Hardware Assignment 1

2XMultiplexer and 4XMultiplexer

Design Decisions

2XMultiplexer

A 2x1 multiplexer (MUX) has two data inputs (`D1`, `D2`), one select input (`S`), and one output (`O`). The select input (`S`) determines which data input is passed to the output.

Boolean Expression: $Y=S'\cdot D1+S\cdot D2$ Where:

- S′ is the complement of `S` (created using a NOT gate).
- '·' represents an AND operation.
- \+ represents an OR operation.

Circuit Design:

1.NOT Gate: Invert the select line `S` to get `S'`.
2.AND Gates:
- AND gate 1: Connect `S'` and `D1` to this gate. The output will be S′·D1.
- AND gate 2: Connect `S` and `D1` to this gate. The output will be S·D2.

3.OR Gate: Combine the outputs of the two AND gates to get the final output $Y=S'\cdot D1+S\cdot D2$.

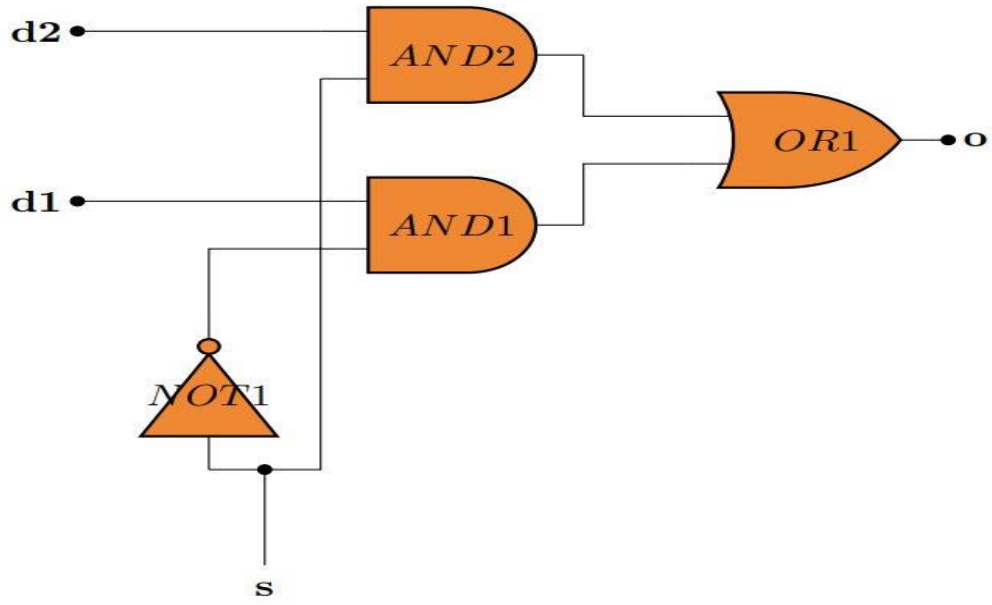


Figure 1: Circuit diagram of 2x1 MUX

Truth Table

Select	Inputs		Output
0	0	0	0
0	0	1	1
1	1	0	1
1	1	1	1

Resource Utilization Table

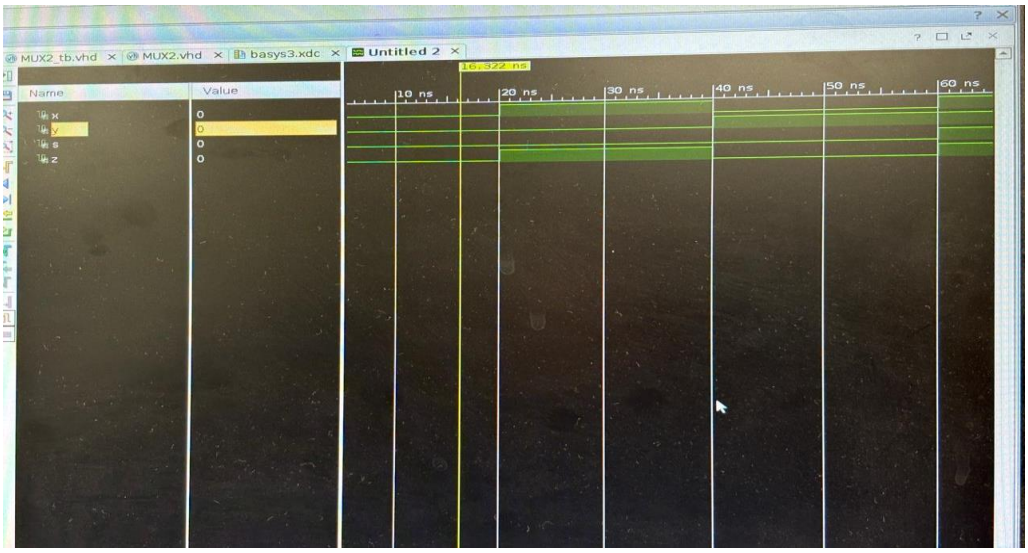
Resource Type	Used	Fixed	Available	Utilization (%)
Slice LUTs*	1	0	20,800	<0.01
- LUT as Logic	1	0	20,800	<0.01
- LUT as Memory	0	0	9,600	0.00
Register as Flip Flop	0	0	41,600	0.00
Register as Latch	0	0	41,600	0.00

Memory Type	Used	Fixed	Available	Utilization (%)
Block RAM Tile	0	0	50	0.00
RAMB36/FIFO*	0	0	50	0.00
RAMB18	0	0	100	0.00

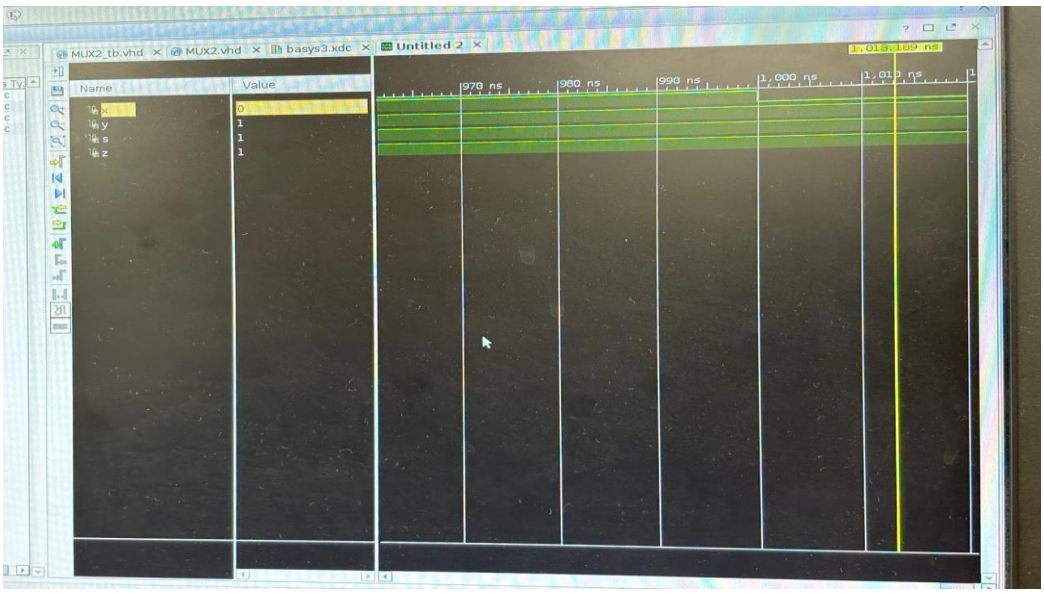
DSP Resource	Used	Fixed	Available	Utilization (%)
DSPs	0	0	90	0.00

