COL215 – Digital Logic and System Design
Department of Computer Science & Engineering, IIT Delhi
Semester I, 2025-26
Lab Assignment 1

1 Basic Linux commands

- To check the list of files and directories: **ls -al**

- To create a new file: **touch <filename>**

- Open a file in editor: **vim <filename> or gedit <filename> or nano <filename>**

- To remove file: **rm <filename>**

- Change directory to home: **cd** or **cd /home/<course type><id>**, e.g., /home/btech24/cs1249999

2 One-time GCL setup for Vivado

To know whether the given system is old or new, run the following command in the terminal after login into your GCL account:

§ lsb_release -a

If the Ubuntu version is 20.xx.x then you are using new system, otherwise you are working on the older machine. Also notice the background color in case of new system, whether it is **Purple** or **Gray**.

Students on old systems in the DHD lab should follow the following instructions for **vivado** setup.

- Logon to the machine using your GCL account.

- In your home folder, open the .bashrc file in home folder (create if not already present). Use (gedit .bashrc) to open the file and add the lines below (create a new file if .bashrc doesn't exist):

  - export XILINXD_LICENSE_FILE=2100@10.208.22.170:/extra/xilinxlic/Xilinx.lic
  - source /opt/Xilinx/14.7/ISE_DS/settings64.sh
  - source /opt/Xilinx/Vivado/2016.4/settings64.sh

- Alternatively, you can add the alias below and whenever you open a shell, run the command "enable_xilinx":

  - alias enable_xilinx="export XILINXD_LICENSE_FILE=2100@10.208.22.170:/extra/xilinxlic/Xilinx.lic; source /opt/Xilinx/14.7/ISE_DS/settings64.sh; source /opt/Xilinx/Vivado/2016.4/settings64.sh"

- After making changes to .bashrc, run "source .bashrc" in the terminal

- To run the required application, type **vivado** in the terminal.

Students on new systems don't need the license file and can directly log into their GCL account to start working on **vivado**.

- In your home folder, open the .bashrc file and add the following line (create a new file if .bashrc doesn't exist)

- Execute below command based on the background screen you see on logging in:
  - **Purple**: source /opt/Xilinx/Vivado/2022.1/settings64.sh
  - **Gray**: source /opt/vivado/Vivado/2022.1/settings64.sh

- After making changes to .bashrc, run "source .bashrc" in the terminal

- To run the required application, type **vivado** in the terminal.

3   Launching and creating a project in Vivado

1. To launch Vivado, type **vivado** in the terminal.

2. Once Vivado is launched, logout and login to the system. This step is required only the first time you login.

3. Launch Vivado once again.

4. When the tool opens, click on **Create New project** and click Next.

5. Select the project directory and name.

6. In the next dialog box, select **RTL project.**
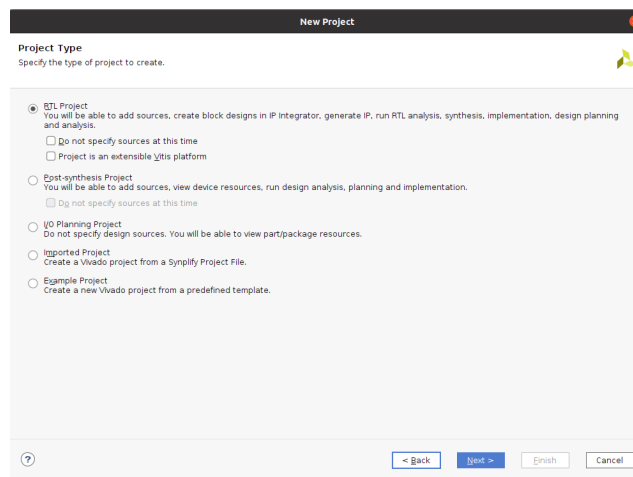
7. Create a new file named **AND_gate.** Select the file type as verilog**.** Ensure that Target and simulator language is verilog.

8. In the next dialog window for **adding constraints**, add the provided basys3.xdc file using "**Add Files**".

9. Select part number **xc7a35tcpg236-1** and click next. After this the project creation is done.

10. Under **design source**, double click **AND_gate** file to open it in the editor and add the following code for the basic AND gate.

