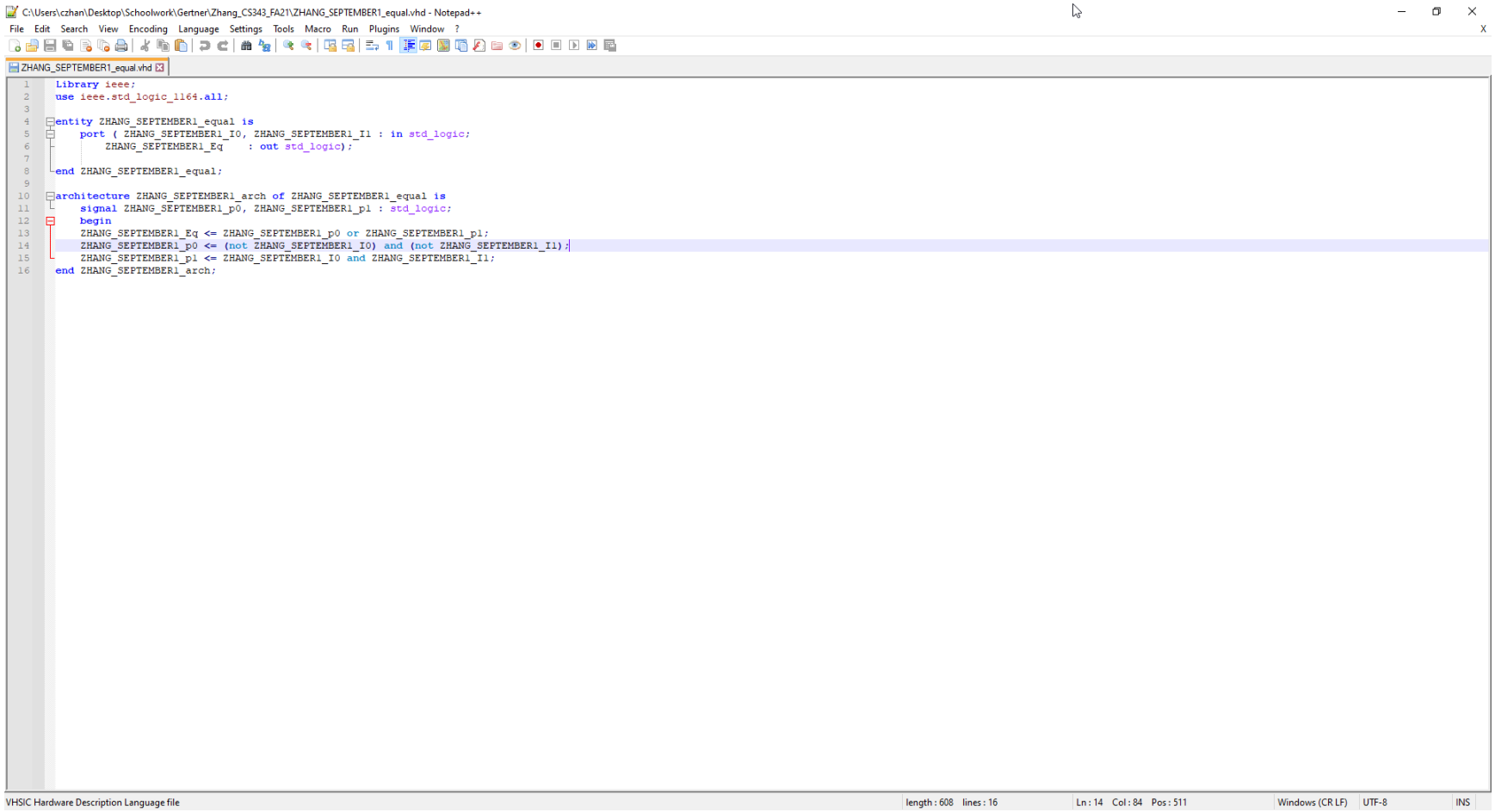
Laboratory Project 2: Comparators

Zhang Chue

9/1/2021

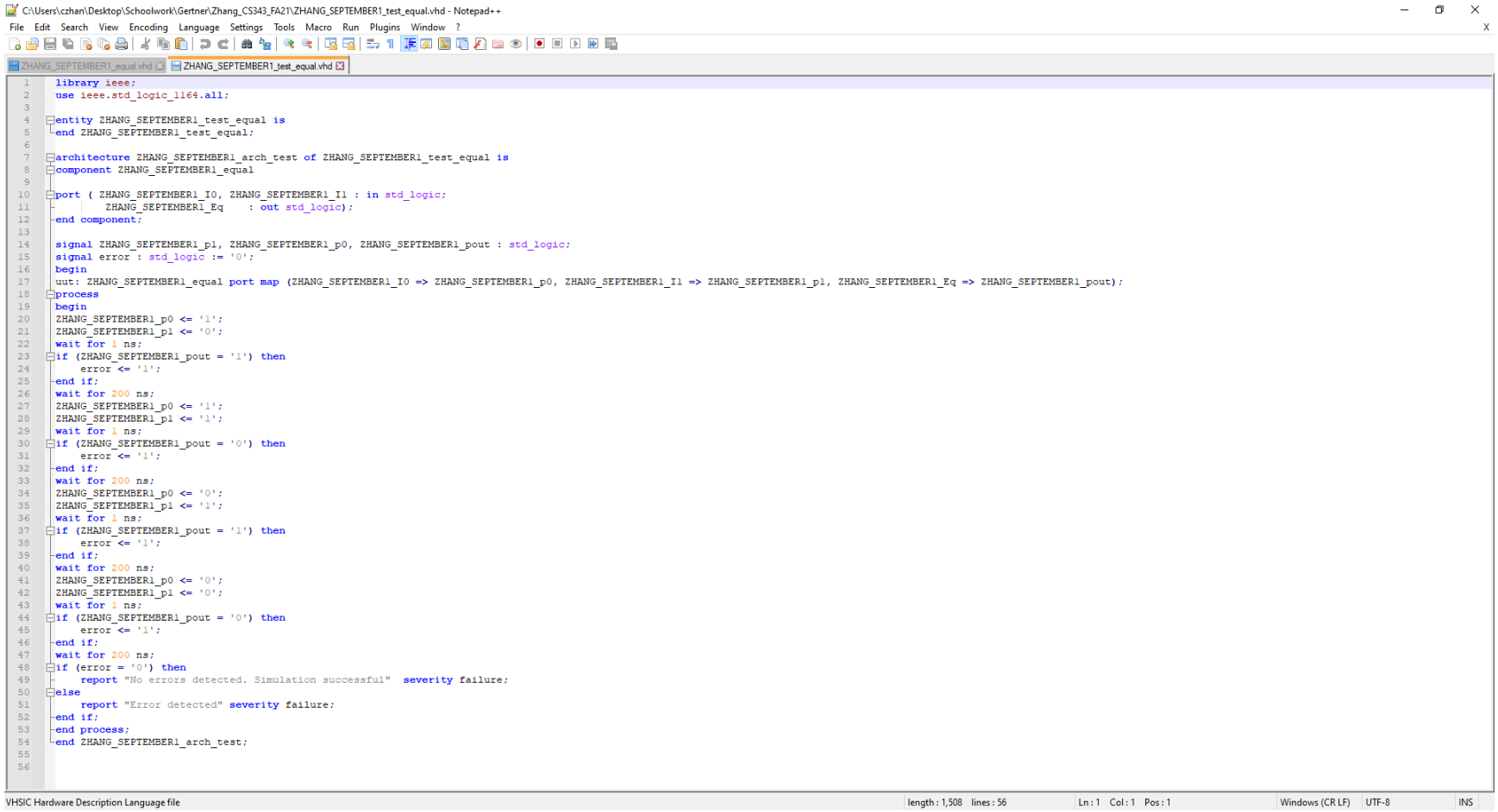
Cs343/Cs342, Professor Isidor Gertner

Task 0 : 1-bit comparator , Chue Zhang



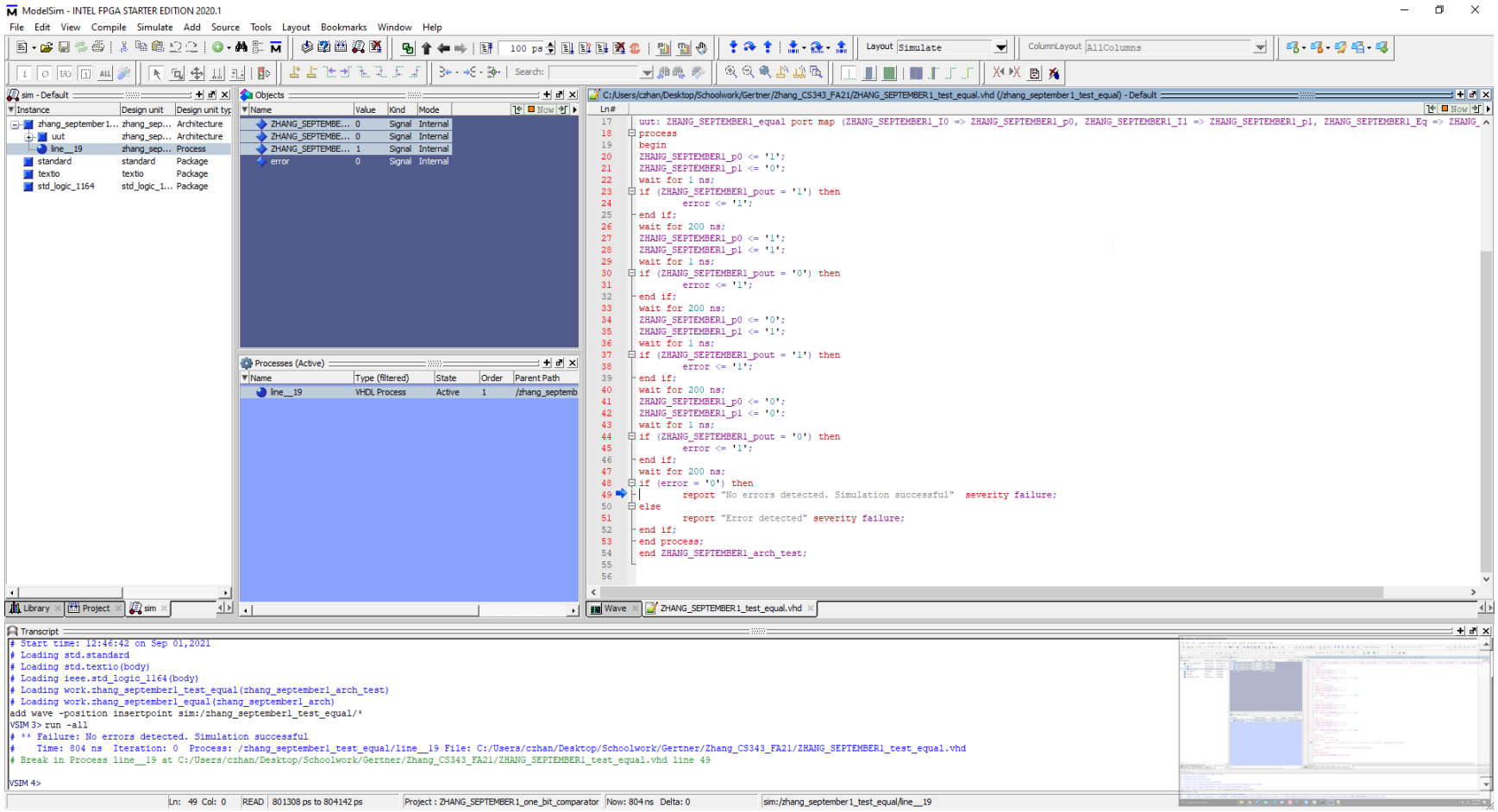
In here is the 1-bit comparator VHDL code that is unoptimized

Task 0 : 1-bit comparator , Chue Zhang



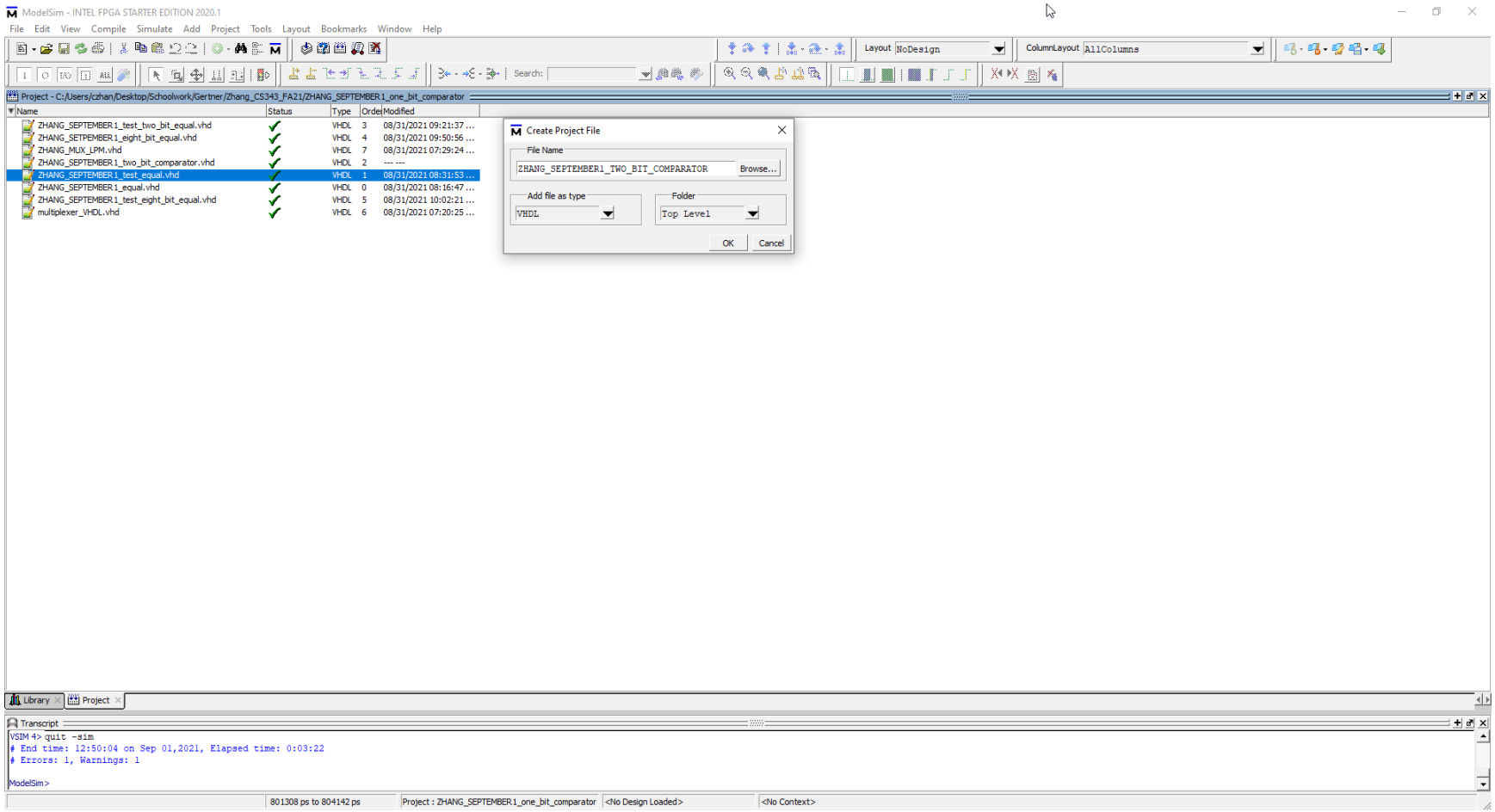
This here is the VHDL code for the test bench of the 1-bit comparator

Task 0 : 1-bit comparator , Chue Zhang



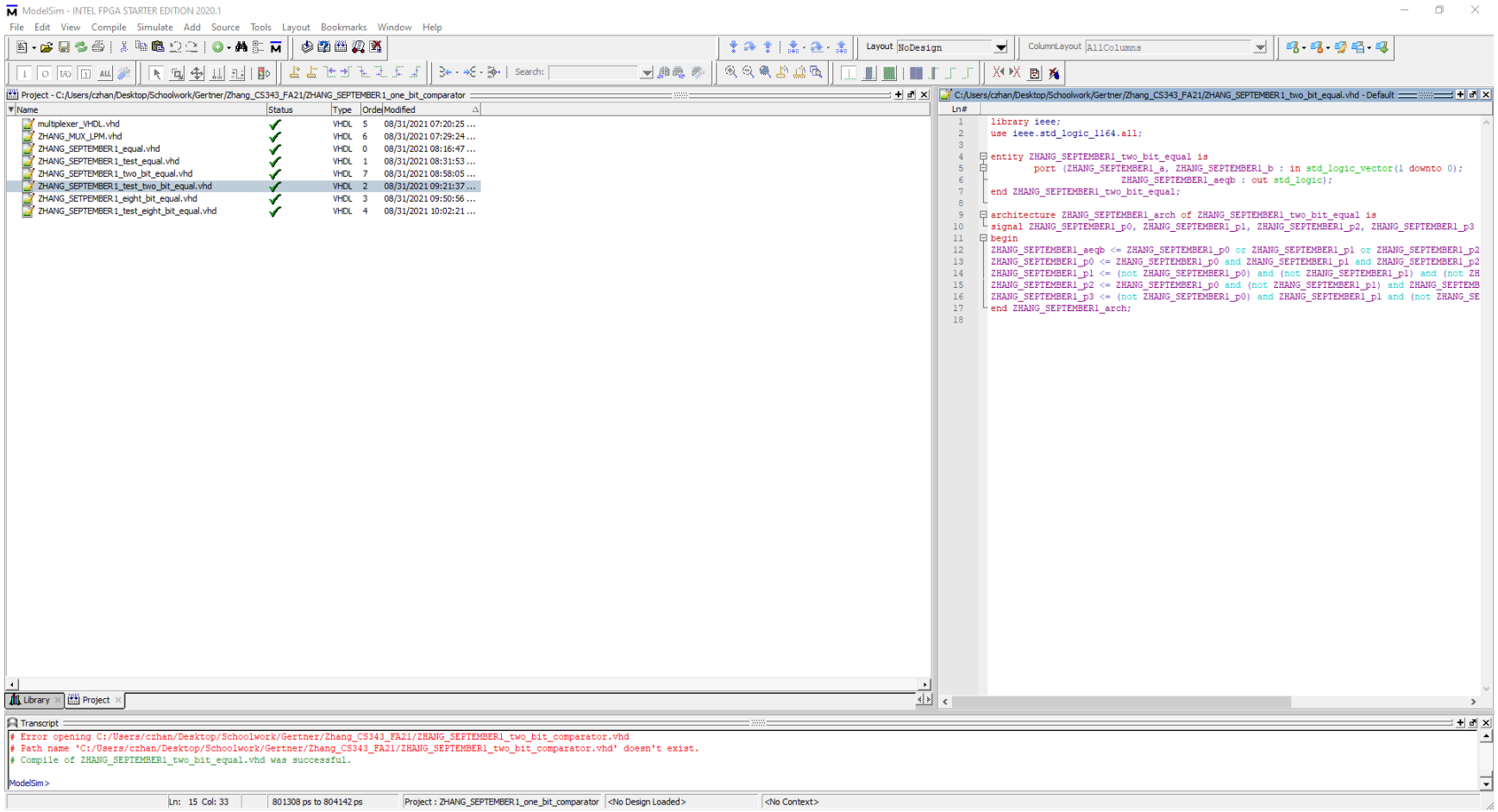
Verification of correctness through the use of test bench. In VHDL code, there is an arrow pointing to the line stating that there were no errors detected.

