**JAYPEE INSTITUTE OF INFORMATION** **TECHNOLOGY, NOIDA**

Summer Internship Report

****

|  |  |  |
| --- | --- | --- |
| **Submitted To:** | **Submitted By:** |  |
| **Dr. RishiBrind Upadhyay** | **Enrolment No.** | **Name** |
| **Dr. Shivani** | **22102236** | **Priyanshu Aggarwal** |

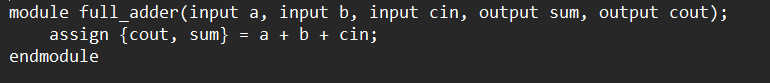
VLSI design ﬂow explained RTL to GDS II

1. **RTL (Register-Transfer Level) Design**
   * Write HDL (Hardware Description Language) code in Verilog or VHDL to describe the digital circuit.
   * RTL code describes the circuit's behavior in terms of registers, combinational logic, and clocked operations.
2. **Synthesis**
   * Convert RTL code into a netlist using a synthesis tool (e.g., Design Compiler).
   * The netlist contains gates and connections.
   * Optimize the netlist for area, power, and timing.
3. **DFT (Design for Testability)**
   * Insert scan chains and test logic to improve testability.
   * Ensure the design is testable and can be debugged.
4. **Place and Route**
   * **Floorplanning**: Plan the placement of blocks and IOs on the chip.
   * **Placement**: Place the standard cells (logic gates) on the chip.
   * **Clock Tree Synthesis**: Design the clock distribution network.
   * **Routing**: Connect the placed cells with wires.
5. **Timing Analysis and Optimization**
   * Verify the design meets timing constraints (setup and hold times).
   * Optimize the design for timing, area, and power.
6. **Physical Veriﬁcation**
   * **DRC (Design Rule Checking)**: Verify the design meets foundry-speciﬁc design rules.
   * **LVS (Layout Versus Schematic)**: Verify the layout matches the netlist.
   * **Antenna Effect**: Check for antenna effects and ﬁx them.
7. **GDSII Generation**
   * Create the ﬁnal GDSII ﬁle for tape-out.
   * The GDSII ﬁle contains the physical layout of the design.

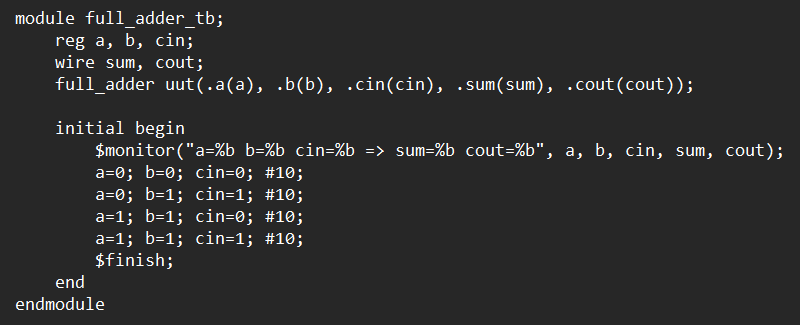
# WEEK 1

**Cadence Entry and Simulations Verilog Codes and Waveforms**

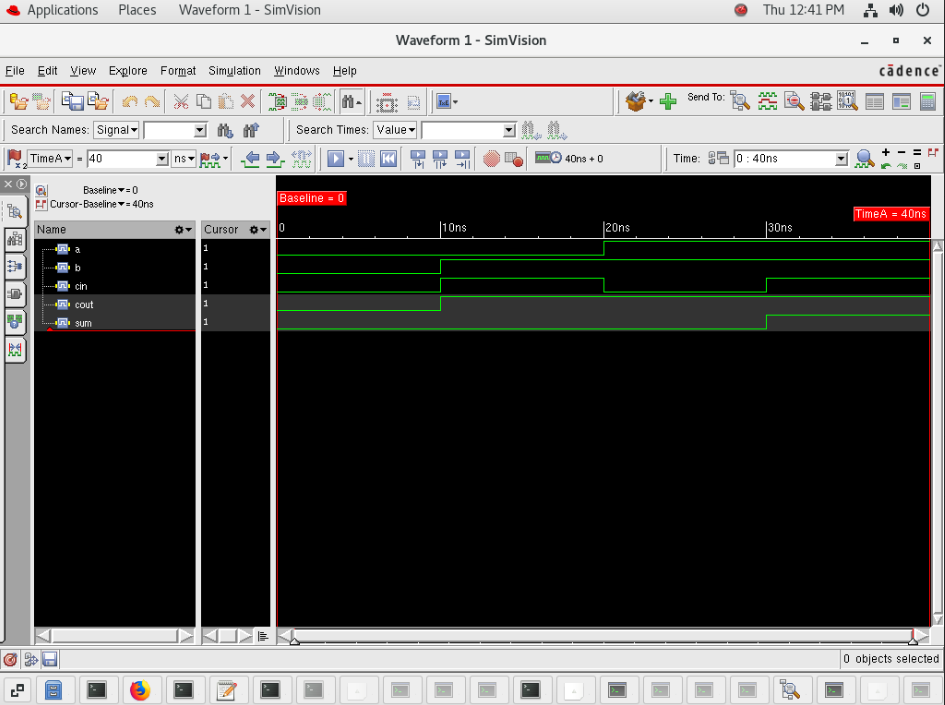
## FULL ADDER:



## Test bench:



**Waveform:**

****

**Traffic Light Controller:**

## 

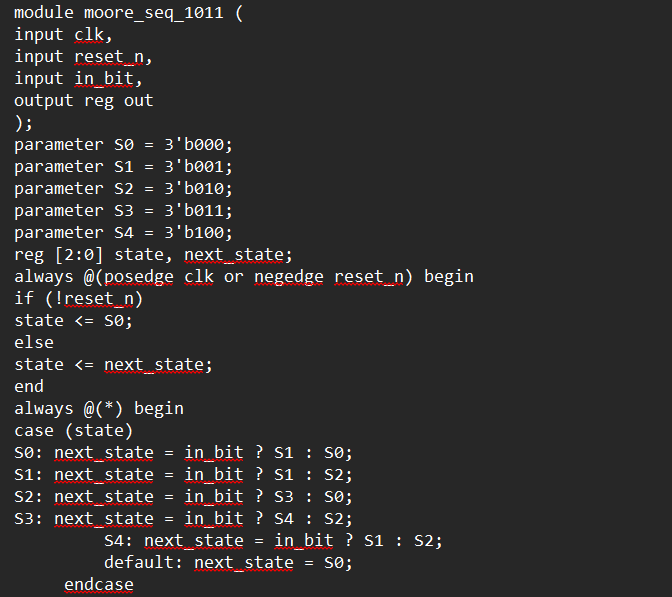
## Test Bench:

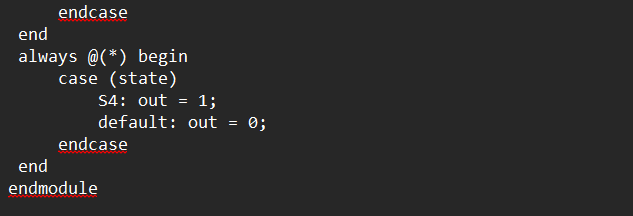
## 

## Waveform:

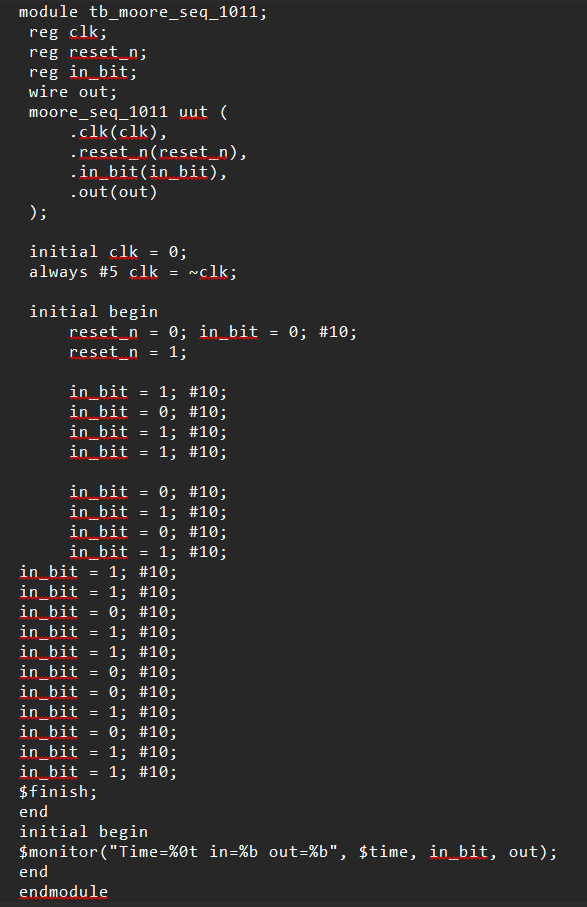
## 

**MEALY SEQUENCE DETECTOR:**

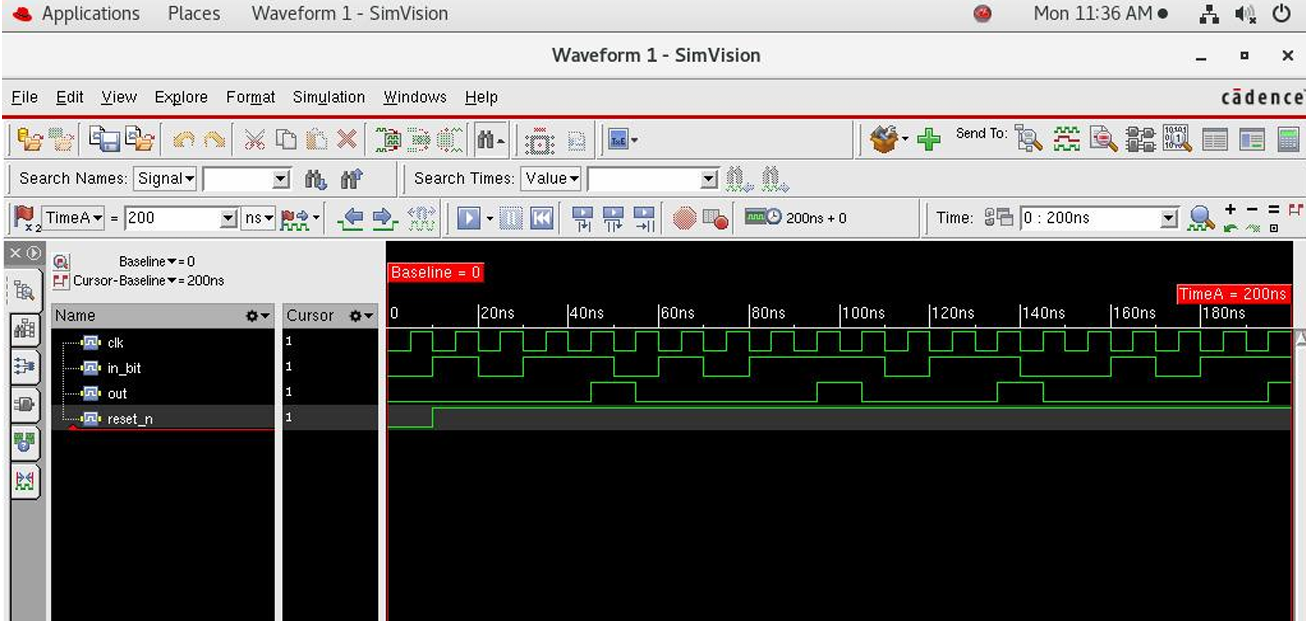
****

****

**Test Bench:**

****

**Waveform :**

****

# WEEK 2

# Synthesis

# FULL ADDER

# 

# TRAFFIC LIGHT CONTROLLER

# 

# JK Flip Flop

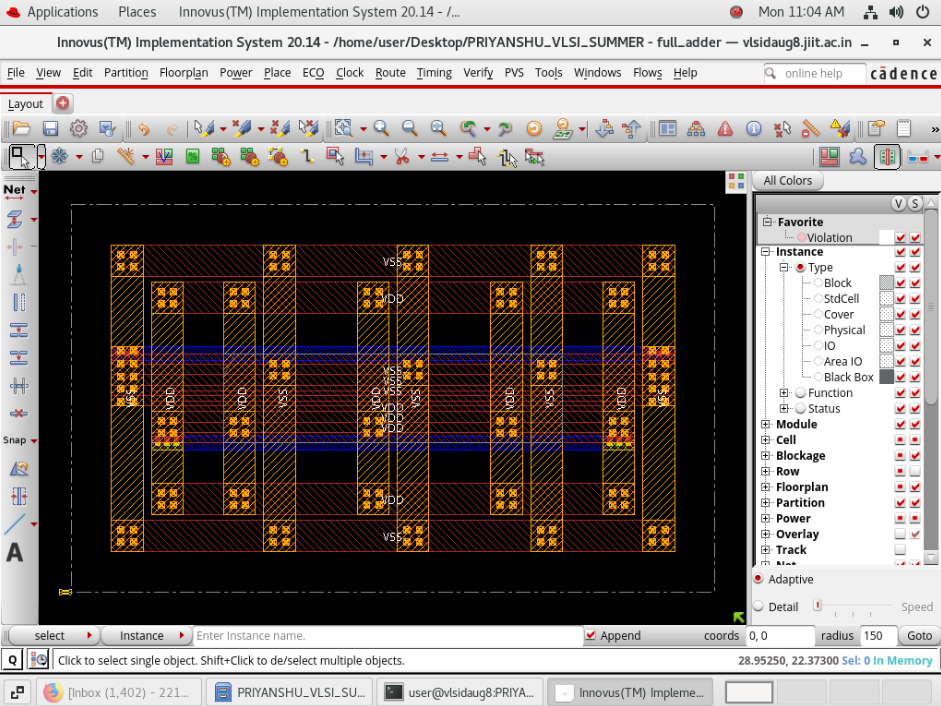
# 

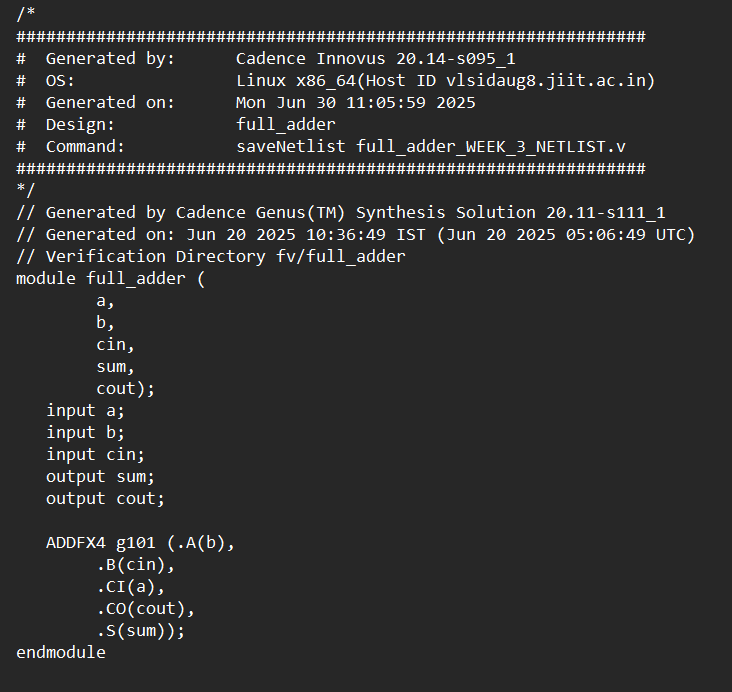
Similarly, Simulated the Synthesis of 10 Digital circuits, including Combinational as well as Sequential Circuits

