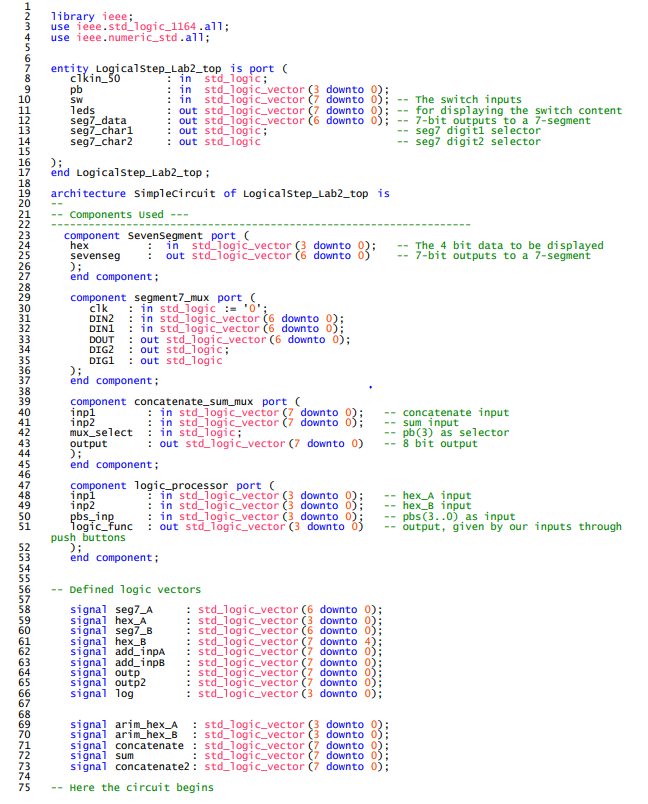
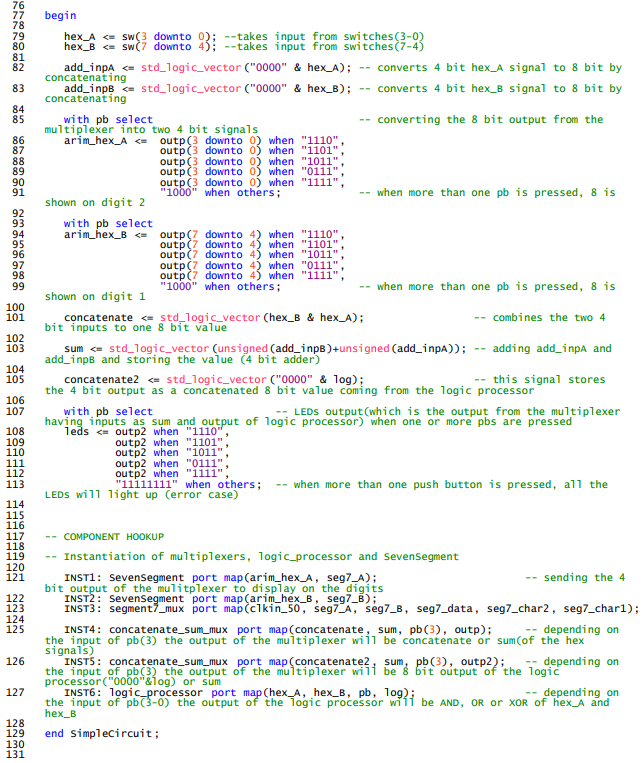
**Lab 2 Group 12 Session 205 Report**

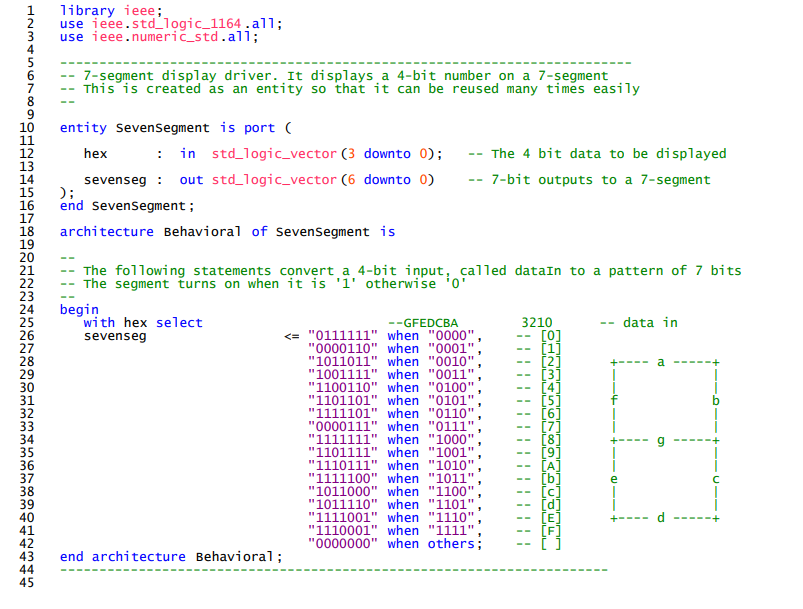
**Part 1**: LogicalStep\_Lab2\_top.vhd File



