Lab Experiment #6

ECE 282 - 002

Friday PM Lab

Carlos Sanchez and Connor Raasch

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer: Connor Raasch

//

// Create Date: 13:44:00 11/16/2018

// Design Name:

// Module Name: Adder

// Project Name: Lab 6 part 1a

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module Adder(S, Cout, A, B, Cin);

output reg [3:0] S;

output reg [1:0] Cout;

input [3:0] A, B;

input [1:0] Cin;

always @ (A, B, Cin)

{Cout, S} = A + B + Cin;

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer: Carlos Sanchez

//

// Create Date: 13:50:00 11/16/2018

// Design Name:

// Module Name: Adder

// Project Name: Lab 6 part 1

// Target Devices:

// Tool versions:

// Description: 282 - 002

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module Adder(S, Cout, A, B, Cin);

output reg [3:0] S;

output reg [1:0] Cout;

input [3:0] A, B;

input [1:0] Cin;

always @ (A, B, Cin)

{Cout, S} = A + B + Cin;

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer: Connor Raasch

//

// Create Date: 13:54:41 11/16/2018

// Design Name:

// Module Name: Adder\_Subtractor

// Project Name: Lab 6

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module Adder\_Subtractor(S, Cout, A, B, Cin);

output reg Cout;

output reg [3:0] S;

input Cin;

input [3:0] A, B;

always @ (A, B, Cin)

if (Cin == 0)

{Cout, S} = A + B;

else

{Cout, S} = A - B;

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer: Carlos Sanchez

//

// Create Date: 13:59:59 11/16/2018

// Design Name:

// Module Name: Adder\_Subtractor

// Project Name: Lab 6 Part 2

// Target Devices:

// Tool versions:

// Description: ECE 282 - 002

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module Adder\_Subtractor(S, Cout, A, B, Cin);

output reg Cout;

output reg [3:0] S;

input Cin;

input [3:0] A, B;

always @ (A, B, Cin)

if (Cin == 0)

{Cout, S} = A + B;

else

{Cout, S} = A - B;

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer: Connor Raasch

//

// Create Date: 14:12:46 11/16/2018

// Design Name:

// Module Name: Adder\_subtractor

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module Adder\_subtractor(S, C, V, A, B, op);

output [3:0] S;

output C;

output V;

input [3:0] A, B;

input op;

wire f0, f1, f2, f3, b0, b1, b2, b3;

xor(b0, B[0], op);

xor(b1, B[1], op);

xor(b2, B[2], op);

xor(b3, B[3], op);

xor(C, f3, op);

xor(V, f3, f2);

full\_adder fa0(S[0], f0, A[0], b0, op);

full\_adder fa1(S[1], f1, A[1], b1, f0);

full\_adder fa2(S[2], f2, A[2], b2, f1);

full\_adder fa3(S[3], f3, A[3], b3, f2);

endmodule

module full\_adder(S, Cout, A, B, Cin);

output S;

output Cout;

input A;

input B;

input Cin;

wire w1;

wire w2;

wire w3;

wire w4;

xor(w1, A, B);

xor(S, Cin, w1);

and(w2, A, B);

and(w3, A, Cin);

and(w4, B, Cin);

or(Cout, w2, w3, w4);

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer: Carlos Sanchez

//

// Create Date: 14:15:32 11/16/2018

// Design Name:

// Module Name: Adder\_subtractor

// Project Name:

// Target Devices:

// Tool versions:

// Description: ECE 282 - 002

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module Adder\_subtractor(S, C, V, A, B, op);

output [3:0] S;

output C;

output V;

input [3:0] A, B;

input op;

wire f0, f1, f2, f3, b0, b1, b2, b3;

xor(b0, B[0], op);

xor(b1, B[1], op);

xor(b2, B[2], op);

xor(b3, B[3], op);

xor(C, f3, op);

xor(V, f3, f2);

full\_adder fa0(S[0], f0, A[0], b0, op);

full\_adder fa1(S[1], f1, A[1], b1, f0);

full\_adder fa2(S[2], f2, A[2], b2, f1);

full\_adder fa3(S[3], f3, A[3], b3, f2);

endmodule

module full\_adder(S, Cout, A, B, Cin);

output S;

output Cout;

input A;

input B;

input Cin;

wire w1;

wire w2;

wire w3;

wire w4;

xor(w1, A, B);

xor(S, Cin, w1);

and(w2, A, B);

and(w3, A, Cin);

and(w4, B, Cin);

or(Cout, w2, w3, w4);

endmodule

`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer: Connor Raasch

//

// Create Date: 14:21:43 11/16/2018

// Design Name: Adder\_subtractor

// Module Name: E:/Lab6/Part2/Adder\_subtracotr\_tb.v

// Project Name: Part2

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: Adder\_subtractor

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

////////////////////////////////////////////////////////////////////////////////

module Adder\_subtracotr\_tb;

