Edge Detection using Mealy & Moore Machines

1. Design

```
module edgeDetector(
    input clk,
    input level,
    input rst,
    output reg mealy_tick,
    output reg moore_tick
    );
    localparam zeroMealy = 1'b0, oneMealy = 1'b1;
    localparam [1:0]
        zeroMoore = 2'b00,
        edgeMoore = 2'b01,
        oneMoore = 2'b11;
    reg stateMealy_reg, stateMealy_next;
    reg [1:0] stateMoore_reg, stateMoore_next ;
    //Reset State
    always@(posedge clk, posedge rst) begin
        if (rst) begin
            stateMealy_reg <= zeroMealy;
            stateMoore_reg <= zeroMoore;
        end
        else begin
            stateMealy_reg <= stateMealy_next;
            stateMoore_reg <= stateMoore_next;
        end
    end
    //mealy Design
        always @(stateMealy_reg, level) begin
            //store current state
```

```
    stateMealy_next = stateMealy_next;
    mealy_tick <= 1'b0;
    case (stateMealy_reg)
      zeroMealy:
        if(level) begin   //when level=1
          stateMealy_next <= oneMealy;
          mealy_tick <= 1'b1;
        end
      oneMealy:
        if (~level) begin     //when level=0
          stateMealy_next <= zeroMealy;
        end
    endcase
  end
//Moore Design
  always @(stateMoore_reg, level) begin
    //Store the current state
    stateMoore_next = stateMoore_reg;

    moore_tick <= 1'b0;// set moore_tick to 0
    case (stateMoore_reg)
      zeroMoore:
      if (level) begin
        stateMoore_next <= edgeMoore;
      end
      edgeMoore: begin
        moore_tick <= 1'b1;
        if(level)
        stateMoore_next <= oneMoore;
        else //level=0
```

```
                stateMoore_next <= zeroMoore;
            end
            oneMoore:
            if (~level) begin
                stateMoore_reg <= zeroMoore;
            end
        endcase
    end
endmodule
```

2. Test bench:-

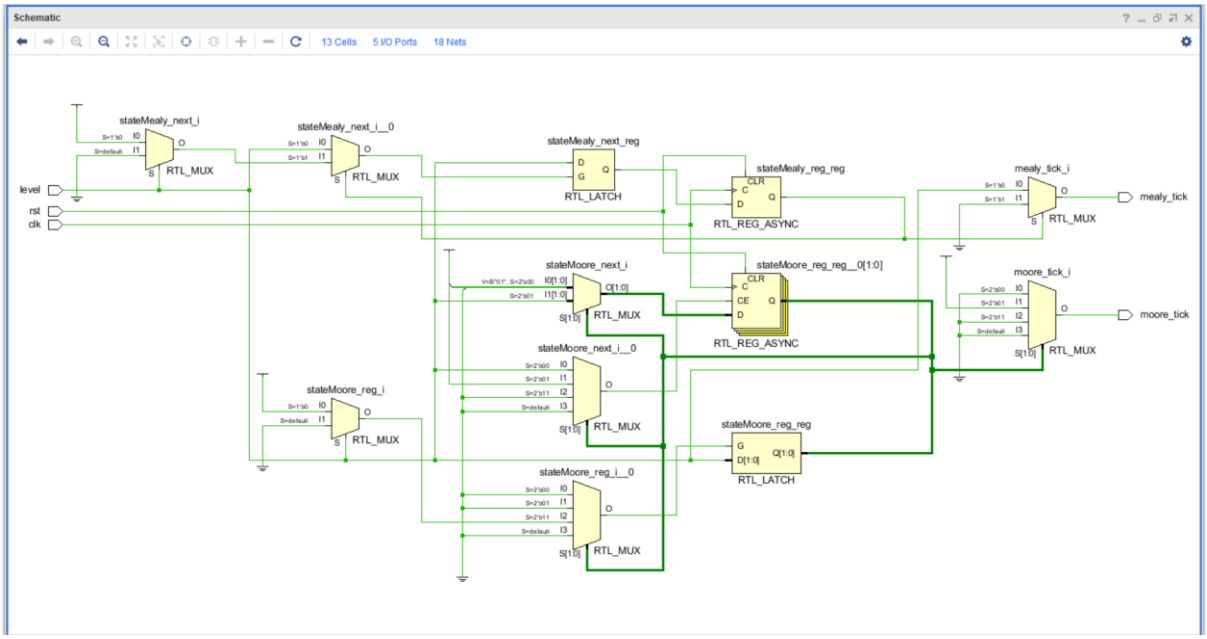
```
module edge_detect_tb;
 // Inputs
 reg clk, level,reset;
// Outputs
 wire moore_tick, mealy_tick;

 edgeDetector uut(.clk(clk), .level(level), .rst(reset), .mealy_tick(mealy_tick), .moore_tick(
moore_tick));
 initial begin
 clk = 0;
 forever #5 clk = ~clk;
 end
 initial begin
  // Initialize Inputs
  level = 0;
  reset = 1;
  #30;
     reset = 0;
  #40;
  level = 1;
```