Simulation Snapshots of 2x1 Multiplexer

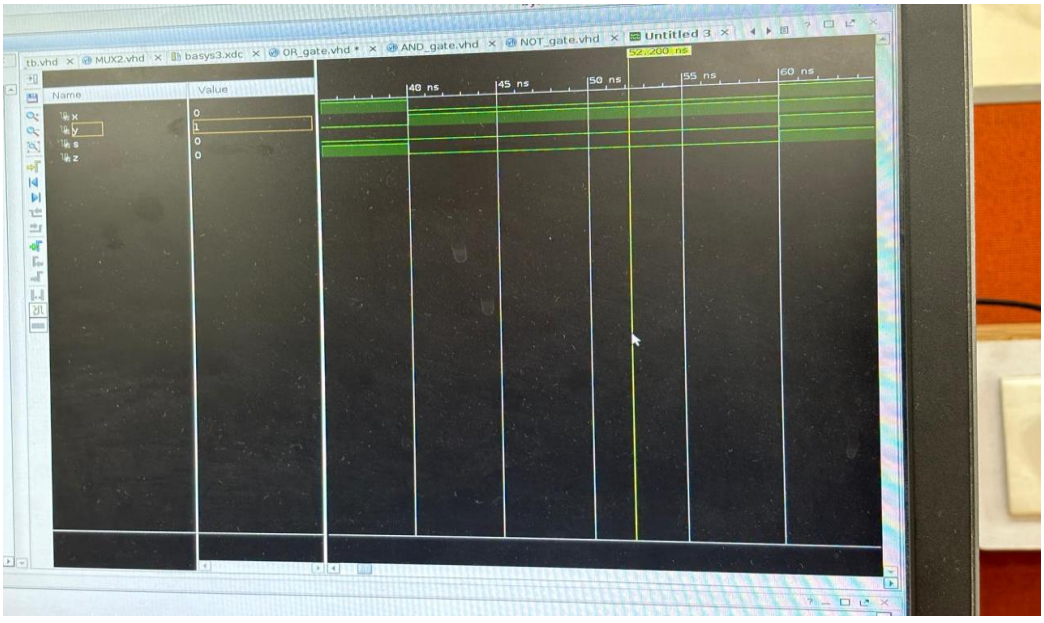
1.



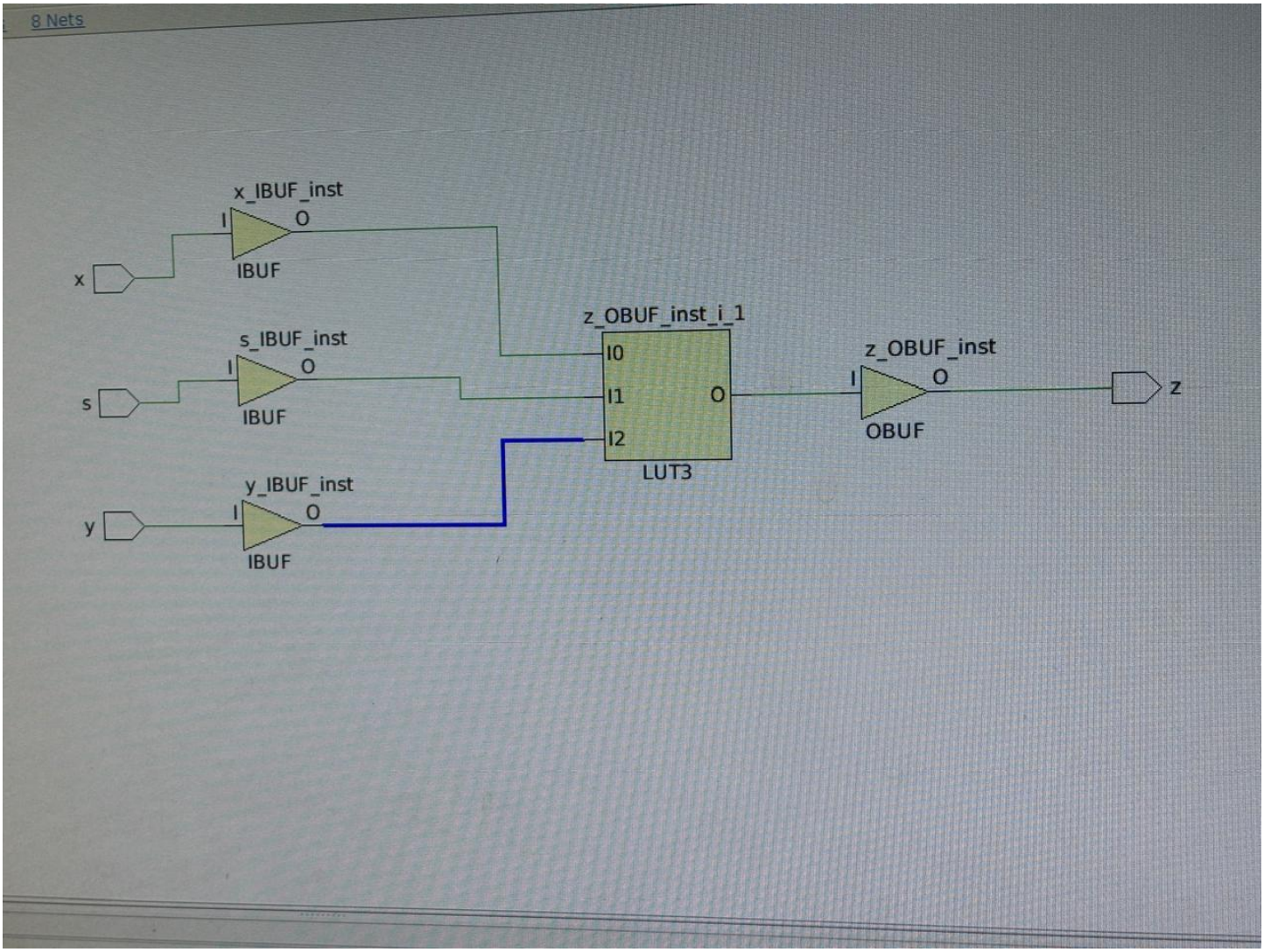
2.



3.

