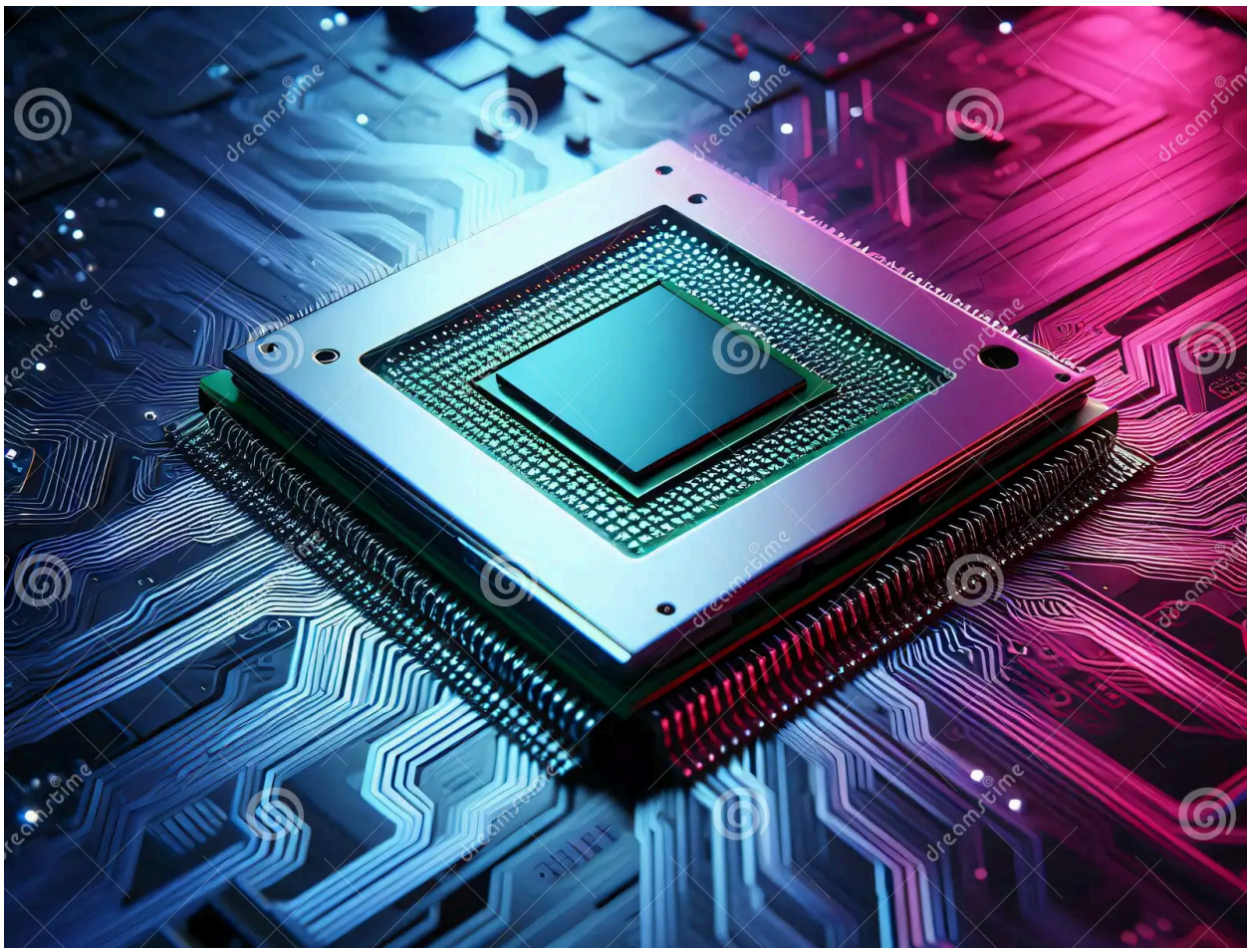
COL215

Laboratory Assignment Report 1

Basic Digital Gates (AND, OR and NOT)



Created by:

Sambhav Singh Aditya 2024CS10177

Jay Rami 2024CS10510

Lab Assignment

In this lab, we aimed for

1. Setting up GCL account for Vivado and creating a new project in verilog HDL.
2. Writing the modules in .v source design files for basic gates like OR, AND and NOT gate.
3. Writing the testbench for the simulation appropriately such that many possible cases of values of variable can be accessed.
4. Simulating the gates, in the software of Vivado using the `run Simulation` Command.
5. Synthesizing the modelled design.
6. Implementing it, and taking a look on the description of formed hardware.
7. Generating bitstream file for the programming of the board **Basys 3™ FPGA Board.**

Finally, the board should have mapped total of 5 switches (2 for AND, 2 for OR and 1 for NOT) with 3 LEDs respectively, so that it can demonstrate the gates.

