Lab 02 - First Verilog

In this lab, you've learned how to do an initial and simple design in Verilog to learn the Vivado tooling and process involved in RTL/FPGA design.

Rubric

Item	Description	Value
Summary Answers	Your writings about what you learned in this lab.	25%
Question 1	Your answers to the question	25%
Question 2	Your answers to the question	25%
Question 3	Your answers to the question	25%

Lab Summary

Summarize your learnings from the lab here.
From the first lab, there was a lot of information and new UI thrown towards myself which took a while to get my head around. The general idea of the layout is the left hand side will run simulations and build the code itself along with the bitstream (which needs to be recreated after any code changes). The middle panel will hold all relevant files or boards attached. The right side will show either the output of the simulation or the code itself. Finally the bottom half is a standard log/ report section.

The creation of the code is straightforward with only a small amount of new syntax to be learnt. Apart from that it is very similar to java, including issues when you forget to add ';' after each statement.

Lab Questions

1 - Describe the stages of building a Verilog project in Vivado.

First, the code that we want to run/ test is created. This can be later when the files are uploaded to the project. But I did it first.
Then a new project is created from here we simply name the project and specify that it is an RTL project ensuring that "do not specify sources at this time" is unchecked.
Next we add all code files needed and hit next. On this screen we change any test files to "simulation only" and any working files to "synthesis & Simulation"

We hit next again and add any constraint files.
Our penultimate step upon pressing next is to add the board type. In our case it is a “Basys3”. If it does not appear when searching, we press the refresh button and download the board when available.
After hitting the next button one final time we are presented with a summary screen. If all is correct we press finish and we can begin testing our code.

2 - What is the value in looking at the elaborated design schematic?

By looking at the elaborated design schematic we get a visual representation of our code logic. With which we can follow through and find any errors easier than by just reading the code. It also helps us check that our code creates an expected end result.

3 - Why should we simulate our designs frequently? What does the simulation do?

We should simulate our designs frequently due to the fact that it is easier and quicker than running a full bitstream implementation. This means that I we are making small code changes we are not constantly waiting a few minutes each time when the same results can be gained faster by running a simulation.

Code Submission

Upload a .zip of all your code or a public repository on GitHub.