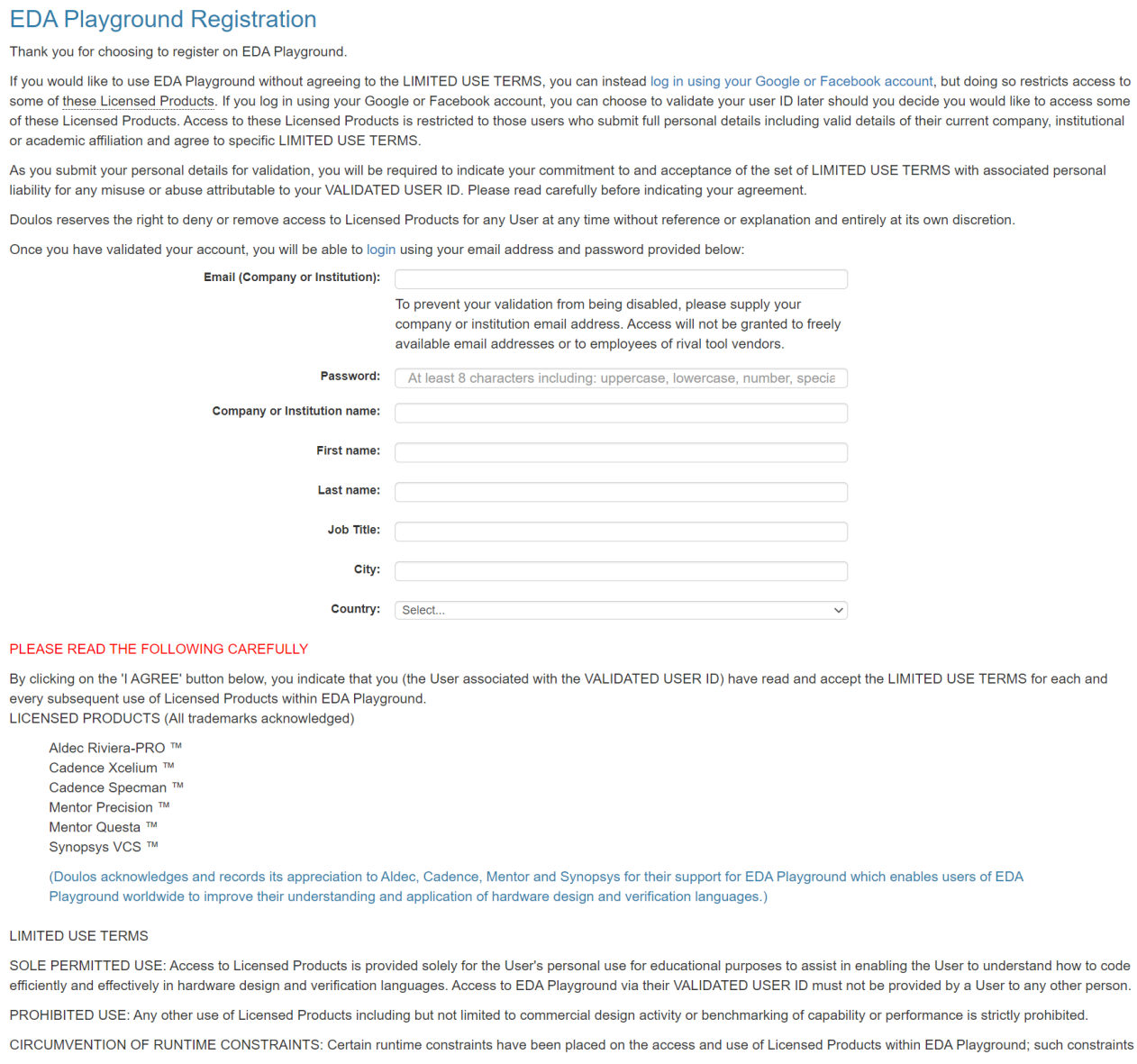
LABORATORY # 0

Intro to EDA Playground

EDA Playground Setup

Creating your account:

Create your EDA Playground account https://www.edaplayground.com/register
The link will direct you to the following webpage



EDA Playground Registration

Thank you for choosing to register on EDA Playground.

