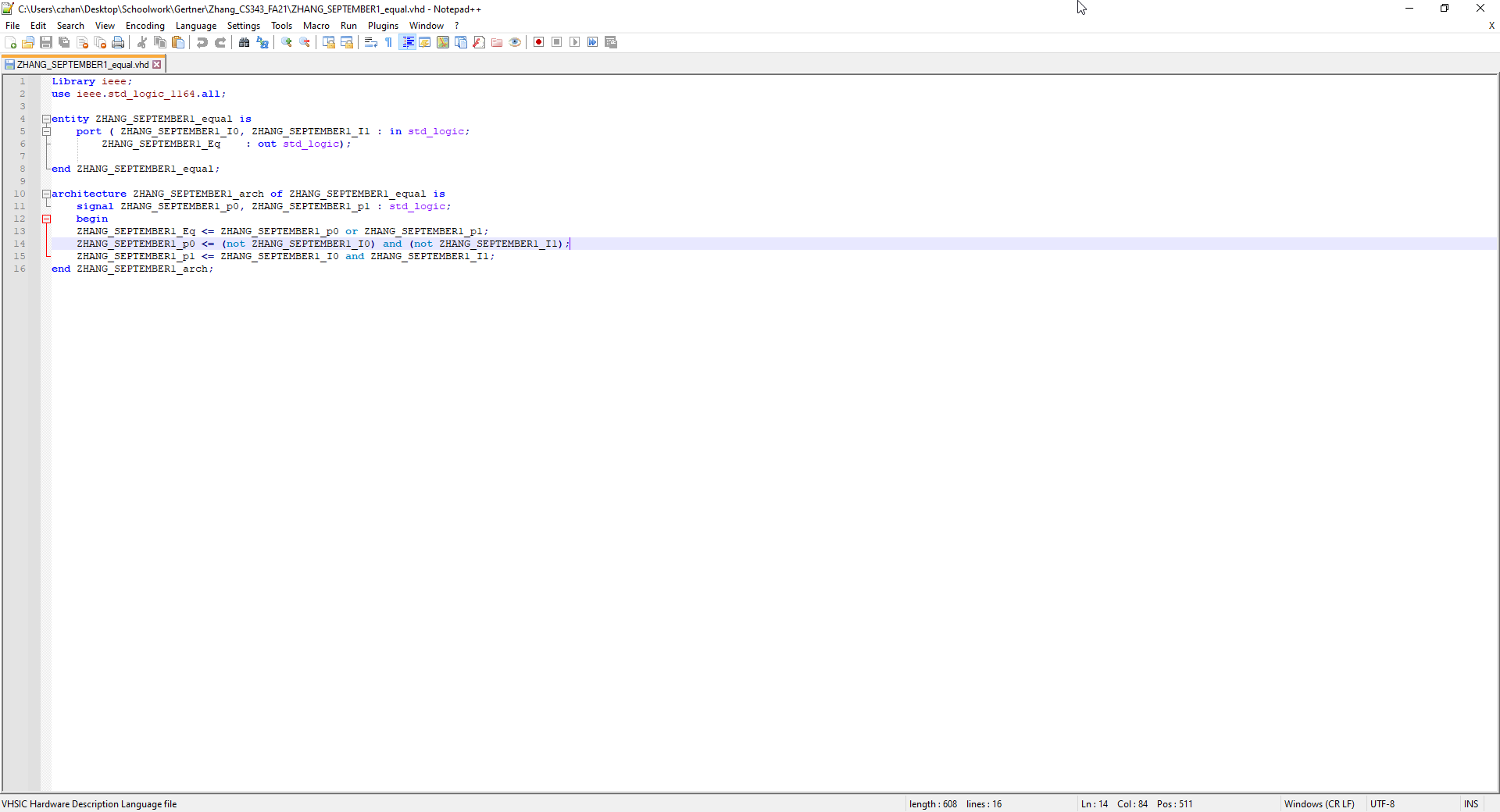
Laboratory Project 2: Comparators

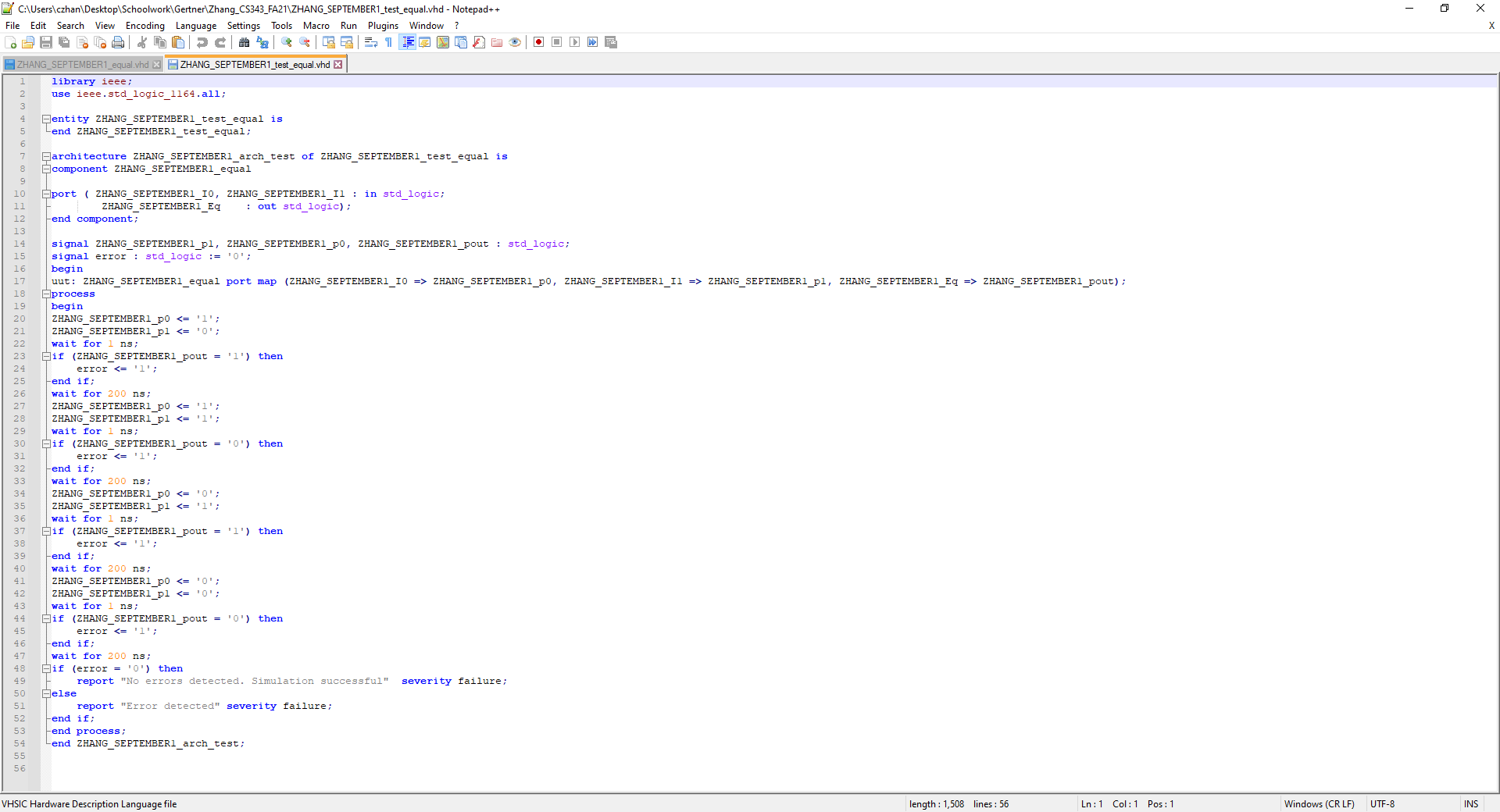
Zhang Chue

9/1/2021

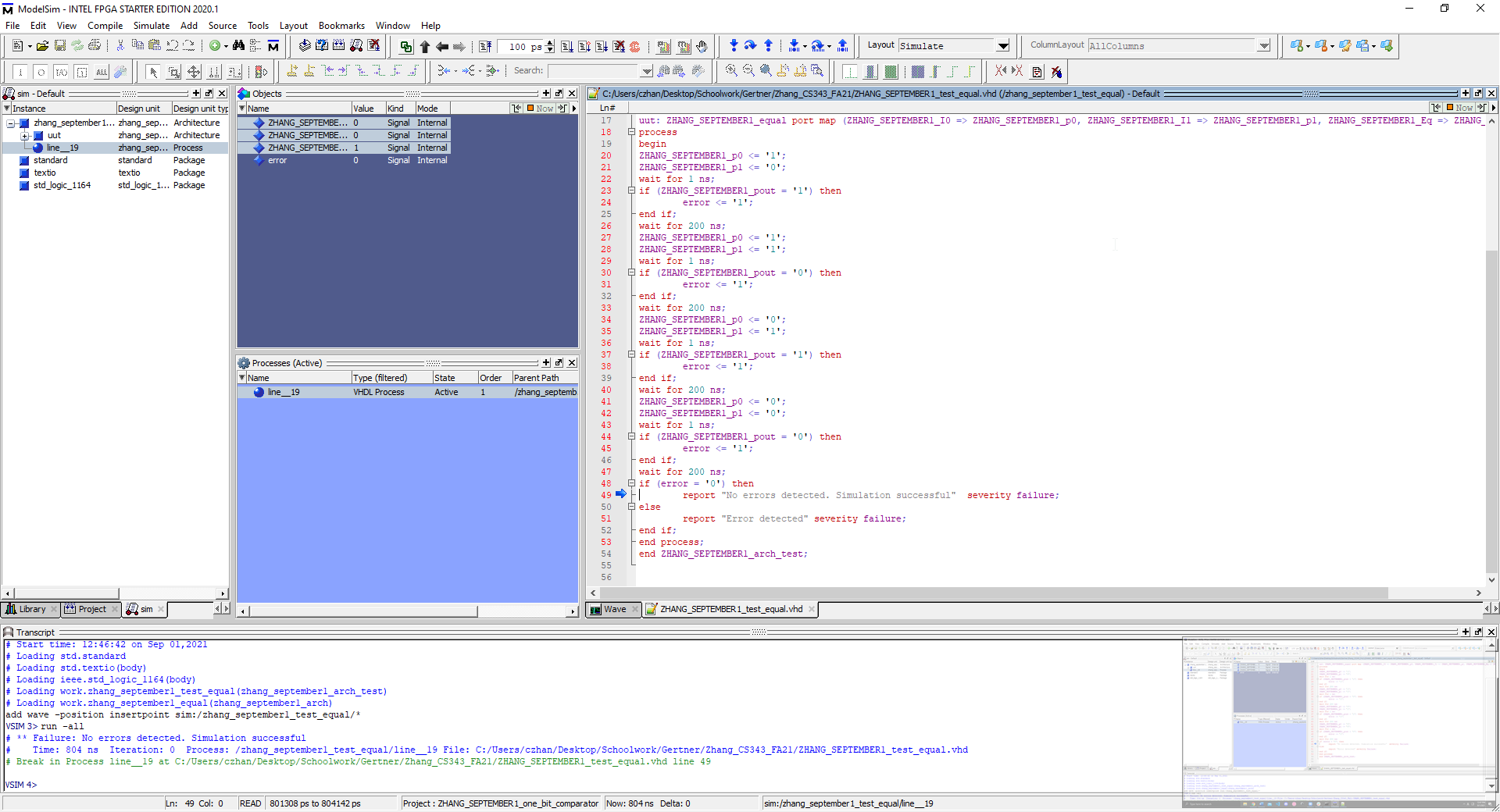
Cs343/Cs342, Professor Isidor Gertner



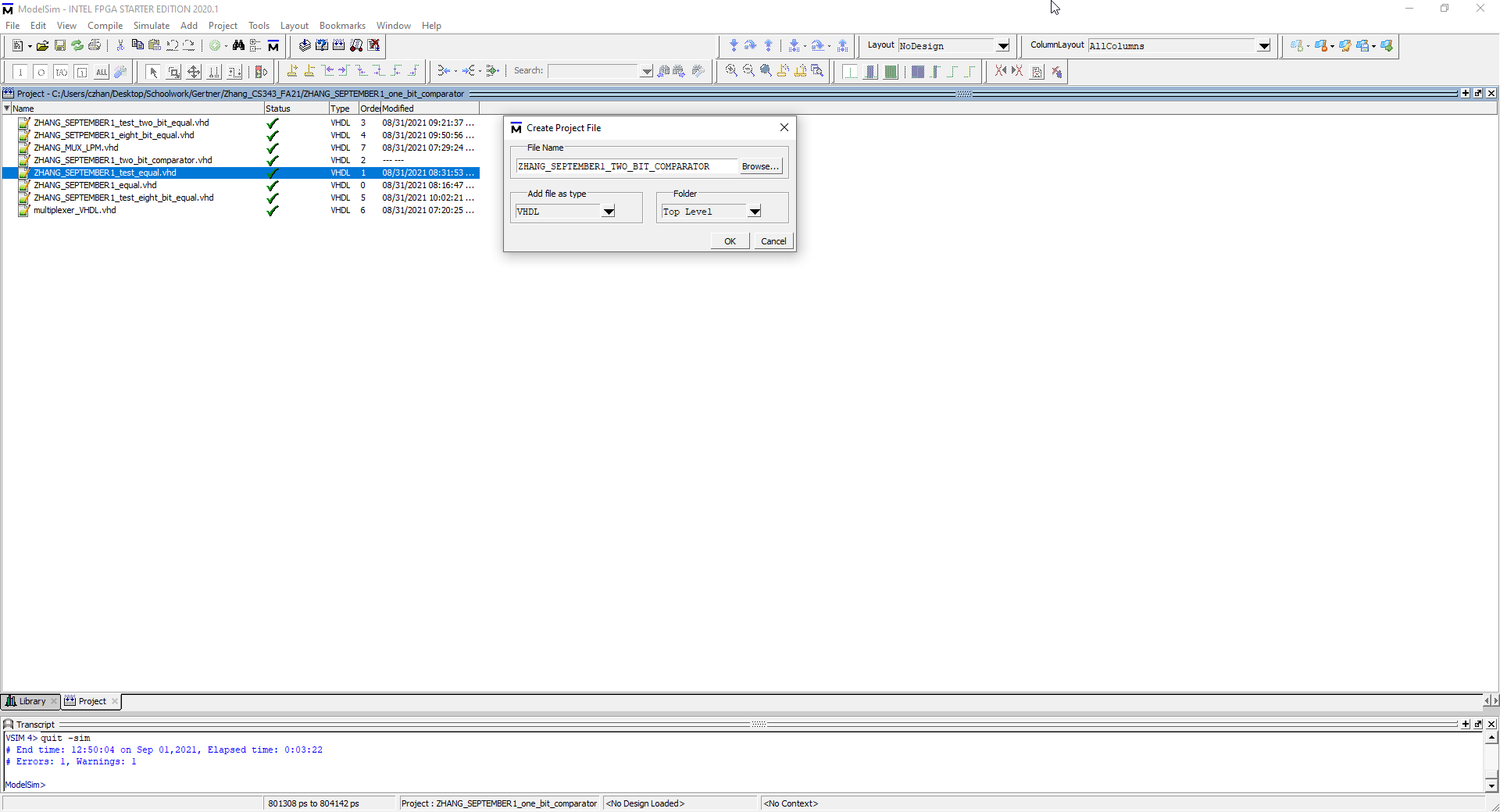
In here is the 1-bit comparator VHDL code that is unoptimized



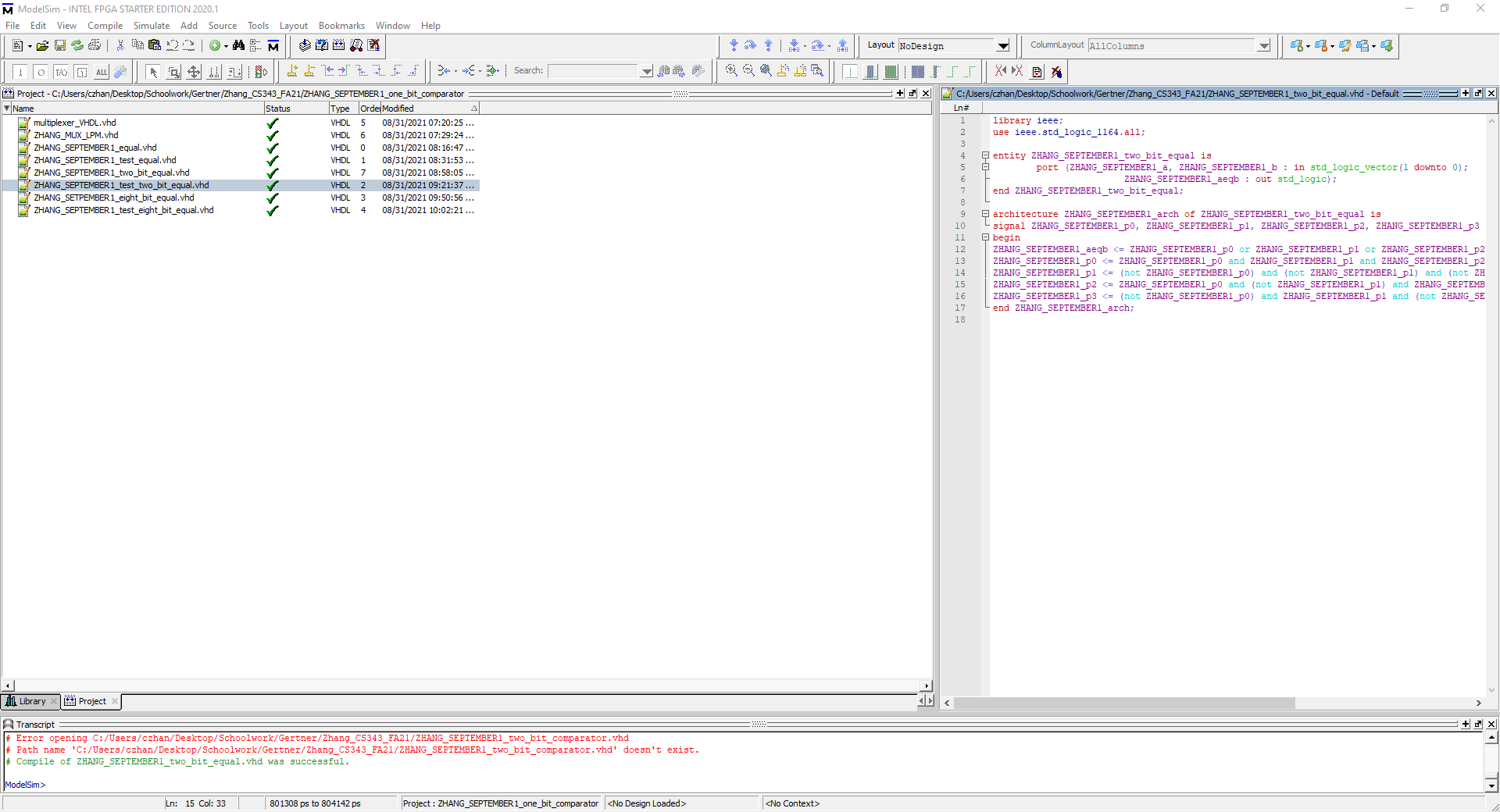
This here is the VHDL code for the test bench of the 1-bit comparator



