Project: Product Machine and State Enumeration

Instructions:

1) The report should be typed in MS word. Figures should be plotted with a software. Scanned or cut/paste hand drawn figures and writing are not acceptable.
2) Turn in your report to the D2L by the deadline.
3) Each group (less than 3 students) should submit one report.

Problem 1: Design a sequential circuit S1 with the following requirements

a) The combinational logic part for the next state function should contain **at least 2 NAND gates, 2 XOR gates, and one NOR.**
b) The number of the flip-flops should be at least 2.
c) The sequential circuit should have at least one output.
d) Make sure that your design be different from those used by other students. If your design is identical to some design used by other students, further investigation will be conducted and you will be asked to revise your design and redo the work.

Task:

1) Draw your sequential circuit S1.
2) Create the next state function table for S1.
3) From an initial state (you decide), explore the entire state space and draw the state transition graph.

Problem 2: Get a copy of S1 and name it as S2.

a) Except the inputs, please add the index 1 to all the wire names of S1, while adding the index 2 to all the wire names of S2.
b) To use the shared input product machine model, please make sure that the corresponding inputs have the same variable names.
c) Build the product machine with the shared input $P=S1\times S2$ (as defined in class).

Task:

1) From an initial global state of P where S1 and S2 are in the same individual state, explore the entire state space of P and draw the state transition graph.

2) *Equivalence checking by state enumeration*: checking the outputs at each global state to draw a conclusion.

3) *Equivalence checking by symbolic simulation*: Perform the symbolic simulation for the product machine for $n>4$ cycles. Create a table to keep all the symbolic expressions on the output for simulation cycles. Use the Boolean algebra to demonstrate the equivalence of the corresponding outputs for each simulation cycle.

4) Modify one gate in S2, so that S1 is not the same as S2. Perform both 2) and 3) to disprove the equivalence of S1 and S2.