Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería Eléctrica
IEE2713 - Sistemas Digitales

Tarea Nº4

27 de Junio de 2019

1. Diagrama de bloques

El sistema requiere de dos comportamientos los cuales implementaremos como dos maquinas de estado. El sistema general queda de la siguiente forma:

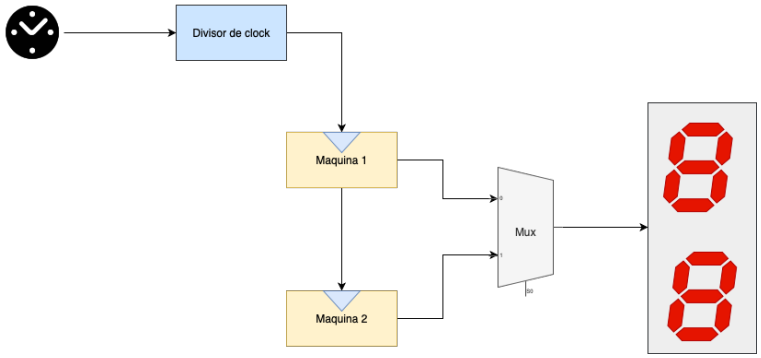


Figura 1: Diagrama de bloques

1.1. Divisor de clock

El tiempo mínimo común divisor es 1s por lo que se utilizará un divisor de clock tal que la salida sea un clock de frecuencia de 1Hz.

El divisor de clock se implementará utilizando un contador de modulo **M**, el cual posee el siguiente diagrama de hardware:

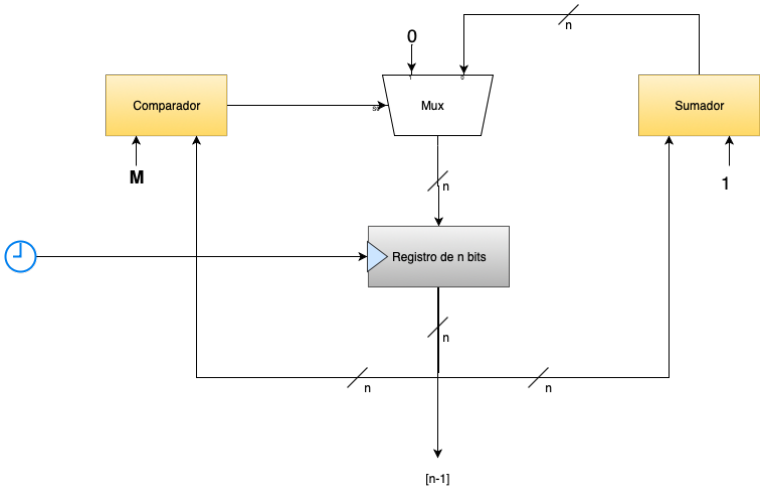


Figura 2: Diagrama de divisor de clock

La frecuencia de salida esta dada por:

$$f = \frac{f_{clk}}{\mathbf{M}}$$

por lo que necesitamos $M = 100000000$.

1.2. Maquina 1

Esta maquina comandará los semáforos durante la mañana. A partir de la tabla de tiempos podemos definir los siguientes estados:

Estado	Semáforo sur-norte	Semáforo este-oeste	Tiempo
S_0	Verde	Rojo	6 [s]
S_1	Amarillo	Rojo	2 [s]
S_2	Rojo	Verde	6 [s]
S_3	Rojo	Amarillo	2 [s]

