ECE 451-LAB2

Fall 2019

Ram Rohit Gannavarapu

Contents

[List of Figures 1](#_Toc21417630)

[1. List of items completed 2](#_Toc21417631)

[2. Lab Procedures 2](#_Toc21417632)

[3. Schematics/Verilog models 3](#_Toc21417633)

[4. Simulation Results 5](#_Toc21417634)

[5. Analysis and Explanations 6](#_Toc21417635)

[6. Conclusion 6](#_Toc21417636)

[7. Answers to questions posted in the lab 6](#_Toc21417637)

# List of Figures

[Figure 1: Logic for Light controller 3](#_Toc21417638)

[Figure 2: Schematic for a Subway Controller with Tri-state Buffers 4](#_Toc21417639)

[Figure 3 Schematic for a Subway Controller test bench 4](#_Toc21417640)

[Figure 4: Simulation Result for D=0 (Rigt to Left Direction) 5](#_Toc21417641)

[Figure 5: Simulaton Result for D=1(Left to Right Direction) 5](#_Toc21417642)

# List of items completed

* 1. Truth Table of Subway controller (Light Controller)
  2. Logic Design of Subway signal controller on Cadence
  3. Design of Tri-State Buffer

# Lab Procedures

Truth Table for Light Controller

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **D** | **P** | **Ai** | **Bi** | **R** | **G** | **Y** | **A0** | **B0** |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |

The lab design starts with designing of a Tri State Buffer. Then followed by designing a Light controller which is responsible in deciding the states of the lights (Red, Yellow and Green) based on the inputs and Aiin and Bin. It also Outputs Aout and Bout which are later connected to the next/previous stations based on the value D. To achieve this we have used a combination of Tri-state Buffers to decide if Aout goes to Ai or A0 and similarly Bout goes to either Bi or B0.

