Christopher Lall

Lab #2 - Introduction to VHDL, ModelSim and Quartus using Comparators

CSc 342/343 – Professor Gertner

Due February 23, 2022

Table of Contents

[Objective 3](#_Toc96879143)

[Screenshots 3](#_Toc96879144)

[1 Bit: 3](#_Toc96879145)

[2 Bit 12](#_Toc96879146)

[8 Bit: 17](#_Toc96879147)

[Explanation & analysis 18](#_Toc96879148)

[Conclusion 18](#_Toc96879149)

# Objective

This goal of this lab is to further understand Quartus, ModelSim, by using comparators. In this lab we will design 1-bit, 2-bit and 8-bit comparators.

# Screenshots

## 1 Bit:

Graphical user interface, text, application

Description automatically generated

Figure 1. Compilation Report for 1 bit Quartus

Graphical user interface, text, application, email

Description automatically generated

Figure 2. Create modelsim proj 1 bit

Graphical user interface, application

Description automatically generated

Figure 3. Adding File to Project

Graphical user interface, text, application, email

Description automatically generated

Figure 4. Compiling Project in Modelsim 1 bit

Text

Description automatically generated

Figure 5. Successful Compile of 1 bit

Graphical user interface, text, application, email

Description automatically generated

Figure 6. Starting Simulation 1 bit

Graphical user interface, application

Description automatically generated

Figure 7. Dragging object for simulation 1 bit

Graphical user interface, application

Description automatically generated

Figure 8. Force editing selected objects 1 bit

