

## Lab 07

### Implementation of a 3 bit up and down counter using FPGA's source clock and Clock Divider



Spring 2025

Submitted by: **Mohsin Sajjad**

Registration No: **22pwsce2149**

Class Section: **A**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

A handwritten signature in black ink that reads "Mohsin Sajjad".

Student Signature: \_\_\_\_\_

Submitted to:

**Engr. Faheem Jan**

Month Day, Year (20 04, 2025)

Department of Computer Systems Engineering  
University of Engineering and Technology, Peshawar

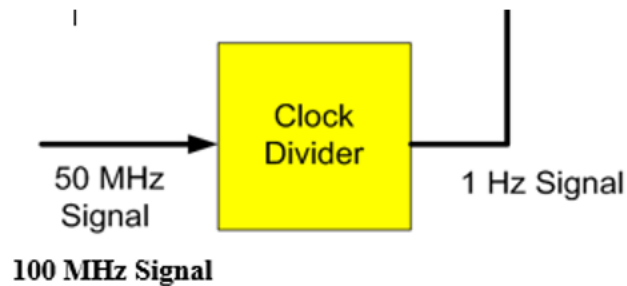
## Implementation of a 3 bit up and down counter using FPGA's source clock and Clock Divider

### Objective:

- To become familiarized with behavior level modeling
- To be able to implement sequential circuits using Verilog
- To Implement an 8 Bit Ring Counter on Spartan 6 FPGA starter kit.

**Block Diagram:** The Spartan 6 kit has a clock source of 100 Mhz. If we use it in applications like counters, the counter will count at an incredibly fast speed and we will not be able to see the output. Your task is to divide the 100 MHz frequency into a 1 Hz frequency. The Module Clock divider is responsible will be responsible for it.

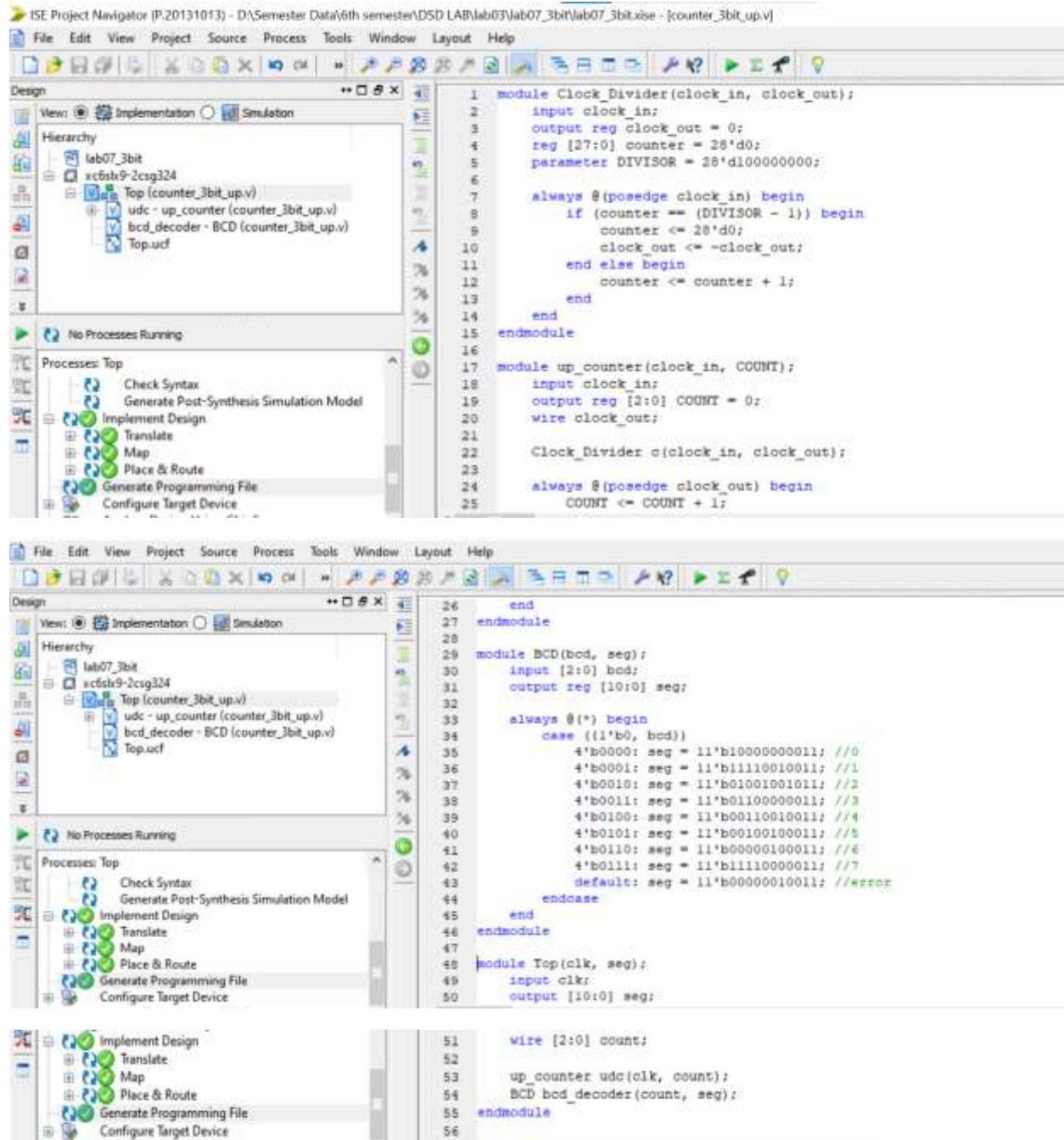
The functional detail of the 8-bit Ring Counter is shown in the following figures.