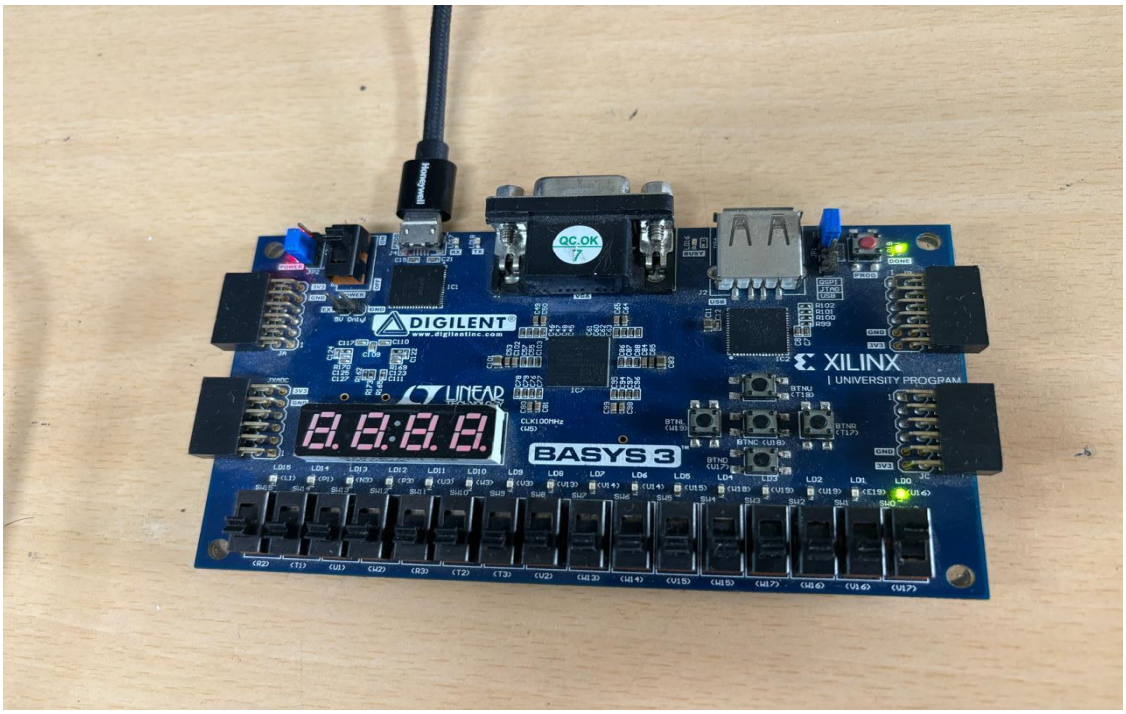


Schematic Snapshot

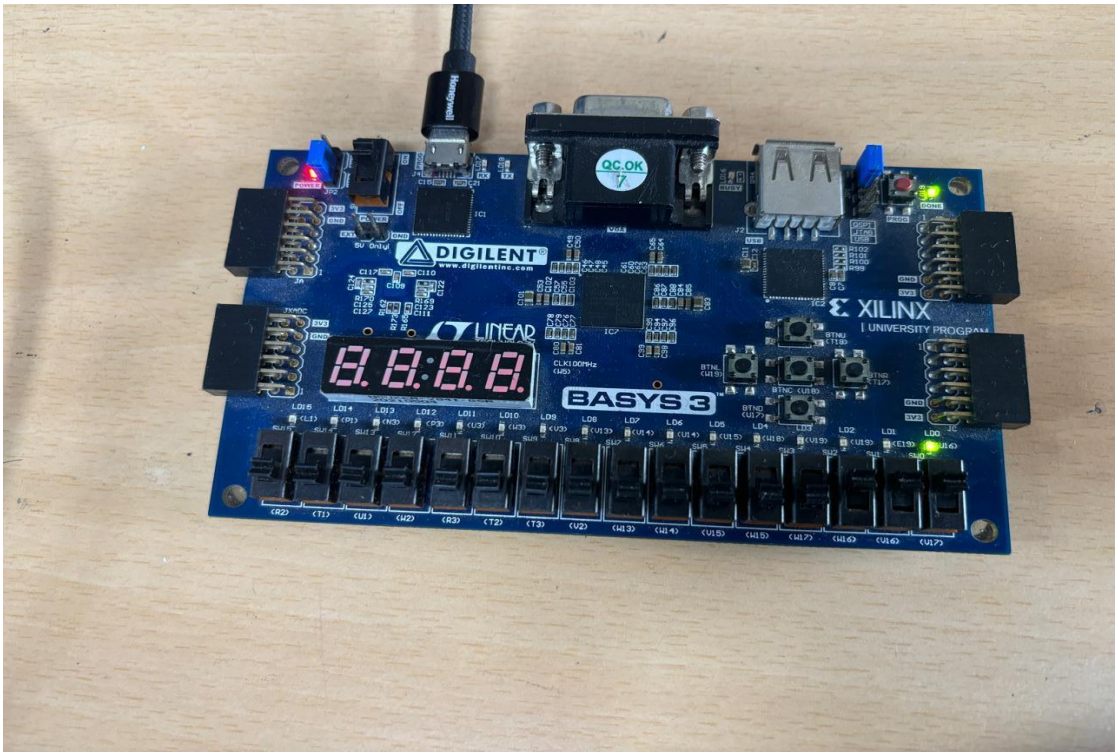


Board Snapshot

1.



2.



4x1 Multiplexer Design using 2x1 Multiplexers

A 4x1 multiplexer has four data inputs (`D1`, `D2`, `D3`, `D4`), two select inputs (`S1`, `S0`), and one output (`Y`). You can build a 4x1 MUX using three 2x1 multiplexers.

Design Steps:

1.First Level (Two 2x1 MUXs):

- **MUX1:** Select between `D1` and `D2` based on `S0`.
- **MUX2:** Select between `D3` and `D4` based on `S0`.

2.Second Level (One 2x1 MUX):

- **MUX3:** Select between the outputs of `MUX1` and `MUX2` based on `S1`.

Detailed Connection:

- **MUX1:**
  - Inputs: `D0`, `D1`
  - Select: `S0`
  - Output: `Y0` (Intermediate output)
- **MUX2:**
  - Inputs: `D2`, `D3`
  - Select: `S0`
  - Output: `Y1` (Intermediate output)
- **MUX3:**
  - Inputs: `Y0`, `Y1`
  - Select: `S1`
  - Output: `Y` (Final output)

In this configuration:

- If `S1` = 0, MUX3 selects the output from `MUX1`, which in turn selects between `D1` and `D2` based on `S0`.
- If `S1` = 1, MUX3 selects the output from `MUX2`, which in turn selects between `D3` and `D4` based on `S0`.

This hierarchical design approach allows to create a 4x1 MUX using smaller 2x1 MUX components.

