ECE 571
Introduction to SystemVerilog
Spring 2021
Homework 4

1. Using the design of Homework 3, create a pipelined version of your (or my) second multiplier (using carry save adders). There should be a pipeline register between each state to hold intermediate results as well as those portions of the original inputs that will be needed in subsequent stages. There should be a pipeline register after the final stage to keep outputs stable during one clock cycle.

   Create a testbench for your pipelined multiplier module. Use a queue to hold the stimulus (multiplier/multiplicand pairs) for all operations until their product has been produced and verified when it emerges from the pipeline.

2. Create a **package** to support complex numbers. Recall that complex numbers have a real and an imaginary component. Assume these components are 32-bit floating point numbers. Your package should create a user-defined data type for complex numbers called **`complex`**, so that users of the package can declare variables of type complex. It should also define the following functions:

   ```
   function complex AddComplex(input complex M, N);
   ```
   Add two complex numbers, returning a complex number

   ```
   function complex MultComplex(input complex M, N);
   ```
   Multiply two complex numbers, returning a complex number

   ```
   function complex CreateComplex(input shortreal RealPart, ImaginaryPart);
   ```
   Create and return a complex number from two **`shortreal`** components (real and imaginary)

   ```
   function void PrintComplex(input complex C);
   ```
   Accept a complex number and print its components using the format (r: number, i: number) where each component is a **`shortreal`**

   ```
   function void ComplexToComponents(input complex C, output shortreal
   RealPart, ImaginaryPart);
   ```
   Accept a complex number and return (as outputs) the components (real and imaginary)

   Create a module that imports the package and demonstrates the use and correct behavior of each of the functions.

   Recall the rules for addition and multiplication of complex numbers:
   (a+bi) + (c+di) = (a+c) + (b+d)i
   (a+bi) x (c+di) = a x c + (a x d)i + (b x c)i + (b x d)i^2 = (a x c – b x d) + (a x d + b x c)i

All design files and your testbench must be in a single directory that is your username. This directory should be zipped and submitted to D2L as HW4.zip. Individual files should be named as follows. Note that from this assignment forward all files should use .sv (SystemVerilog) suffix and you should make use of all appropriate SystemVerilog constructs we've covered in class.

`multiplypipelined.sv`	Your pipelined multiplier using carry save adders (CSA)
`fulladdr.sv`	Your full adder design
`multiplytb.sv`	Testbench for your pipelined multiplier

complexpkg.sv	Your complex package
complexm.sv	module with initial block to demonstrate package and functions

Your pipelined multiplier testbench and top level module in complex.sv should be declared as follows:

```
module top();
```