Learning Objectives

1. Introduction to Verilog syntax
2. Introduction to Verilog tools (iverilog, gtkwave)
3. Practice with Verilog coding, testing, and debugging

Work that needs to be handed in

1. **First deadline:** Implement the `sc2_block` in Verilog.

   Using the `sc_block` module we provide (shown on page 2 of the section handout, in SVN in `sc_block.v`) implement the `sc2_block` module as described on the last page of the section handout. Instantiate the `sc_block` module directly; **do not** inline it.

   Your code should be placed in the `sc2_block.v` file in SVN; **it should be the only module in that file.**

   In the file `sc2_block_tb.v`, write a test bench for `sc2_block` using `sc_block_tb.v` as a guideline.

   You should be able to compile, run, and debug your code by running (note that these are three separate commands):

   ```
   iverilog -o sc2 -Wall sc2_block_tb.v sc2_block.v sc_block.v
   ./sc2
   gtkwave sc2.vcd &
   ```

   The final command assumes that you use the command

   ```
   $dumpfile("sc2.vcd");
   ```

   in your test bench. If you name the dump file differently, adjust the last command accordingly.

   Make sure you commit both `sc2_block.v` and `sc2_block_tb.v` into SVN to handin.

2. **Second deadline:** Test a "black box" module to determine its truth table. In your SVN repository you have a circuit called `blackbox.v`. This Verilog module was designed to be not easily analyzed by visual inspection. You should write a test bench that allows you to discover the truth table for this module and fill in the truth table in the `blackbox.tt` file. Make sure you commit this file.