### Lab Task:

1-Implement 3 bit Up Counter using FPGA's clock source and clock divider

### **CODE:**



**OUTPUT:**



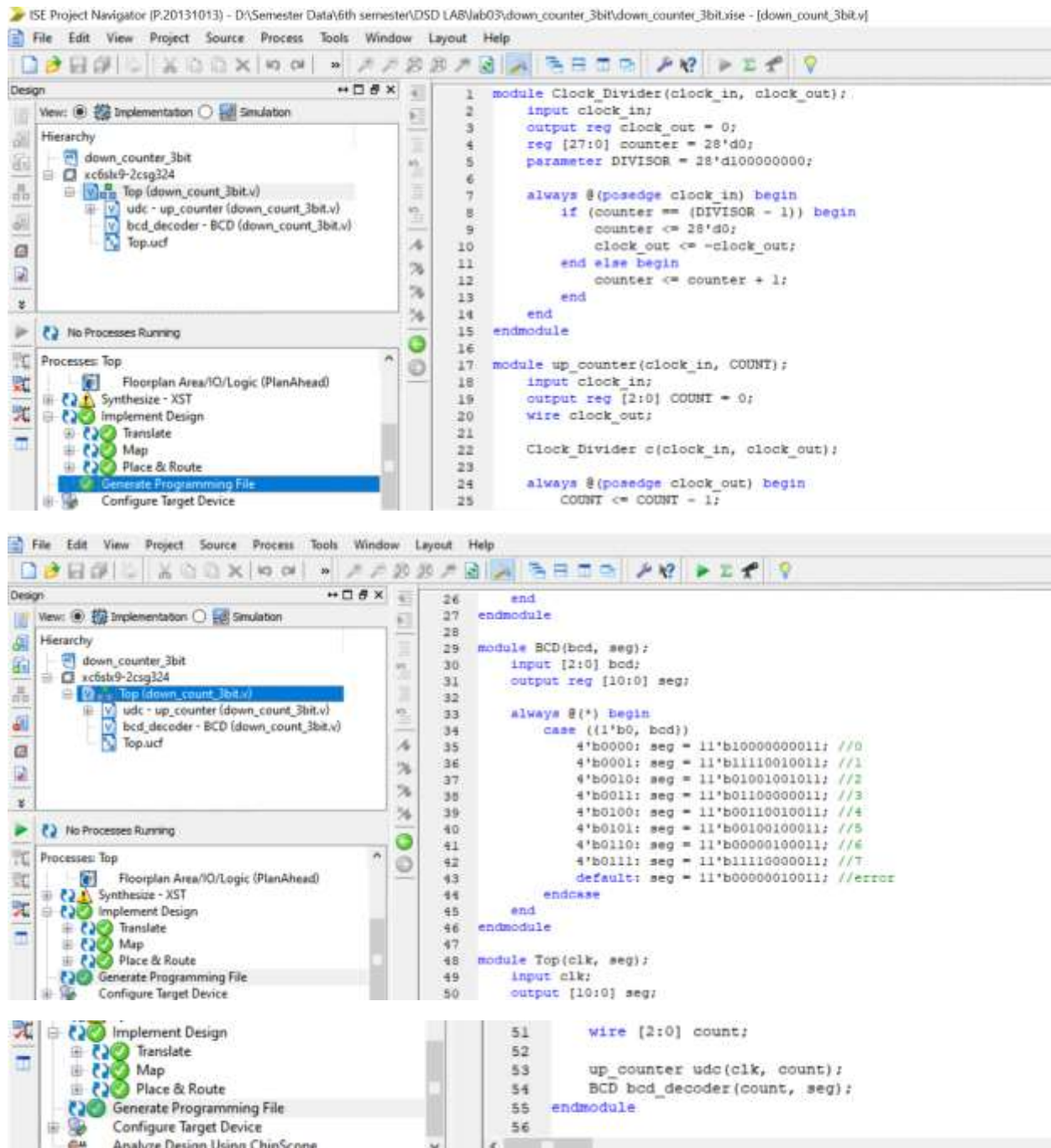
## Conclusion:

This Verilog code implements a simple digital system that counts from 0 to 7 using a clock divider and displays the count on a 7-segment display. The `Clock_Divider` module reduces the input clock frequency, the `up_counter` module increments the 3-bit count, and the `BCD` module decodes this count to drive the display. The `Top` module connects all components together.

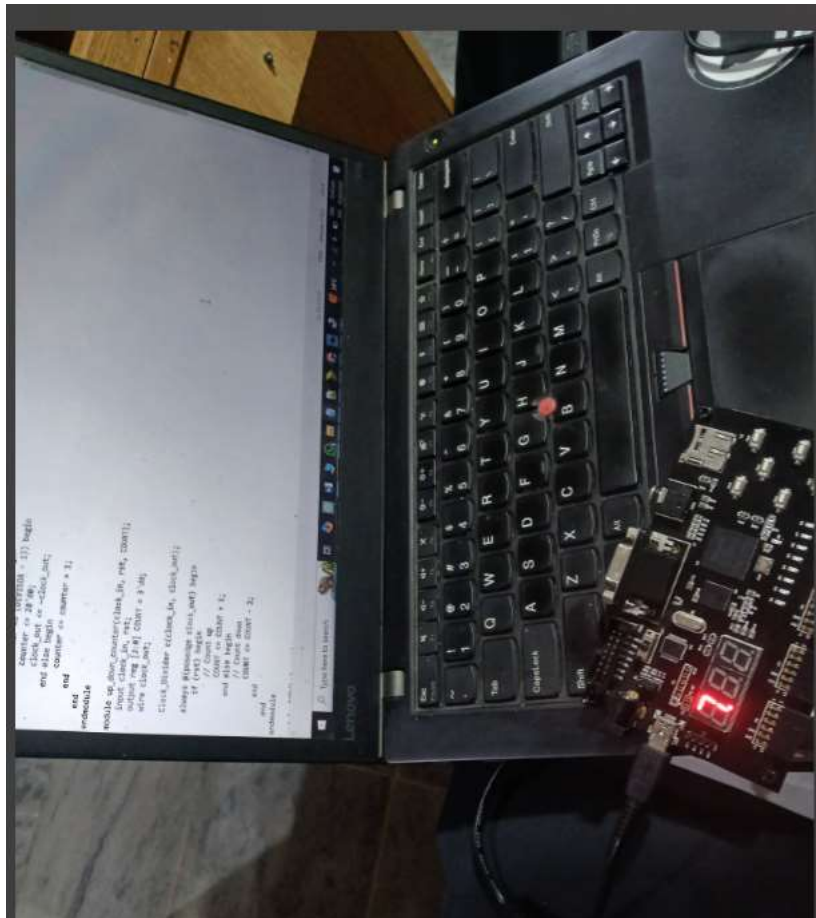
## TASK 02:

Implement 3-bit Down Counter using FPGA's clock source and clock divider

## CODE:



**OUTPUT:**



## Conclusion:

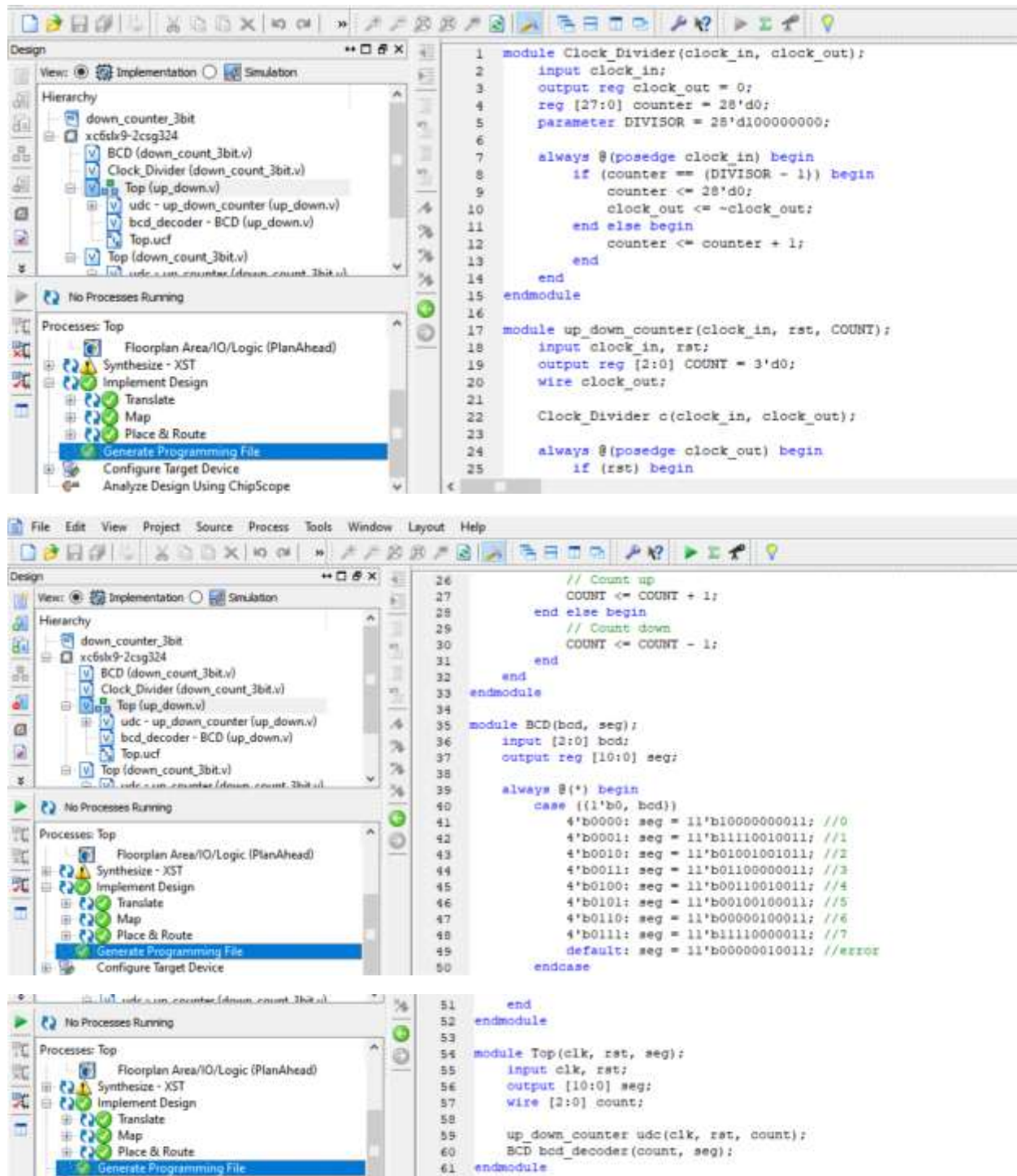
This updated Verilog design implements a **down counter** that counts from 7 to 0 repeatedly. It uses a `Clock_Divider` to slow the input clock, an `up_counter` module (actually counting down), and a `BCD` module to display the count on a 7-segment display. The `Top` module integrates all components into one system.