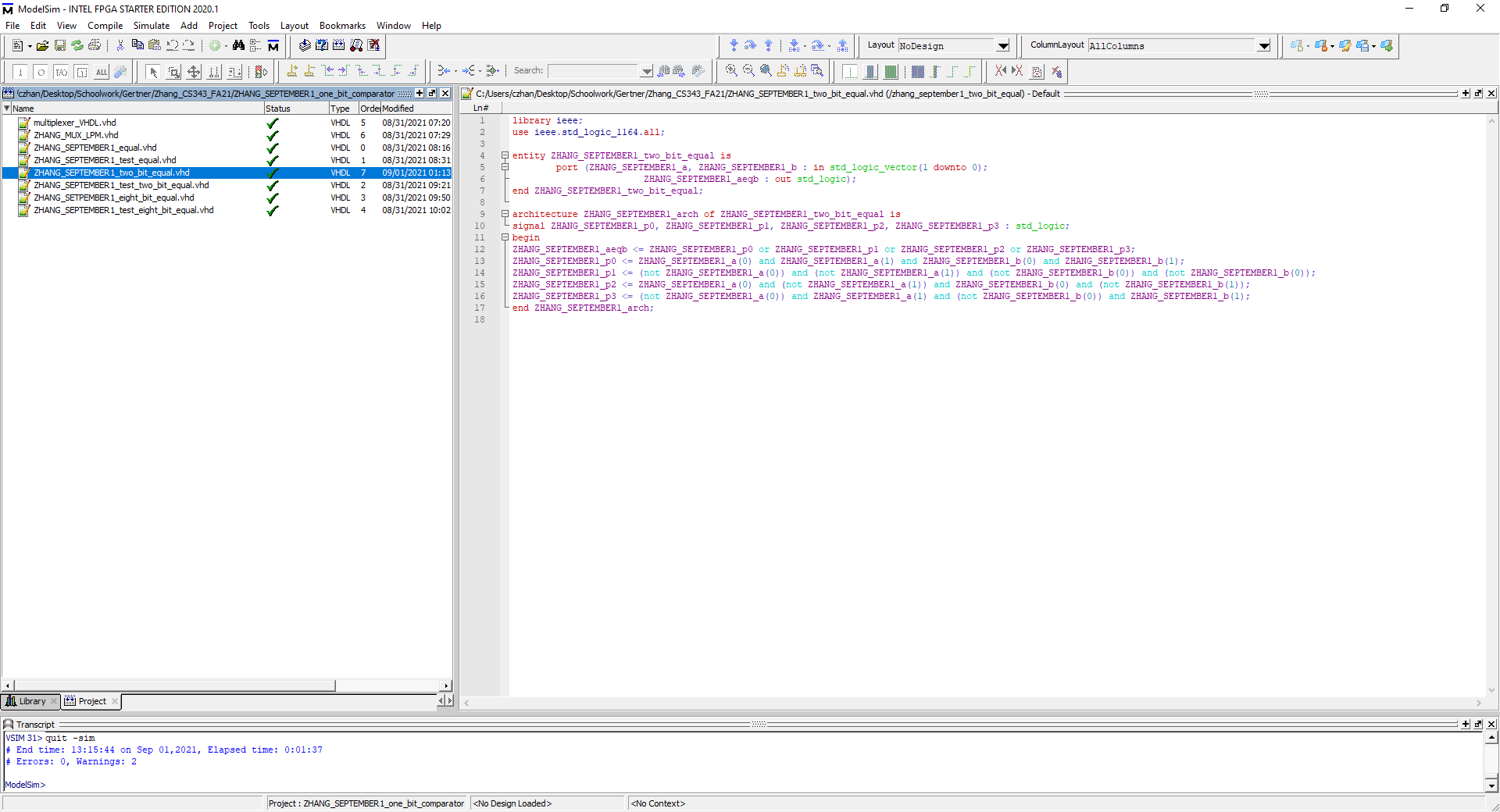
Verification of correctness through the use of test bench. In VHDL code, there is an arrow pointing to the line stating that there were no errors detected.



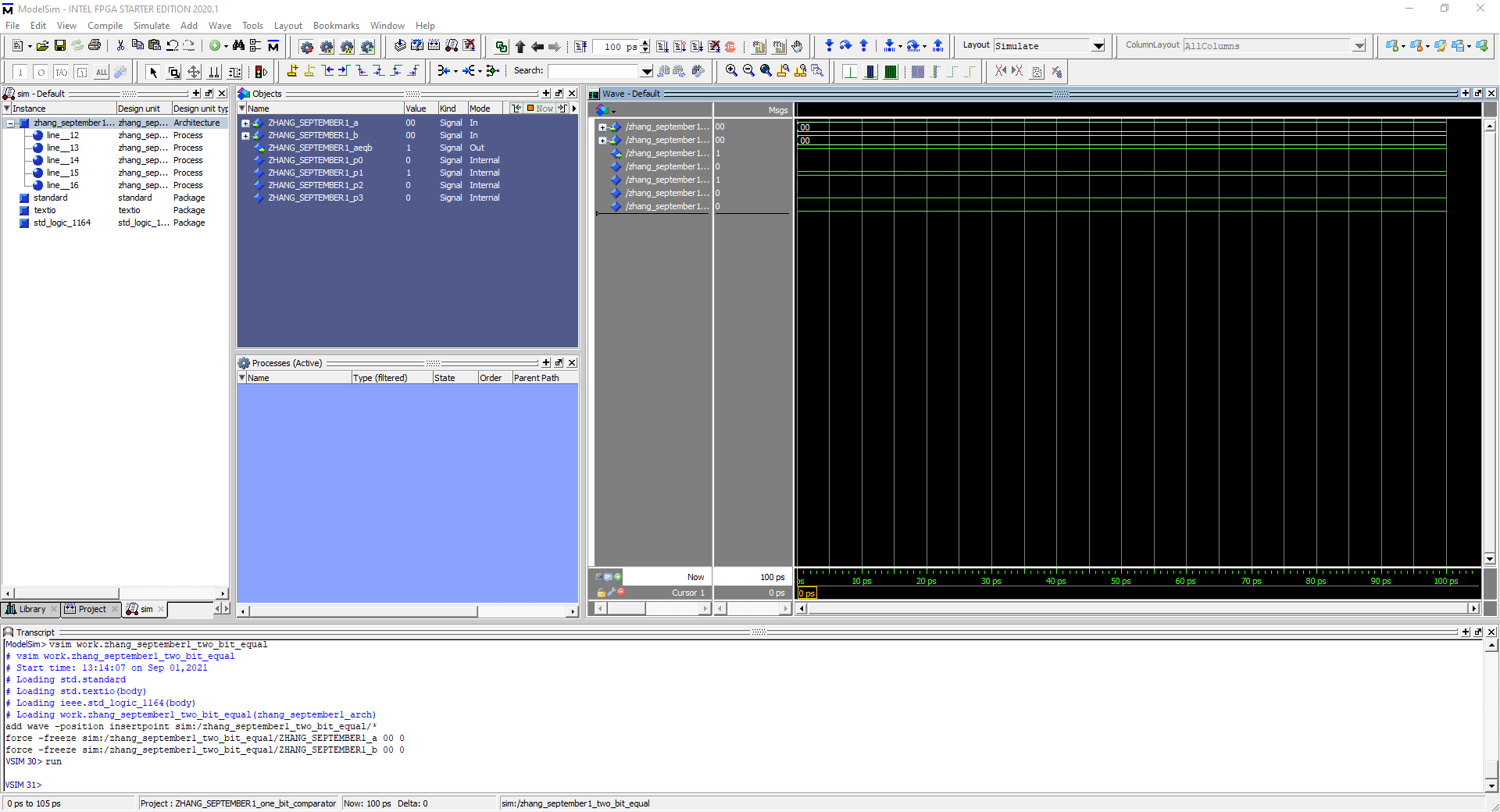
I first create a new file for the 2-bit comparator



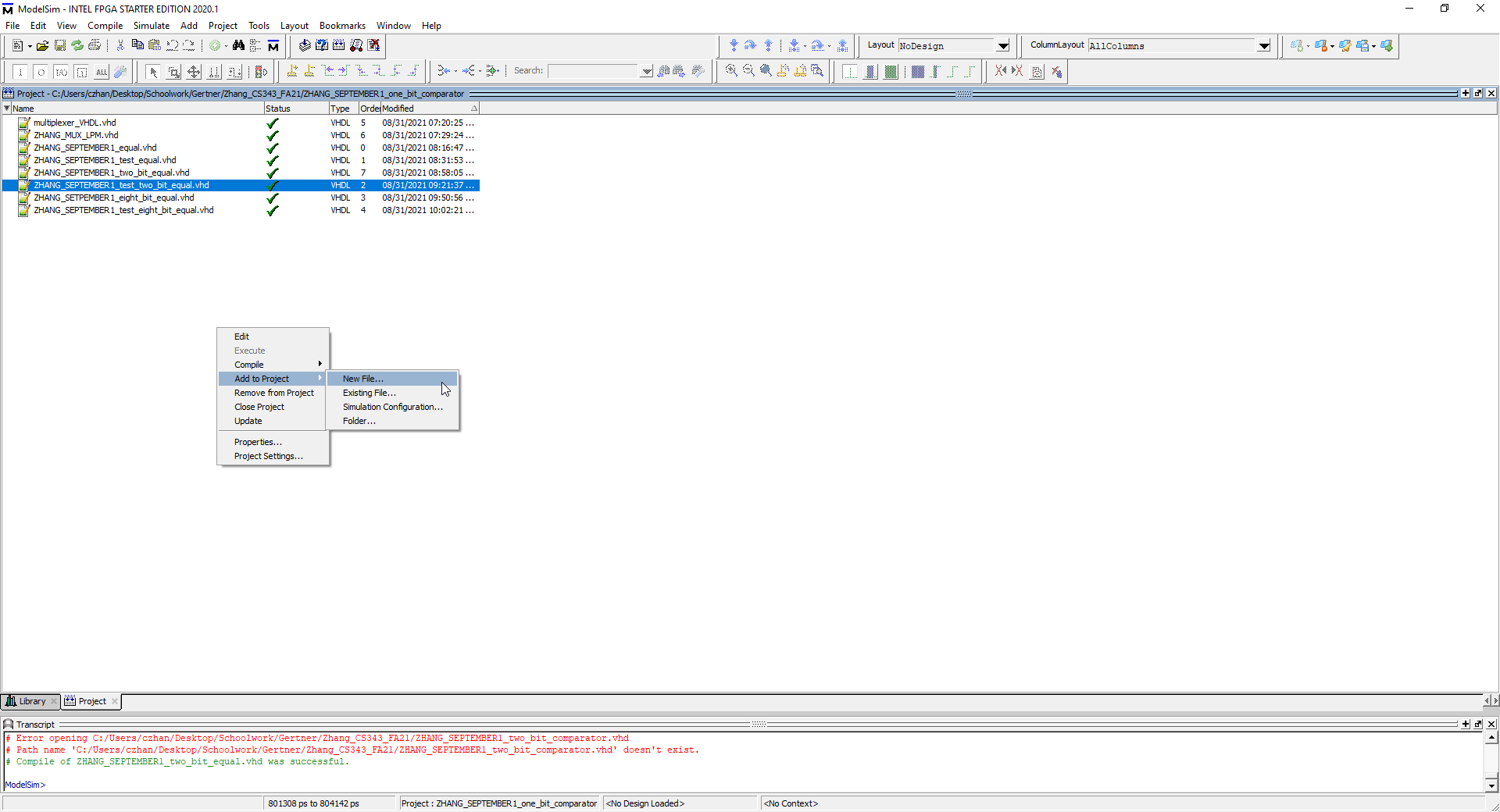
I double click the file and begin writing my VHDL code as seen on the right



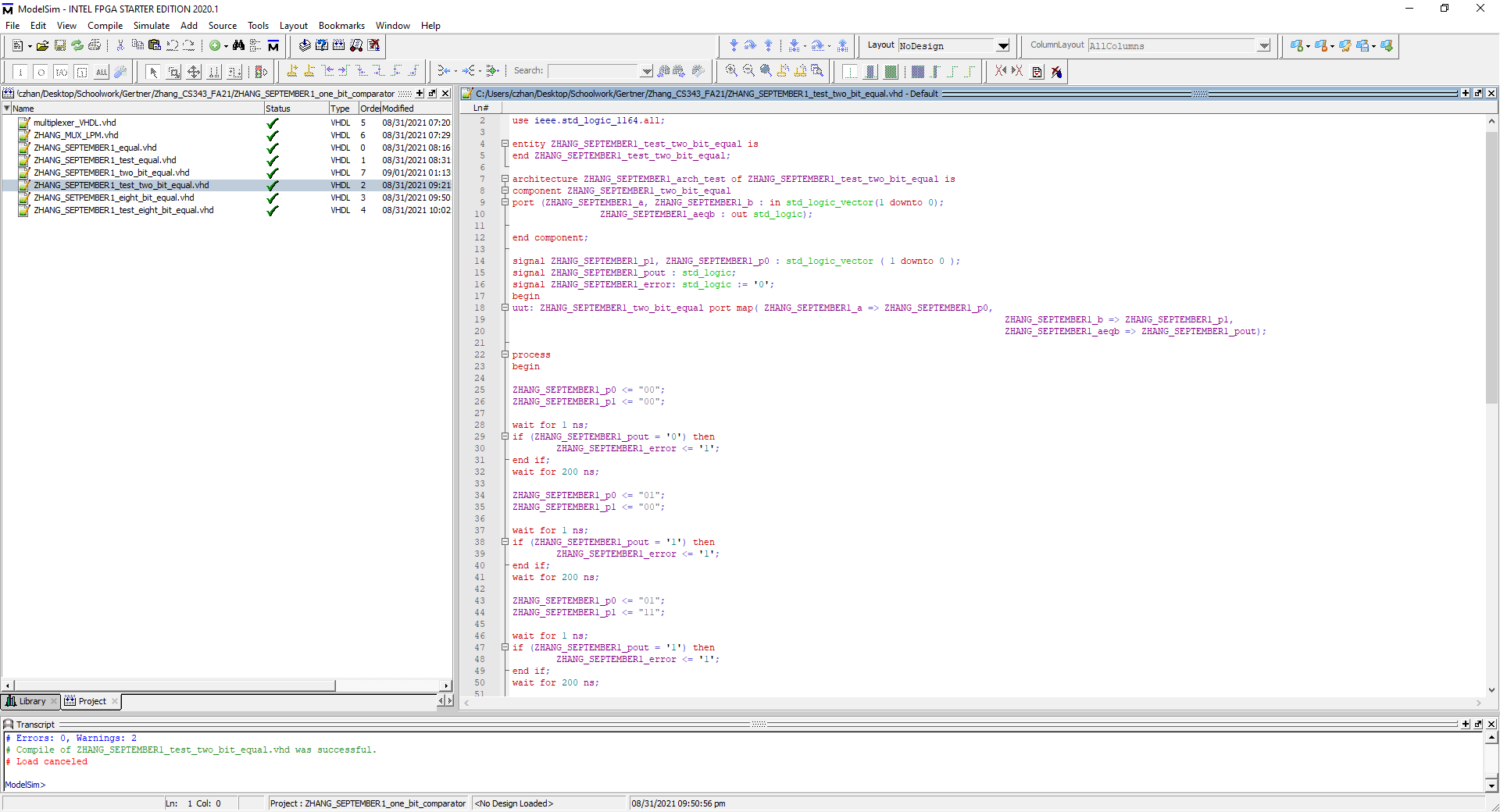
I select the two bit comparator VHDL file and right click 🡪 compile 🡪 compile selected. At the bottom it should show GREEN with no errors stating that the VHDL file was compiled successfully



