Digital Design

CSCE 2114-L007

Blake Fasse

May 1, 2017

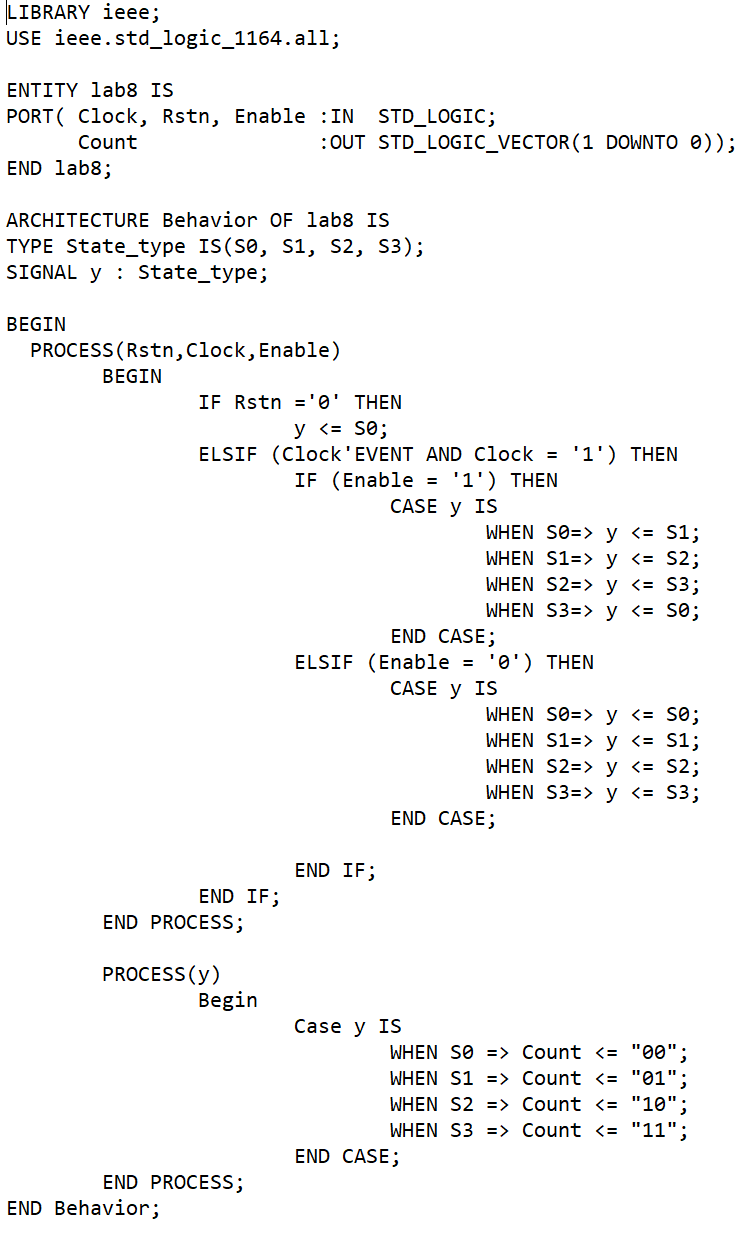
[bafasse@uark.edu](mailto:bafasse@uark.edu)

**Introduction**

This lab consisted of writing VHDL code from scratch to implement a Moore type state machine. The Moore type state machine will cycle through values 0 to 3 and change on the rising edge of the clock as long as enable is equal to 1. If enable is ever equal to 0 then the count value will loop that same input until enable is equal to 1 again.

