Erich Viebrock

Marvin Soriano

ECPE 174

Lab #5

Asynchronous Communication

**Problem Summary:**

For this lab assignment, we are implementing VHDL code to create communication between two FPGA boards. The communication will be asynchronous, and will resemble machines acknowledging receiving and sending information.

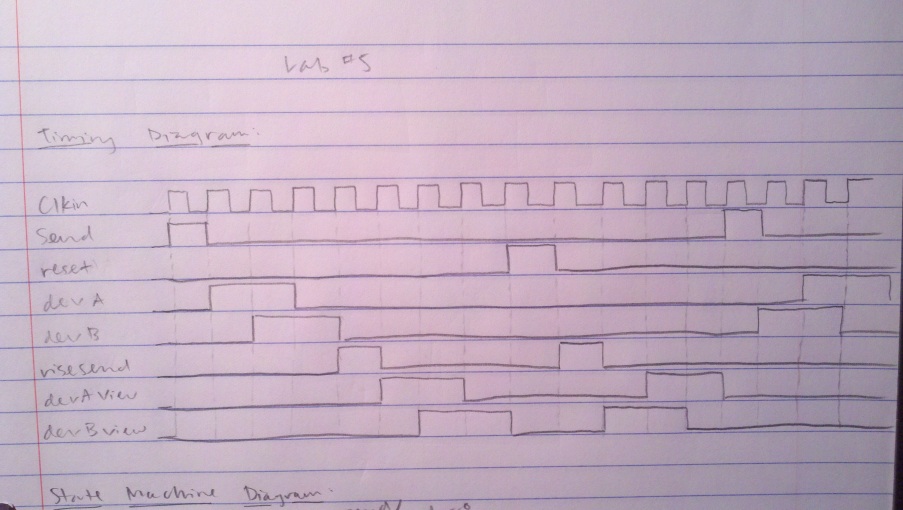
Each device has two signals, devA and devB. The send key will begin the transfer of information between the two devices. The communication begins with device A pulling devA high, then devB. When devB is high, devA will go low. Then, device B will pull devB high. When devB of device B is high, devB of device A will go low. Finally, devA of device B will go high and devB of device B will go low. This sequence will conclude the transaction between device A and device B. The same logic pertains to device B communicating with device A, however all of the signals are opposite. For example, device B begins the communication with devB, rather than devA.

**Assumptions:**

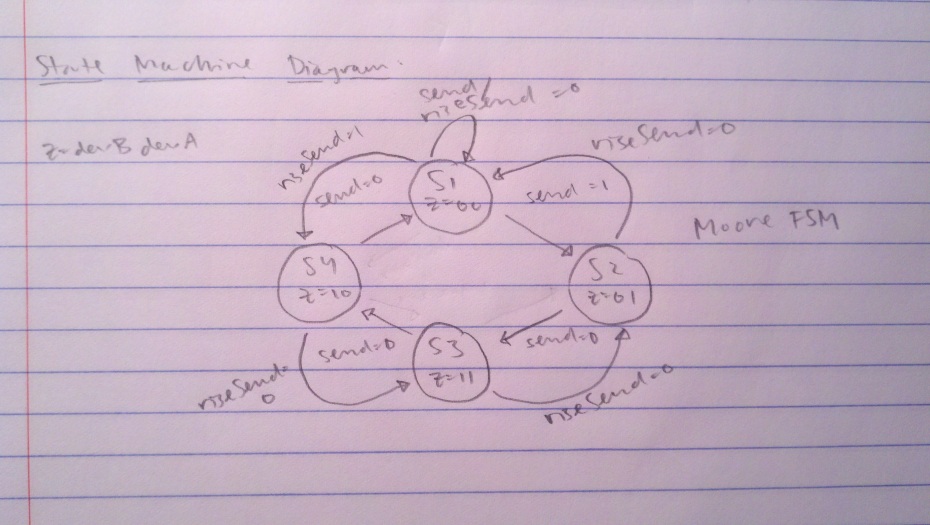
Once the data transfer sequence begins, user cannot enter opposite data transfer sequence during initial data transfer. In reality this situation is common, however this scenario is ignored in the interest of time.