# WEEK 3

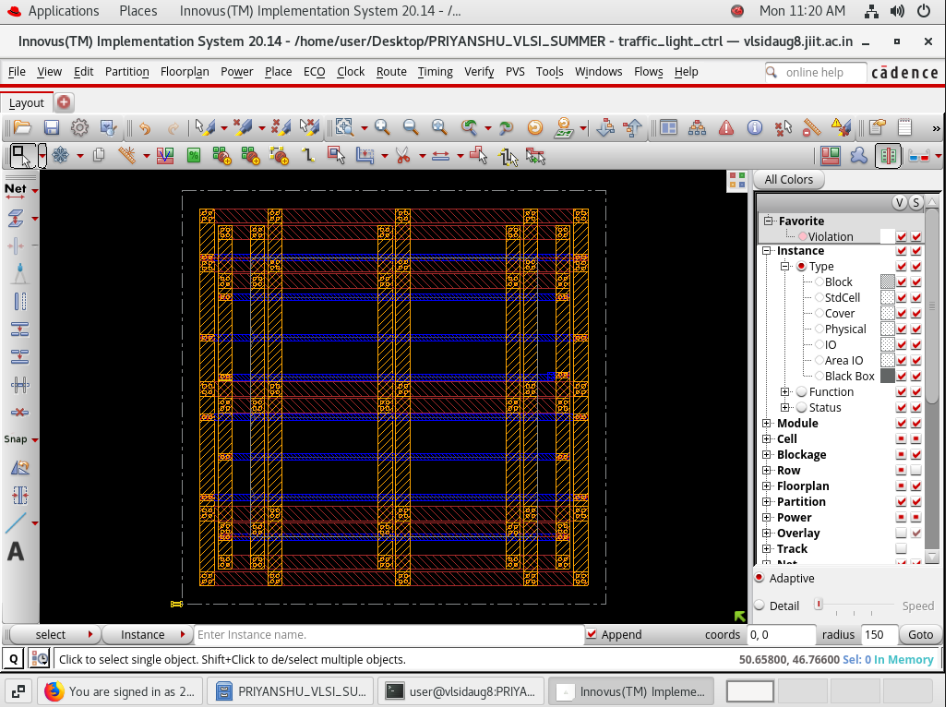
**Floor Planning and Power Planning**

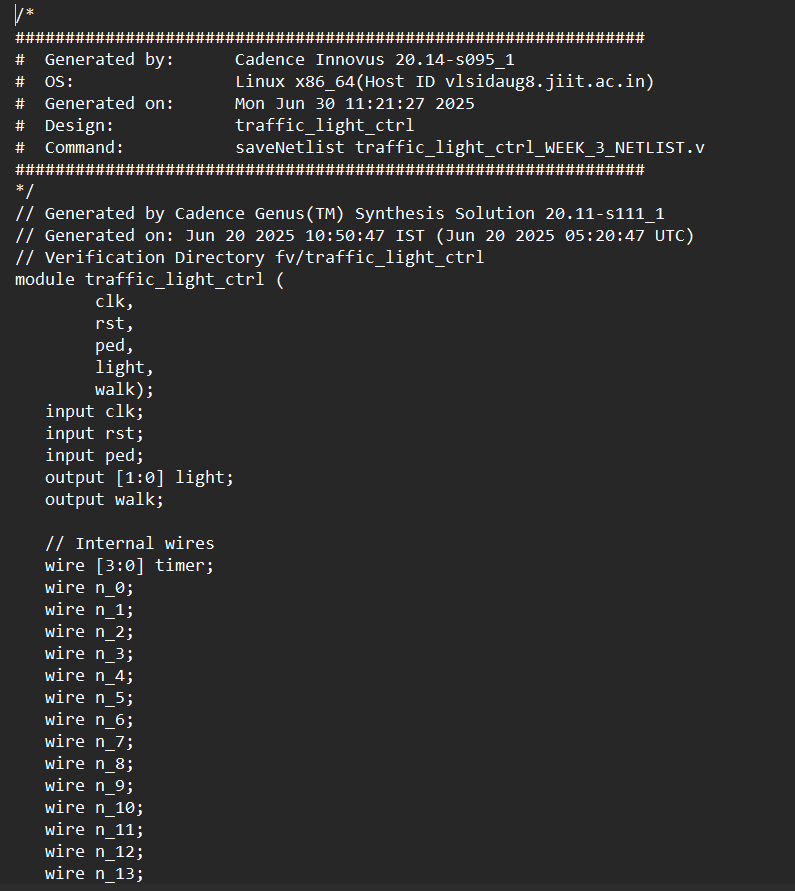
**FULL ADDER**

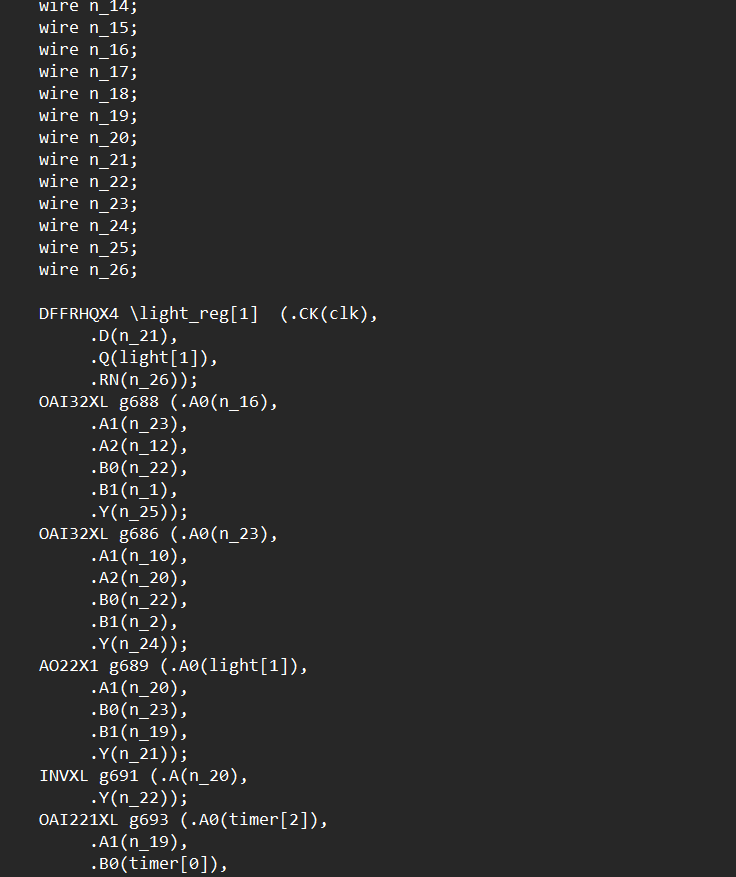
****

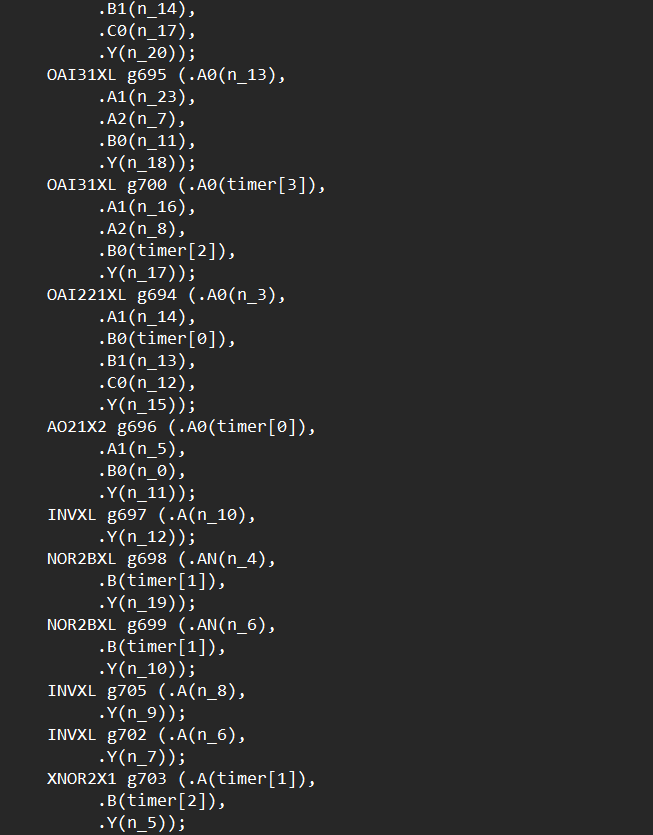
****

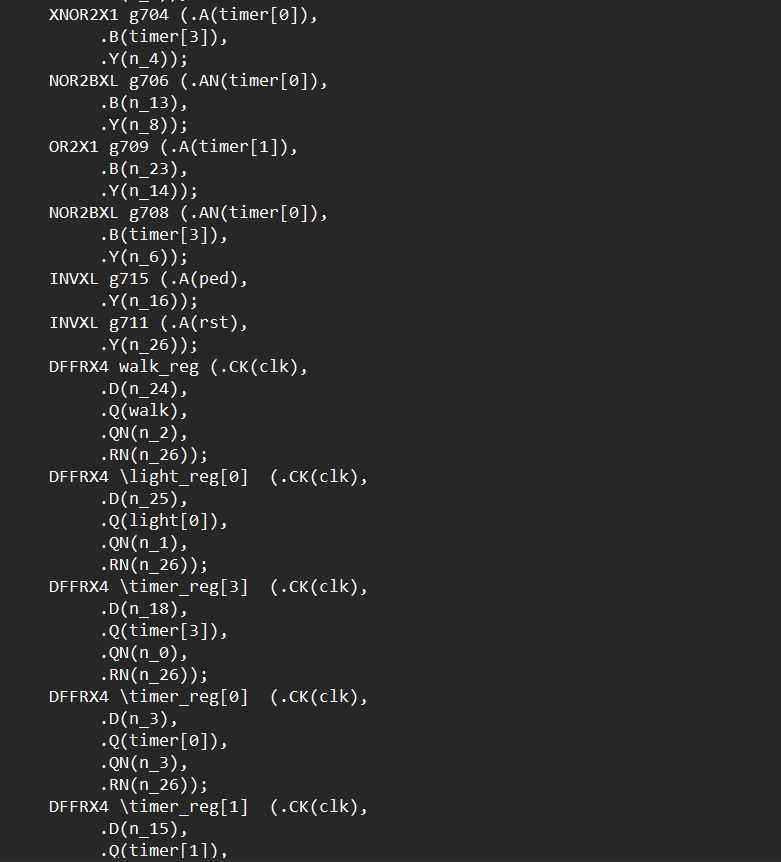
**TRAFFIC LIGHT CONTROLLER**

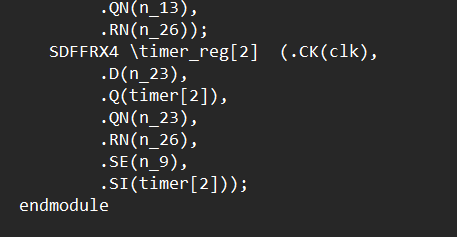
****

****

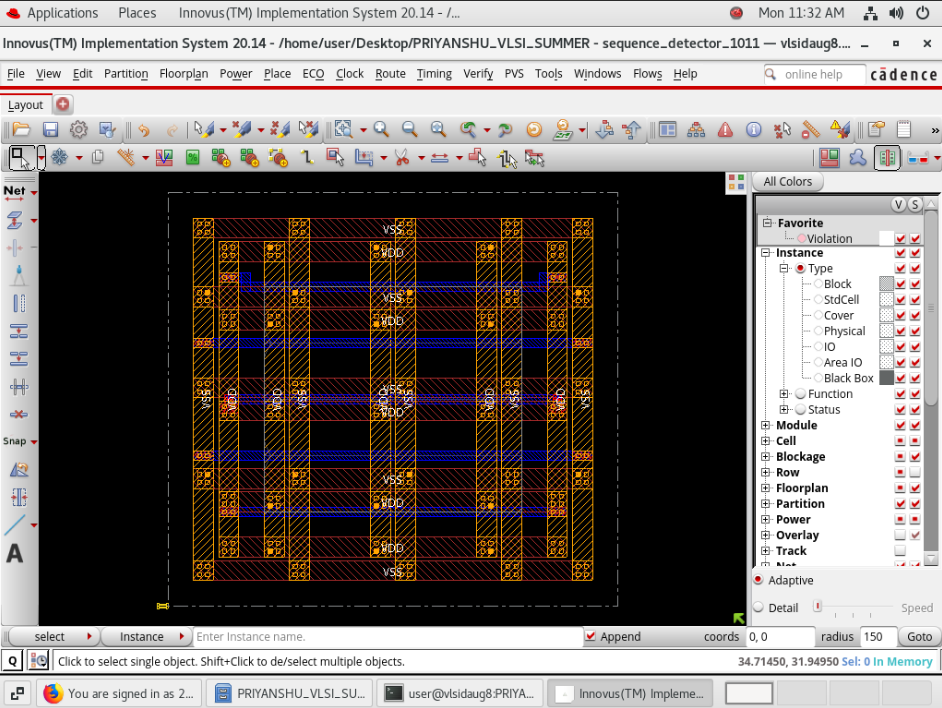
****

****

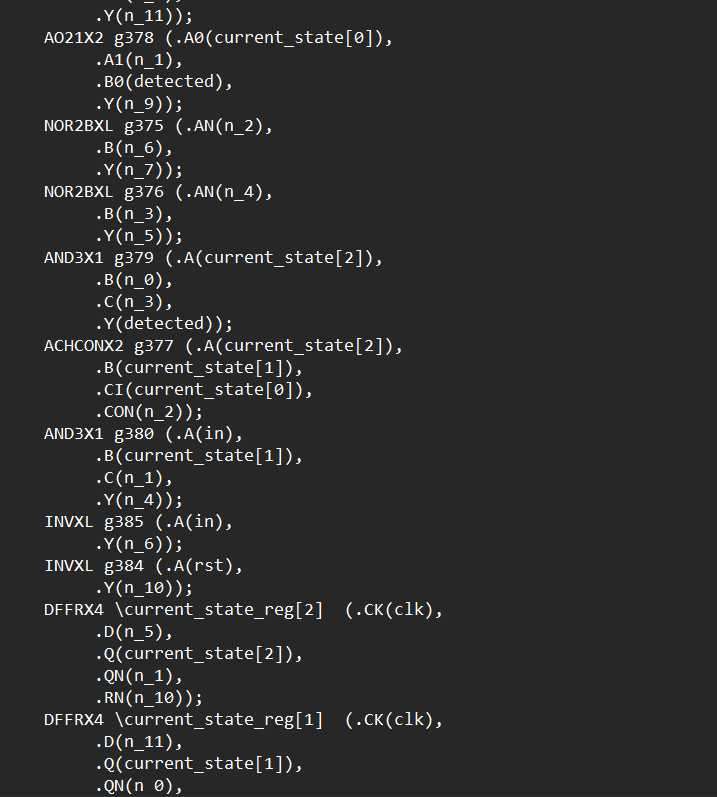