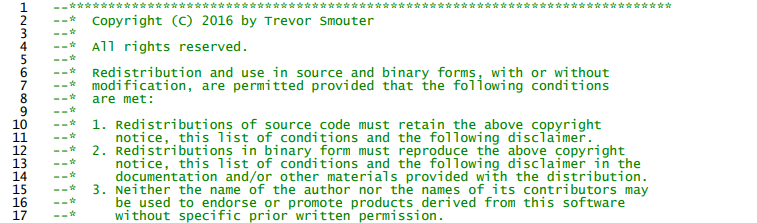


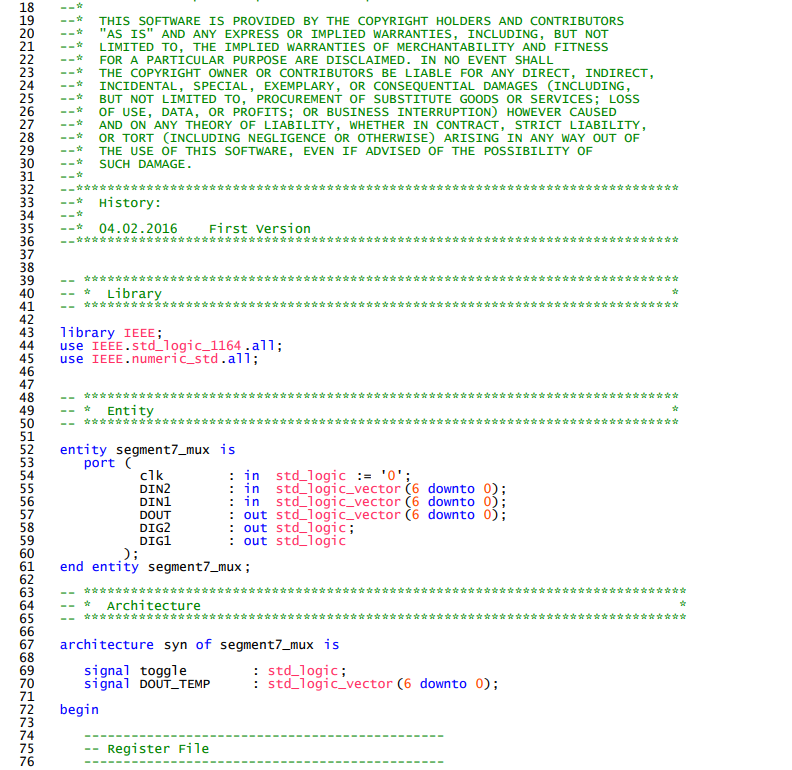
**Part 2**: Subordinate VHDL Files

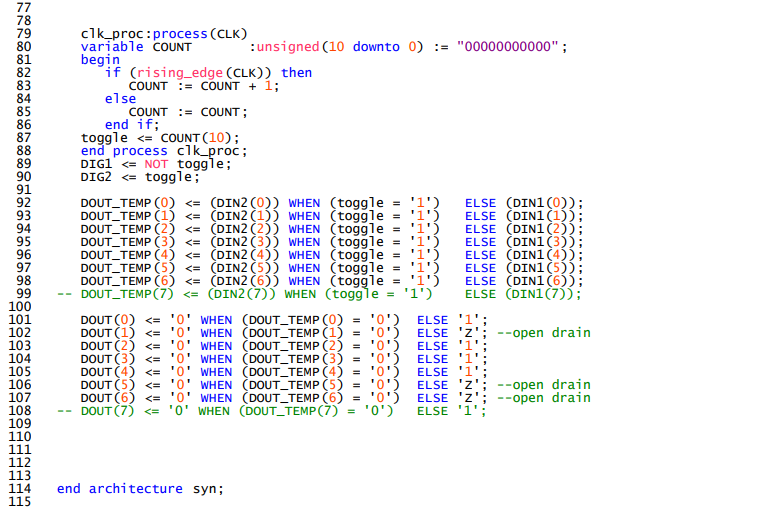
1. SevenSegment.vhd File



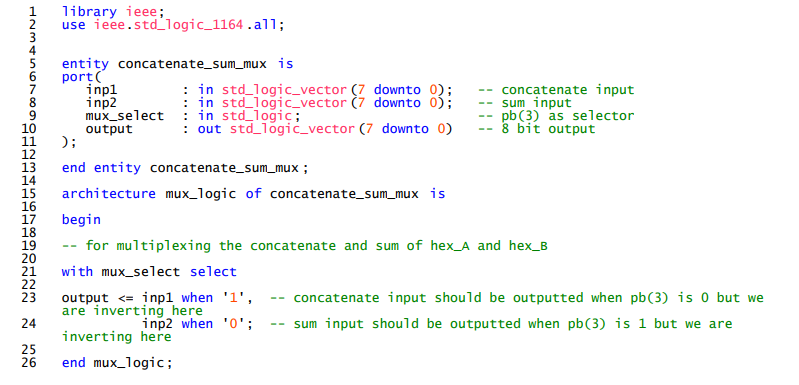
1. segment7\_mux.vhd File



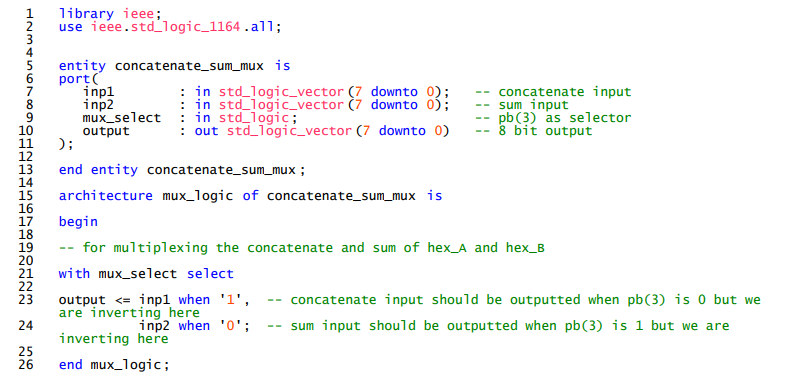




1. concatenate\_sum\_mux.vhd File



1. logic\_processor.vhd File



**Part 3:** Supporting Documentation

We have taken an example of inputs 5 and 6 to run our simulations. The number can be written in binary as:

‘A’ = “1010” and ‘6’ = “0110”.

Now, using the Simulation Waveform Editor, we will generate the outputs as seen on the LogicalStep by changing the values of pb(3-0). Before that let’s look at what the outputs should be:

When pb(0) is pressed, logical AND of operands should be performed, so:

pb(0) : “0101” AND “0110” = “0100”

Similarly, when pb(1) is pressed, logical OR of operands should be performed, so:

pb(1) : “0101” OR “0110” = “0111”

Similarly, when pb(2) is pressed, logical XOR of operands should be performed, so:

pb(2) : “0101” XOR “0110” = “0011”

When pb(3) is pressed, binary ADD of operands should be performed, so:

pb(3) : “0101” + “0110” = “1011”

Note: Since we haven’t inverted our pb’s, so, the input will be 0 instead of 1. For example, if pb(0) is pressed, that means the real input for the pb’s will be “1110” because in the LogicalStep, a push button’s value becomes 0 when it is pressed.

