Universidad de las Américas Puebla



Sistemas Embebidos

Tarea 2: Informe Tecnico

Emilio Iván Jiménez López 179543

Planteamiento

En esta segunda tarea, realizaremos 2 circuitos secuenciales, ambos con el propósito de realizar un corrimiento a la derecha de un registro de bits, sin embargo, ambos tendrán un funcionamiento y complejidad muy diferentes.

El primer problema tendrá entrada y salida de registros paralelos de 8 bits, deberá además incluir un input para carga/corrimiento. El segundo problema es similar, sin embargo, deberá producir un output serial, no paralela, y de 4 bits en lugar de 8. Para ello, se deberán incluir también una señal de inicio de conversión, y una señal también de data_ready para señalizar la conclusión del proceso.

Ambos deberán incluir un reloj y activación en el flanco positivo del reloj, un reset asíncrono, y se deberán hacer los testbenches correspondientes.

Procedimiento

Previo a la implementación en Verilog, se realizaron trabajos de prototipado en papel, los cuales pueden verse bajo solicitud al estudiante.

La implementación del problema 1 se ve de la siguiente forma:

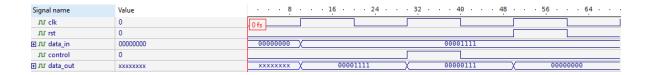
La implementación del problema 2 se ve de la siguiente forma:

```
//Problema 2: Registro paralelo-serial
35
    module registro_paralelo_serie (
                                       // Señal de reloj
36
         input wire clk,
37
38
         input wire rst,
                                       // Reset asincrono
         input wire start,
                                     // Señal de inicio de conversión
39
        output reg data_out,
         input wire [3:0] data_in,
                                     // Entrada paralela de 4 bits
40
                                       // Salida serial de 1 bit
41
                                      // Señal de conversión completa
42
    );
43
44
         // Registros internos
45
        reg [3:0] shift_reg;
reg [2:0] bit_count;
                                       // Registro de desplazamiento
46
                                       // Contador de bits enviados
47
         reg sending;
                                       // Estado de transmisión
48
49
         // Ciclo activado por flanco positivo del reloj o estimulo reset
50
         always @(posedge clk or posedge rst) begin
51
             if (rst) begin
52
                 // Reinicio de todos los registros
53
                 shift reg <= 4'b0000;</pre>
54
                 bit count <= 3'b000;
55
                 data out <= 1'b0;
56
                 data ready <= 1'b0;
57
                 sending <= 1'b0;
58
             end else begin
59
                 if (start && !sending) begin
60
                      // Inicio de transmisión: carga de datos
61
                     shift_reg <= data_in;
62
                     bit count <= 3'b000;
63
                     sending <= 1'b1;
64
                     data ready <= 1'b0;
65
                 end else if (sending) begin
66
                     // Transmision bit a bit
67
                     data_out <= shift_reg[bit_count];</pre>
68
                     bit_count <= bit_count + 1;
69
70
                      if (bit_count == 3) begin
71
                          // Ultimo bit transmitido
72
                          sending <= 1'b0;
73
                          data_ready <= 1'b1;</pre>
74
                      end
75
                 end else begin
76
                      // Limpieza de registro de listo cuando no se transmite
77
                      data_ready <= 1'b0;</pre>
78
                 end
79
             end
         end
    endmodule
```

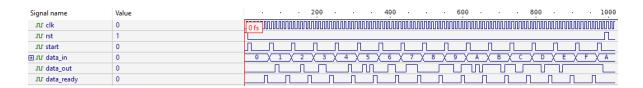
La implementación de los testbenches se realizo con apoyo de herramientas de inteligencia artificial.

Resultados

Los resultados del testbench del problema 1 son los siguientes:



Los resultados del testbench del problema 2 son los siguientes:



Análisis

Tras revisar los resultados de los testbenches, es claro que los ejemplos aplicados efectivamente dan resultados adecuados, por ejemplo, en el caso del circuito con salida paralela, 00001111 nos da como resultado 00000111, mientras que en el circuito secuencial es un poco mas complicado revisarlo, sin embargo, también se puede notar que efectivamente, la salida es adecuada.

Conclusiones

Esta tarea explora temas que en clase no habíamos podido revisar adecuadamente, por tanto represento una gran oportunidad para aprender bastante acerca de el lenguaje Verilog, sus capacidades, el paradigma de programación para descripción de hardware, y en lo personal, me sirvió para aplicar mis habilidades programáticas al paradigma de los lenguajes de descripción.