## TASK 03:

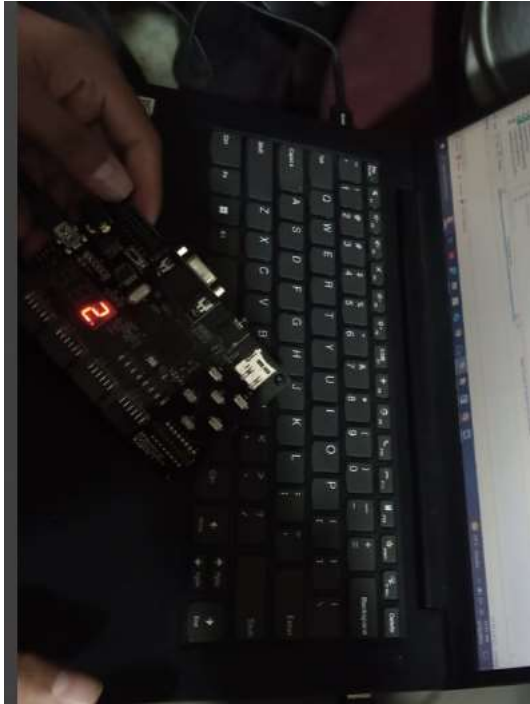
Implement 3-bit Up and Down Counter using RST if the RST is high it should Count Up if it is low it should Count Down and display the count in the seven segment display.

