VHDL Assignment 3 Report

Yutong Wang 260839723
Hongtao Xu 260773785

1.Briefly explain your VHDL code implementation of all circuits.

First of all, it is a 4-bit comparator with A and B inputs. I found the circuit of the 4-bit comparator and below is the truth table I found online:

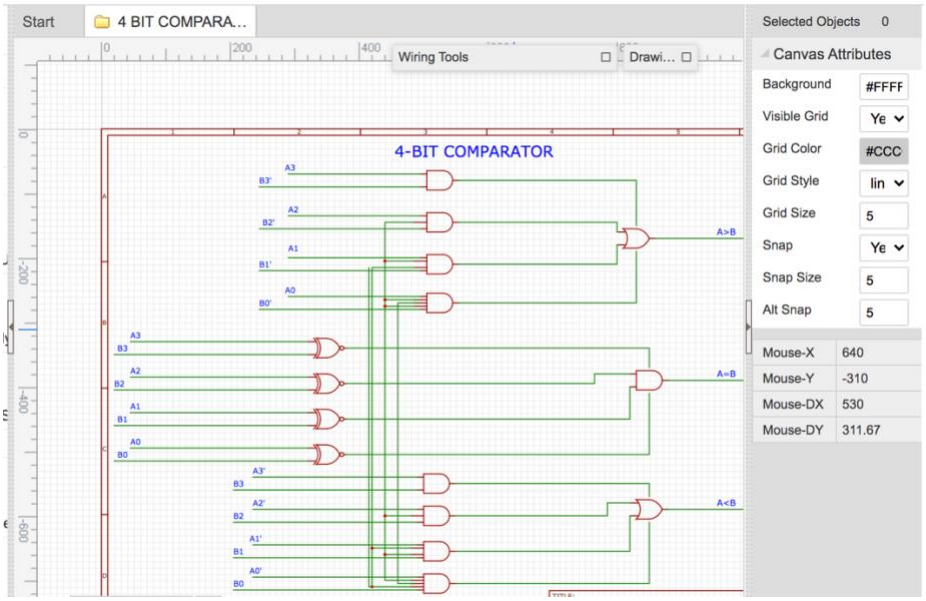


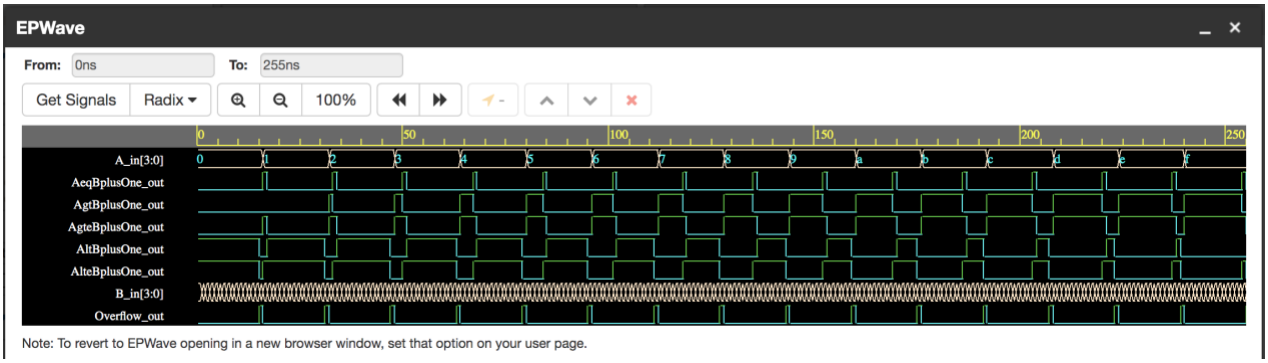
Table 1: Truth table for 4-bit

INPUTS								OUTPUTS		
A3	A2	A1	A0	B3	B2	B1	B0	A > B	A = B	A < B
0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	1	0	0	1
0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	1	0	1	0	0	1
0	0	0	0	0	1	1	0	0	0	1
0	0	0	0	0	1	1	1	0	0	1
0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	1	0	0	1	0	0	1
0	0	0	0	1	0	1	0	0	0	1
0	0	0	0	1	0	1	1	0	0	1
0	0	0	0	1	1	0	0	0	0	1
0	0	0	0	1	1	0	1	0	0	1
0	0	0	0	1	1	1	0	0	0	1
0	0	0	0	1	1	1	1	0	0	1
0	0	0	1	0	0	0	0	1	0	0
0	0	0	1	0	0	0	1	0	1	0
0	0	0	1	0	0	1	0	0	0	1
0	0	0	1	0	0	1	1	0	0	1
⋮			⋮			⋮			⋮	
1	1	1	1	1	1	1	1	0	1	0

From the truth table we know that there are totally 8 inputs (4 for A and 4 for B) and 3 outputs in the right. The 3 cases labelled in the right -hand side: A=B, A>B and A<B. While in this lab, we also need to think about the cases A>=B, A<=B and overflow.