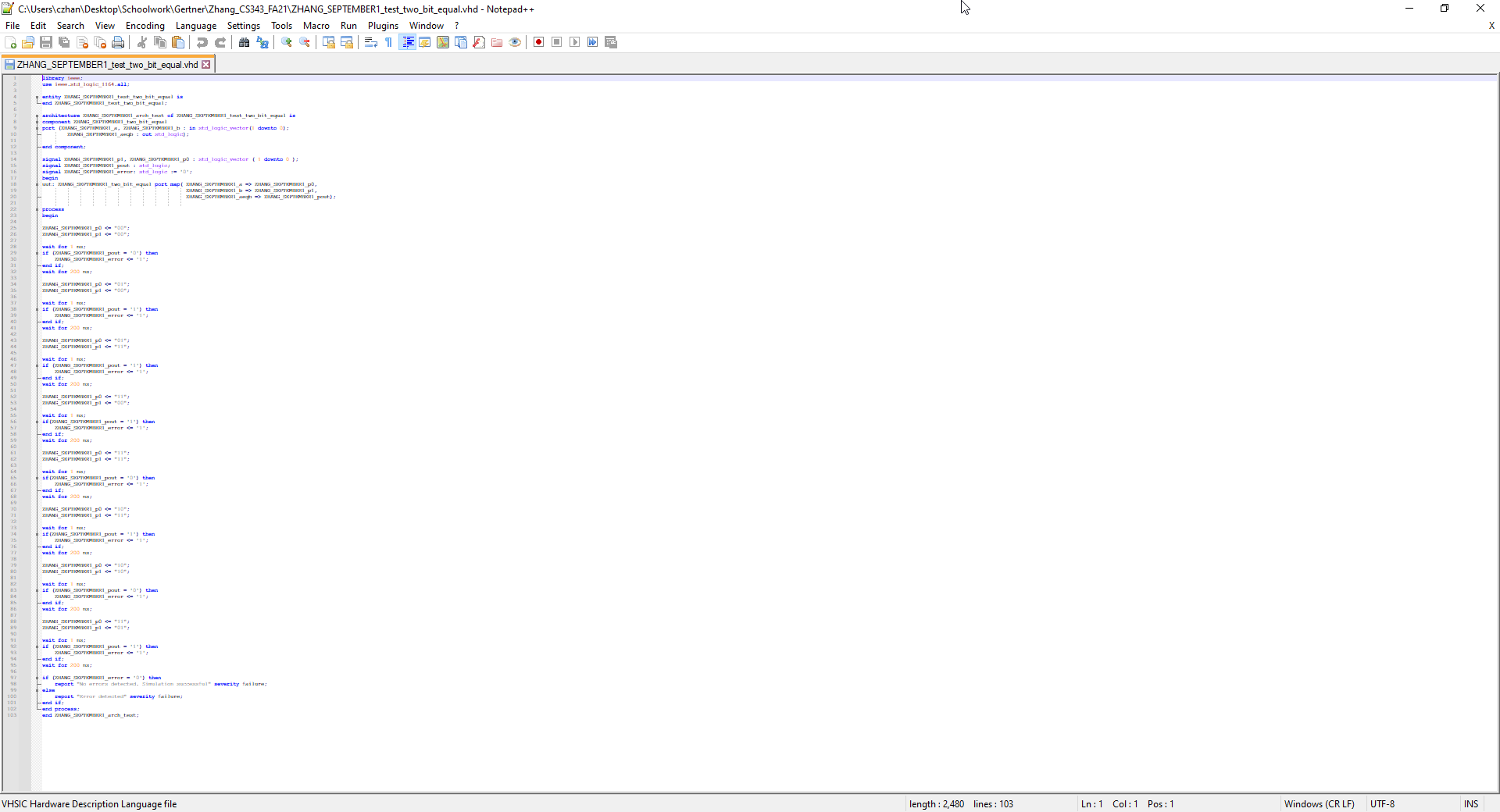
In this figure, what is shown is the waveform model for testing the inputs 00,00 which shows the appropriate output



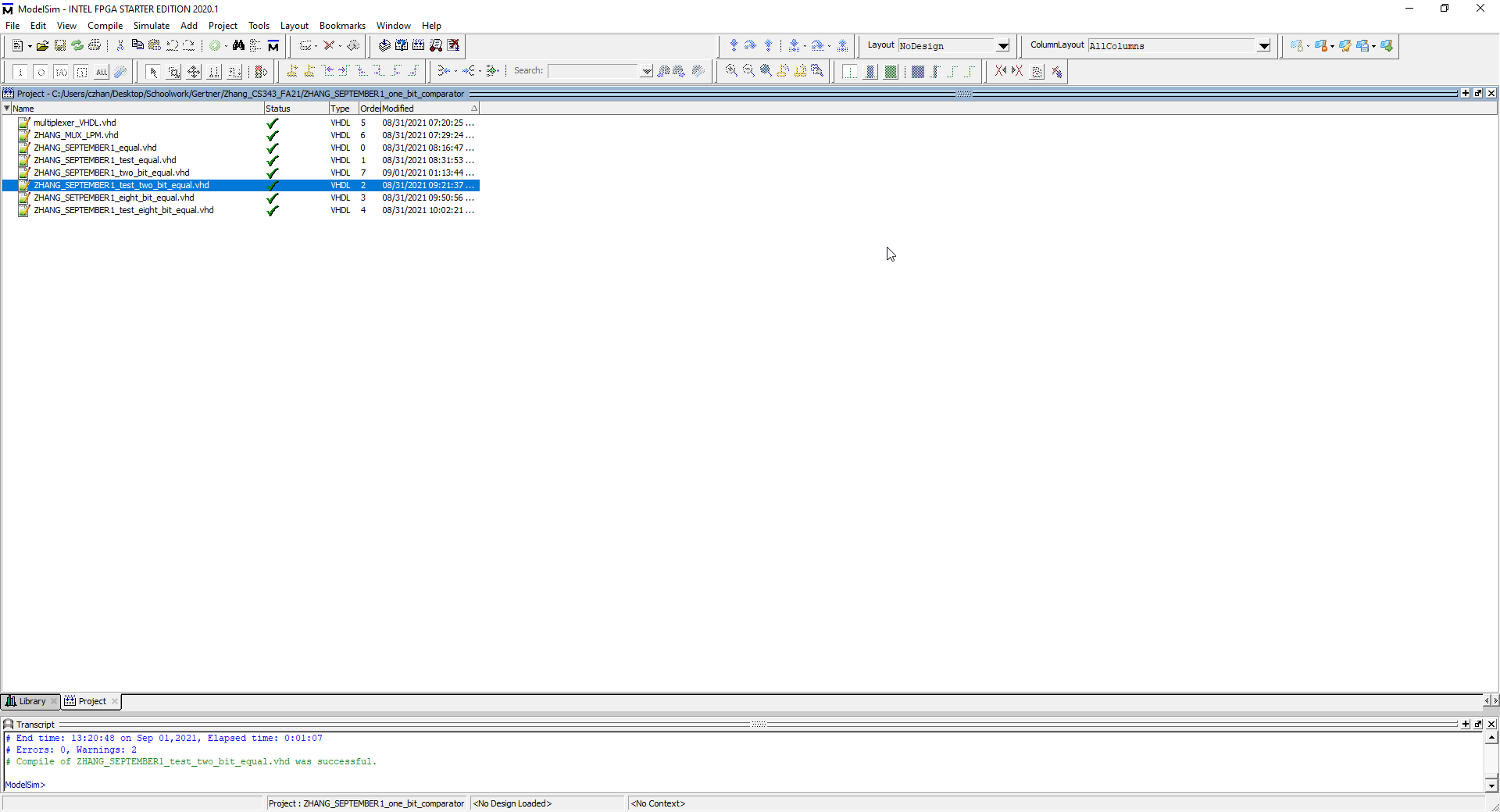
I am now going to create a new file for the test bench



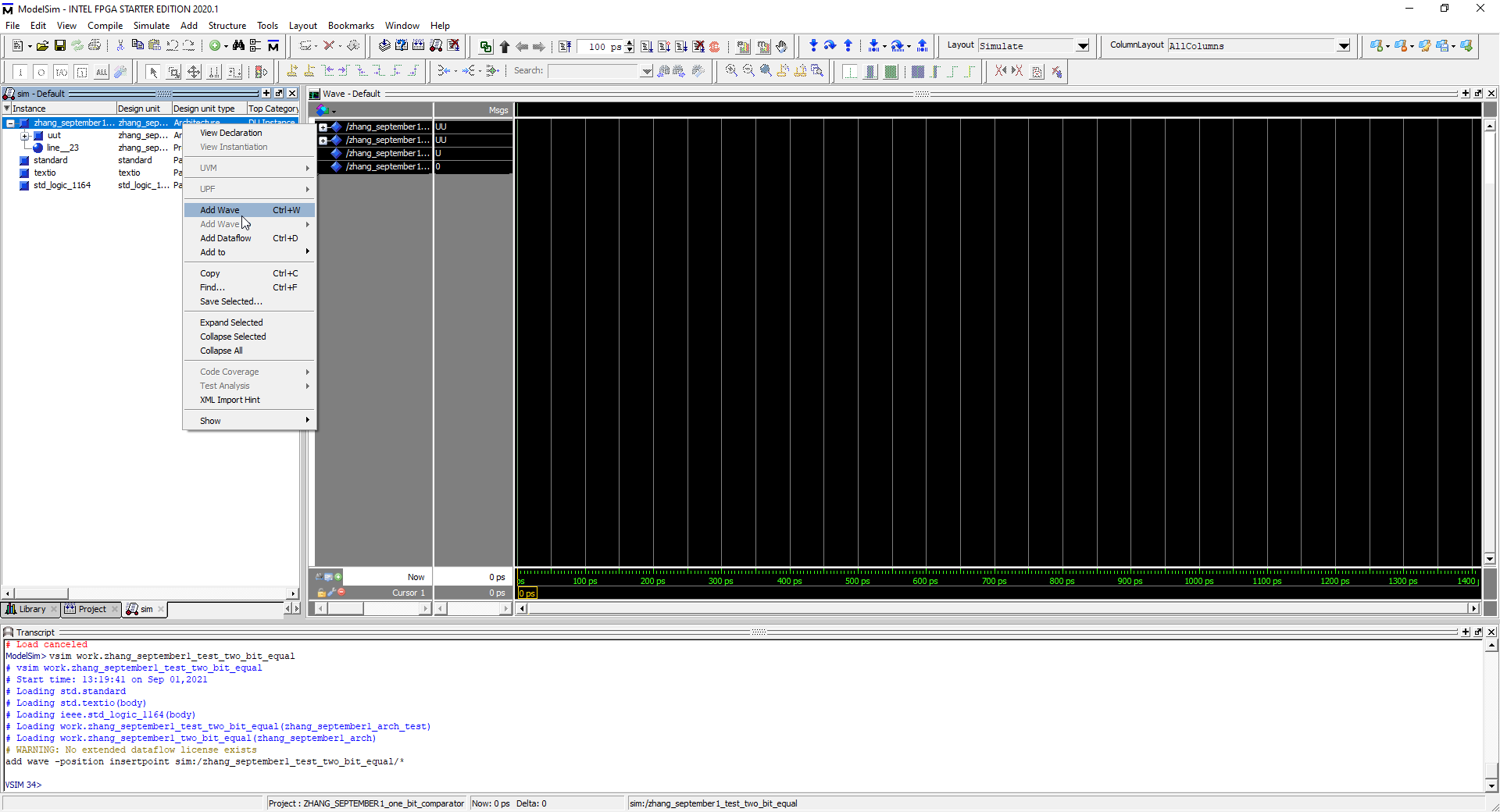
Over here, I have the written test bench code and I will now proceed to compile to check for correctness in code.



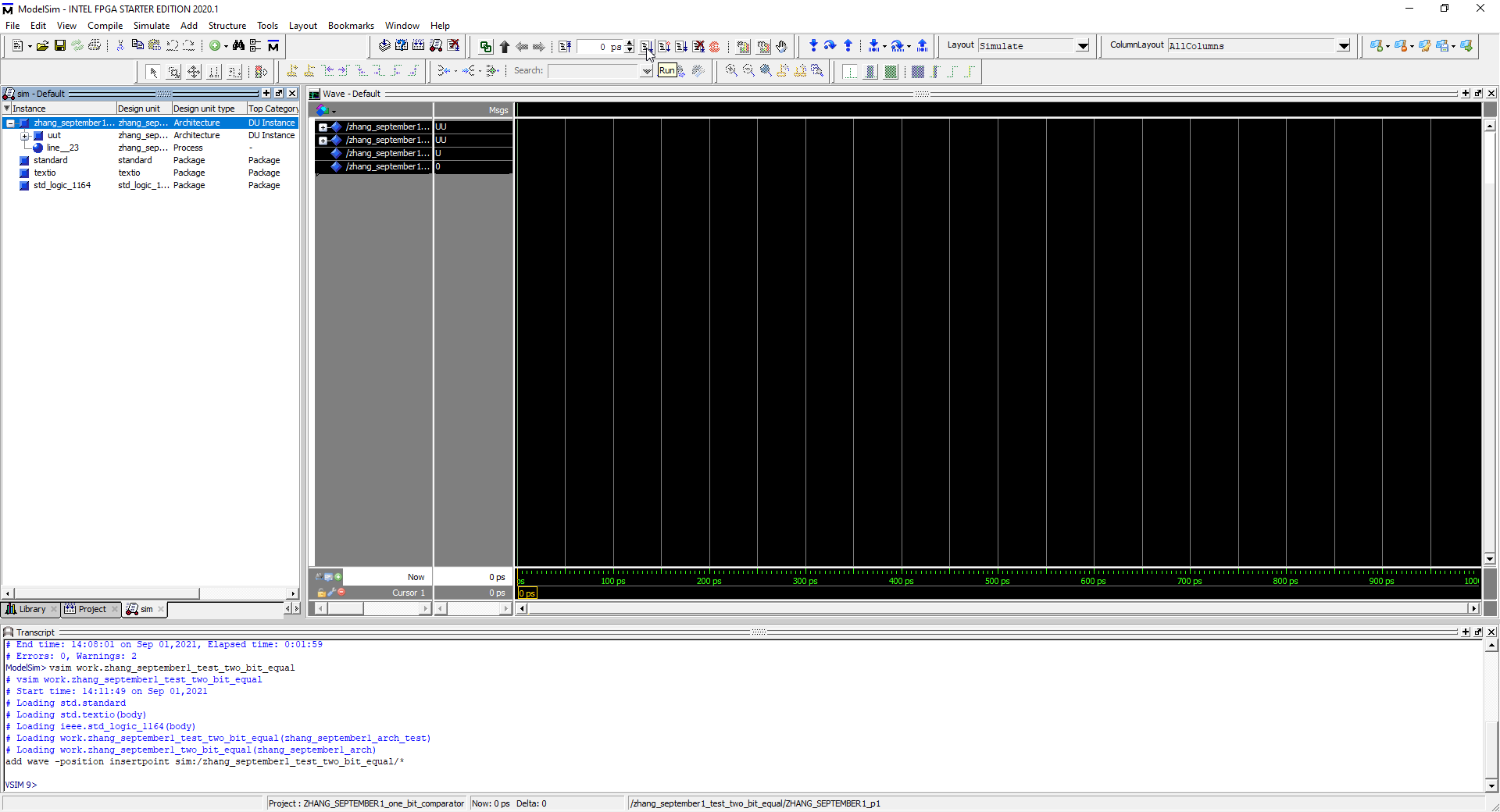
This is the entire code in a screen shot



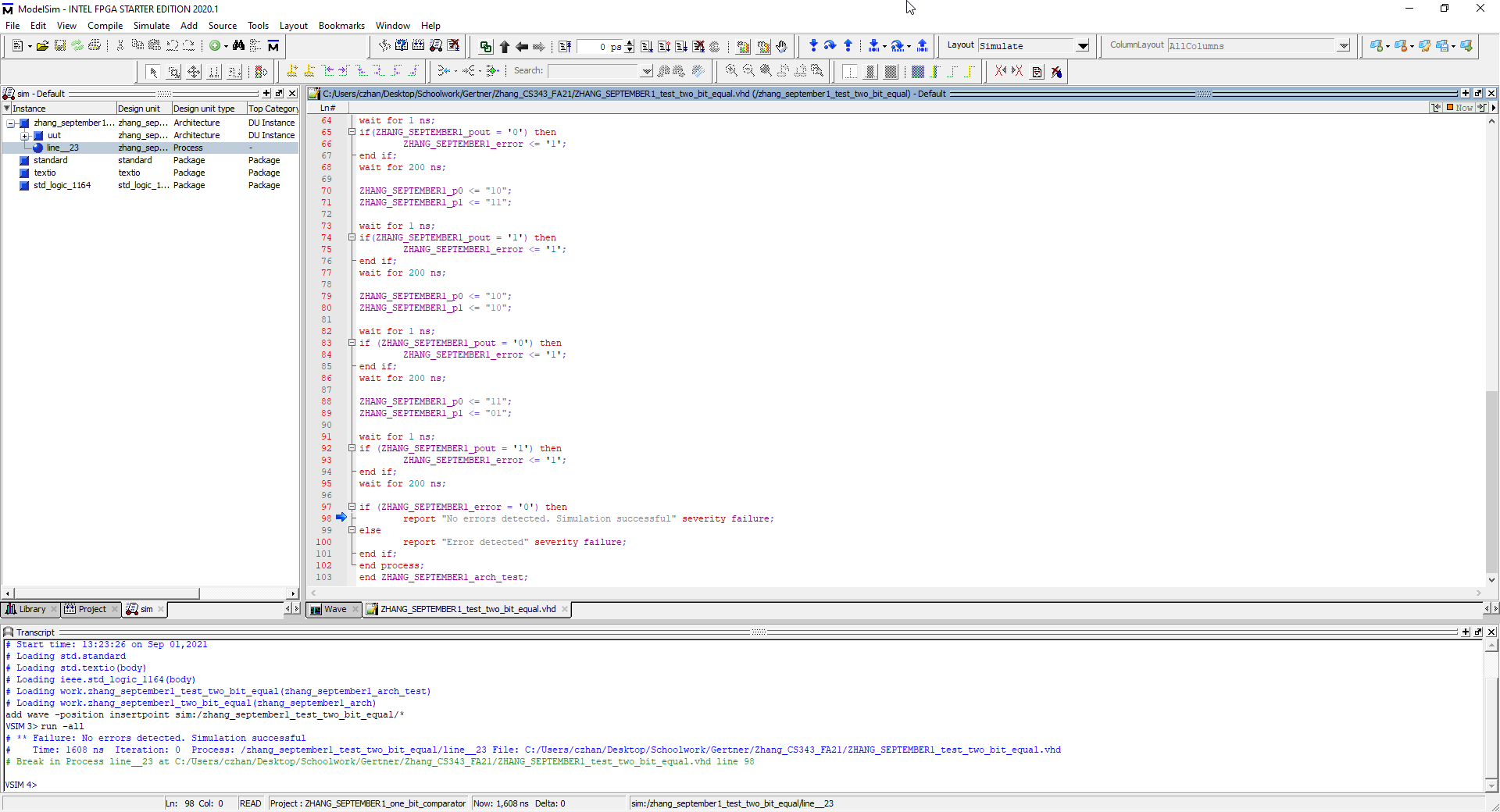
Compilation successful, will proceed to simulation on model sim



