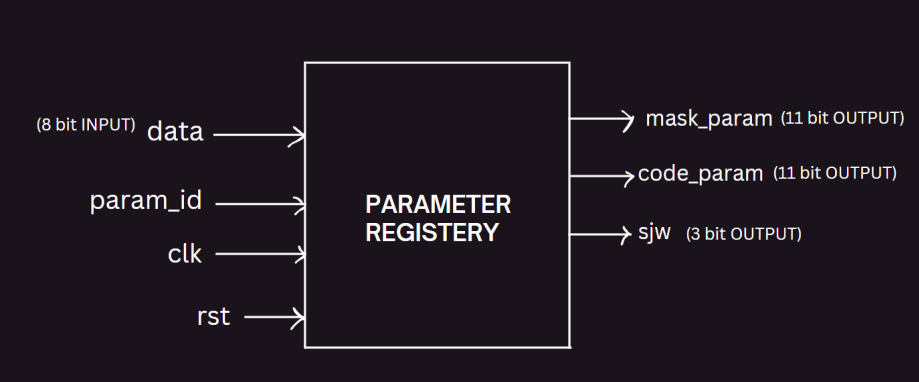


**The Problem Statement**

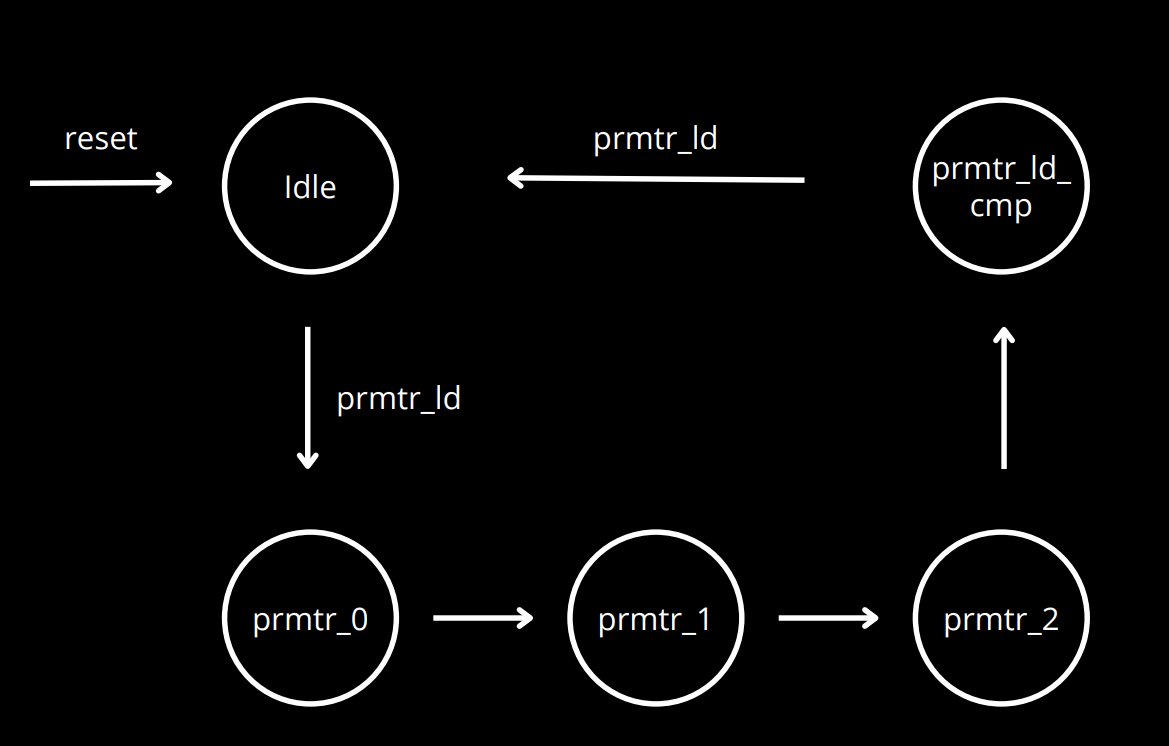
A registry block implements a finite state machine (FSM) model for

managing parameter loading and manipulation based on input data. The inputs to this block include a clock signal, a global reset, a data input (data\_in) and a parameter load signal (prmtr\_ld). The outputs from the block are a masked parameter (mask\_param), coded parameter (code\_param) and a synchronisation jump width (sjw). A finite state machine is used to implement this logic. The task is to take the 8-bit data input (data\_in) and convert it into 11-bit encoded outputs for mask\_param and code\_param.