****

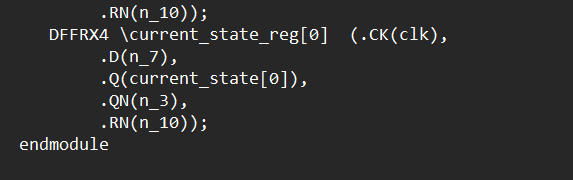
****

**Mealy Sequence Detector**

****

****

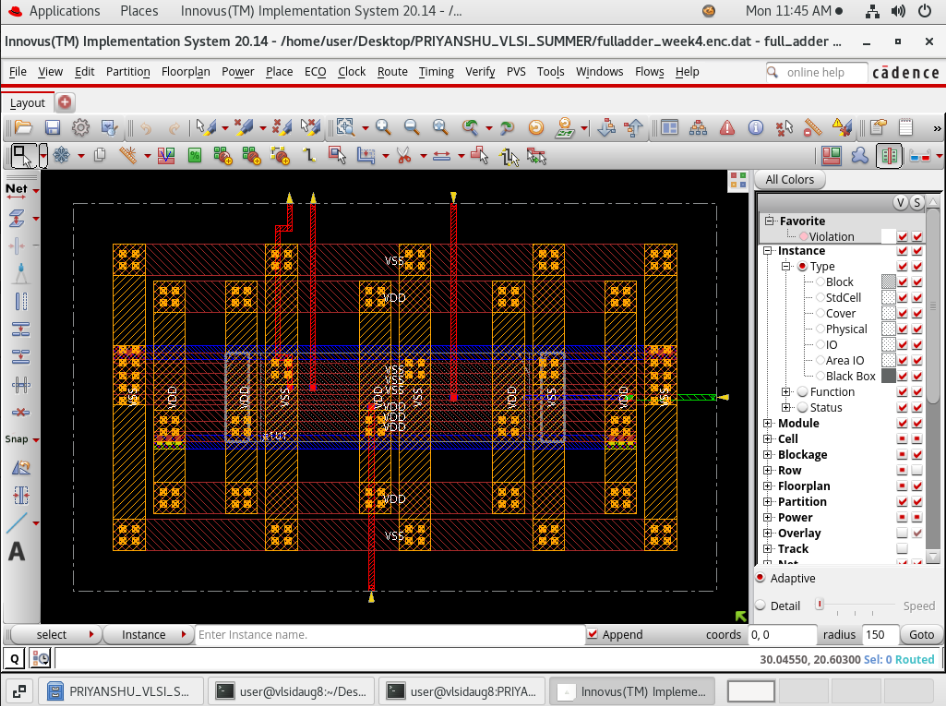
****

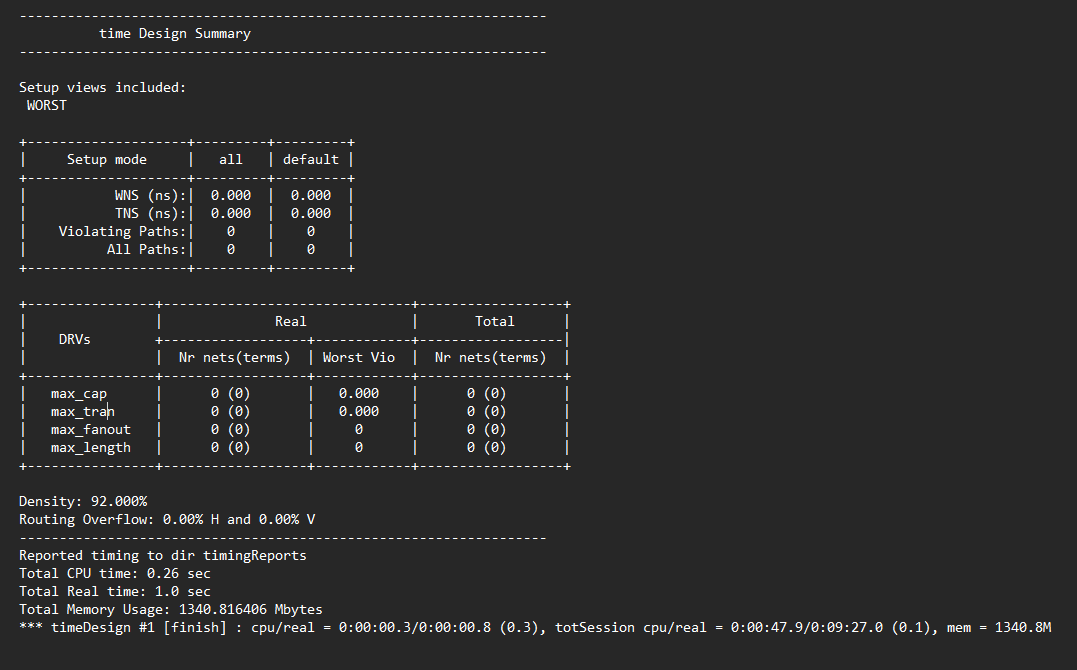
****

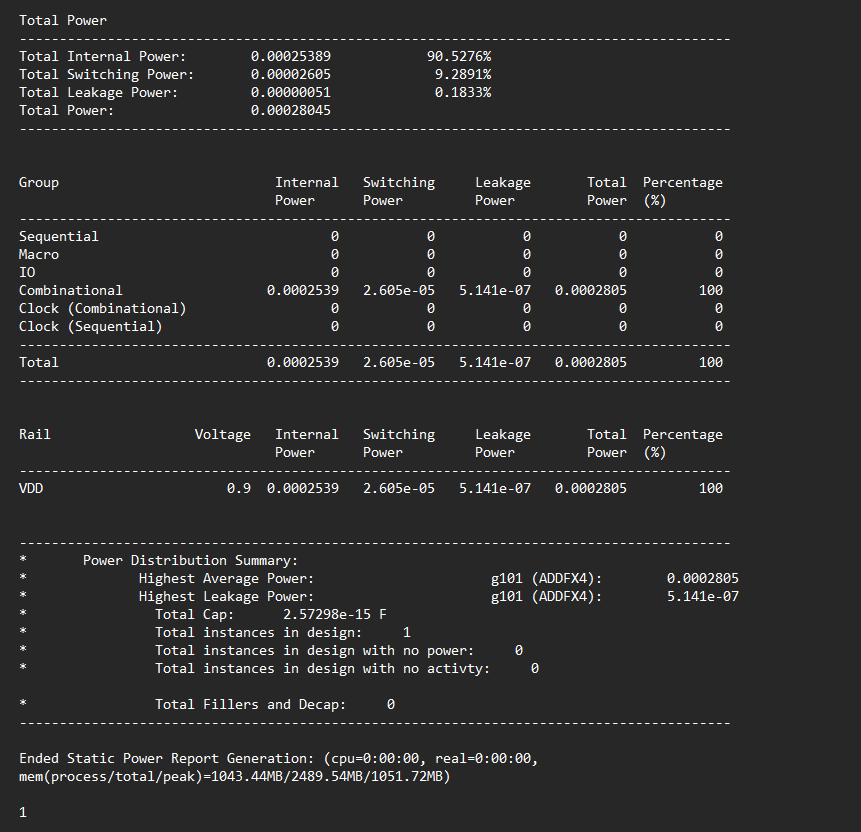
# WEEK 4

**Placement, CTS, Routing, RC Extraction**

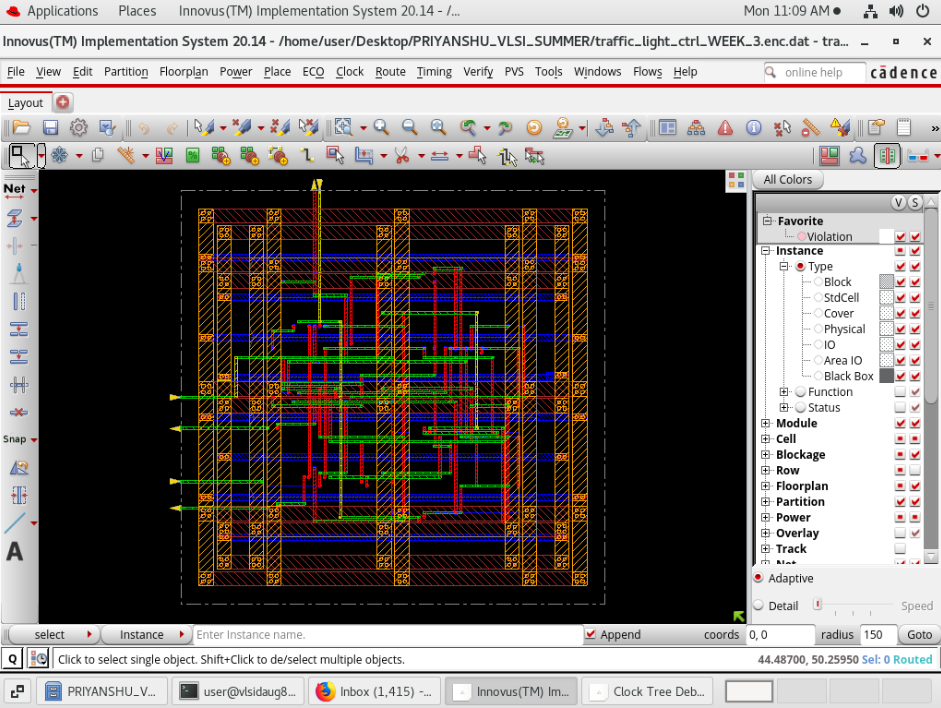
**FULL ADDER**

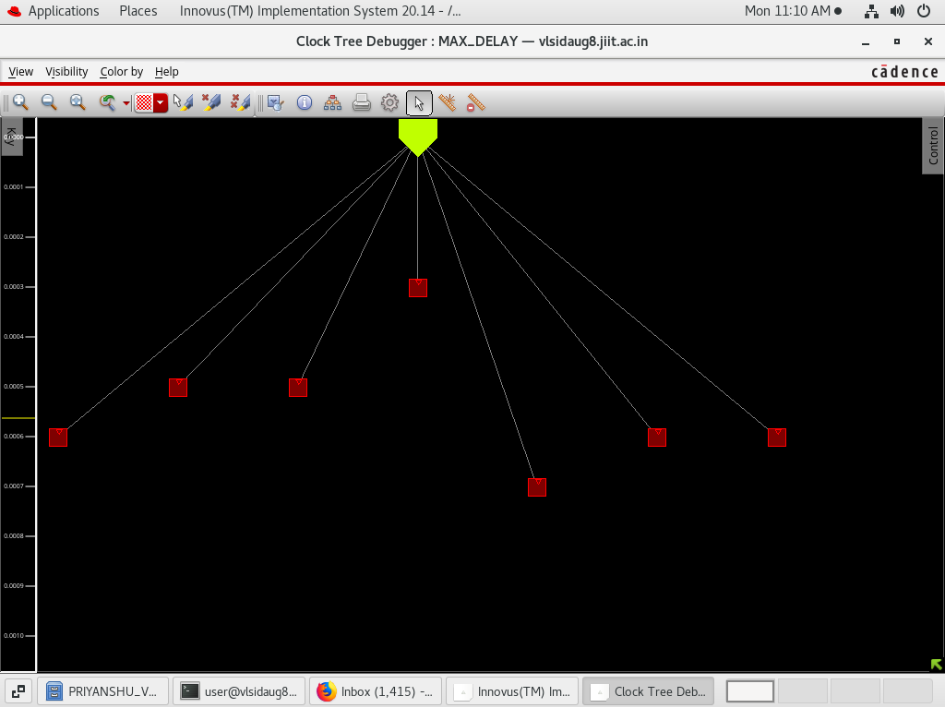
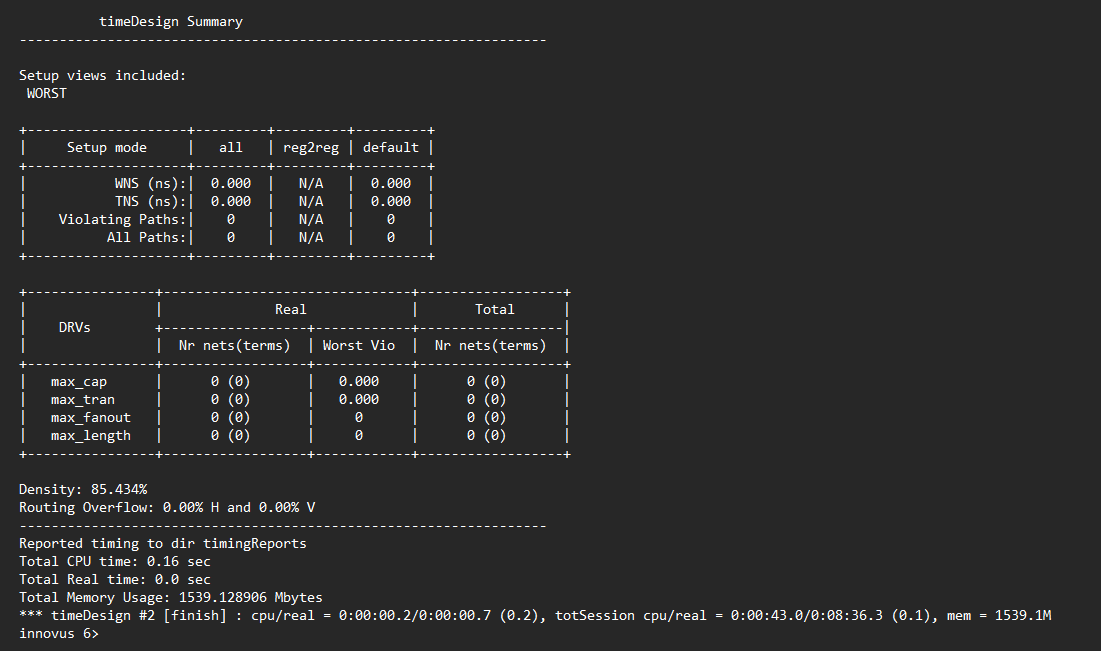
****