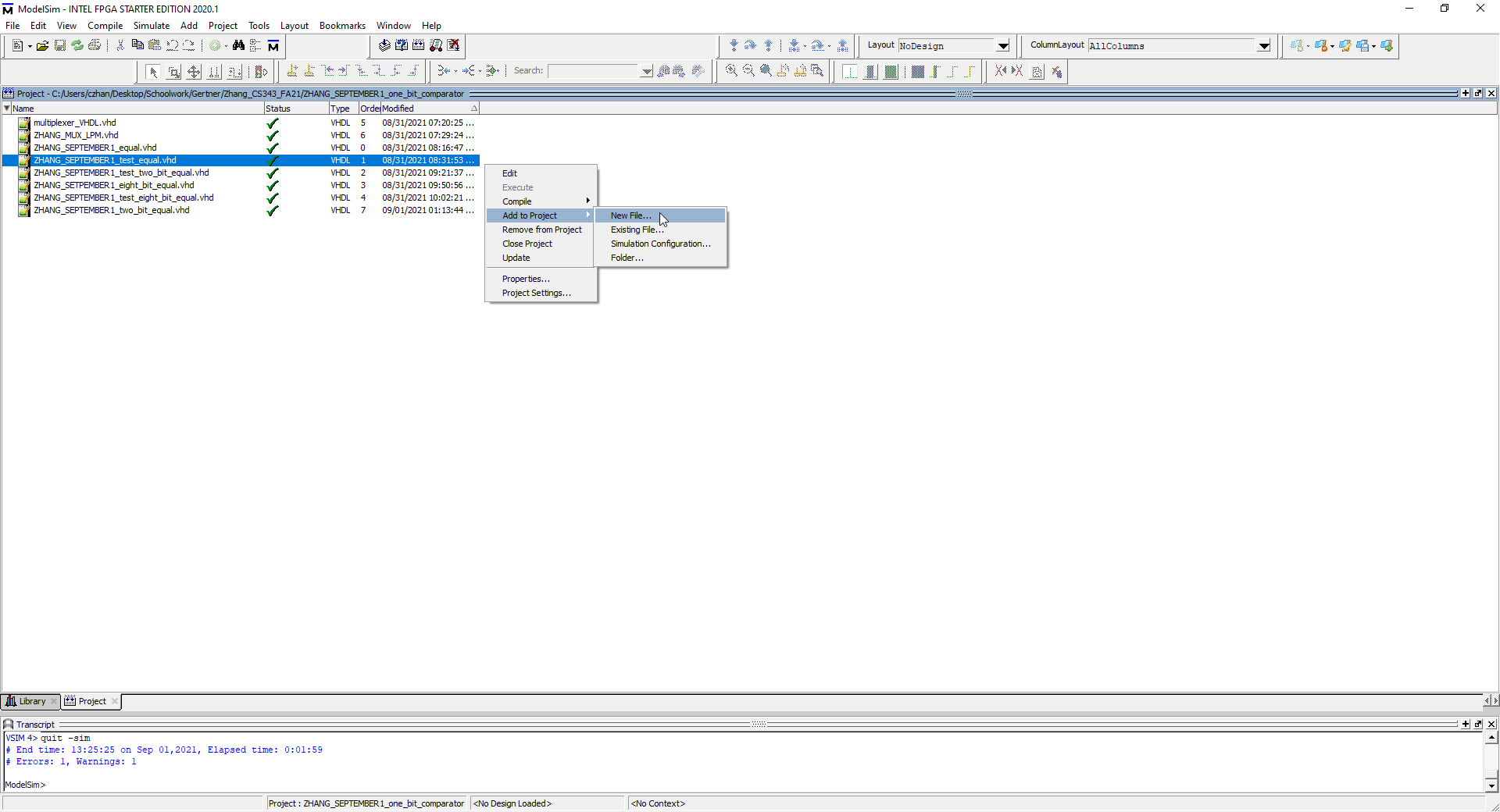
In modelsim I add the waves by pressing right click 🡪 add wave



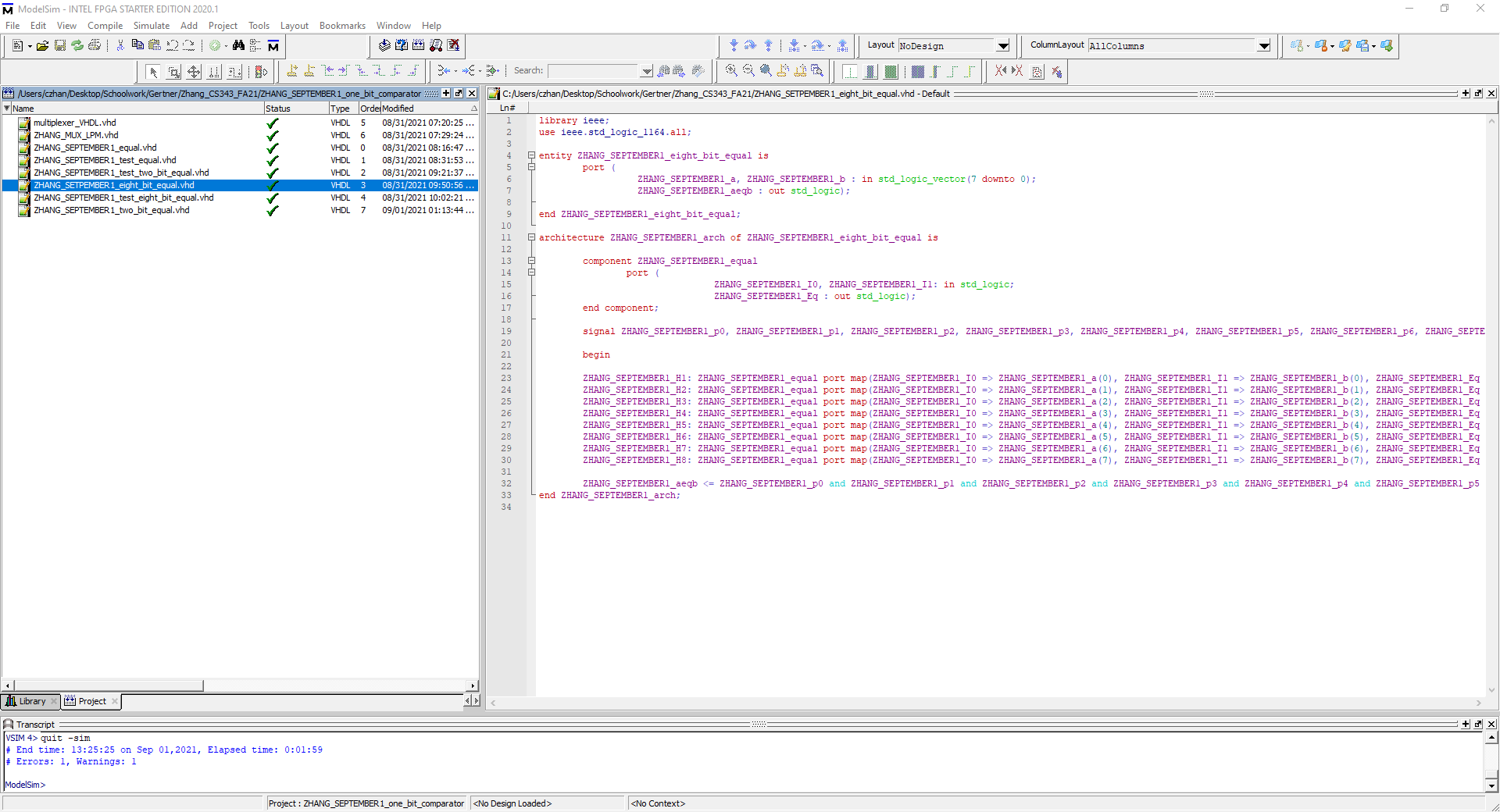
Appropirate waves have been added and I proceed to run-all



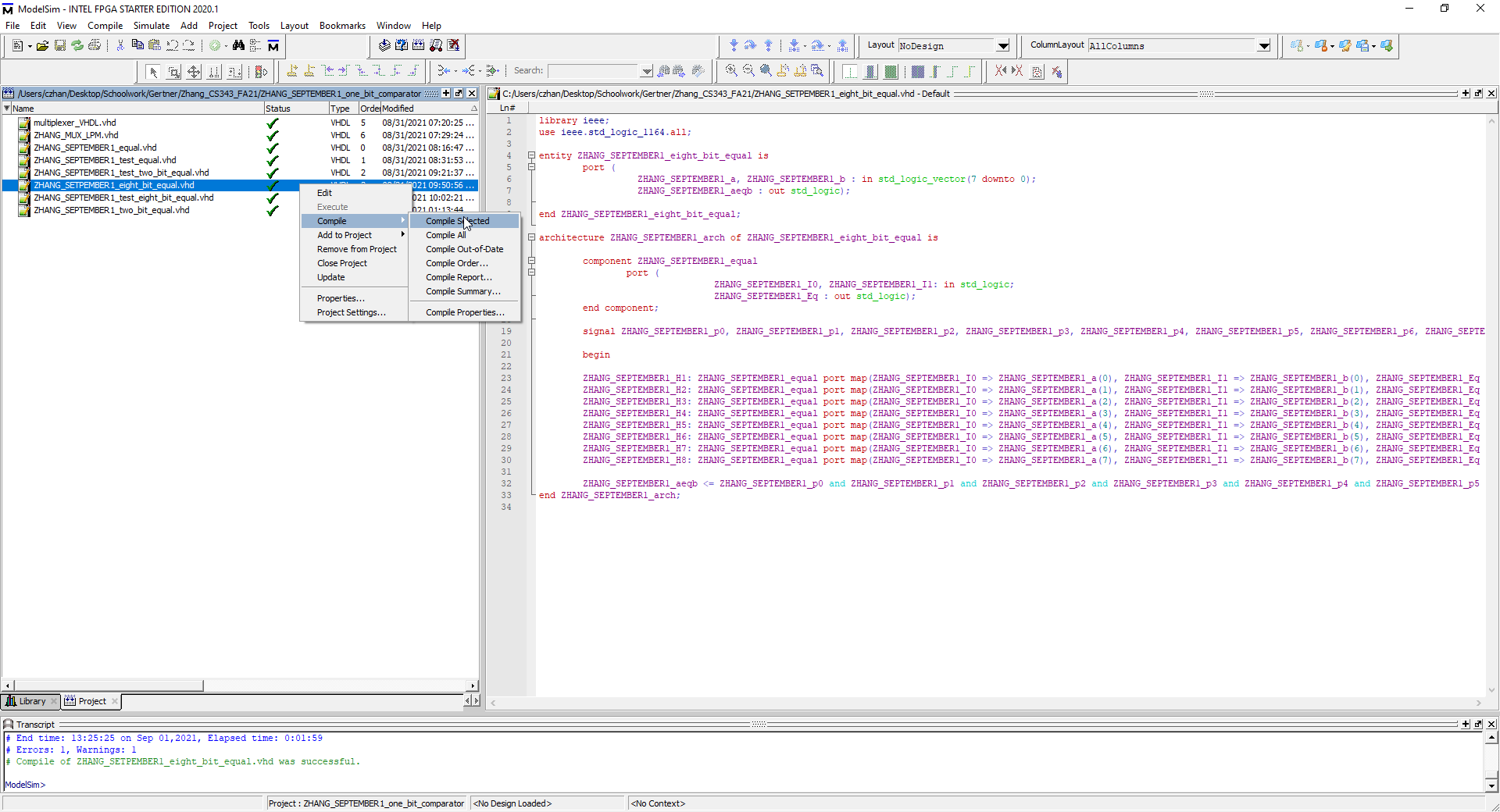
I press run all and there is no errors created so test bench is successful.



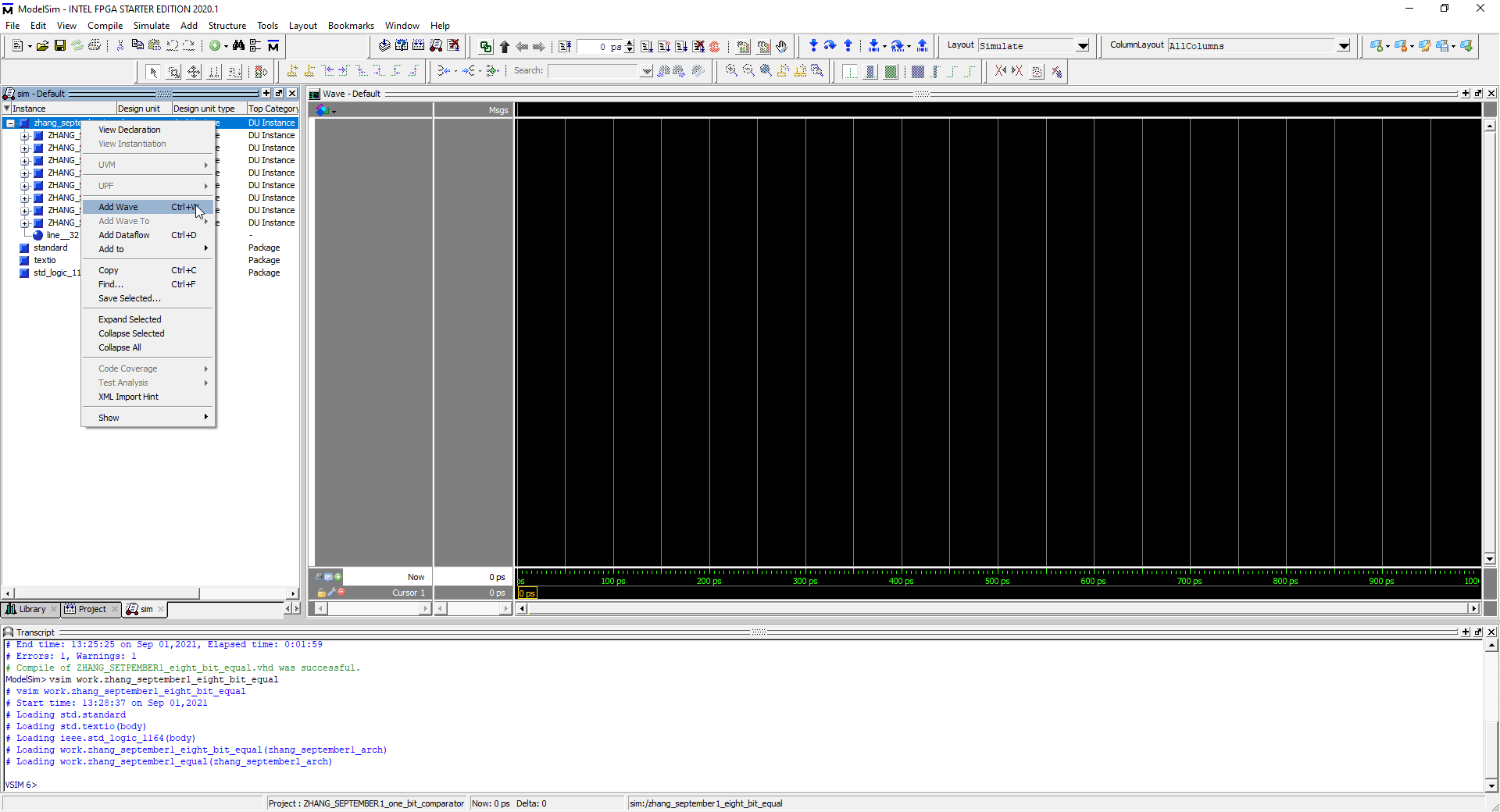
I am now going to create a new VHDL file for 8 bit comparator



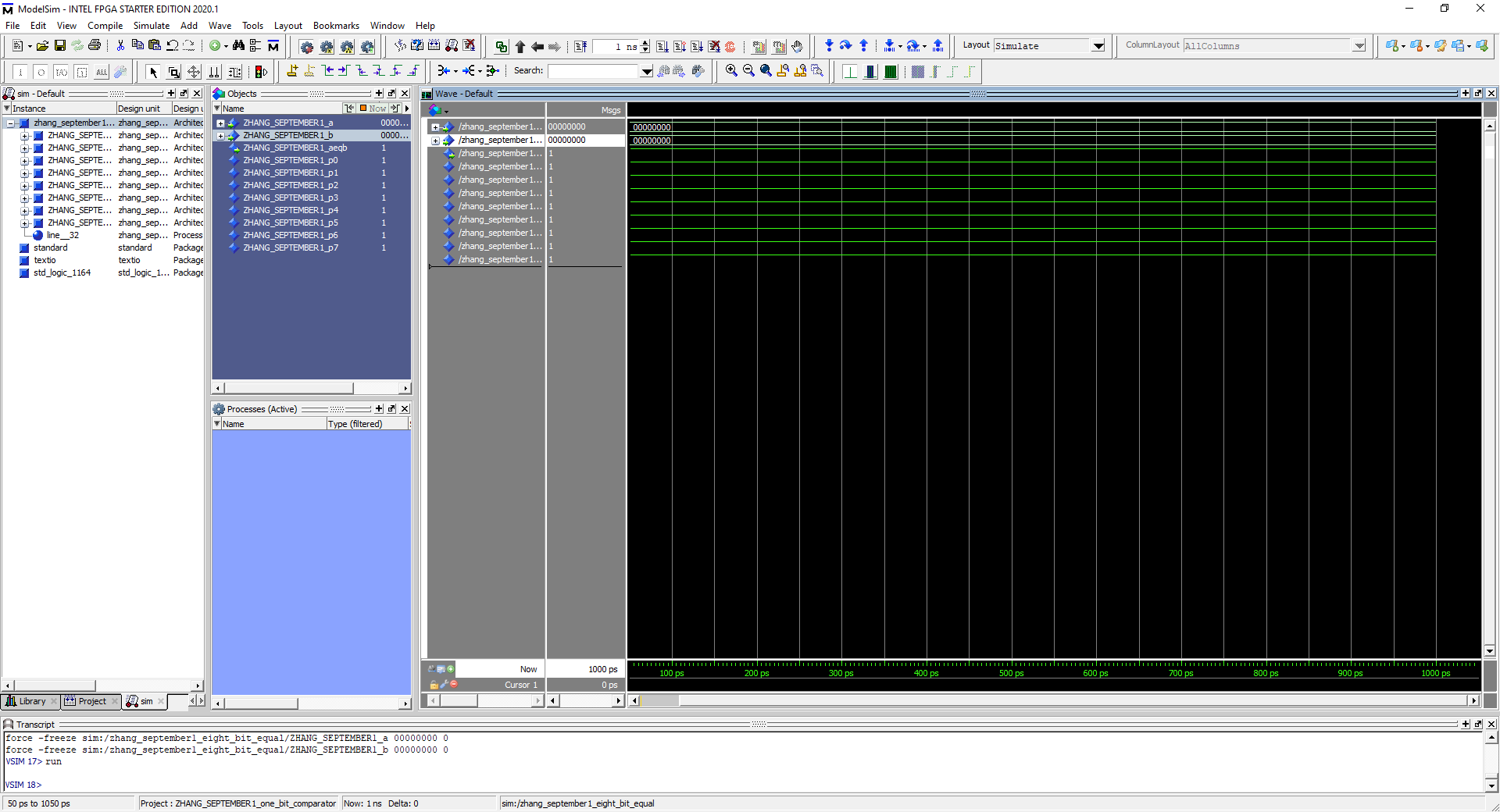
This is the vhdl code created for the eight bit comparator that I have written