Theory

AND gate

It is simple logic gate that would be taking input 2 bool value and giving single bool output. The truth table according to which this works is here:

Input a	Input b	Output c
0	0	0
0	1	0
1	0	0
1	1	1

OR gate

It is again a basic binary logic gate, that would take 2 input and provide a output according to this truth table:

Input a	Input b	Output c
0	0	0
0	1	1
1	0	1
1	1	1

NOT gate

It is a basic logic gate that would take a input and return the negation of that input. The truth would look like:

Input a	Output b
0	1
1	0

Modelling the gates in Vivado

For modelling then inside the software of Vivado, verilog HDL was used to define different modules for the gates separately.

We had the option to choose from:

1. Generate a module that would take 5 inputs and give outputs in other variables, which we would map with the buttons of the board.

2. Generate separate module for the gates respectively and the instantanize them inside another module which is to be programmed inside the board.

I chose the first one for synthesis while other one for the simulation. There are 3 types of files to be created:

- Source design files (.v)
- Simulation files (test bench files (.v))
- Constraints file (basys3.xdc in this case)

All the source design files, simulation files and __ are attached to assignment zip.

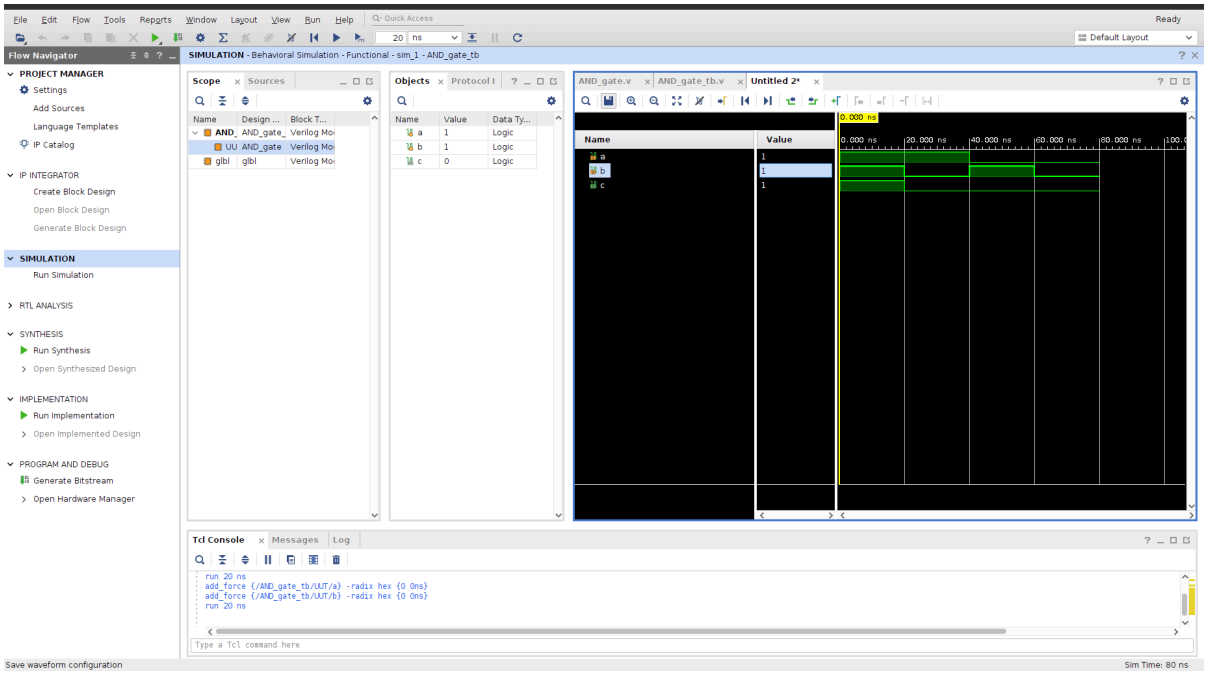
Source file is exactly what have the real code logic of a hardware.