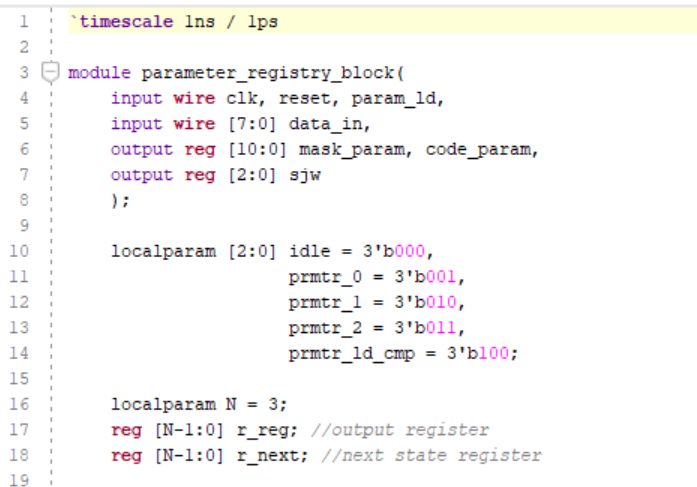
**Block Diagram**

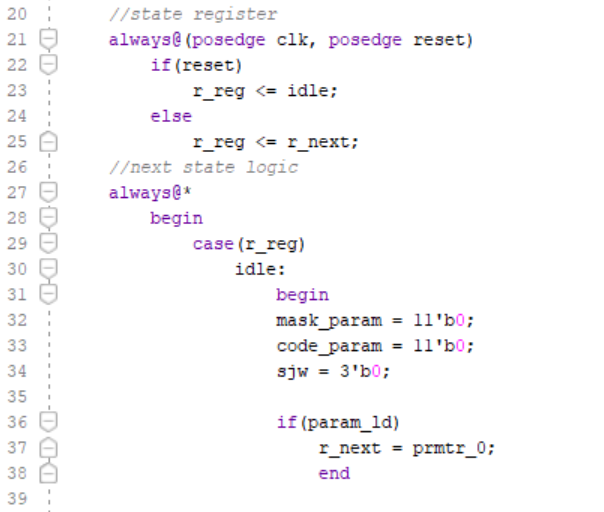


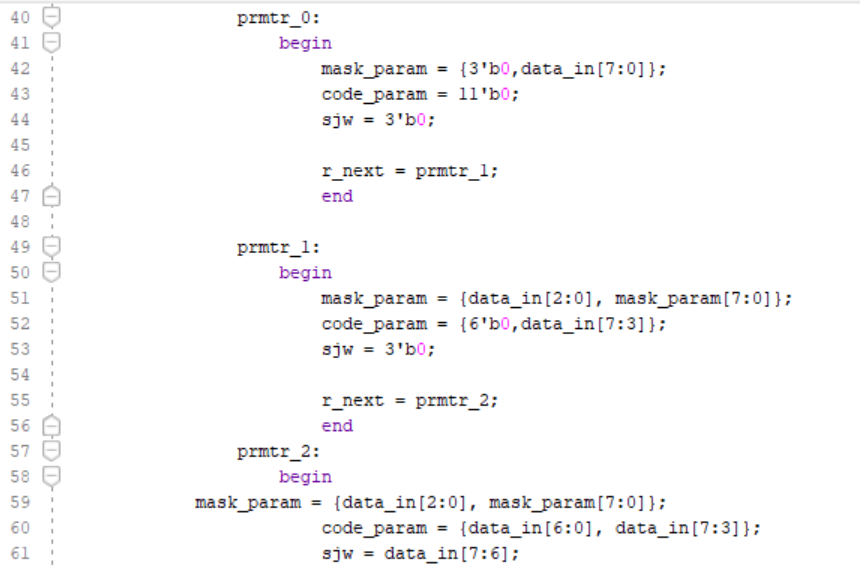
