Daniel Delgado Acosta   
Andrew Goldbaum   
Eric Duong   
3/24/2022

Lab 7: Designing a Full-Adder with Verilog

Objective: To build a full adder at the gate level, design and test a full-adder with Verilog.  
Provided: TTL7486(1) (XOR), TTL7408 (1) (AND) (1), 7432 (1) (OR), digital trainer  
kit.

Diagram

Description automatically generatedFull Adder Logic Circuit:

A picture containing table

Description automatically generatedFull Adder Truth Table:

A screenshot of a computer

Description automatically generatedHalf Adder Screenshot:

HalfAdder.v Code:

`timescale 1ns / 1ps

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

// Company: CSUSB

// Engineer: Haiyan Qiao, Sean A. Finucane

//

// Create Date: 08:19:36 09/29/2012

// Design Name: Gate-Level Half-Adder (200X)

// Module Name: half\_adder

// Project Name: CSE 310 LAB

// Target Devices: NONE

// Tool versions: XILINX ISE (13.4)

// Description:

//

// Dependencies:

//

// Revision: 1.0 - First Functional Release

// Revision 0.01 - File Created

// Additional Comments:

//

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

module half\_adder(

input wire x,

input wire y,

output wire S,

output wire C

);

assign S = x ^ y; // XOR Gate

assign C = x & y; // AND Gate

endmodule

Test HalfAdder.v Code:

`timescale 1ns / 1ps

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

// Company:

// Engineer:

//

// Create Date:   11:12:05 03/22/2022

// Design Name:   half\_adder

// Module Name:   C:/Users/006896598/FullAdder1/test\_halfadder1.v

// Project Name:  FullAdder1

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: half\_adder

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

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

module test\_halfadder1;

// Inputs

reg x;

reg y;

// Outputs

wire S;

wire C;

// Instantiate the Unit Under Test (UUT)

half\_adder uut (

.x(x),

.y(y),

.S(S),

.C(C)

);

initial begin

// Initialize Inputs

x = 0;

y = 0;

#2

x = 1;

y = 0;

#2

x = 0;

y = 1;

#2

x = 1;

y = 1;

#100 ;

// Wait 100 ns for global reset to finish

// Add stimulus here

end

endmodule

A screenshot of a computer

Description automatically generatedFull Adder Screenshot

FullAdder.v Code:

`timescale 1ns / 1ps

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

// Company: CSUSB

// Engineer: Haiyan Qiao, Sean A. Finucane

//

// Create Date: 08:29:08 09/29/2012

// Design Name: Full-Adder using Half-Adders, Gate-Level (200X)

// Module Name: full\_adder

// Project Name: CSE 310 LAB

// Target Devices: NONE

// Tool versions: XILINX ISE (13.4)

// Description:

//

// Dependencies: half\_adder

//

// Revision: 1.0 - First Functional Release

// Revision 0.01 - File Created

// Additional Comments:

//

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

module full\_adder(

    input wire x,

    input wire y,

    input wire z,

    output wire S,

    output wire C

    );

// signal interconnects:

wire S0, C0, C1;

// internal module declarations (sub-modules):

half\_adder ha\_0( .x(x), .y(y), .S(S0), .C(C0) ); // FIRST HALF-ADDER

half\_adder ha\_1( .x(S0), .y(z), .S(S), .C(C1) ); // SECOND HALF\_ADDER

// output assignments (where still needed):

assign C = C0 | C1;

endmodule

Test FullAdder.v Code:

`timescale 1ns / 1ps

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

// Company:

// Engineer:

//

// Create Date:   11:17:06 03/22/2022

// Design Name:   full\_adder

// Module Name:   C:/Users/006896598/FullAdder1/test\_fulladder1.v

// Project Name:  FullAdder1

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: full\_adder

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

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

module test\_fulladder1;

// Inputs

reg x;

reg y;

reg z;

// Outputs

wire S;

wire C;

// Instantiate the Unit Under Test (UUT)

full\_adder uut (

.x(x),

.y(y),

.z(z),

.S(S),

.C(C)

);

initial begin

// Initialize Inputs

x = 0;

y = 0;

z = 0;

      #2

x =0;

y =0;

z =1;

#2

x =0;

y =1;

z =0;

#2

x =0;

y =1;

z =1;

#2

x =1;

y =0;

z =0;

#2

x =1;

y =0;

z =1;

#2

x =1;

y =1;

z =0;

#2

x =1;

y =1;

z =1;

#2

// Wait 100 ns for global reset to finish

#100;

// Add stimulus here

end

endmodule