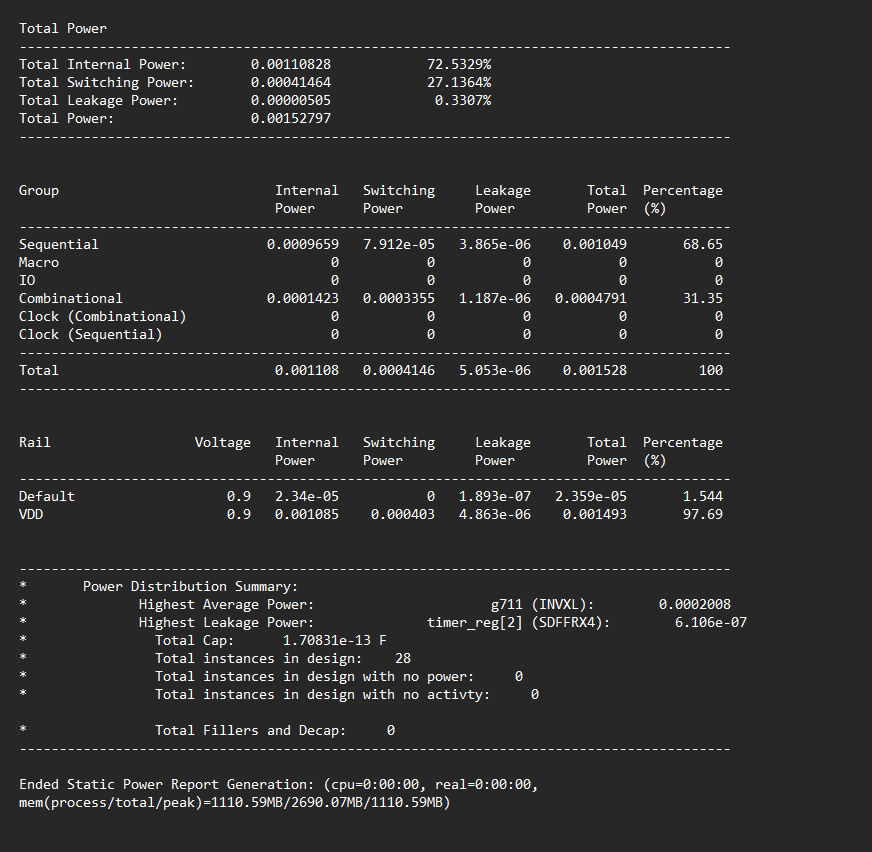
****

****

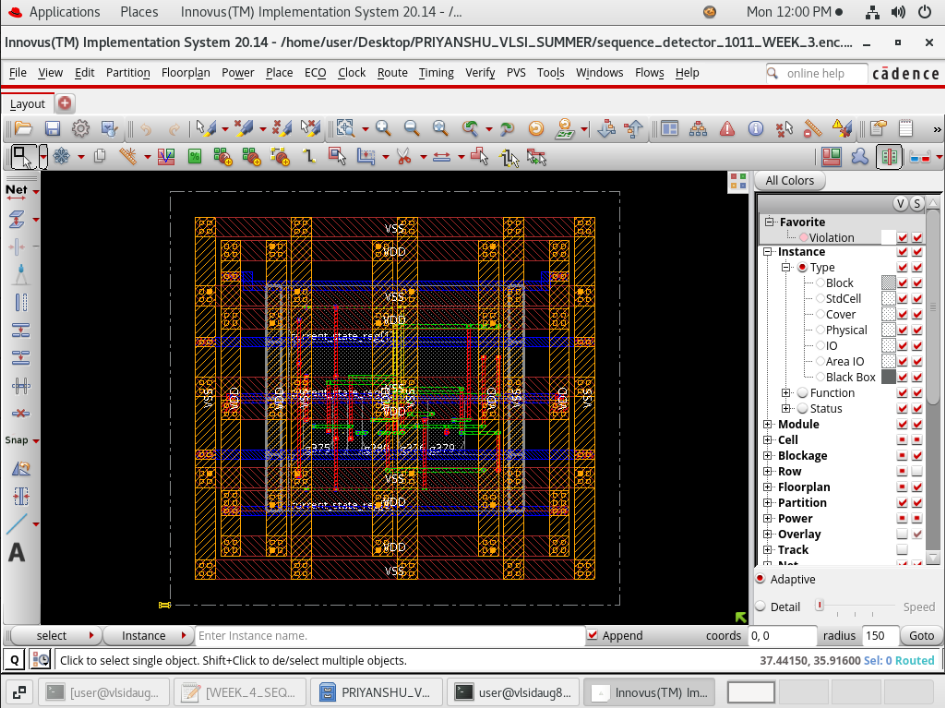
**TRAFFIC LIGHT CONTROLLER**

****

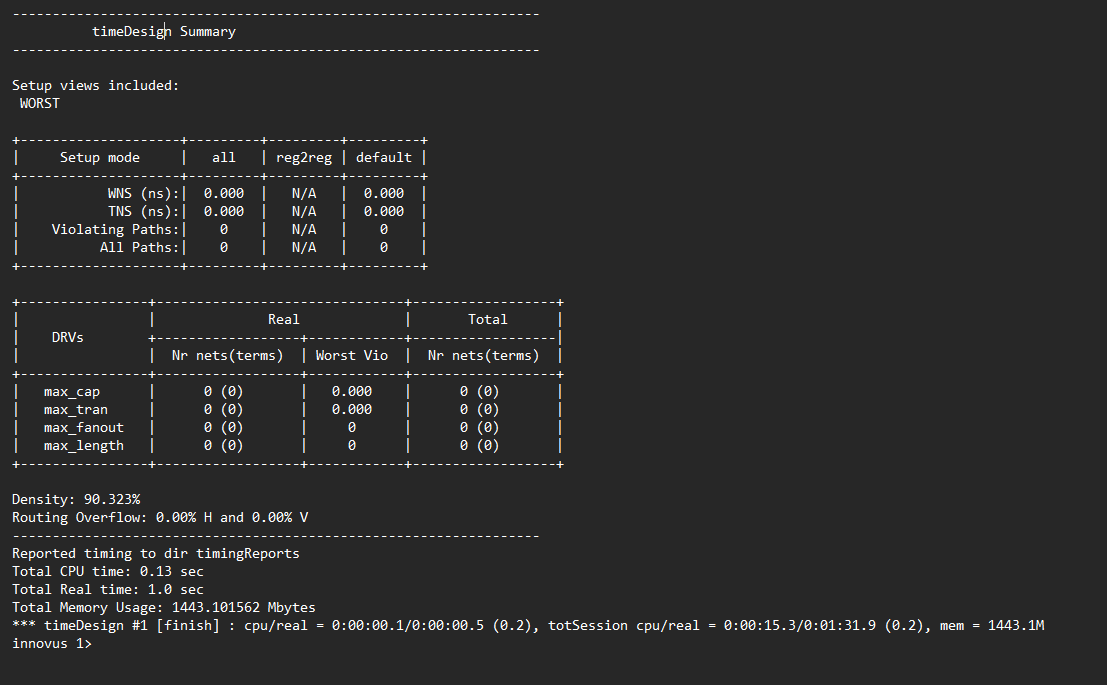
**** ****

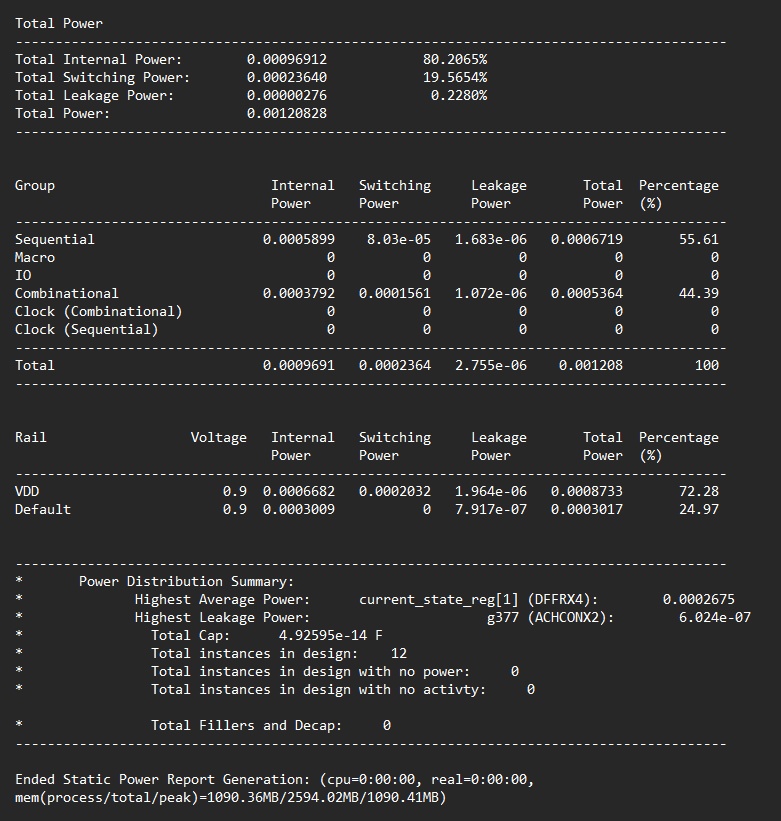
****

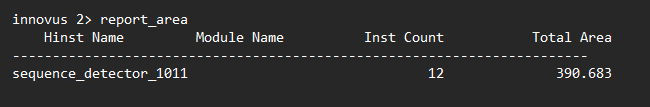
**MEALY SEQUENCE DETECTOR**

****

****

****

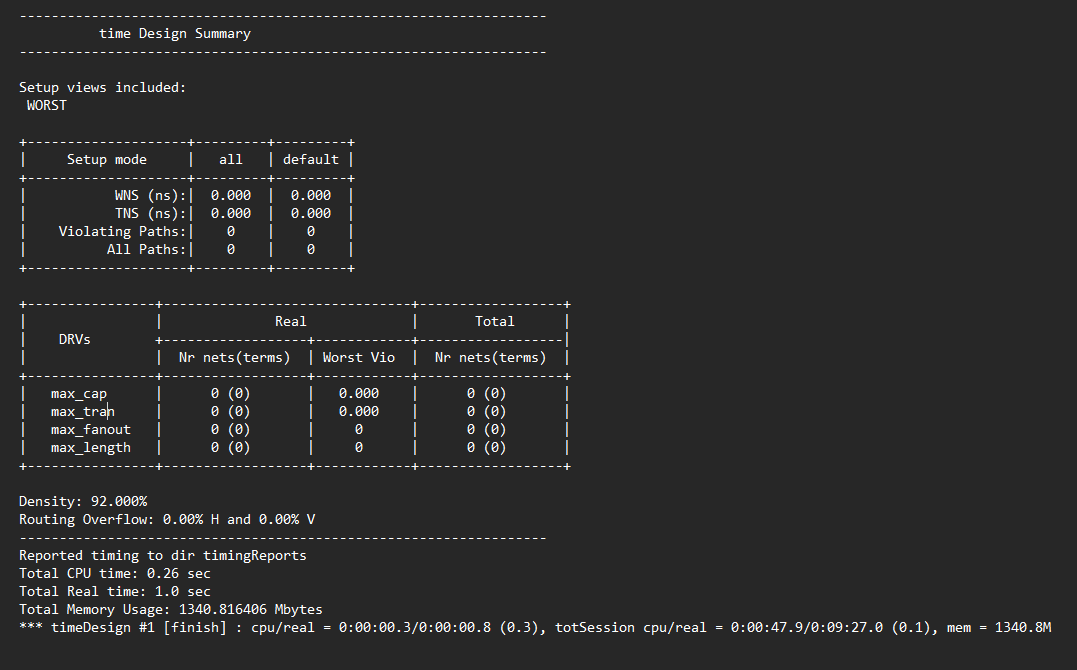
****

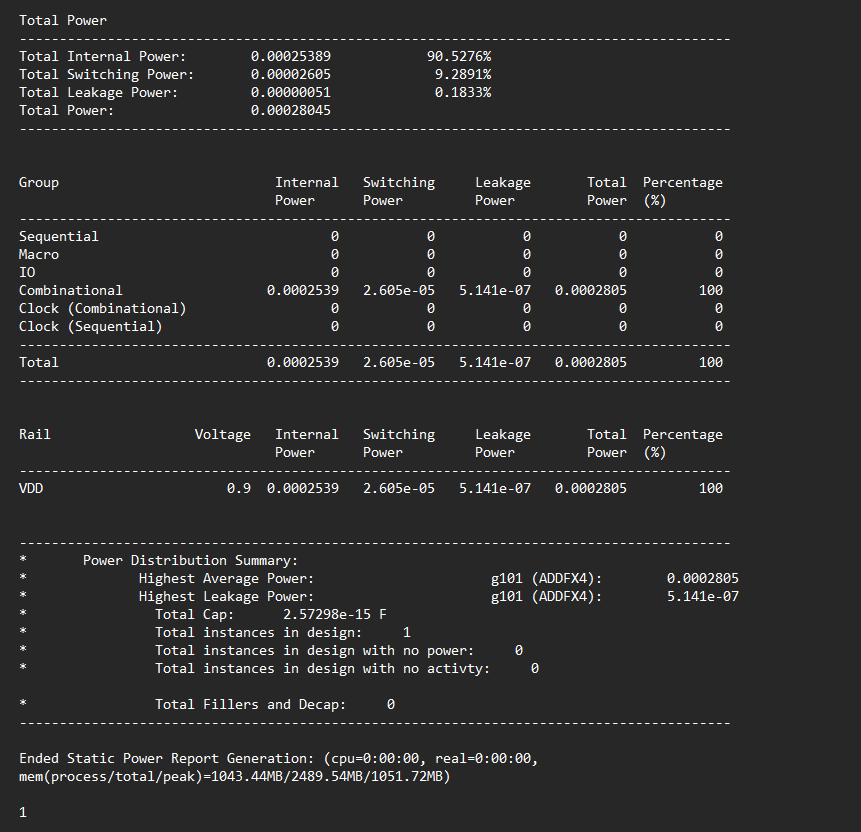
****

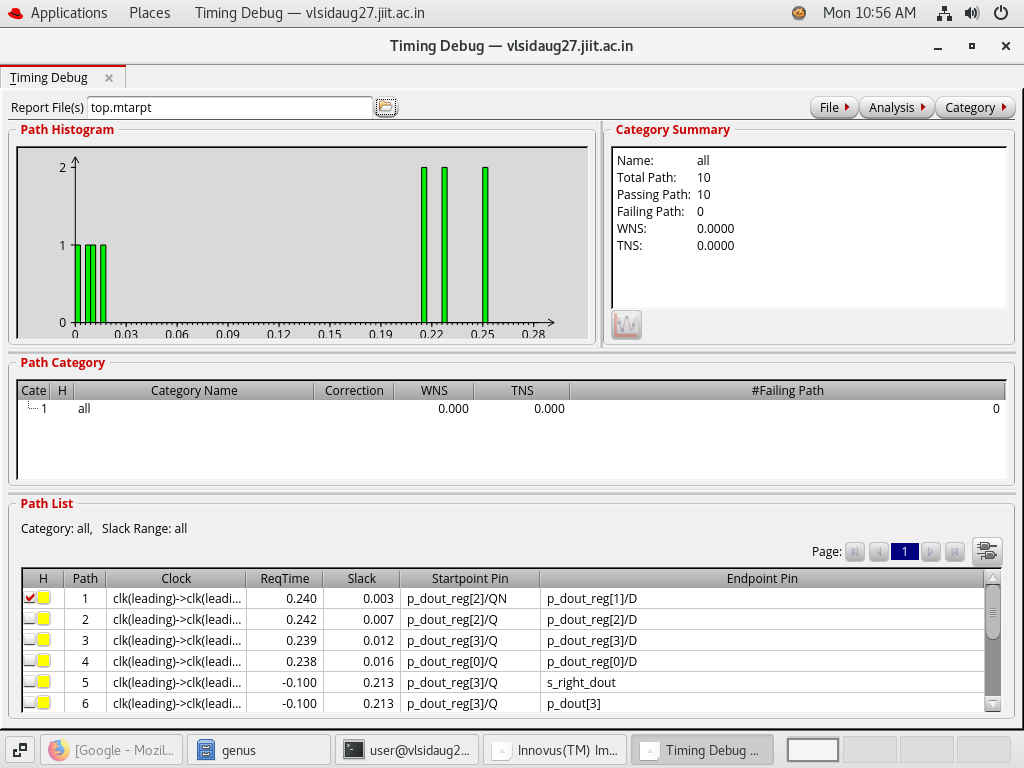
**WEEK 5**

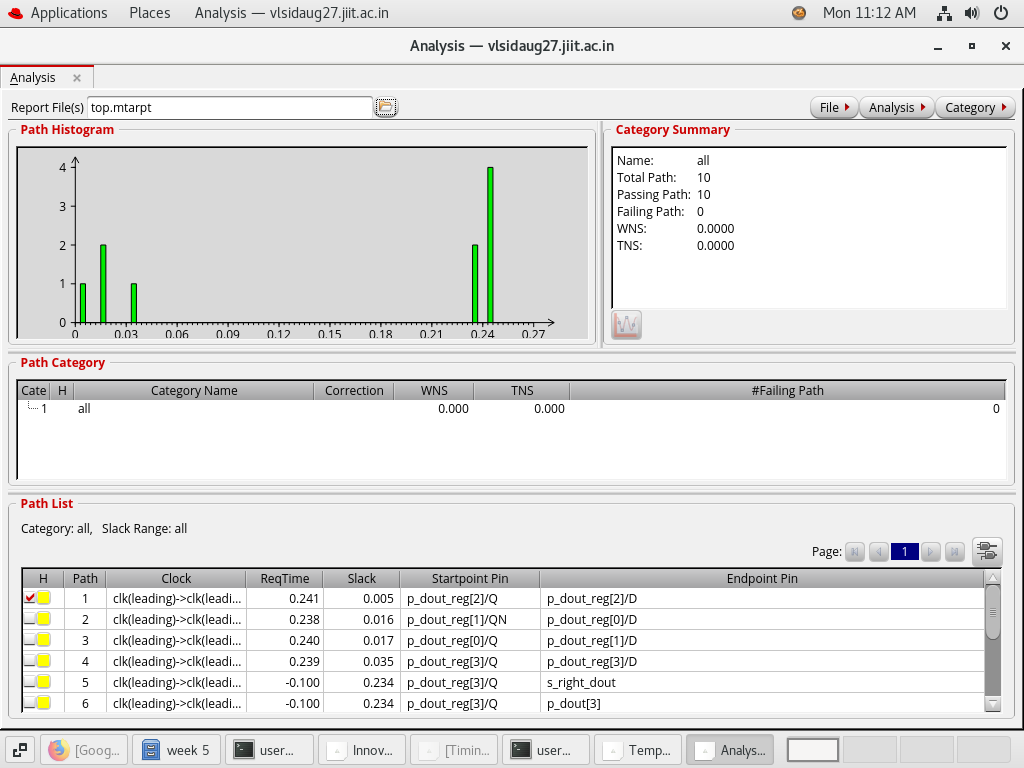
**STATIC TIMING ANALYSIS**

**FULL ADDER**

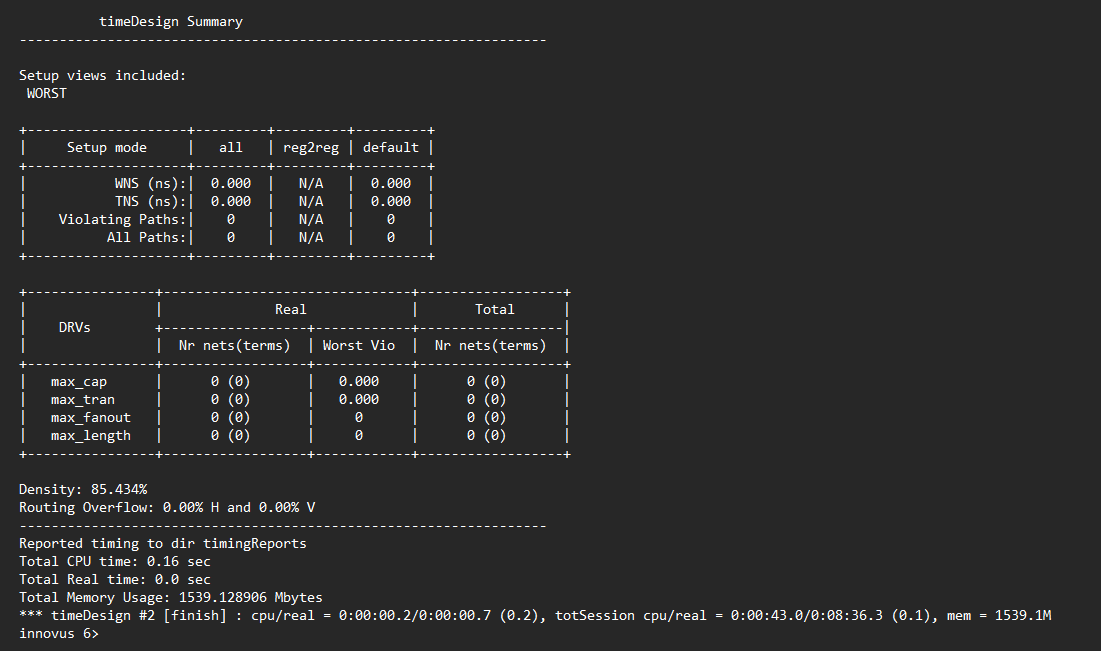
****

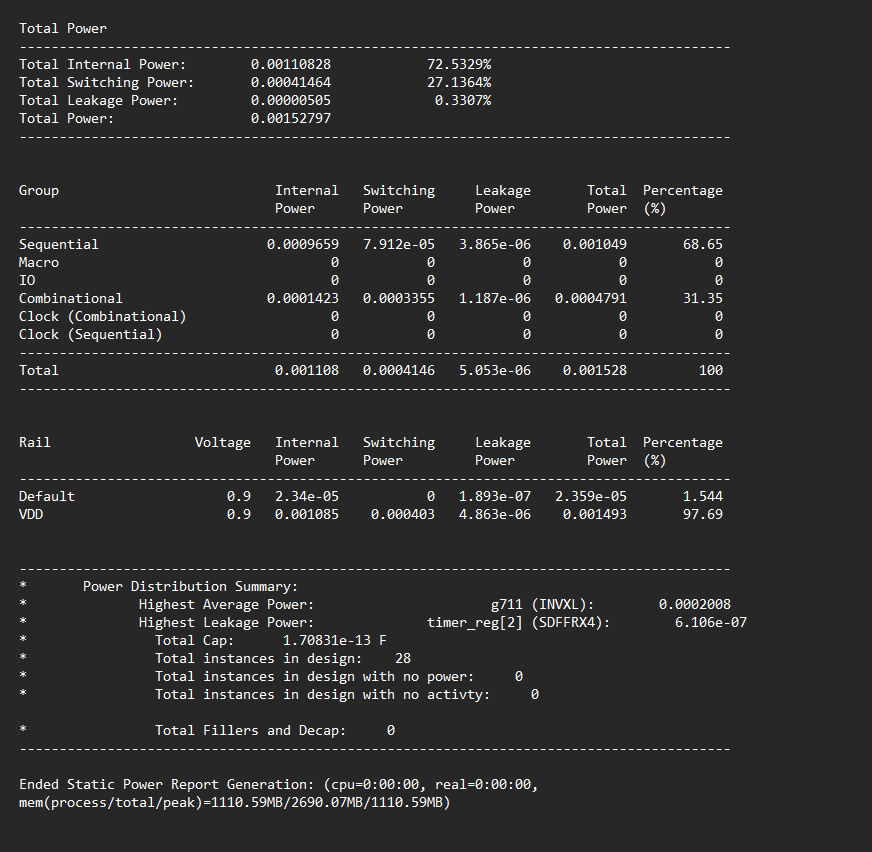
****

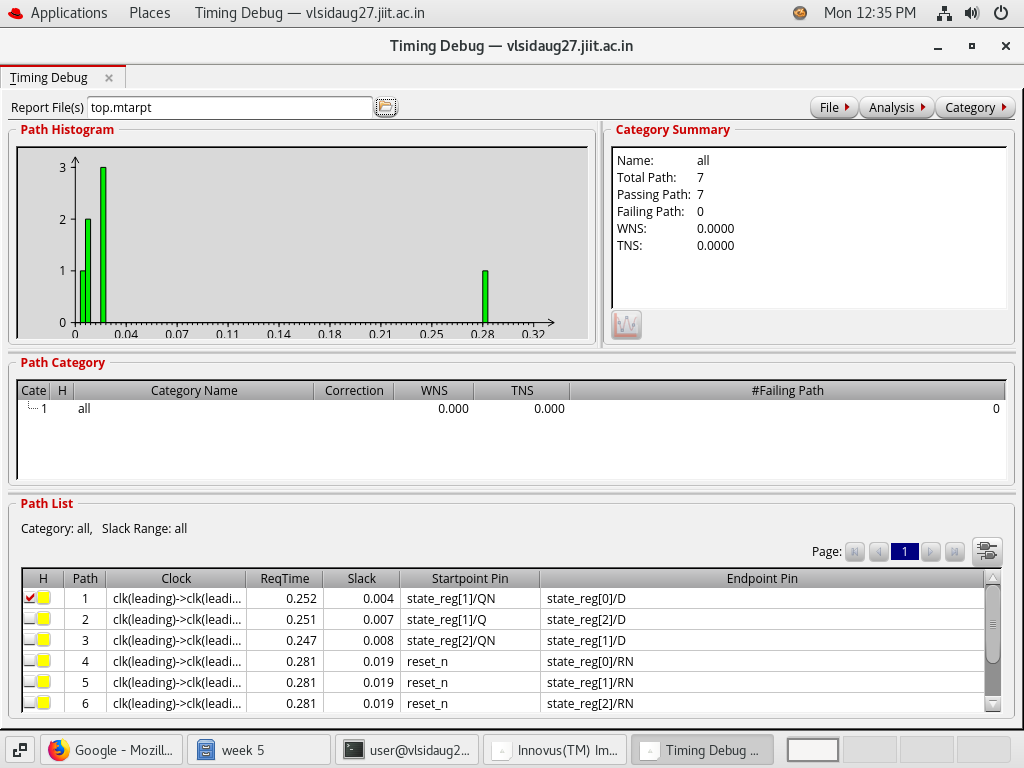


****

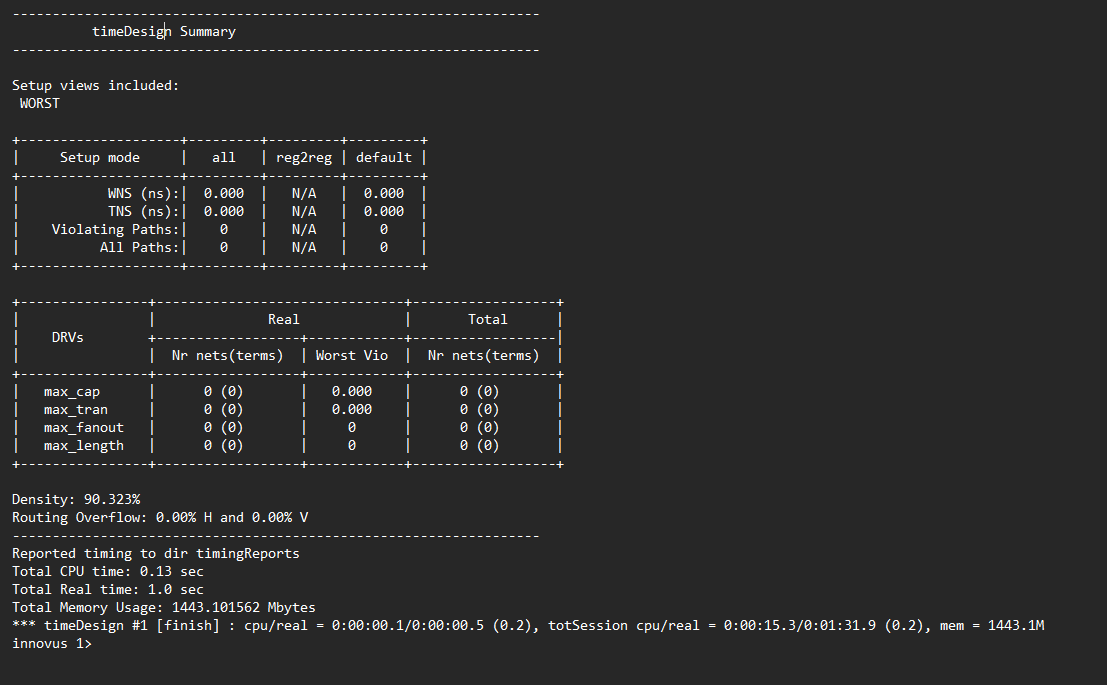
**TRAFFIC LIGHT CONTROLLER**

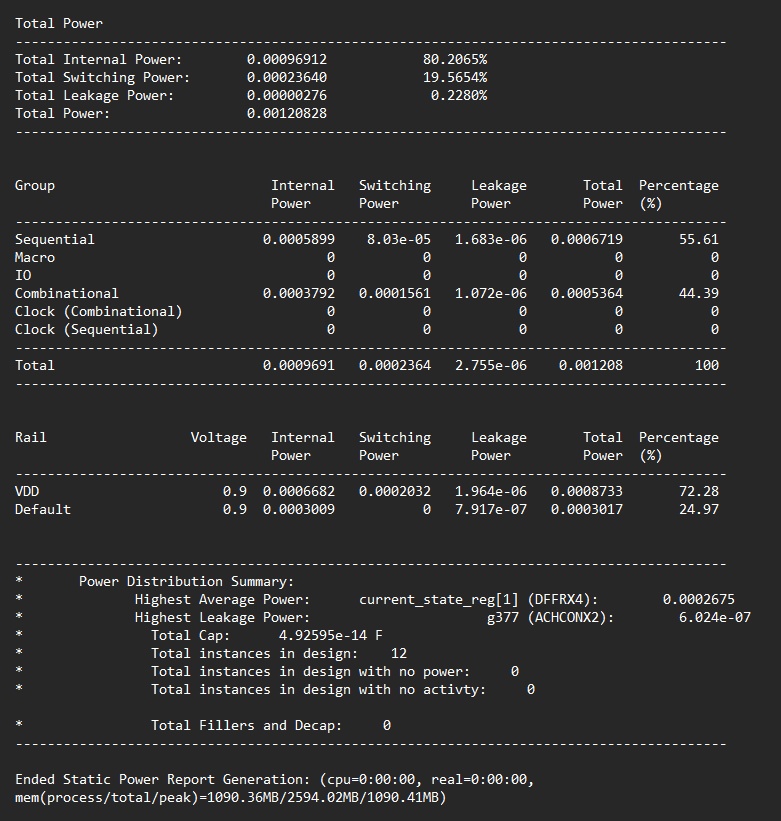
****

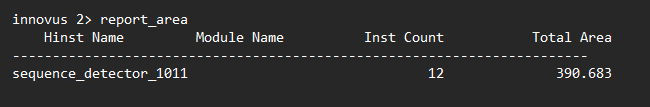
****

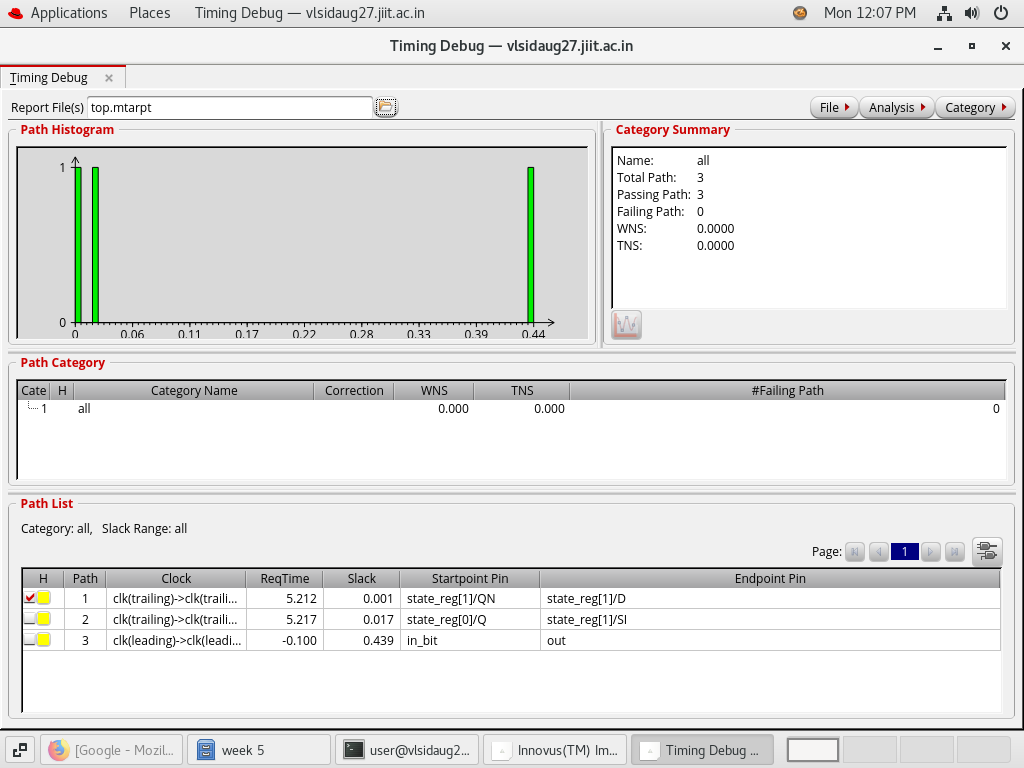


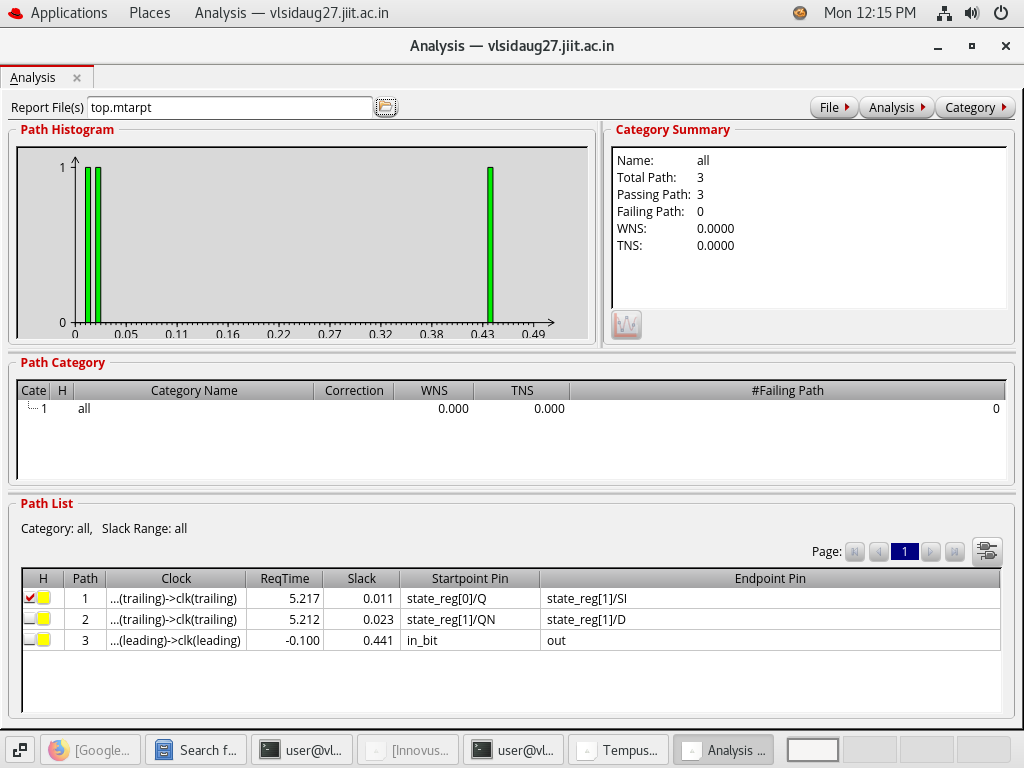
**MEALY SEQUENCE DETECTOR**

****

****

****

****

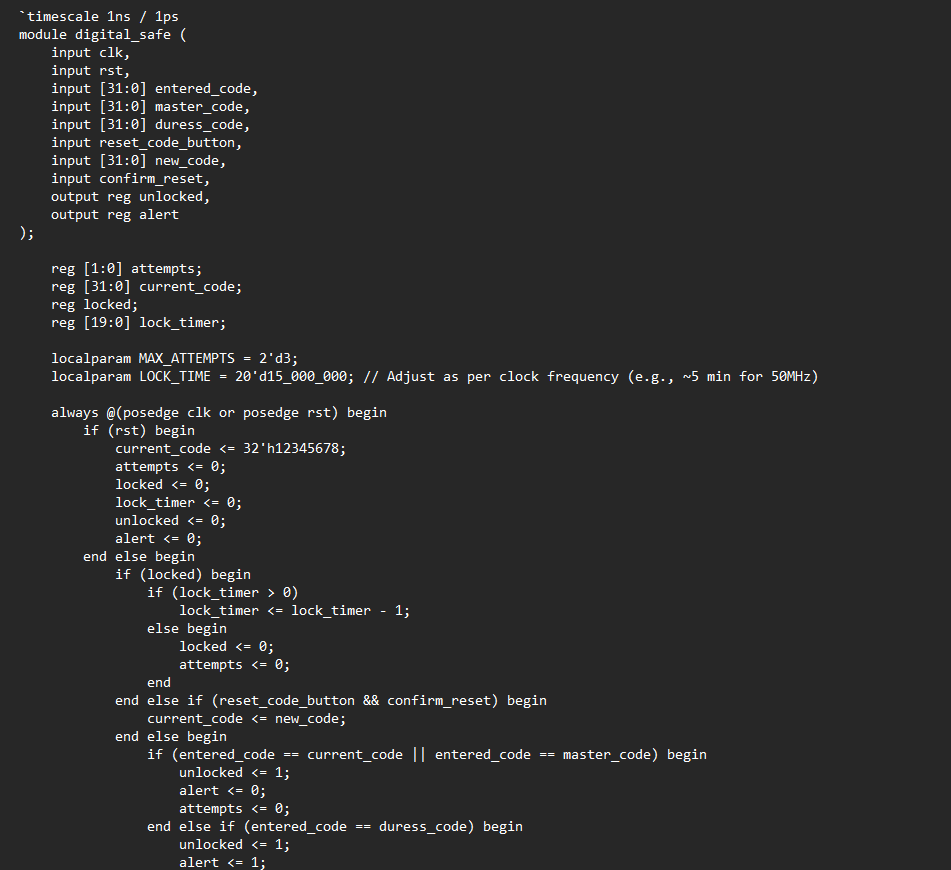


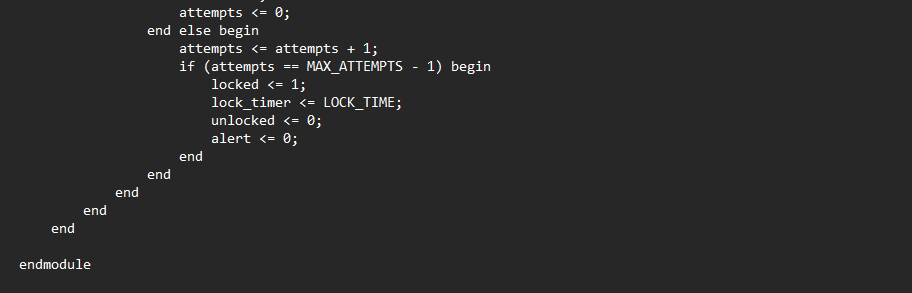
**WEEK 6**

**PROJECT**

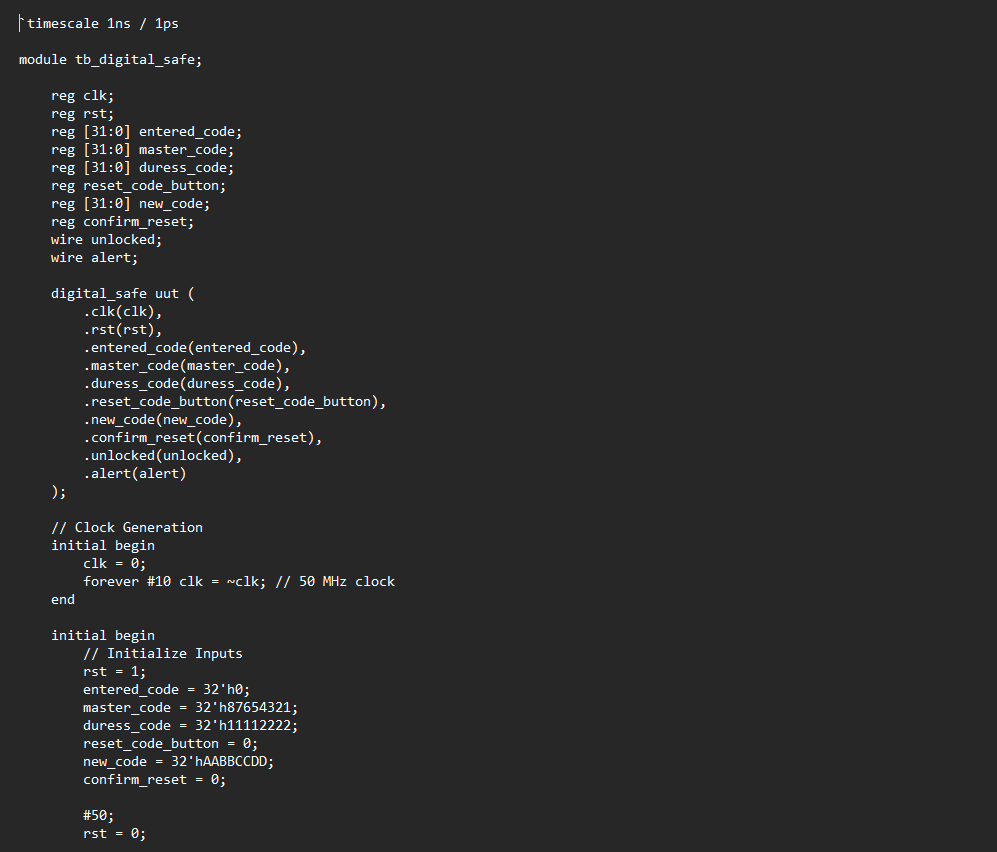