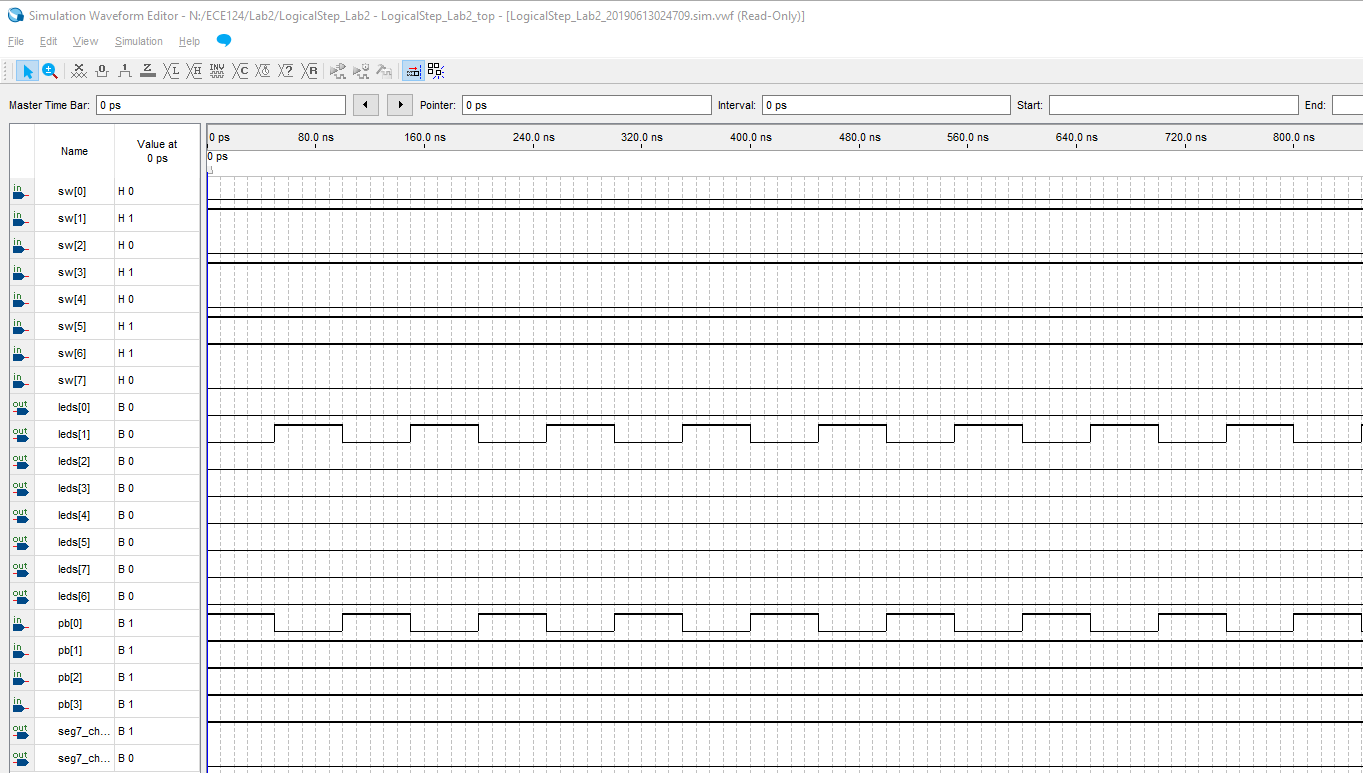


Figure 2: Functional Simulation of AND operation

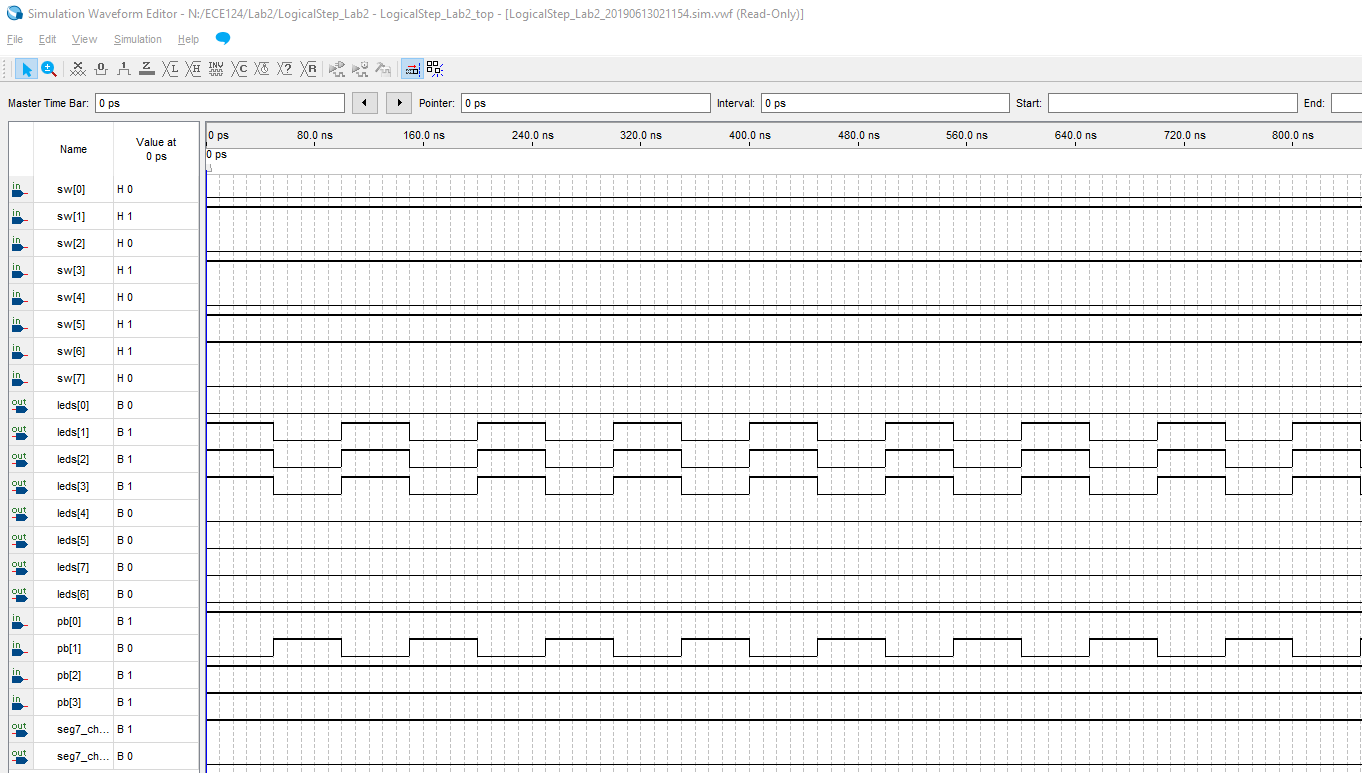


Figure 2: Functional Simulation of OR operation

Here we can see that when pb(1) is pressed, the output of the LEDs(0-3) is “0111” which is the desired output as mentioned above.