## CODE:





**Output:**



## Conclusion:

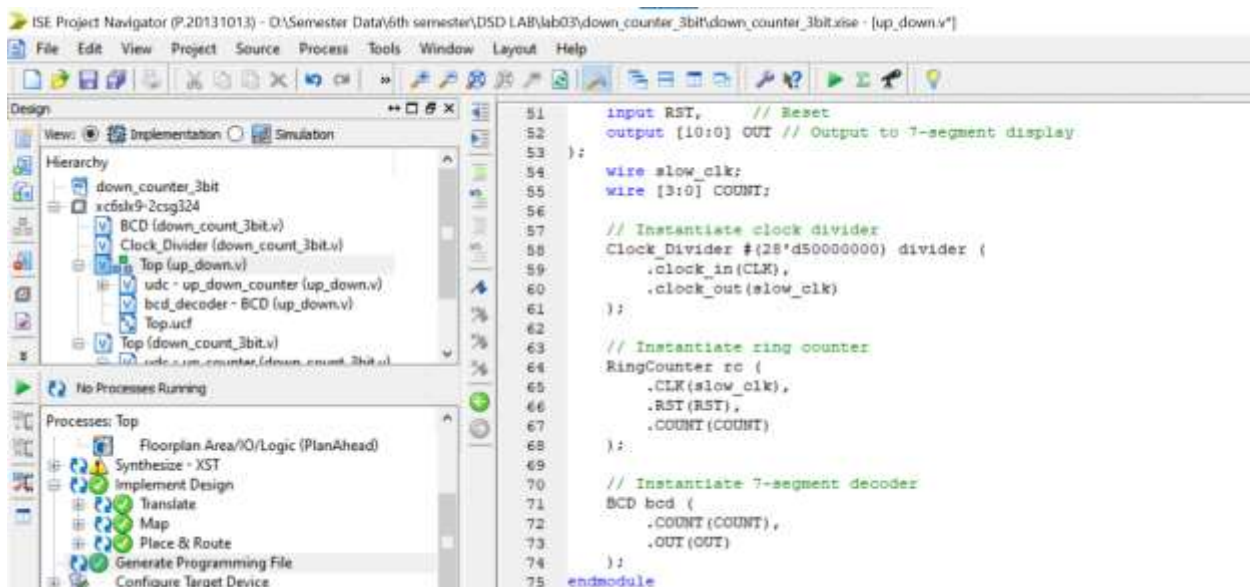
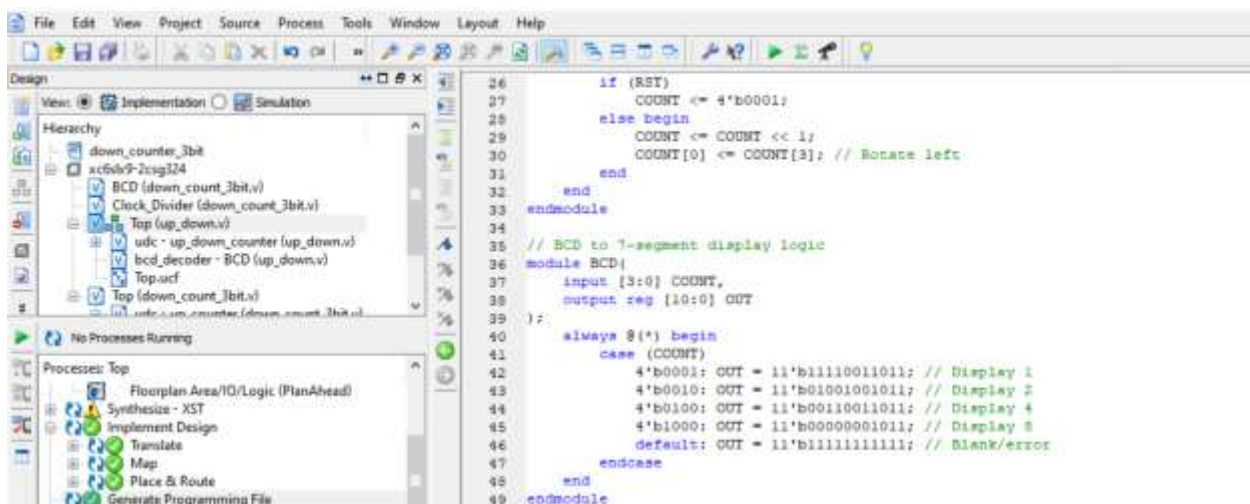
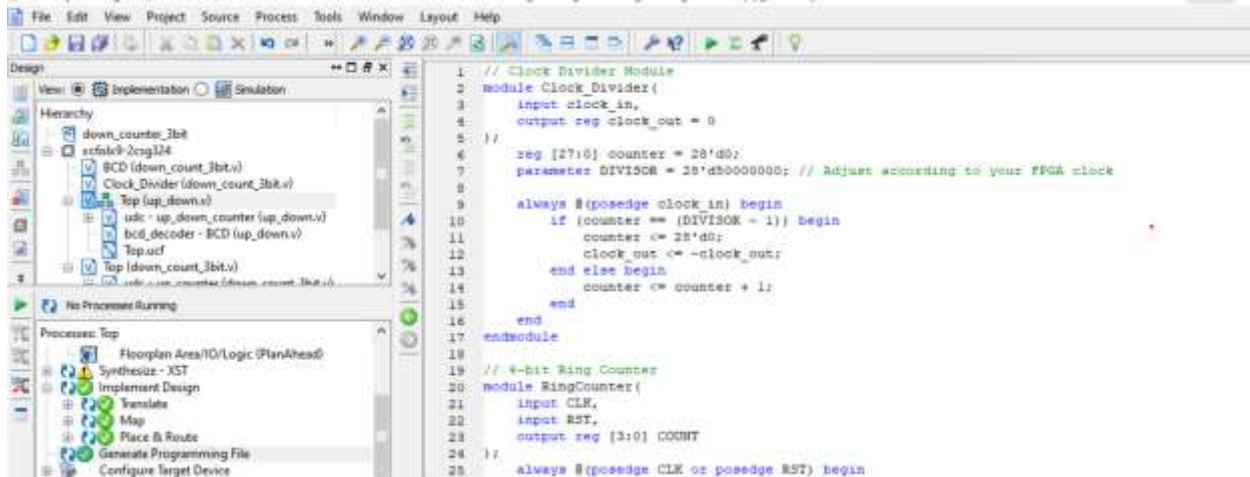
This Verilog code implements a 3-bit up/down counter system with display. A `Clock_Divider` slows the clock signal, and the `up_down_counter` counts up when `rst` is high and down when `rst` is low. The `BCD` module decodes the count to a 7-segment format. The `Top` module integrates all components, making the count direction controllable by the `rst` input.

