**ECE 4250/ 7250: VHDL and Programmable Logic Devices  
Laboratory**

**Lab #5   
Lab Title: Generate a Clock**

**Group #2  
Group Names: Chris Smith, Benjarit Hotrabhavananda**

**Teaching Assistant Use Only:**

**Points Earned Reasons for Deduction**

**Pre-lab:**

**Post Lab report:**

**Demonstration:**

**Final Lab Grade:**

**Comments to students:**

**Objective:**

The objective of lab 5 was to use the 50 Mhz clock of the spartan 3 FPGA board to generate our own clocks at several different frequencies. To accomplish this our VHDL programs utilize the built in clock and a counting mechanism to establish the clocks of different frequencies. For example, establishing a frequency of half of a hertz first required knowing how many clock ticks of the FPGA clock will tick before we set our established clock low (this will occur at half of the period). This allows us to count up until a specified value (depending on the period) and set our clock accordingly. We also began working with generate statements and generics to implement the four different clock frequencies for testing purposes, and then implement them to the FPGA hardware.

**Lab work:**

It was near impossible for us to run an accurate simulation of the generate\_clock.vhd file, as the simulator allows execution only for specific times. We were unsure how to allow the simulation to run for seconds at a time, and with the frequency of the 50 Mhz clock being so high, it was nearly impossible to see the clock transition occurring at the right time. We tried to simulate several times but moved onto the hardware implementation for most of our testing purposes.

**Conclusion:**

There were some issues that we encountered for testing purposes. Initially, we could only get one of the LEDs of the board to activate. We were confused at first as we generate four different clocks the testClock.vhd file. We discovered that the issue was an incorrect mapping of the UCF file. We had simply incremented the pin numbers but some of these pins were not in fact wired to any LED, so no output would be written. Once we had fixed the issue of the LEDs lighting up we had to fix the period of the clocks. We had initially written our code using hex representations of the numbers that we would need to count to for each clock process. We think these numbers were incorrect and caused the LED frequencies to be consistent, however they were not synchronized over a 4 second period which should not occur since they are all multiples of the lowest frequency. Source code including comments has been appended in the zip file containing this report.