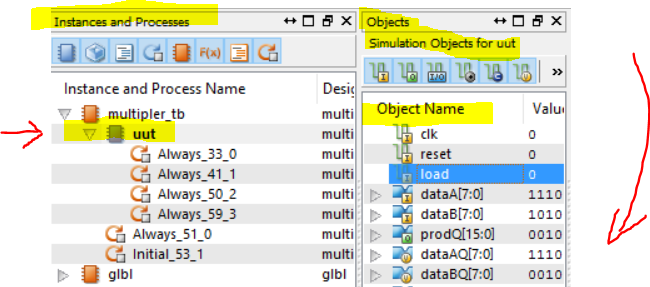
**Verification Report:** Using only simulation we are to test two types of simulation checks. Before the simulation, we test the Top Level RTL view of the design and with confidence, everything seems to be in order and connected. Top Level RTL view is not a reliable source to inspect design but used to give you a simple glance of the design being tested. For the simulation test, we are required to test not only the behavioral simulation but to generate the Post-Synthesis simulation model.

***\*Observation:***

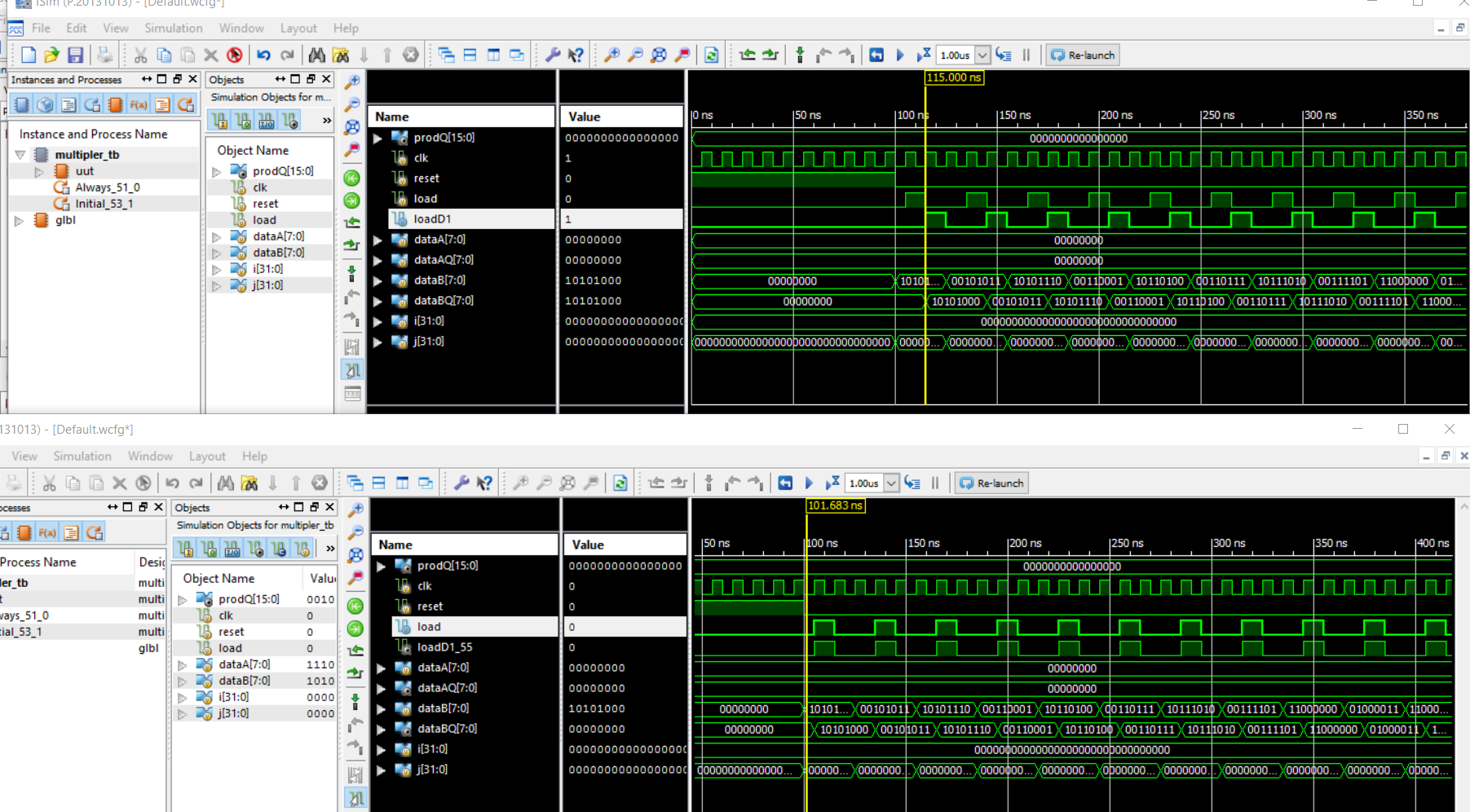
**Comparing Behavioral vs Post-translation simulation. This is looking at it in a glance.** Identically they look the same in terms of wavelengths.