While simulation files are that gives the specification and details needed for simulation to to like automating and triggering the input variables with time.

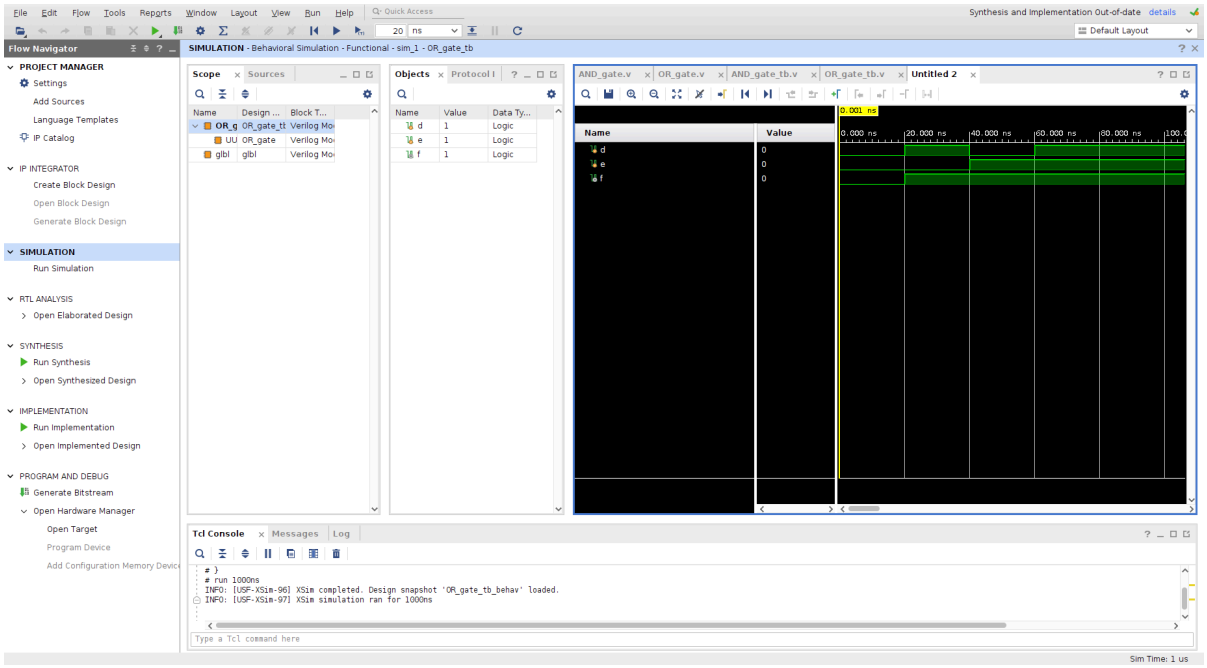
Constraint file is actually used to map variable to the buttons of board