Task 1, Chue Zhang



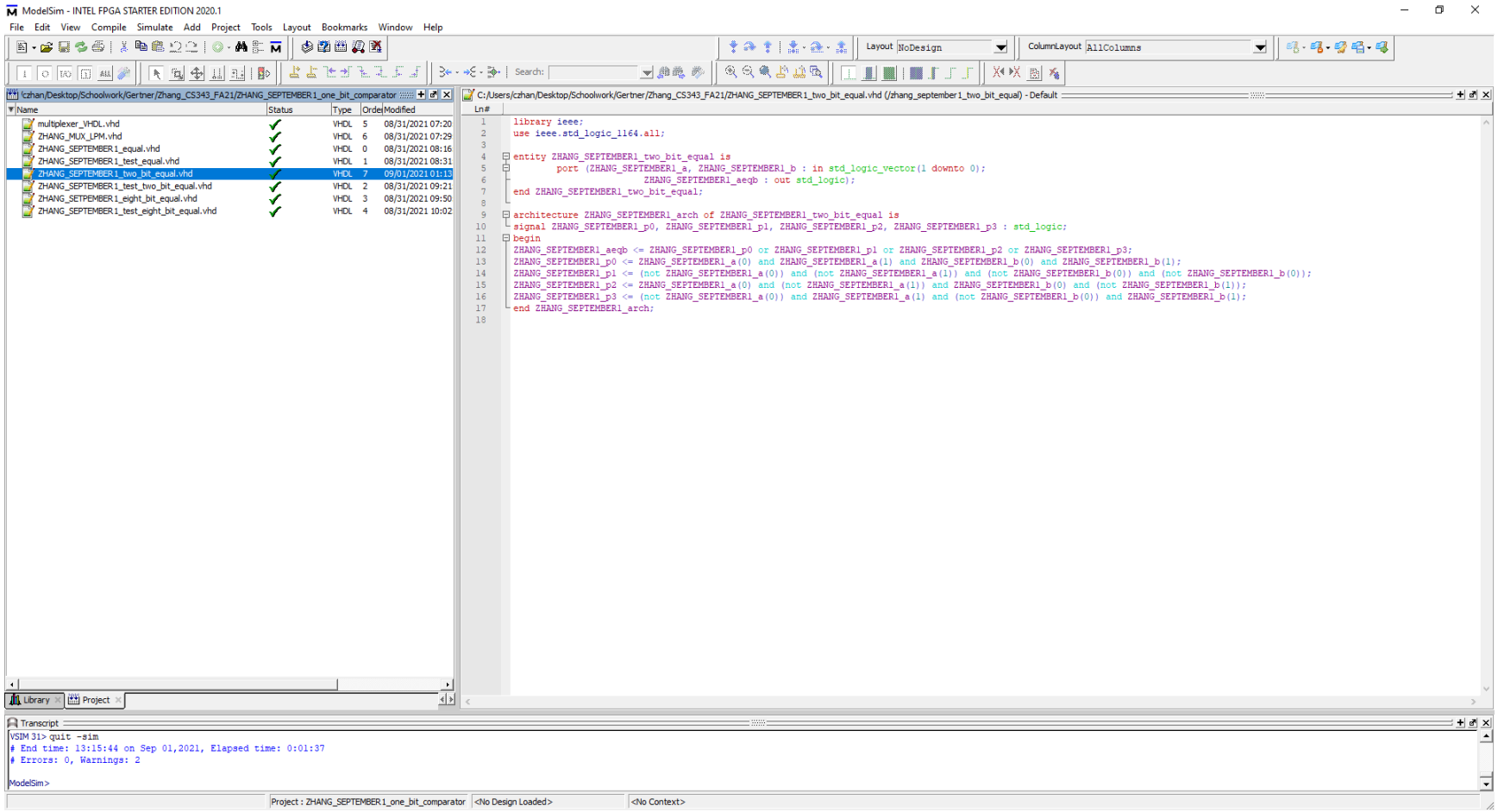
I first create a new file for the 2-bit comparator

Task 1 : 2-bit Comparator, Chue Zhang



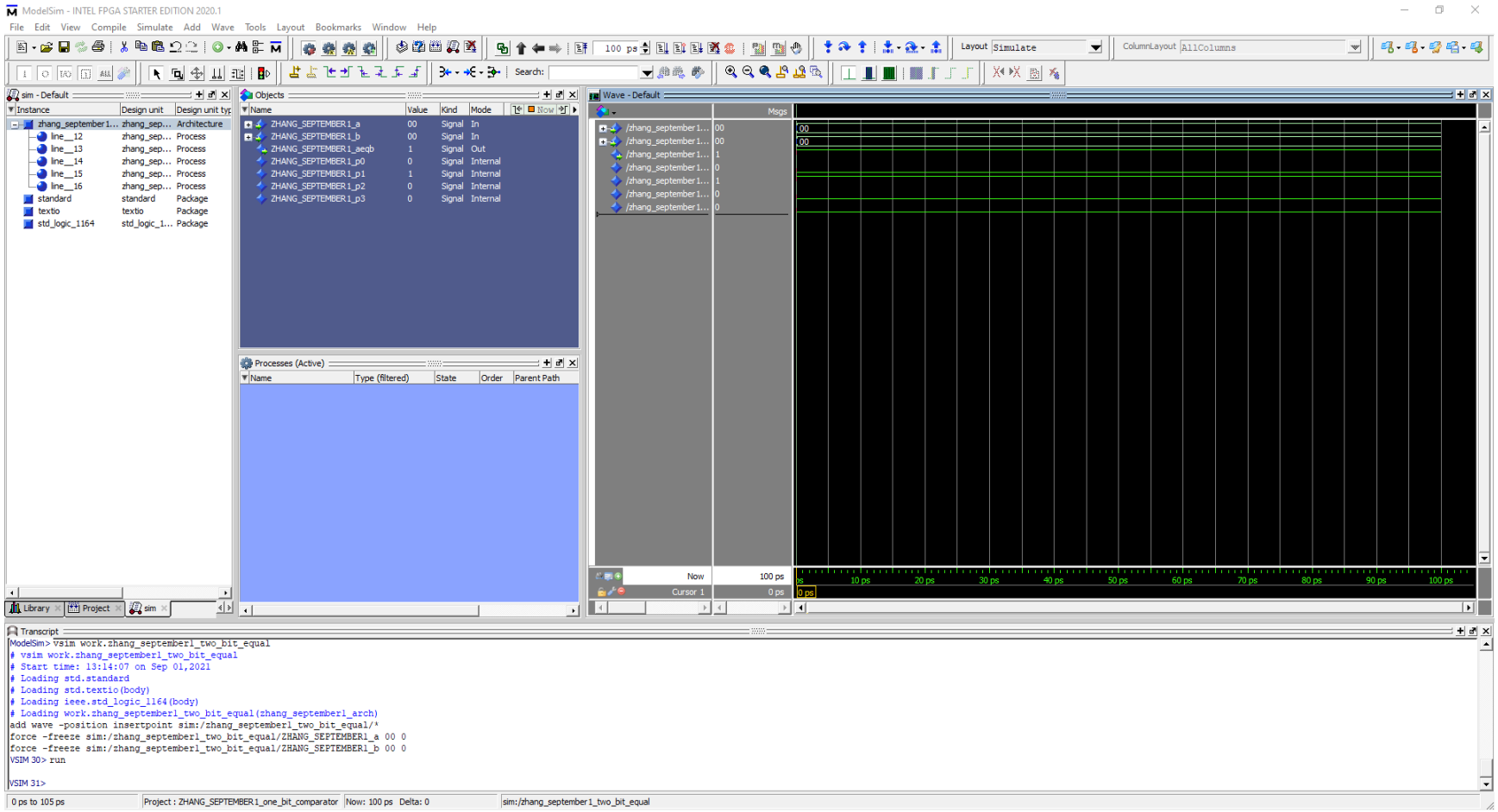
I double click the file and begin writing my VHDL code as seen on the right

Task 1 : 2-bit Comparator, Chue Zhang



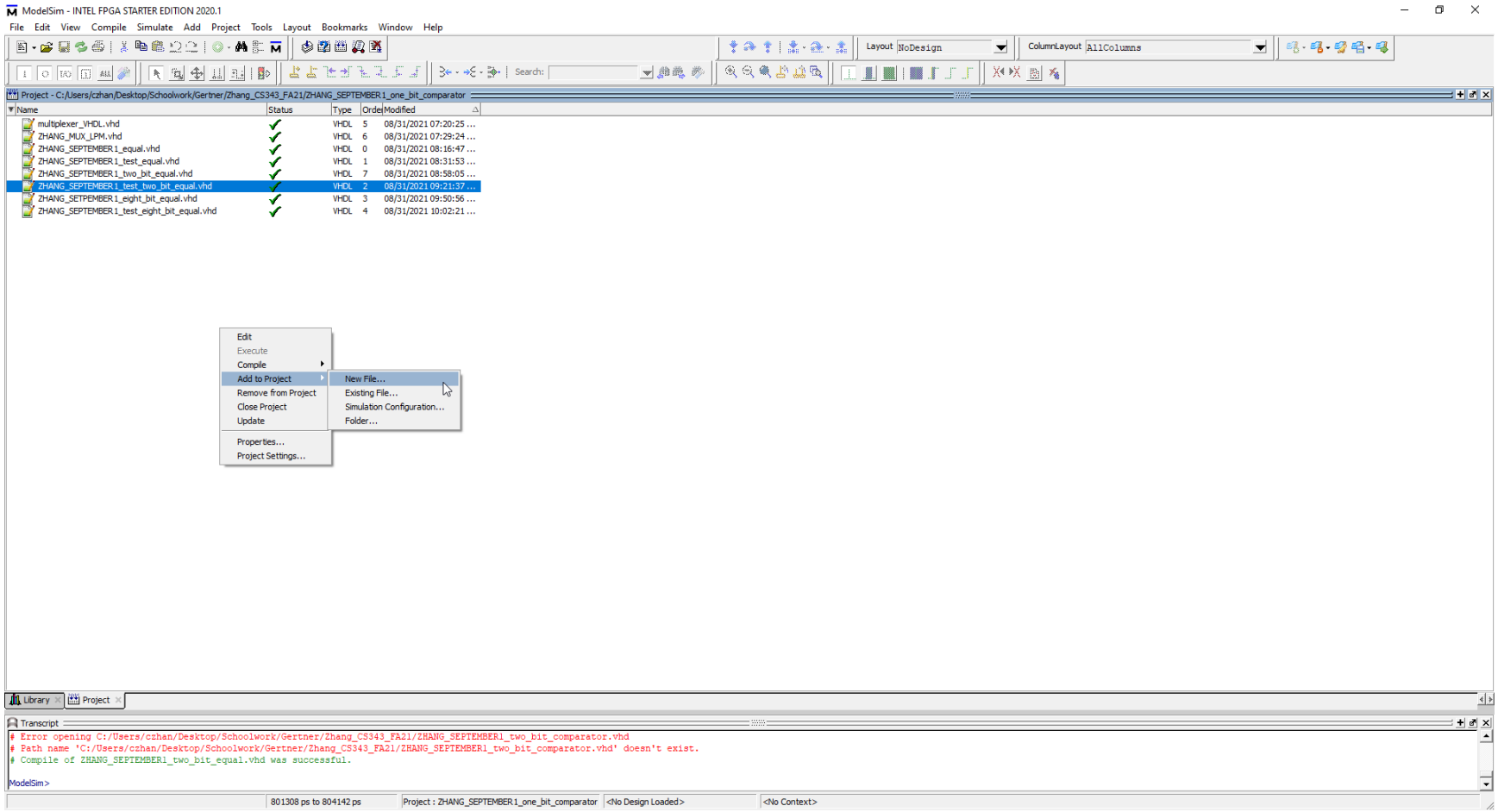
I select the two bit comparator VHDL file and right click → compile → compile selected. At the bottom it should show GREEN with no errors stating that the VHDL file was compiled successfully

Task 1 : 2-bit Comparator, Chue Zhang



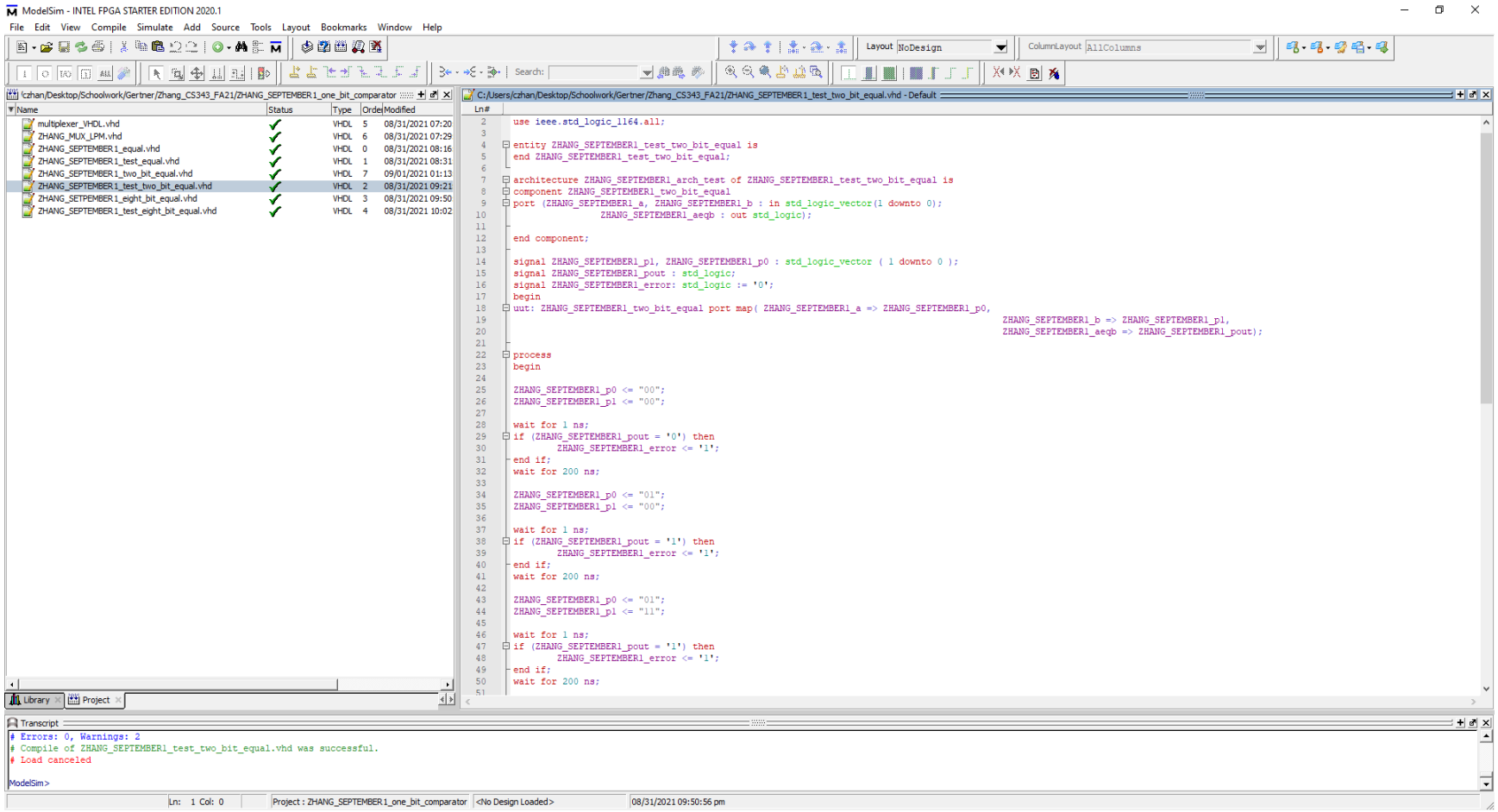
In this figure, what is shown is the waveform model for testing the inputs 00,00 which shows the appropriate output

Task 1 : 2-bit Comparator, Chue Zhang



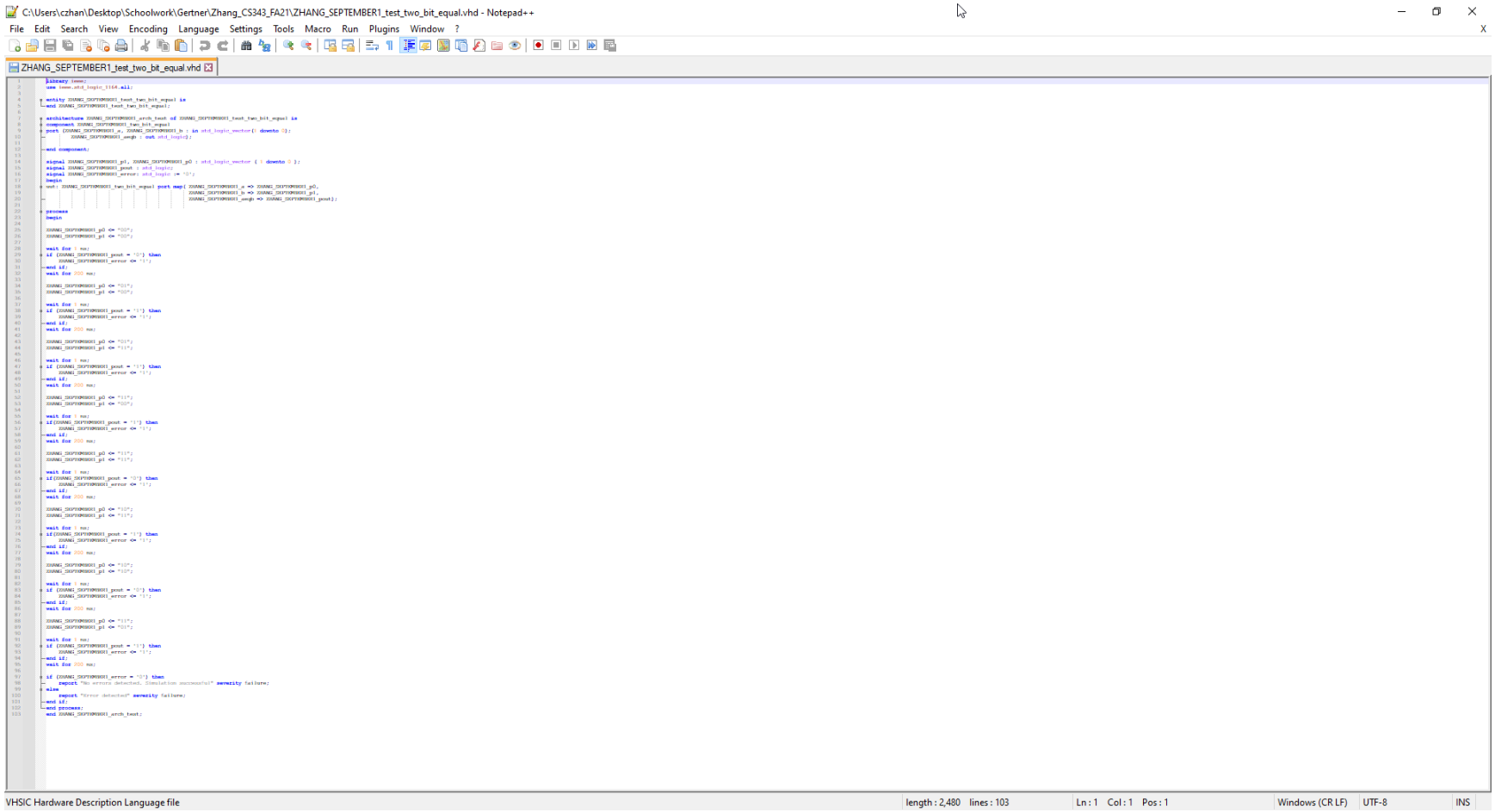
I am now going to create a new file for the test bench