Graphical user interface, application

Description automatically generated

Figure 9. Running Project using run 100 for 1 bit

Graphical user interface

Description automatically generated

Figure 10. Waves for both forces as 0 for 1 bit

Graphical user interface

Description automatically generated

Figure 11. Waves for i0 as 1 force for 1 bit

Graphical user interface

Description automatically generated with medium confidence

Figure 12. Waves for both 1 force for 1 bit

Graphical user interface, application

Description automatically generated

Figure 13. Ending the Simulation

Graphical user interface, text, application

Description automatically generated

Figure 14. 1 bit test equal file in quartus, compilation report

Graphical user interface, application, Word

Description automatically generated

Figure 15. Adding Test Equal to modelsim proj for 1 bit

Graphical user interface, text, application

Description automatically generated

Figure 16. Compile test equal in modelsim

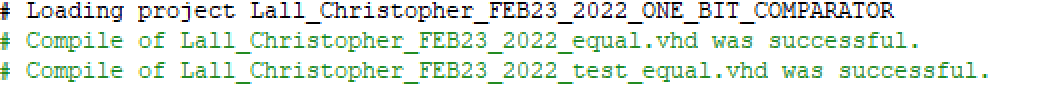


Figure 17. Successful compilation for 1 bit test equal

Graphical user interface, text, application

Description automatically generated

Figure 18. Starting Simulation for test equal 1 bit in modeslim

A screenshot of a computer

Description automatically generated

Figure 19. Waves for test equal in modelsim simulation

Graphical user interface, text

Description automatically generated

Figure 20. File Popup



Figure 21. No errors detected 1 bit test equal, 1 bit test

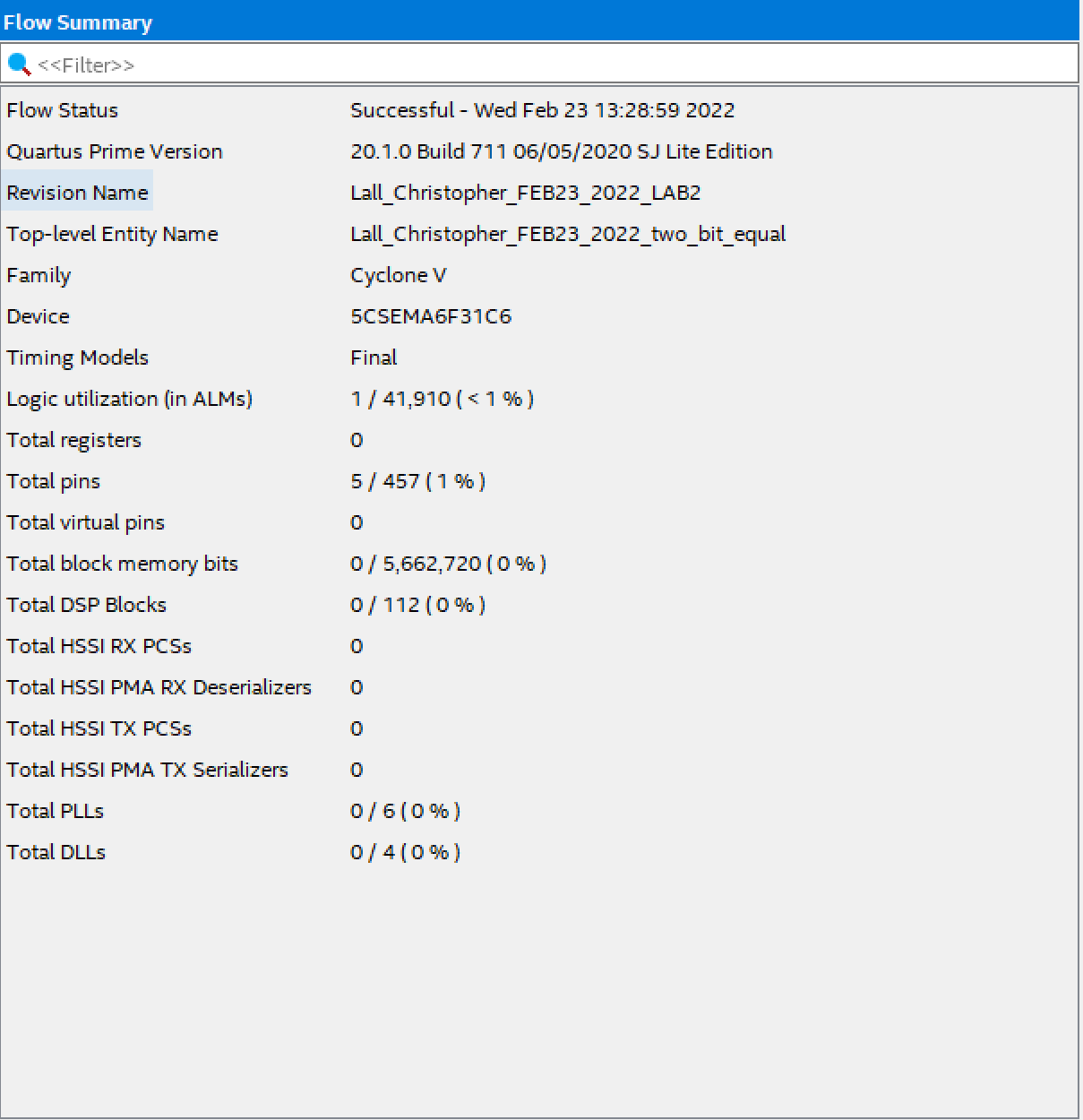
2 Bit:

Figure 22. Compilation Report 2 bit

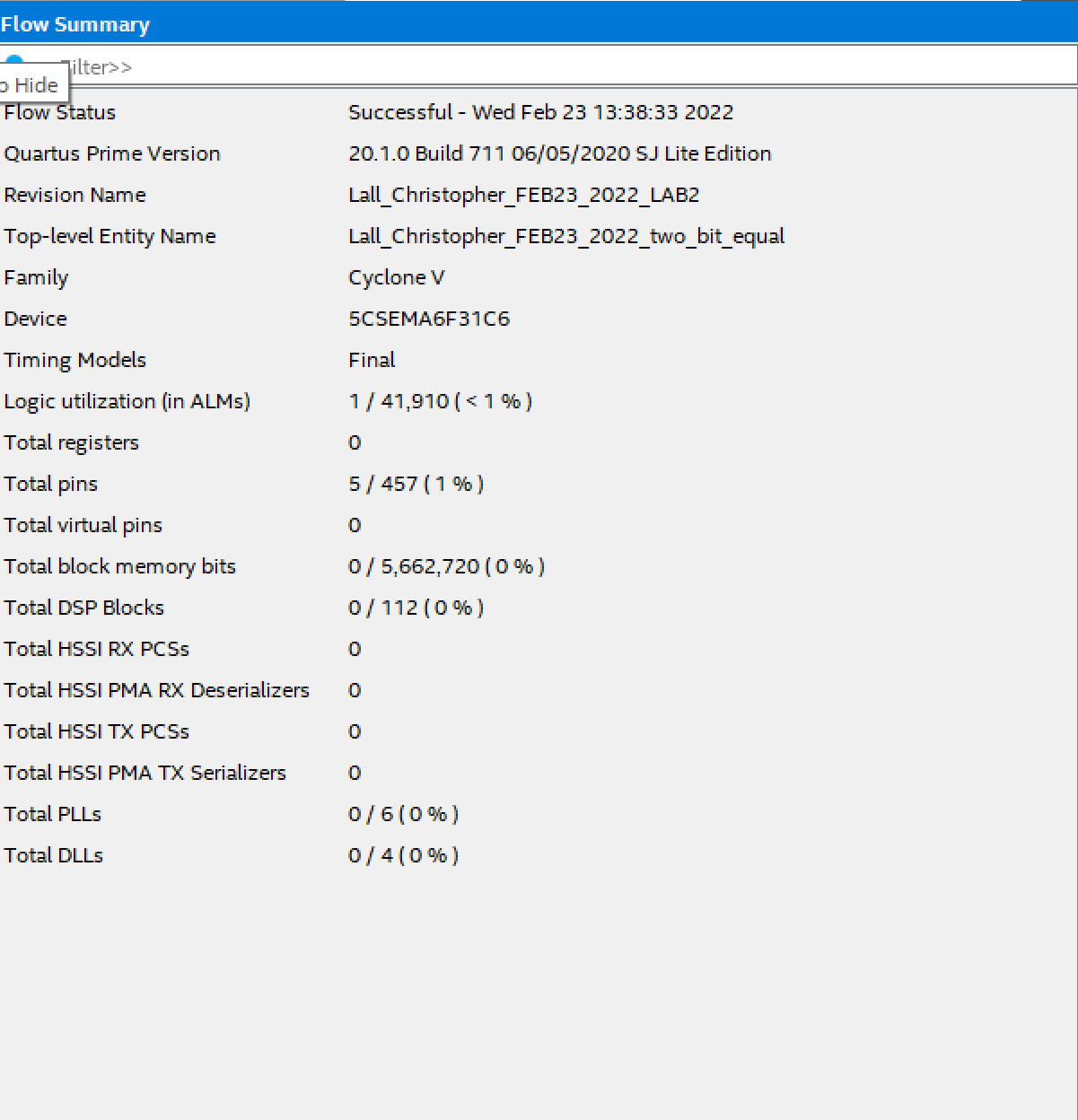


Figure 23. Compilation Report 2 bit test

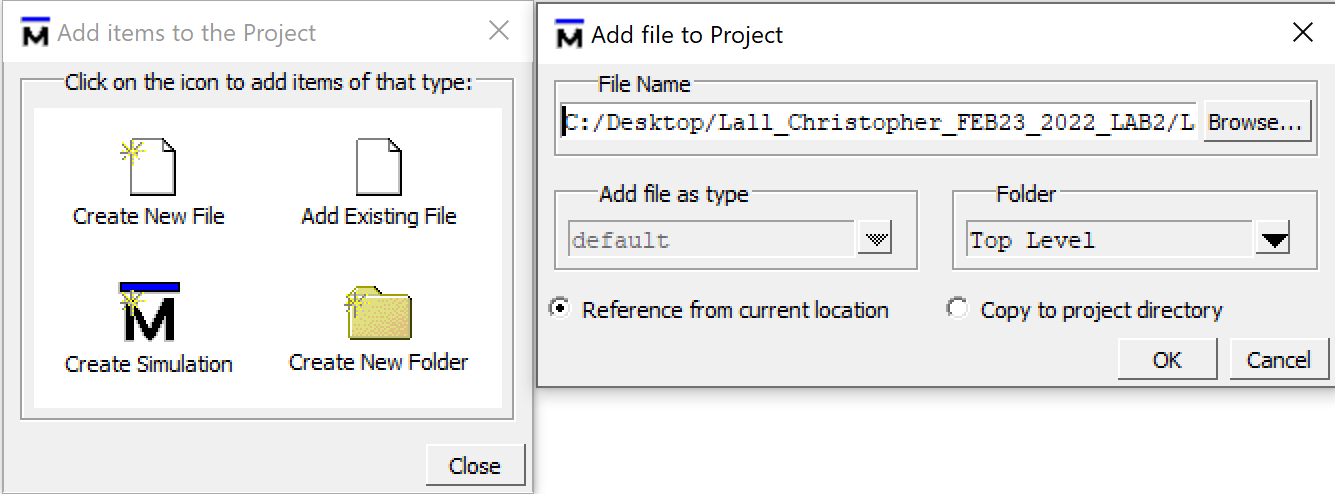


Figure 24. Adding Files to Project 2 bit

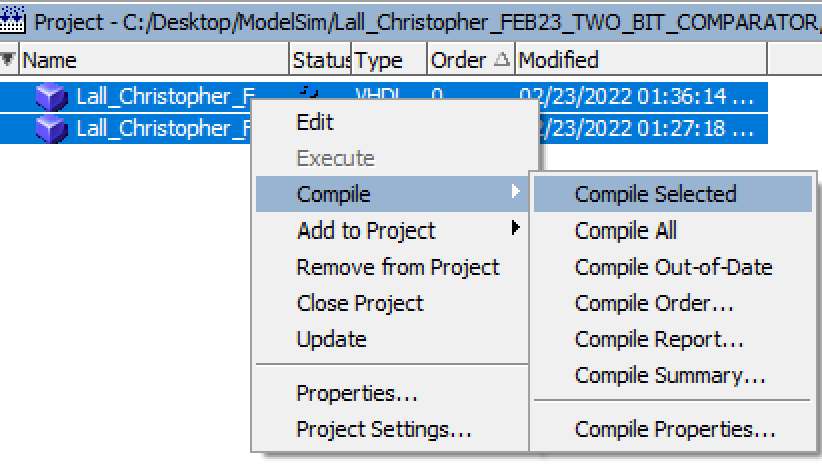


Figure 25. Compiling 2 bit files



Figure 26. Compilation

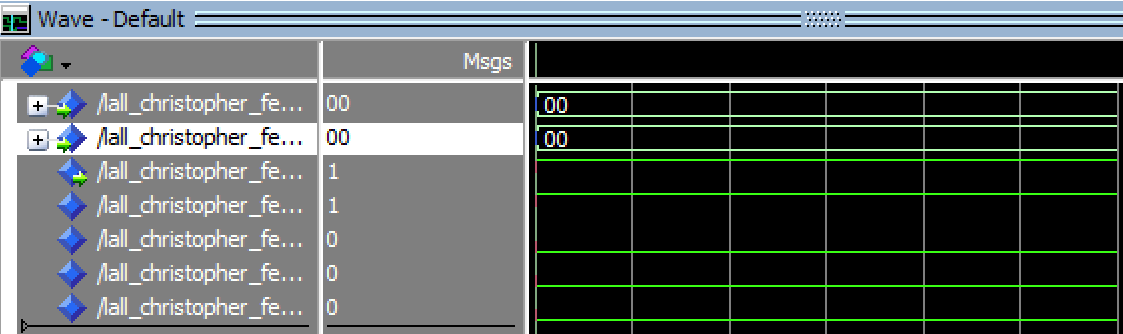


Figure 27. All 0's