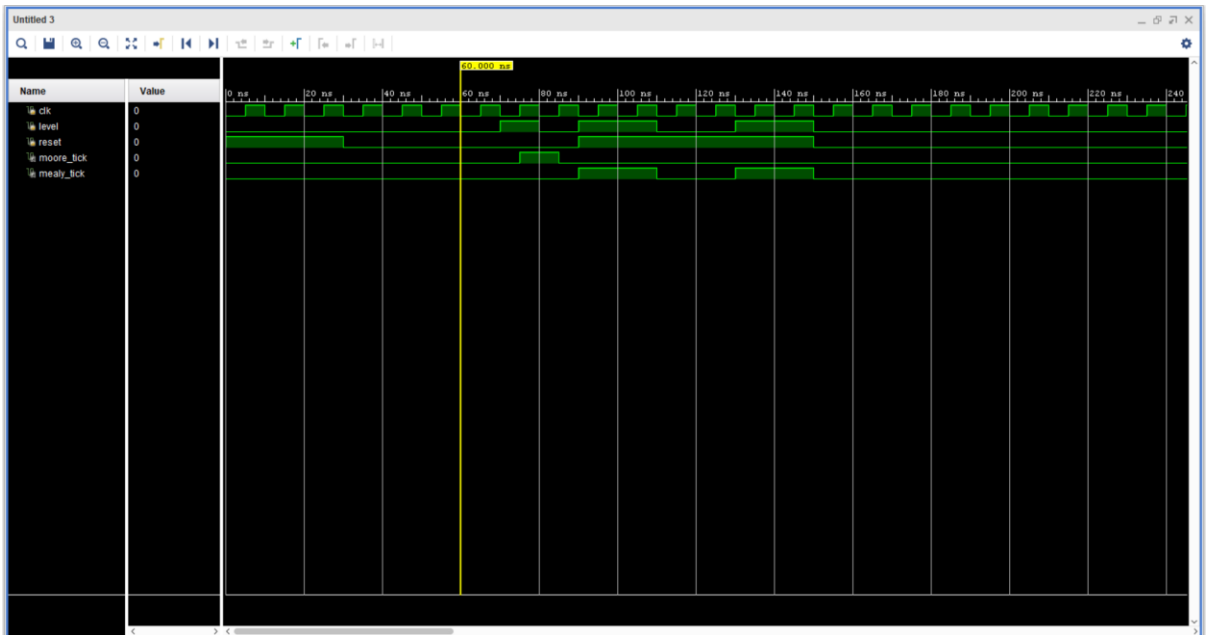
```
#10;
level = 0;
#10;
level = 1;
reset = 1;
#20;
level = 0;
#20;
level = 1;
#20;
reset = 0;
level = 0;

end
endmodule
```

3. Output:-



4. Waveforms:-


Untitled 3
Name
Value
clk
level
reset
moore_tick
mealy_tick
0
60.000 ns
0 ns
20 ns
40 ns
60 ns
80 ns
100 ns
120 ns
140 ns
160 ns
180 ns
200 ns
220 ns
240