Task 1 : 2-bit Comparator, Chue Zhang



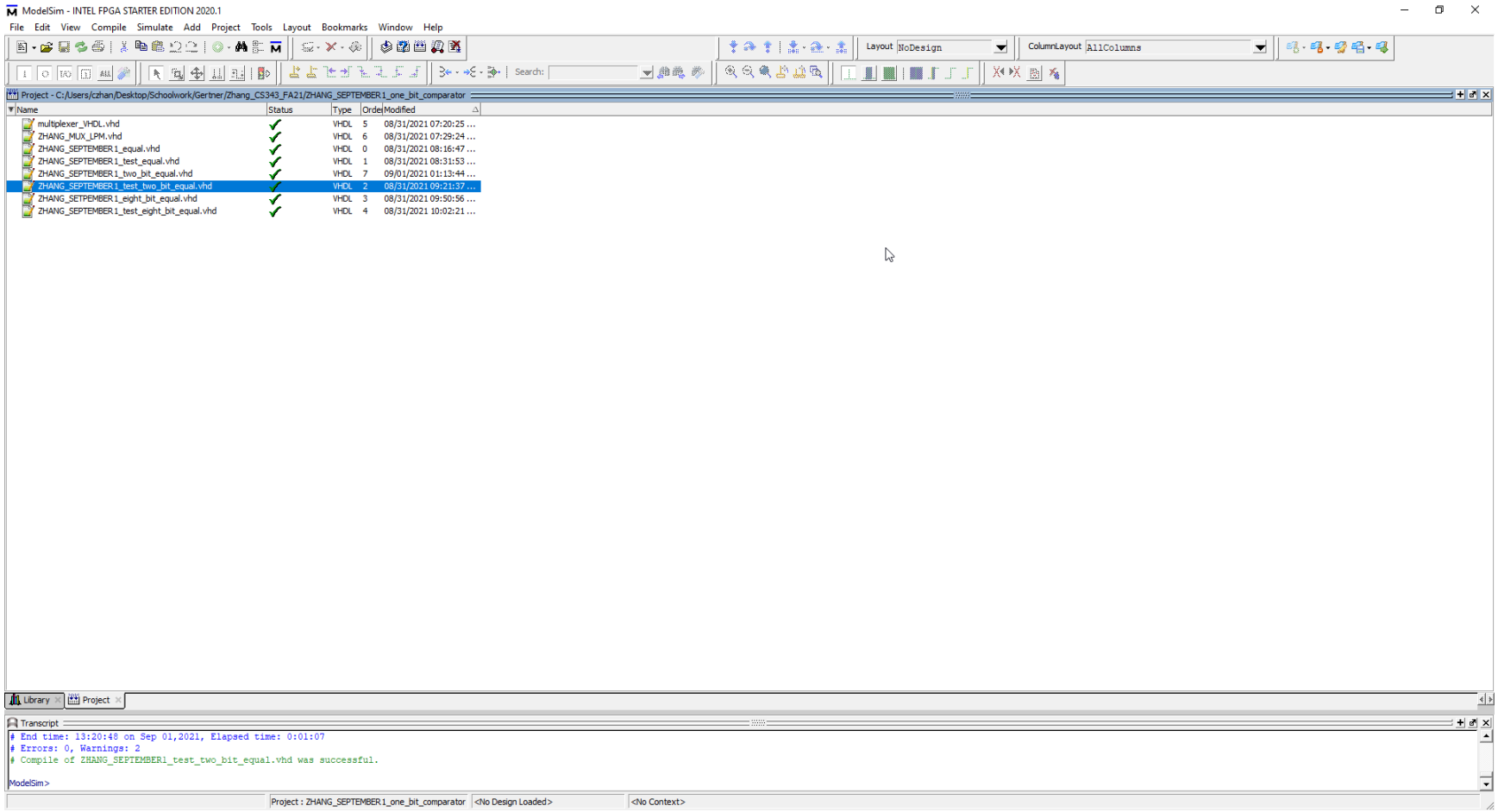
Over here, I have the written test bench code and I will now proceed to compile to check for correctness in code.

Task 1 : 2-bit Comparator, Chue Zhang



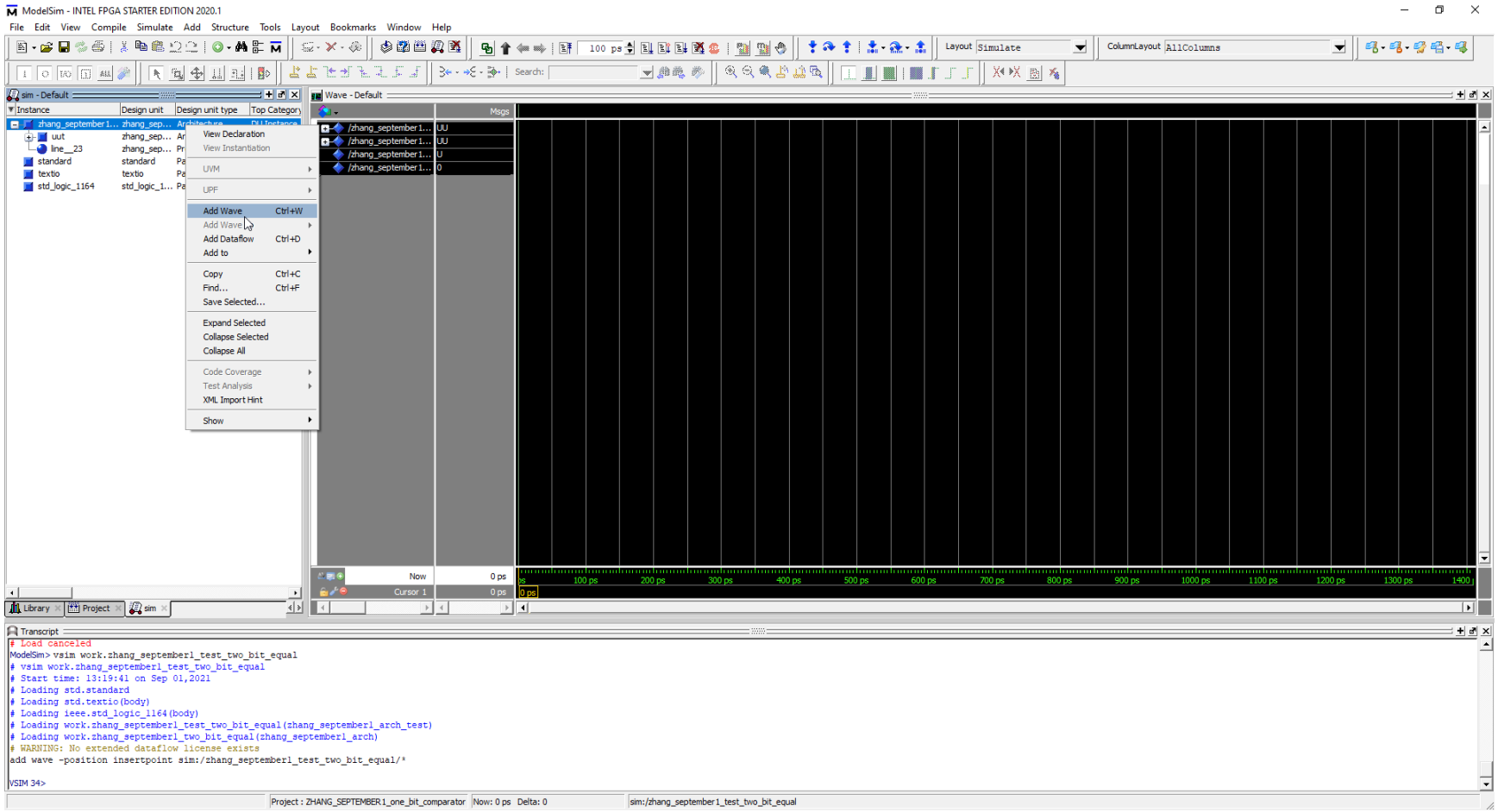
This is the entire code in a screen shot

Task 1 : 2-bit Comparator, Chue Zhang



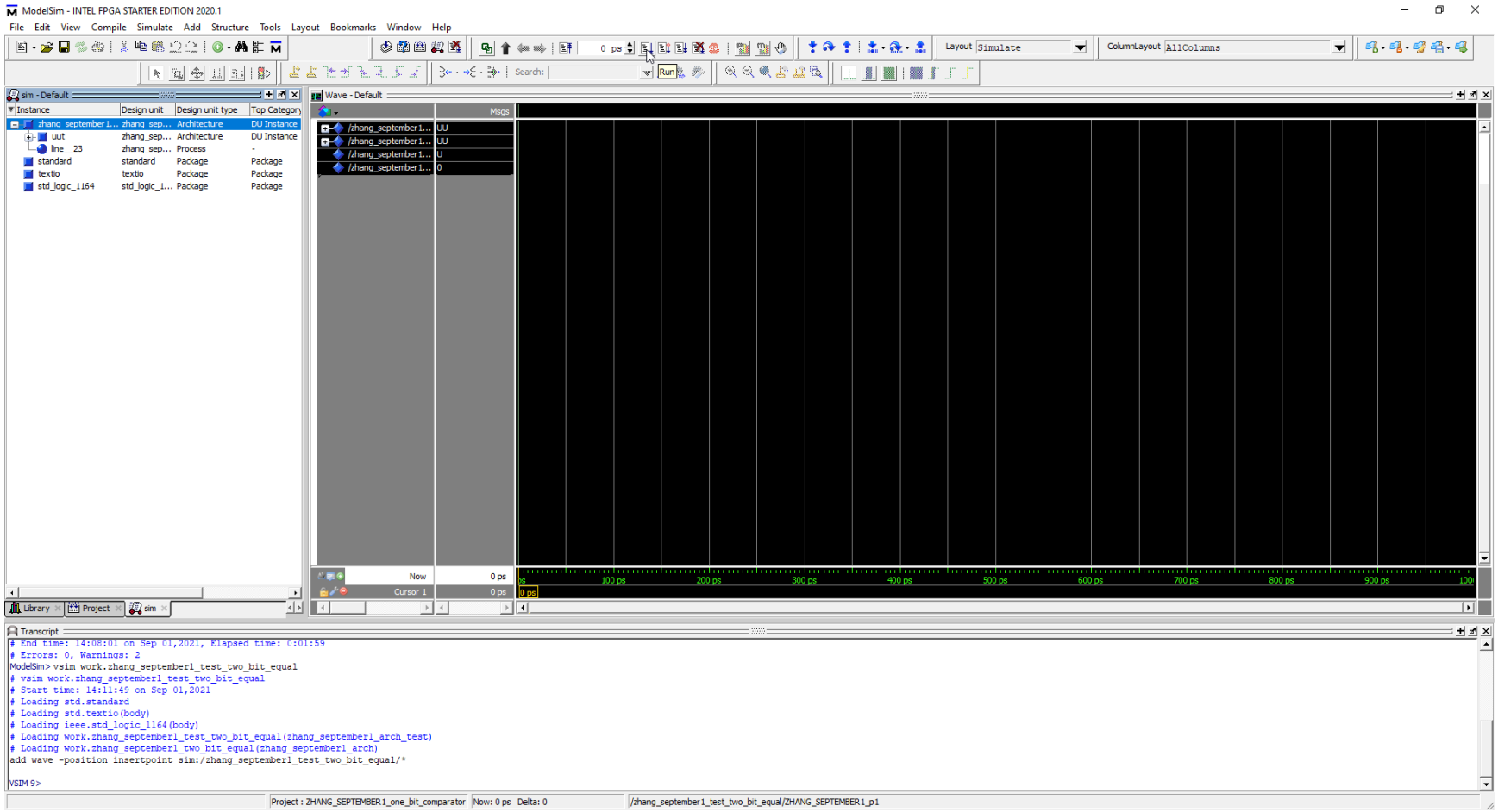
Compilation successful, will proceed to simulation on model sim

Task 1 : 2-bit Comparator, Chue Zhang



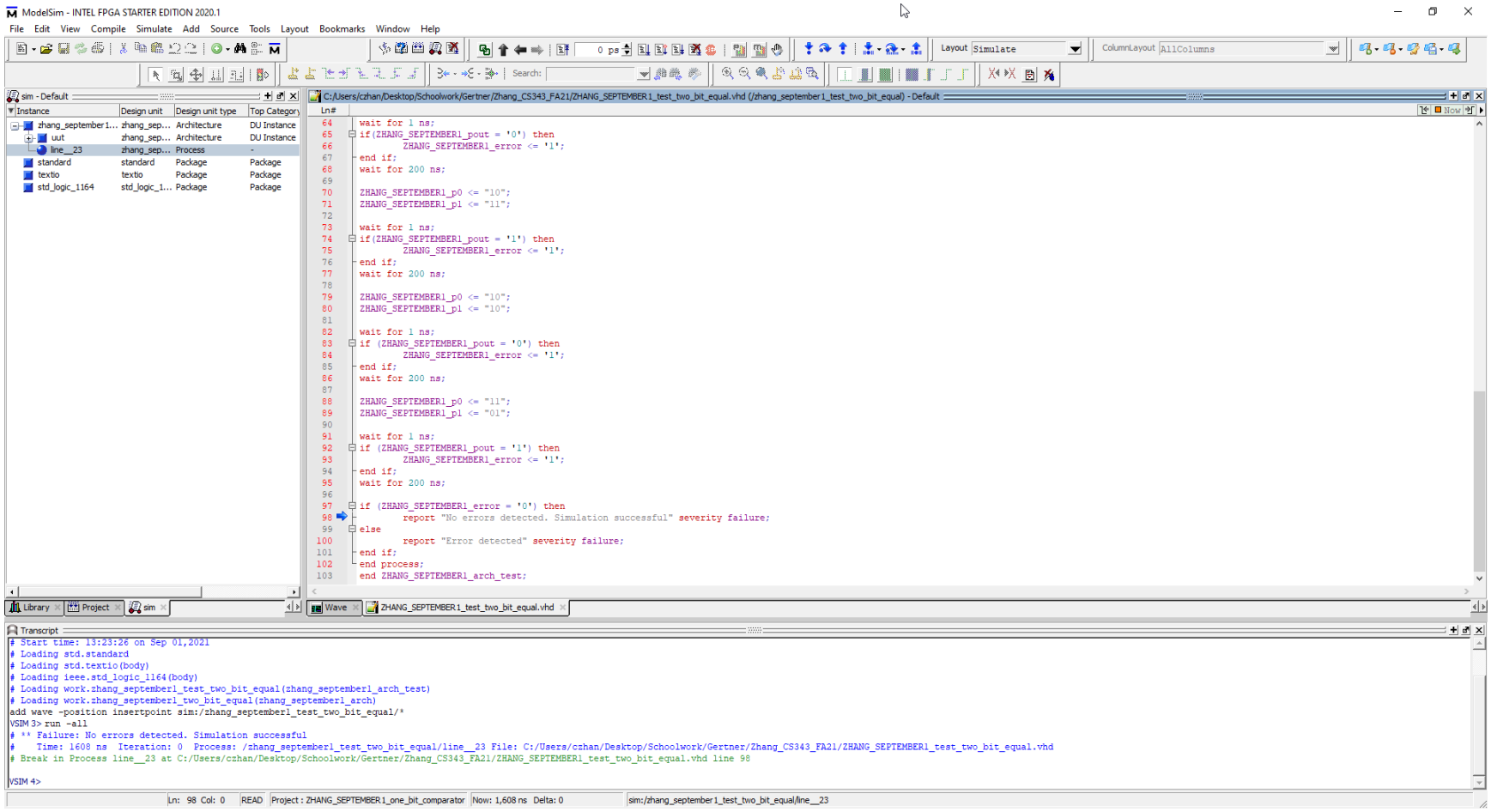
In modelsim I add the waves by pressing right click → add wave

