# Lab 7 Introduction to Verilog

**Objective**

The objective of this lab is to learn the techniques of Verilog Design and Simulation for the design of half adder and full adder. Your job is to design the circuit using Verilog description and simulate using a test bench.

**Introduction**

In this lab assignment, you will design and test a full adder using verilog. We will focus on building the design from the bottom up, giving you practice in hierarchical design. We will start with the design of a half adder, then use two half adders to build a full adder.

• Half-adders in terms of gates.

• Full-adders in terms of half-adders

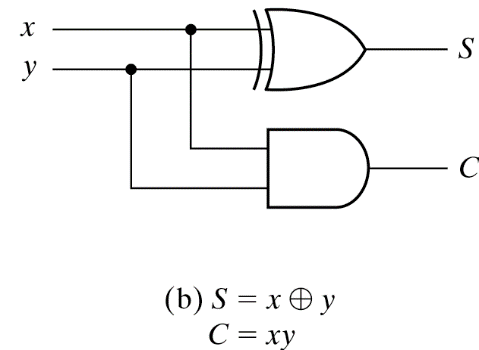
**Lab Procedure**

1. **Build a Half Adder**:

Below is the circuit and Boolean equations for a Half Adder. In this part, we will design the half adder structurally using basic logic gates, exactly following the topology of the circuit diagram. Edit the definition of halfadder to make it **purely structural** as follows. A skeleton of the Verilog description is provided, with some details missing. ***Your task is to fill in the blanks*.** Be sure that your Verilog description exactly matches the circuit below.

**module** **halfadder** (S, C, x, y);

U1



**input** x, y;

**output** S, C;

//Instantiate primitive gates

U2

**xor** U1(S, x, y);

**and** U2(C, x, y);

**endmodule**

For your information, you can alternatively use **dataflow modelling** by expressing each output as a Boolean equation in Verilog:

assign S = x^y;

assign C = x&y;

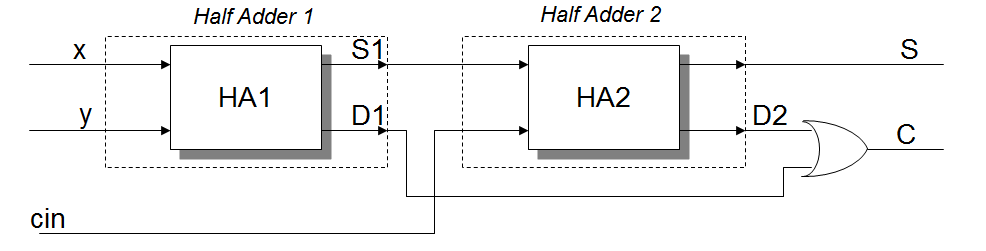
or you can **use behavioral modelling**

assign {C, S} = x+y;

1. **Build a Full Adder using two Half Adders and an OR gate.**

Again, a skeleton of the Verilog description of the full adder is provided, with some details erased. ***Your task is to fill in the blanks***. Be sure that your Verilog description exactly matches the circuit below.

**module** **fulladder** (S, C, x, y, cin);



**input** x, y, cin;

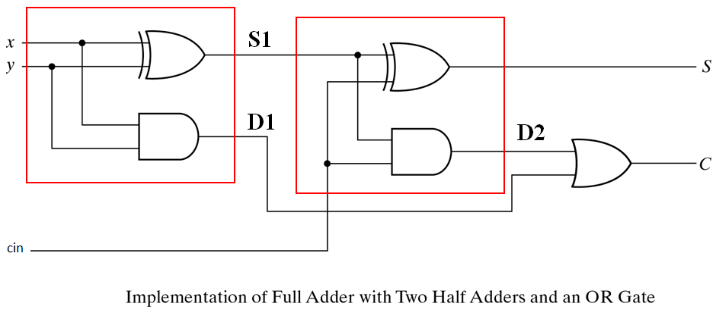
**output** S, C;

U3

//internal signals

**wire** S1, D1, D2;

//Instantiate the half adders



**halfadder** HA1 (S1, D1, x, y);

**halfadder** HA2 (S, D2, S1, cin);

U3

**or** U3(C, D2, D1);

**endmodule**

1. **Test Bench**: Write a testbench program to test the full adder. Add your **statement (all other test cases)** afterthe comment **“**// Initialize Inputs”

**module** test;

**reg** x, y, cin; // Inputs

**wire** S, C; // Outputs

// Instantiate the Unit Under Test (UUT)

fulladder uut (S, C, x, y, cin);

**initial**

**begin**

//the following 2 statements are needed for producing waveform in edaplayground.com

**$dumpfile("dump.vcd"); $dumpvars(1, test);**

// display the inputs and outputs

**$monitor**( "x + y + cin = %b + %b + %b = C S = %b %b", x, y, cin, C, S );

// Initialize Inputs

x = 0; y = 0; cin = 0;

#10 x = 0; y = 0; cin = 1;

#10 x = 0; y = 0; cin = 0;

#10 x = 0; y = 1; cin = 0;

#10 x = 0; y = 1; cin = 1;

