# 4-bit Ripple Carry Adder

DDCO Assignment 1

The addition of two numbers is perhaps the most fundamental computation task. Your task in this assignment is to design and simulate a 4-bit[1] adder.

## 1 Design

Once you have designed the adder logic, it needs to be specified using the basic Verilog syntax you have been taught[2] in the file add.v. The entire adder should be composed of purely combinational logic elements that have been discussed earlier. Also, only the Verilog module definition and instantiation syntax discussed in class should be used.

## 2 Simulate

Once the adder is designed, it needs to be simulated to verify proper functionality. In order to do so, appropriate inputs need to be applied to the adder. The inputs are specified as test vectors, four of which are specified on lines 14 to 17 of tb_add.v. You may add additional test vectors to test the adder for additional inputs although it is not mandatory. Be sure to change value of TESTVECS on line 2 whenever the number of test vectors is modified.

To simulate the adder with the test vectors, run the commands:

```
iverilog -o tb_add lib.v add.v tb_add.v
vvp tb_add
```

Above commands should produce the tb_add.vcd file. To view the waveforms, run the command:

```
gtkwave tb_add.vcd
```

For more gtkwave info see the accompanying README file.

[1] Instead of a 4-bit adder, it is reasonably straightforward to make a 32-bit or even 64-bit adder. But in order to keep wiring requirements manageable, we stick to 16-bits.

[2] Essentially, use only instantiation of modules in lib.v to form more complex modules, which in turn get instantiated in even more complex modules, eventually constructing the adder module.