Task 1 : 2-bit Comparator, Chue Zhang



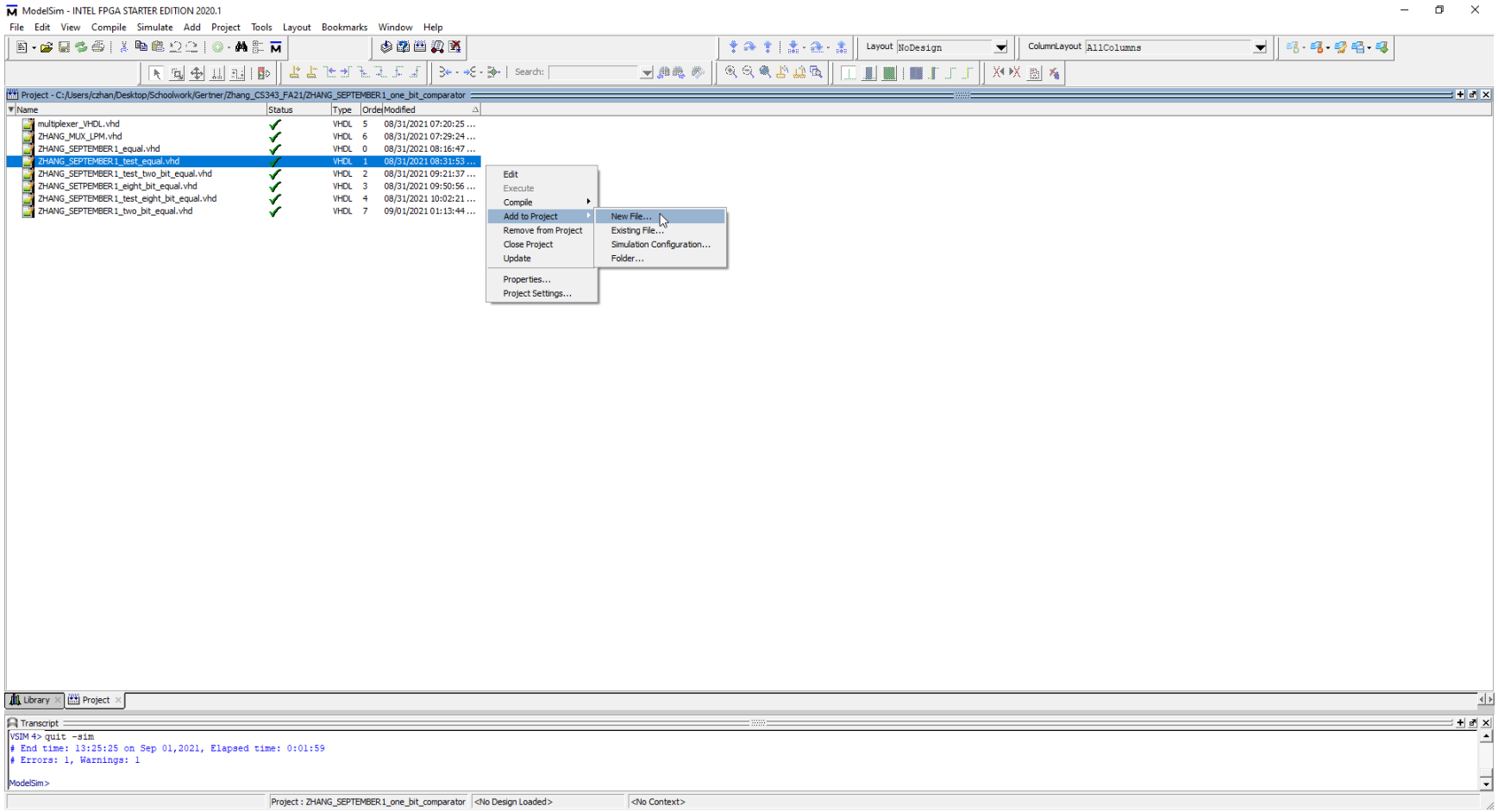
Appropirate waves have been added and I proceed to run-all

Task 1 : 2-bit Comparator, Chue Zhang



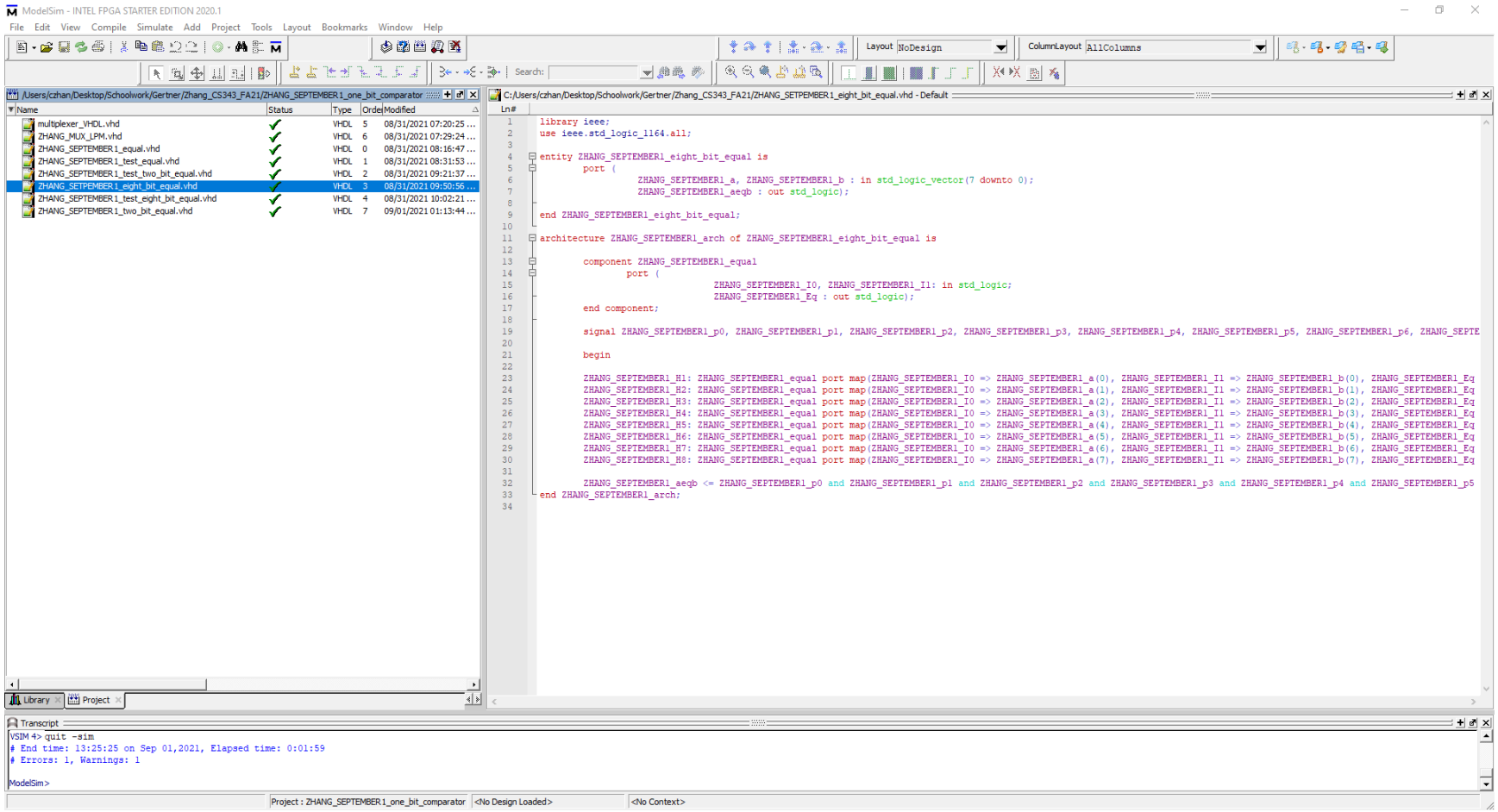
I press run all and there is no errors created so test bench is successful.

Task 2 : 8-bit Comparator , Chue Zhang



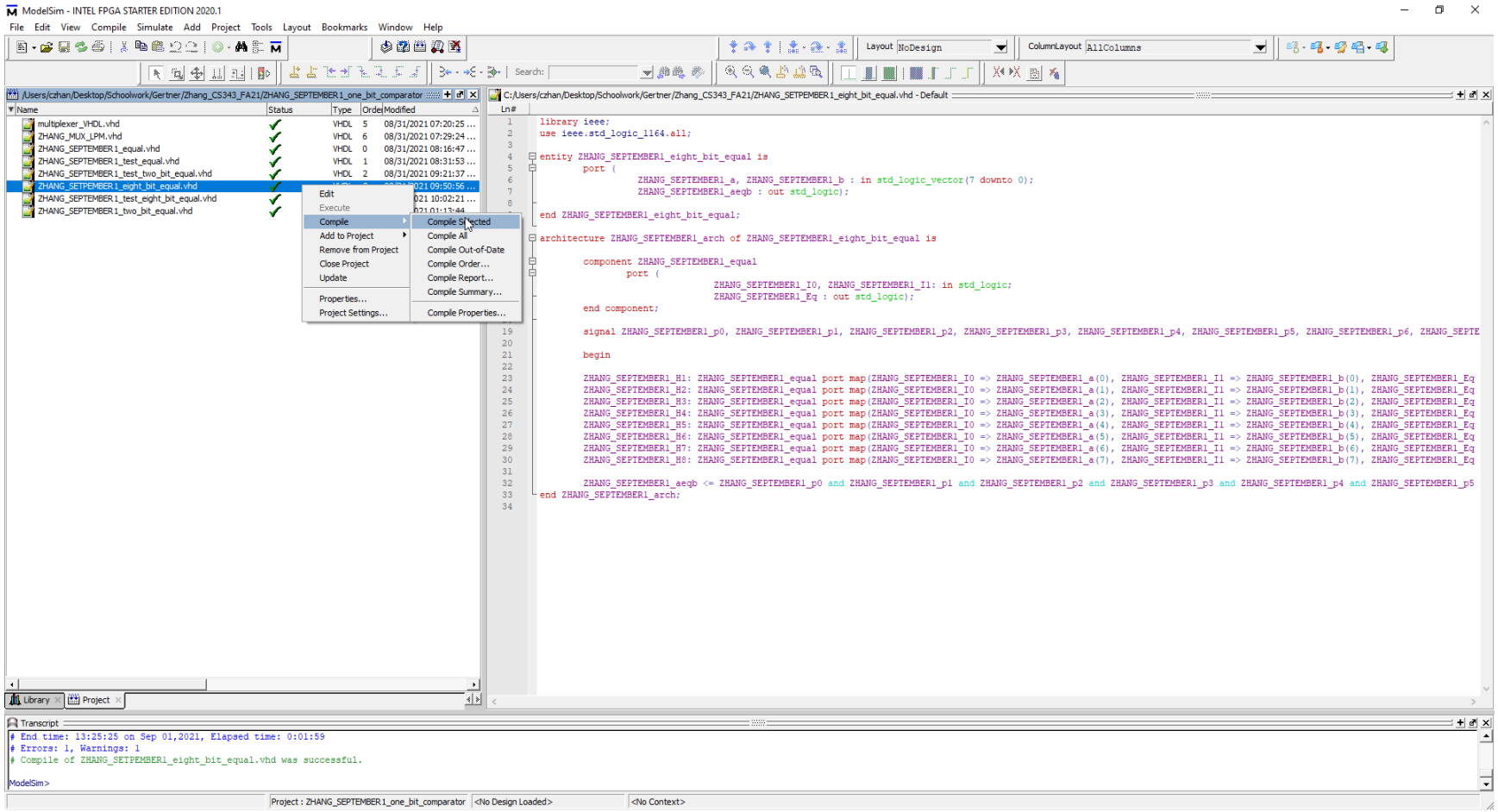
I am now going to create a new VHDL file for 8 bit comparator

Task 2 : 8-bit Comparator , Chue Zhang



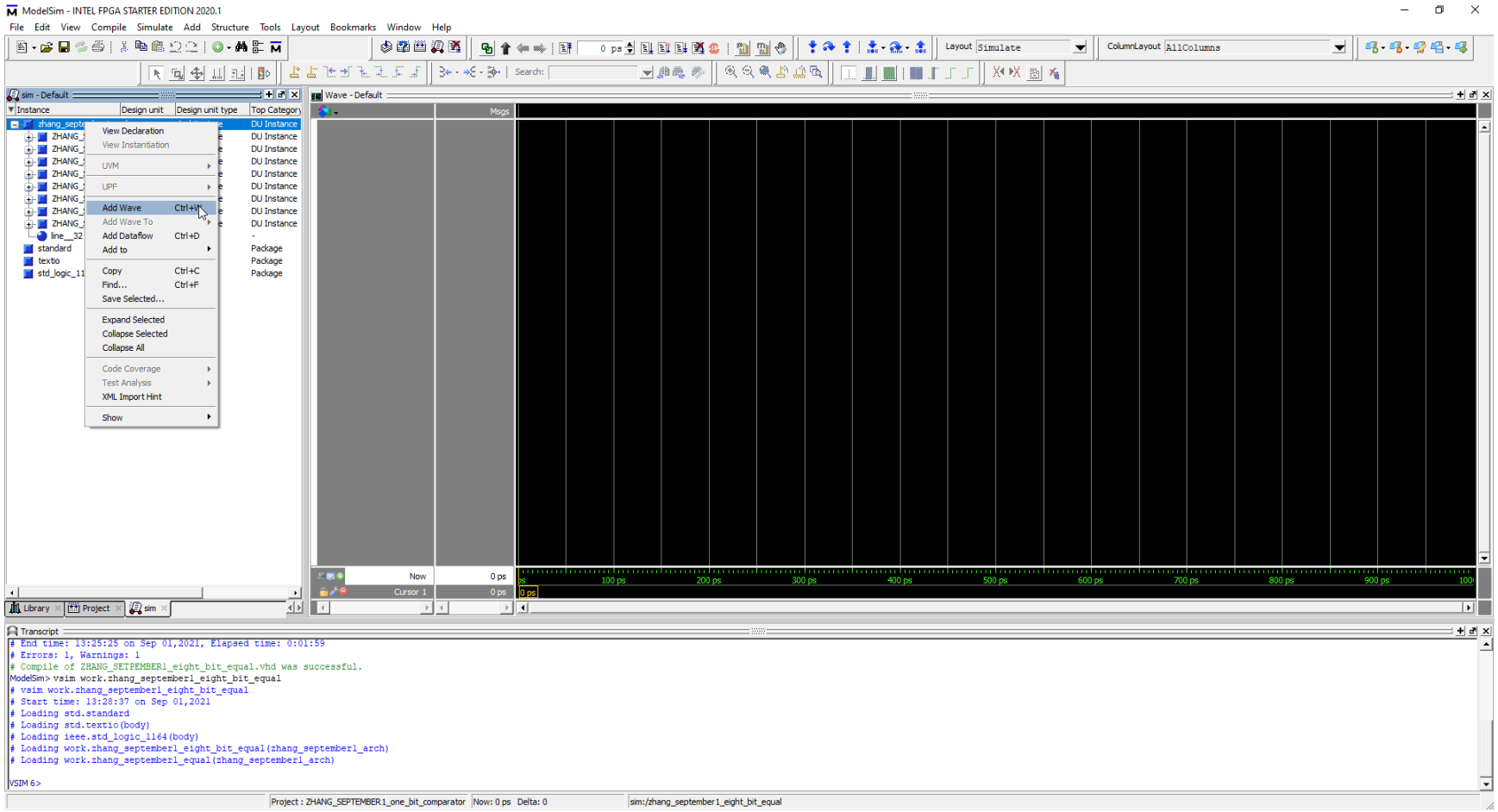
This is the vhdl code created for the eight bit comparator that I have written

Task 2 : 8-bit Comparator , Chue Zhang



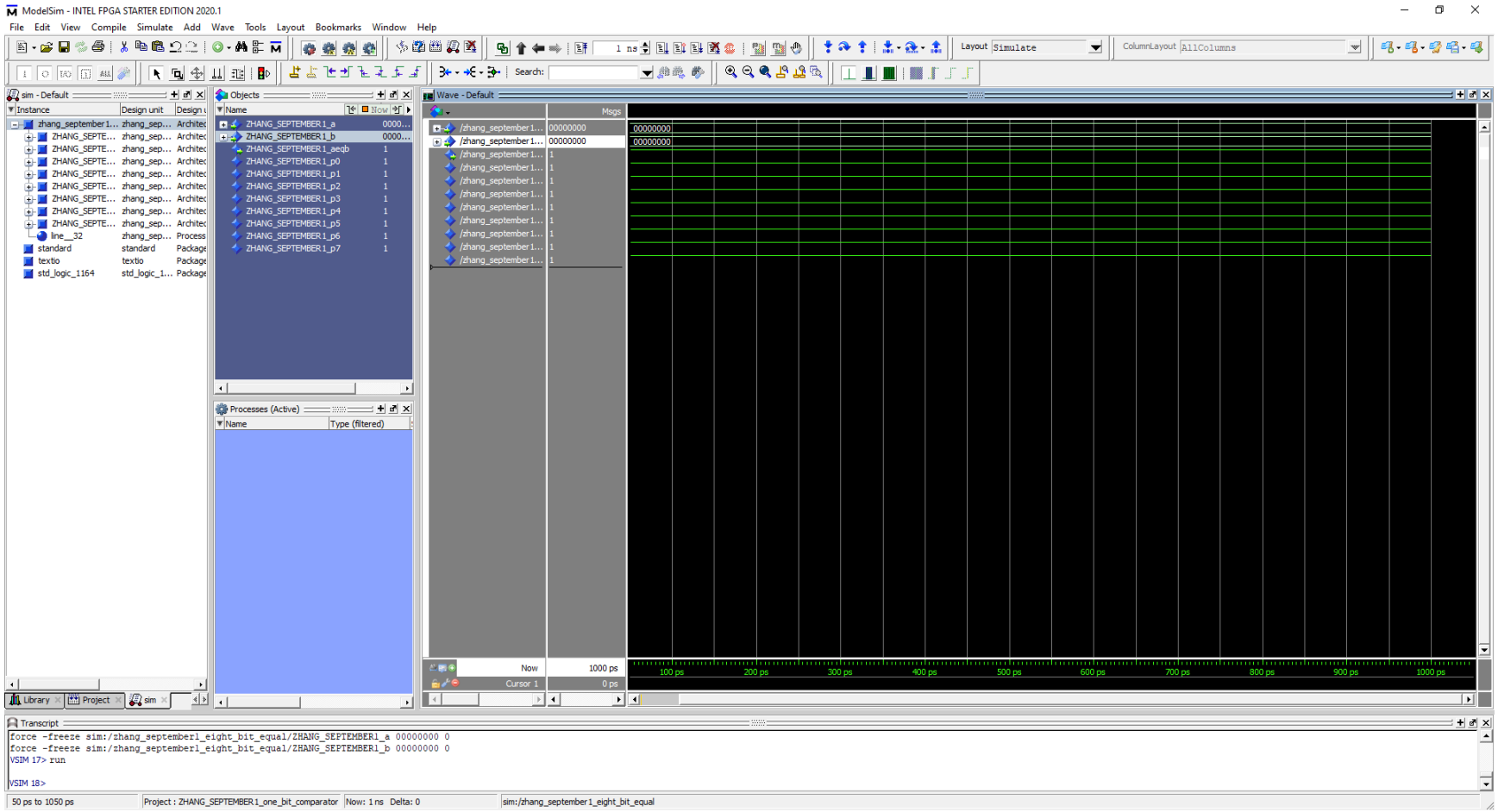
I compile with success as shown in the textbox at the bottom of the figure

