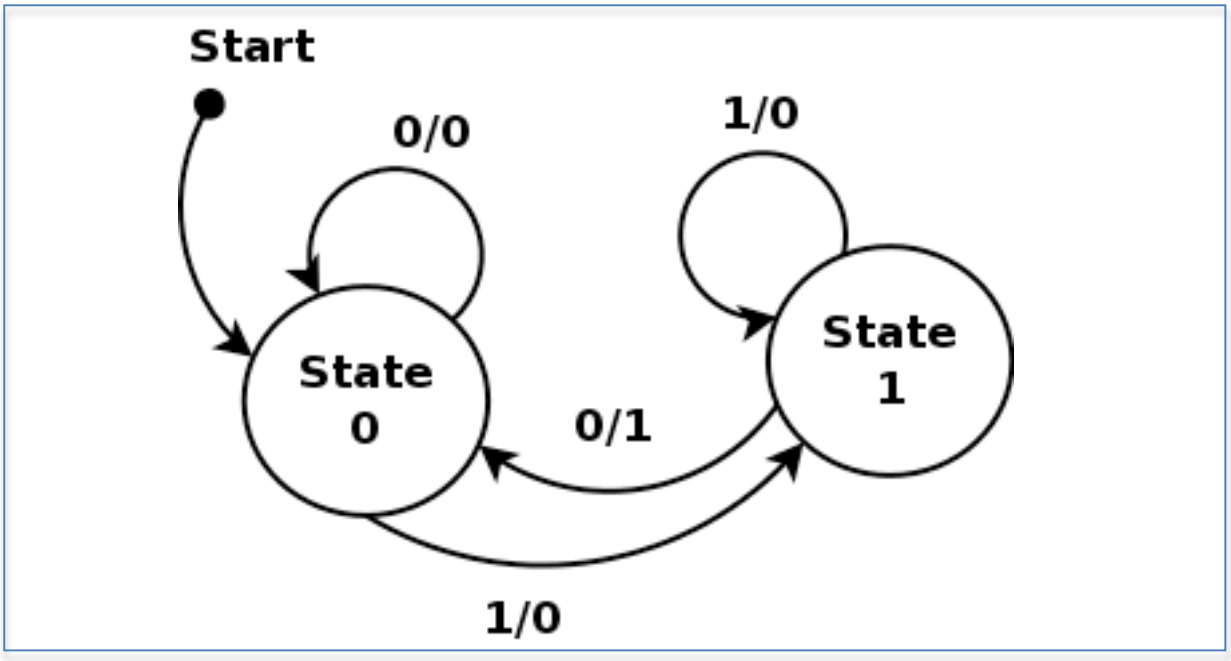
# Simple State Machine

This is an example of a simple state machine for cs270.

**Instructions:** Design a Logisim circuit that implements the state machine shown below:

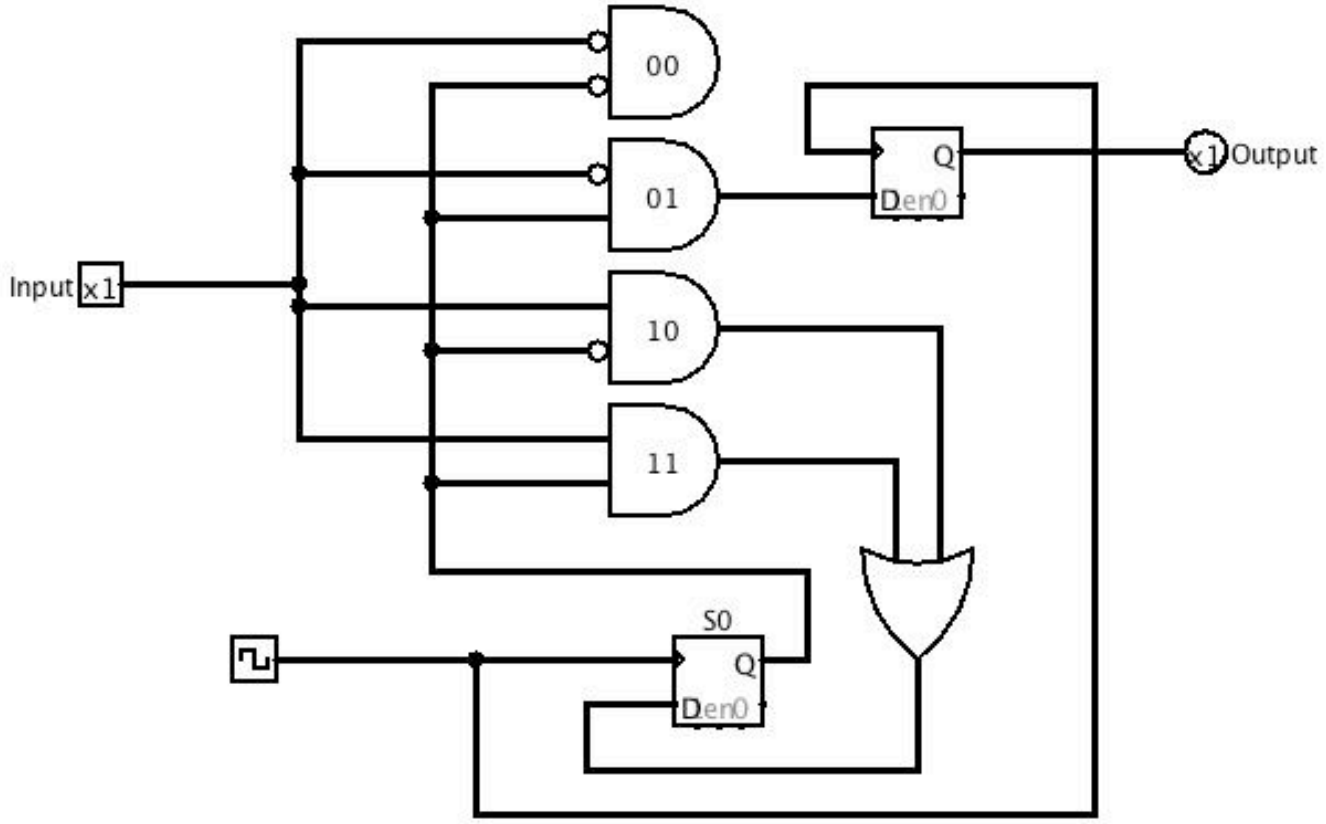



Here are a few clarifications that may help you, and questions you should be able to answer:

- What is the sequence of bits that is detected by this state machine? 10
- State should be stored in D-latches, how many do you need? **1, since only 2 states**
- The truth table is combinational logic similar to what you have already done.
- You must use a D-latch the output, since it's on a transition.
- Optimization is allowed and encouraged, the fewer gates the better!

| *Input* | *Current State* | *Output* | *Next State* |
|---|---|---|---|
| *0* | *0* | *0* | *0* |
| *0* | *1* | *1* | *0* |
| *1* | *0* | *0* | *1* |
| *1* | *1* | *0* | *1* |

Answer Sheet:




Note that an optimized circuit could do without three of the AND gates pictured above!