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

Ayudantía # 8

Felipe Sánchez Varas

Pregunta 1

Vea el código anexo que recibe una entrada de 16 bits y muestra a la salida el número hexadecimal en el display de 7segmentos, indique a grandes rasgos para qué son cada una de las entradas y salidas. ¿Cuál es la diferencia de usar en un bloque `always` la asignación `=` versus `<=`.

Pregunta 2

Utilizando el código `x7seg.v` y `binbcd14.v` cree los modulos necesarios para realizar un contador decimal que cuente de 0 - 9999 y muestre el resultado en el display de 7seg, el contador debe cambiar a una velocidad lo más cercano posible a 50 Hz. Para ello debe implementar el divisor de clock. Muestre el resultado en la FPGA.

Pregunta 3

Utilizando el display de 7 segmentos, cree los módulos necesarios para realizar mostrar un desplazamiento del número telefónico `248-656-141`, la idea es hacer algo que para cada flanco del reloj muestre lo siguiente (considere una velocidad de movimiento suficiente para mostrar con claridad el movimiento):



Intente ahora mostrar su nombre de esta misma forma[1]

Pregunta 4

Realice en Verilog una máquina de estados que simule el funcionamiento de un semáforo (luces roja, amarilla y verde) de una intersección cuyos cambios están solo controlados por un temporizador. El diagrama de estado de la misma es el siguiente:

[1] Puede guiarse por la imagen `https://johansuarezarduino.files.wordpress.com/2013/07/alfanumericos.png`

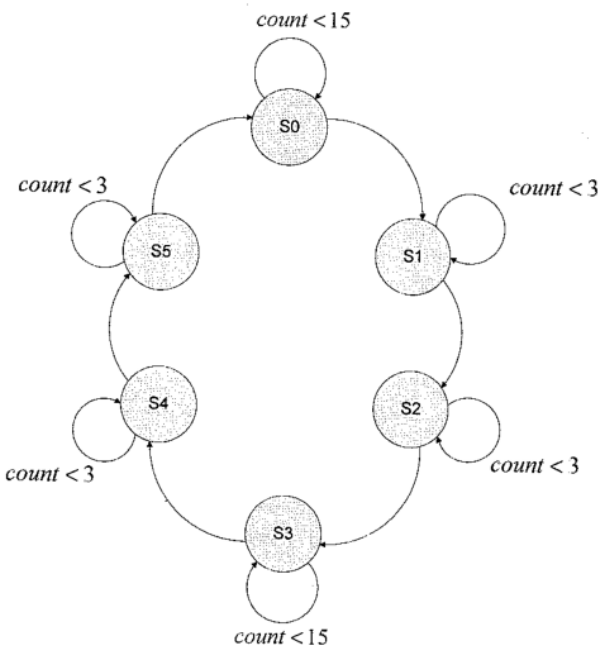


Figura 1: Semáforo pregunta pregunta 4

Propuestos

Modifique el contador de la pregunta 4, para que ahora muestre una serie de Fibonacci, ajuste la velocidad para que el cambio sea visible.

1. Anexos

1.1. x7seg.v

```
1  //////////////////////////////////////////////////////////////////////////////////
2  // Sistemas Digitales 2018-1
3  // Author:     Felipe Sanchez Varas
4  // Module Name:    x7seg
5  //
6  //////////////////////////////////////////////////////////////////////////////////
7  module x7seg(
8      input wire [15:0] x,
9      input wire clk,
10     input wire clr,
11     output reg [6:0] a_to_g,
12     output reg [3:0] an,
13     output wire dp
14     );
15
16 reg [1:0] s;
17 reg [3:0] digit;
18 wire [3:0] aen;
19
20 assign dp = 1;
21 assign aen = 4'b1111;
22
23 // Seleccionador de digito
24 always @(*)
25     case(s)
26         0: digit = x[3:0];
27         1: digit = x[7:4];
28         2: digit = x[11:8];
29         3: digit = x[15:12];
30         default: digit = x[3:0];
31     endcase
32
33 // Decodificador de 7 segmentos: hex7seg
34
35 always @(*)
36     case(digit)
37         0: a_to_g = 7'b1000000;
38         1: a_to_g = 7'b1111001;
39         2: a_to_g = 7'b0100100;
40         3: a_to_g = 7'b0110000;
41         4: a_to_g = 7'b0011001;
42         5: a_to_g = 7'b0010010;
43         6: a_to_g = 7'b0000010;
44         7: a_to_g = 7'b1111000;
45         8: a_to_g = 7'b0000000;
46         9: a_to_g = 7'b0010000;
47         'hA: a_to_g = 7'b0001000;
48         'hb: a_to_g = 7'b0000011;
49         'hC: a_to_g = 7'b1000110;
50         'hd: a_to_g = 7'b0100001;
51         'hE: a_to_g = 7'b0000110;
52         'hF: a_to_g = 7'b0001110;
53         default: a_to_g = 7'b1000000; // 0
54     endcase
55
56 // Seleccionador de numero
57 always @(*)
58     begin
59         an = 4'b1111;
60         if(aen[s] == 1)
61             an[s] = 0;
```

```
62          end
63
64  // 2-bit counter
65  always @(posedge clk or posedge clr)
66      begin
67          if(clr == 1)
68              s <= 0;
69          else
70              s <= s + 1;
71      end
72
73  endmodule
```

1.2. bincbd14.v

```
1   //////////////////////////////////////////////////////////////////////////////////
2   // Sistemas Digitales 2018-1
3   // Author:     Felipe Sanchez Varas
4   // Module Name:    binbcd14
5   //
6   //////////////////////////////////////////////////////////////////////////////////
7
8   module binbcd14(
9       input wire [13:0] b,
10      output reg [16:0] q
11  );
12
13  // variables intermedias
14  reg [32:0] z;
15  integer i;
16
17  always @(*)
18      begin
19          for (i = 0; i <= 32; i = i+1)
20              z[i] = 0;
21          z[16:3] = b;
22
23          repeat(11)
24              begin
25                  if(z[17:14] > 4)
26                      z[17:14] = z[17:14] + 3;
27                  if(z[21:18] > 4)
28                      z[21:18] = z[21:18] + 3;
29                  if(z[25:22] > 4)
30                      z[25:22] = z[25:22] + 3;
31                  if(z[29:26] > 4)
32                      z[29:26] = z[29:26] + 3;
33                  z[32:1] = z[31:0];
34              end
35          q = z[30:14];
36      end
37  endmodule
```