Truth Table

S_1	S_0	A	B	C	D	Out
0	0	0	x	x	x	0
0	0	1	x	x	x	1
0	1	x	0	x	x	0
0	1	x	1	x	x	1
1	0	x	x	0	x	0
1	0	x	x	1	x	1
1	1	x	x	x	0	0
1	1	x	x	x	1	1

*Where x can be ‘0’ or ‘1’

A=D1 B=D2

C=D3

D=D4

Resource Utilization Table

Resource Type	Used	Fixed	Available	Utilization (%)
Slice LUTs*	1	0	20,800	<0.01
- LUT as Logic	1	0	20,800	<0.01
- LUT as Memory	0	0	9,600	0.00
Register as Flip Flop	0	0	41,600	0.00
Register as Latch	0	0	41,600	0.00

Memory Type	Used	Fixed	Available	Utilization (%)
Block RAM Tile	0	0	50	0.00
RAMB36/FIFO*	0	0	50	0.00
RAMB18	0	0	100	0.00

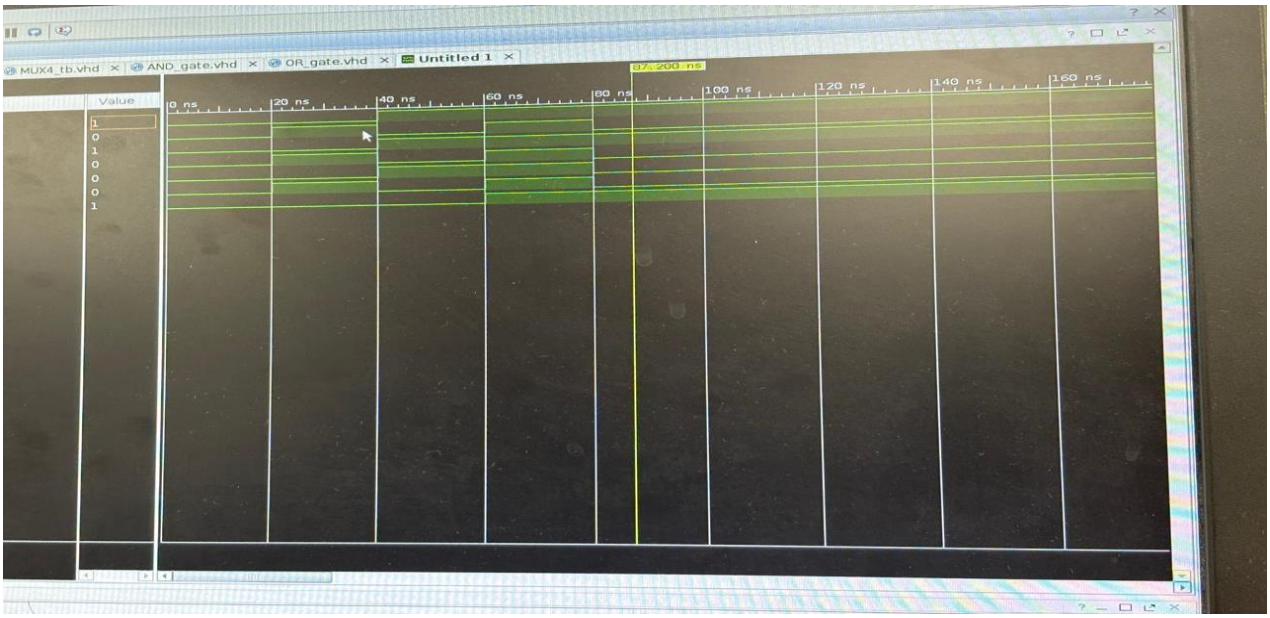
DSP Type	Used	Fixed	Available	Utilization (%)
DSPs	0	0	90	0.00

AND gate

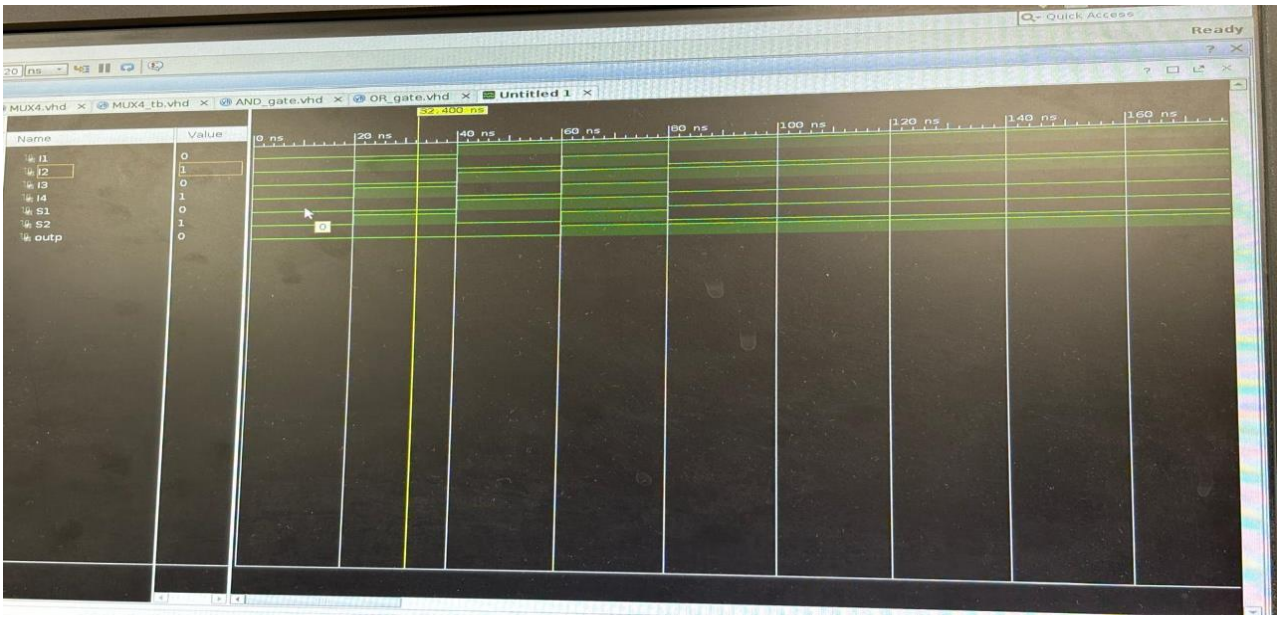
Simulation Snapshots

Simulation Snapshots of 4x1 Multiplexer

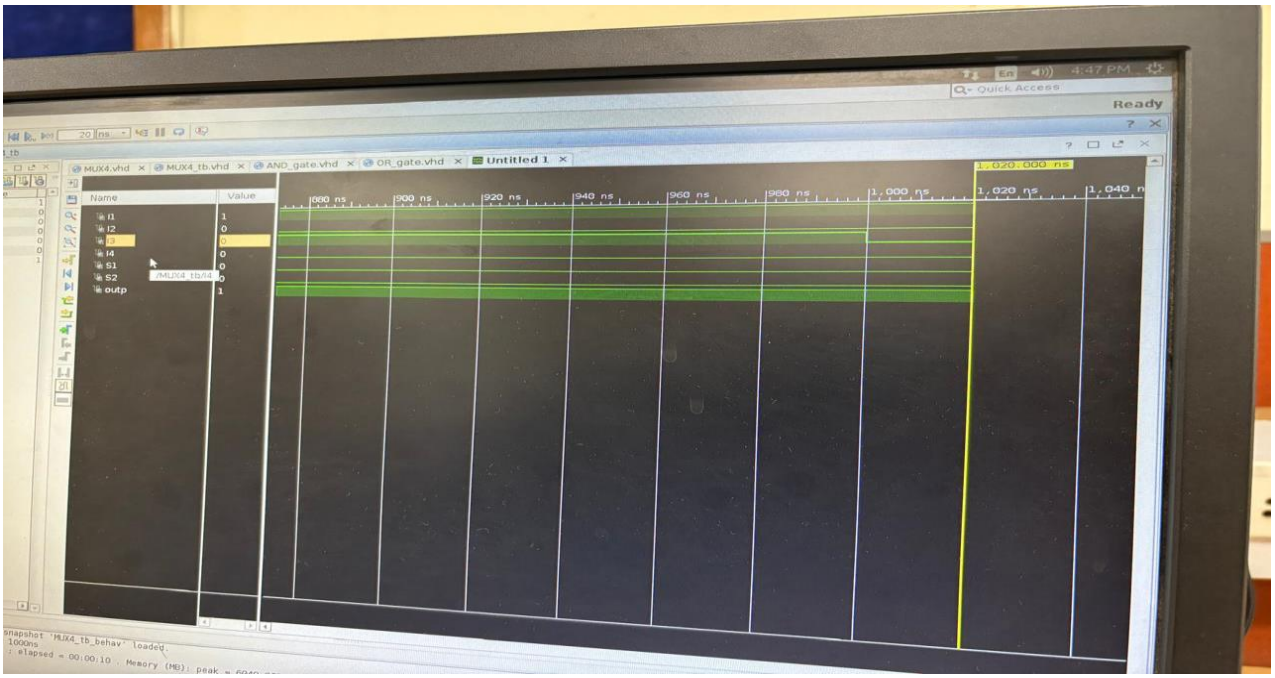
1.



