Finite State Machine Practice Problem

Your task is to construct a circuit that detects the sequence '1101' in a bitstream[1].

a) **Draw** the state diagram for a finite state machine (FSM) that detects the sequence. Your FSM will process 1 bit at a time.
b) **Encode** the states of the FSM by numbering each state. List your encoding in a table or show the encoding on your diagram.
c) Write the **truth table for the next state function**. Your truth table should have one row for every combination of possible input values and state.
d) Write the **truth table for the output function**. Recall that the output is 1-bit (i.e., whether the pattern was seen or not). Remember that the output depends only on the current state of the FSM.
e) **Implement** the circuit in *Logisim*. You may use combinational analysis or K-Maps to simplify your circuits.

For part e, be sure that you use a clock, a button to let you test different inputs, and one or more D flip-flops to store the state.

Tip: Under the simulate menu item, you can can specify a clock tick frequency (e.g., 0.5Hz) and enable ticks. This will allow you do easily test your circuit. As the clock ticks, poke your circuit's button to make sure that it correctly detects the proper bit pattern.

[1] I.e., the circuit should detect two consecutive ones followed by a zero followed by a one. The last one that is seen may itself by part of another valid sequence. E.g., if the circuit sees “1011011” then the circuit will output a one twice.