L4 - REGISTER

Any assignment to a signal on a rising clock edge will be synthesized as a register, where the LHS is the output of the register, and the RHS is the input. This might seem obvious for this simple example, but it is critically important to remember this rule for bigger examples because adding or omitting a register is a very common source of bugs.

When we "design the circuit" before writing the code, a huge part of that design is determining the number of registers. To ensure that the synthesized circuit has the same number of registers as our design circuit, we need to understand how registers get created during synthesis.

Use VHDL to describe the following four register types:

- a register with an active high, asynchronous reset
- a register with an active high, synchronous reset
- a register with an enable, and an active high, asynchronous reset; the register holds its value when enable isn't asserted
- a register with an enable, and an active high, synchronous reset; the register holds its value when enable isn't asserted.

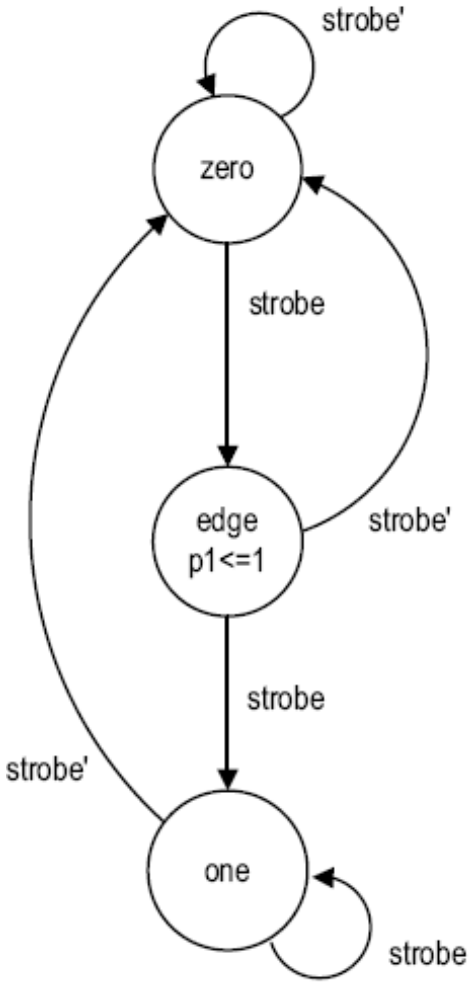
L5 - FSM

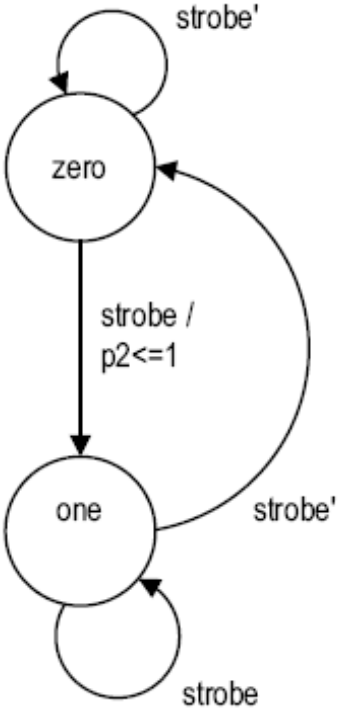
We assume that a synchronous system is connected to a slowly varying input signal, strobe, which can be asserted to ‘1’ for a long time (much greater than the clock period of the FSM). An edge detection circuit is used to detect the rising edge of the strobe signal. It generates a “short” pulse when the strobe signal changes from ‘0’ to ‘1’.

The basic design idea is to construct an FSM that has a zero state and a one state, which represent that the input has been ’0’ or ’1’ for a long period of time respectively. The FSM has a single input signal, strobe, and a single output signal. The output will be asserted “momentarily” when the FSM transits from the zero state to the one state.

We first consider a design based on a Moore machine: there are three states since in addition to the zero and one states, the FSM also has an edge state. When strobe becomes ’1’ in the zero state, it implies that strobe changes from ’0’ to ’1’. The FSM moves to the edge state, in which the output signal, p1, is asserted.

The second design is based on a Mealy machine: it consists of only the zero and one states. When strobe changes from ‘0’ to ‘1’ in the zero state, the FSM moves to the one state. The output signal, p2, is asserted in the zero state whenever strobe is ‘1’. When the FSM moves to the one state, p2 will be deasserted.





Use VHDL to describe the two FSMs