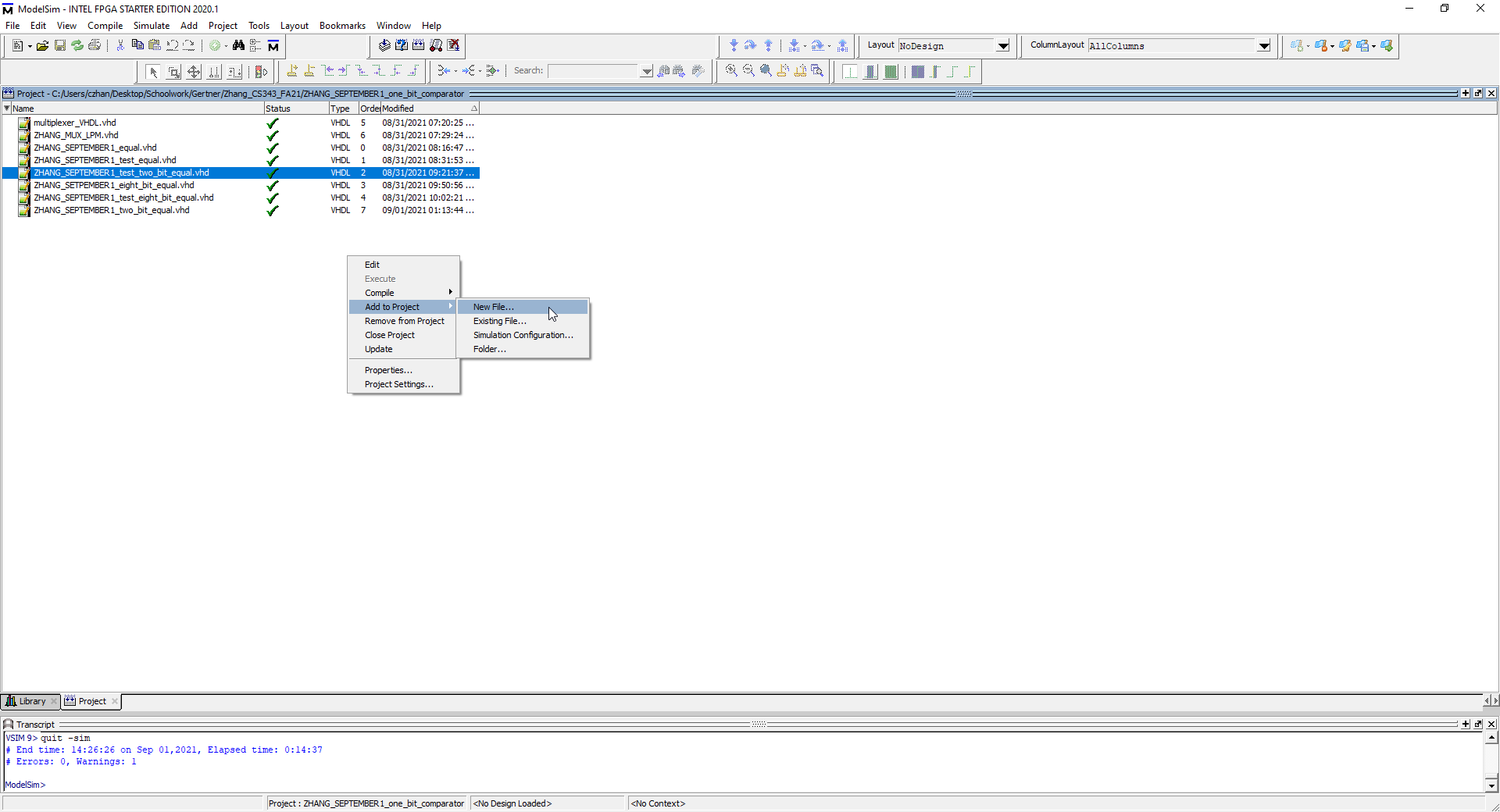
I compile with success as shown in the textbox at the bottom of the figure



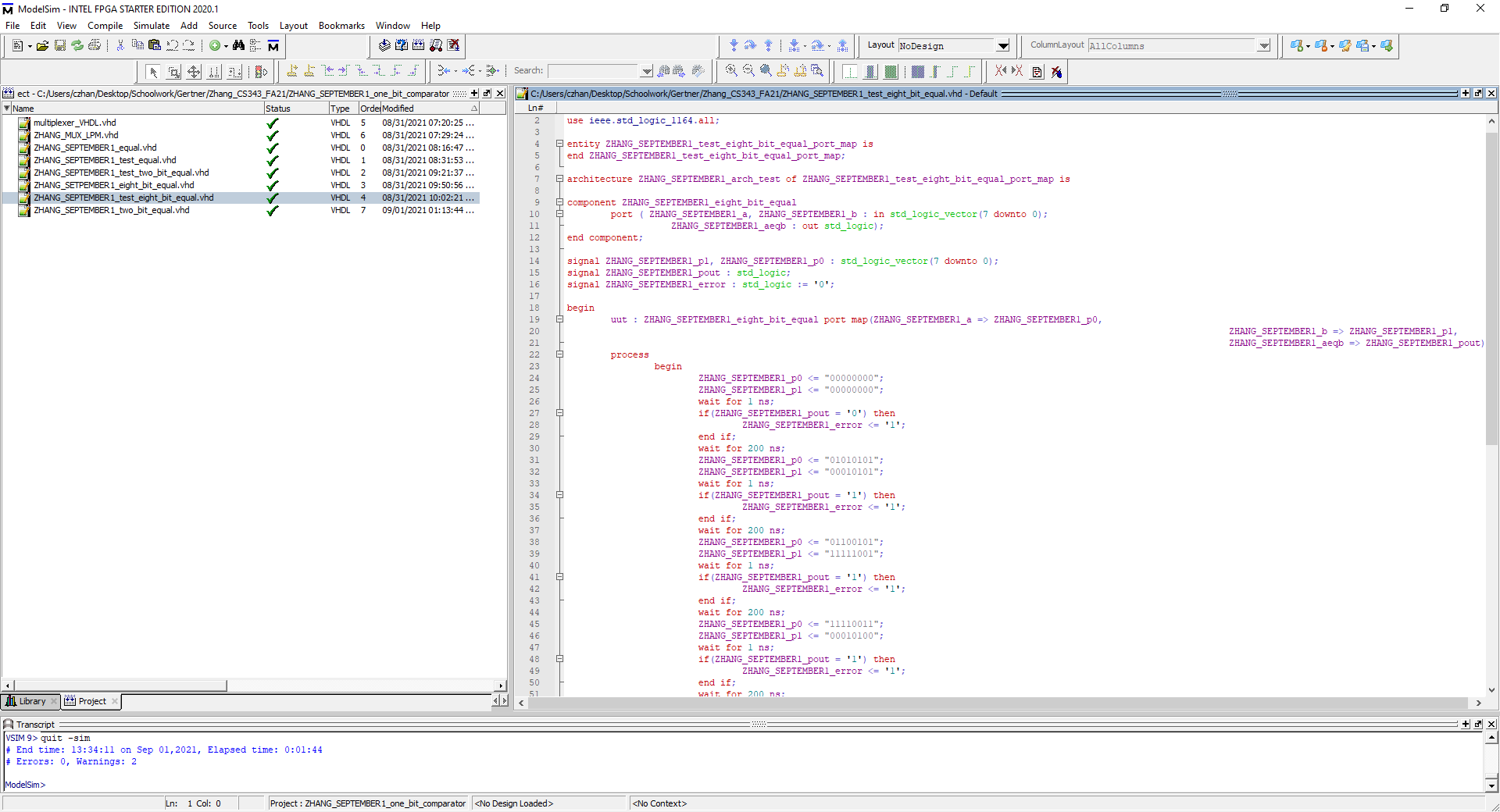
I begin simulating in modelsim by adding the waves to be simulated with



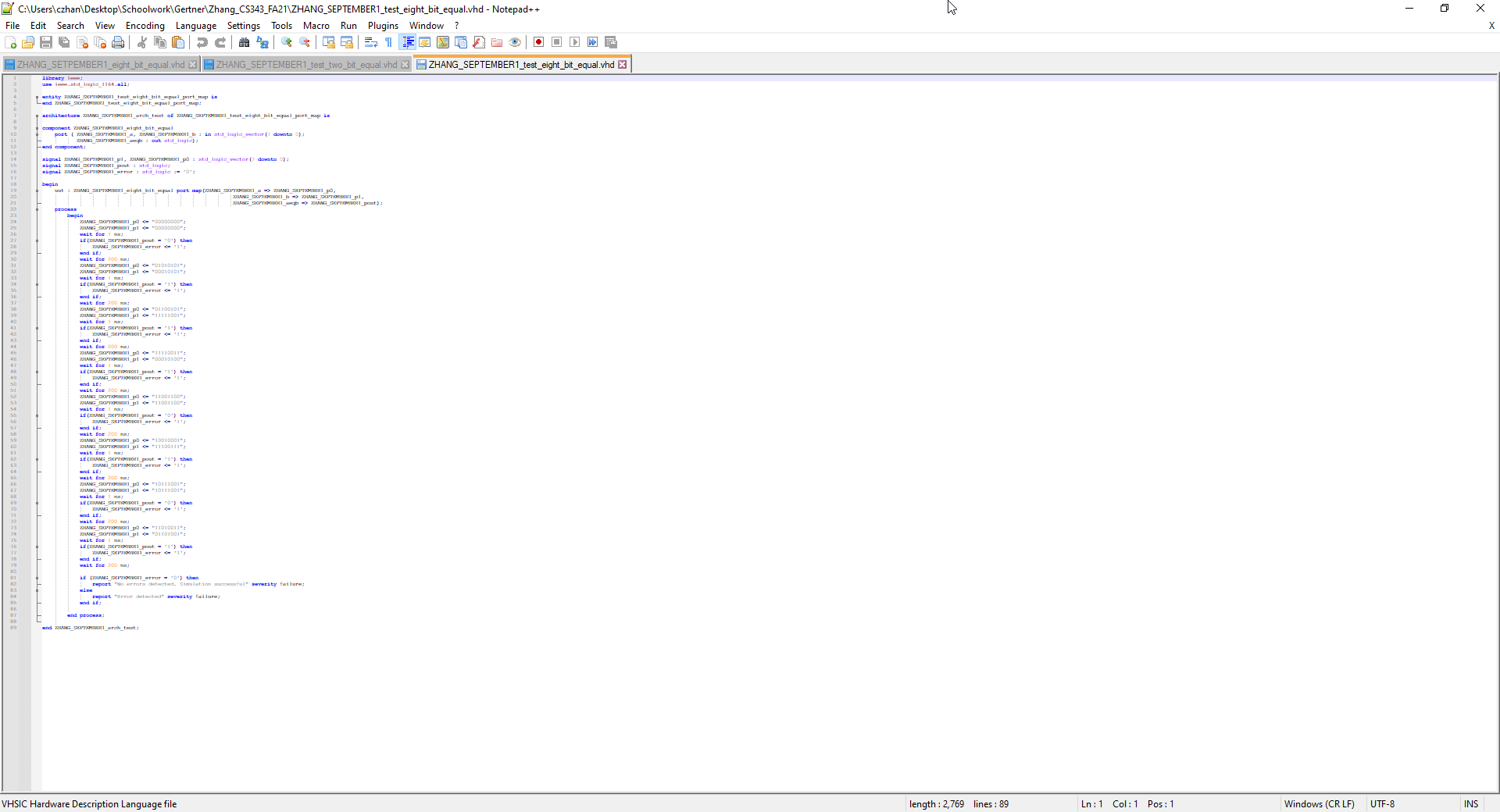
This was the modelsim Output, as you can see in it, output is 1 which is correct



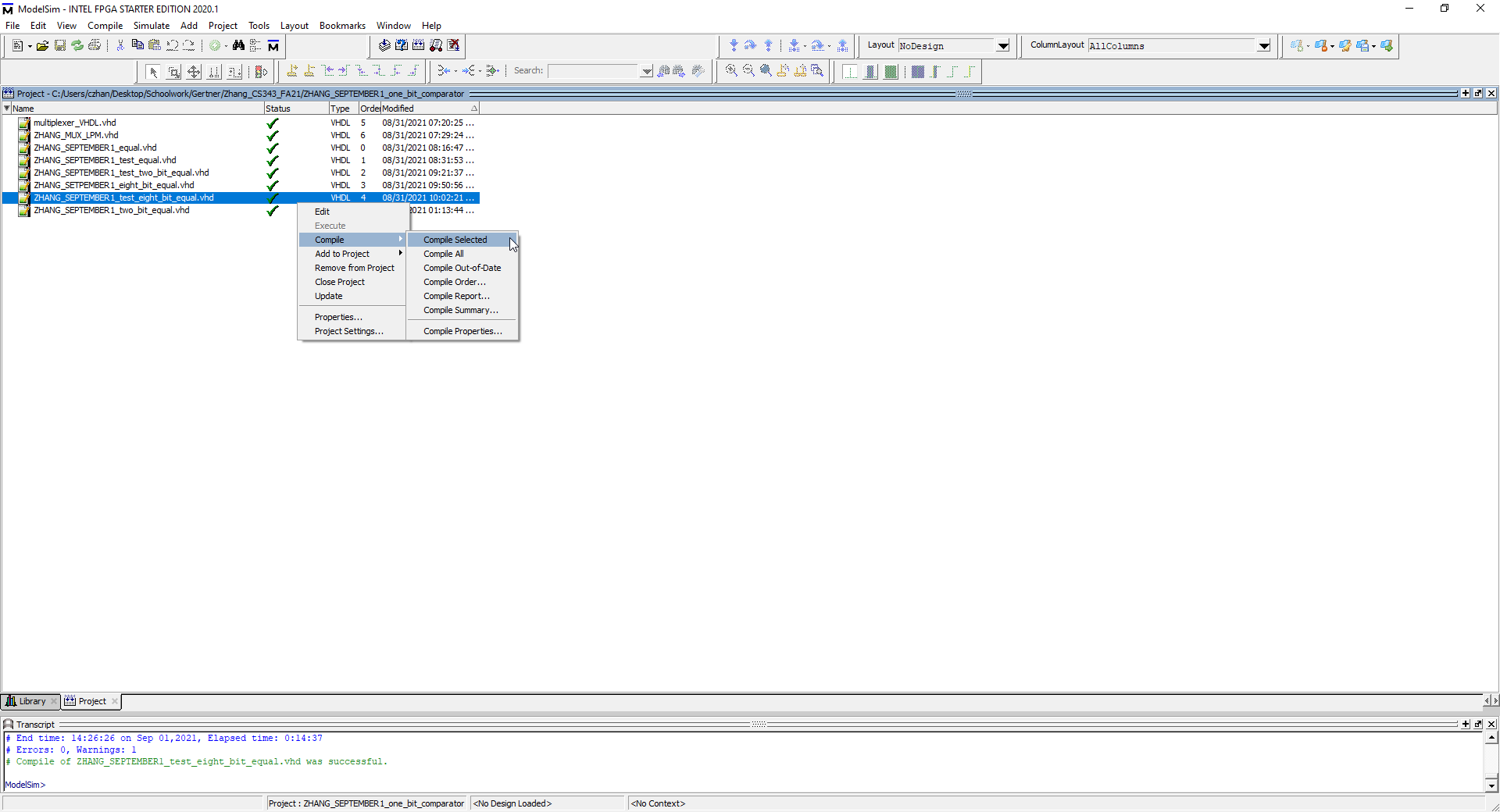
Next I create another file for the test bench



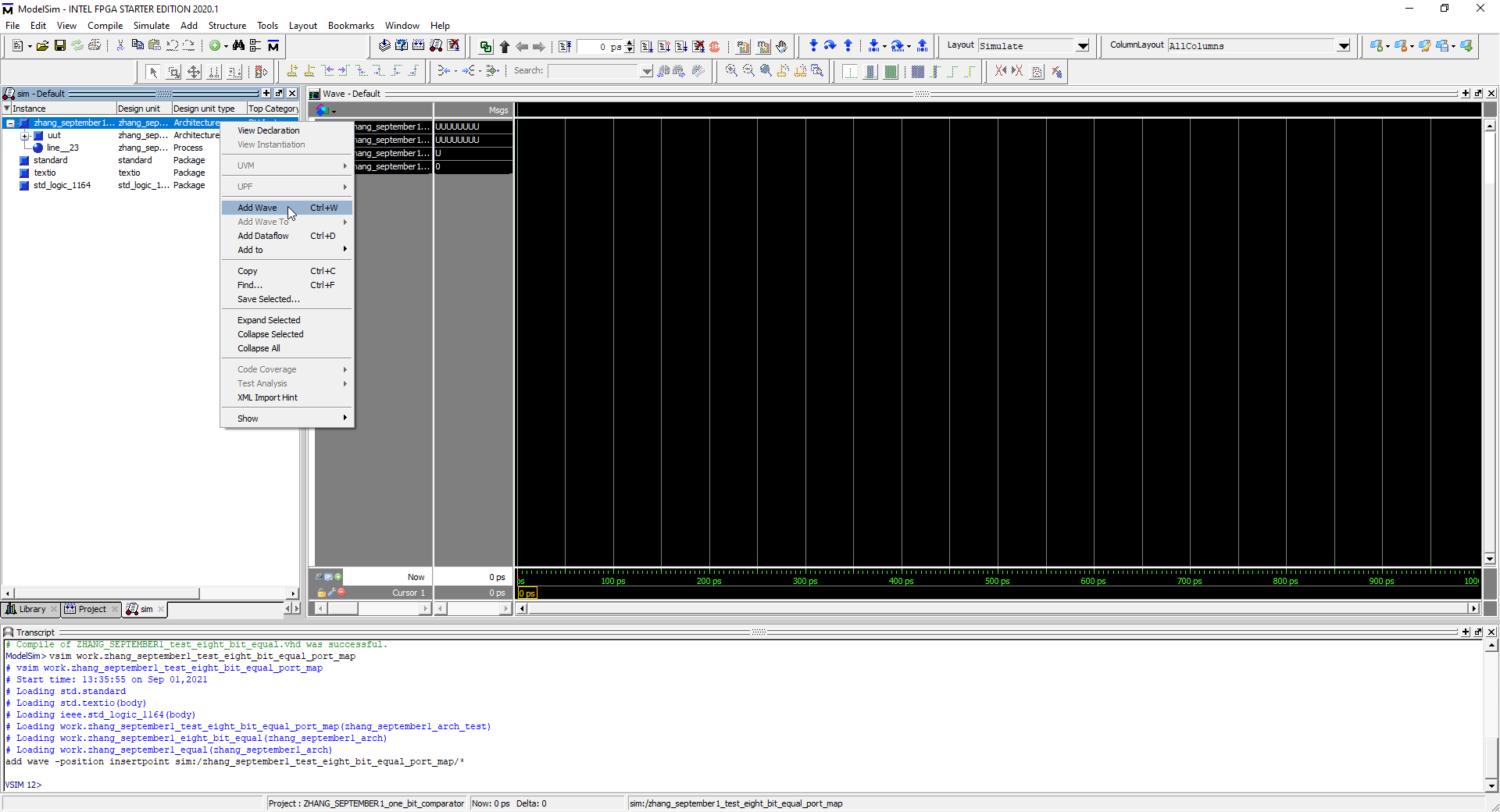
I begin writing VHDL code in the new file I have created



