ECE 571
Introduction to SystemVerilog
Spring 2021
Homework 3

1. [25] Create a combinational N-bit multiplier declared as

```
module multiply(multiplicand, multiplier, product);
parameter N = 16;
input [N-1:0] multiplicand, multiplier;
output [2*N-1:0] product;
```

   Use the first multiplier design from the lecture slides that uses ordinary ripple carry adders to sum the partial products. Create a Full Adder module and instantiate it within a module called MCell. Use instances of MCells to create the multiplier. Model the Full Adder module and the MCell modules using behavioral dataflow style. Both the Full Adder and the AND function in the MCell should have a propagation delay of 1 time unit.

2. [25] Create a testbench to verify your multiplier.

3. [25] Create a multiplier that uses a carry save adder (consult the third slide of the multiplier lecture slides). You do not need to use the MCell module from above.

4. [25] Verify this multiplier using the same testbench from (2) above.

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 HW2.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.

multiplyripple.sv	Your multiplier using ripple carry adders
multiplyrcsa.sv	Your multiplier using carry save adders (CSA)
fulladdr.sv	Your full adder design
mcell.sv	Your mcell design
multiplytb.sv	Testbench for your multipliers

Your multiply testbench should be declared as follows:

```
module top();
```

You do not need to submit the testbenches for your FullAdder or MCell modules.