Learning Objectives

1. Introduction to Verilog syntax
2. Introduction to Verilog tools (iverilog, gtkwave)
3. Practice with Verilog coding, testing, and debugging
4. Practice designing combinational circuits using a truth table

Work that needs to be handed in

1. **First deadline:** Implement the `sc2_block` in Verilog.
   Using the `sc_block` module we provide in git in `sc_block.v` (shown on page 2 of the discussion section handout) implement the `sc2_block` module as described on the last page of the discussion section handout. Instantiate the `sc_block` module directly; **do not** inline it.
   Your code should be placed in the `sc2_block.v` file in git; **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 git to handin.
2. **Second deadline:** Complete **Verilog Black Box** problem on PrairieLearn. Commit `farmer.tt` to git
   Construct a truth table to help you solve the riddle. Once upon a time a farmer (f) went to a market and purchased a fox (x), a goose (g), and a bag of beans (b). On his way home, the farmer came to the left bank of a river and rented a boat. When crossing the river by boat, the farmer could carry only himself and only one of his purchases: the fox, the goose, or the bag of beans.
   If the farmer does not stay with the fox or the goose, the fox would eat the goose if possible and the goose would eat the beans if possible.
   The farmer's challenge was to carry himself and his purchases from the left bank (0) of the river to the right bank (1) of the river, leaving each purchase intact.
   Use the variables f, x, g, and b to represent the position of the farmer, fox, goose, and bag of beans respectively. When a variable's value is 0, the object it represents is on the left bank of the river. When a variable's value is 1, the object it represents is on the right bank of the river. Fill in the truth table in `farmer.tt` with output e that is 1 for all combinations of positions for which something gets eaten.