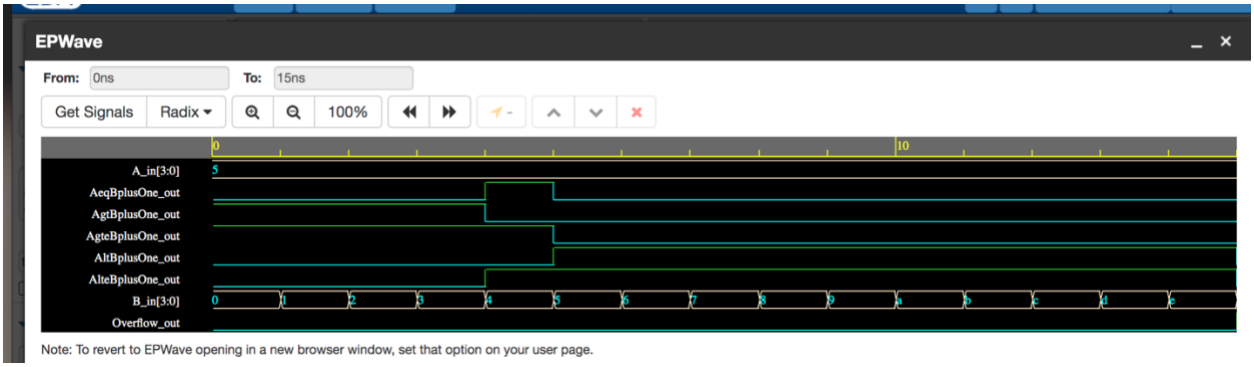
To be specific, A and B are assigned to 4 bits input respectively, they counted as <3 downto 0>. For the output: AgtBplusOne, AgteBplusOne, AltBplusOne, AlteBplusOne and AeqBplusOne. In this way, we can setup the testbench. The number of inputs for A and B are set into 16 loops respectively. A: for m in 0 to 15 loop; B: for n in 0 to 15 loop; a_in <= std_logic_vector(to_unsigned(m, 4)); b_in <= std_logic_vector (to_unsigned(n, 4));

Now is the design part. First of all, there are totally 5 cases: AgtBplusOne; AgteBplusOne; AltBplusOne; AlteBplusOne and AeqBplusOne. For the overflow= '1': B= '1111', if B+1, it would out of space for truth table. In this way, all of those cases call '0' which means overflow. Otherwise, if overflow is not happened (overflow = '0'), we compare the value A and B+1 and select the corresponding cases. For example, if A<B+1 means that AltBplusOne = '1', otherwise AltBplusOne = '0'. Similar for all the other cases. In this way, we use the [if-then-else] statement check all the cases and get the corresponding statement. Finally, we could get the correct EPWave:

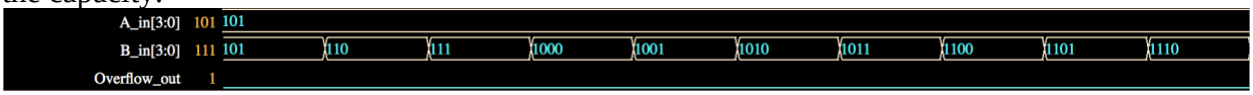


2. Given that A = 510 , provide a separate simulation plot that demonstrates all possible cases for the 4-bit comparator, including a separate plot for the case where overflow occurs. A total of four plots should be included. Explain each plot and mark all inputs and outputs clearly.

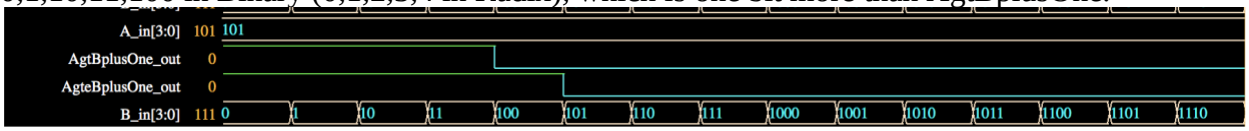
When A=5 with base of 10, the overall EPWave:



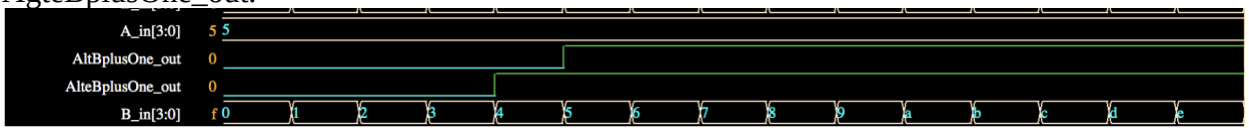
From this plot, we know that overflow would occur at B = ‘1110’, cause B+1 then would out of the capacity.