**Design Approach:**

**Timing Diagram:**

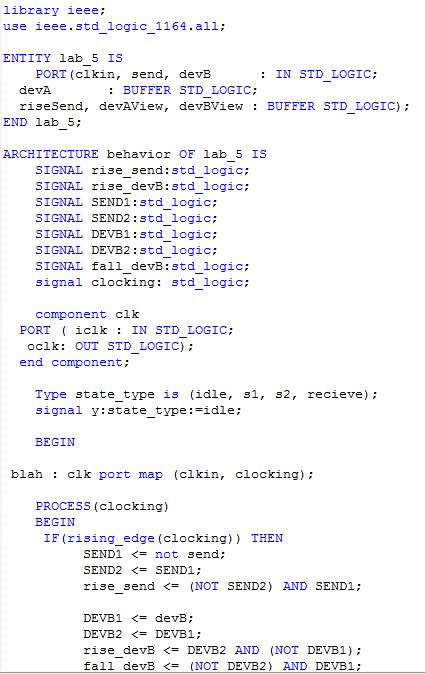
****

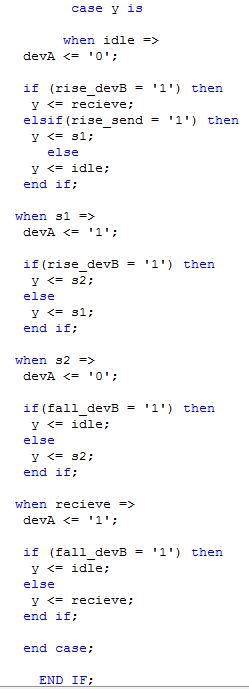
**State Machine:**

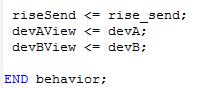
****

**VHDL:**

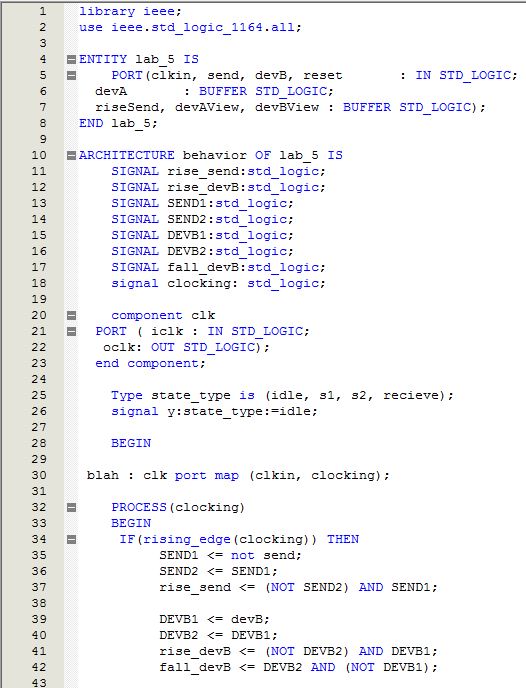
**One device**

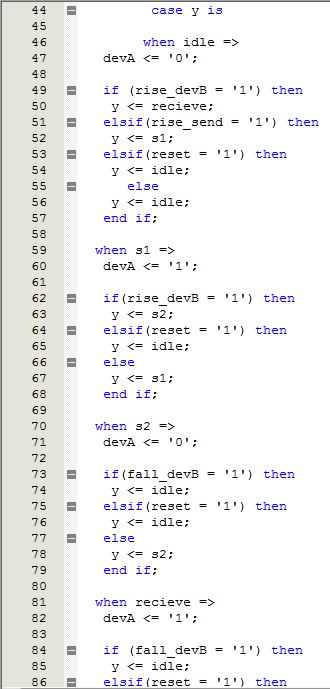
****

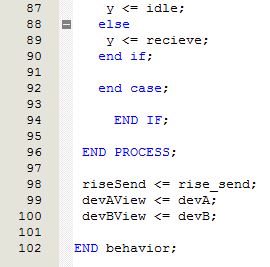
****

****

**Two devices:**

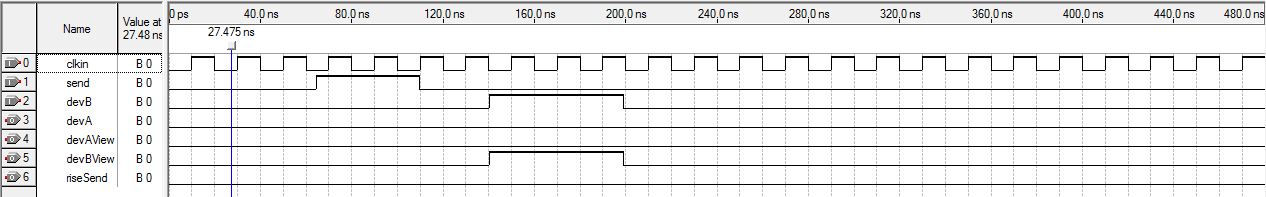
****

****

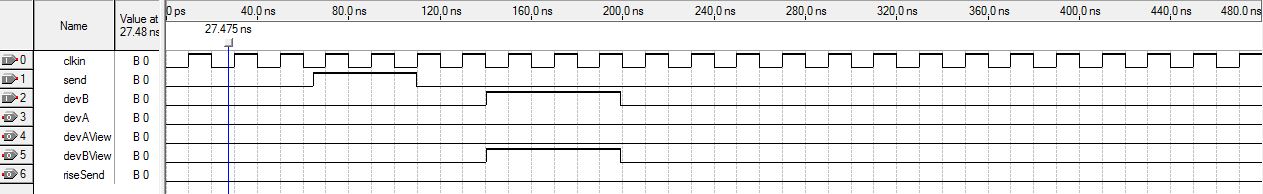
****

**Timing Diagram:**

**One Device:**

****

**Two Devices:**

****

**Verification Procedure:**

The initial problem we encountered was confusion of the problem statement itself. Given each device has two output signals, “devA” and “devB”, we were unsure how the state machine operated when only two outputs were mentioned. After conversation with the professor, and fellow classmates, we determined the problem statement was much easier than previously anticipated. Once we discovered how to perform the laboratory, we modified our entire VHDL to suit the guidelines of the asynchronous communication. This demanded some time, being we completely remodeled our VHDL. During the remodel process, we ran into very few errors. However, after determining our VHDL was completed, desired output was not seen. Transaction from device A to device B did not occur, but device A showed desired output. After some conversation with fellow classmates, we were able to determine our VHDL was correct, and device B was waiting on us to allow the transaction to take place. By sending “riseSend” HIGH, this was device B’s method of acknowledging, and accepting information from device A. Once this problem was resolved, asynchronous communication worked effectively between device A and device B.

**Post Lab Questions:**

1. When both devA and devB are asserted at the same time, no communication is initiated. This makes sense, because the FPGA boards are only able to operate in one direction, rather than two directions simultaneously. In order to modify the protocol to ensure no problems ensue the design, VHDL is used to determine how the FPGA board will dictate these unwanted scenarios.
2. When the frequency of one device was modified, the fundamental frequency was used for communication through the system. This was expected, because even if one device is clocked faster than another device, the slower clocked device will limit the entire system to that specific clocking.
3. In order to modify a system to communicate data, the underlying question is what is transferring data to what? If the designer is creating communication between two computers, the designer is going to use a high frequency, in order to utilize the computer’s speed and efficiency. If the designer is creating communication between a vending machine and output LEDs on the vending machine, the designer will use a slower frequency. Although this will make the system run at a slower speed, a lower frequency enables the user to view LEDs at a reasonable rate.