Simulation Result

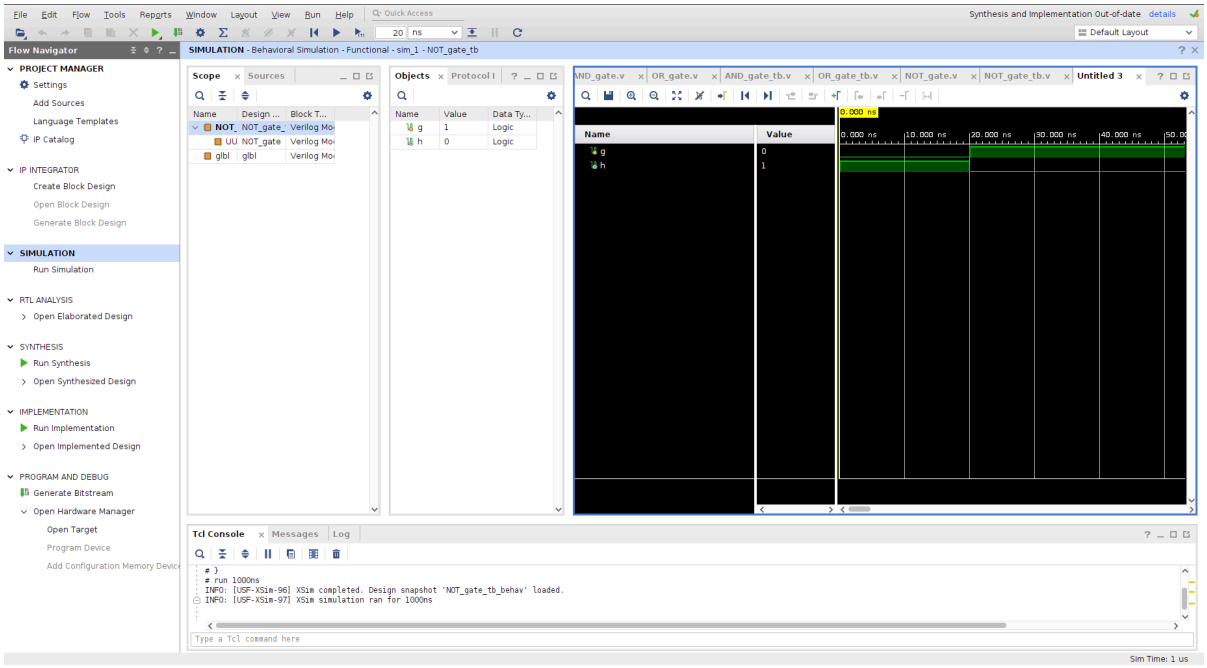
1. AND Gate



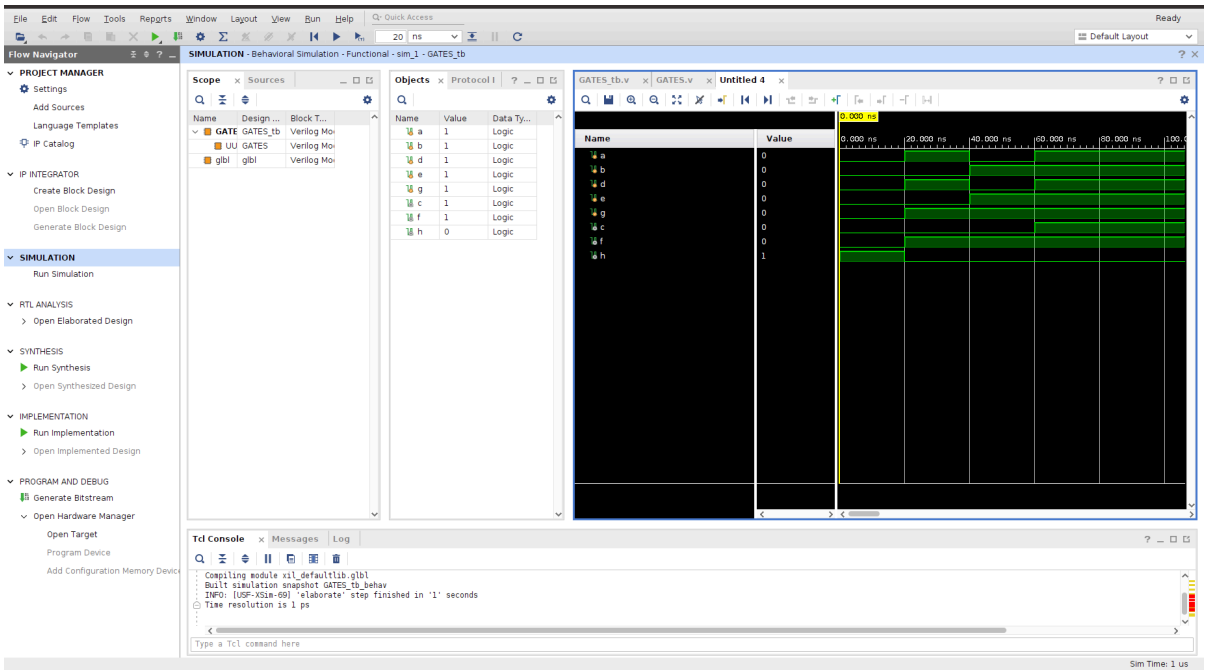
2. OR Gate



3. NOT Gate