**State Diagram**

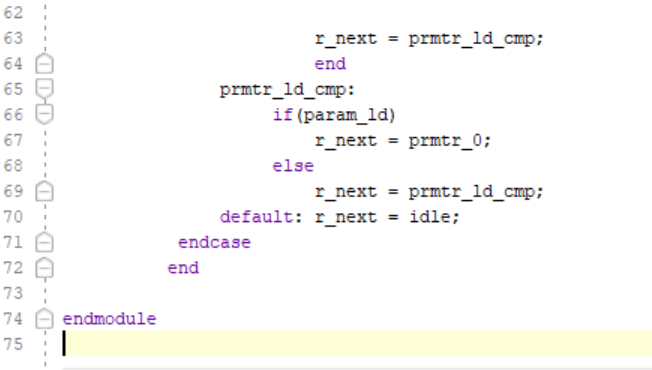
****

**Design Code**

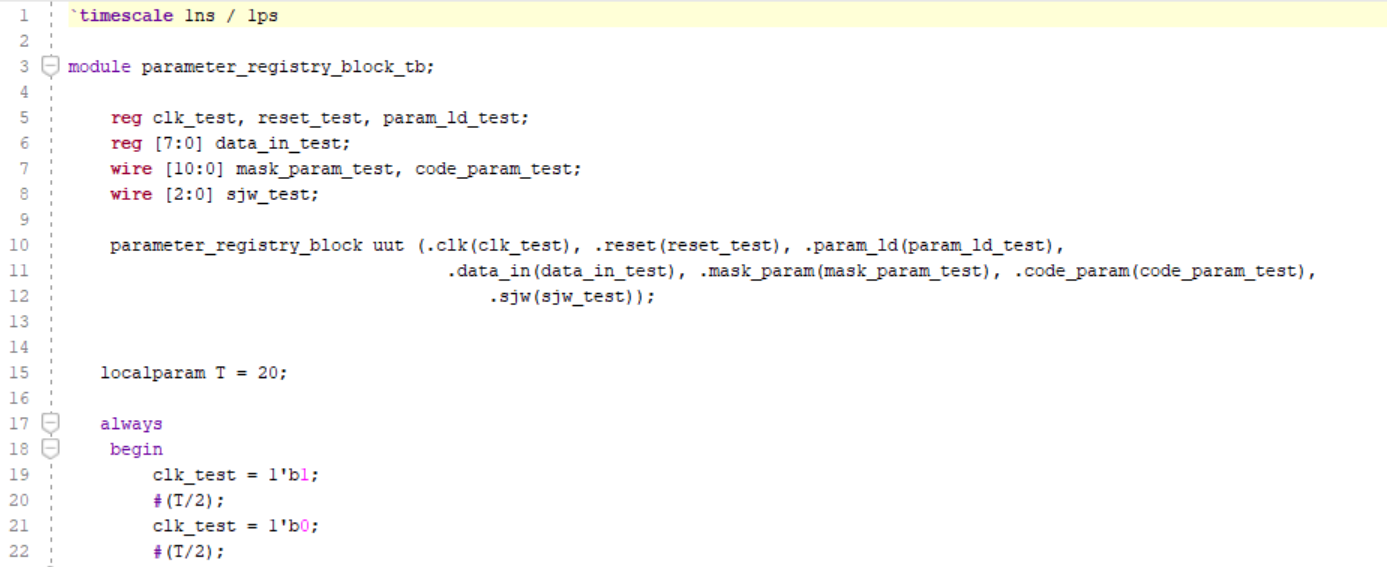
****

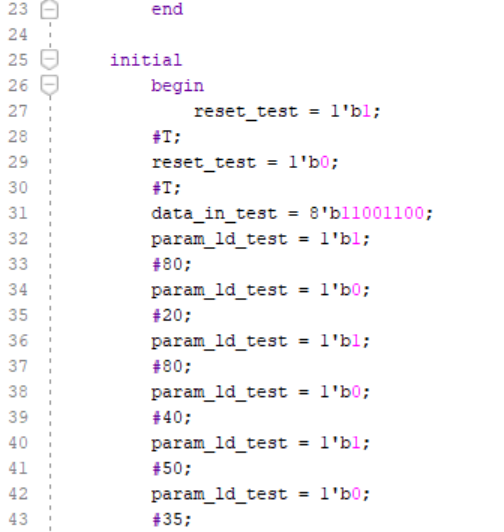