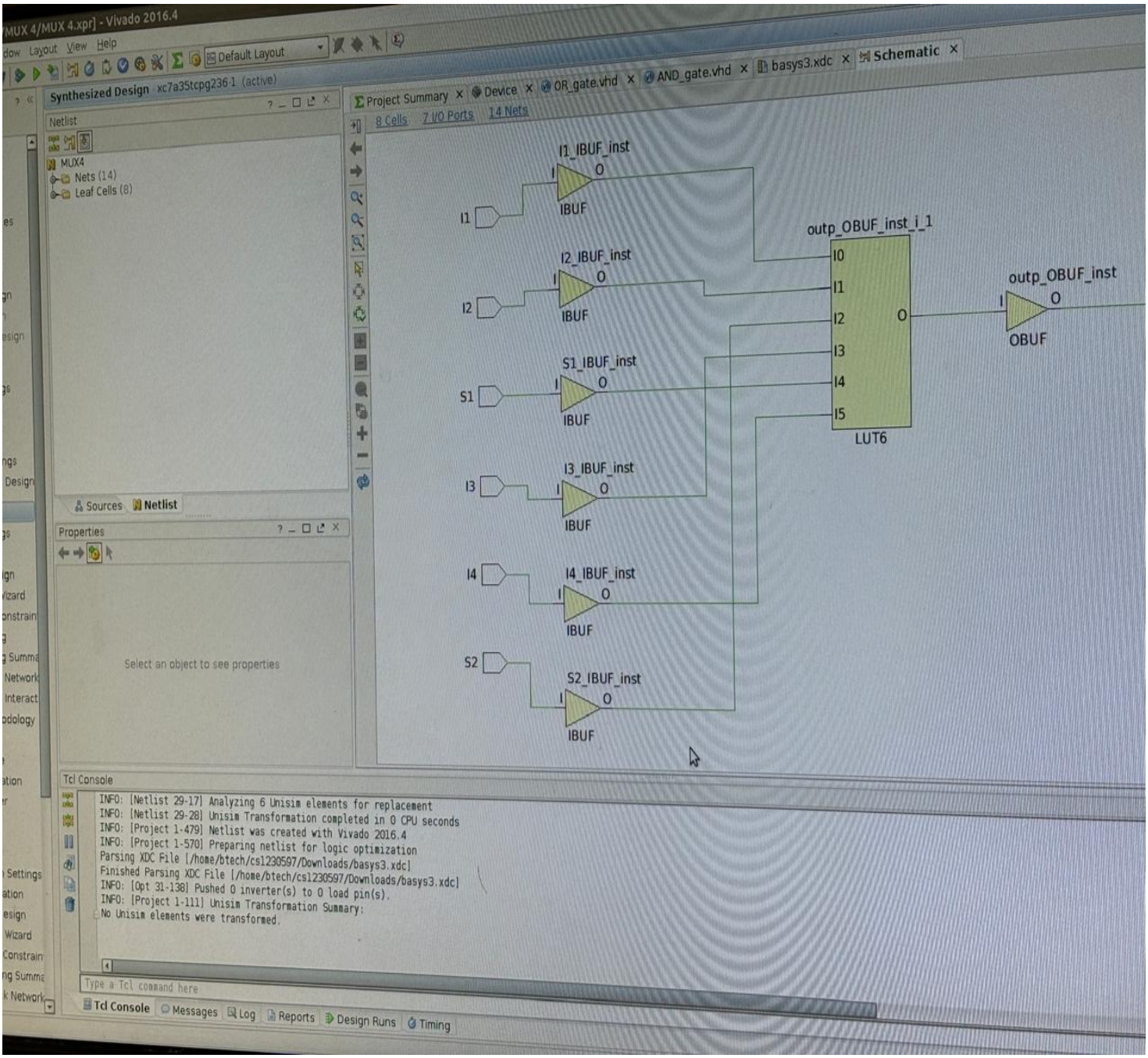
2.



3.