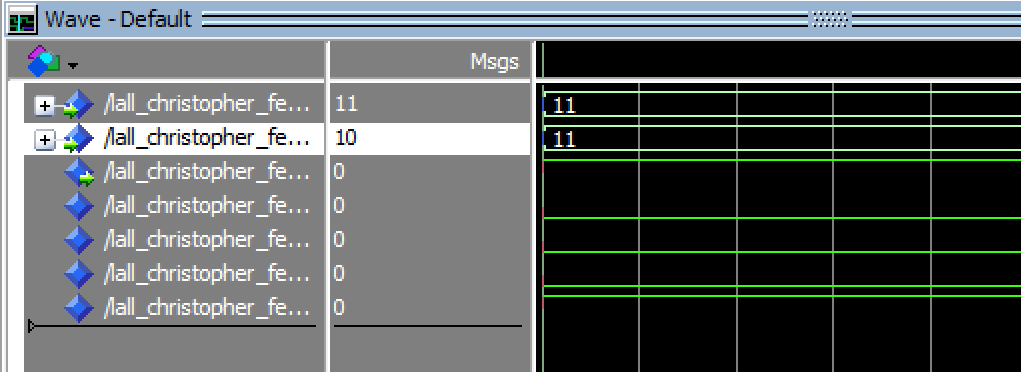


Figure 28. All 1's

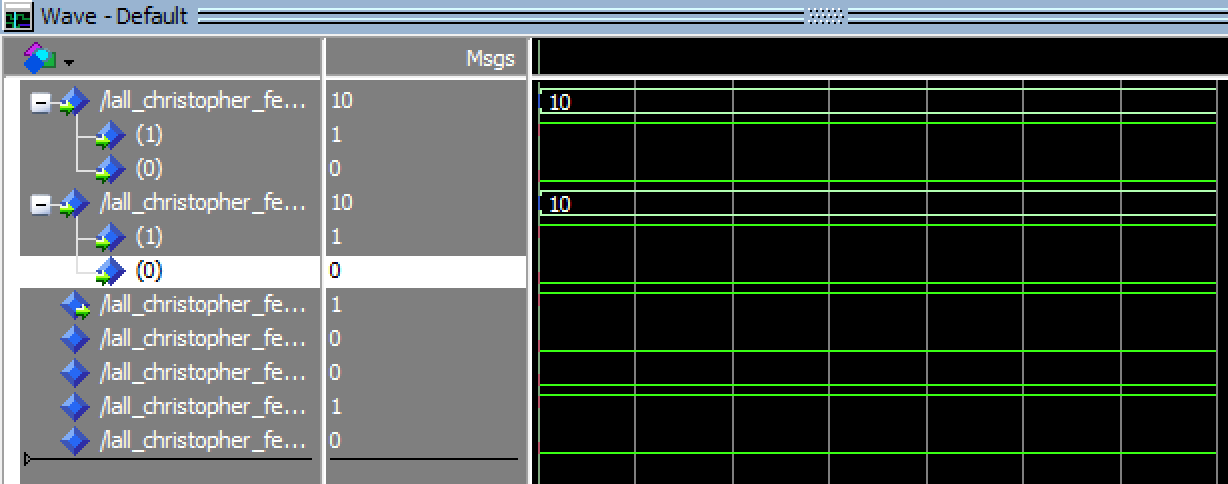


Figure 29. 10 10 2 bit

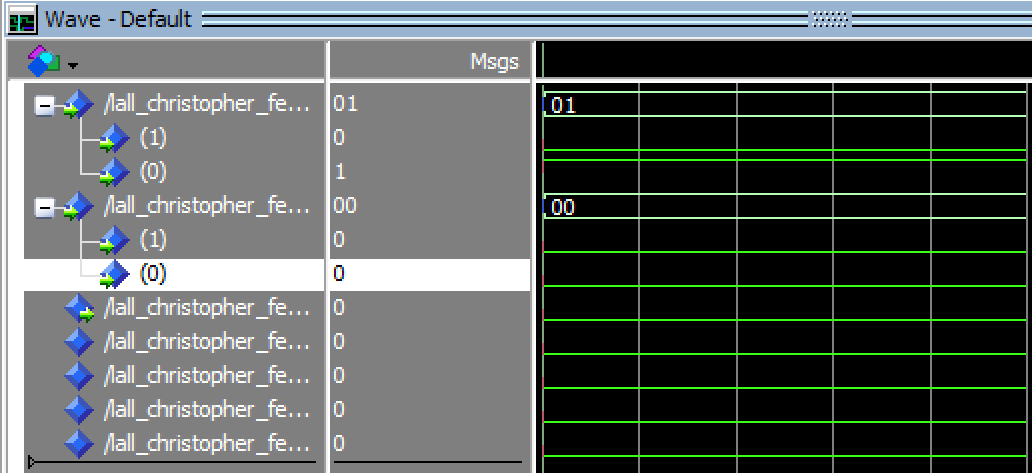


Figure 30. 01 00 1 bit

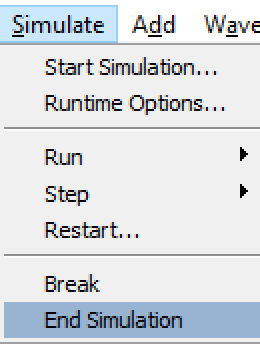


Figure 31. End 2 bit Sim

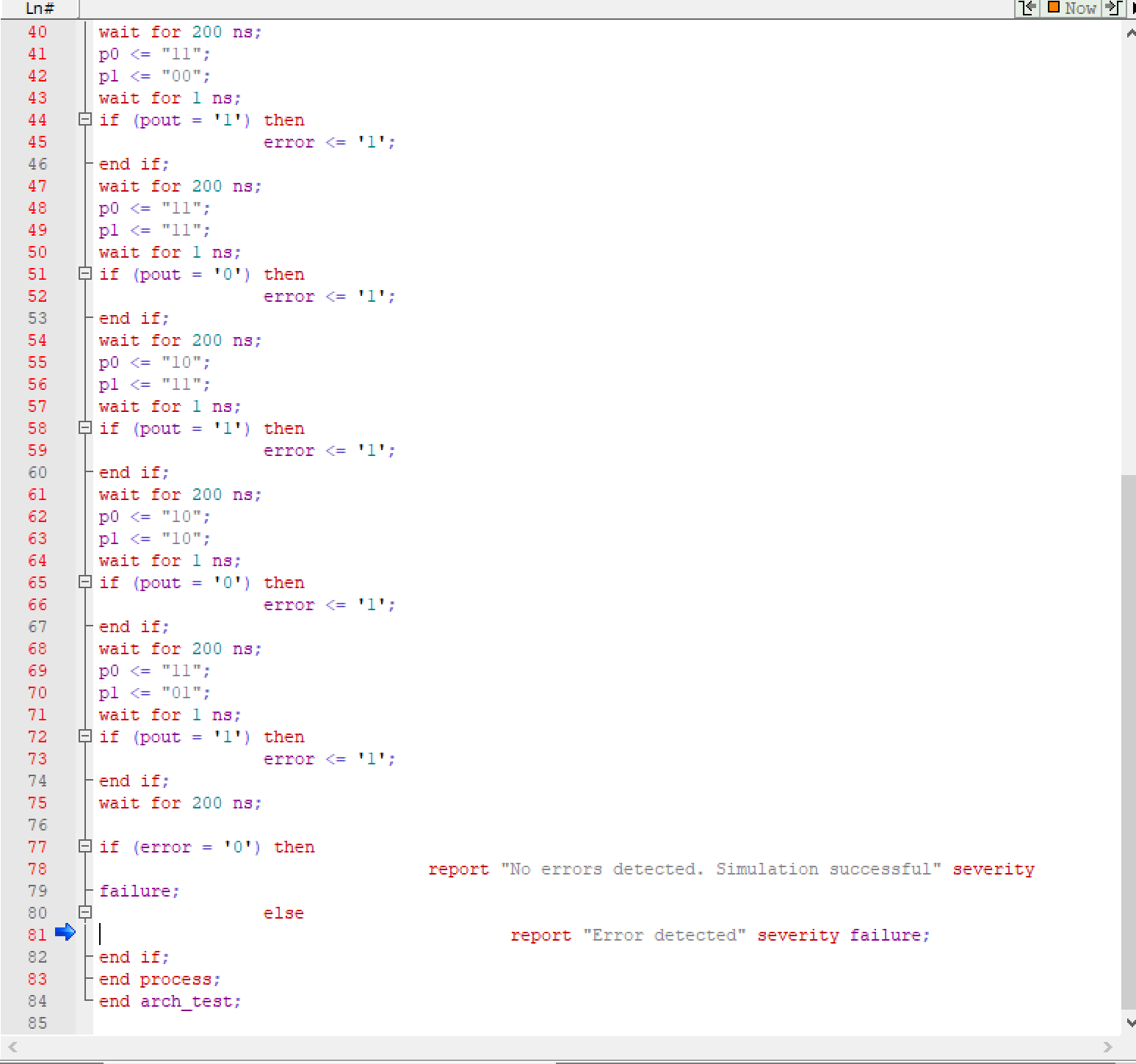


Figure 32. File Error

## 8 Bit:

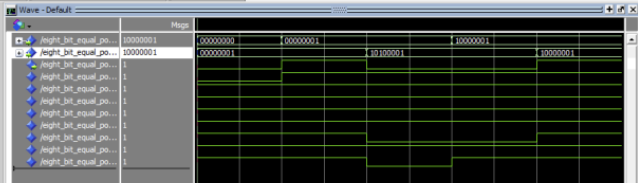