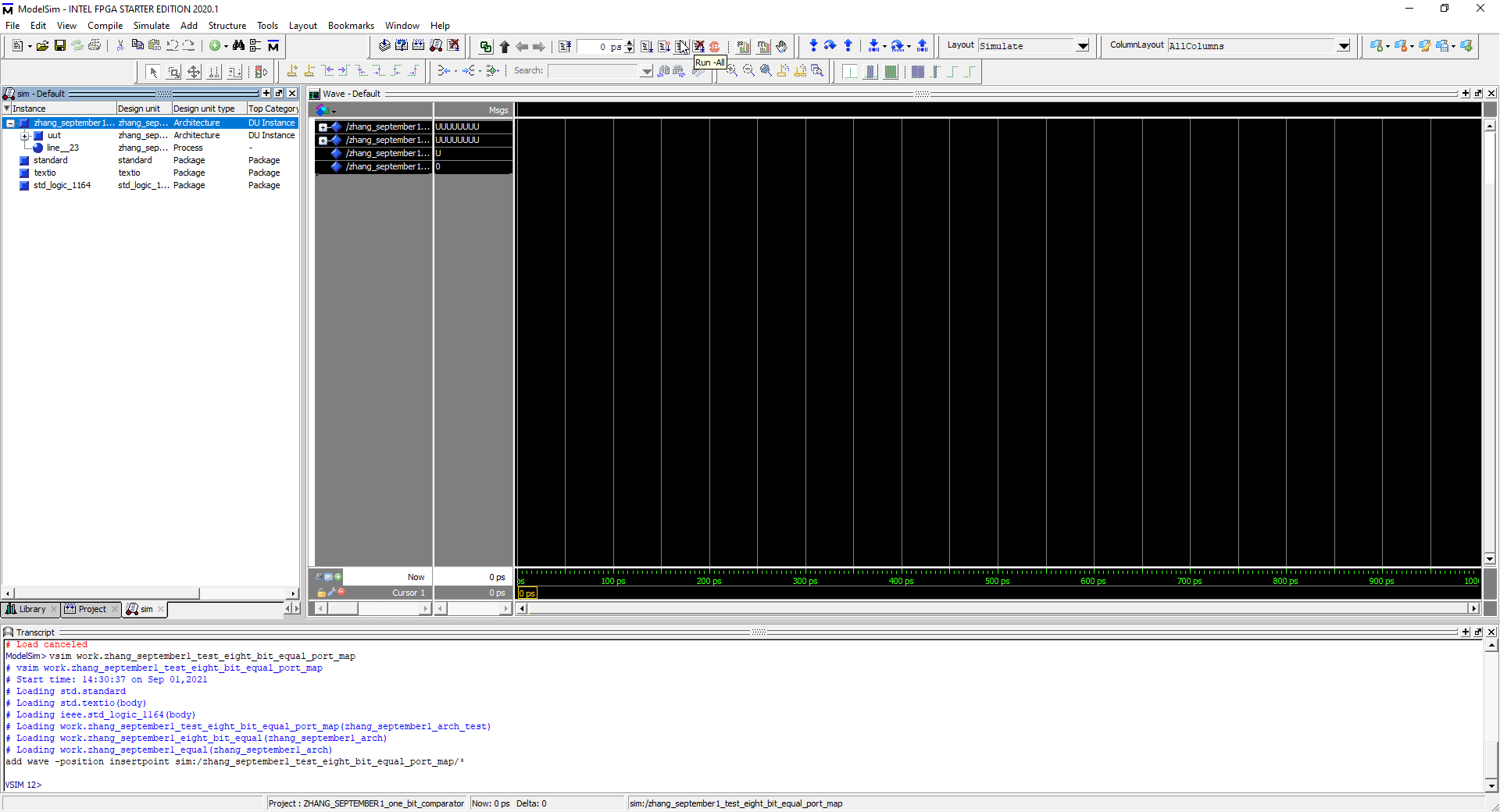
This is the entire code in a screenshot



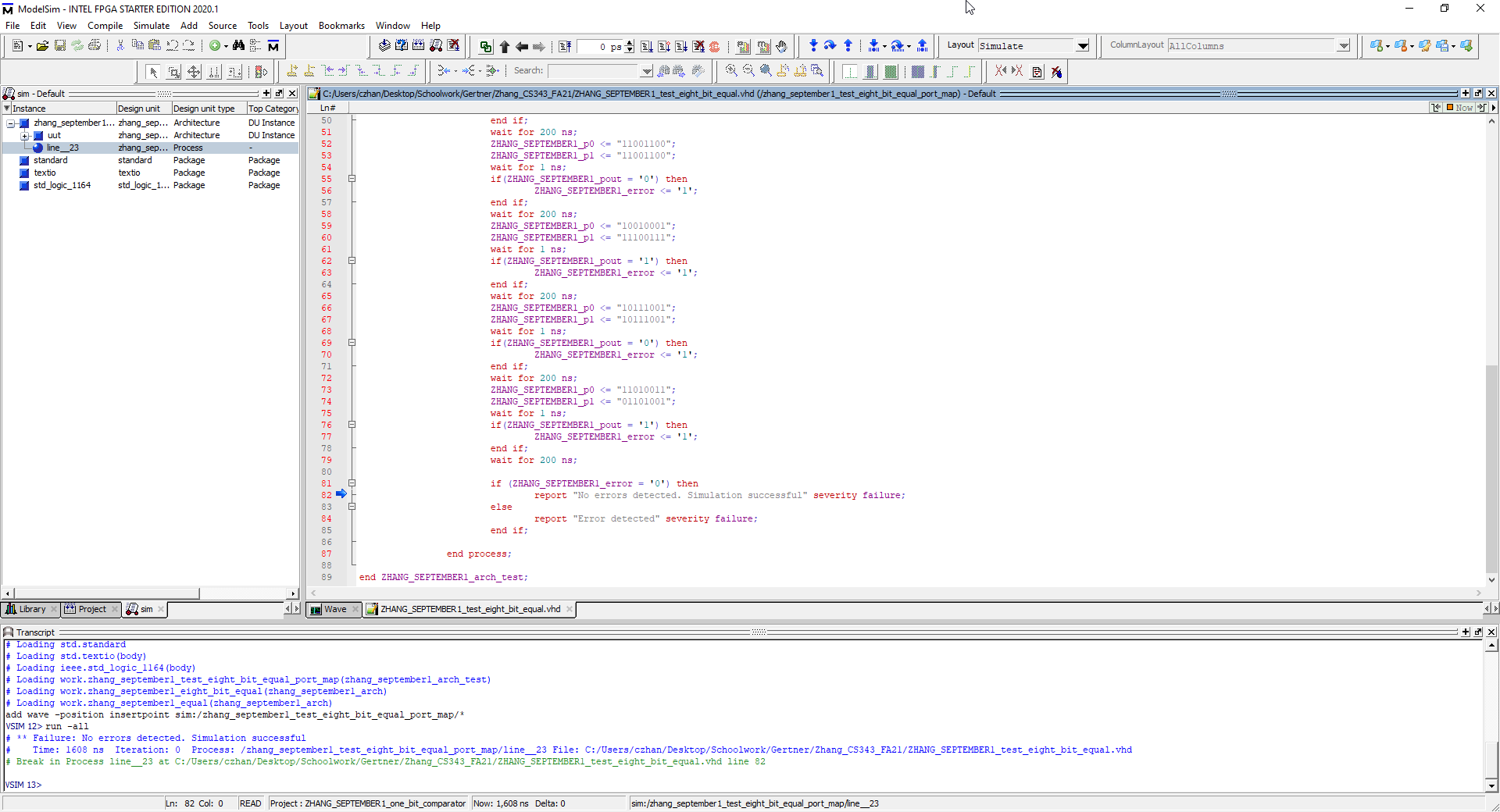
I compile the test bench VHDL file for the 8 bit comparator with no issues. Refer to green text on the bottom of image for verification of success



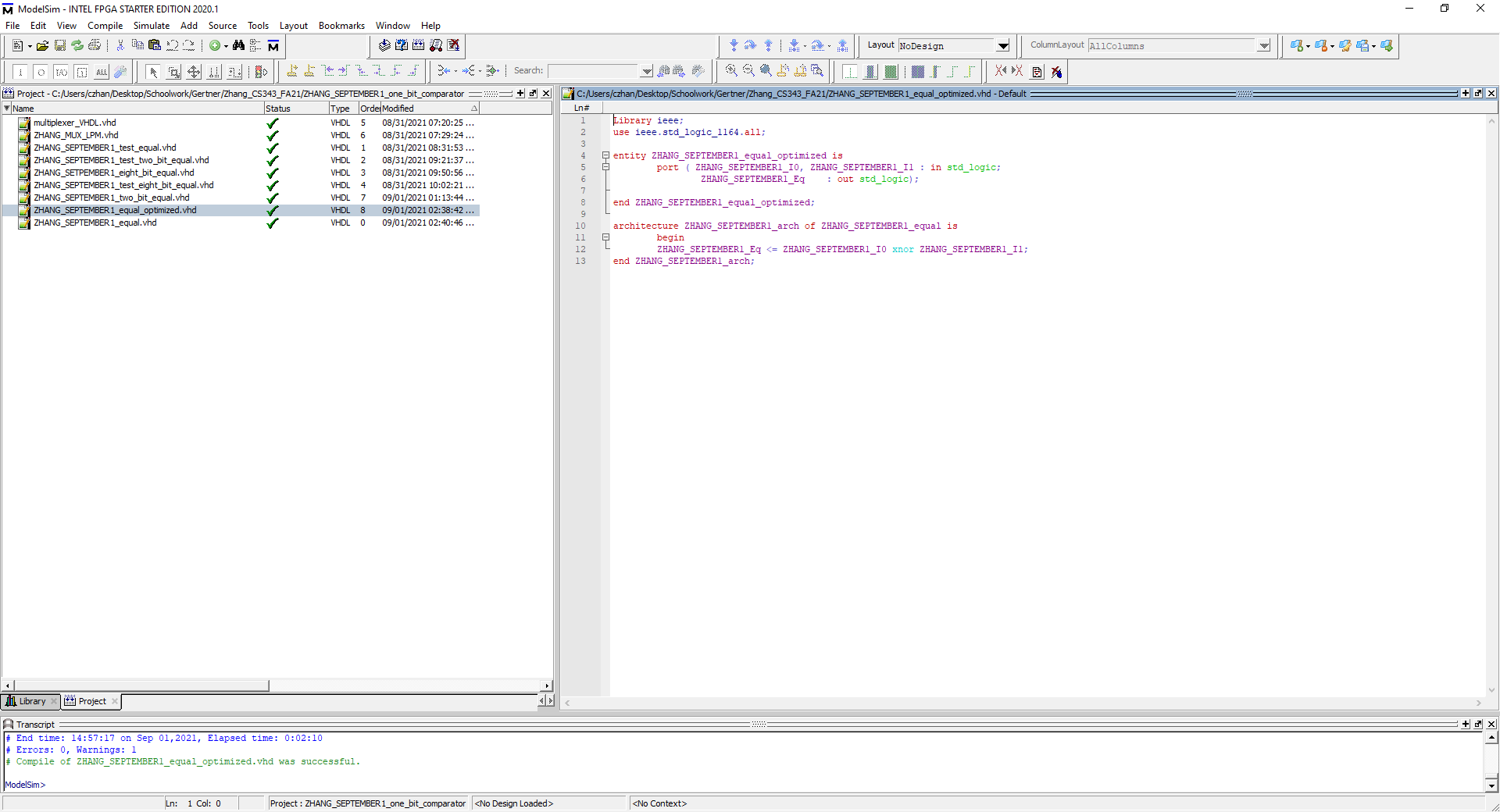
I add waves that are required to be simulated



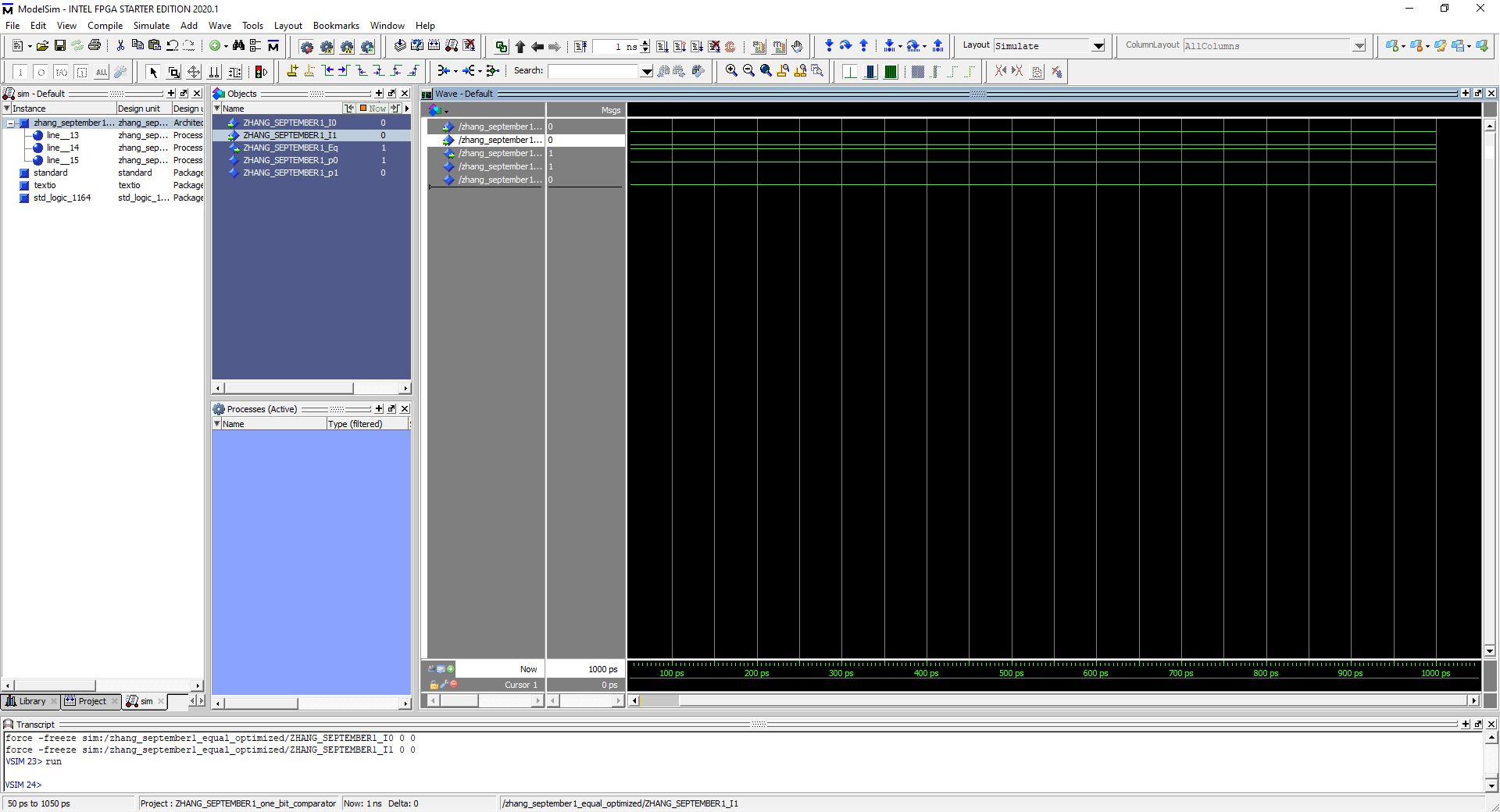
I press run all



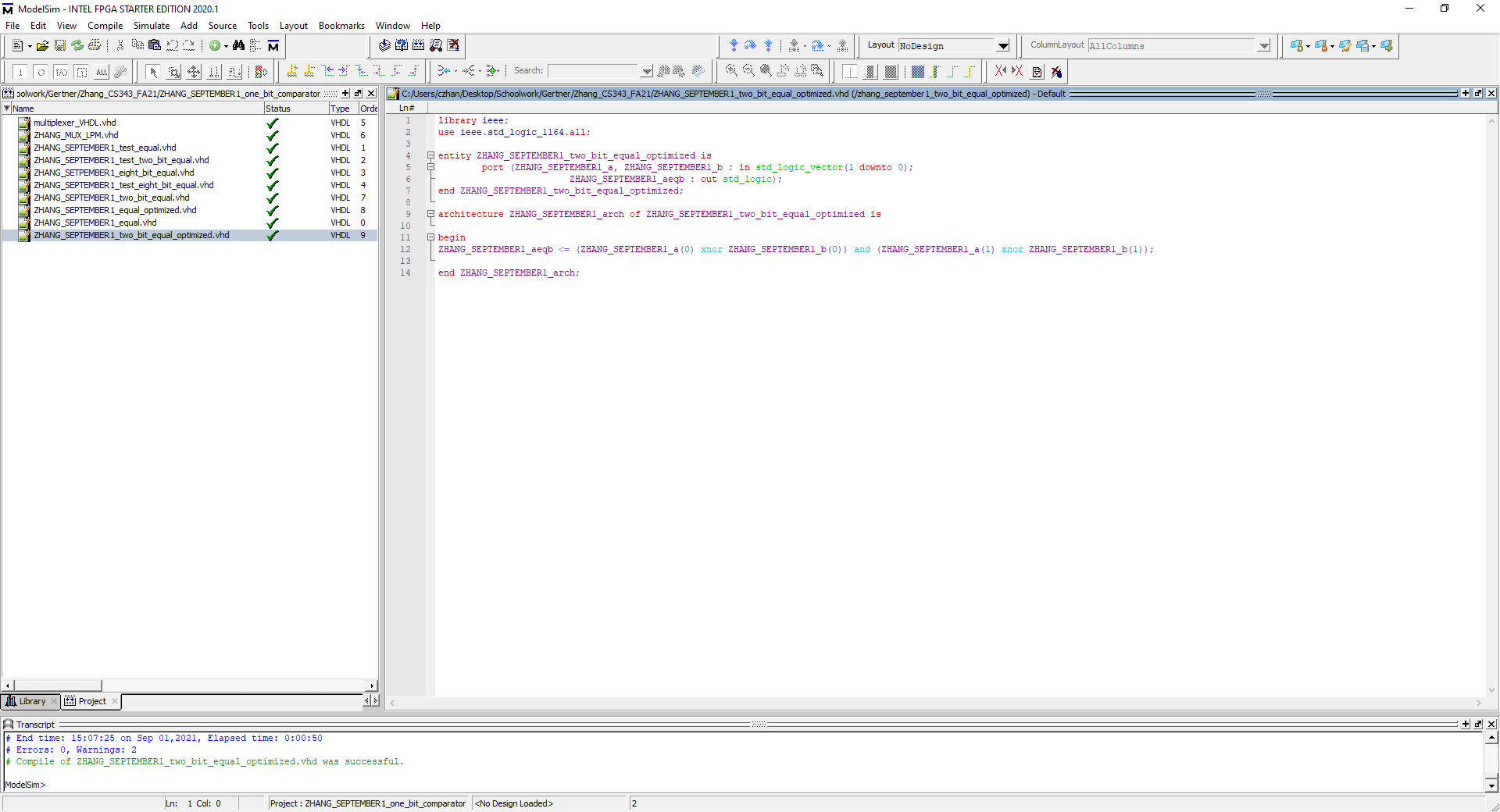
I press run all in modelsim and this page is prompted. There is an arrow at line 62 pointing that there were no errors with the simulation therefore correctness for the 8 bit comparator is verified