Task 2 : 8-bit Comparator , Chue Zhang



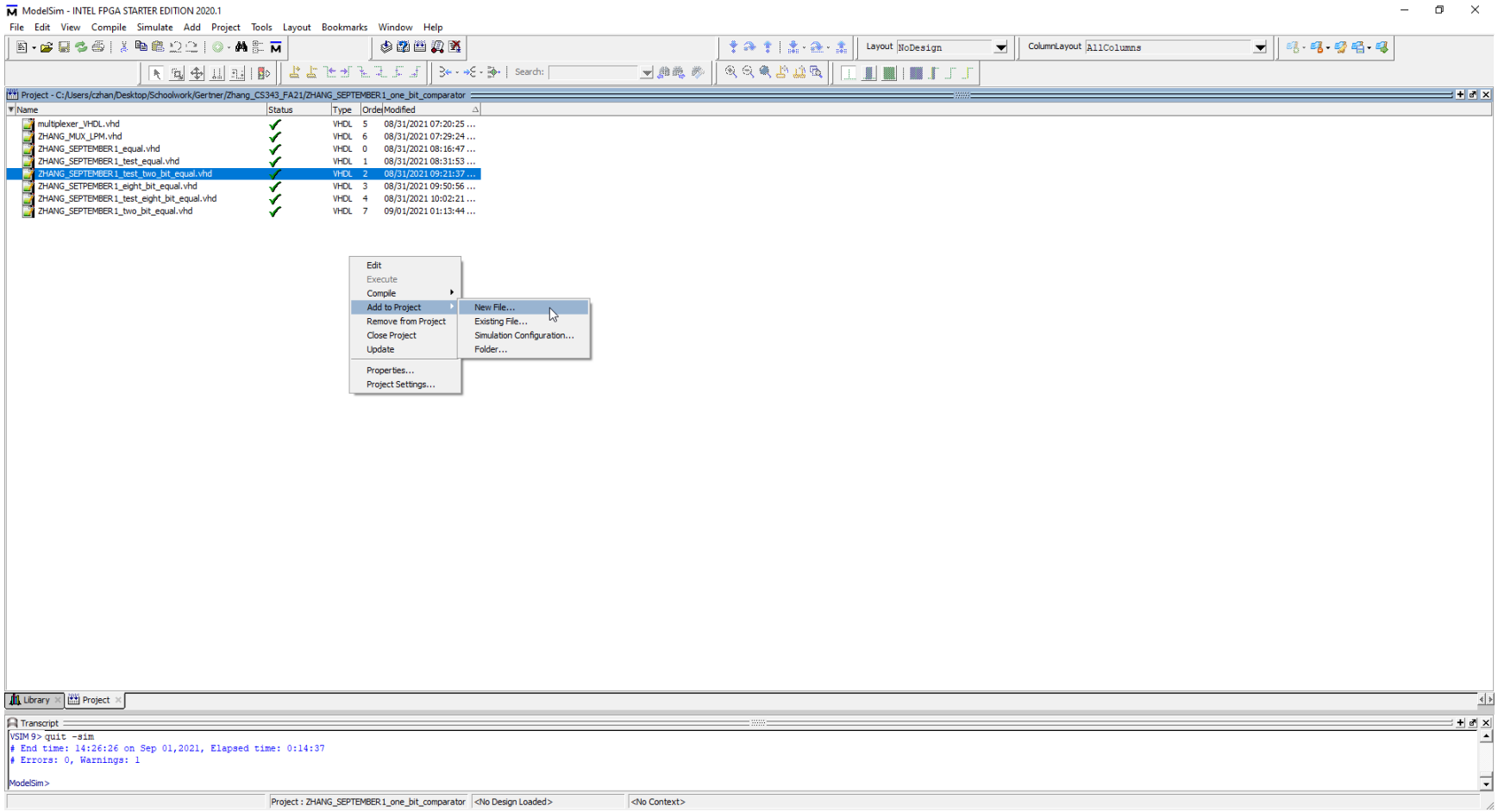
I begin simulating in modelsim by adding the waves to be simulated with

Task 2 : 8-bit Comparator , Chue Zhang



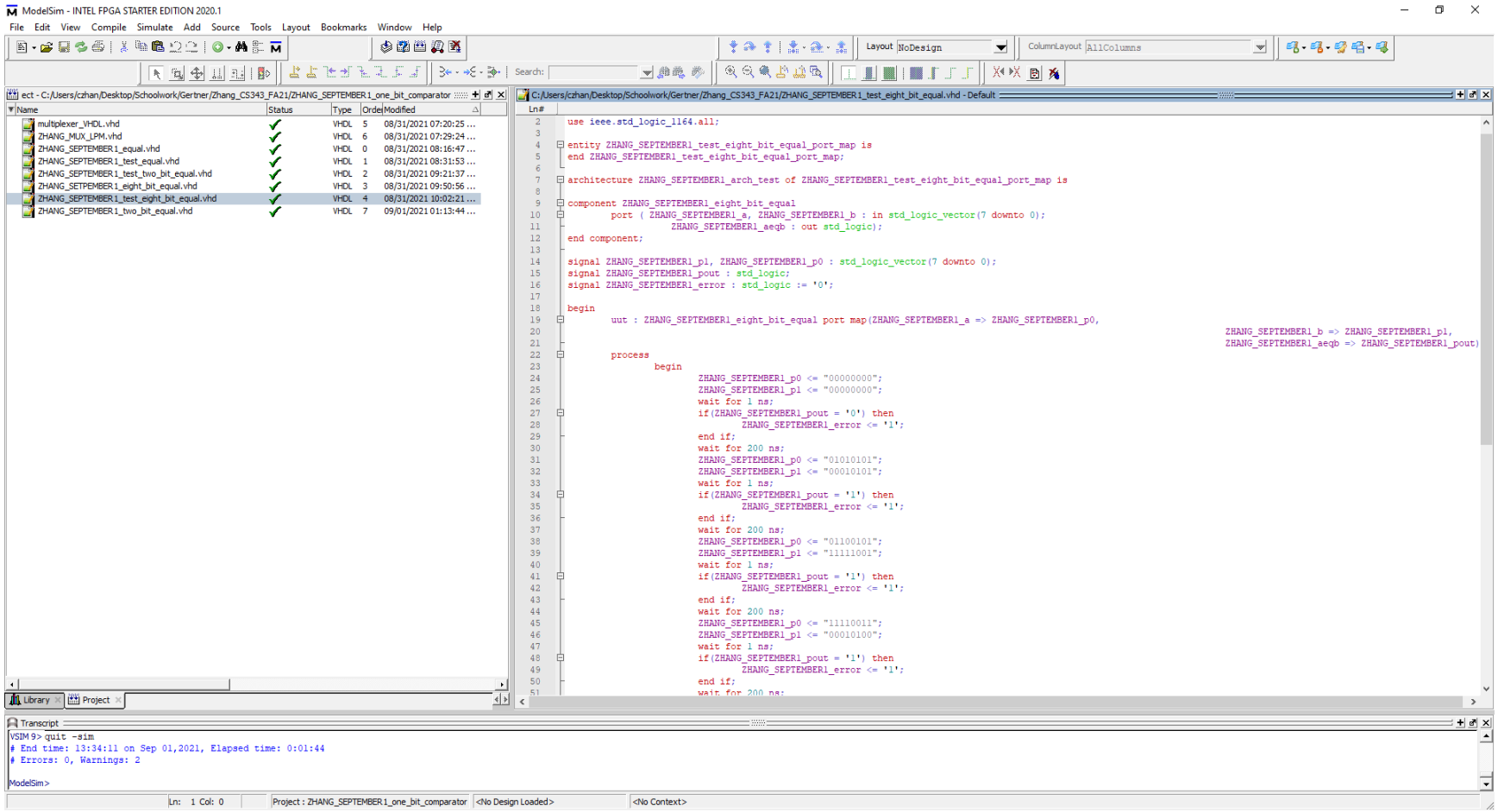
This was the modelsim Output, as you can see in it, output is 1 which is correct

Task 2 : 8-bit Comparator , Chue Zhang



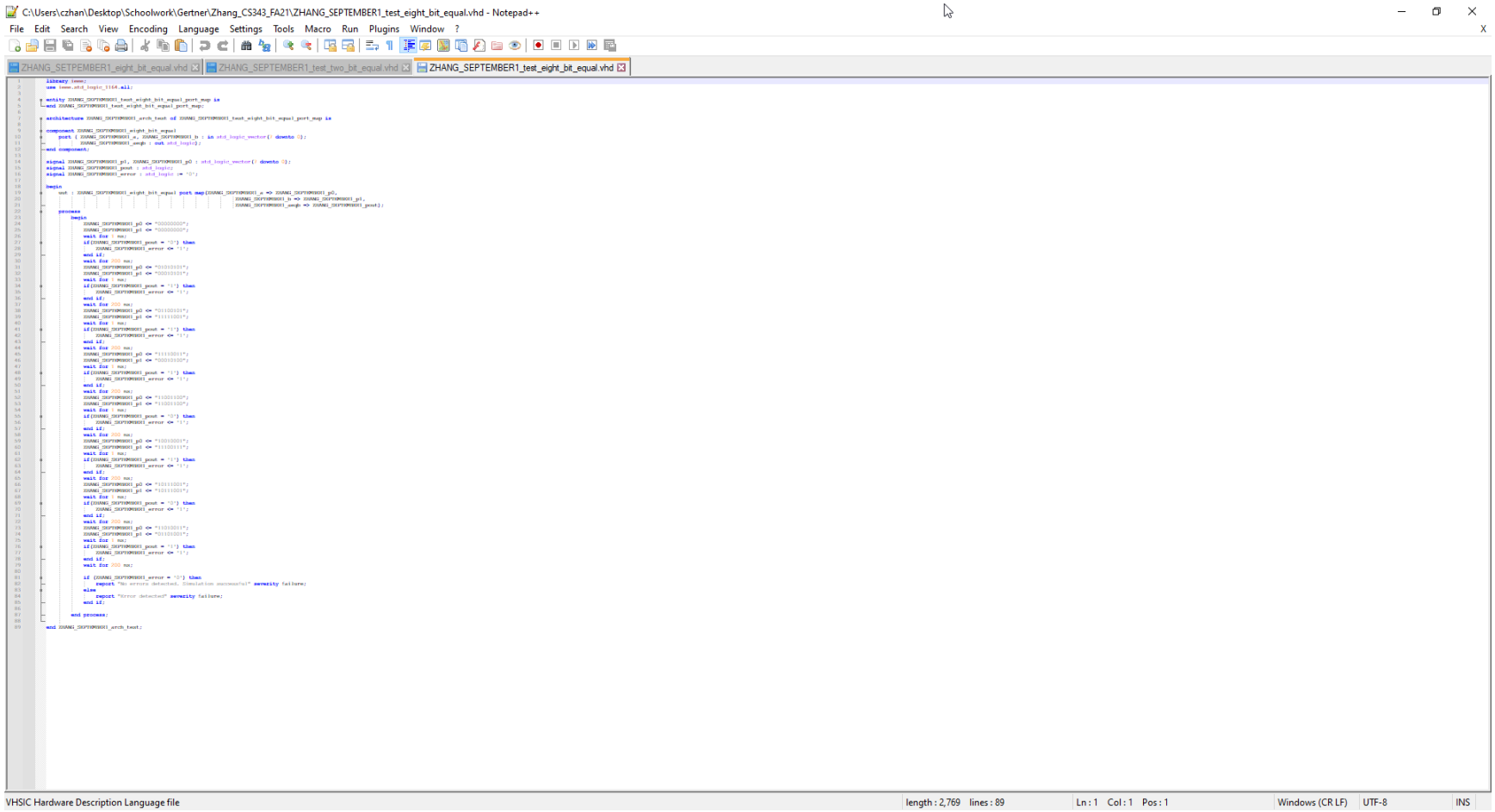
Next I create another file for the test bench

Task 2 : 8-bit Comparator , Chue Zhang



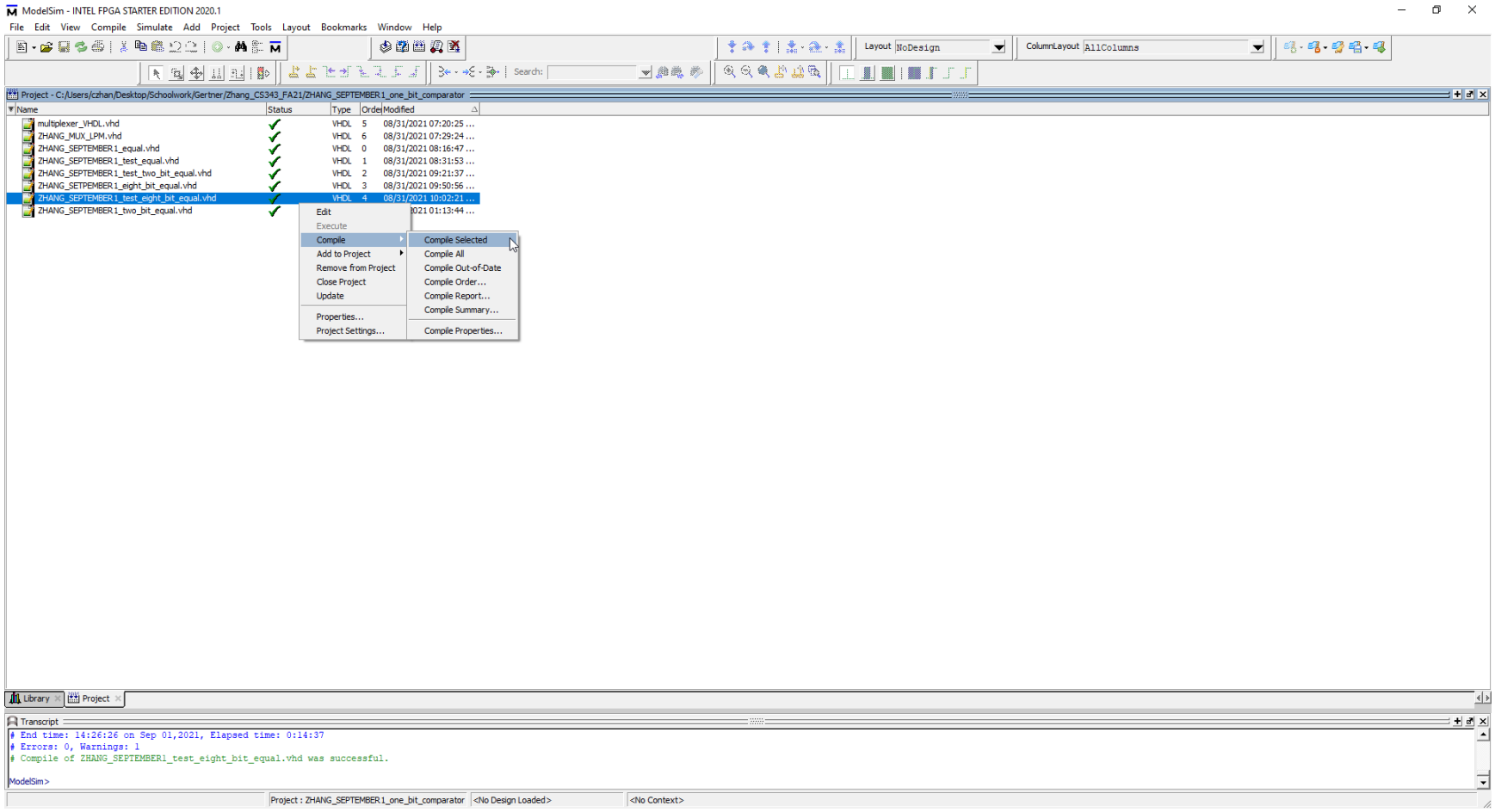
I begin writing VHDL code in the new file I have created