If you would like to use EDA Playground without agreeing to the LIMITED USE TERMS, you can instead log in using your Google or Facebook account, but doing so restricts access to some of these Licensed Products. If you log in using your Google or Facebook account, you can choose to validate your user ID later should you decide you would like to access some of these Licensed Products. Access to these Licensed Products is restricted to those users who submit full personal details including valid details of their current company, institutional or academic affiliation and agree to specific LIMITED USE TERMS.

As you submit your personal details for validation, you will be required to indicate your commitment to and acceptance of the set of LIMITED USE TERMS with associated personal liability for any misuse or abuse attributable to your VALIDATED USER ID. Please read carefully before indicating your agreement.

Doulos reserves the right to deny or remove access to Licensed Products for any User at any time without reference or explanation and entirely at its own discretion.

Once you have validated your account, you will be able to login using your email address and password provided below:

Email (Company or Institution):

To prevent your validation from being disabled, please supply your company or institution email address. Access will not be granted to freely available email addresses or to employees of rival tool vendors.

Password: At least 8 characters including: uppercase, lowercase, number, specia

Company or Institution name:

First name:

Last name:

Job Title:

City:

Country: Select...

PLEASE READ THE FOLLOWING CAREFULLY

By clicking on the 'I AGREE' button below, you indicate that you (the User associated with the VALIDATED USER ID) have read and accept the LIMITED USE TERMS for each and every subsequent use of Licensed Products within EDA Playground.
LICENSED PRODUCTS (All trademarks acknowledged)

Aldec Riviera-PRO ™
Cadence Xcelium ™
Cadence Specman ™
Mentor Precision ™
Mentor Questa ™
Synopsys VCS ™

(Doulos acknowledges and records its appreciation to Aldec, Cadence, Mentor and Synopsys for their support for EDA Playground which enables users of EDA Playground worldwide to improve their understanding and application of hardware design and verification languages.)

LIMITED USE TERMS

SOLE PERMITTED USE: Access to Licensed Products is provided solely for the User's personal use for educational purposes to assist in enabling the User to understand how to code efficiently and effectively in hardware design and verification languages. Access to EDA Playground via their VALIDATED USER ID must not be provided by a User to any other person.

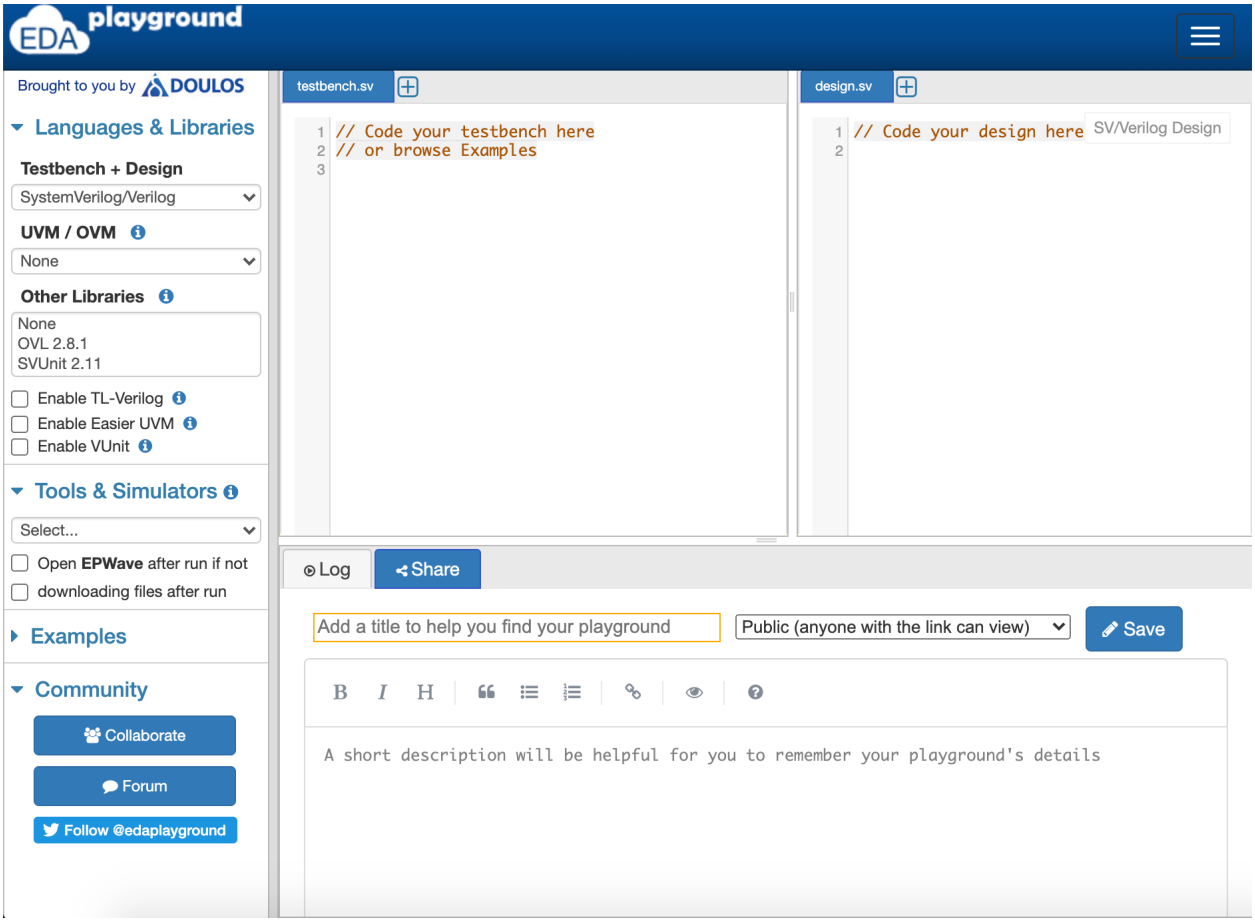
PROHIBITED USE: Any other use of Licensed Products including but not limited to commercial design activity or benchmarking of capability or performance is strictly prohibited.

