CMU 18-447: Introduction to Computer Architecture

Lab 1.5 (Warmup): Synthesis Workflow and SystemVerilog ALU units

Not Due

In this lab, you review the basics of SystemVerilog and the synthesis tools you will be using. First you will develop three different modules, an adder, a multiplier, and a shifter. Then you will write a testbench to test these modules, while learning the synthesis workflow that you will use in the labs. This lab is not required, but is reccommended if you are unsure about your systemverilog. We will release solutions and a comprehensive testbench later in the week to allow you to check your solutions.

Task 1: Write the Modules

The first module you will write is an adder. It takes in two 32-bit operands and outputs the sum of the two. Again the add operation should be able to hand the different addition instructions specified in ARM: ADD, SUB, RSB, ADC, SBC, and RSC. Since this is supposed to mimic the Lab 2 adder you will use in your execute stage, try to think about what conditions and other status flags the adder will have to output. Use the interface provided in the file src/arith_units.sv.

The other two modules you will implement are the multiplier and shifter. We will provide the interface, but try to think about what cases you need to cover for each.

Task 2: Write a Testbench

It is essential that you understand what cases to test for and how to debug for systemverilog. By writing a testbench, you will think about the different cases you should cover for edge cases on each operation. Remember to include a *$finish* statement or your code will run forever, or until interuppted. This will be in src/tb.sv.

Task 3: Compile and Run

First run this command:

source /afs/ece/class/ece447/bin/setup_bash

Then use the vcs command. For the purpose of this class, it should be sufficient to compile sv files with the command:

vcs –sverilog <*file1.sv*> <*file2.sv*> …

And then you can run the file with ./simv. Feel free to explore other options that vcs has to offer.

Other options for compilation and running may be added later.

The starter code can be found at: /afs/ece/class/ece447/labs/lab1.5/