#10 x = 1; y = 0; cin = 0;

#10 x = 1; y = 0; cin = 1;

#10 x = 1; y = 1; cin = 0;

#10 x = 1; y = 1; cin = 1;

#10 **$finish;**

**end**

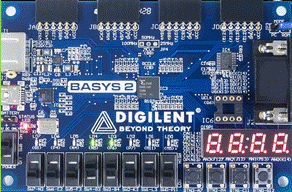
**endmodule**

1. **Test on EDAplayground.com**

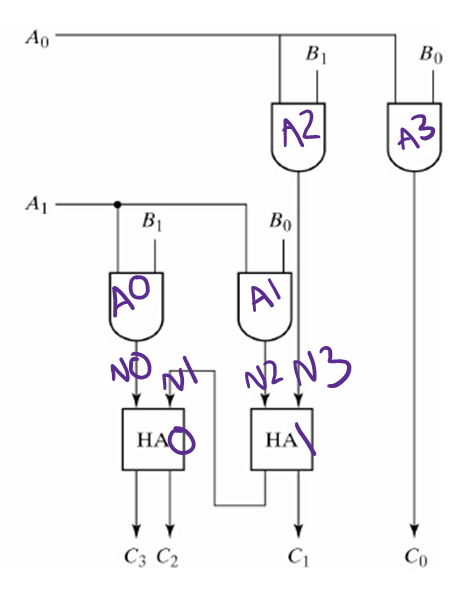
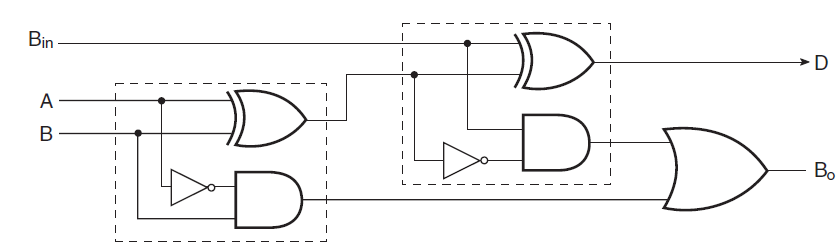
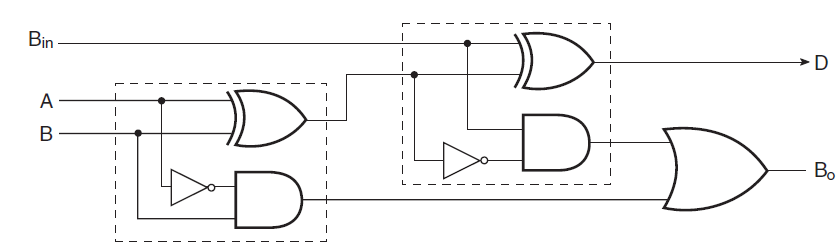
* Log into your EDAplayground.com account.
* Edit your Verilog design code and testbench code in the right and left windows respectively, enter a name such as “Lab 8” for your project in the edit box at the bottom and click the “Save**”** button to save your project.
* On the left panel, under Tools and Simulations, select Icarus Verilog 0.9.7 and check the box of “Open **EPwave** after Run”
* Click “Run” at the top to run the simulation, watch for waveforms and verify your full adder works correctly.

1. **Pin Assignments, Synthesis and FPGA board programming**

* not able do online



1. **Homework: write Verilog design and test bench codes for a 2-bit multiplier based on the following diagram**



