Temple University

College of Engineering

ECE 4612: Advanced Processor Systems

A red square with white letters

Description automatically generated

Adam Love

Program Counter Design

11/2/2023

**Objective**

This report details the design and implementation of a program counter (PC) component, programmed in Verilog. The PC that is designed will be used as one of the components to create a single-cycle processor.

**Tools/Equipment**

The PC and all test benches were programmed using Verilog and were programmed within VS code. The PC was tested using Vivado to generate output waveforms.

**Procedure**

The PC is a register with very few requirements. It outputs the current instruction address and reads in the next instruction address. It sets the current instruction address to be the next instruction address whenever the positive edge of the clock is detected. The PC should start at address 0x00000000.

**Testing**

To test the PC, a very simple test bench was created. The test bench is essentially how the PC will be implemented in the single-cycle processor. The test bench used the PC module, a 32-bit adder module that was used as part of the design in the 64-bit adder, and a repeating clock signal. In the actual implementation of the single-cycle processor, the PC will generally increase by 4 on each clock cycle. For the test bench, this same property was implemented. The adder reads the current PC value, as well as the value 0x00000004. This means that in the test bench, the PC should increase by 4 on each clock cycle.

**Results/Observations**

The first implementation and test of the PC did not work correctly. The PC would initialize to 0xZZZZZZZZ, and would remain there for the whole test. To correct this behavior, an initial condition had to be added to the PC, where it would initialize to 0x00000000, done with an initial block in the Verilog module. Once this was done, the PC would behave as expected, increasing by 4 on each rising clock edge. The behavior of the PC is shown below.

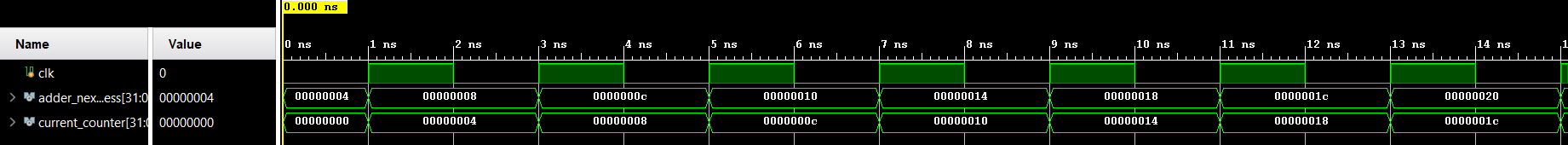


Figure : Program counter behavior increasing by 4 each clock cycle

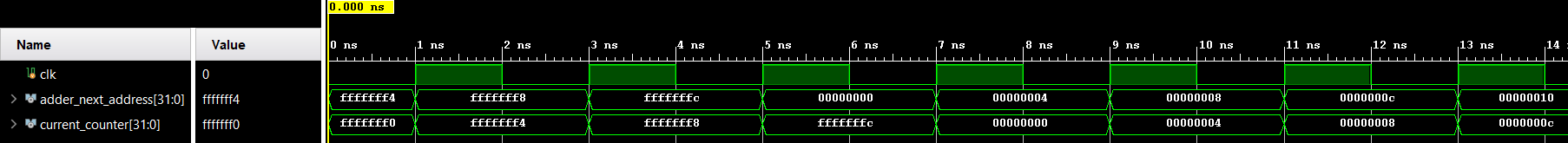


Figure : Program counter behavior upon reaching the maximum value (0xffffffff)

**Conclusion**

Designing the PC module taught me a very important lesson about initializing values in Verilog. I spent a lot of time trying to troubleshoot and attempting to setup the test bench in a way that loaded the initial value of 0x00000000 into the PC. Eventually I realized that defining the initial value in the actual PC module was the correct way to solve this problem. I will need to keep this in mind during the design of the rest of the modules in this project so that the final single-cycle processor is put together more easily.