Task 2 : 8-bit Comparator , Chue Zhang



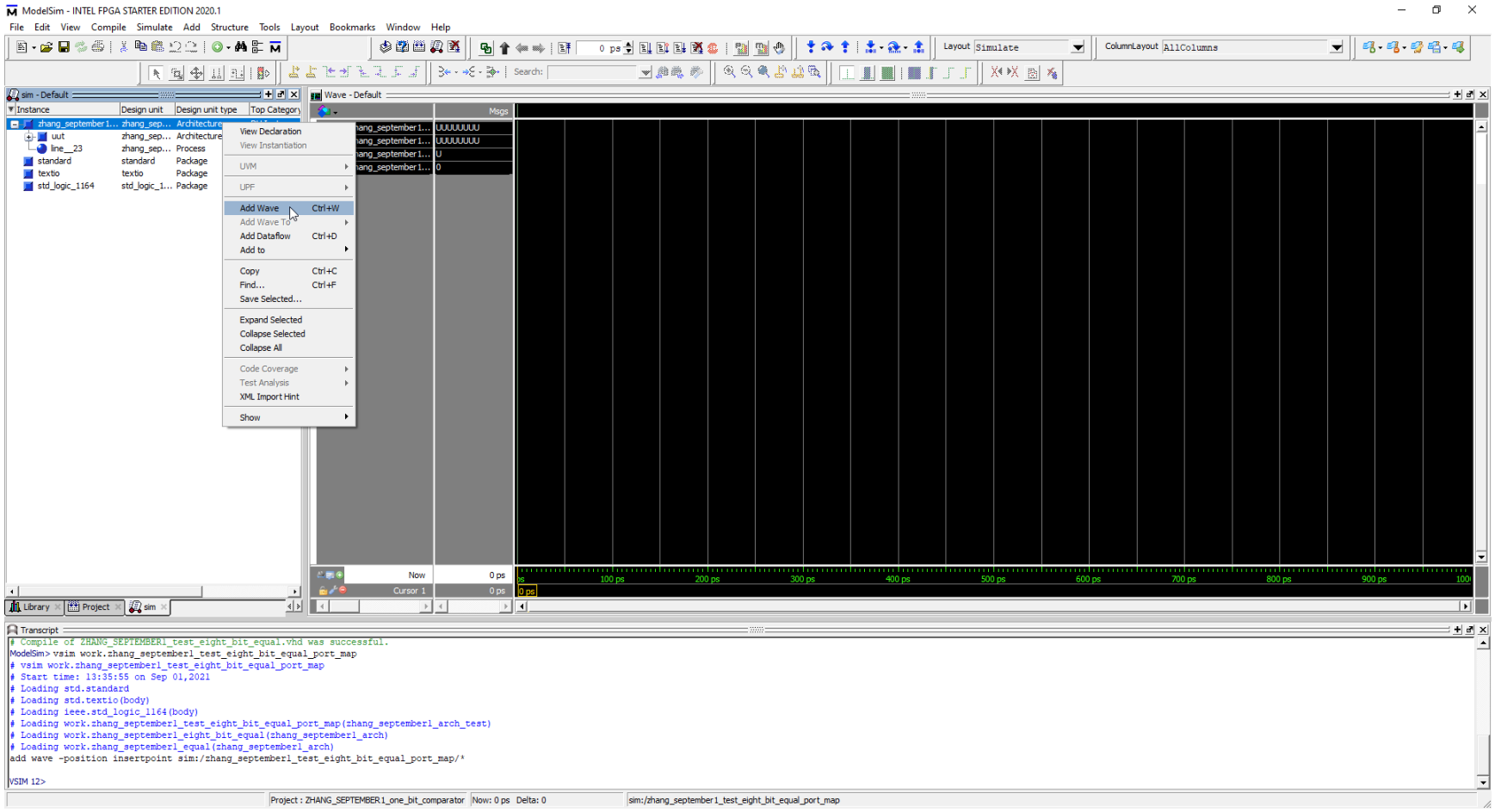
This is the entire code in a screenshot

Task 2 : 8-bit Comparator , Chue Zhang



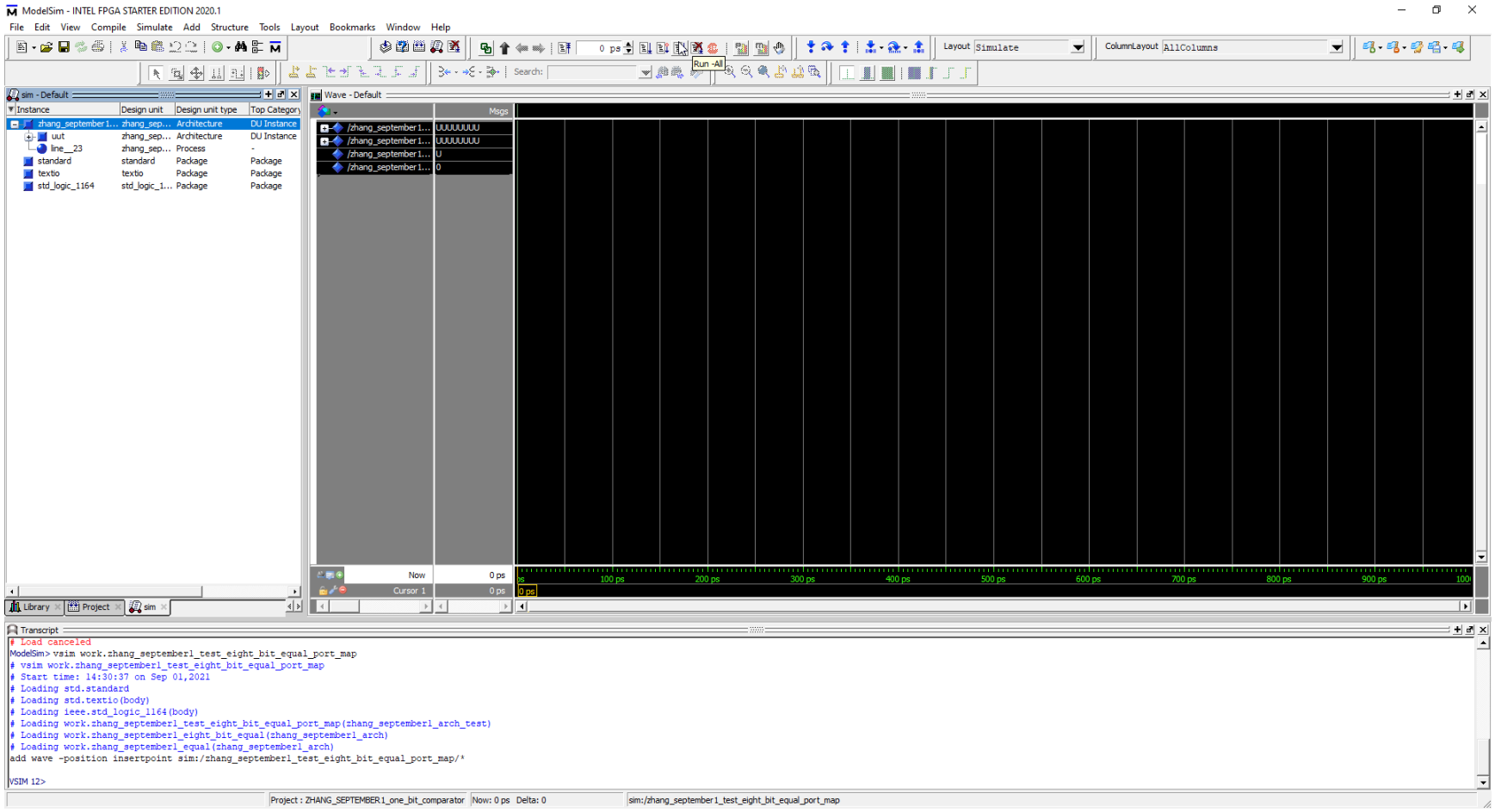
I compile the test bench VHDL file for the 8 bit comparator with no issues. Refer to green text on the bottom of image for verification of success

Task 2 : 8-bit Comparator , Chue Zhang



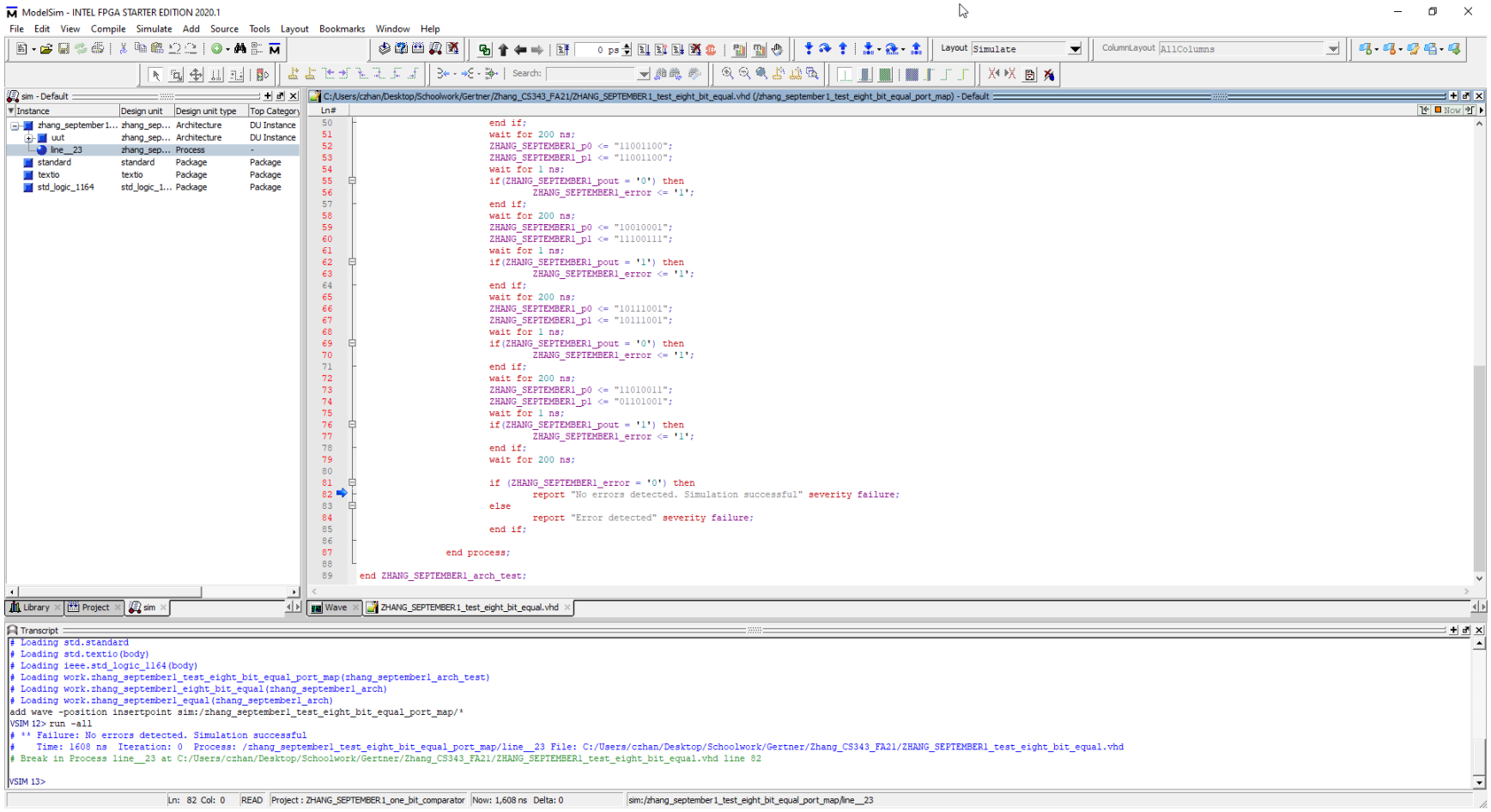
I add waves that are required to be simulated

Task 2 : 8-bit Comparator , Chue Zhang



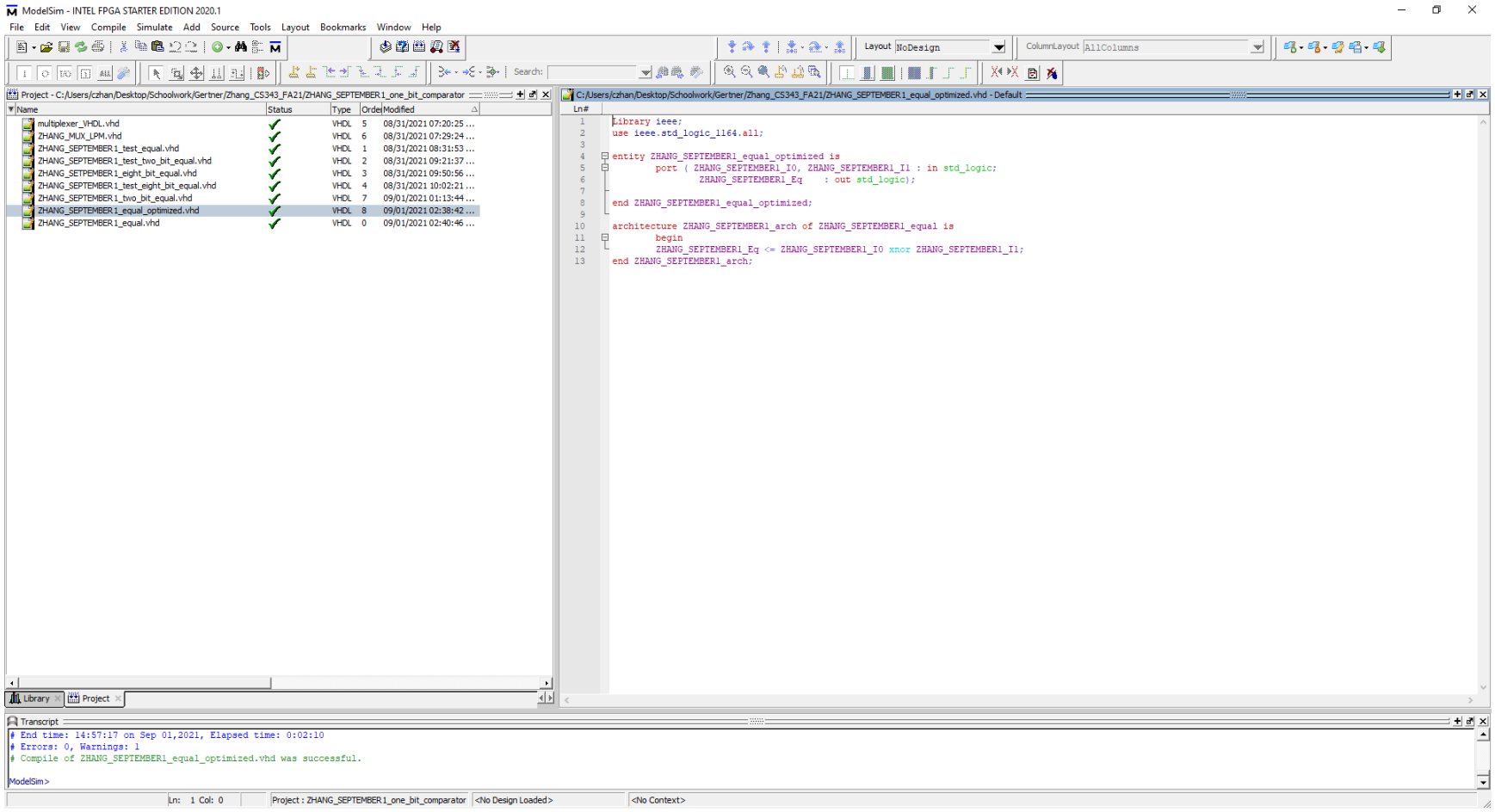
I press run all

Task 2 : 8-bit Comparator , Chue Zhang



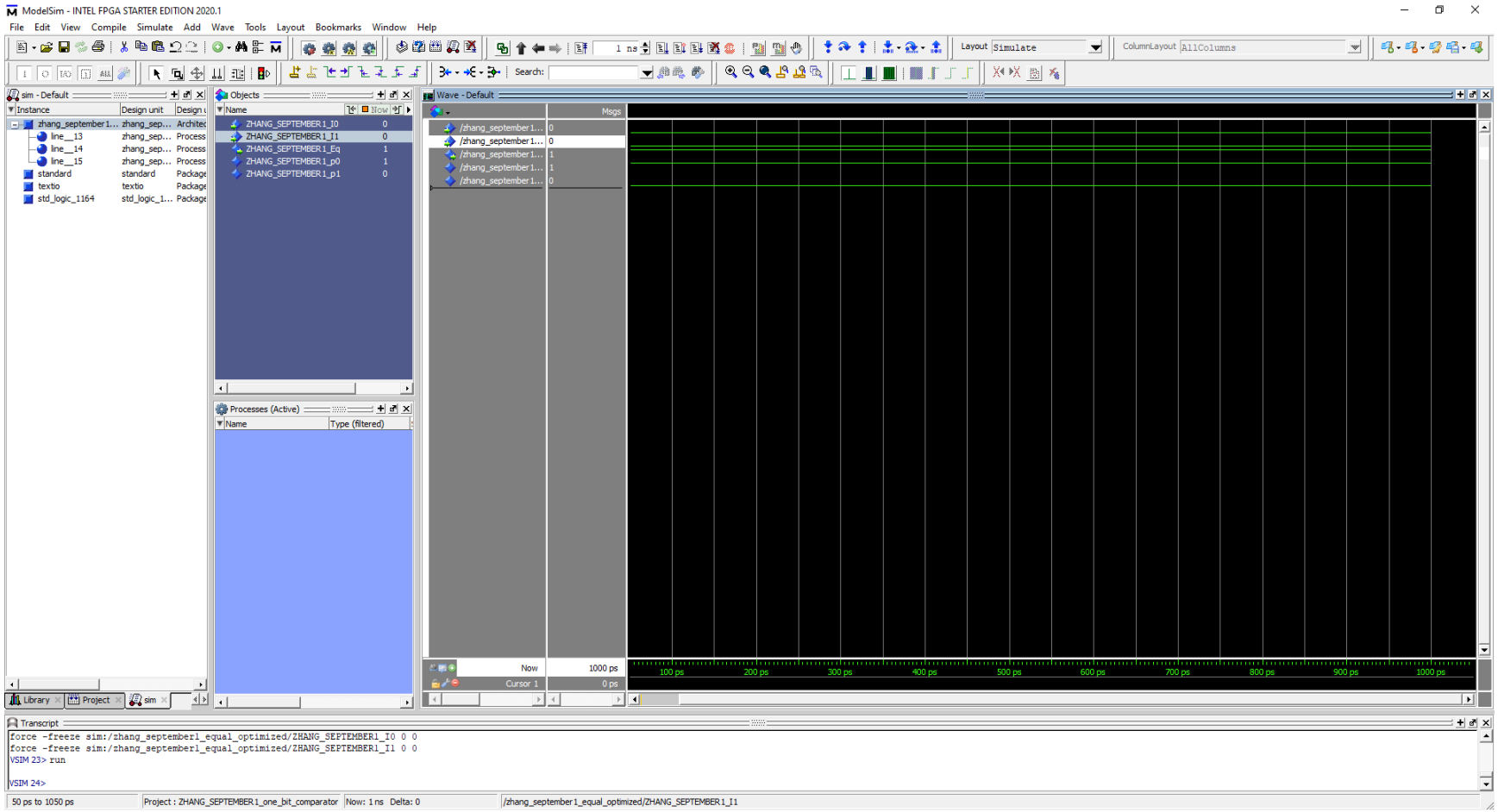
I press run all in modelsim and this page is prompted. There is an arrow at line 62 pointing that there were no errors with the simulation therefore correctness for the 8 bit comparator is verified

Task A1 : 1-bit Comparator optimized, Chue Zhang



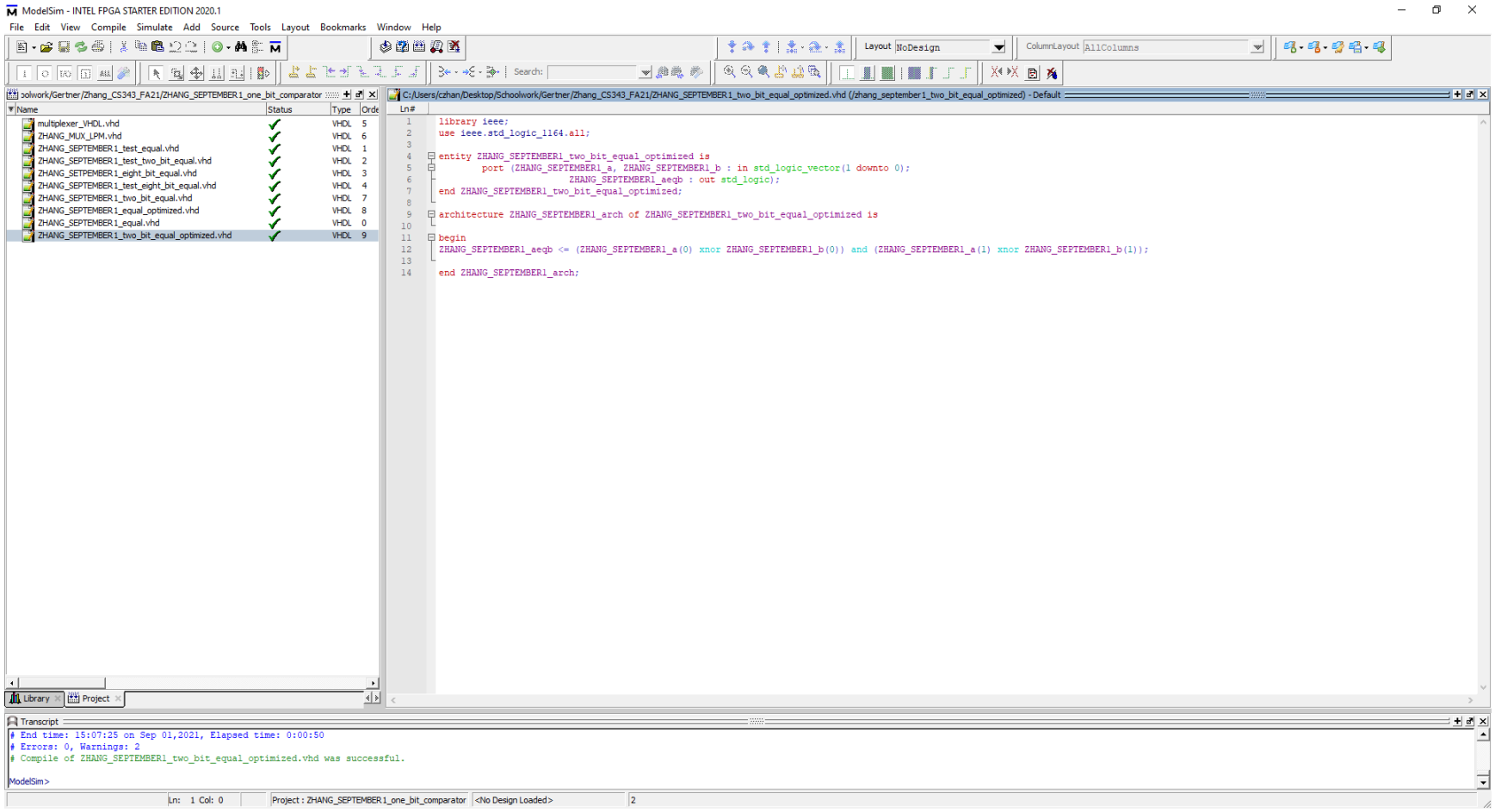
This is the 1 bit comparator that is optimized using XNOR. At the bottom of the image is also a successful compilation OF the optimized 1 bit comparator