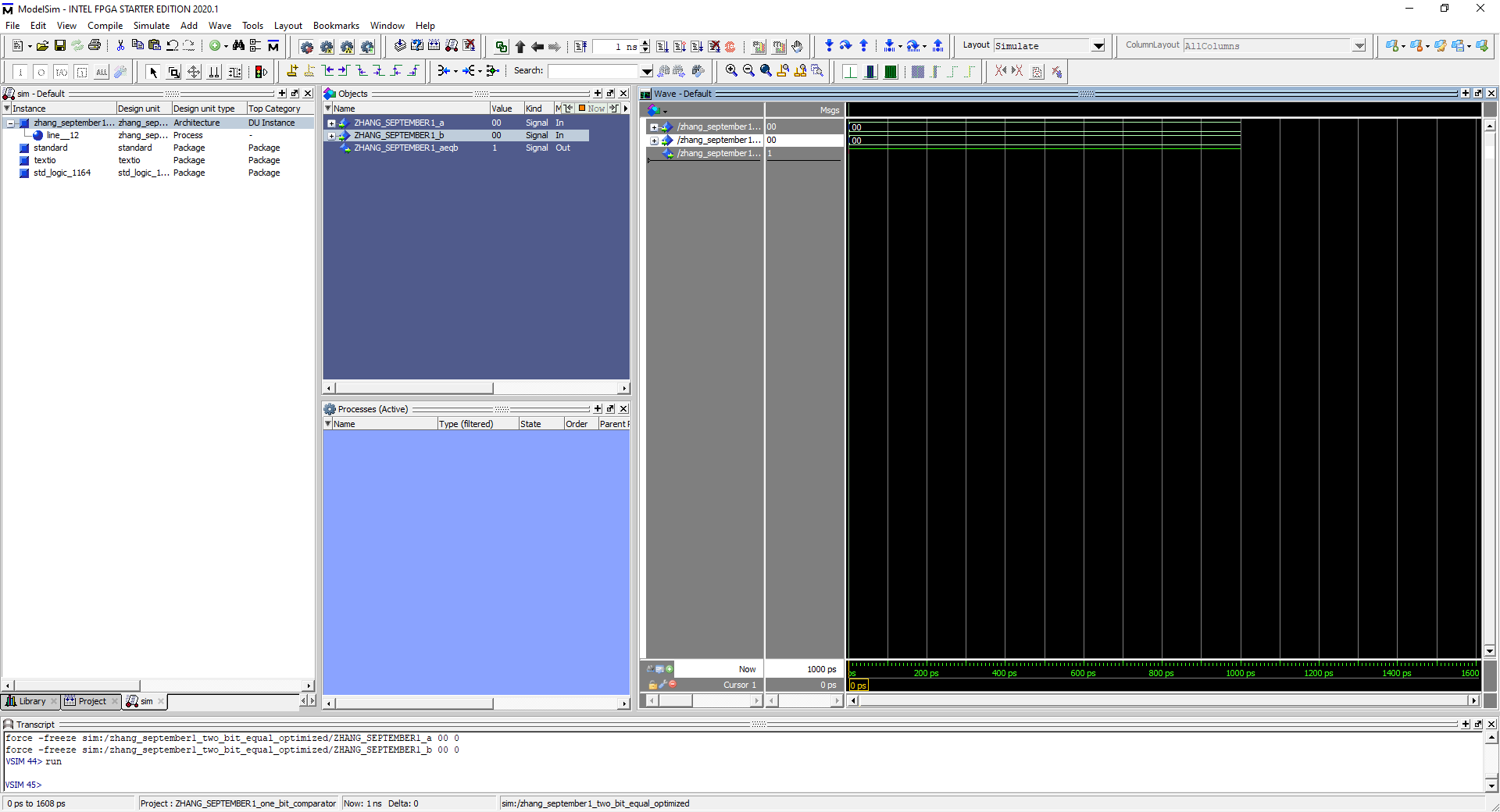
This is the 1 bit comparator that is optimized using XNOR. At the bottom of the image is also a successful compilation OF the optimized 1 bit comparator



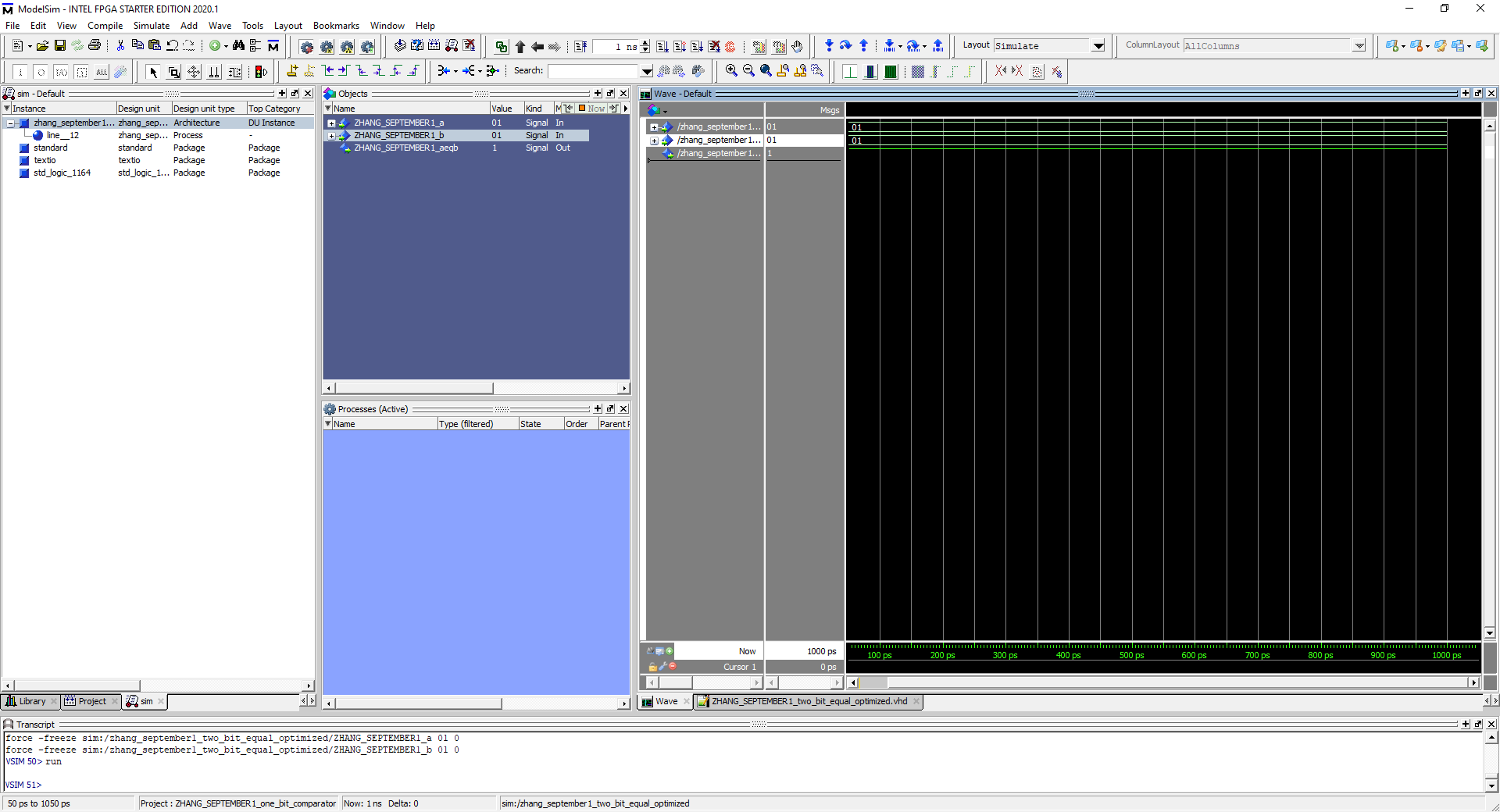
To test for correctness, in modelsim I used inputs 0 and 0 for inputs “a” and “b” respectively and received the output 1 which is correct according to xnor’s truth table.



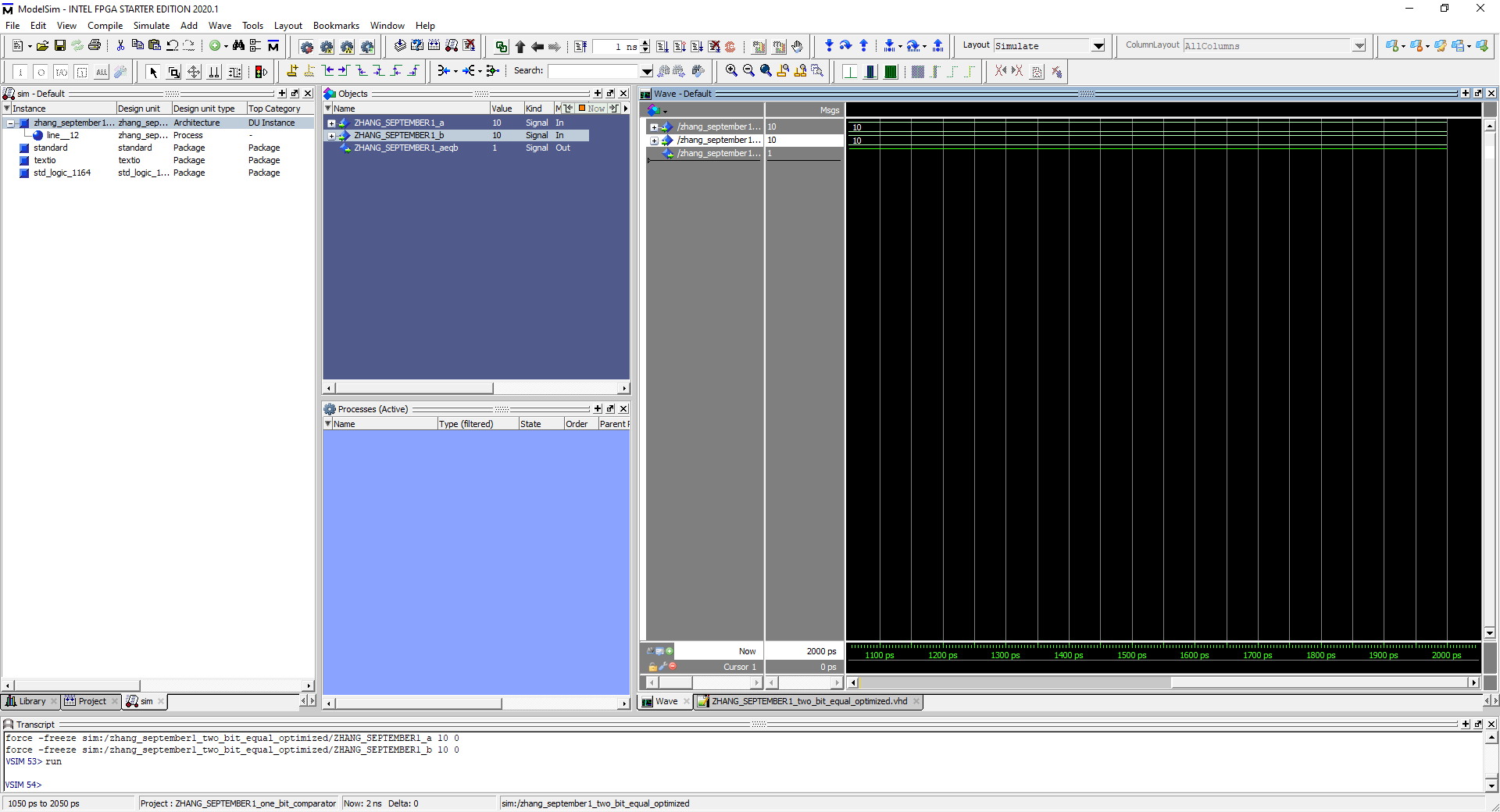
In this image, what is shown is the optimized code for a 2-bit comparator as well as a successful compilation to ensure that the code we will be simulating is correct



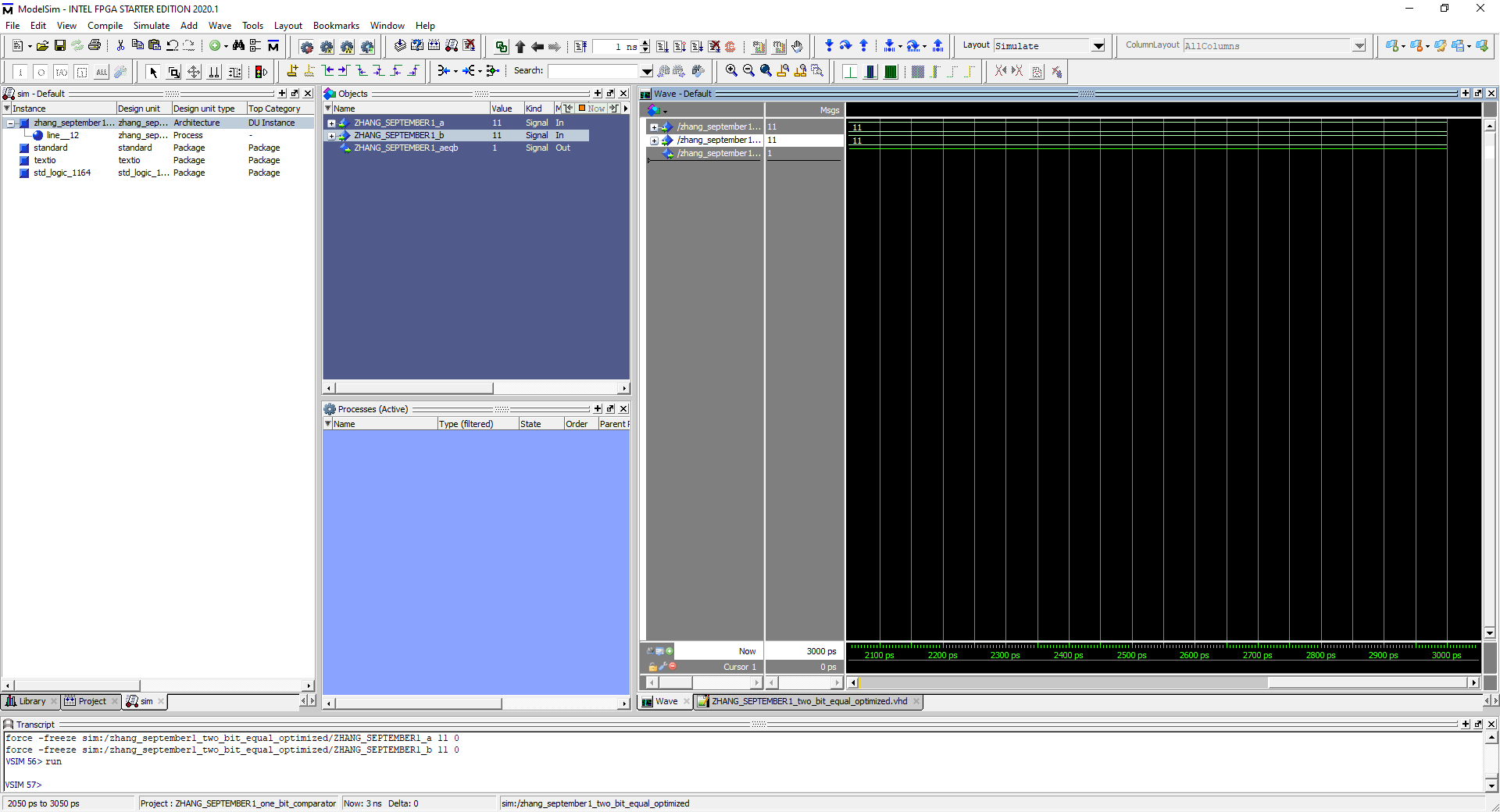
In the image above is the modelsim simulation of the optimized 2 bit comparator running with the inputs “0000” with the output of 1 which is correct.