11. Functionality and implementation for **AND_gate** is given below,

a	b	a AND b
0	0	0
0	1	0
1	0	0
1	1	1

```verilog
// AND GATE
module AND_gate (
    input a,
    input b,
    output c);

assign c = a & b;

endmodule
```

12. **Simulation of the design**: Now to simulate your design you need to make a *test bench* and test your code.
    Test Benches are used to test the correctness of the Design Under Test (DUT). They can be used to provide inputs to the design and observe the outputs

    As a start, you can force the values of the inputs and observe their output in the waveform.
    A sample testbench is given for simulation of AND gate. You need to create a simulation source file for the test bench.
    Note: Test Bench is only for simulation. Please do not run the Synthesis/implementation flow on it, as introduced in the next steps.

```verilog
// AND_gate_tb.v (testbench)

module AND_gate_tb ();

    // In this TB modeling Style, the test bench instantiates the DUT as a component
    // and passes the inputs from a separate verilog module at instantiation

    reg a, b; // a, b are storage elements
    wire c; // c is the output

    // connecting testbench signals with AND_gate
    AND_gate UUT (
        .a (a),
        .b (b),
        .c (c)
    );

  initial begin
    // inputs
    // 00 at 0 ns
```

3

```
    a = 0;
    b = 0;

    // 01 at 20 ns, as b is 0 at 20 ns and a is changed to 1 at 20 ns
    #20 a = 1;

    // 10 at 40 ns
    #20 b = 0; a = 0;

    // 11 at 60 ns
    #20 a = 1; b = 1;
  end

endmodule
```
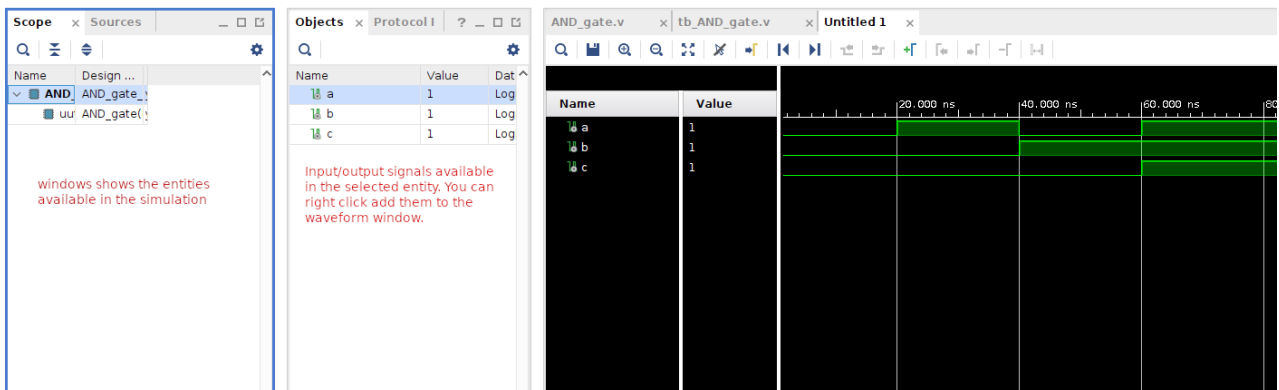