# Schematics/Verilog models

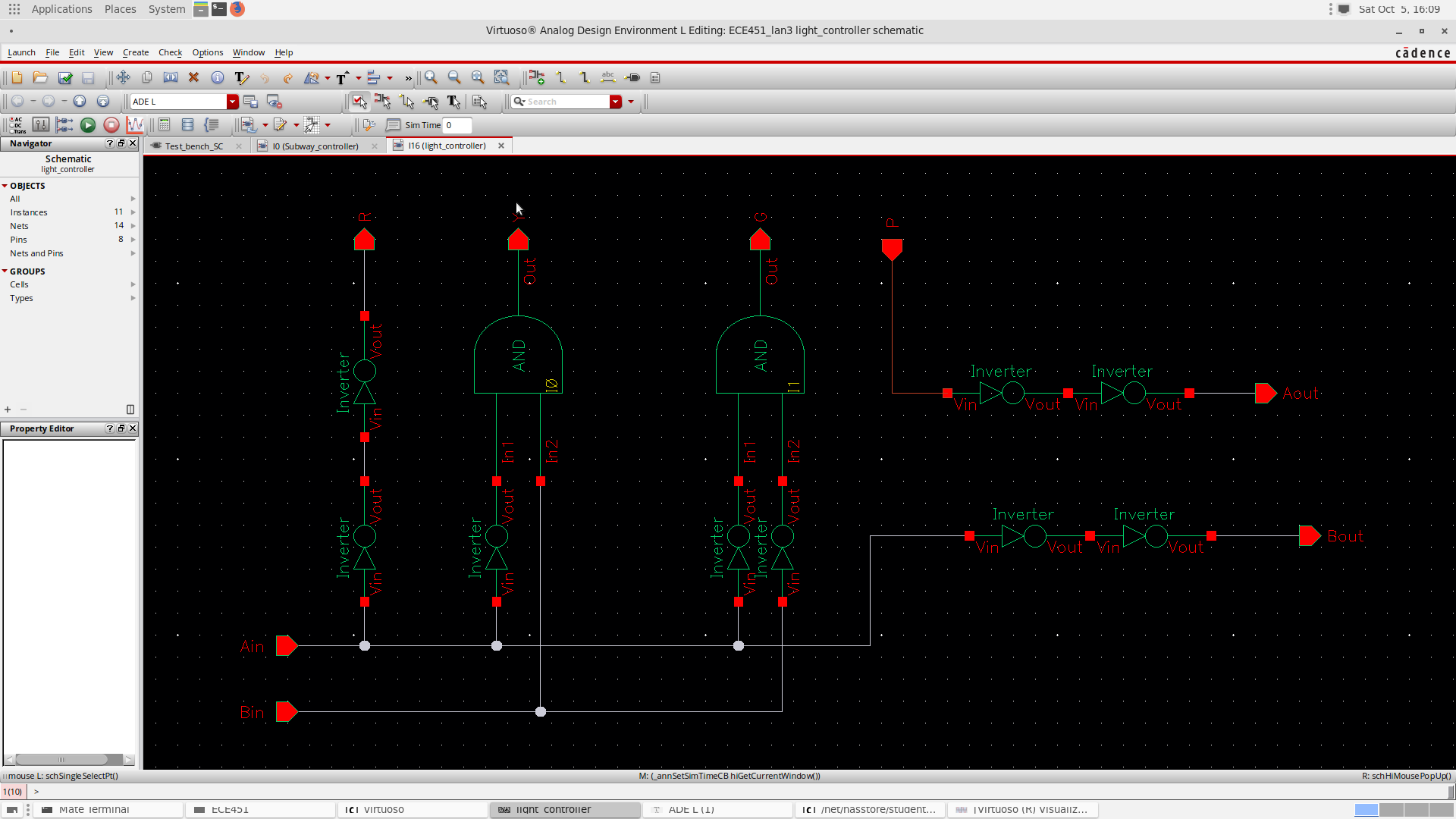


Figure 1: Logic for Light controller

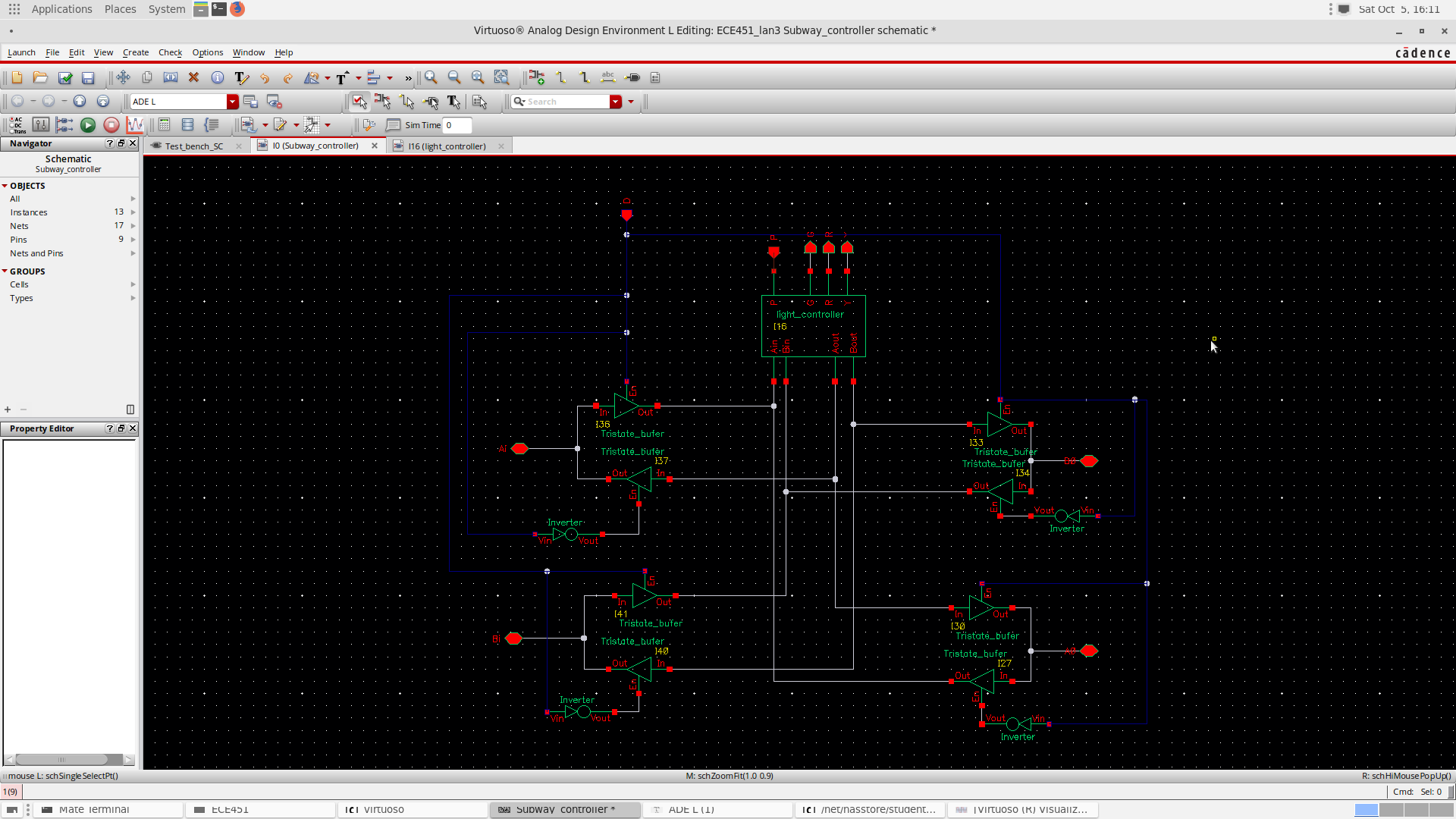


Figure 2: Schematic for a Subway Controller with Tri-state Buffers

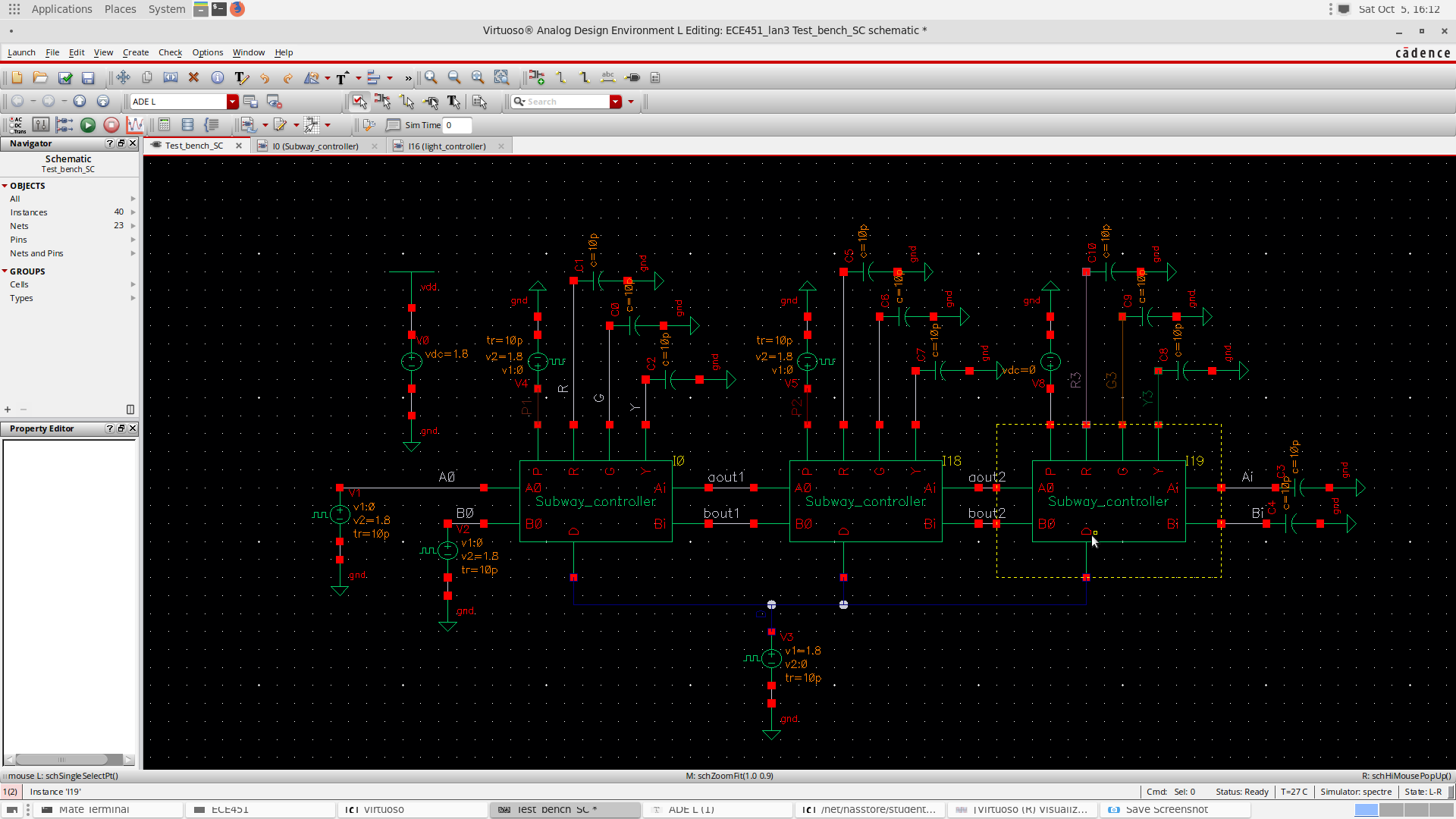


Figure 3 Schematic for a Subway Controller test bench

# Simulation Results

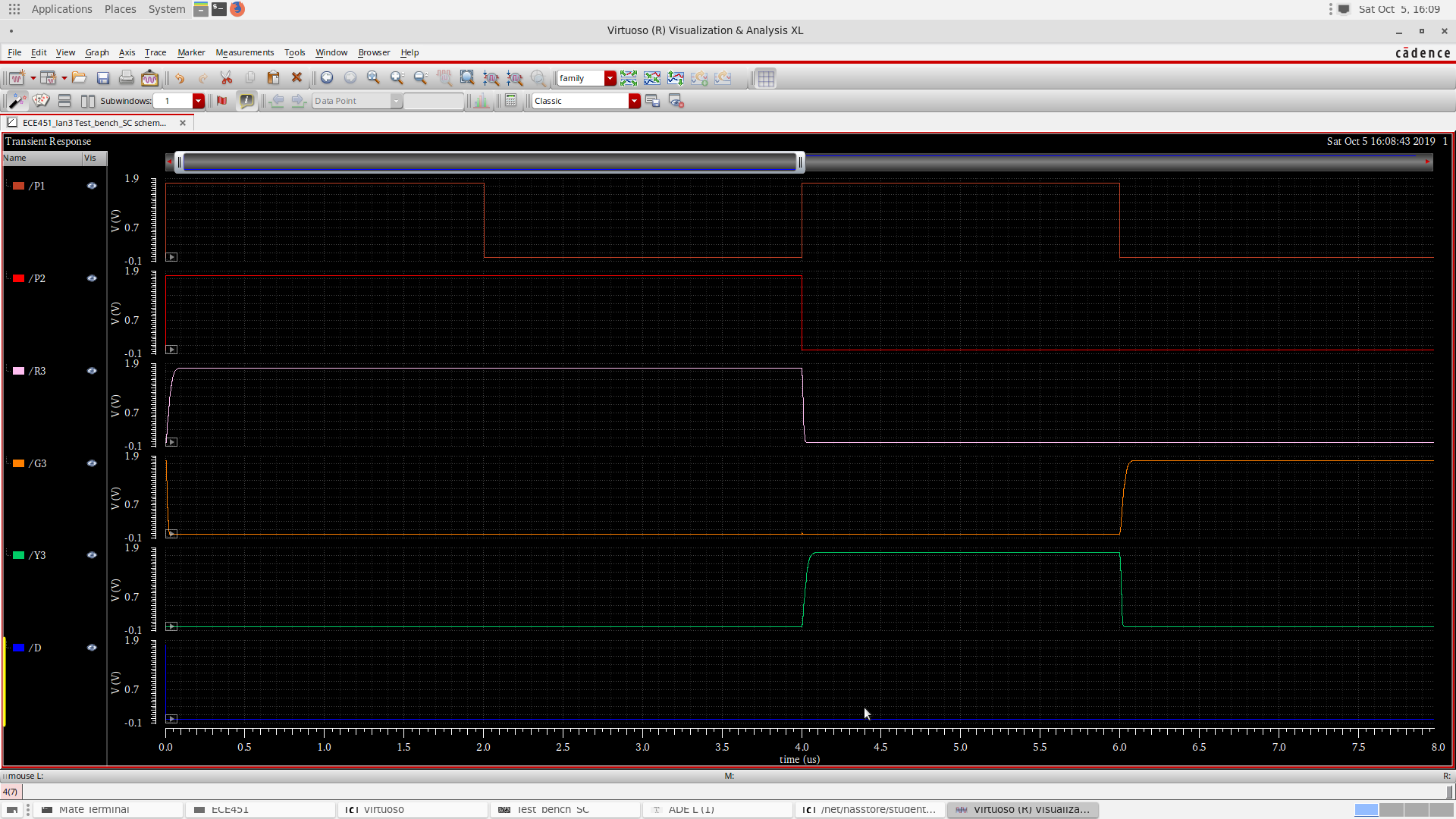


Figure 4: Simulation Result for D=0 (Rigt to Left Direction)



Figure 5: Simulaton Result for D=1(Left to Right Direction)

# Analysis and Explanations

It is evident from the simulation results that Subway Controller shows Red signal when there is train in the next station. The station shows a yellow signal when there is a Train on track of the station next to the adjacent station. The subway controller shows green when there is not train on either of the stations.

# Conclusion

The Subway controller is worked as expected in both directions i.e left – Right (D=1) and Right – left(D=0)

# Answers to questions posted in the lab

Q) What is a static 1-hazard and a static 0-hazard?

A) A static hazard is a condition in which when there is a change in one of the inputs the outputs changes momentarily before becoming stable. There are 2 types of static hazards.

Static 1- hazard: the output changes from 1 – 0 – 1 instead of staying a constant 1

Static 0-hazard: the output changes from 0 – 1 – 0 instead of staying a constant 0

Q) Discuss how you remove a static 1-hazard and a static 0-hazard.

A) Static 1 hazards can be usually solved by adding a redundant term which includes the adjacent 1’s which are not already covered in the initial Boolean equation(SOP form). Static 0 hazards can be solved by similar method where instead of using a SOP equation we use a POS expression and add redundant term that covers adjacent 0’s.