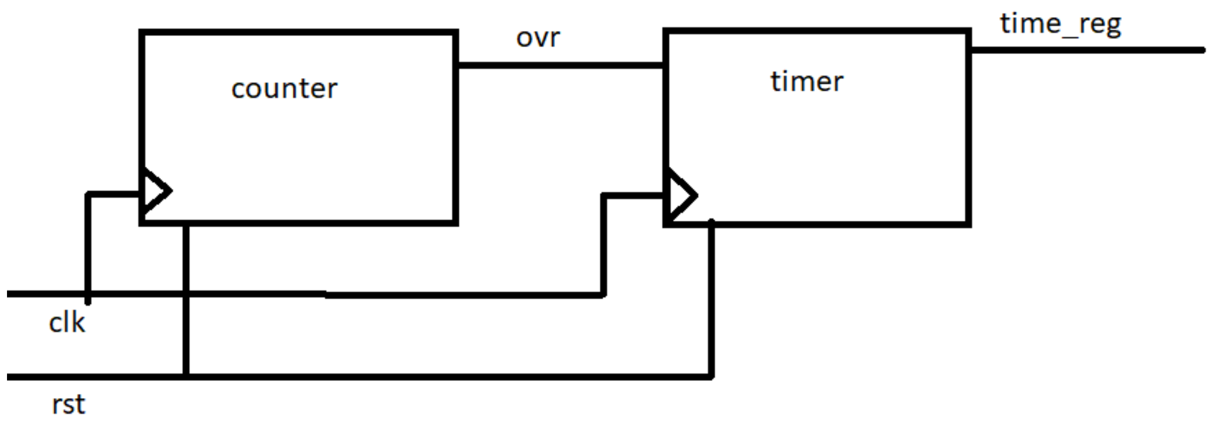

```
module cnt #(parameter FREQ_KHZ = 5000) (clk, rst, ovr);
  input clk;
  input rst;
  output ovr;

  reg [31:0] count_reg, count_nxt;
  reg ov_reg;
  localparam LIMIT = FREQ_KHZ;

  //Bloc secvential
  always @(posedge clk or negedge rst)
   begin
    if(rst == 0) begin
      count_reg <= 0;
      ov_reg <= 0;
    end
    else count_reg <= count_reg + 1;
   end

  //Bloc combinational
  always @(count_reg)
  begin
    ov_reg = 0;
    if(count_reg == LIMIT)
    begin
      count_reg = 0;
      ov_reg = 1;
    end
  end

    assign ovr = ov_reg;

endmodule


module tmr #(parameter FREQ_KHZ = 5000) (clk, rst, time_reg);
  input clk;
  input rst;
  output reg [31:0] time_reg;  //time in ms

  reg [31:0] time_nxt;
  wire ov;
  cnt # (.FREQ_KHZ(FREQ_KHZ)) counter(.clk(clk), .rst(rst), .ovr(ov));

  //Bloc secvential
  always @(posedge clk or negedge rst)
   begin
    if(rst == 0) begin
      time_reg <= 0;
    end
    else time_reg <= time_nxt;
   end

  //Bloc combinational
  always @(ov)
  begin
```

```
    if(ov == 1)
    begin
      time_nxt <= time_reg + 1;
    end
     else time_nxt <= time_reg;
  end

endmodule


module tb;
  reg clk, rst;
  wire [31:0] time_reg;

  `define freq_khz 10000

  initial begin
    $monitor("%0t\t%d", $time, time_reg);
  end

  tmr # (.FREQ_KHZ(`freq_khz)) timer(.clk(clk), .rst(rst), .time_reg(time_reg));

  localparam CLKP = 1000000/`freq_khz;

  initial begin
    rst = 0;
    #1
    rst = 1;
  end

  initial begin
    clk = 0;
    forever #(CLKP/2) clk = ~clk;
  end
endmodule
```