13. You can change the signal value in the simulation window, (i) Right click the signal value and select force constant, (ii) change the value under force constant and (iii) select the simulation time and press play. After running the simulation for the selected time, you can observe the change in output signal c in the waveform window (highlighted in the red box). Refer to video: `https://drive.google.com/file/d/1ozKwQ-5wwQQR7VayAWX1QrbPDb-SC65W/view?usp=sharing`
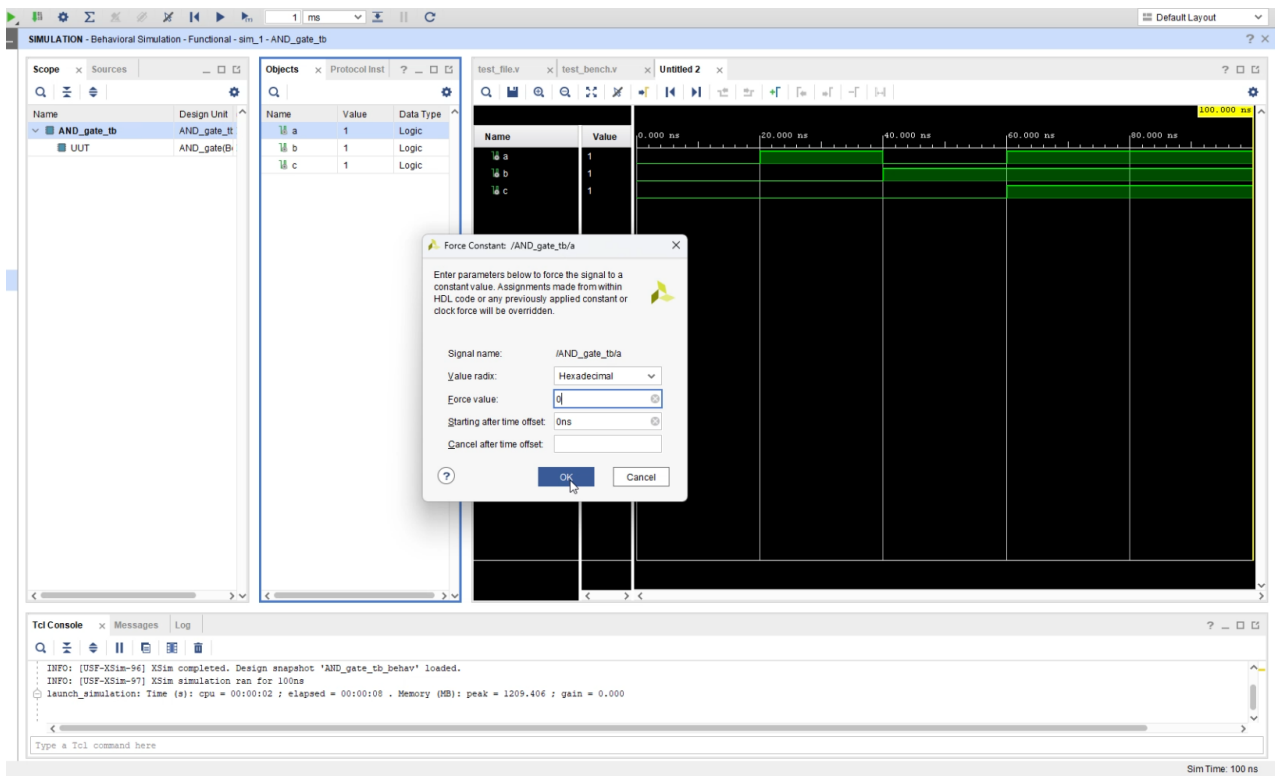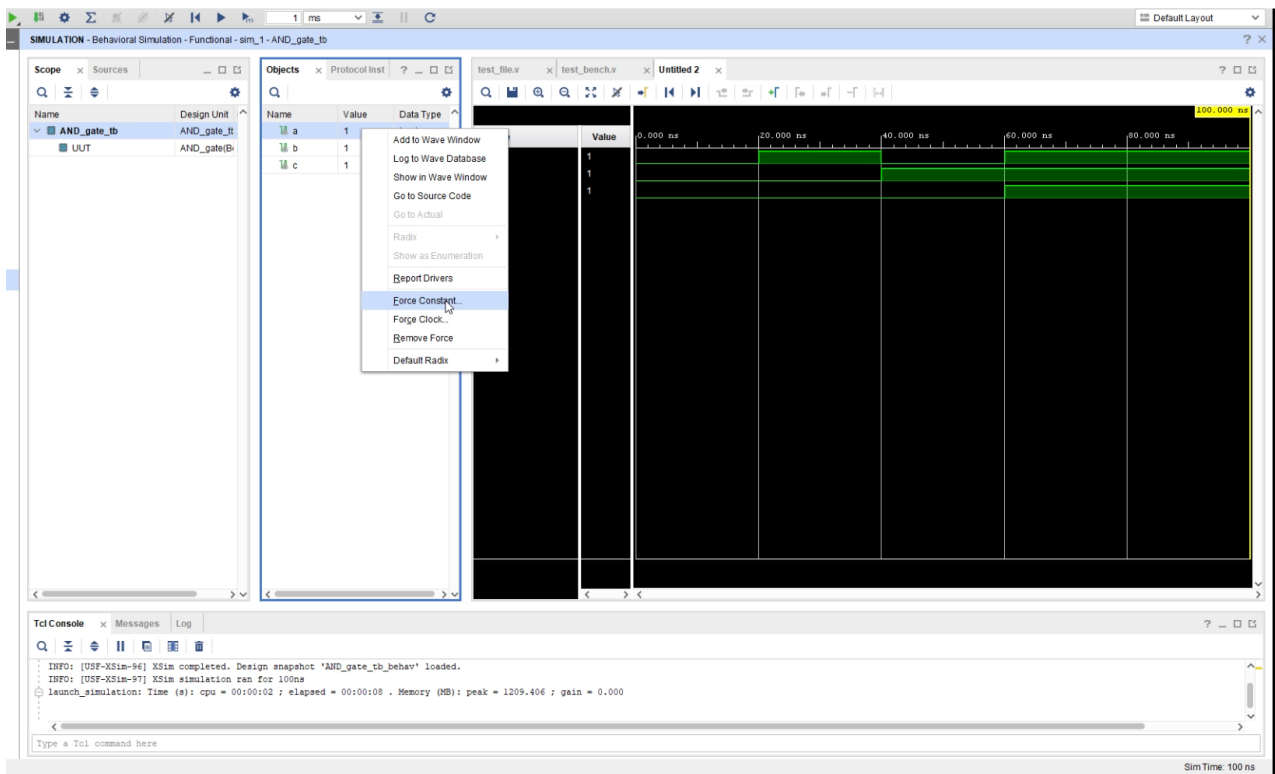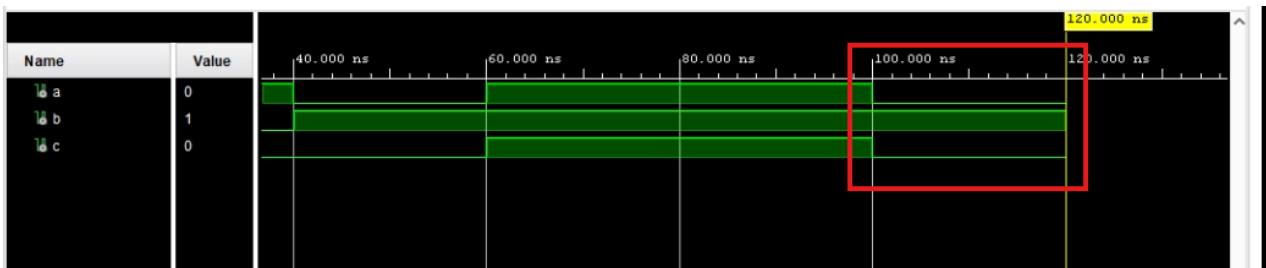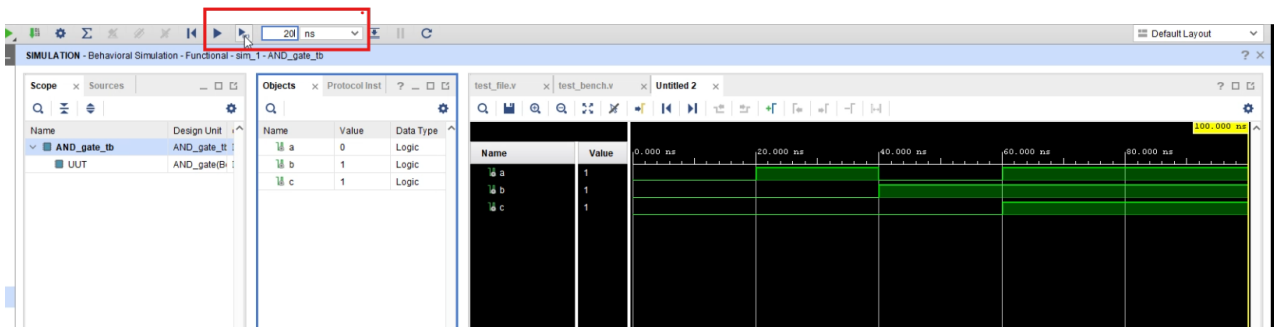
14. Next under the **Constraints section**, edit the **basys3.xdc** file to connect the switches and LED to the declared port. In this module, switches V17 and V16 act as inputs to the gate and LED U16 as the output of the gate.

```
set_property PACKAGE_PIN V17 [get_ports {a}]
        set_property IOSTANDARD LVCMOS33 [get_ports {a}]
set_property PACKAGE_PIN V16 [get_ports {b}]
        set_property IOSTANDARD LVCMOS33 [get_ports {b}]
....
....
....
set_property PACKAGE_PIN U16 [get_ports {c}]
        set_property IOSTANDARD LVCMOS33 [get_ports {c}]
....
....
```