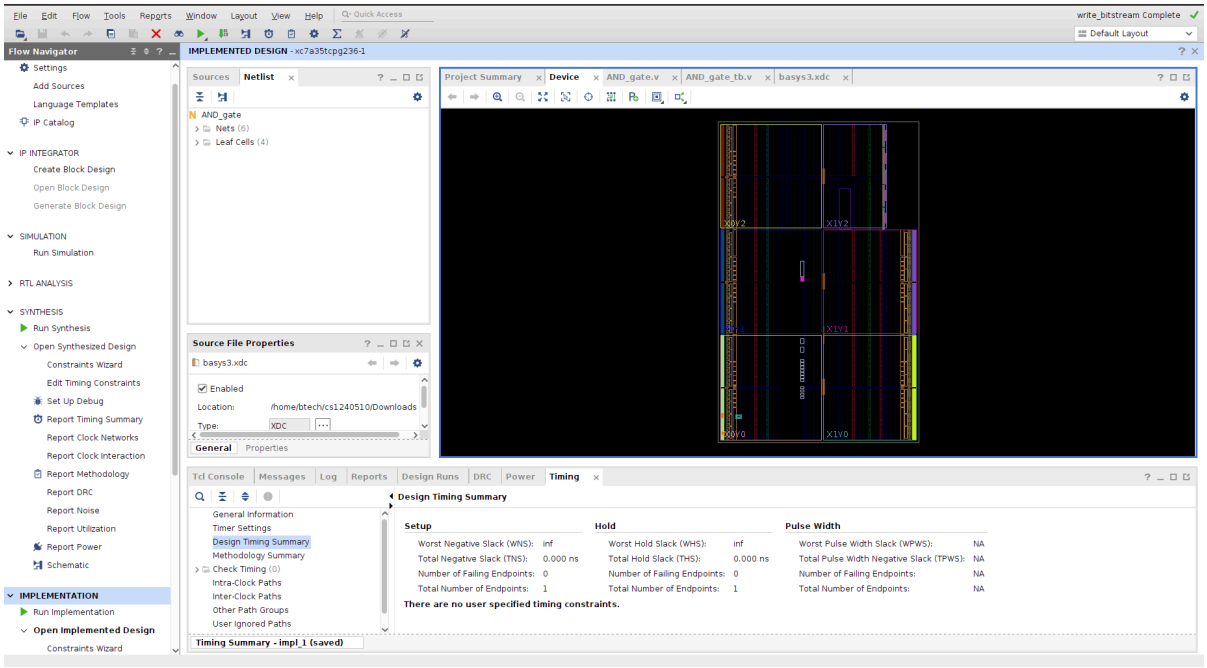
4. All together



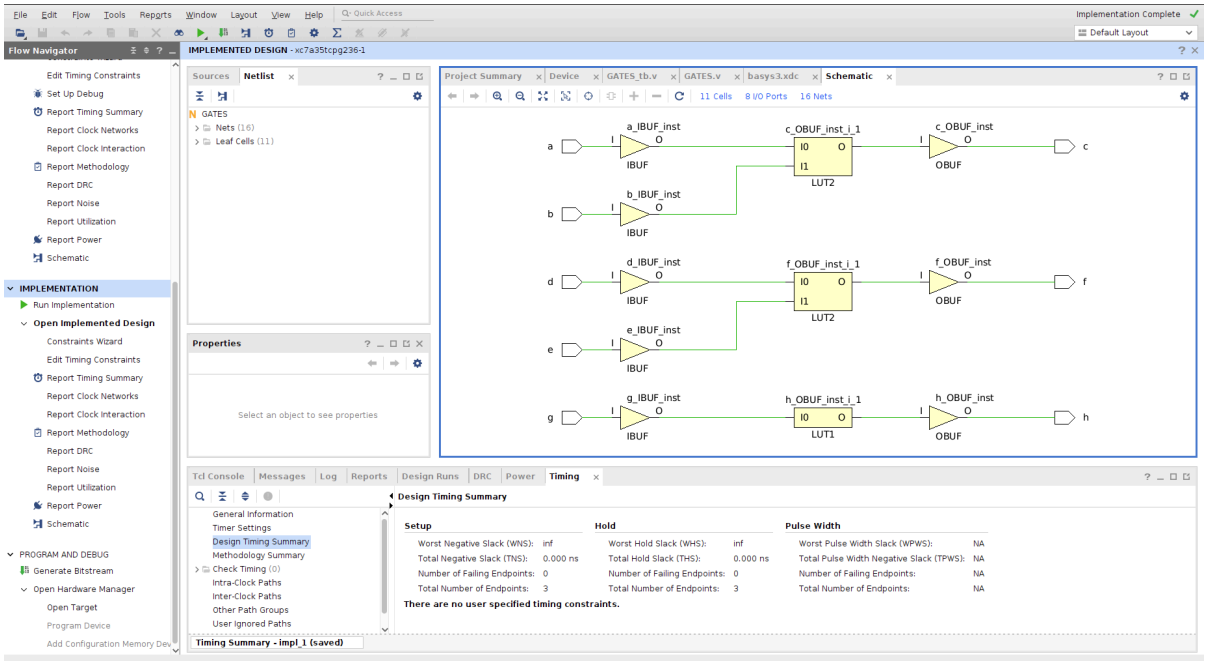
The respective gates in the combined simulation are:

1. AND gate:
    - input of a, b;
    - and output of c;
2. OR gate:
    - Input of d, e;
    - Output of f;
3. NOT gate:
    - Input of g;
    - Output of h;

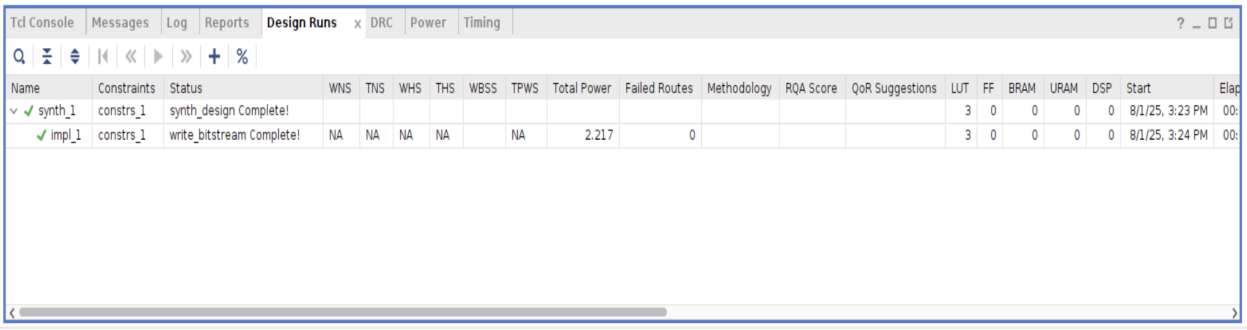
Design suite



Schematic Diagram (of combined gates)



Synthesis Report



Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM	DSP	Start	Elap
✓ synth_1	constrs_1	synth_design Complete!												3	0	0	0	0	8/1/25, 3:23 PM	00:
✓ impl_1	constrs_1	write_bitstream Complete!	NA	NA	NA	NA		NA	2.217	0				3	0	0	0	0	8/1/25, 3:24 PM	00:

Number of

1. LUTs=3
2. Flip flops=0
3. BRAM=0
4. DSP=0

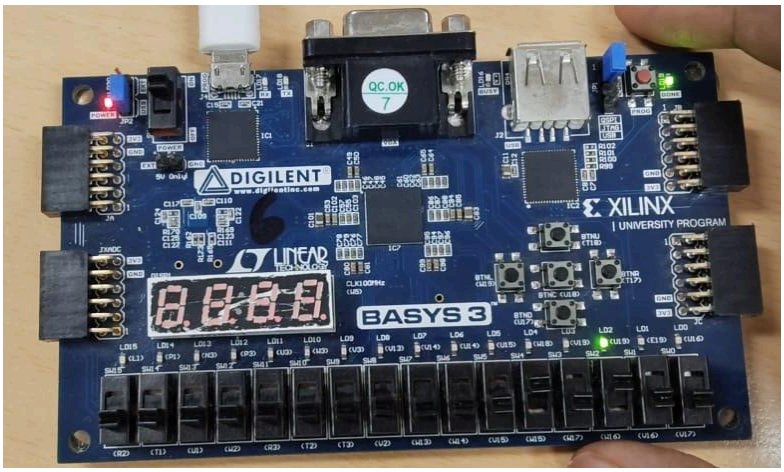
Programming the Basys 3™ FPGA Board

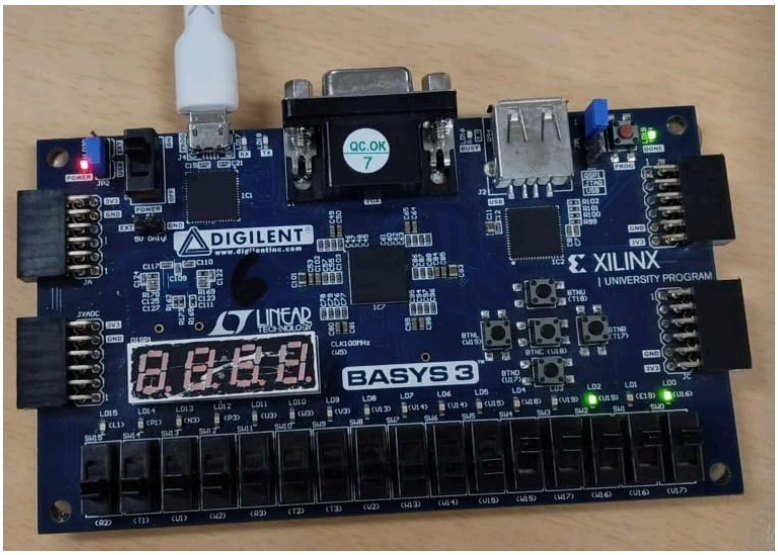
After generating the bitstream of the combined gate, using the constraint files and source file:

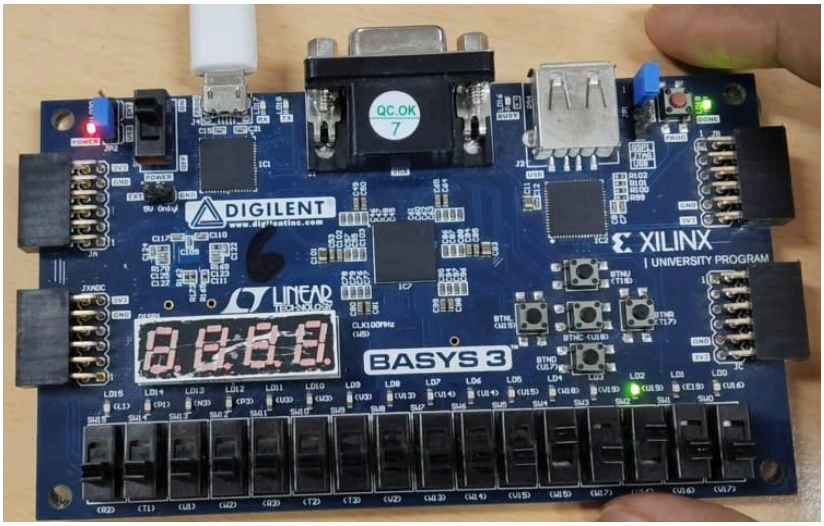
- Constraint file containing the data of mapping and active switches/LEDs.
- Source file containing the logic according to which the board will be programmed.

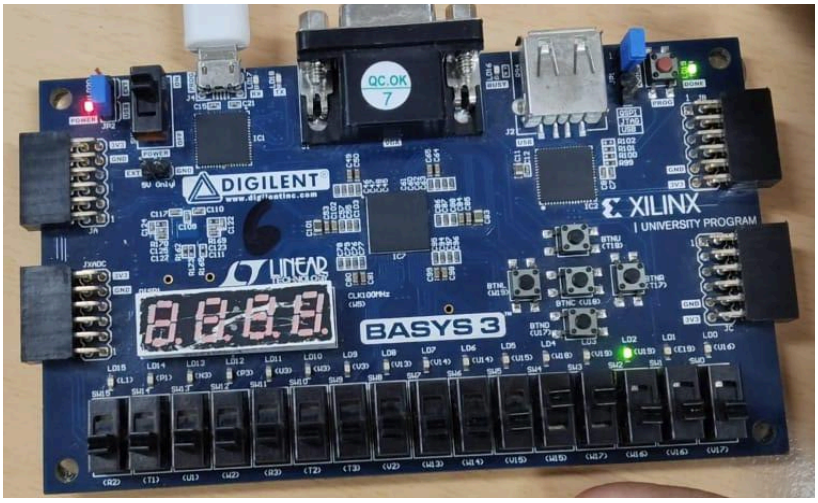
The .bit file is programmed inside the **Basys 3™ FPGA Board.** We used 5 different switches for input and 3 LEDs for the output.