**Problem Statement :** Digital Safe Lock System Design Problem  
Design a digital controller for an electronic safe lock system used to secure valuables in homes, banks, or offices. The safe lock is operated via a digital keypad where the user enters a numeric passcode, typically 4 to 8 digits long. The safe’s controller stores a master code in non-volatile memory and compares every entered code to this stored code.  
When the user enters the correct passcode, the controller energises a solenoid to unlock the safe door for a fixed duration, such as 10 seconds, after which it automatically relocks. The controller must keep track of the door’s status using a door sensor to ensure it closes properly before relocking.  
If a user enters an incorrect passcode, the system increments a failed-attempt counter. After three consecutive wrong entries, the safe enters a lockout period lasting 5 minutes, during which the keypad is disabled and the user cannot make further attempts. During lockout, the system should display a “LOCKED OUT” message or light an LED indicator.  
The safe must also support a duress mode. If the user enters a special duress code known only to authorised users, the safe unlocks normally, but the system quietly triggers a hidden alarm output to notify authorities of a possible coercion situation.  
Furthermore, the safe should have a setup mode accessible only to the master user. This mode allows the master user to change the safe’s passcode, enable or disable duress mode, and test the solenoid lock mechanism. The controller should store the new passcode securely and prevent unauthorised access to setup mode.  
Inputs include keypad digit inputs, an enter key, door sensor signals, master setup mode activation, duress code entry, clock, and reset. Outputs include signals to control the solenoid lock, display or LED outputs for status messages (e.g., “ACCESS GRANTED,” “WRONG CODE,” “LOCKED OUT”), and a hidden alarm signal for duress conditions.  
Design Constraints  
The digital safe lock system must be implemented as a Mealy finite-state machine in a fully synthesisable RTL style suitable for ASIC synthesis and physical design. All logic must avoid combinational loops and should separate FSM control from the datapath handling code comparisons, counters, memory storage for codes, and timing management for lockout periods.