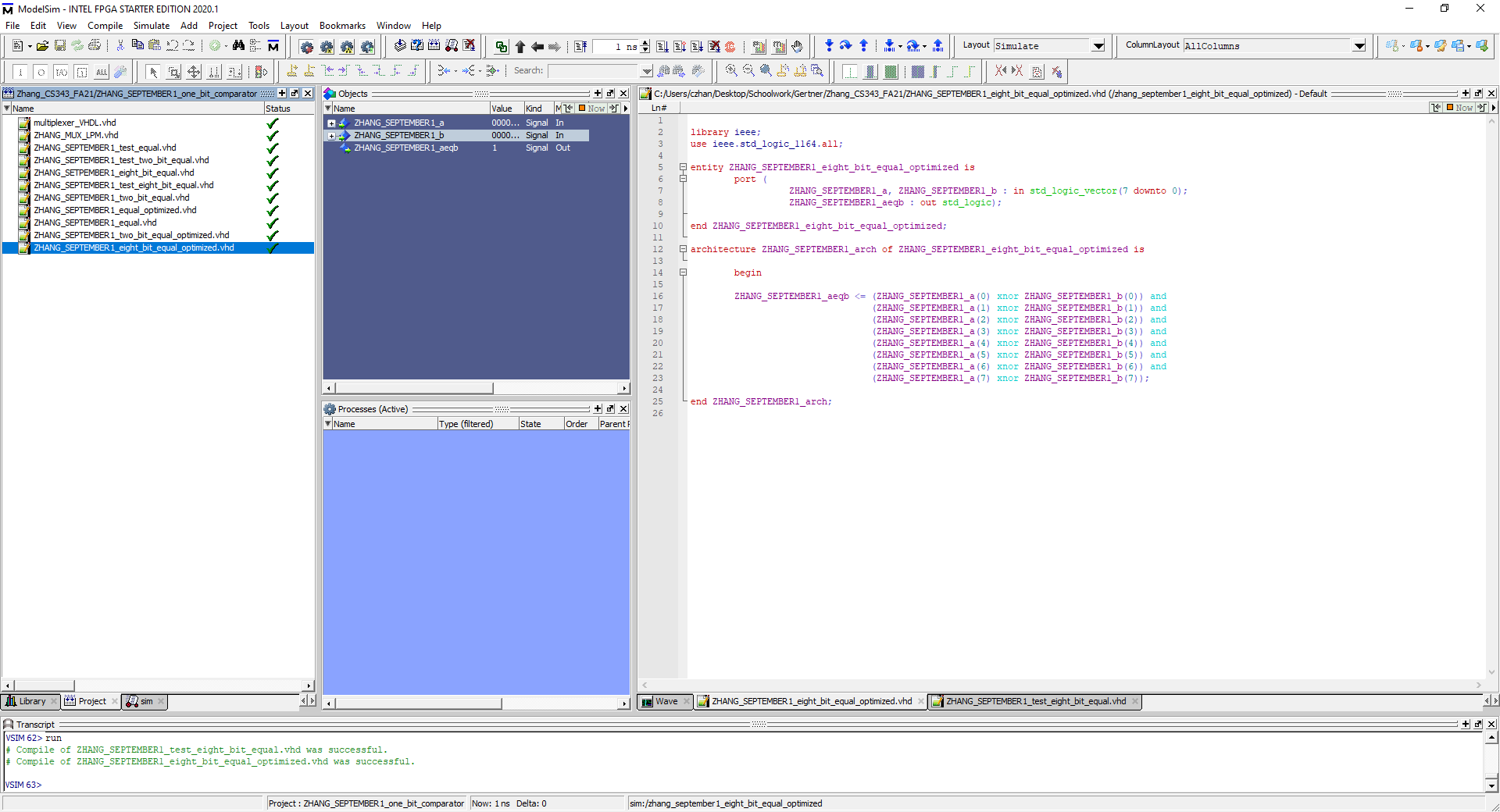
Input with 0101 returning 1



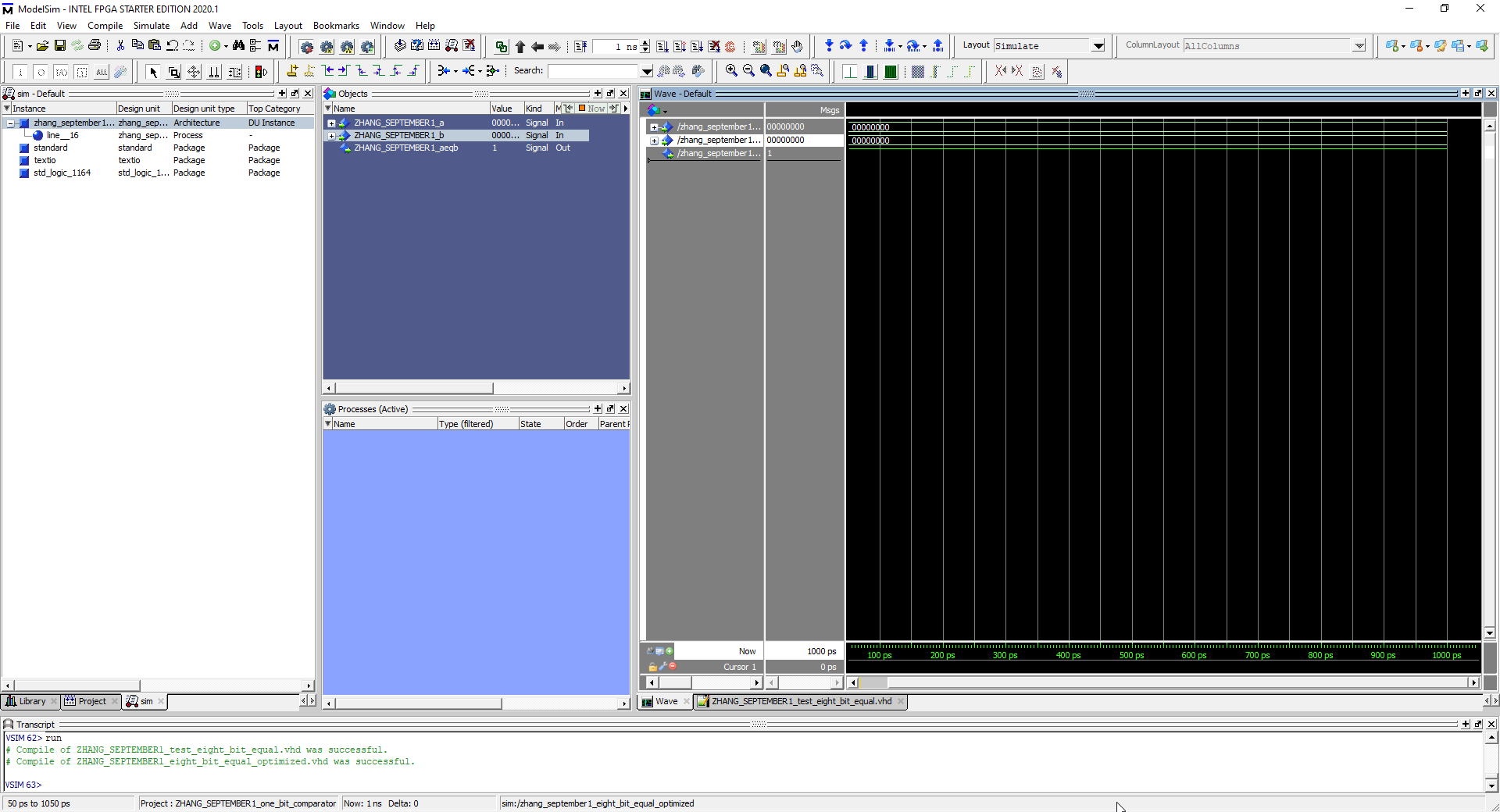
Input with 1010 returning 1



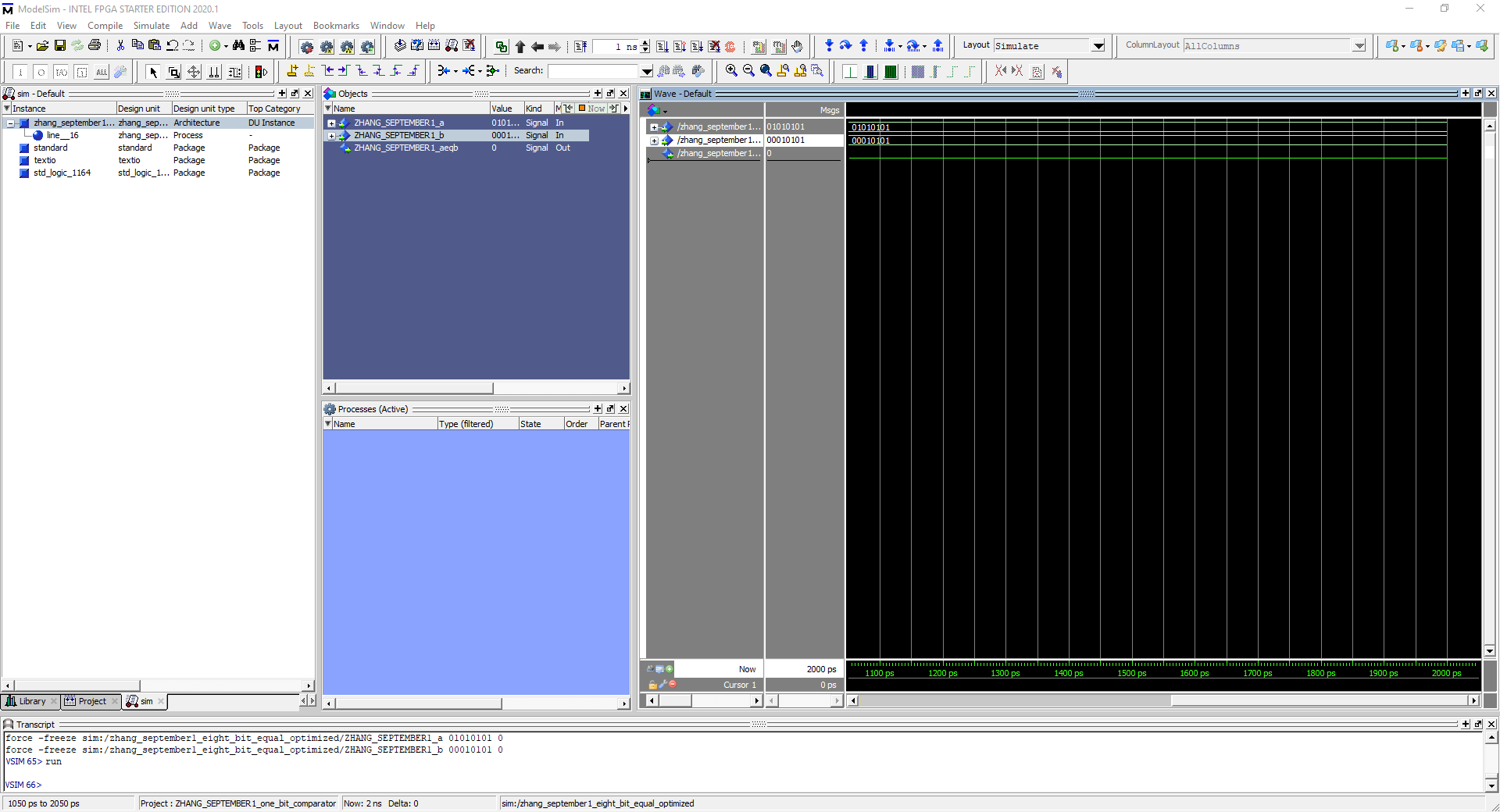
Input with 1111 returning 1



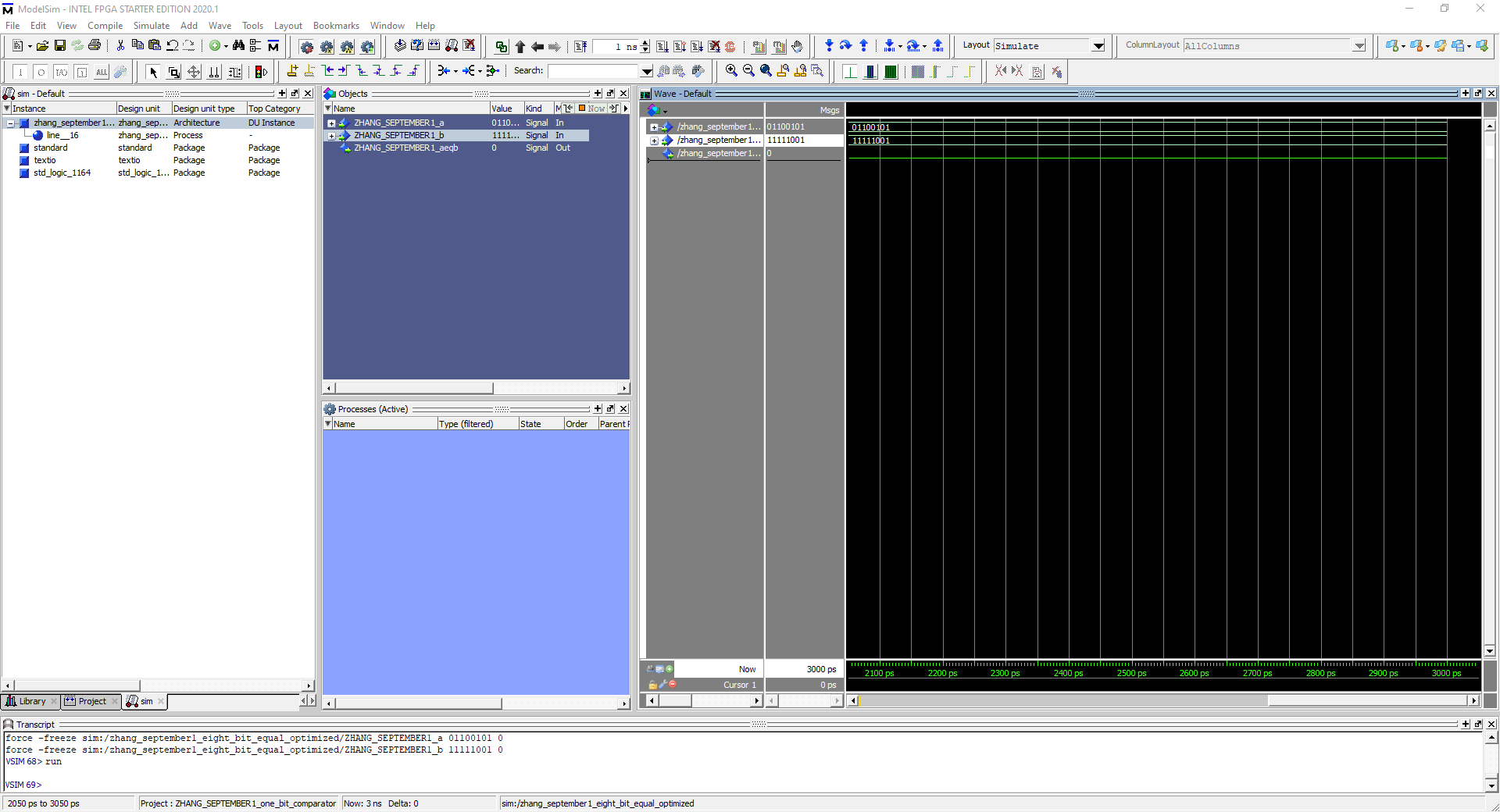
In the image above, it shows the 8 bit comparator with optimized code and successful compilation to verify correctness in the code



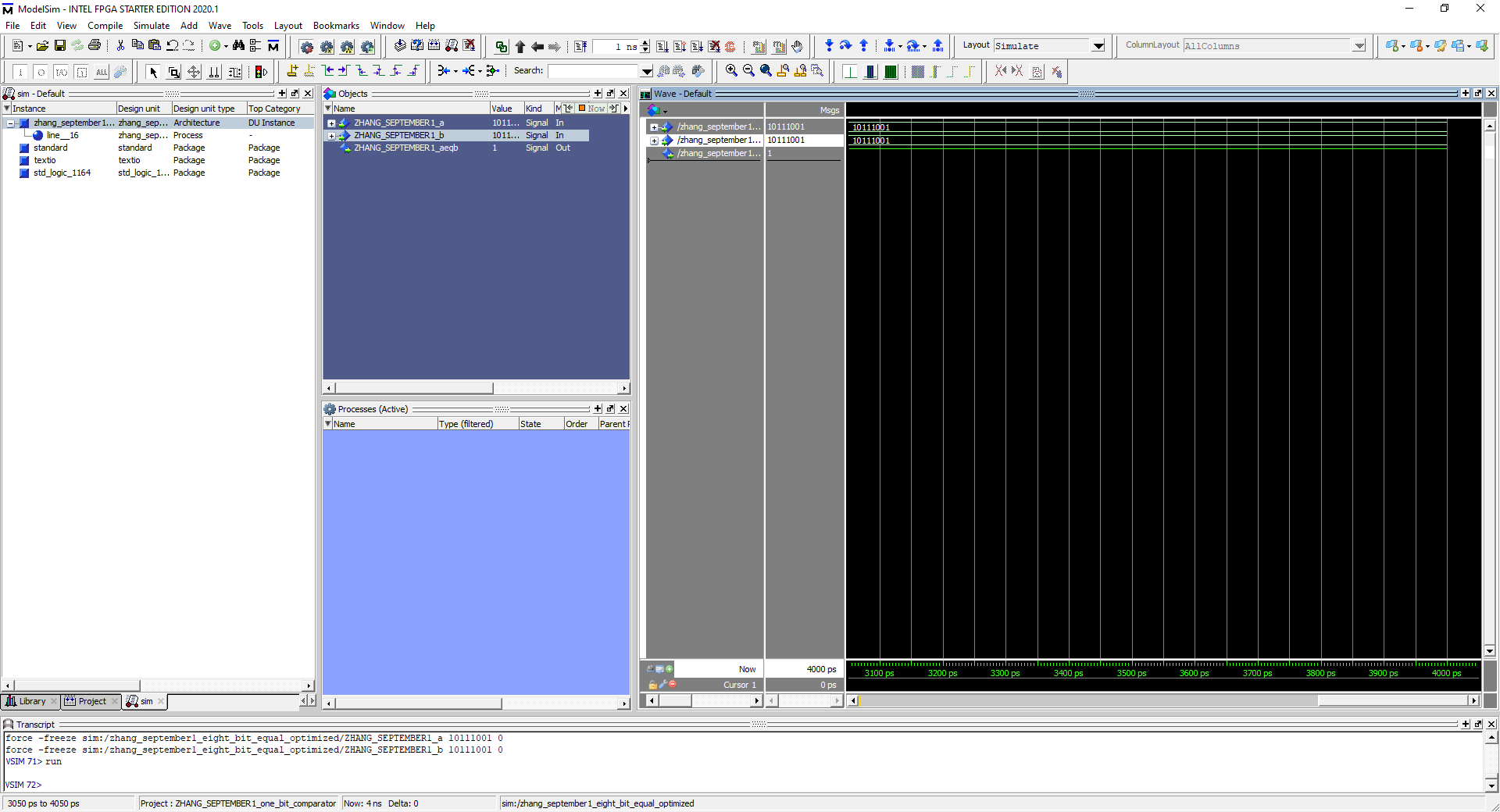
Input of 00000000, 00000000 for an output of 1



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



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



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