At closer inspection you find that they are not the same from the **prodQ** **output** as they slightly misaligned, but even conclusively, you can inspect how the wavelengths from the load, by searching into the **“Instances and Processes”** and clicking to the **“uut”** from the testbench you can select any of the **simulation object** to compare into the simulation as shown here:



By selecting which register I can compare from the simulation results, I decided to compare the loads, and the outputs of dataA and dataB to do a side by side wavelength view. In a couple zooms, here being observed are propagation delays from the post-translation simulation. **The top simulation is the behavioral simulation and the bottom is the post-translation simulation.**

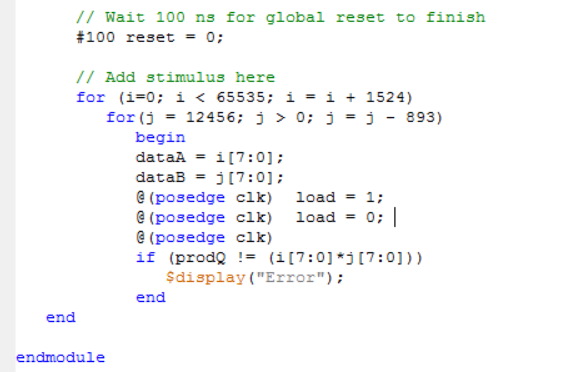


Behavioral simulation only give you an ideal perfect simulation test while using the post-translation simulation will simulate a more of closer gate, register inspection of a design.

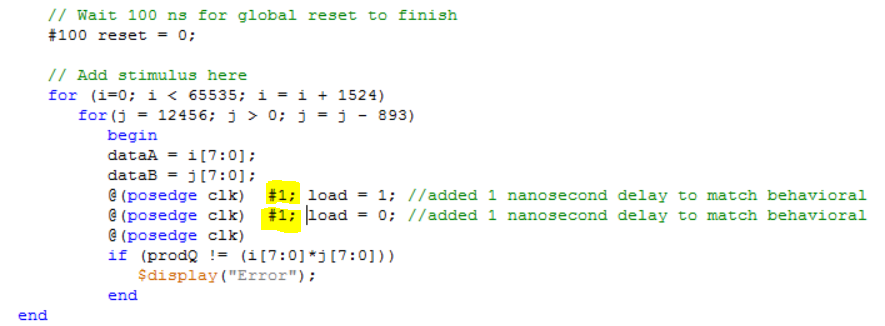
**Results:**

To remedy the design, I went to inspect the testbench stimulus and look to modify the stimulus code to match the behavioral simulation. Since, the **load** is performing the same as the **load output** in wavelengths, I went on ahead and add an **one nanosecond delay to the clock to give the load output time to be delayed so the load output value will be displayed correctly.**