## Task 04:

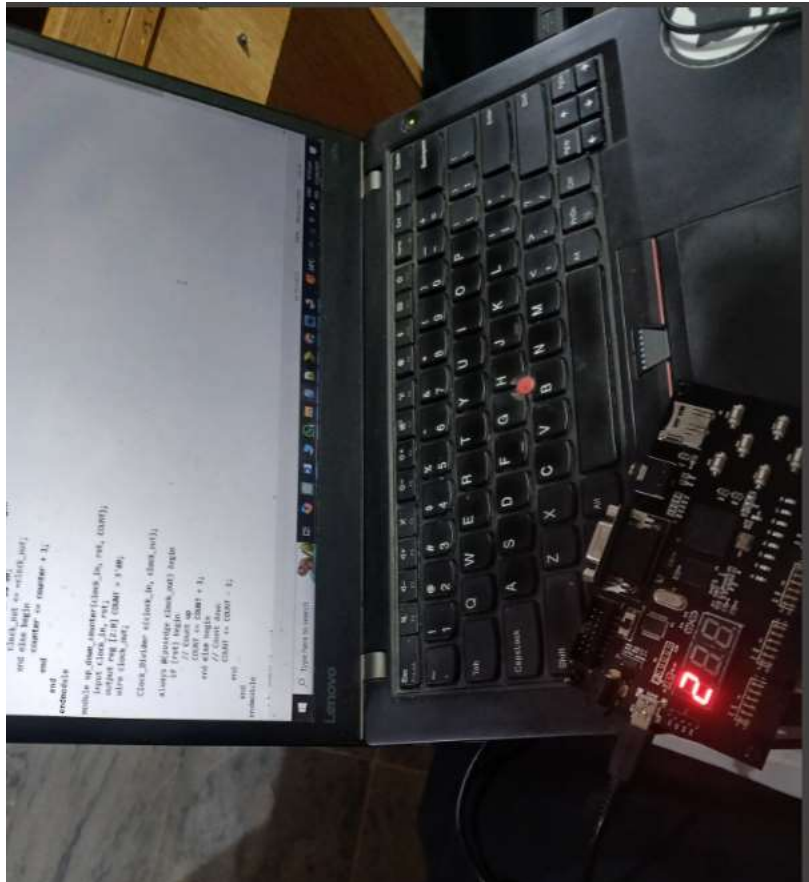
Implement Lab 6 using FPGA's clock source.

## CODE:





## Output:



## Conclusion:

This Verilog design creates a 4-bit ring counter system using a `Clock_Divider` to slow down the input clock, a `RingCounter` that shifts a single high bit through 4 positions (rotating left), and a `BCD` module that converts the counter output into a 7-segment display pattern. The `Top` module ties everything together, and the count resets to 0001 when `RST` is high.