Practice problems on Verilog (FSM)

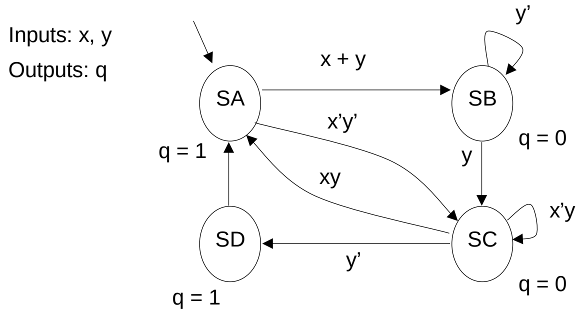
For each problem, write Verilog code and the testbench (make sure that it works in both behavioral simulation and post-synthesis functional simulation)

Notes:

a) the example that we did in class are on D2L: Content->Unit Verilog-> Sequential_circuits_example_Practice_problems

b) you practiced writing Verilog code for FSM in lab 3 and lab 4

1) Write the code and testbench for



Note: Your circuit should also have an active-high synchronous reset (Rst).

When writing testbench, create the sequence of inputs x any y such that several cases can be shown.

For example,

Case 1: come up with x and y such that the state changes (at each clock cycle) from

SA -> SB -> SC -> SD ->SA

Case 2 (continue from case 1 in the testbench)

SA -> SB -> SB -> SC ->SC -> SA

Case 3 (continue from Case 2 in the testbench)

SA -> SC -> SC -> SD -> SA -> SC->SA

Reminder: the purpose of the testbench is to test whether your Verilog code describes the circuit that correctly behave according to your design (state diagram).

2) Design a sequence detector, which has one input b and one output w, that accepts a sequence of bits (one bit (0 or 1) at a time) and outputs 1 when target sequences have been detected. The circuit outputs w = 1 when the previous three values of b were **110 or 101** (target sequences). **NON-overlapping** sequence is detected. Note: Your circuit should also have an active-high synchronous reset (Rst).

b 01**110101101**0100**110**0

w 0000**1**00**1**00**1**000000**1**0

When writing testbench, create sequence of b as shown above.

3) Design a synchronous sequential circuit which has an input w and an output z. The circuit outputs z = 1 when the previous four values of w were 1001 or 1111; otherwise z = 0. **Overlapping** input pattern is detected. Note: Your circuit should also have an active-high synchronous reset (Rst).

An example of the desired behavior is

w: 010**1111**001**1001**1111

z: 000000**1**00**1**000**1**00**1**1

When writing the testbench, create sequence of w as shown above.

4) Design a sequential circuit which has 2 inputs w1 and w2 and an output z. Its function is to compare the input sequences on the two inputs. If w1 = w2 during any four consecutive clock cycles, the circuit produces z = 1; otherwise z = 0.
Note: Your circuit should also have an active-high synchronous reset (Rst).
For example,

w1: 0110111000110

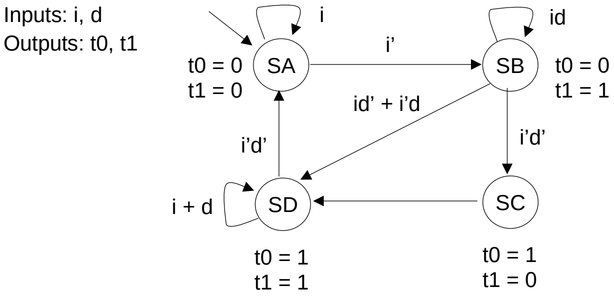
w2: 1110101000111

z: 0000**1**0000**1****1****1**0

Hint: Let k = w1 xor w2 be an input to your sequential circuit -> if w1 = w2, k = 0 (based on XOR truth table).

When writing the testbench, create sequence of w1 and w2 as shown above.

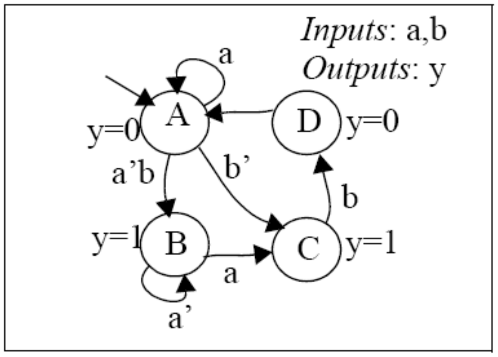
5) Write the code and testbench for



When writing testbench, create sequence of i, d to test several cases (similar idea as discussed in prob 1))

Note: Your circuit should also have an active-high synchronous reset (Rst).

6) Write the code and testbench for



When writing testbench, create sequence of i, d to test several cases (similar idea as discussed in prob 1))

Note: Your circuit should also have an active-high synchronous reset (Rst).