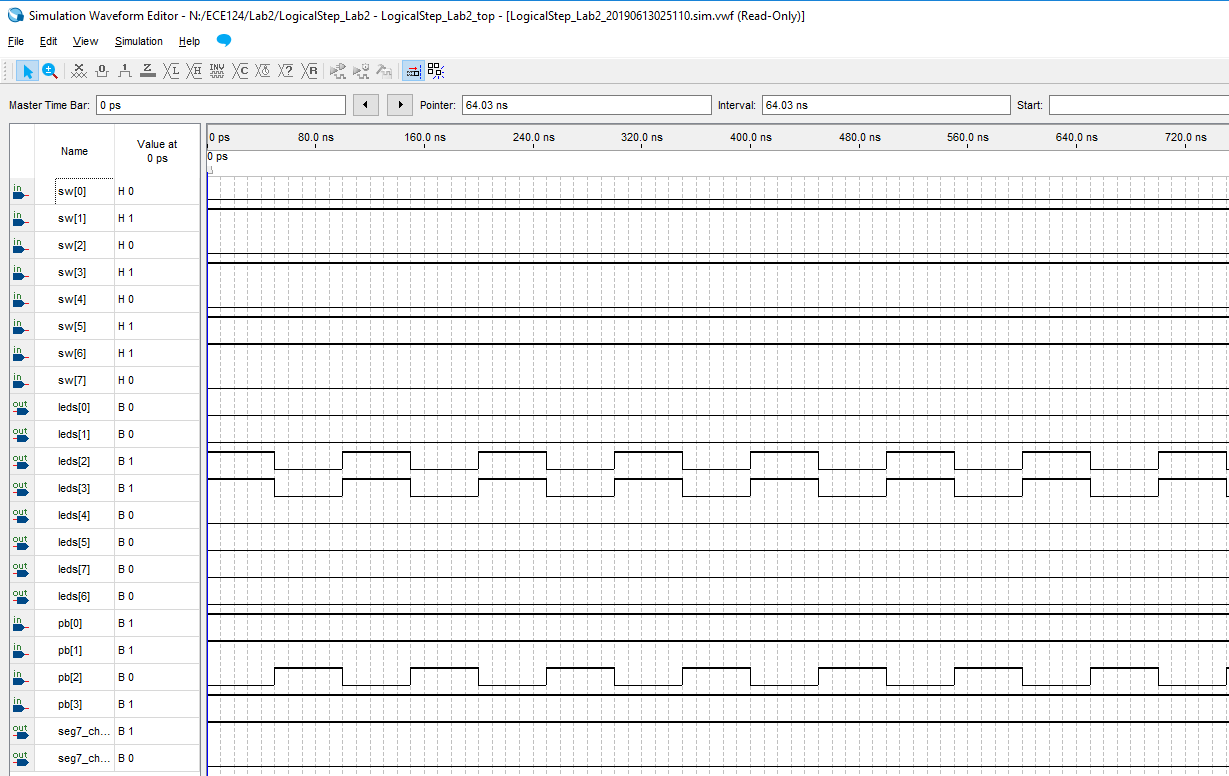


Figure 2: Functional Simulation of XOR operation

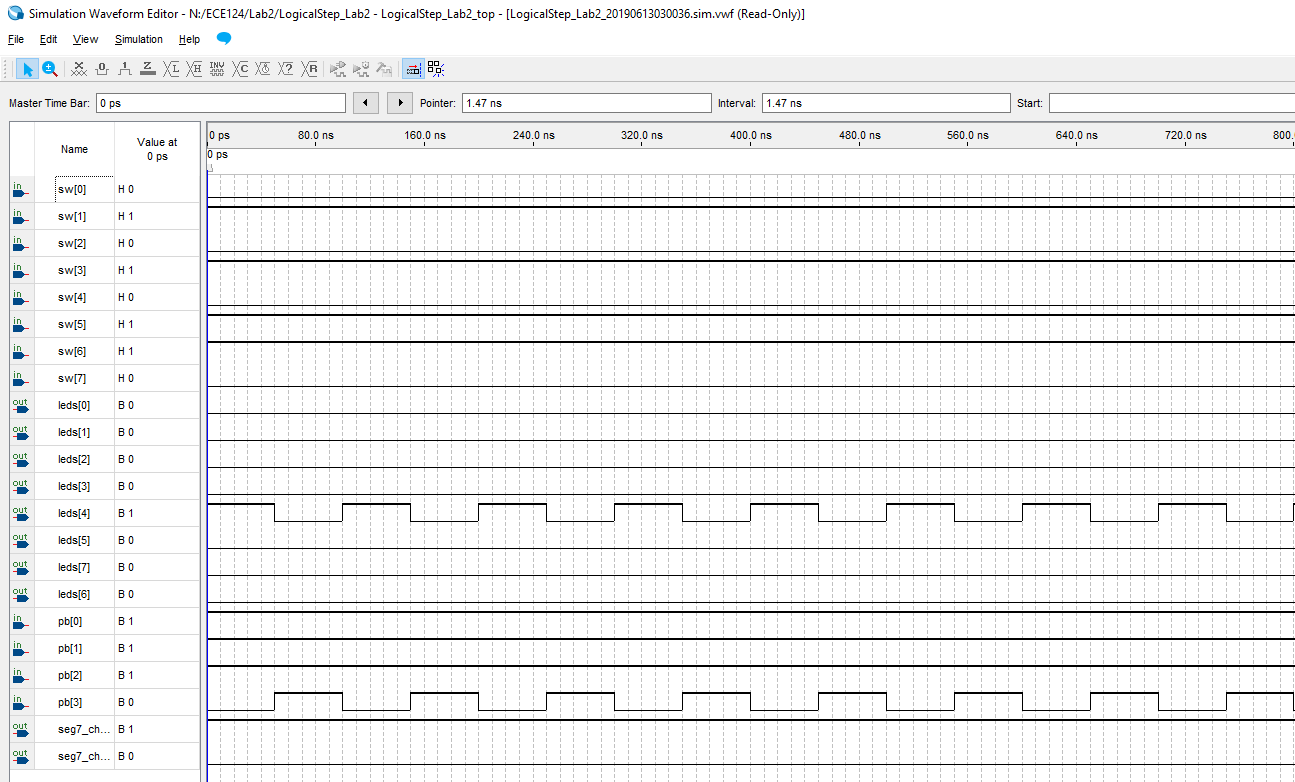


Figure 2: Functional Simulation of ADD operation