15. Under the Flow Navigator, Click on **Run synthesis** then **Run implementation**. **Please open the synthesized design to analyze the generated output. Look at the resource utilization. Observe the number of LUTs used in your design. Open Synthesized Design => Report utilization**
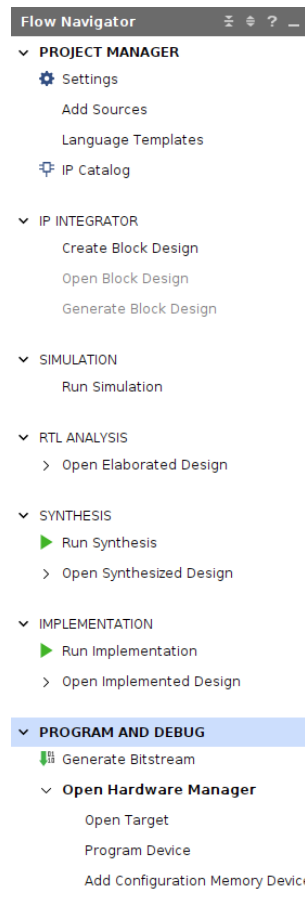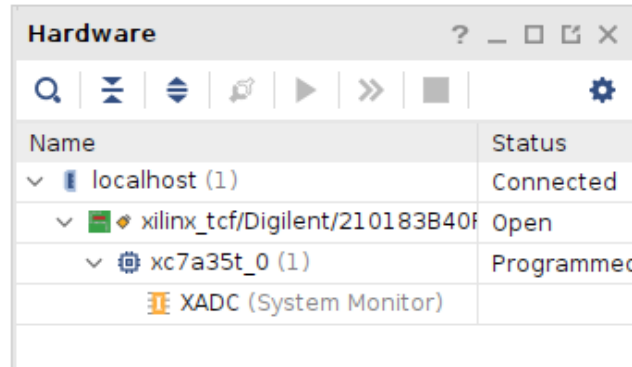
(a) Additional info: Number of jobs can be increased depending upon the available cores. This may improve the running time. For now, use the default value given in the dialog box.
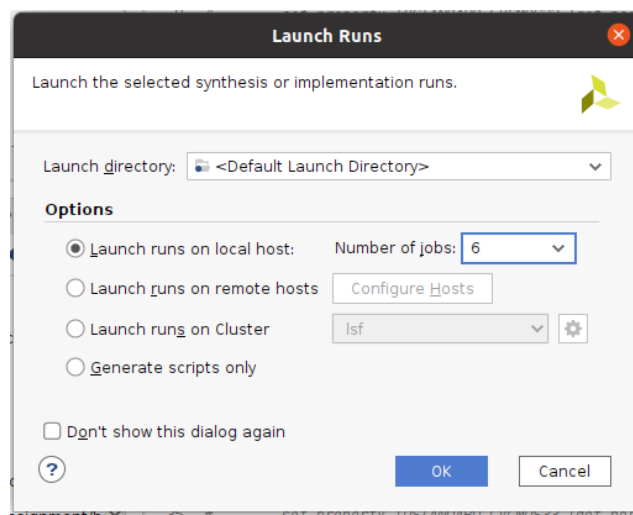