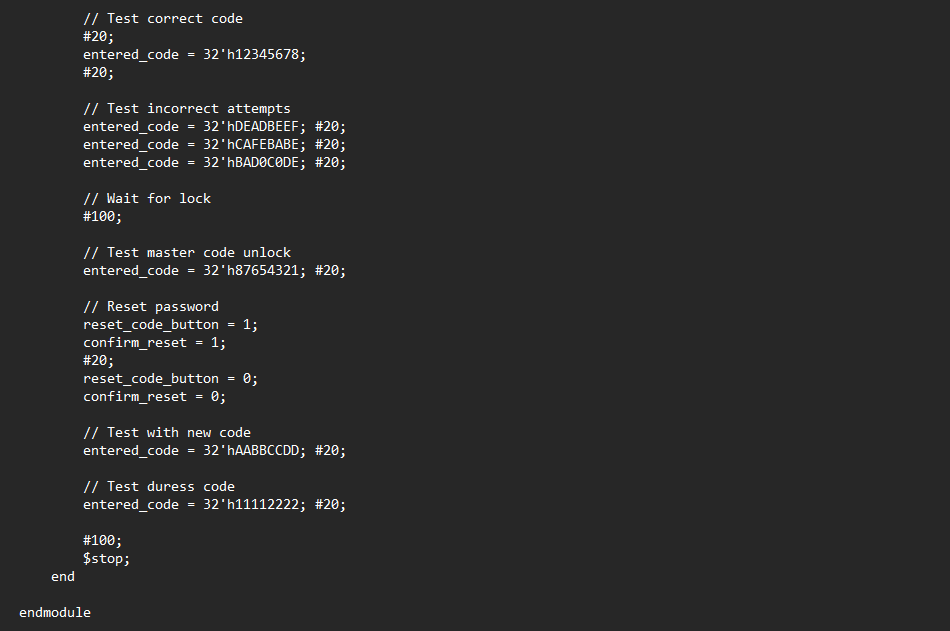
**Project**

****

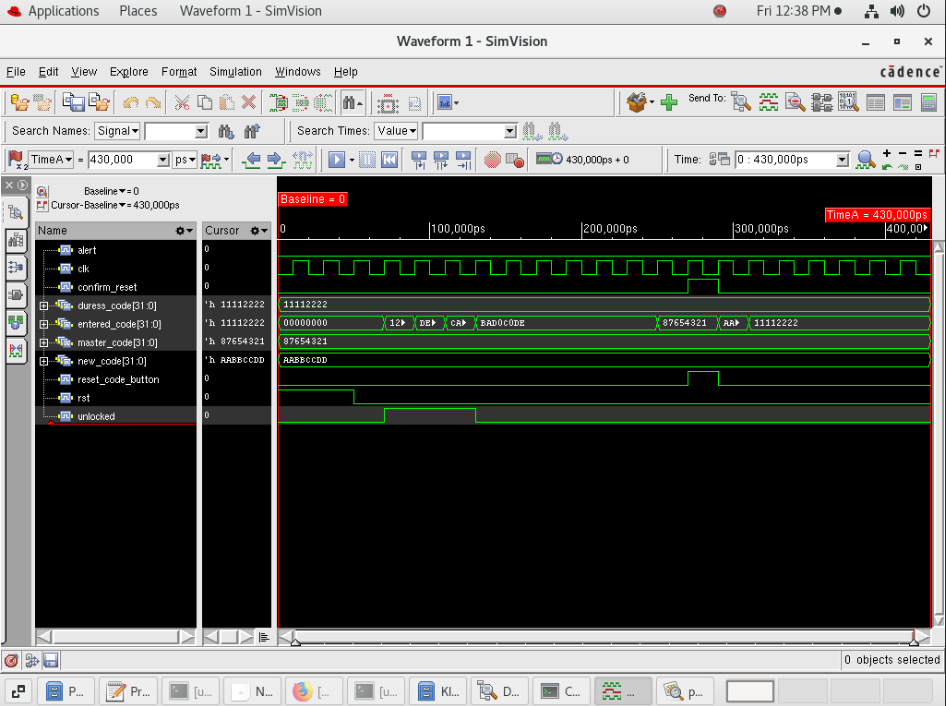
****

**Test Bench**

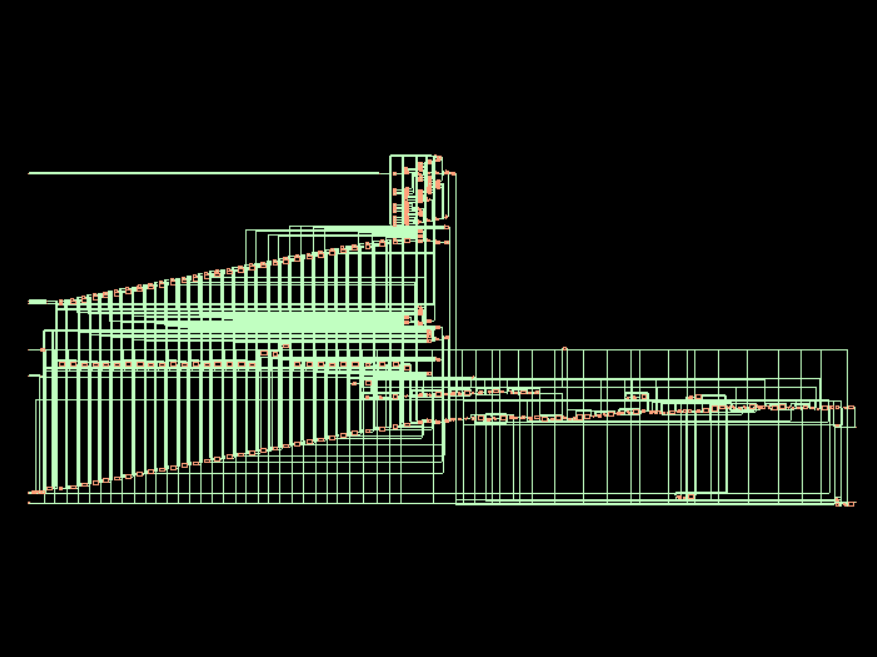
****

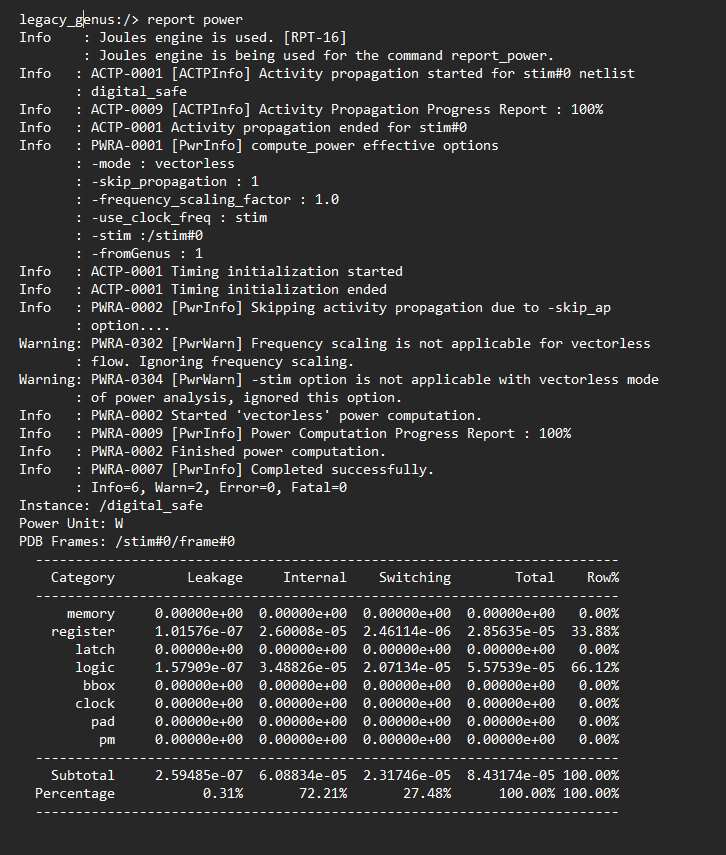
****

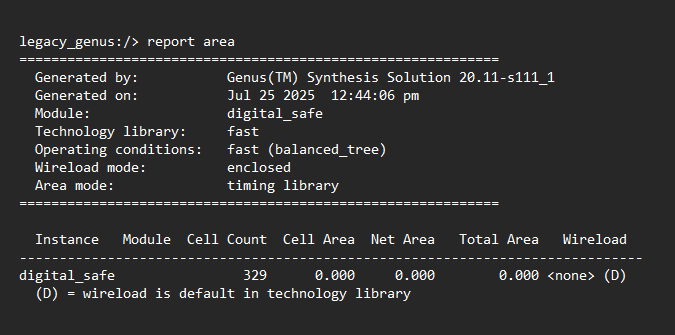
**Wave Form**

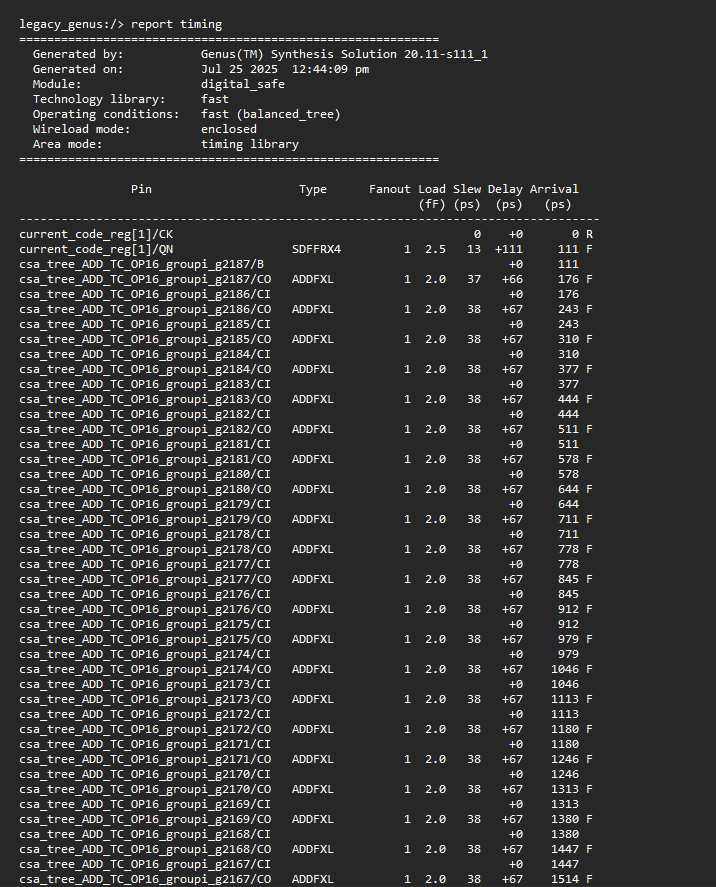
****

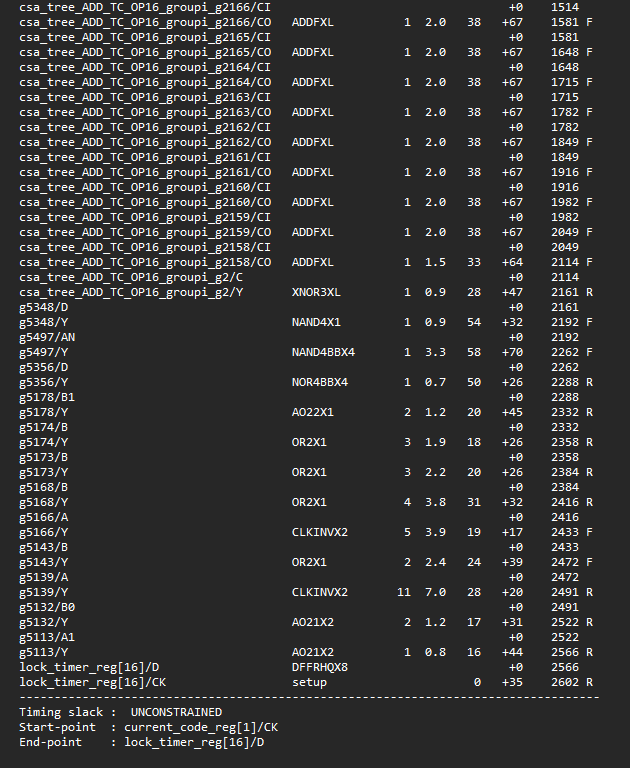