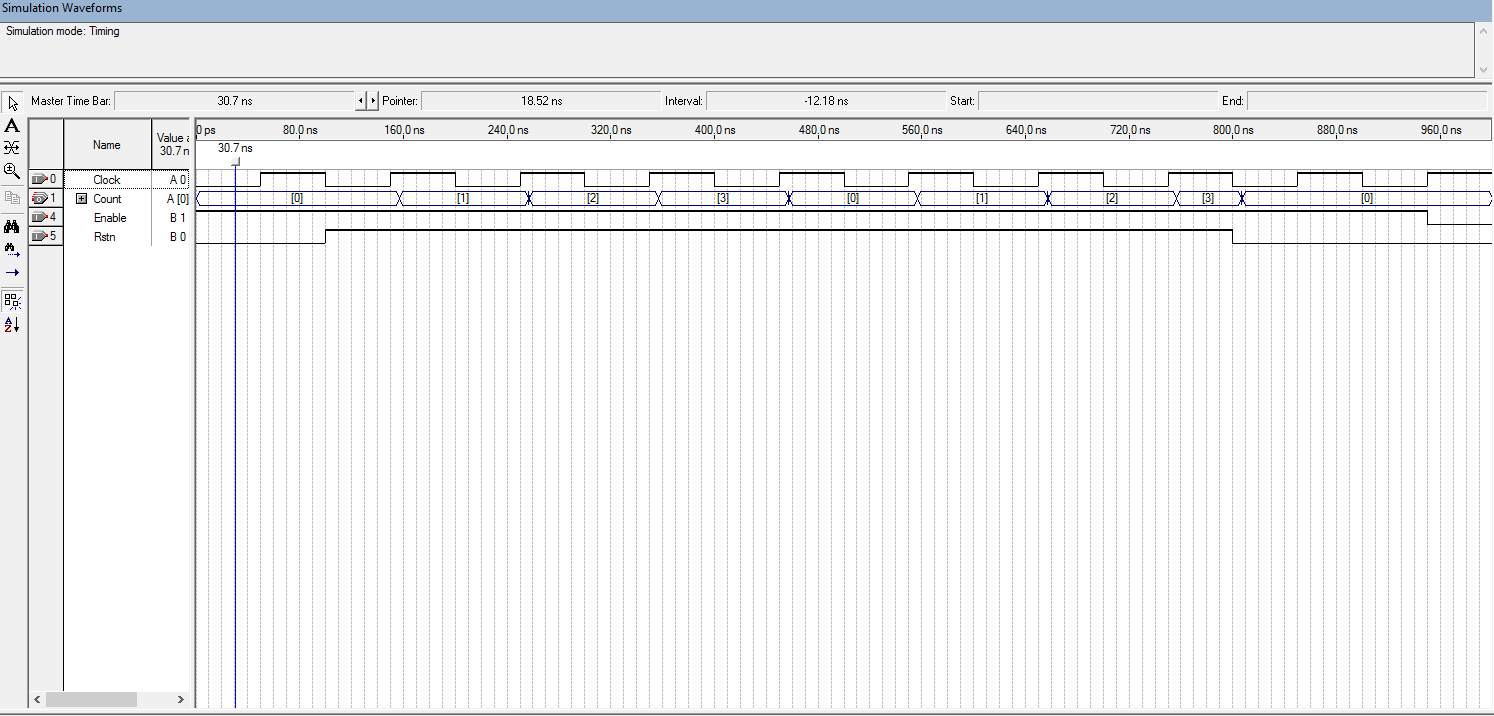
**Design**

There are three inputs, clock, reset, and enable, and one output, count. In place of the traditional counting of bits, S0, S1, S2, and S3 are used instead to mean 00, 01, 10, and 11, respectively. Anytime the reset value is set equal to 0 then the y state type variable used is set equal to S0. In order to move onto the next state enable must be set equal to one and anytime it is set equal to 0 then the y state type variable will remain in its current state. All of that is done within its own process. Since y is not an output another process was made in order to set the values equal to count so that the changing state can be seen on the vector waveform graph. A picture of the code will be shown below.



**Results**

The results can be seen in the vector waveform below. In the vector waveform it is shown that whenever the reset value and enable are equal to 1 then the count value is allowed to freely cycle through its values. It is also shown that any change in the value of count is rising edge triggered but when the reset value is equal to 0 then count value is forced to S0 and isn’t allowed to change on the rising edge of the clock.



**Conclusion**

The Moore type state machine is not a difficult concept to understand. Anytime the enable is equal to 1 then it moves onto the next state but if the enable is equal to 0 the it stays within its current state. Whenever the reset value is equal to 0 then the value of count is set equal to S0 and not even the enable can change its value.