Advanced Programming
Homework Assignment 7

Due Wednesday, March 5, at 3PM

Preminimaries

1. Read the QuickCheck paper. (Note that the paper refers to a slightly older version of QuickCheck than the one I described in class — for example, instead of a single `Arbitrary` class with two methods `arbitrary` and `coarbitrary`, the paper describes separate classes `Arbitrary` and `Coarbitrary` with one operation each. But the fundamental ideas and most of the details are identical.)

2. Grab the file `circuits.hs` from the course web site and rename it `YourName(s)7.hs`. Read through the file and make sure you understand what's there. (Here, again, I've made quite a few small changes from the version that I showed in class on Monday — e.g., the type `Signal` is no longer polymorphic, and some of the points where there was confusion on Monday have been made cleaner and clearer — but the ideas have not changed and the code is pretty simple.)

3. Experiment a little with running the QuickCheck tests included in the file. Try introducing mistakes in either the tests or the implementations being tested and see whether QC catches them.

Main assignment

1. Using `prop_bitAdder_Correct` as a model, write a specification for a single-bit subtraction function that takes as inputs a multi-bit binary number and a single bit to be subtracted from it and yields as outputs a binary number with the same number of bits as the input. Subtracting one from zero should yield zero.

2. Using the `bitAdder` circuit as a model, define a `bitSubtractor` circuit that implements this functionality and use QC to check that it behaves correctly.

3. Using `prop_Adder_Correct` as a model, write down a QC specification for a multiplier circuit that takes two binary numbers of arbitrary width as input and outputs their product.

4. Define a `multiplier` circuit and check that it satisfies your specification. (Looking at how `adder` is defined will help with this, but you'll need a little more wiring. To get an idea of how the recursive structure should work, think about how to multiply two binary numbers on paper.)

5. Using `prop_Toggle_Correct` as a model, invent a QC specification for an n-bit counter and use it to check the behavior of the `counter` circuit.

6. *Optional:* Invent a QC specification that characterizes the behavior of the `memory` circuit.

Submission instructions

- Submit your code in a file `YourName(s)6.hs`.

  Put your name(s) in a comment at the top of the file. Also, please put the approximate number of hours that you spent on this assignment. Give separate numbers for time spent reading and time spent programming.

- Your submission should define a module `Main` that includes an action `main` that runs all your test functions.

- Email the file to both `jschorr@seas.upenn.edu` and `bcpierce@cis.upenn.edu`.