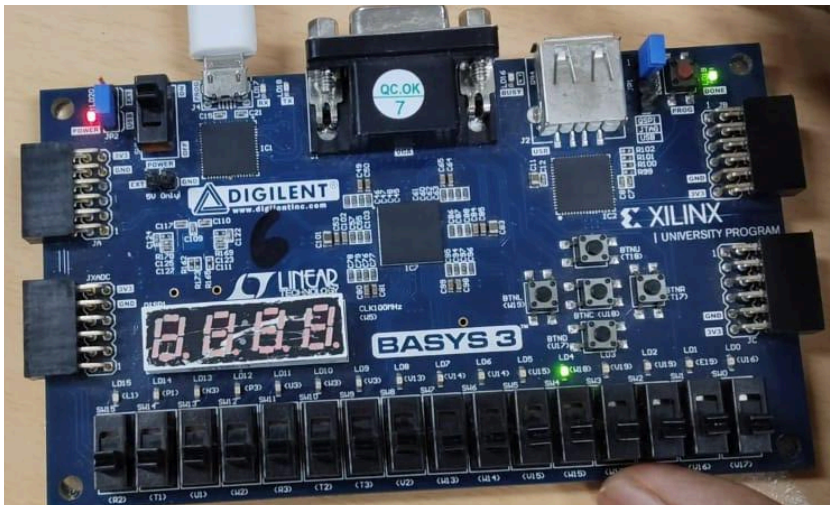
Here are the final results for different combination:

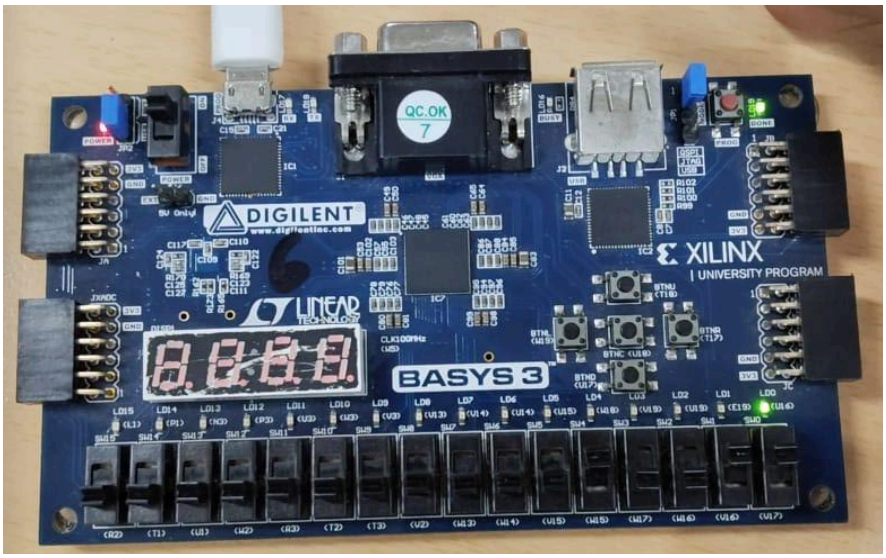

QC.OK
7
DIGILENT
XILINX
UNIVERSITY PROGRAM
LINEAR TECHNOLOGY
BASYS 3


QC.OK
7
DIGILENT
XILINX
UNIVERSITY PROGRAM
LINEAR TECHNOLOGY
BASYS 3


QC.OK
7
DIGILENT
XILINX
UNIVERSITY PROGRAM
LINEAR TECHNOLOGY
BASYS 3


QC.OK
7
DIGILENT
XILINX
UNIVERSITY PROGRAM
BASYS 3


QC.OK
7
DIGILENT
XILINX
UNIVERSITY PROGRAM
BASYS 3



Thanking you!