**module test;**

**reg A0, A1, B0, B1; // Inputs**

**wire C0, C1, C2, C3; // Outputs**

**// Instantiate the Unit Under Test (UUT)**

**multiplier uut (A0, A1, B0, B1, C0, C1, C2, C3);**

**initial**

**begin**

**//the following 2 statements are needed for producing waveform in edaplayground.com**

**$dumpfile("dump.vcd"); $dumpvars(1, test);**

**// display the inputs and outputs**

**$monitor("%b %b %b %b %b %b %b %b", A0, A1, B0, B1, C0, C1, C2, C3);**

**// this for loop auto inputs for me woooo**

**for(int i = 0; i < 16; i = i + 1) begin**

**{A0, A1, B0, B1} = i;**

**#10; // 10 delay units**

**end**

**#10 $finish;**

**end**

**endmodule**

**// Code your design here**

**module multiplier(A0, A1, B0, B1, C0, C1, C2, C3);**

**input A0, A1, B0, B1;**

**output C0, C1, C2, C3;**

**//internal signals (labeled on diagram)**

**wire N0, N1, N2, N3;**

**//Instantiate the half adders**

**halfadder HA0 (C3,C2,N0,N1);**

**halfadder HA1 (C1,N1,N2,N3);**

**and A0(N0, A1, B1);**

**and A1(N2, A1, B0);**

**and A2(N3, A0, B1);**

**and A3(C0, A0, B0);**

**endmodule**

**module halfadder (S, C, x, y);**

**input x, y;**

**output S, C;**

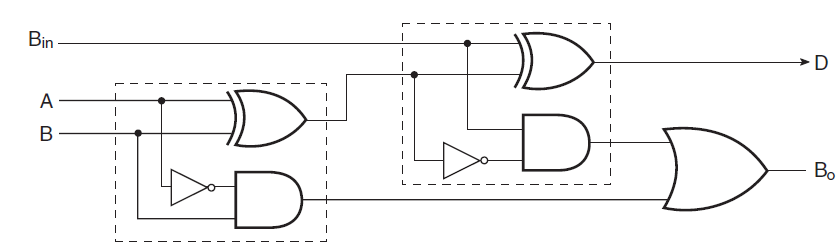
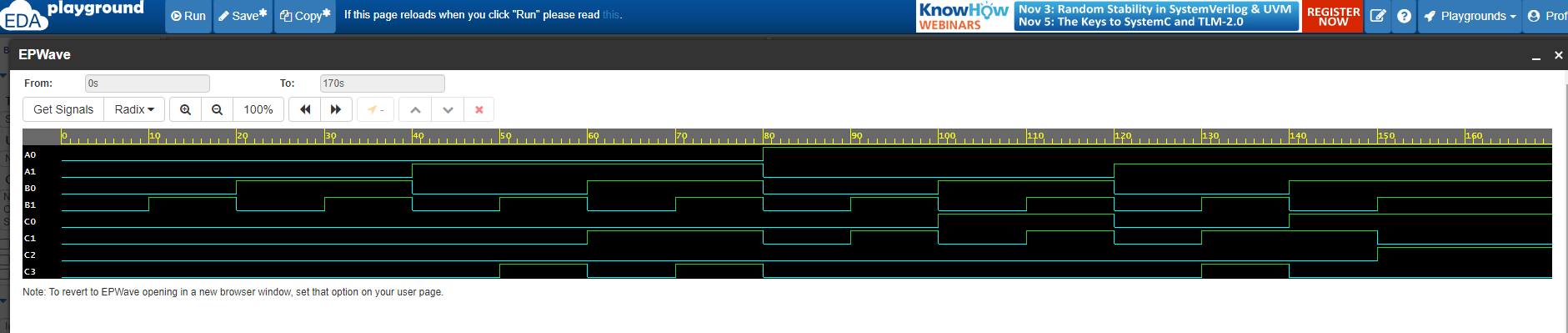
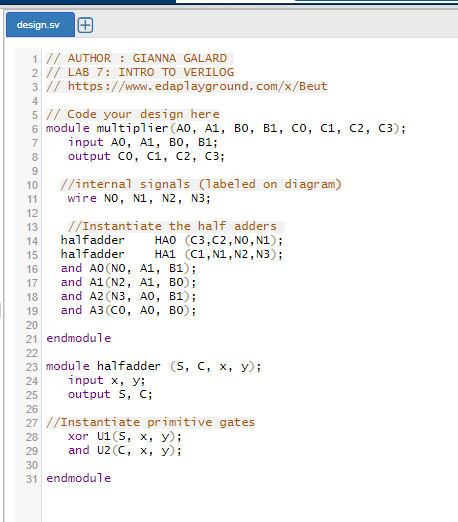
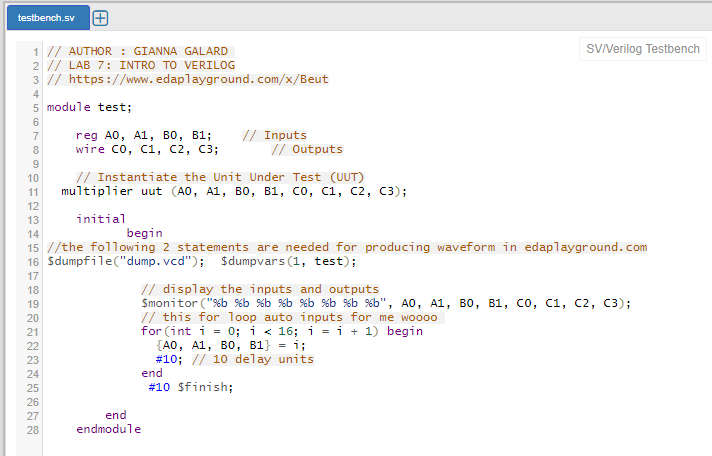
**//Instantiate primitive gates**

**xor U1(S, x, y);**

**and U2(C, x, y);**

**endmodule**

[**https://www.edaplayground.com/x/Beut**](https://www.edaplayground.com/x/Beut)



**Submission Instructions:**

Lab work submission

1. Take a screenshot of your waveforms.
2. Add the following information as comments to the beginning of your code. Make sure to click the “Save” button to save your project, then take a screenshot of your code.

// Author: Name

// Lab 7: Introduction to Verilog

// Link to your project

1. Copy the link of your design from the address bar of the browser.
2. On the Blackboard, click on Lab 7. Attach the screenshot from the first two steps and paste the link from Step 3 into the Comments area, then click the “Submit” button.

Lab report submission

1. Lab report is needed for this lab. Please follow the guidelines and sample report on the Blackboard when you are writing your lab report. Click on **Lab 7 Report Submission** to submit your report. It is due one week after the lab is done.