Duran, Eudis

eudisd@gmail[.com](mailto:myemail@something.com)

CSC 343

TA: Dave Chisholm

**Lab 1 – Comparators, Adders, Multiplexers**

September 6th, 2011

**TABLE OF CONTENTS**

[**Objective**](#_Toc302063794)[**3**](#_Toc302063794)

[**Circuit 1 – One Bit Comparator**](#_Toc302063795)[**3**](#_Toc302063795)

[**Circuit 2 – Two Bit Comparator**](#_Toc302063796)[**5**](#_Toc302063796)

[**Circuit 3 – Eight Bit Comparator**](#_Toc302063797)[**7**](#_Toc302063797)

[**Circuit 4 – Two Bit Adder**](#_Toc302063798)[**8**](#_Toc302063798)

[**Circuit 5 – Four Bit Adder**](#_Toc302063799)[**9**](#_Toc302063799)

[**Circuit 6 – Four To One Multiplexer**](#_Toc302063800)[**10**](#_Toc302063800)

[**Circuit 7 – Multiplexer and**](#_Toc302063801)[**Adder**](#_Toc302063801)[**11**](#_Toc302063801)

[**Demo on the Board**](#_Toc302063802)[**13**](#_Toc302063802)

[**Conclusion**](#_Toc302063803)[**13**](#_Toc302063803)

[**Appendix 1 –**](#_Toc302063804)[**test\_eight\_bit\_comparator.vhd**](#_Toc302063804)[**14**](#_Toc302063804)

[**Appendix 2 – four\_bit\_adder.vhd**](#_Toc302063805)[**15**](#_Toc302063805)

**Objective**

The goal of this lab is to familiarize myself with the VHDL language and the ModelSIM development environment. I will implement comparator circuits, adders, multiplexers and a more complex circuit that combines these elements. Finally, all functionality will be tested via test bench files and manual testing in ModelSIM, and I will interpret the signal graphs and text output produced by these simulations to verify my results.

**Circuit 1 – One Bit Comparator**

**Specification:**

Inputs

**InputA:** a single bit that is compared by the circuit  
**InputB**: a single bit that is compared by the circuit

Outputs:   
**AEqualsB**: a single bit indicating whether or not the inputs are equal

**Functionality:**

This circuit compares the values supplied in the two inputs (InputA and InputB) and outputs a true value (indicated by ‘1’ in AEqualsB if these inputs are exactly equal, or a false value otherwise.

**Design:**

A truth table and design diagram were provided in the handout and are included below for reference. The functionality was implemented in the file “one\_bit\_comparator.vhd”

|  |  |  |
| --- | --- | --- |
| **Inputs** | | **Outputs** |
| ***InputA*** | ***InputB*** | ***AEqualsB*** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



**Test:**

This circuit was tested with the testing script “test\_one\_bit\_comparator.vhd”. This script supplied all four possible inputs to the circuit and verified that the resulting output correctly indicated whether the inputs were equal. If any of the outputs are incorrect, the script halts and reports a failure message.

**Simulation:**

The timing diagram below shows the results of the test file. As expected, the console output displayed “Failure: NONE. TEST SUCCESSFUL! End of simulation.” Which indicated that all tests were successful.



**Circuit 2 – Two Bit Comparator**

**Specification:**

Inputs

**InputA:** a two bit vector that is compared by the circuit  
**InputB**: a two bit vector that is compared by the circuit

Outputs:   
**AEqualsB**: a single bit indicating whether or not the inputs are equal

**Functionality:**

Like the one bit comparator, this circuit compares the values supplied in the two inputs (InputA and InputB) and outputs a true value (indicated by ‘1’ in AEqualsB if these inputs are exactly equal, or a false value otherwise. The only difference is that each input is two bits wide.

**Design:**

I created the following truth table to describe the behavior of the two bit comparator.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | | | | **Output** |
| **a(0)** | **a(1)** | **b(0)** | **b(1)** | **aeqb** |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

**Test:**

This circuit was tested with the testing script “test\_two\_bit\_comparator.vhd”. This script supplied all possible permutations for 2 2-bit inputs and verified that the resulting output correctly identified whether the two 2-bit inputs where equal. If there are any incorrect outputs a error message of "TEST FAILED! Simulation halted" would be reported.

