Design semafor.v

```
//------------------------------------------------------------------------
-----
// Universitatea Transilvania din Brasov
// Facultatea IESC
// Proiect     : Laborator ED
// Modul       : semafor
// Autor       : Cordonaș Cristian
// Data        : Apr. 29, 2023
//------------------------------------------------------------------------
-----

module semafor (
input   clk             , // semnal de ceas,
input   rst_n           , // semnal de reset asincron activ 0
input   buton           , // buton pietoni
output  semafor_masini  ,
output  semafor_pietoni
);

reg [1:0] stare_prezenta;
reg [1:0] stare_viitoare;
reg [8:0] counter       ;
reg buton_apasat        ;
reg ok1                 ;
reg ok2                 ;

// coduri stari (unice)
localparam stare1   = 2'b01;  //rosu pentru masini, verde pentru pietoni
localparam stare2   = 2'b10;  //verde pentru masini, rosu pentru pietoni
localparam stare3   = 2'b00;  //galben pentru masini, rosu pentru pietoni

initial begin
counter      <= 8'b0;
buton_apasat <= 1'b0;
ok1 <= 1'b0;
ok2 <= 1'b0;
end

always @(posedge clk)
counter <= counter + 1;

// modeleaza registrul de stare
always @(posedge clk or negedge rst_n)
if (~rst_n) begin
            stare_prezenta <= stare1;  // stare initiala
            counter        <= 8'b0;
end
else        stare_prezenta <= stare_viitoare;

// modeleaza circuitul combinational de stare
always @(*)
case (stare_prezenta)
stare1  : begin
          ok2 <= 1'b0;
          ok1 <= 1'b1;
          buton_apasat <= 1'b0;
          if (counter == 'd30) begin
```

```
                stare_viitoare <= stare3;
                counter <= 8'b0;
            end
            end
stare2   : begin
            ok1 <= 1'b0;
            ok2 <= 1'b1;
            if (buton && !buton_apasat) begin
                buton_apasat <= 1'b1;
                counter <= 8'b0;
            end
            else if (buton_apasat == 1'b1 && counter == 'd60) begin
                     counter        <= 8'b0;
                     stare_viitoare <= stare3;
            end
            end
stare3   : begin
            buton_apasat <= 1'b0;
            if (counter == 'd5) begin
                stare_viitoare <= stare1;
                counter        <= 8'b0;
                if(ok1)
                   stare_viitoare <= stare2;
                else if(ok2)
                   stare_viitoare <= stare1;
            end
            end
default : stare_viitoare <= stare1;
endcase

// modeleaza circuitul combinational de iesire
assign semafor_masini  = (stare_prezenta == stare2);
assign semafor_pietoni = (stare_prezenta == stare1);

endmodule // semafor
```

Testbench semafor_test.v

```
//------------------------------------------------------------------------
-----
// Universitatea Transilvania din Brasov
// Facultatea IESC
// Proiect     : Laborator ED
// Modul       : semafor_test
// Autor       : Cordonaș Cristian
// Data        : Apr. 29, 2023
//------------------------------------------------------------------------
-----

`timescale 1s/1s

module semafor_test;
wire   clk            ;
wire   rst_n          ;
reg    buton          ;
wire   semafor_masini ;
wire   semafor_pietoni;
```

```
ck_rst_tb #(
.CK_SEMIPERIOD ('d5)
) i_ck_rst_tb (
.clk     (clk   ),
.rst_n   (rst_n )
);

initial begin
  buton <= 1'bx;
  @(negedge rst_n);
  buton <= 1'b0;
  @(posedge rst_n);
  @(posedge clk);
  buton <= 1'b1;
  #1;
  buton <= 1'b0;
  #3;
  buton <= 1'b1;
  #1;
  buton <= 1'b0;
  #20;
  buton <= 1'b1;
  #1;
  buton <= 1'b0;
  #10;
  buton <= 1'b1;
  #1;
  buton <= 1'b0;
  #5;
  buton <= 1'b1;
  #3;
  buton <= 1'b0;
  #30;
  buton <= 1'b1;
  #1;
  buton <= 1'b0;
  #22;
  buton <= 1'b1;
  #1;
  buton <= 1'b0;
  #100;
  $display ("%M %0t INFO: Final simulare.", $time);
  $stop;
end

semafor i_semafor(
.clk               (clk            ),
.rst_n             (rst_n          ),
.buton             (buton          ),
.semafor_masini    (semafor_masini ),
.semafor_pietoni   (semafor_pietoni)
);

endmodule // semafor_test
```

ck_rst_tb.v

```
//------------------------------------------------------------------------
-----
// Universitatea Transilvania din Brasov
// Facultatea IESC
// Proiect     : Laborator ED
// Modul       : ck_rst_tb
// Autor       : Cordonaș Cristian
// Data        : Apr. 29, 2023
//------------------------------------------------------------------------
-----

`timescale 100ms/100ms

module ck_rst_tb #(
parameter CK_SEMIPERIOD = 'd10        // semi-perioada semnalului de ceas
)(
output reg              clk         , // ceas
output reg              rst_n         // reset asincron activ 0
);

initial
begin
  clk = 1'b0;               // valoare initiala 0
  forever #CK_SEMIPERIOD    // valoare complementata la fiecare semi-
perioada
    clk = ~clk;
end

initial begin
  rst_n <= 1'b1;    // initial inactiv
  @(posedge clk);
  rst_n <= 1'b0;    // activare sincrona
  @(posedge clk);
  @(posedge clk);
  rst_n <= 1'b1;    // inactivare dupa doua perioade de ceas
  @(posedge clk);   // ramane inactiv pentru totdeauna
end

endmodule // ck_rst_tb
```

Explicatii: in modulul semafor avem intrari pentru clock, reset asincron activ in 0 si butonul pentru pietoni. Iesirile sunt semaforul pentru masini si cel pentru pietoni. Intern am declarat starea prezenta, starea viitoare, verificarea daca butonul a fost apasat, un counter si doi semnalizatori pentru a memora starea in care ne-am aflat. Cele 3 stari sunt: rosu pentru masini-verde pentru pietoni, verde pentru masini-rosu pentru pietoni si galben pentru masini-rosu pentru pietoni. Starea initiala este starea 1, urmata de starea 3 dupa 30s, apoi starea 2 dupa 5s, iar starea 3 dupa minimum 60s si daca butonul a fost apasat si apoi revenim in starea 1. Semaforul pentru masini este activ in starea 2 si semaforul pentru pietoni este activ in starea 1. Ambele sunt 0 in starea 3.

In testbench am generat stimuli pentru semnalul butonului, iar in ck_rst_tb am generat semnalul de reset asincron si cel de ceas cu frecventa de 1Hz (initializate in testbench).

Formele de unda generate: