

Lesson 18

Finite State Machine

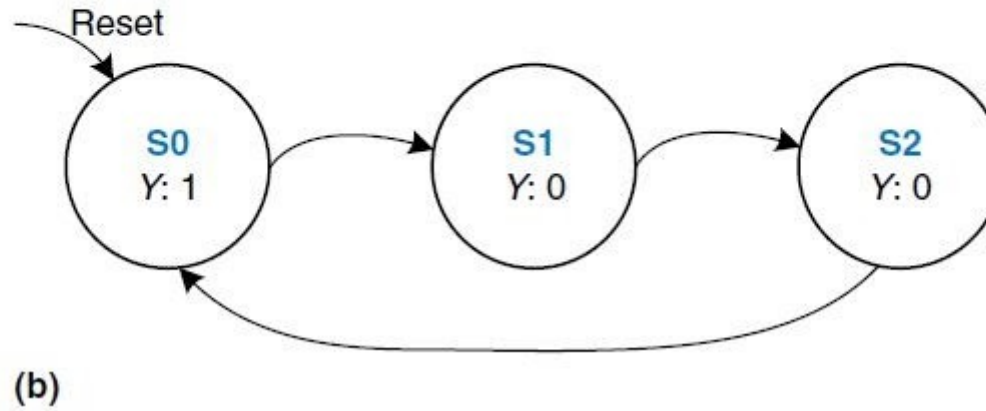
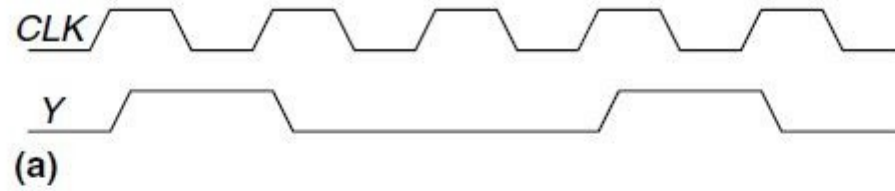
Mohd Saufy Rohmad

Technical Trainer and Consultant

FSM

- Rather than going through VHDL examples of mealy and moore machines, I would like to show you an example how it is done.
- We will take a look at a divide-by-3 finite state machine.
- First, I would like to explain what is a divide-by-3 counter.
- A divide-by-3 counter has one output and no inputs.
- The output divides the frequency of the clock by 3.
- The waveform and the state diagram is shown below.

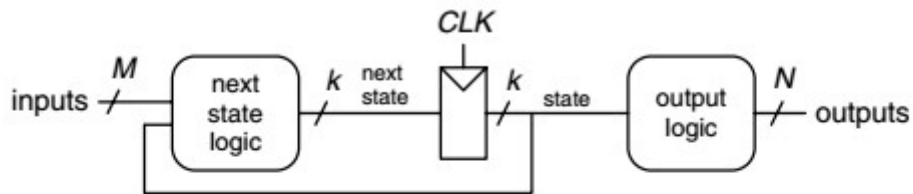
FSM



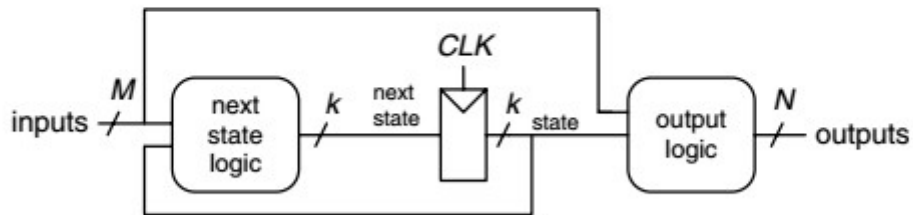
FSM

- If you cant remember the Mealy and Moore machines please revisit Hour 3.
- Regardless of the types, FSMs generally consist of three parts.
 1. Register (that saves the states)
 2. Logic to compute next state
 3. Output logic
- The parts of an FSM is shown below.

FSM



(a)



(b)

Moore (a) and mealy (b) machine

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity FSM_example1 is
port (clk, reset: in STD_LOGIC;
      y: out STD_LOGIC);
end;
architecture synth of divideby3FSM is
    type statetype is (S0, S1, S2);
    signal state, nextstate: statetype;
begin
    — — state register
    process (clk, reset) begin
        if reset = '1' then state <= S0;
        elsif clk'event and clk = '1' then
            state <= nextstate;
        end if;
    end process;
    — — next state logic
    nextstate <= S1 when state = S0 else
        S2 when state = S1 else
        S0;
    — — output logic
    y <= '1' when state = S0 else '0';
end;

```

FSM

- The code also has three parts, state register, next state logic and output logic.
- A new enumeration data type, `statetype` is declared with three states `S0`, `S1` and `S2` which is similar to the states defined in the figure above.
- `S0` is the reset state and an asynchronous reset is used in the VHDL code.

Thank You