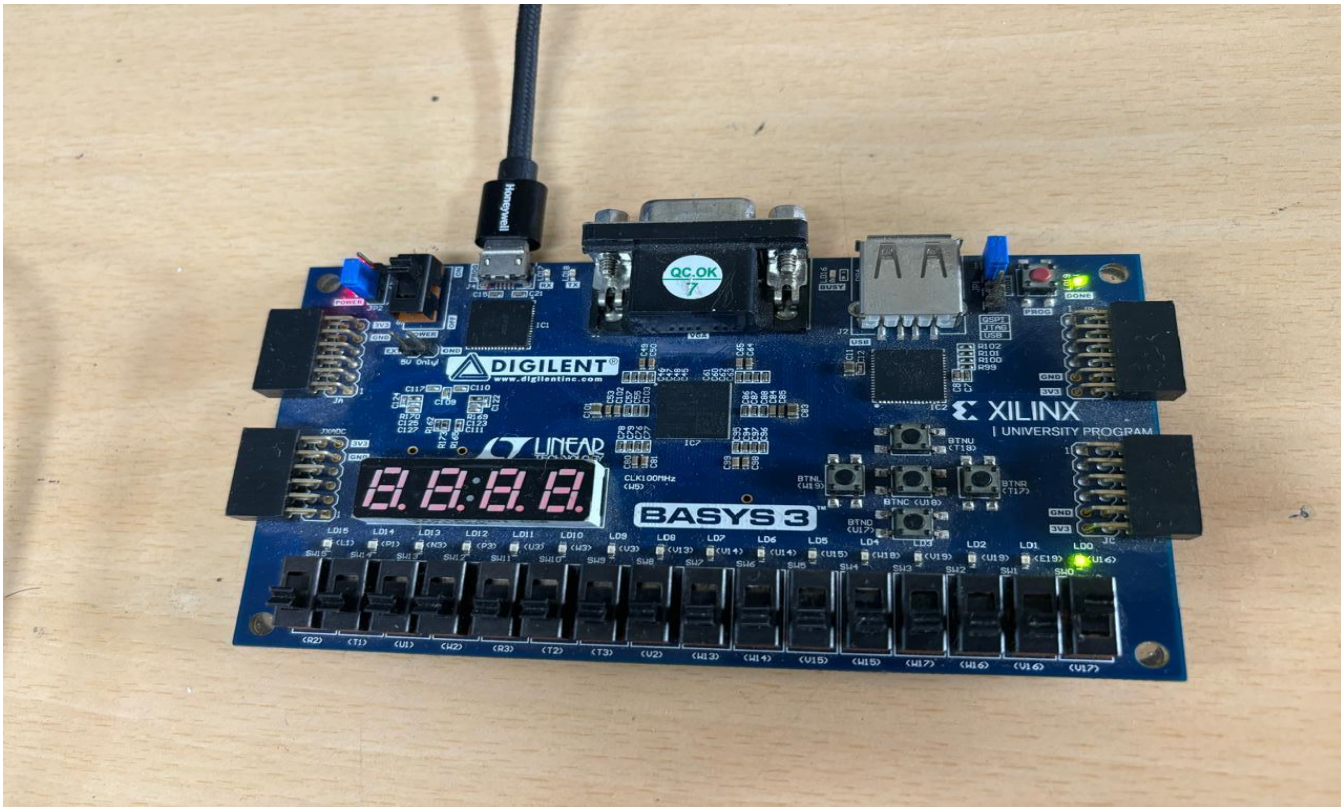


Schematic Snapshot

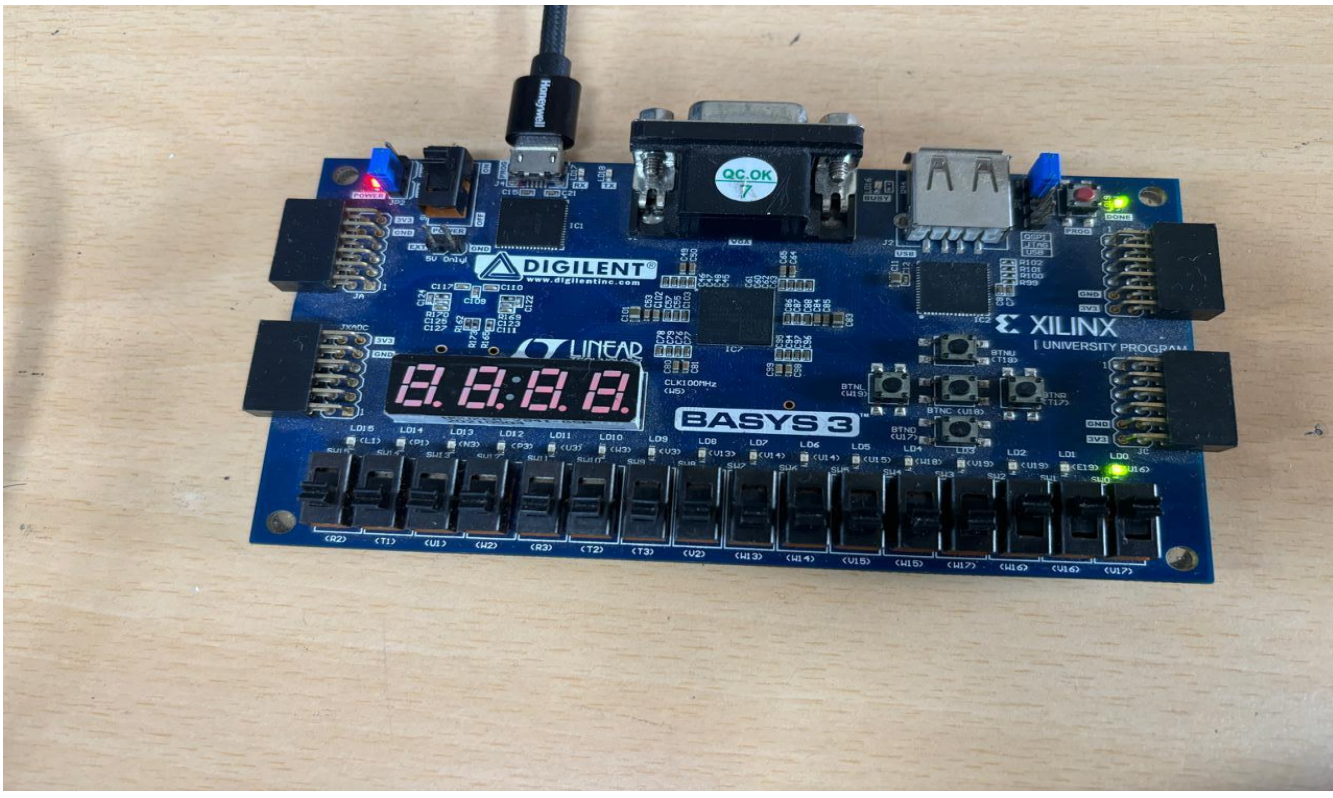


Board Snapshot

1.

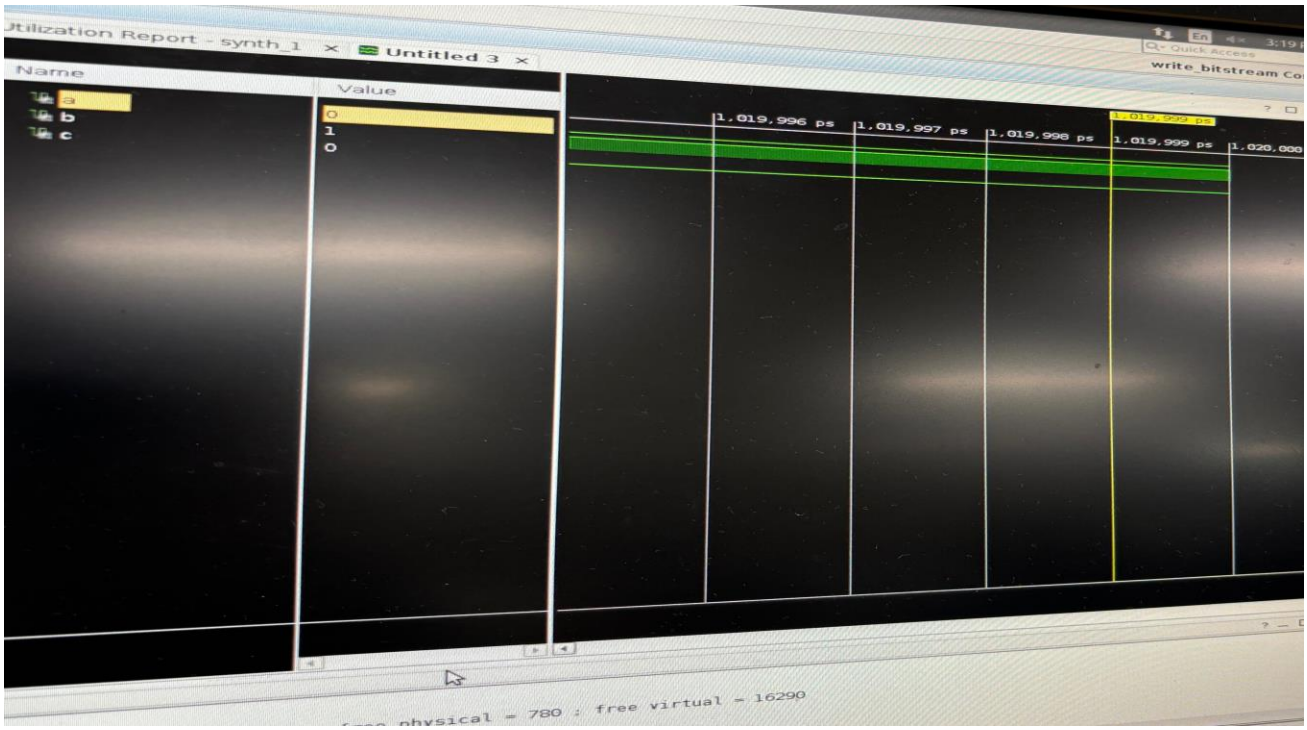


2.