Task A1 : 1-bit Comparator optimized, Chue Zhang



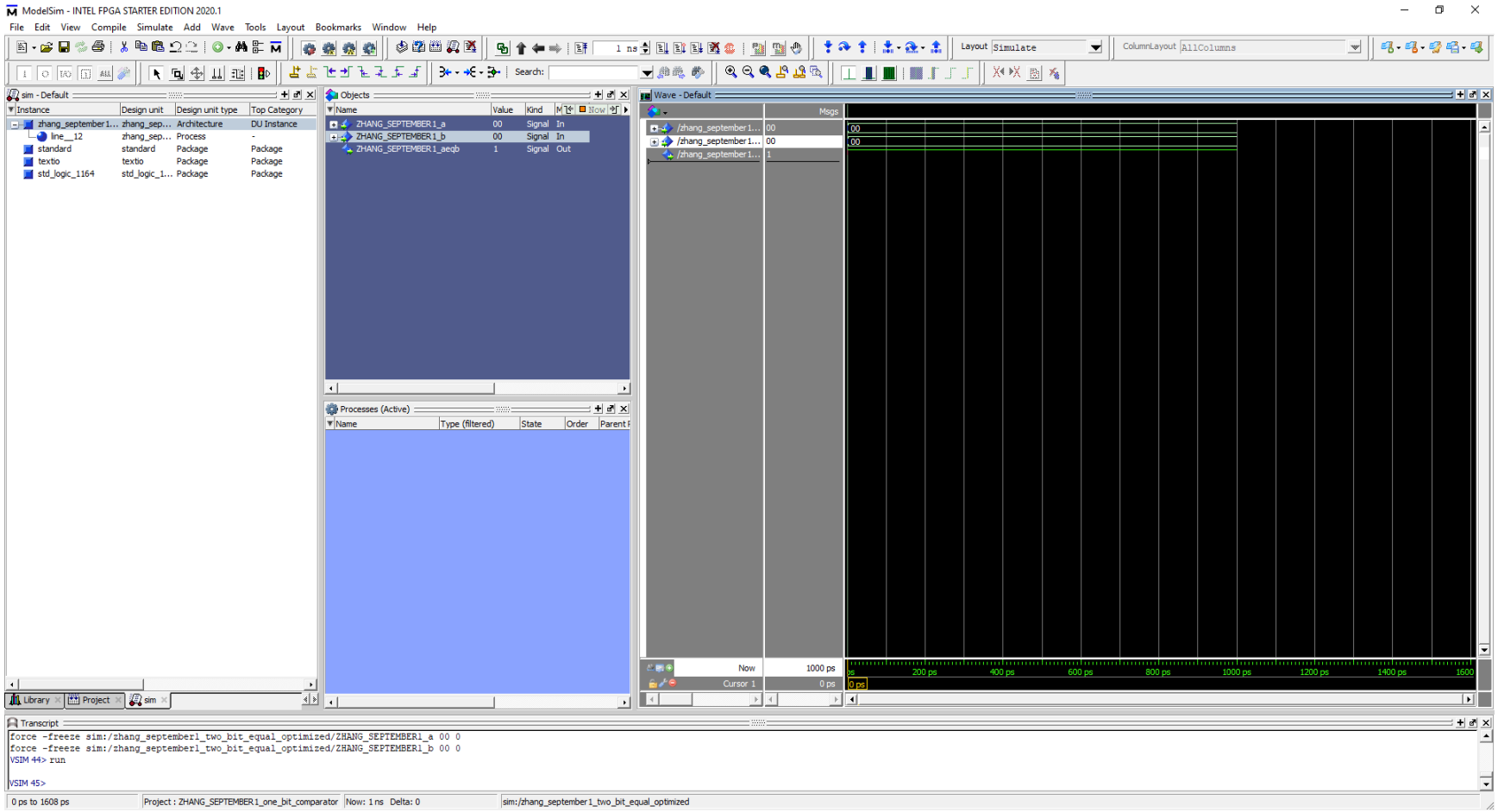
To test for correctness, in modelsim I used inputs 0 and 0 for inputs “a” and “b” respectively and received the output 1 which is correct according to xnor’s truth table.

Task A2 : 2-bit Comparator optimized, Chue Zhang



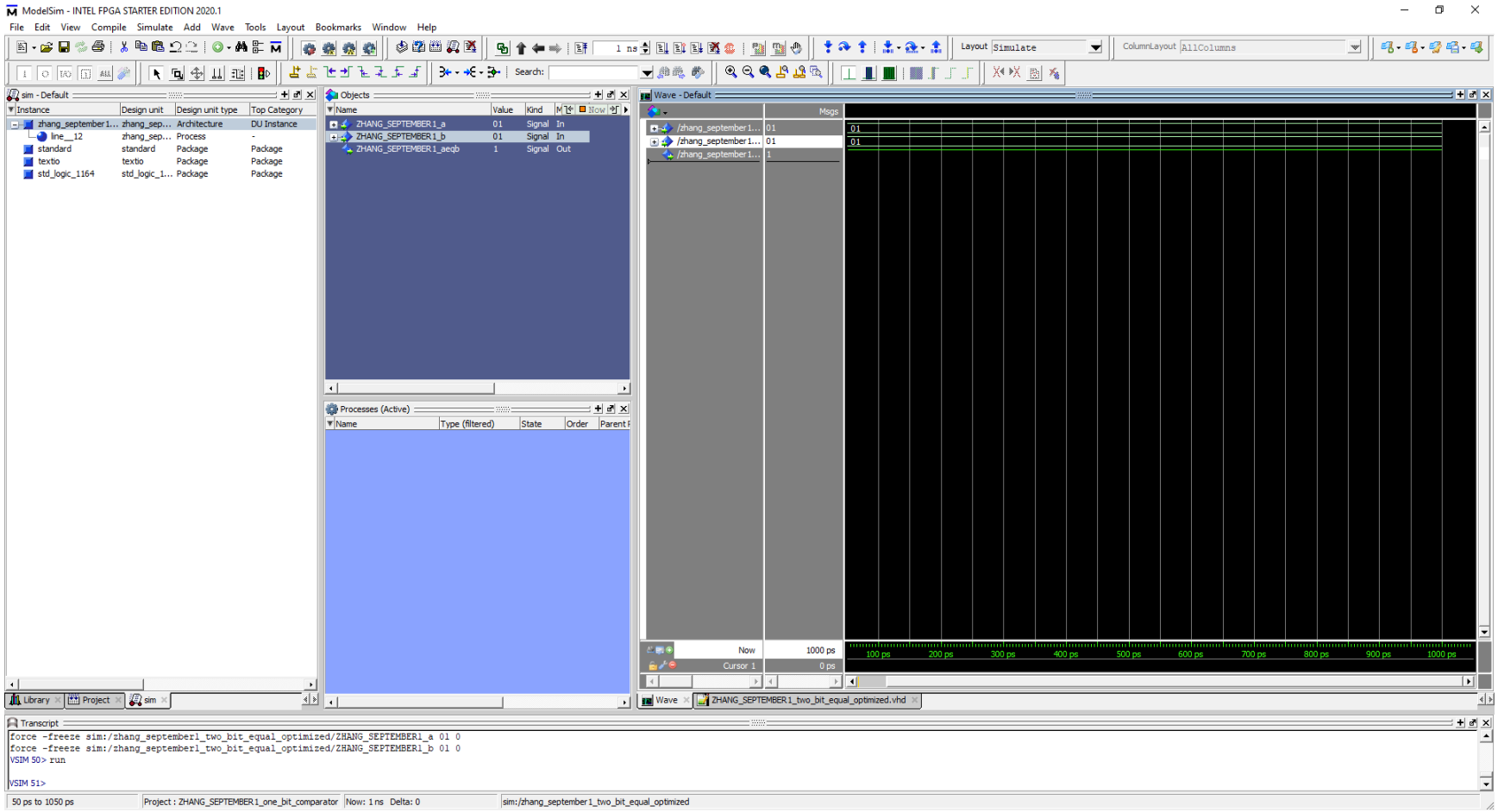
In this image, what is shown is the optimized code for a 2-bit comparator as well as a successful compilation to ensure that the code we will be simulating is correct

Task A2 : 2-bit Comparator optimized, Chue Zhang



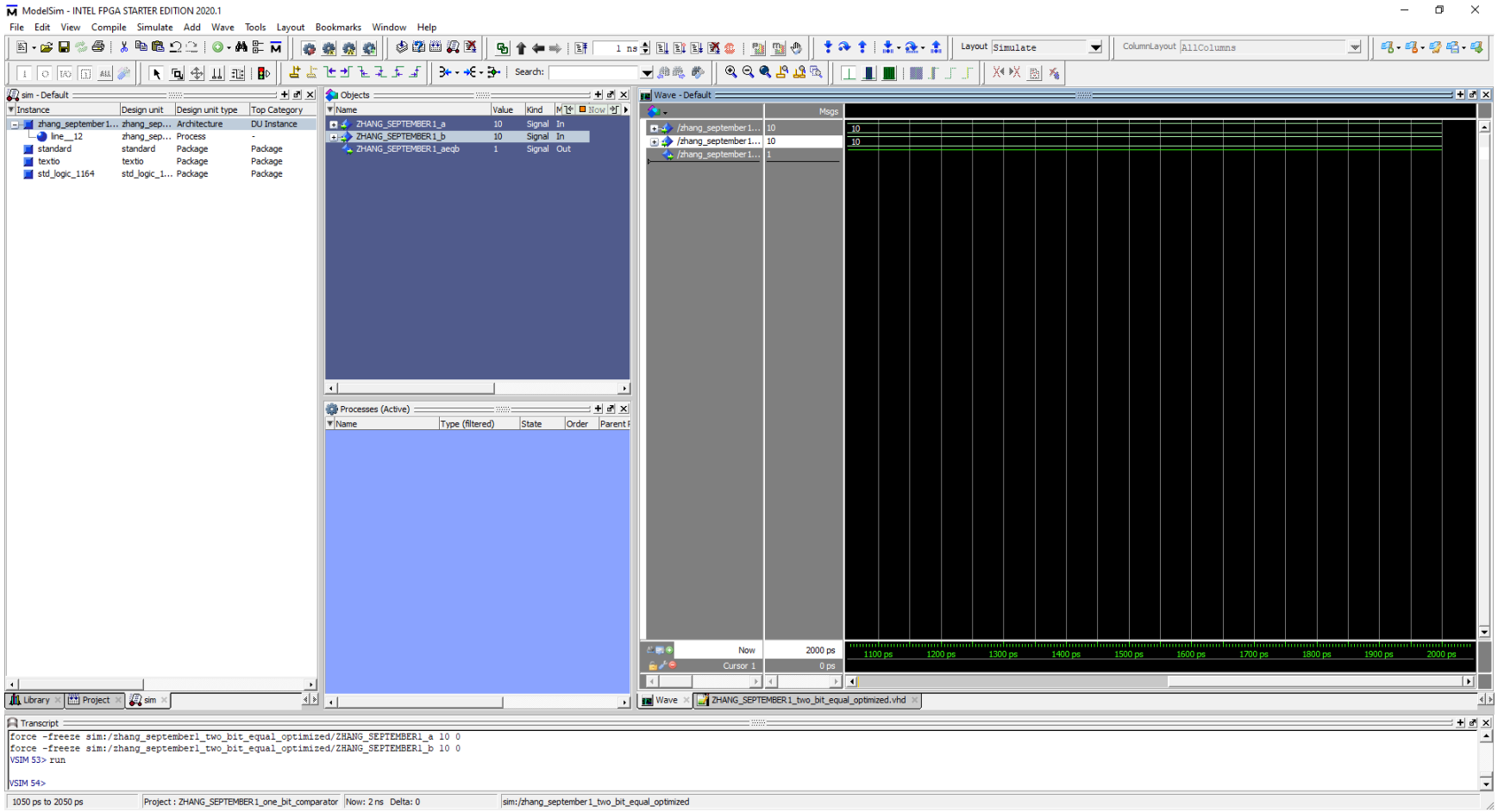
In the image above is the modelsim simulation of the optimized 2 bit comparator running with the inputs “0000” with the output of 1 which is correct.

