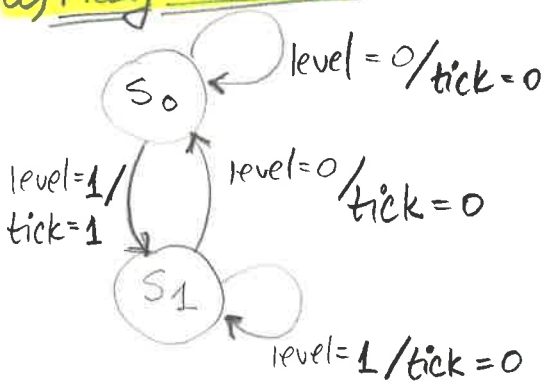
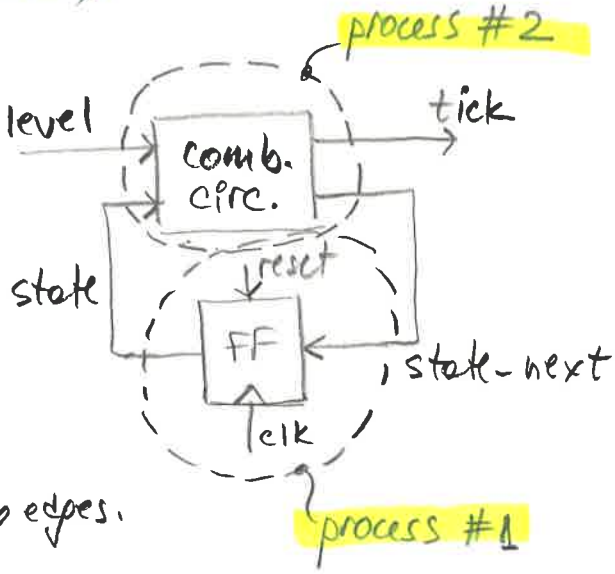
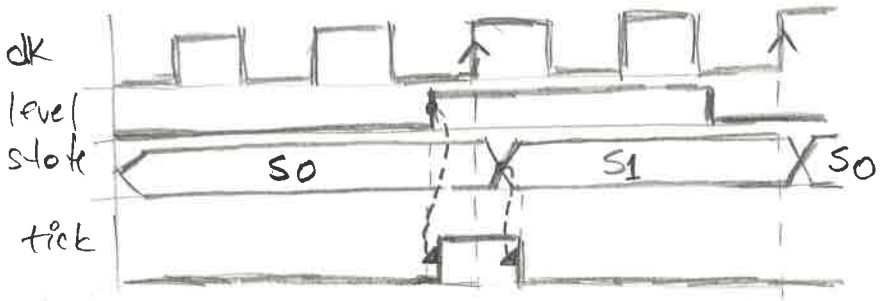
More on FSM's (Finite State Machines)

a) Mealy machine





Mealy machine : edge detector of rising edges.

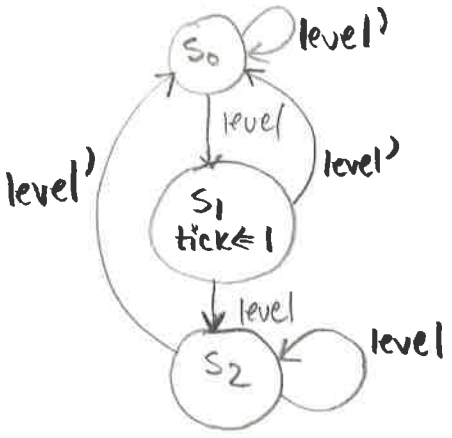


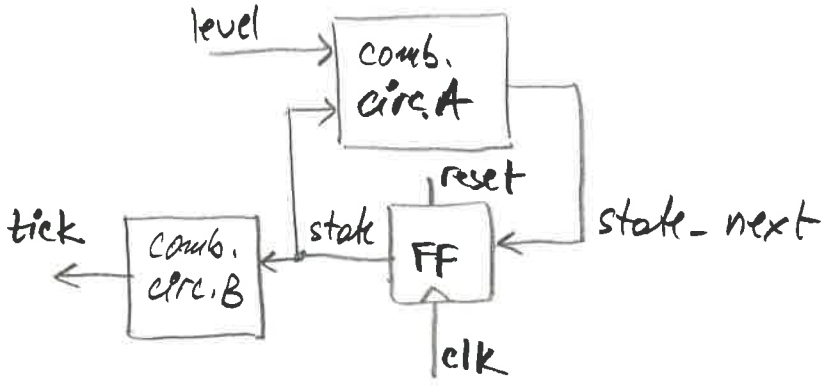

```
library ieee;
use ieee.std_logic_1164.all;
entity edge_detect is
    port(
        clk, reset : in std_logic;
        level : in std_logic;
        tick : out std_logic
    );
end edge_detect;

architecture mealy_arch of edge_detect is
    type state_type (S0, S1);
    signal state, state_next: state_type;
begin
    -- state register ; process #1
    process (clk, register)
    begin if (reset = '1')
            state <= S0;
        elseif (clk'event and clk = '1') then
            state <= state_next;
        end if;
    end process;
```

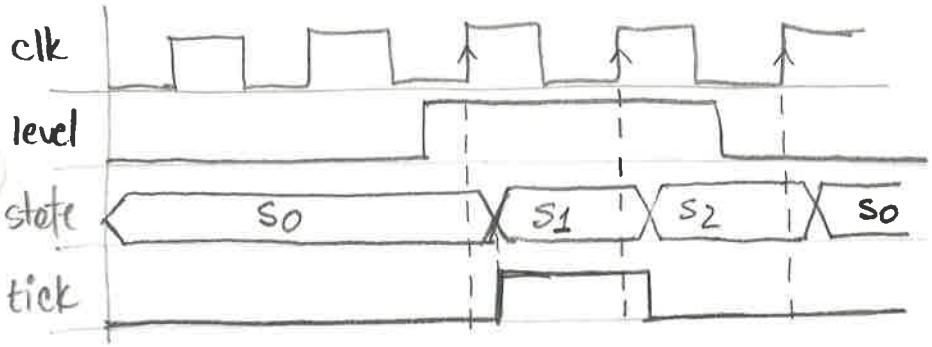
← use two processes model!

```
-- next state & output logic ; process#2
process ( state, level)
begin
    state_next <= state;
    tick <= '0';
    case state is
        when S0 =>
            if level = '1' then
                state_next <= S1;
                tick <= '1';
            end if;
        when S1 =>
            if level = '0' then
                state_next <= S0;
            end if;
    end case;
end process;
end mealy_arch;
```

b) Moore machine:







Note: Entity declaration is the same!

```
architecture moore_arch of edge_detect is
    type state_type is (S0, S1, S2);
    signal state, state_next : state_type;

begin
    -- state register; process #1
    begin
        if (reset = '1') then
            state <= S0;
        elseif (clk'event and clk = '1') then
            state <= state_next;
        end if;
    end process;
    -- next state and output logic; process #2
    process (state, level)
    begin
        state_next <= state;
        tick <= '0';
        case state is
            when S0 =>
                if (level = '1') then
                    state_next <= S1;
                end if;
            when S1 =>
                tick <= '1';
                if (level = '1') then state_next <= S2;
                else state_next <= S0;
                end if;
            when S2 =>
                if (level = '0') then
                    state_next <= S0;
                end if;
        end case;
    end process;
end moore_arch;
```