CIRCUMVENTION OF RUNTIME CONSTRAINTS: Certain runtime constraints have been placed on the access and use of Licensed Products within EDA Playground; such constraints

Proceed with creating your account (**make sure to use your UCR email address**).

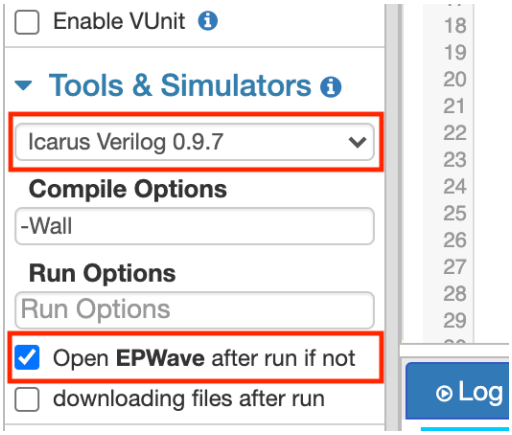
After you register, go back to the EDA playground website if you haven't been automatically re-directed there: https://www.edaplayground.com/

You will now see the following user interface (UI)



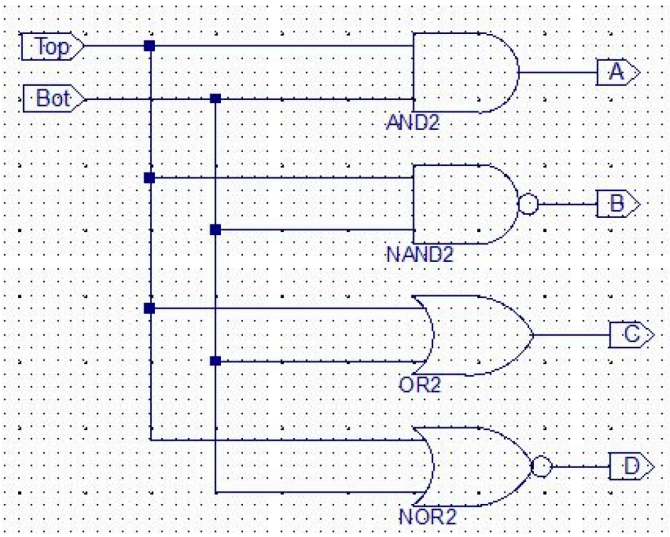
If you do not see this, make sure your account has been successfully created and you are logged in.