// Inputs

reg [3:0] A;

reg [3:0] B;

reg op;

// Outputs

wire [3:0] S;

wire C;

wire V;

// Instantiate the Unit Under Test (UUT)

Adder\_subtractor uut(S, C, V, A, B, op);

initial begin

// Initialize Inputs

op = 0;

A = 3;

B = 2;

#20 // 5

A = 5;

B = 4;

#20 // 9

A = 3;

B = 3;

#20 // 6

A = 15;

B = 9;

#20 // 24

op = 1;

A = 6;

B = 5;

#20 // 1

A = 7;

B = 2;

#20 // 5

A = 11;

B = 4;

// 7

#20; $finish;

end

endmodule

`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer: Carlos Sanchez

//

// Create Date: 14:23:22 11/16/2018

// Design Name: Adder\_subtractor

// Module Name: E:/Lab6/Part2/Adder\_subtracotr\_tb.v

// Project Name: Part2

// Target Device:

// Tool versions:

// Description: ECE 282 - 002

//

// Verilog Test Fixture created by ISE for module: Adder\_subtractor

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

////////////////////////////////////////////////////////////////////////////////

module Adder\_subtracotr\_tb;

// Inputs

reg [3:0] A;

reg [3:0] B;

reg op;

// Outputs

wire [3:0] S;

wire C;

wire V;

// Instantiate the Unit Under Test (UUT)

Adder\_subtractor uut(S, C, V, A, B, op);

initial begin

// Initialize Inputs

op = 0;

A = 3;

B = 2;

#20 // 5

A = 5;

B = 4;

#20 // 9

A = 3;

B = 3;

#20 // 6

A = 15;

B = 9;

#20 // 24

op = 1;

A = 6;

B = 5;

#20 // 1

A = 7;

B = 2;

#20 // 5

A = 11;

B = 4;

// 7

#20; $finish;

end

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer: Connor Raasch

//

// Create Date: 15:08:52 11/16/2018

// Design Name:

// Module Name: State\_Diagram

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module State\_Diagram(out, in, clk, rst);

output out;

reg [1:0] state, next;

input in, clk, rst;

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

assign out = in ^ state[0] ^ state[1];

always @ (posedge clk, negedge rst)

if (!rst) state <= S0;

else state <= next;

always @ (state, in)

case(state)

S0: if(in) next = S0; else next = S1;

S1: if(in) next = S3; else next = S2;

S2: if(in) next = S1; else next = S0;

S3: if(in) next = S2; else next = S3;

endcase

Endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer: Carlos Sanchez

//

// Create Date: 15:12:12 11/16/2018

// Design Name:

// Module Name: State\_Diagram

// Project Name:

// Target Devices:

// Tool versions:

// Description: ECE 282 - 002

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module State\_Diagram(out, in, clk, rst);

output out;

reg [1:0] state, next;

input in, clk, rst;

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

assign out = in ^ state[0] ^ state[1];

always @ (posedge clk, negedge rst)

if (!rst) state <= S0;

else state <= next;

always @ (state, in)

case(state)

S0: if(in) next = S0; else next = S1;

S1: if(in) next = S3; else next = S2;

S2: if(in) next = S1; else next = S0;

S3: if(in) next = S2; else next = S3;

endcase

Endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer: Connor Raasch

//

// Create Date: 15:40:46 11/16/2018

// Design Name:

// Module Name: bianary\_Counter

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//////////////////////////////////////////////////////////////////////////////////

module bianary\_Counter(outA, outB, in\_A, in\_B, in\_x, clk, rst);

output outA, outB;

input in\_A, in\_B, in\_x, clk, rst;

wire w1, w2;

xor(w1, in\_A, in\_B, in\_x);

not(w2, w1);

DFF Aff(outA, clk, rst, in\_B);

DFF Bff(outB, clk, rst, w2);

endmodule

module DFF (out, clk, rst, D);

output reg out;

input clk, rst, D;

always @ (posedge clk)

if (rst == 1)

out = 0;

else

out = D;

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer: Carlos Sanchez

//

// Create Date: 15:44:50 11/16/2018

// Design Name:

// Module Name: Binary\_Counter

// Project Name:

// Target Devices:

// Tool versions:

// Description: ECE 282 - 002

// Dependencies:

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//////////////////////////////////////////////////////////////////////////////////

module Binary\_Counter(outA, outB, in\_A, in\_B, in\_x, clk, rst);

output outA, outB;

input in\_A, in\_B, in\_x, clk, rst;

wire w1, w2;

xor(w1, in\_A, in\_B, in\_x);

not(w2, w1);

DFF Aff(outA, clk, rst, in\_B);

DFF Bff(outB, clk, rst, w2);

endmodule

module DFF (out, clk, rst, D);

output reg out;

input clk, rst, D;

always @ (posedge clk)

if (rst == 1)

out = 0;

else

out = D;

endmodule