16. After this click on **Generate Bitstream**.

17. Once this is done, click on Open **Hardware Manager => Open Target => Auto Connect**. Ensure that basys3 board is connected to the system via micro-USB cable.

18. Click on **Program Device**. Make sure the correct bit file is selected.

19. Once the process is complete, toggle the bottom right corner switches (V16 and V17, the AND gate inputs). This should change the LED (LD0/U16) according to the AND gate logic.
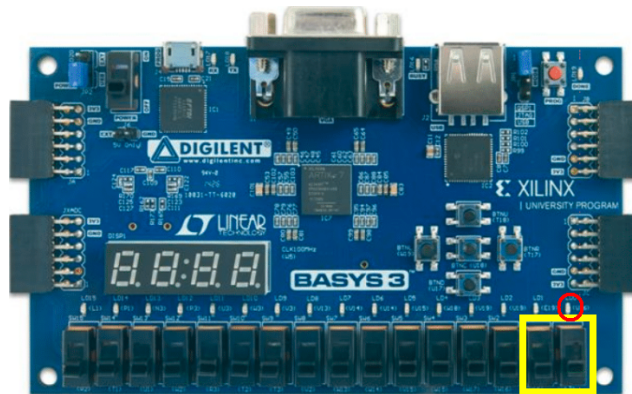
20. After completing the simulation and testing of AND gate, the next task is to realize a two input OR and single input NOT gate using the same template. Map different Switches on the board for OR and NOT gate. All three gates should be running using different switches on the basys 3 board.

Important points to note

- At the end of each lab, send the saved files and snapshots to your institute email account and remove the files. The space is limited and the vivado logfiles occupy a lot of disk space. If not done and the quota is reached, you wouldn't be able to do anything further.

- You should use the same machine in all subsequent labs to avoid doing setup again and again.

---

4 Assignment Submission Instructions

The general instructions below need to be followed for all assignments. Only one partner needs to submit. Mention all team member names and entry IDs during the submission.

1. Name the submission file as entryNumber1_entryNumber2.zip

2. Go to Gradescope

3. Upload the following files:

   - Source file (.v)
   - Constraint File (.xdc)
   - Bit file (.bit)
   - Report (.pdf) - The report needs to state your design decisions, lab work, simulation snapshots, synthesis report (particularly resource counts: Flip-flops, LUTs, BRAMs, and DSPs) and generated schematics.