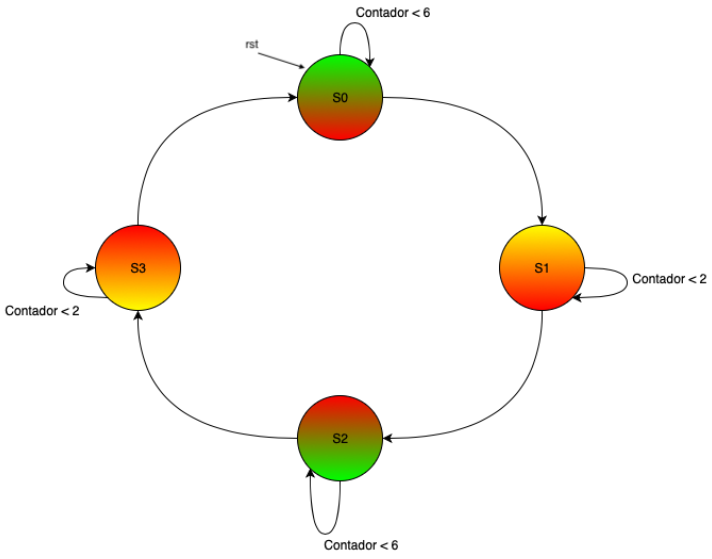


Figura 3: Diagrama de estados maquina 1

1.3. Maquina 2

Esta maquina comandará los semáforos durante la tarde. A partir de la tabla de tiempos podemos definir los siguientes estados:

Estado	Semáforo sur-norte	Semáforo este-oeste	Tiempo
S_0	Verde	Rojo	3 [s]
S_1	Amarillo	Rojo	1 [s]
S_2	Rojo	Verde	3 [s]
S_3	Rojo	Amarillo	1 [s]

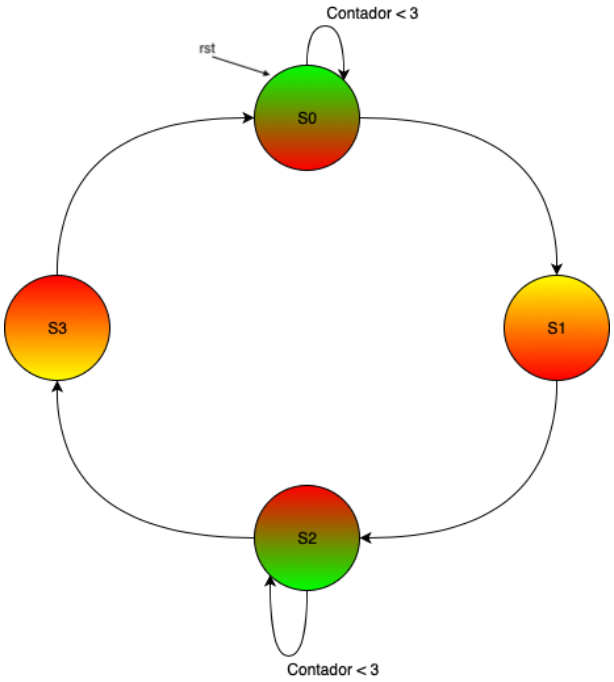


Figura 4: Diagrama de estados maquina 2

2. Códigos

2.1. Divisor de clock

```
module divisor_clk(input logic clk_in, input logic rst, output logic clk_out);

    logic [26:0] contador;

    always_ff @ (posedge clk_in or posedge rst)
    begin
        if (rst == 0)
            begin
                contador <= 0;
            end
        if (contador == 99999999)
            begin
                contador <= 0;
            end
        else
            begin
                contador <= contador + 1;
            end
    end
    assign clk_out = contador[26];

endmodule // divisor_clk
```

