LAB 3: VHDL Single Cycle Calculator

CPEG 324

Jason Reynolds
Vinay Vazir

ABSTRACT

The goal for this project was to develop a simple single cycle processor, combining the skills we learned in the earlier labs into one project.

DIVISION OF LABOR

The lab was broken down into three steps. The first step was developing the datapath, the second path was developing each individual component, and the final step was putting all the components together. Both team members worked on the first step of the project together, so that everyone understood the datapath well for later steps of the project.

The second part of the project was developing each of the subcomponent of the datapath. This step was split between Vinay and Jason, as each component could be made without any knowledge of the other components.

The last part of the project was putting all the components together and developing a testbench. Jason worked on putting all the components developed in step two and connecting them together with signals. Vinay worked on developing a python tool that would generate testbenches for the calculator using the tools created in lab one.

STRATEGY

For the datapath, we initially started off with the register memory, as we decided that the testbench would be acting as the program counter and instruction memory. From the register memory, we decided what ports would be needed for input and output, and continued to the ALU. For the skip logic, we decided to use a 2-bit shift register, as it was easier to implement than other skip logic. We then worked on developing the control signals and muxes for the datapath. After presenting our datapath in class, we decided to go back and change some details of our datapath, such as removing all control signals from the ALU, instead opting for a mux to determine inputs, and keeping all control signals, no matter how small and simple, in a dedicated control block.

For the register memory, we decided that the testbench would provide an opcode to the component, which we would then parse to gather what data would be needed. This data was then piped out to two data lines, *reg1Data* and *reg2Data*.

For the ALU, we initially had a control signal that would determine the operation that it would do, load, add, or subtract. However, we decided that this was not needed, as all operations were just additions, so we instead had a series of muxes at the input to the ALU, which enabled us to remove all control signals from the ALU and instead have three muxes at the inputs.
For the skip logic, we decided to use a shift register were we would load and shift a skip signal every clock signal. This made all the skip instructions very easy to implement, as we would just

set the select signals of a mux to be the skip signal. We went with a three bit shift register, as it would give us the ability to skip the next cycle, and skip the next two cycles, without affecting the current cycle.

RESULTS

We were successfully able to develop a single cycle datapath and implement all the logic needed to control our 8-bit calculator.

CONCLUSION

This project provided a good opportunity to make use of the knowledge and skills gained this semester. The programming of this project required the use of VHDL and an understanding of single cycle datapaths and the components that constitute them. Through the process of designing, laying out, and programming this assignment, these skills were all thoroughly reinforced.

Challenges were encountered after assembling the disparate components into a single entity. Even though each of the components on their own functioned properly and passed all the tests that were designed for them, once assembled into a datapath, there were a few instructions that did not perform as expected. Troubleshooting these issues provided a great opportunity to develop a deeper understanding of how the components interact with one another, especially with regard to time.

NOTES ON EVALUATING

Use the command: *python builder.py bench.jv -s* to build the calculator and run the testbench.

APPENDICES

Appendix I:

Name	Hours spent
Jason Reynolds	20
Vinay Vazir	20

Appendix II:
All code located here:
https://github.com/Syncla/CPEG324/tree/Lab1WorkingV/lab3/final/toSubmit