Setting up the EDA Playground UI:
Under “Tools & Simulators” choose Icarus Verilog 0.9.7 and check “Open EPWave after run”



Lab exercise

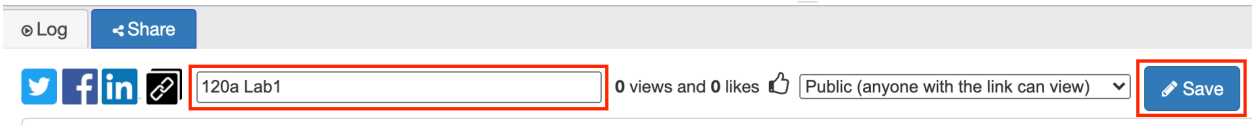
Design:
In this lab, you will be implementing the following design using Verilog.
You will also be creating a Verilog testbench to test the functionality of the design and verify that it is producing the expected results.



The Verilog code for the structural implementation of this circuit is given below.
Copy this to the "SV/Verilog Design" window of your EDA playground UI.

```
`timescale 1ns / 1ps

module lab1_structural(
  // Ports I/O
  input wire Bot,
  input wire Top,
  output wire A,
  output wire B,
  output wire C,
  output wire D
);

  // Place gates and connect I/O
  // Format: gate_type arbitrary_label (output, input1, input2)
  and gate1 (A, Top, Bot);
  nand gate2 (B, Top, Bot) ;
  or gate3 (C, Top, Bot) ;
  nor gate4 (D, Top, Bot) ;

endmodule
```

Make sure to name your project and hit “Save” (Make sure you save periodically).



Before proceeding , look back at the Verilog code and the schematic diagram given in the previous page. Even with limited background in Verilog, you should be able to see how the code describes the given circuit. Use the “Intro to FPGAs and HDLs” slides for reference if needed.

Testbench:

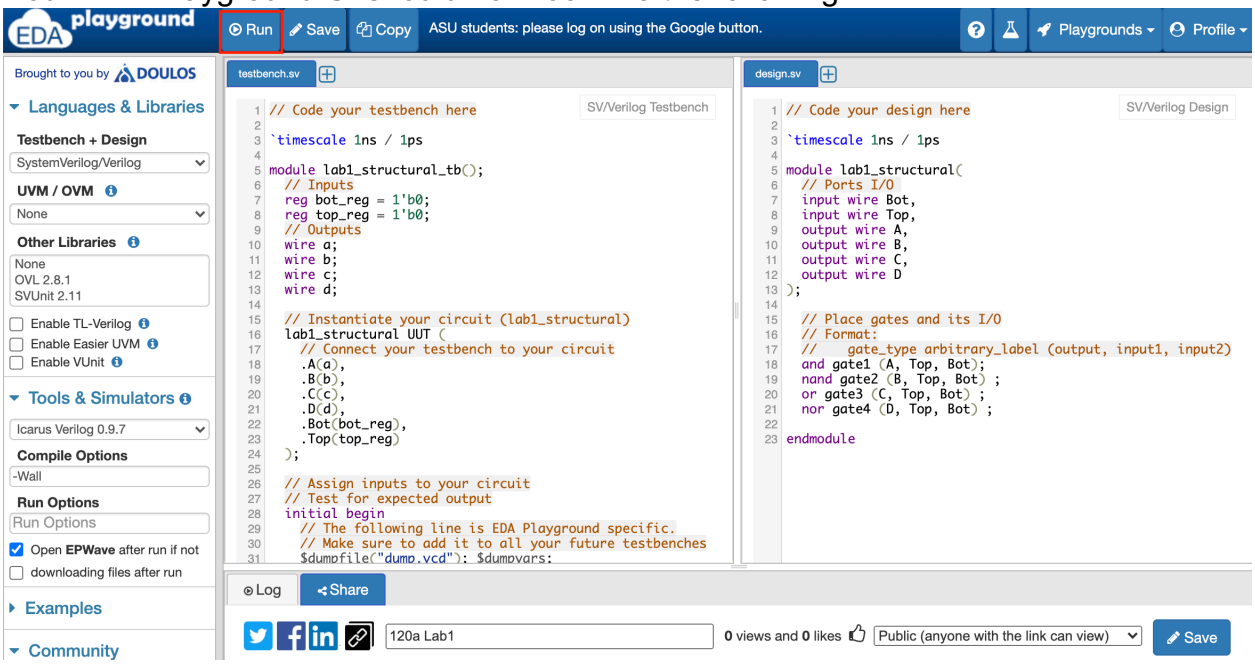
Now that our design is complete, we want to make sure it functions correctly.
To this end, we will create a testbench that generates the input stimulus for our circuit and checks to ensure the circuit produces the desired output.
Verilog can be used to create testbenches as well.

