EE290C Homework 2

Bitwidth Inference

Designing custom hardware gives you absolute control over what functionality you are implementing. One of the main advantages is that you can control the bitwidths of all your computations. For example, if your computation only requires 12 bits, you can save a lot of power and area by making custom 12-bit hardware rather that using a general-purpose 64-bit processor.

Keeping track of the widths everywhere in your design can be tedious, though, especially when you are writing a generator. Chisel (via the FIRRTL compiler) can infer widths based on a set of conservative rules. Using Chisel's width inference rules, figure out the widths of the registers and outputs in the following snippet of FIRRTL. Show your work.

You can check your work with the FIRRTL compiler.

```
circuit Example :
  module Example :
    input clk: Clock
    input a: UInt<4>
    input b: UInt<2>
    input c: UInt<3>
    output out: UInt

    reg sum: UInt, clk
    reg prod: UInt, clk

    sum <= add(a, b)
    prod <= mul(sum, c)
    out <= sub(prod, sum)
```

Bootcamps

Complete bootcamps 2.3-2.5 and 3.1-3.2.

You will upload your completed bootcamp notebooks to bcourses. Please upload both the notebook file (`.ipynb`) and an HTML version (in the notebook, click File->Download As->HTML). Be sure to evaluate your cells!