Figure 33. 8 bit waves

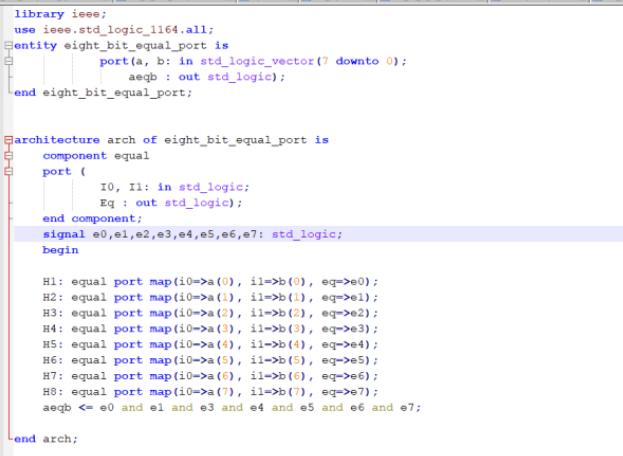


Figure 34.8 bit file

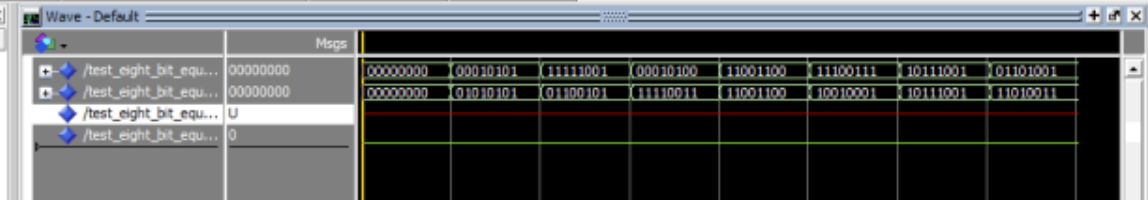


Figure 35. 8 bit test waves, unsure why redline



Figure 36. 8 bit test file

# Explanation & analysis

In the above screenshots, you can see that I went through each 1 bit, 2 bit, and 8 bit comparator vhdl files. I then compiled each to make sure that there were no errors. After doing this, I then imported the files to modelsim to run simulations on them. The results above tell us that we were able to successfully get the right results for the most part. I was also introduced to the comparators and these created comparator files allowed us to visualize the waveforms in which we compared to the truth table. We then verify our results and conclude that it matches with the truth table.

# Conclusion

In conclusion, I was able to created the 1 bit, 2 bit, and 8 bit comparator vhdl files by following the tutorial. Once it was simulated in modelsim, I was able to verify that the waves matched the truth table. With this being said I learned how to more familiarize myself with project creation and simulations.