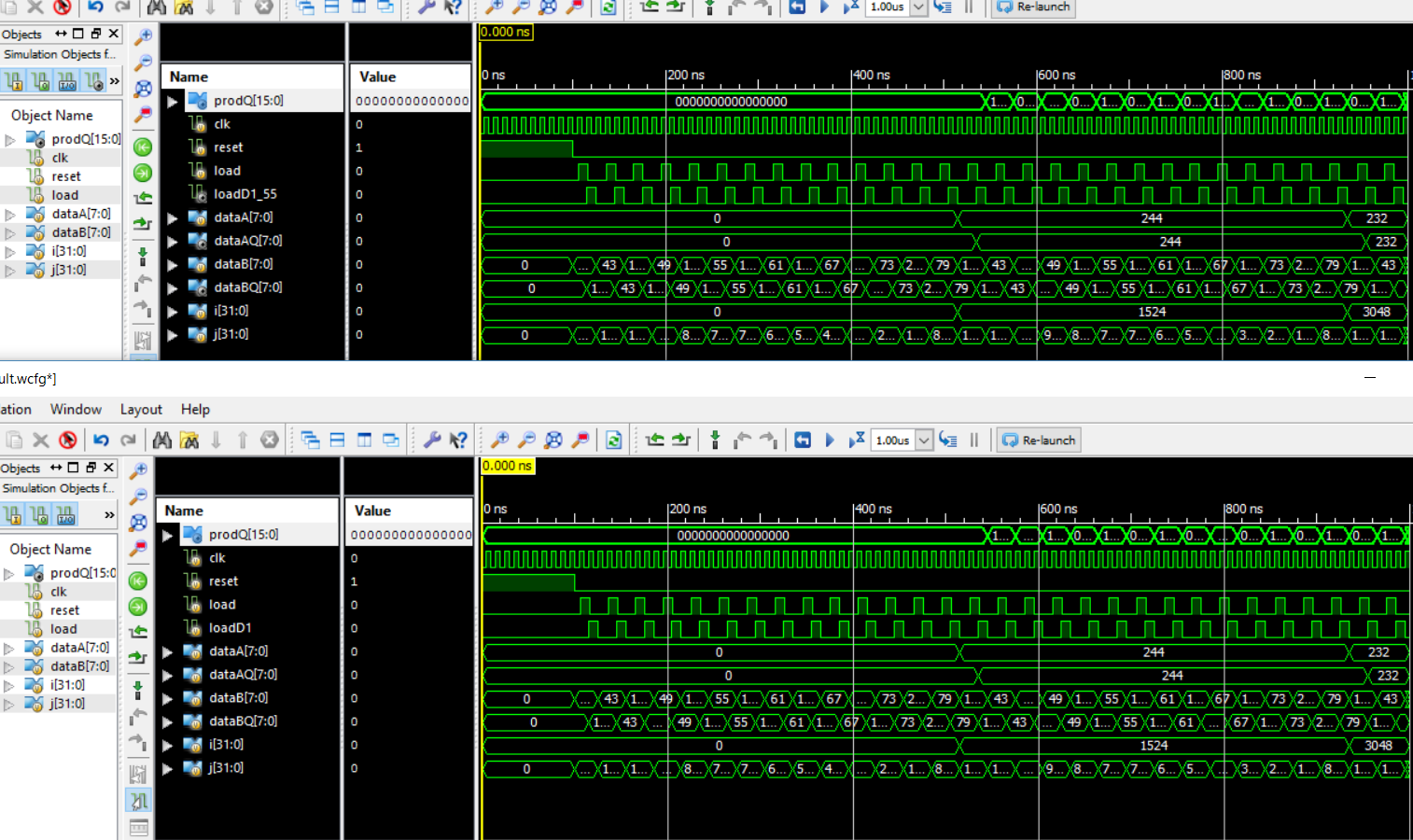
***Before edit:***



***After edit:***



After the adding the nanosecond delay, the post-translate simulation begins to shape up and match nearly to the behavioral design which we want.



**Problems/Issues:** Adding one nanosecond did fix the wavelength design, but if you zoom closer in the post-translation simulation, you can definitely see it does NOT match the behavior design which is much expected. I could fine-tune a value for a delay since the timescale is in nanosecond/picosecond to find the sweet spot match. For learning purposes, I did not spend time doing so.