**Simulation:**

The timing simulations is displayed below. Like the timing simulation for the previous simulation, the console diplayed "Failure: NONE. TEST SUCCESSFUL! End of simulation." Which indicated that all tests were successfull.



**Circuit 3 – Eight Bit Comparator**

**Specification:**

Inputs

**InputA:** an 8 bit vector that is compared by the circuit  
**InputB**: an 8 bit vector that is compared by the circuit

Outputs:   
**AEqualsB**: a single bit indicating whether or not the inputs are equal

**Functionality:**

Like the 2 bit comparator, this circuit compares the values supplied in InputA and InputB. If they are equal, then the AEqualsB would output '1', and '0' otherwise. The difference here is that the inputs are std\_logic\_vector(s) of length 8.

**Design:**

Because we are comparing 8 bit inputs, it would be superfluous to include a truth table here, which would need the display 65,536 comparisons. This is because each 8 bit input has 2^8 = 256 permutations. Therefore, it would require 256 \* 256 comparisions.

**Test:**

FILL IN YOUR MATERIAL HERE!

**Simulation:**

FILL IN YOUR MATERIAL HERE!

**Circuit 4 – Two Bit Adder**

**Specification:**

Inputs

**InputA:** a two bit vector that is to be added by the circuit  
**InputB**: a two bit vector that is to be added by the circuit

Outputs:   
**ASumB**: a two bit vector sum.

**Functionality:**

Like the one bit comparator, this circuit compares the values supplied in the two inputs (InputA and InputB) and outputs a true value (indicated by ‘1’ in AEqualsB if these inputs are exactly equal, or a false value otherwise. The only difference is that each input is two bits wide.

**Design:**

**Test:**

FILL IN YOUR MATERIAL HERE!

**Simulation:**

FILL IN YOUR MATERIAL HERE!

**Circuit 5 – Four Bit Adder**

**Specification:**

FILL IN YOUR MATERIAL HERE!

**Functionality:**

FILL IN YOUR MATERIAL HERE!

**Design:**

FILL IN YOUR MATERIAL HERE!

**Test:**

FILL IN YOUR MATERIAL HERE!

**Simulation:**

FILL IN YOUR MATERIAL HERE!

**Circuit 6 – Four To One Multiplexer**

**Specification:**

FILL IN YOUR MATERIAL HERE!

**Functionality:**

FILL IN YOUR MATERIAL HERE!

**Design:**

FILL IN YOUR MATERIAL HERE!

**Test:**

FILL IN YOUR MATERIAL HERE!

**Simulation:**

FILL IN YOUR MATERIAL HERE!

**Circuit 7 – Multiplexer and Adder**

**Specification:**

FILL IN YOUR MATERIAL HERE!

**Functionality:**

FILL IN YOUR MATERIAL HERE!

**Design:**

FILL IN YOUR MATERIAL HERE!

**Test:**

FILL IN YOUR MATERIAL HERE!

**Simulation:**

FILL IN YOUR MATERIAL HERE!

**Demo on the Board**

Not required for this lab.

**Conclusion**

PUT YOUR CONCLUSION HERE!

**Appendix 1 – test\_eight\_bit\_comparator.vhd**

PUT YOUR CODE HERE!

Note that you only need to include files that you either added to, changed, or created from scratch. If a file was fully provided in the lab handout, you don’t need to copy paste it into your report

For this lab, you should be including

test\_eight\_bit\_comparator.vhd (complete the starter code)

four\_bit\_adder.vhd (complete the starter code)

multiplexer\_4\_to\_1.vhd (complete the starter code)

multiplexer\_and\_adder.vhd (complete the starter code)

test\_multiplexer\_and\_adder.vhd (new file from scratch)

These files were complete in the lab assignment so you don’t need to included them in the appendix

one\_bit\_comparator.vhd

test\_one\_bit\_comparator.vhd

full\_adder.vhd

two\_bit\_adder.vhd

test\_four\_bit\_adder

test\_multiplexer\_4\_to\_1.vhd

multiplexer\_4\_to\_1\_using\_process.vhd

In addition, I am giving you the following files for free to help since it’s a big lab :) Check the website to download them. Since they are being given to you, you don’t need to included them in the appendix

two\_bit\_comparator.vhd

test\_two\_bit\_comparator.vhd

eight\_bit\_comparator.vhd

**Appendix 2 – four\_bit\_adder.vhd**

PUT YOUR CODE HERE!

Likewise, add each file to the appendix under its own heading…