A synchronous finite state machine has an input \( x_{\text{in}} \) and an output \( y_{\text{out}} \). When \( x_{\text{in}} \) changes from 0 to 1, the output \( y_{\text{out}} \) is to assert for three cycles, regardless of the value of \( x_{\text{in}} \), and then de-assert for two cycles before the machine will respond to another assertion of \( x_{\text{in}} \).

The machine is to have active-low asynchronous reset.

(a) Draw the state diagram of the machine.
(b) Write Verilog model of the machine.

module StateDiagram (output y_out, input x_in, clk, reset_b);

Test data will change on the falling edge of the clock and your machine should change state on the rising edge.

Your design must include an asynchronous reset.