The testbench code is given below. Copy this to the “SV/Verilog Testbench” window of your EDA playground UI.

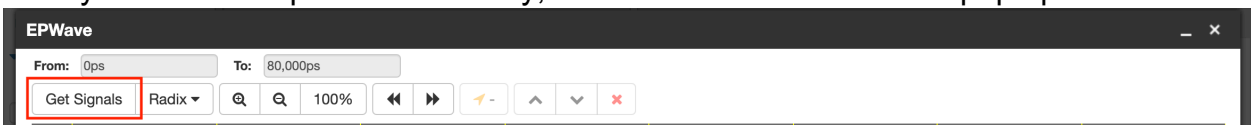
```
`timescale 1ns / 1ps

module lab1_structural_tb();
  // Inputs
  reg bot_reg = 1'b0;
  reg top_reg = 1'b0;
  // Outputs
  wire a;
  wire b;
  wire c;
  wire d;

  // Instantiate your circuit (lab1_structural)
  lab1_structural UUT (
    // Connect your testbench to your circuit
    .A(a),
    .B(b),
    .C(c),
    .D(d),
    .Bot(bot_reg),
    .Top(top_reg)
  );
```

```
  // Assign inputs to your circuit
  // Test for expected output
  initial begin
    // The following line is EDA Playground specific.
    // Make sure to add it to all your future testbenches
    $dumpfile("dump.vcd"); $dumpvars;

    // Testcase #1
    $display("Testcase #1");
    // Assign inputs
    bot_reg = 1'b1;
    top_reg = 1'b1;
    #40; // Wait 40 ns
    // Check for expected output
    if ( {a,b,c,d} != 4'b1010 )
      $display  ("\t Result is wrong %b ", {a,b,c,d});
    else
      $display ("\t Testcase #1 successful");

    // Testcase #2
    $display("Testcase #2");
    // Assign inputs
    bot_reg = 1'b0;
    top_reg = 1'b0;
    #40;
    // Check for expected output
    if ( {a,b,c,d} != 4'b0101 )
      $display  ("\t Result is wrong %b ", {a,b,c,d});
    else
      $display ("\t Testcase #2 successful");

    // Add testcase #3 and #4 below
  end

endmodule
```