**Synthesis**

****

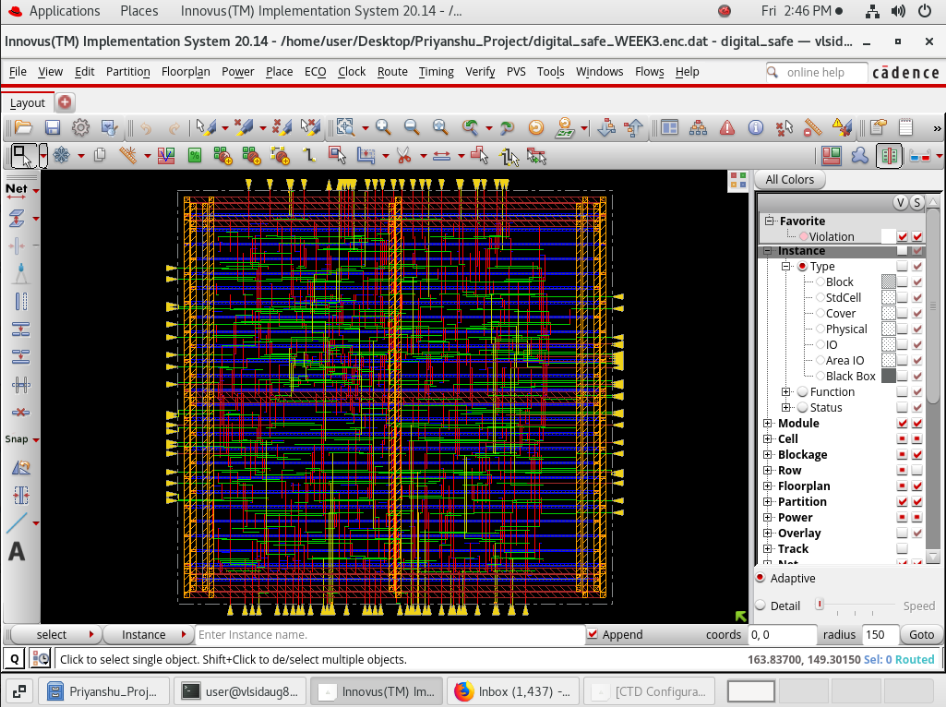
****

****

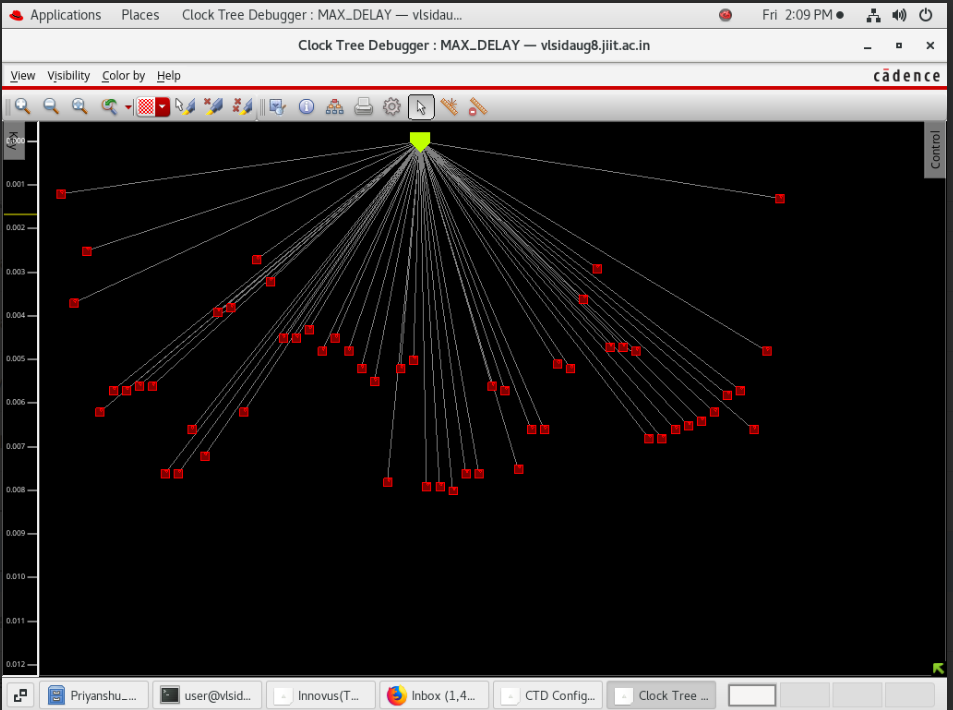
****

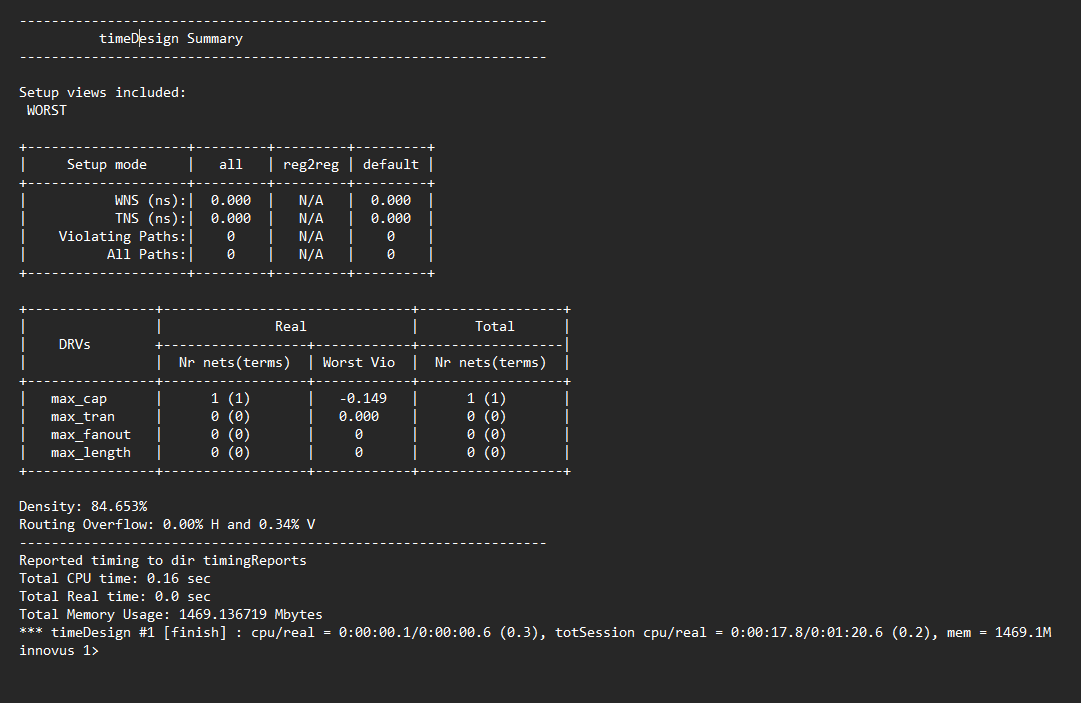
****

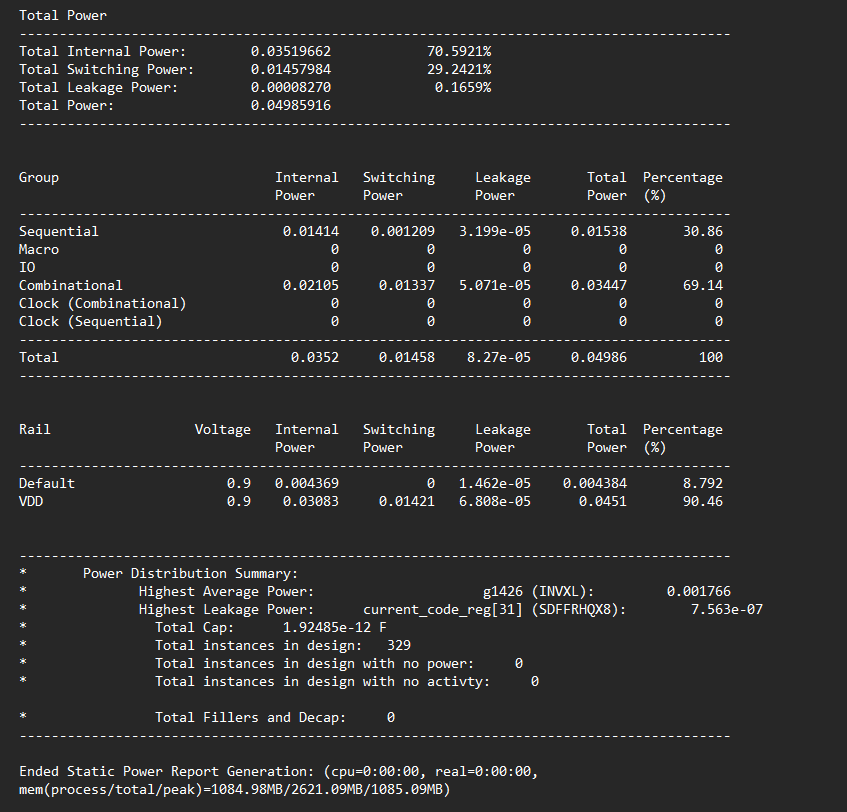
**Floor Planning, Power Planning , Placement , CTS , Routing and RC Extraction**

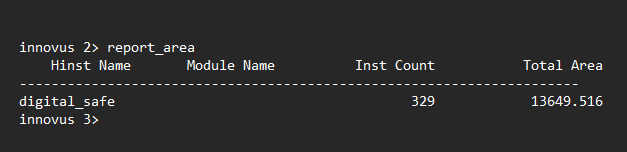
****

****

****

****

****

****