Task A2 : 2-bit Comparator optimized, Chue Zhang



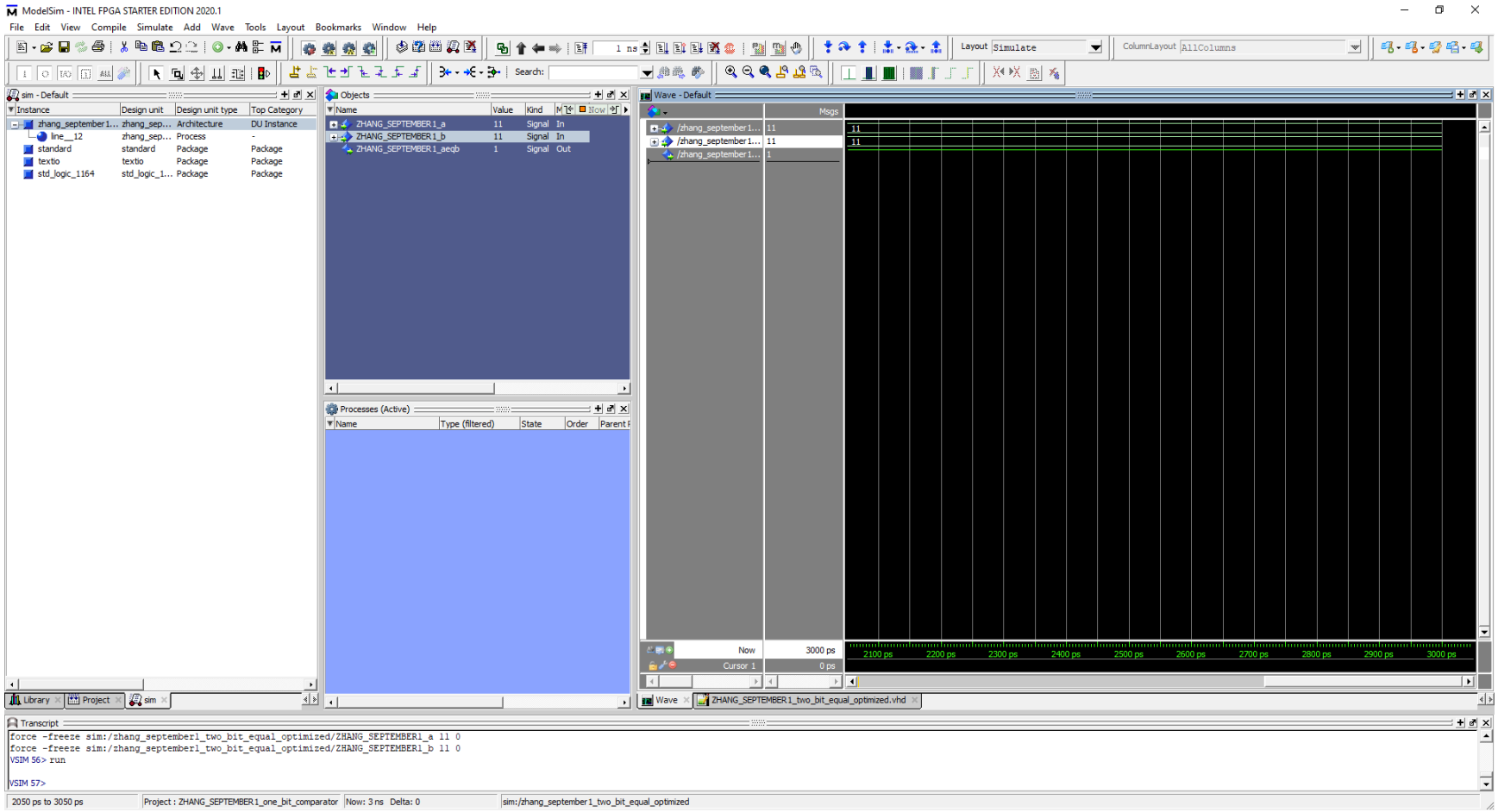
Input with 0101 returning 1

Task A2 : 2-bit Comparator optimized, Chue Zhang



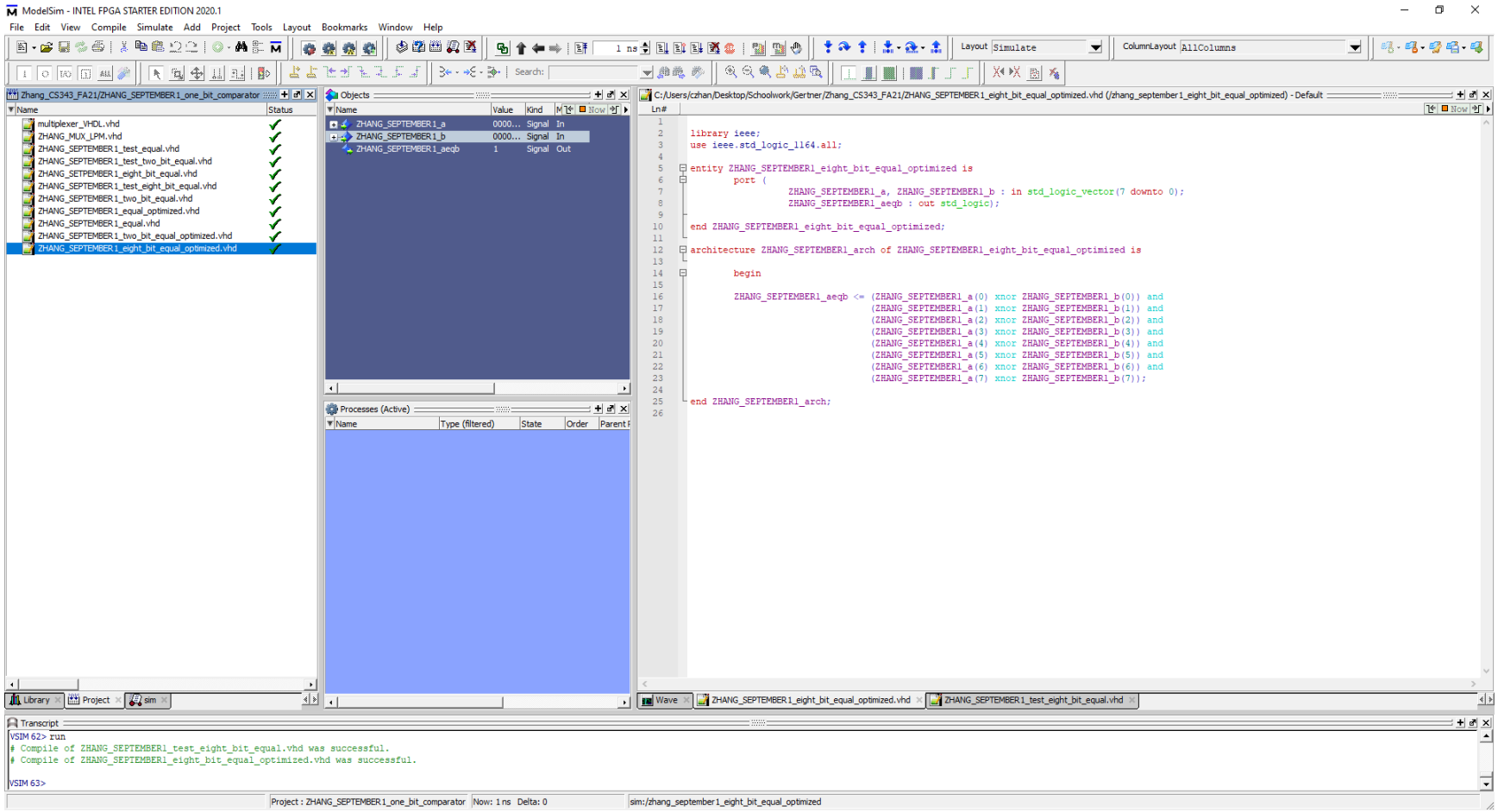
Input with 1010 returning 1

Task A2 : 2-bit Comparator optimized, Chue Zhang



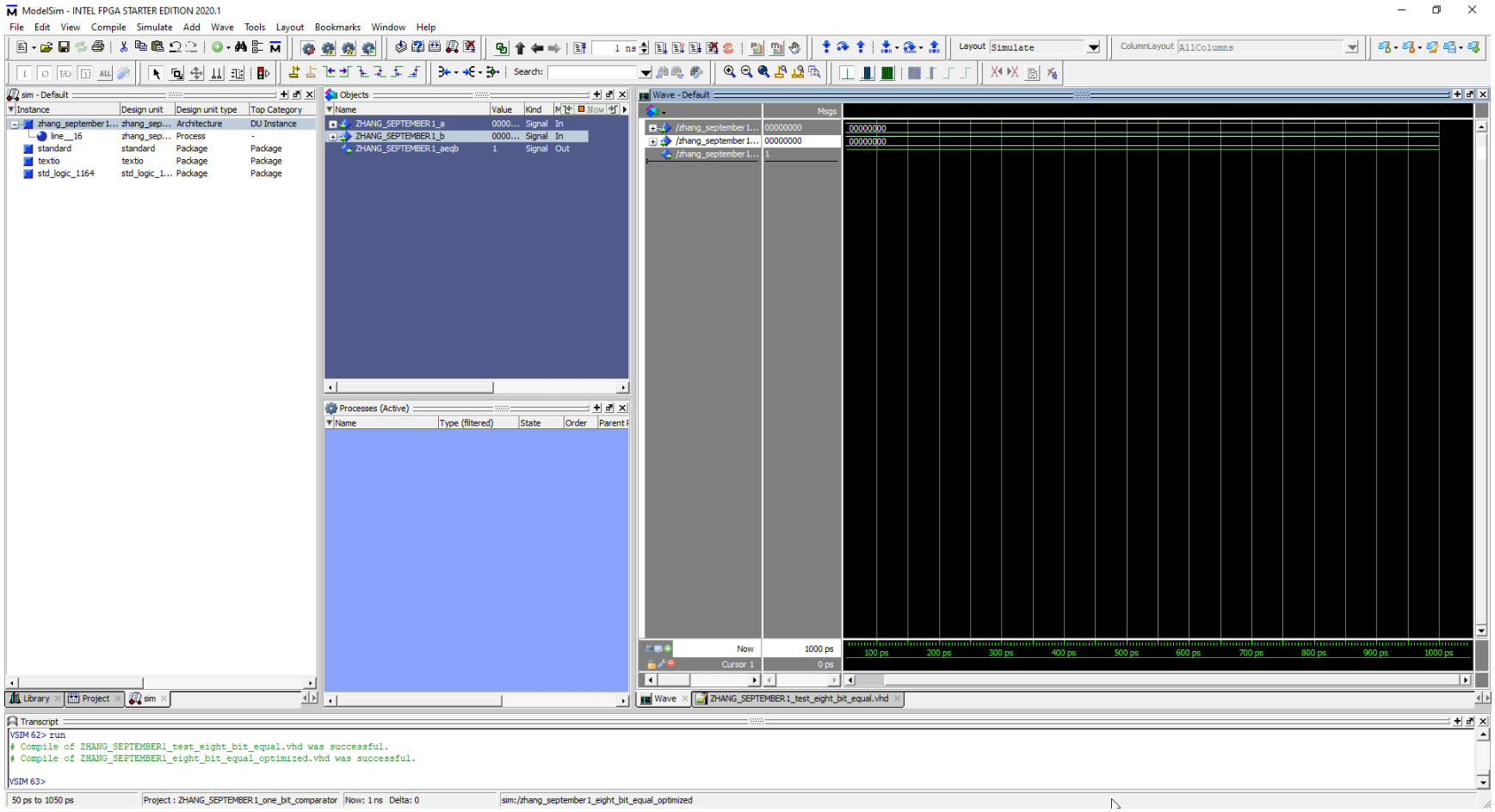
Input with 1111 returning 1

Task A3 : 8-bit Comparator optimized, Chue Zhang



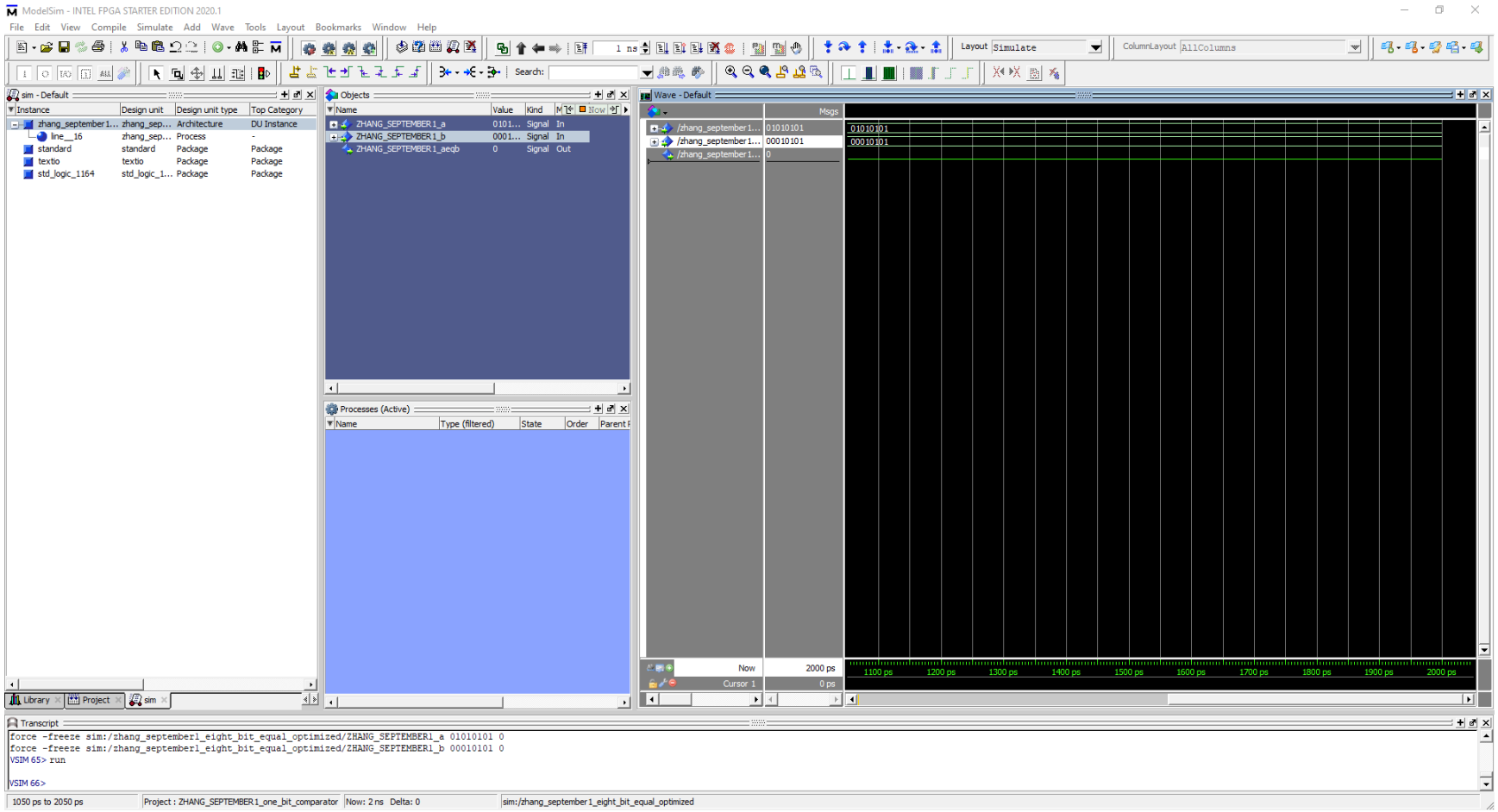
In the image above, it shows the 8 bit comparator with optimized code and successful compilation to verify correctness in the code

Task A3 : 8-bit Comparator optimized, Chue Zhang



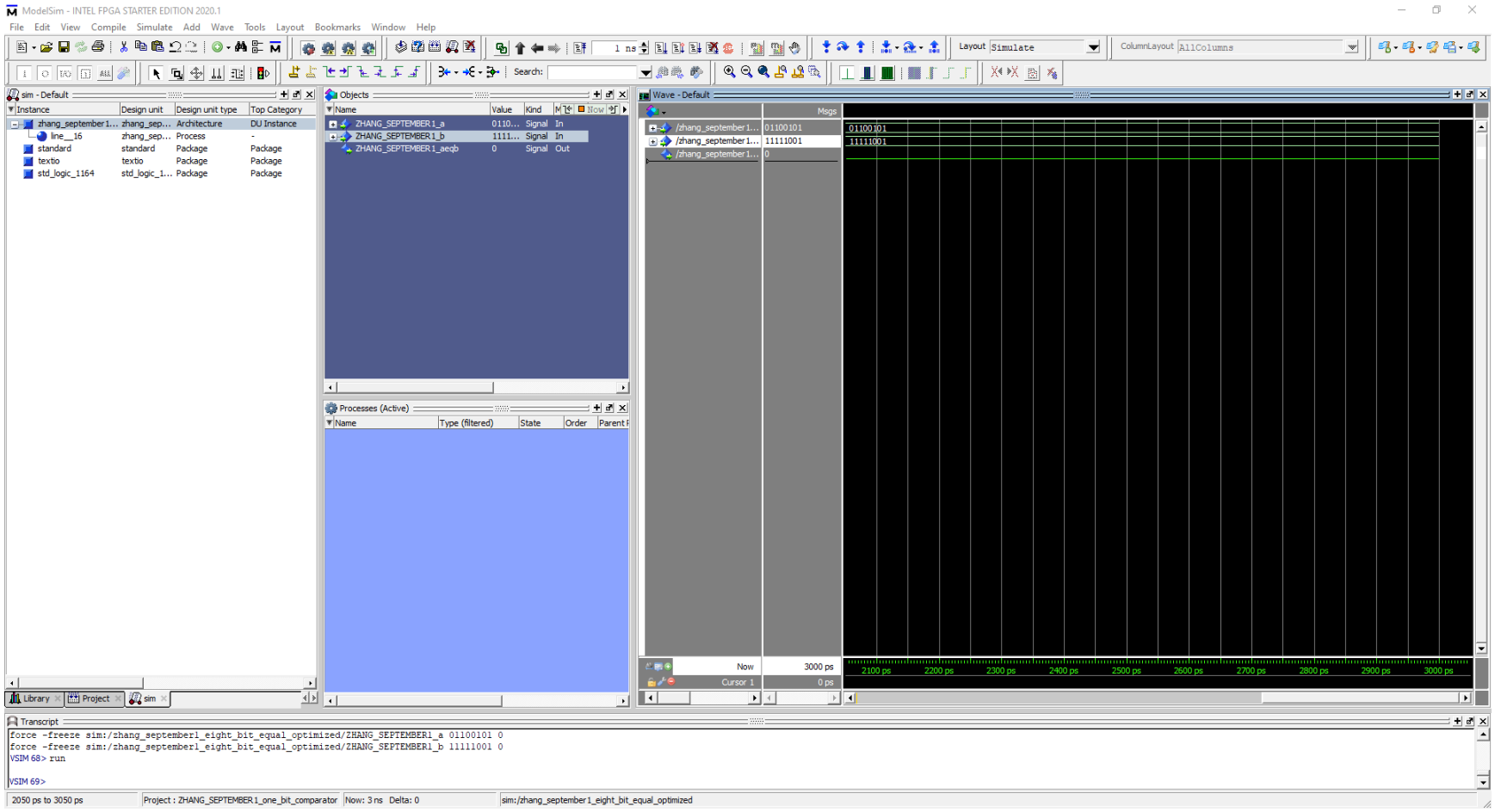
Input of 00000000, 00000000 for an output of 1

Task A3 : 8-bit Comparator optimized, Chue Zhang



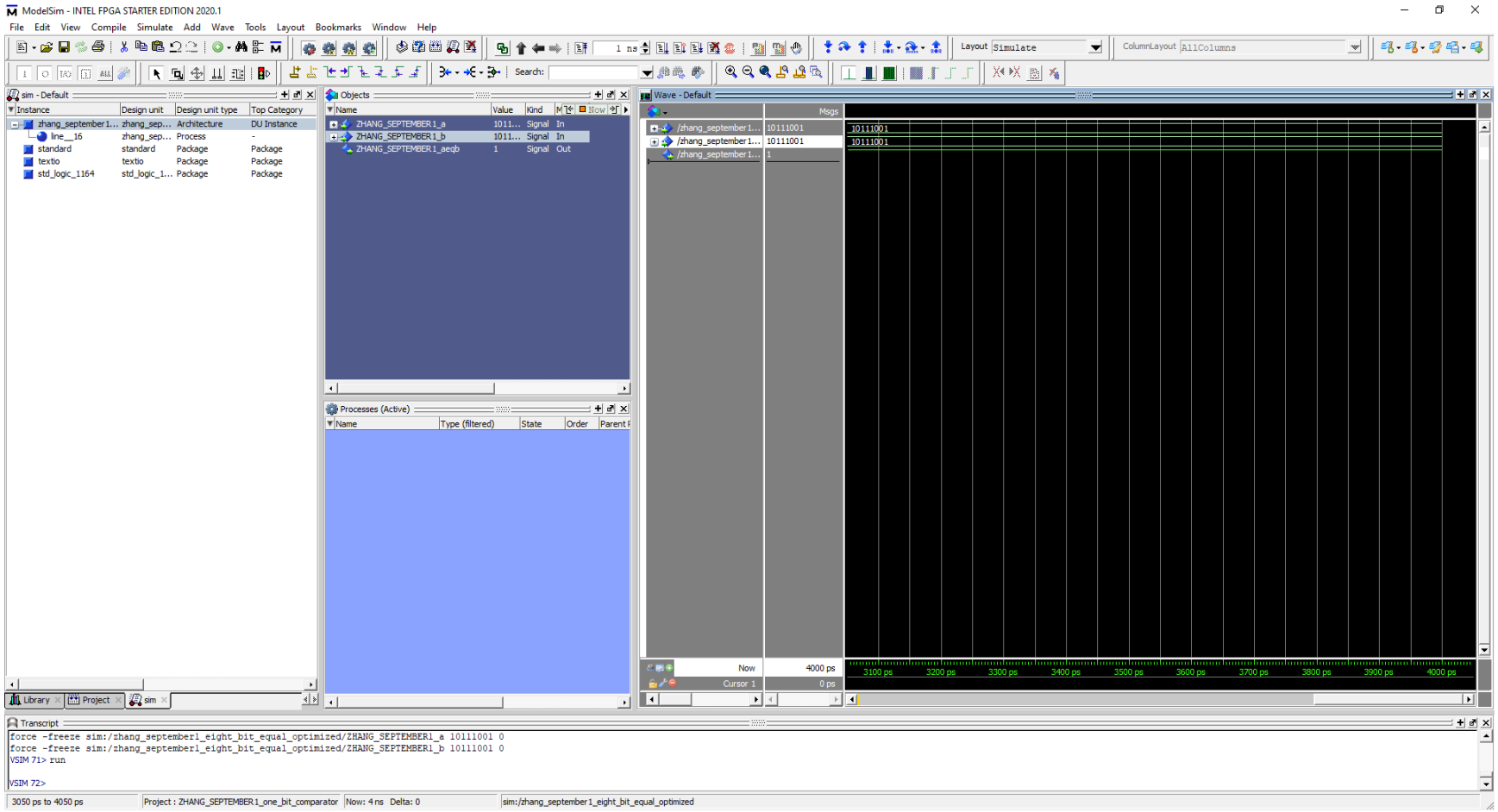
Inputted 01010101, 00010101 which outputted 0. Following the test bench logic, this should be the scenario therefore correct

Task A3 : 8-bit Comparator optimized, Chue Zhang



Inputted 01100101, 11111001 which outputted 0 which is correct according to the test bench.

Task A3 : 8-bit Comparator optimized, Chue Zhang



Input 10111001, 10111001 output 1. So in conclusion, so as long as both inputs share the same values, output 1 is guaranteed and this also helps to verify the correctness of the design of the 8 bit optimized comparator