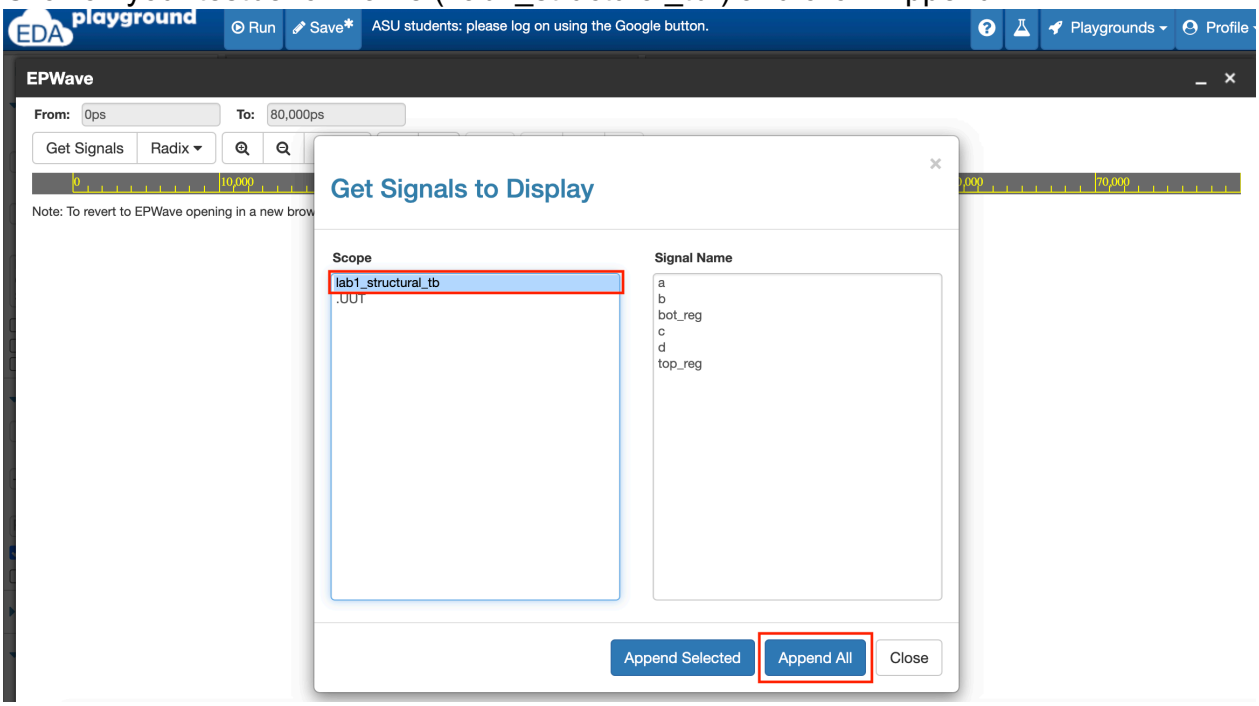
Your EDA Playground UI should now look like the following



Click “Run” to run your testbench.
After your code compiles successfully, the EPWave window should pop up



Click on “Get Signal” to add the desired signals to the waveform
Click on your testbench name (“lab1_structural_tb”) and click “Append All”

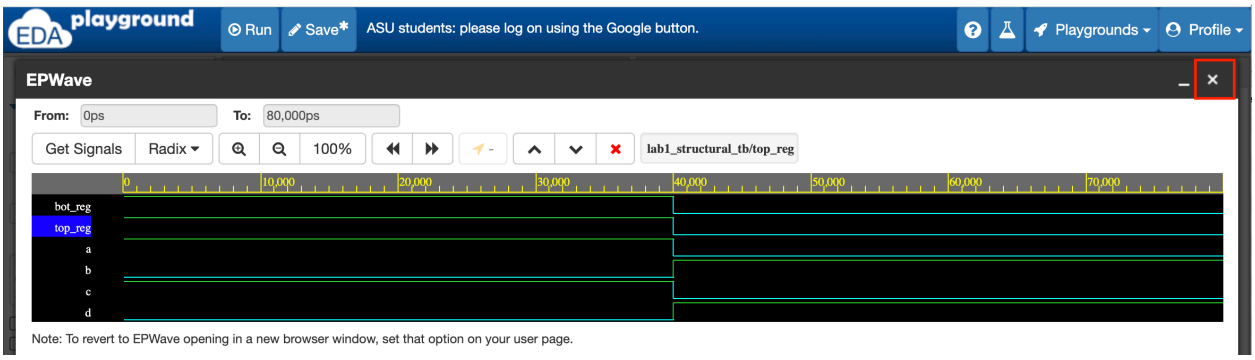


You should now be able to see the simulation waveform.
Does it look correct?

i.e. for the given input (bot_reg, top_reg), is the circuit producing the correct output (a, b, c, d)?

Make sure to discuss this in your lab report.

Now, click on the “x” in the top right corner of your EPWave window to go back to your code.



Note there are 2 more tests you can add to your testbench. Think about what these are and add them to your testbench.

```
53        $display  ("\t Result is wrong %b ", {a,b,c,d});
54      else
55        $display ("\t Testcase #2 successful");
56
57      // Add testcase #3 and #4 below
58    end
59
60  endmodule
```

Generate the final waveform and make sure to include a screenshot of it in your lab report.