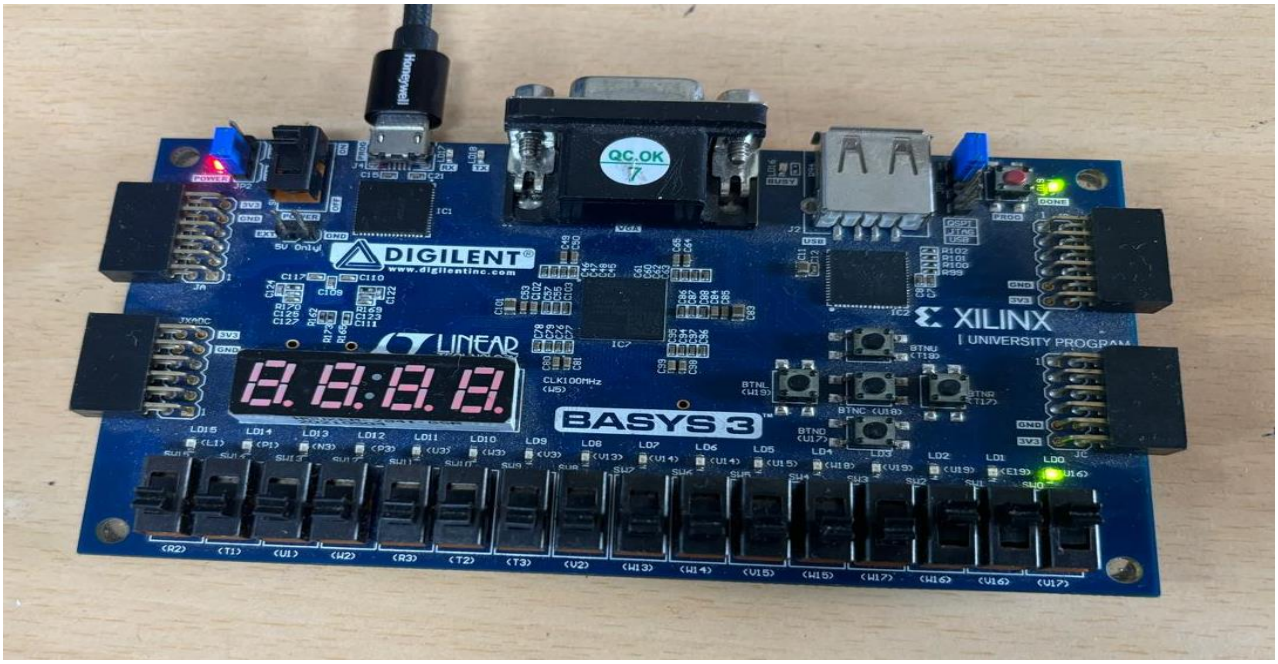


AND GATE

1.Simulation Snapshot

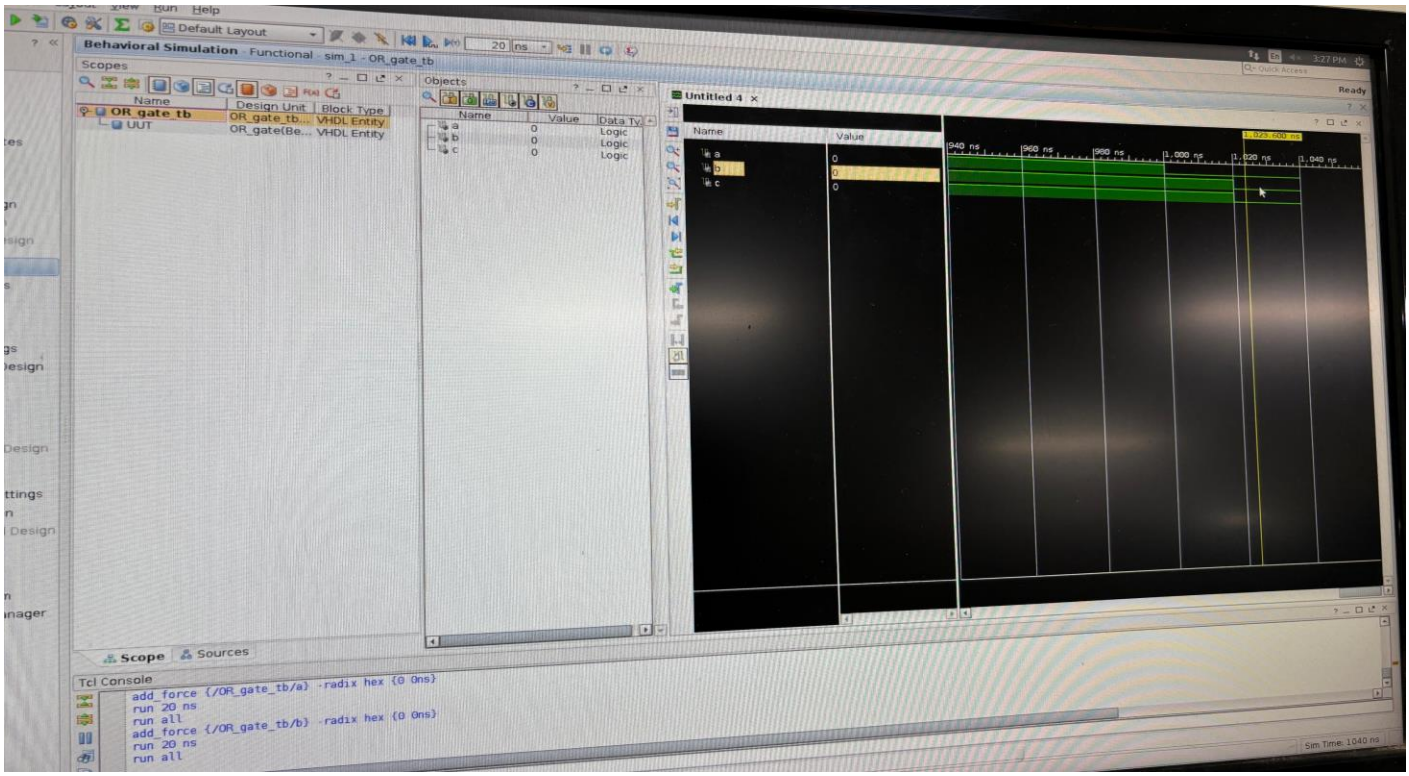


2.Board Snapshot

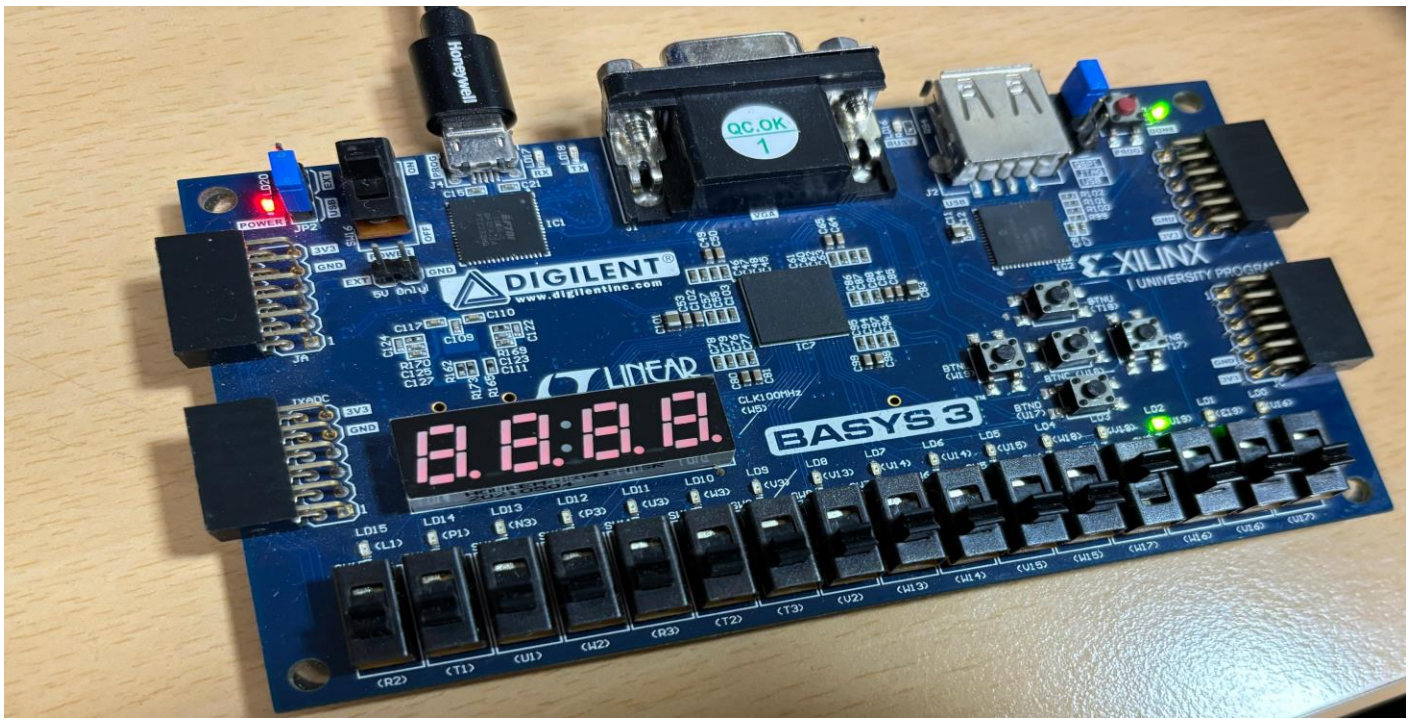


