Homework 3

Note: Include all Verilog codes in your report

1-) Design a circuit, COMP_1_bit, for comparing two 1-bit numbers, a and b. The outputs are (x, y, z) defined below:

- if a>b; x=1, y=0, z=0
- if a=b; x=0, y=1, z=0
- if a<b; x=0, y=0, z=1

a) Write a Verilog code for COMP_1_bit. Save this file by giving name as “COMP_1_bit.v”. Elaborate this module in Cadence Genus and add the resulting schematic to your report. Examine what type of elements are used to model the written HDL code.

b) Write a test bench file with name “COMP_1_bit_tb.v” to test your design. Perform a behavioral simulation for your design in Cadence Xcelium, covering each possible cases. Add the result to your report to prove that the circuit is working as expected.

2-) Design a circuit, POZ_COMPARE, for comparing two 4-bit numbers, A and B. The block diagram for comparing 4-bit numbers is shown in Figure 1. The outputs are (X, Y, Z) defined below:

- if A>B, then X=1, Y=0, Z=0
- if A=B, then X=0, Y=1, Z=0
- if A<B, then X=0, Y=0, Z=1

Figure 1 is the block diagram of the circuit which is used to calculate the following equations:

$$X = x_3 + y_3x_2 + y_3y_2x_1 + y_3y_2y_1x_0$$

$$Y = y_3y_2y_1y_0$$

$$Z = z_3 + y_3z_2 + y_3y_2z_1 + y_3y_2y_1z_0$$

a) Write a Verilog code for the CONNECT block in Figure 1. Save this file by giving name as “CONNECT.v”

b) Write a Verilog code for the CON_COMP_1_bit block by using COMP_1_bit and CONNECT as building blocks. Save this file by giving name as "CON_COMP_1_bit.v".

c) Write a Verilog code for POZ_COMPARE by using CON_COMP_1_bit as the building block. Save this file by giving name as "POZ_COMPARE.v". Elaborate this module in Cadence Genus and add the resulting schematic to your report. Examine what type of elements are used to model the written HDL code.

d) Write a test bench file with name "POZ_COMPARE_tb.v" to test your design. Perform a <u>behavioral simulation</u> for your design in Cadence Xcelium, using at least three input instances for each three possible cases. Add the result to your report to prove that the circuit is working as expected.

3-) Design 4-to-1 line MUX.

a) Write a Verilog code for MUX. Save this file by giving name as "MUX.v"

b) Write a test bench file with name "MUX_tb.v" to test your design. Perform a <u>behavioral simulation</u> for your design in Cadence Xcelium. Add the result to your report to prove that the circuit is working as expected.

4-) Design a circuit, COMPARE, for comparing two 8-bit signed numbers,
$A=(a_7 , a_6 , a_5 , a_4 ,a_3, a_2, a_1 , a_0)_2$, $B=(b_7 , b_6 , b_5 ,b_4 , b_3 , b_2 ,b_1, b_0)_2$.
The ouputs (X, Y, Z) are defined as below:

- If A≥ 0, B≥0
    - ➢ A>B, then X=1, Y=0, Z=0
    - ➢ A=B, then X=0, Y=1, Z=0
    - ➢ A<B, then X=0, Y=0, Z=1

- If A≥ 0, B<0, then X=1, Y=0, Z=0

- If A<0, B≥0, then X=0, Y=0, Z=1
- If A<0, B<0
    - ➢ A>B, then X=1, Y=0, Z=0
    - ➢ A=B, then X=0, Y=1, Z=0

- A<B, then X=0, Y=0, Z=1

a) Write a Verilog code for COMPARE by using POZ_COMPARE and MUX as building blocks. Save this file by giving name as "COMPARE.v". Elaborate this module in Cadence Genus and add the resulting schematic to your report.

b) Write a test bench file with name "COMPARE_tb.v" to test your design. Perform a behavioral simulation for your design in Cadence Xcelium, using at least three input instances for each three possible cases. Add the result to your report to prove that the circuit is working as expected.

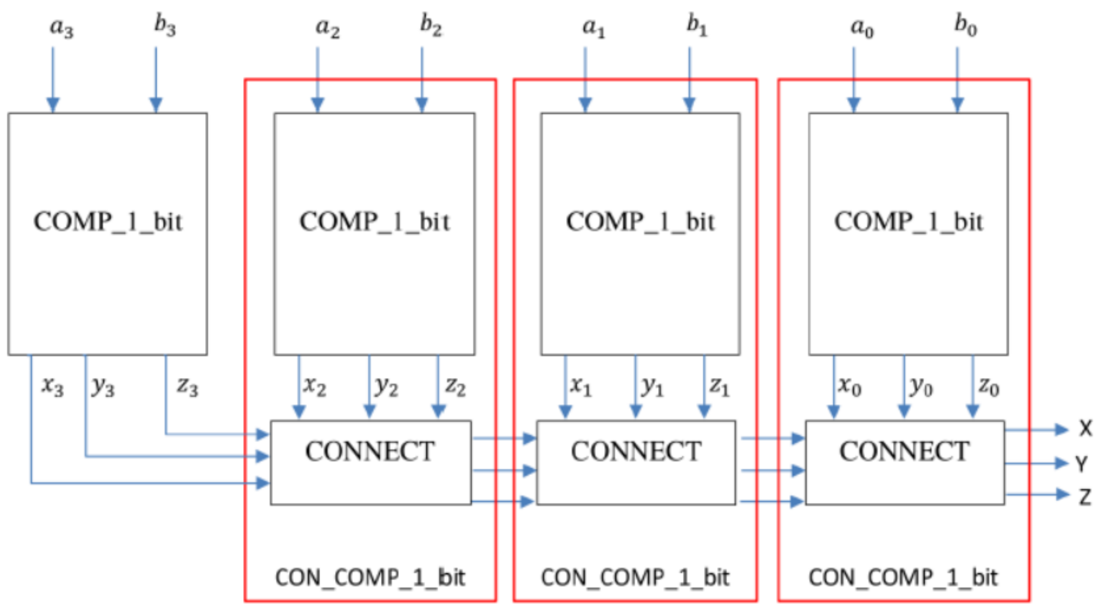

Figure 1