2.2. Maquina 1

```
module maquina1(input logic clk, input logic rst, output logic [5:0] luces);

    logic [1:0] estado;
    logic [2:0] contador;

    parameter S0 = 2'b00;
    parameter S1 = 2'b01;
    parameter S2 = 2'b10;
    parameter S3 = 2'b11;

    parameter seg6 = 3'b110;
    parameter seg2 = 3'b010;



    always_ff @(posedge clk or posedge rst)
    begin
        if (rst == 1)
            begin
                estado <= 0;
                contador <= 0;
            end
        else
            begin
                case(estado)

                    S0: if (contador < seg6)
                            begin
                                estado <= S0;
                                contador <= contador + 1;
                            end
                        else
                            begin
                                estado <= S1;
                                contador <= 0;
                            end

                    S1: if (contador < seg2)
                            begin
                                estado <= S1;
                                contador <= contador + 1;
                            end
                        else
                            begin
                                estado <= S2;
                                contador <= 0;
                            end

                    S2: if (contador < seg6)
                            begin
                                estado <= S2;
                                contador <= contador + 1;
                            end
                        else
                            begin
                                estado <= S3;
                                contador <= 0;
                            end
```

```
59
60                      S3: if (contador < seg2)
61                              begin
62                                  estado <= S3;
63                                  contador <= contador + 1;
64                              end
65                          else
66                              begin
67                                  estado <= S0;
68                                  contador <= 0;
69                              end
70                      default: estado <= S0;
71                  endcase
72              end
73      end
74
75      always_comb
76      begin
77          case(estado)
78              S0: luces = 6'b100001;
79              S1: luces = 6'b010001;
80              S2: luces = 6'b001100;
81              S3: luces = 6'b001010;
82              default: luces = 6'b100001;
83          endcase
84      end
85
86  endmodule // maquina1
```

2.3. Maquina 2

```
module maquina2(input logic clk, input logic rst, output logic [5:0] luces);

    logic [1:0] estado;
    logic [1:0] contador;

    parameter S0 = 2'b00;
    parameter S1 = 2'b01;
    parameter S2 = 2'b10;
    parameter S3 = 2'b11;

    parameter seg3 = 2'b11;

    always_ff @ (posedge clk or posedge rst)
    begin
        if (rst == 1)
            begin
                estado <= 0;
                contador <= 0;
            end
        else
            begin
                case(estado)

                    S0: if (contador < seg3)
                            begin
                                estado <= S0;
                                contador <= contador + 1;
                            end
                        else
                            begin
                                estado <= S1;
                                contador <= 0;
                            end

                    S1: estado <= S2;

                    S2: if (contador < seg3)
                            begin
                                estado <= S2;
                                contador <= contador + 1;
                            end
                        else
                            begin
                                estado <= S3;
                                contador <= 0;
                            end

                    S3: estado <= S0;

                    default: estado <= S0;
                endcase
            end
    end

    always_comb
    begin
        case(estado)
            S0: luces = 6'b100001;
```

```
59              S1: luces = 6'b010001;
60              S2: luces = 6'b001100;
61              S3: luces = 6'b001010;
62              default: luces = 6'b100001;
63          endcase
64      end
65
66  endmodule // maquina2
```

2.4. 7-Segmentos

```
module s_seg(input logic clk, input logic rst, input logic [5:0] luces
             output logic [6:0] display, output logic [3:0] an);

     logic [19:0] contador;

     always_ff @ (posedge clk or posedge rst)
     begin
        if (rst == 1)
            contador <= 0;
        else
            contador <= contador + 1;
     end

     logic s;
     assign s = contador[19];

     always_comb
     begin
        case (s)
            1'b0: begin
                     an = 4'b0111;
                     display = {~luces[5], 1, 1, ~luces[3], 1, 1, ~luces[4]};
                  end

            1'b1: begin
                     an = 4'b1110;
                     display = {~luces[2], 1, 1, ~luces[0], 1, 1, ~luces[4]};
                  end
            default: begin
                     an = 4'b0111;
                     display = {~luces[5], 1, 1, ~luces[3], 1, 1, ~luces[4]};
                  end
        endcase
     end

endmodule // s_seg
```

2.5. Main

```
module main(input logic clk, input logic rst, input logic sw0,
            output logic [6:0] display, output logic [3:0] an);

   logic clk_0;
   divisor_clk div1(clk, rst, clk_0);

   logic [5:0] luces1;
   logic [5:0] luces2;
   maquina1 m1(clk_0, rst, luces1);
   maquina2 m2(clk_0, rst, luces2);

   logic [5:0] luces;
   assign luces = sw0 ? luces2:luces1;

   s_seg semaforos(clk, rst, luces, display, an);

endmodule // main
```