CSE369 Practice Problems (lec 5)

Problem 1: XOR Counter

1.1] Draw the state diagram for an FSM that takes two inputs and has one output. It should pulse its output high if the XOR of its inputs has been high for three cycles straight; that is, if either one of (but not both of) the input bits has been high over the past three cycles. It's ok if the bit that is on "switches" from one cycle to the next.

1.2] How many cycles would we need to **fully** test this FSM from reset?

1.3] Draw a **truth table** for your state diagram. The inputs to the truth table should be the two bits of the present state **PS[1:0]** and and the two bits of the FSM input, **IN[1:0]**. The outputs should be the two next state bits **NS[1:0]** and the output **Out**

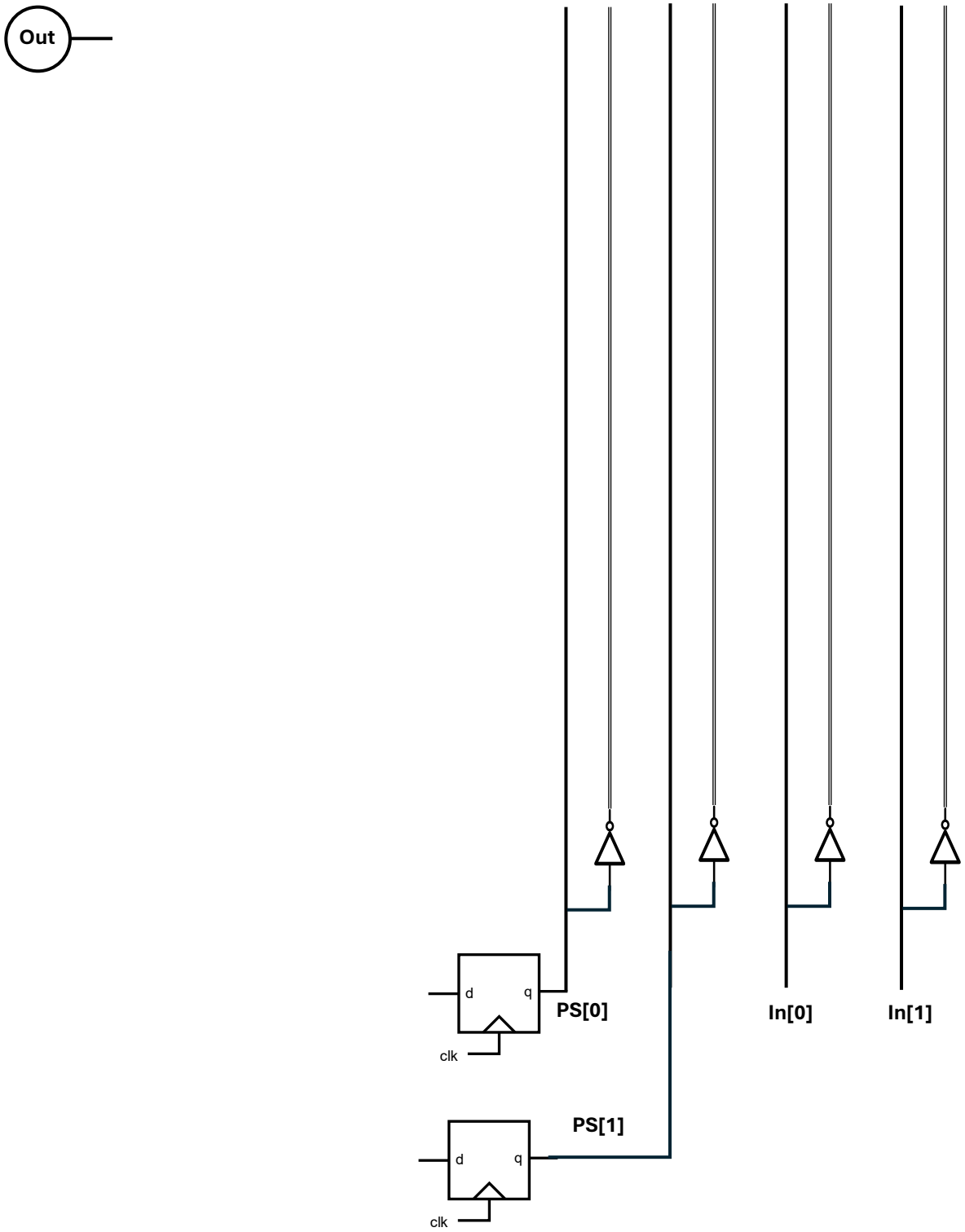
1.4] When you're done with the truth table, solve a k-map for each of the three output bits and use it to make a minimal two-level Boolean equation. Write that equation here:

NS[0] =

NS[1] =

Out =

1.5] Draw gates on the diagram below to complete your FSM design:



Problem 2: Bunny Hopper

2.1] Draw the state diagram for an FSM with no inputs and one output that outputs the following repeating sequence of bits:

0, 1, 0, 1, 1, 0, 1, 1, 1

2.2] How many bits do we need to store the state of this FSM?

2.3] If we were to draw it out (but like, please don't) -- how many rows should the truth table of this FSM have? How many Karnaugh maps would you need to implement the logic for it?

2.4] How many cycles would we need from reset to **fully** test this FSM?