OR GATE

1.Simluation Snapshots

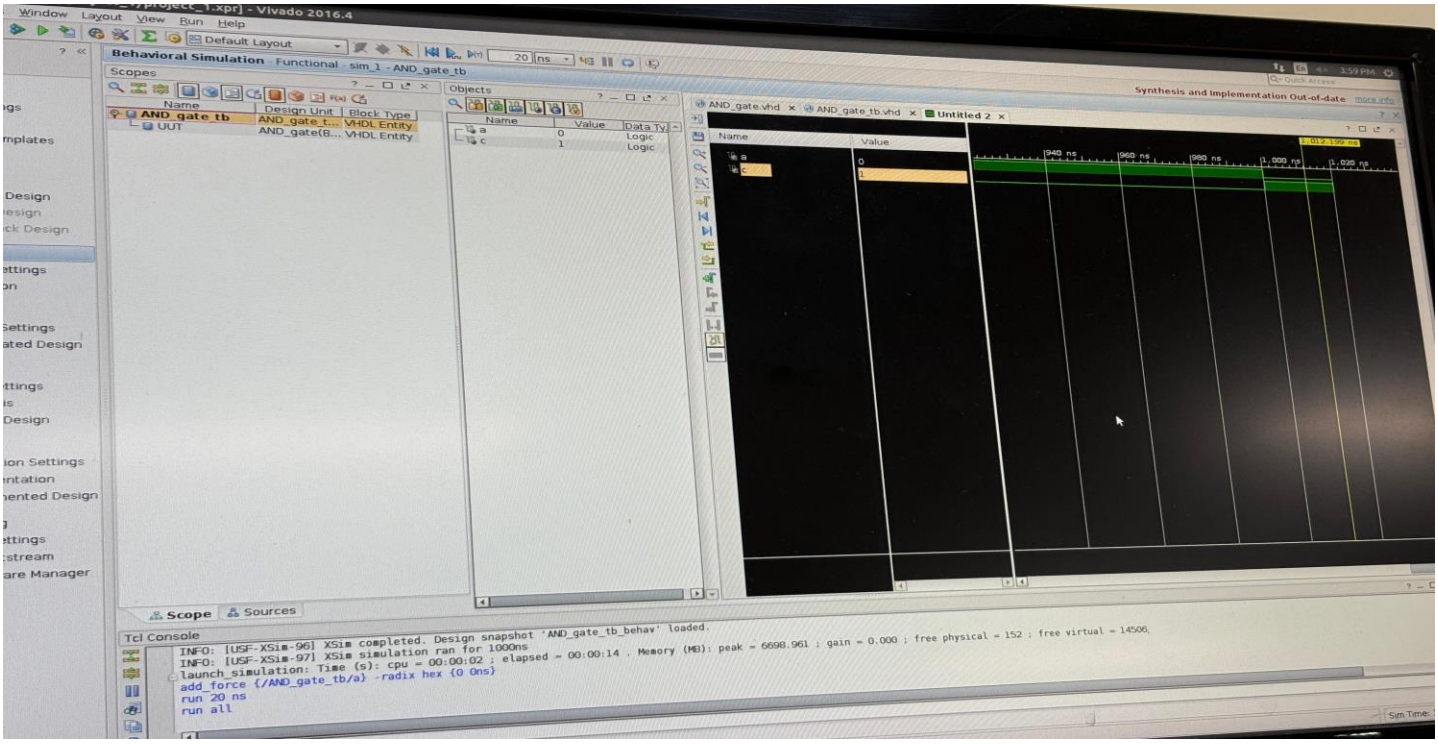


2.Board Diagram



NOT GATE

1.Simulation Snapshots



2.Board Snapshot