That’s the plot for A > B+1 and A>= B+1, for AgtBplusOne_out = ‘1’ happens when Bin = 0,1,10,11 in Binary (0,1,2,3 in Radix) and AgteBplusOne_out = ‘1’ happens when Bin = 0,1,10,11,100 in Binary (0,1,2,3,4 in Radix), which is one bit more than AgtBplusOne.

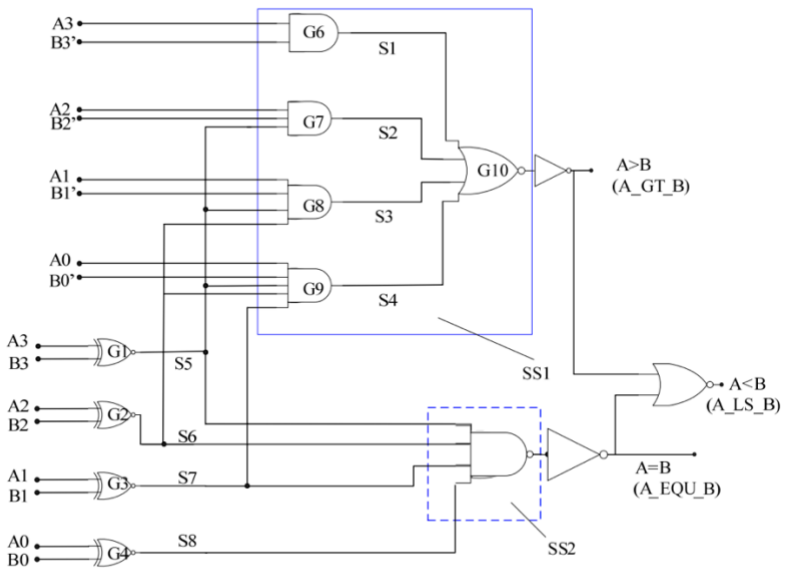


That’s the plot for A < B+1 and A<= B+1. For AltBplusOne_out = ‘1’ happens since Bin = 5 (101) and AlteBplusOne_out = ‘1’ happens since Bin = 4, which is still one more number than AltBplusOne_out. It is noticed that Bin=4 is commonly shared by both AlteBplusOne_out and AgteBplusOne_out.



That’s the plot for A = B+1 happens when Bin = 4, is also commonly shared by both AlteBplusOne_out and AgteBplusOne_out.

3.Perform timing analysis of the 4-bit comparator and find the critical path(s) of the circuit. What is the delay of the critical path(s)?



Optimized logic diagram for a 4-bit comparator

This is the optimized diagram of a 4-bit comparator I found online.

```
# Info: [670]: Finished synthesizing design.
# Info: [11019]: Total CPU time for synthesis: 0.6 s secs.
# Info: [11020]: Overall running time for synthesis: 0.6 s secs.
# Info: /home/runner/impl_1/precision_tech.sdc
# Info: [44856]: Total lines of RTL compiled: 58.
# Info: [44835]: Total CPU time for compilation: 0.0 secs.
# Info: [44513]: Overall running time for compilation: 1.0 secs.
# Info: [667]: Current working directory: /home/runner/impl_1.
# Info: [15333]: Doing rtl optimizations.
```

From the log on EDA playground I found that the running timing is 0.6s.
From the above circuit I found that the critical path is going through a NOR gate, then enter to a AND gate, enter to a OR gate and a NOT gate, finally going through a OR gate. The critical path would go through 5 gates.
From the above log screenshot we calculate delay of critical path = 1s-0.6s = 0.4s

4. Report the number of pins and logic modules used to fit your 4-bit comparator design on the FPGA board.

The number of pins: used:14 and Avial 210.
The number of logic modules: used:9 and Avial:63400

Reference:
http://users.encs.concordia.ca/~asim/COEN_6511/Projects/final6511report.pdf
https://easyeda.com/ankitjain21/4-bit-comparator