****

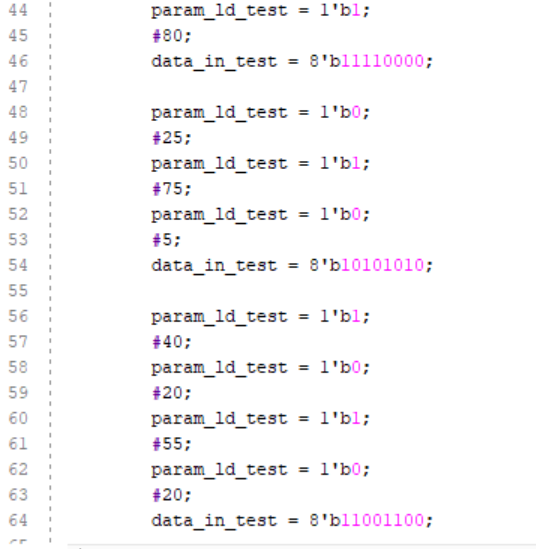
****

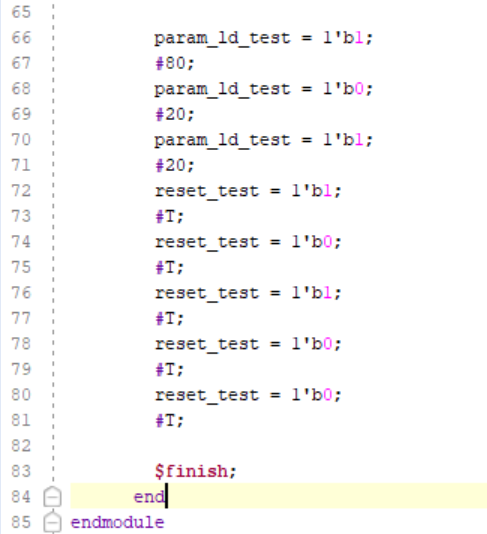
****

**Testbench Code**

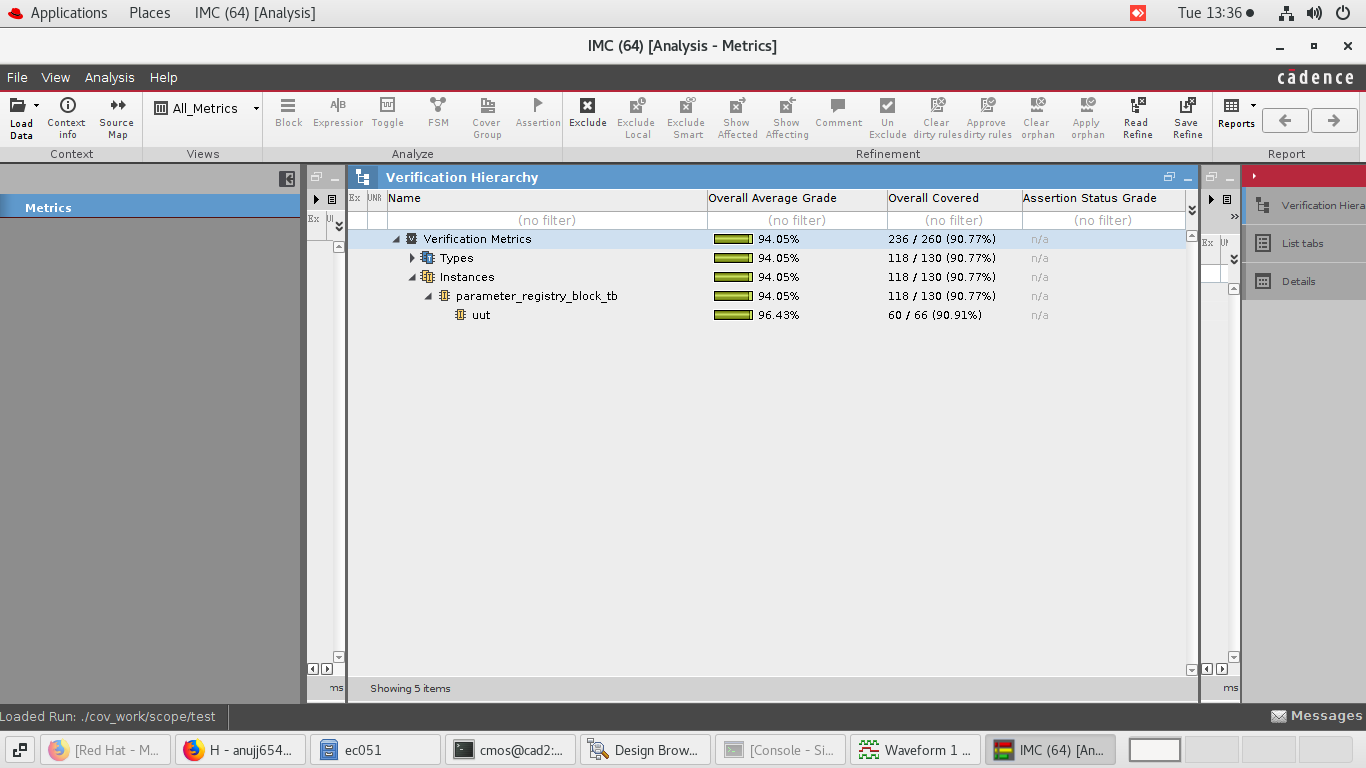


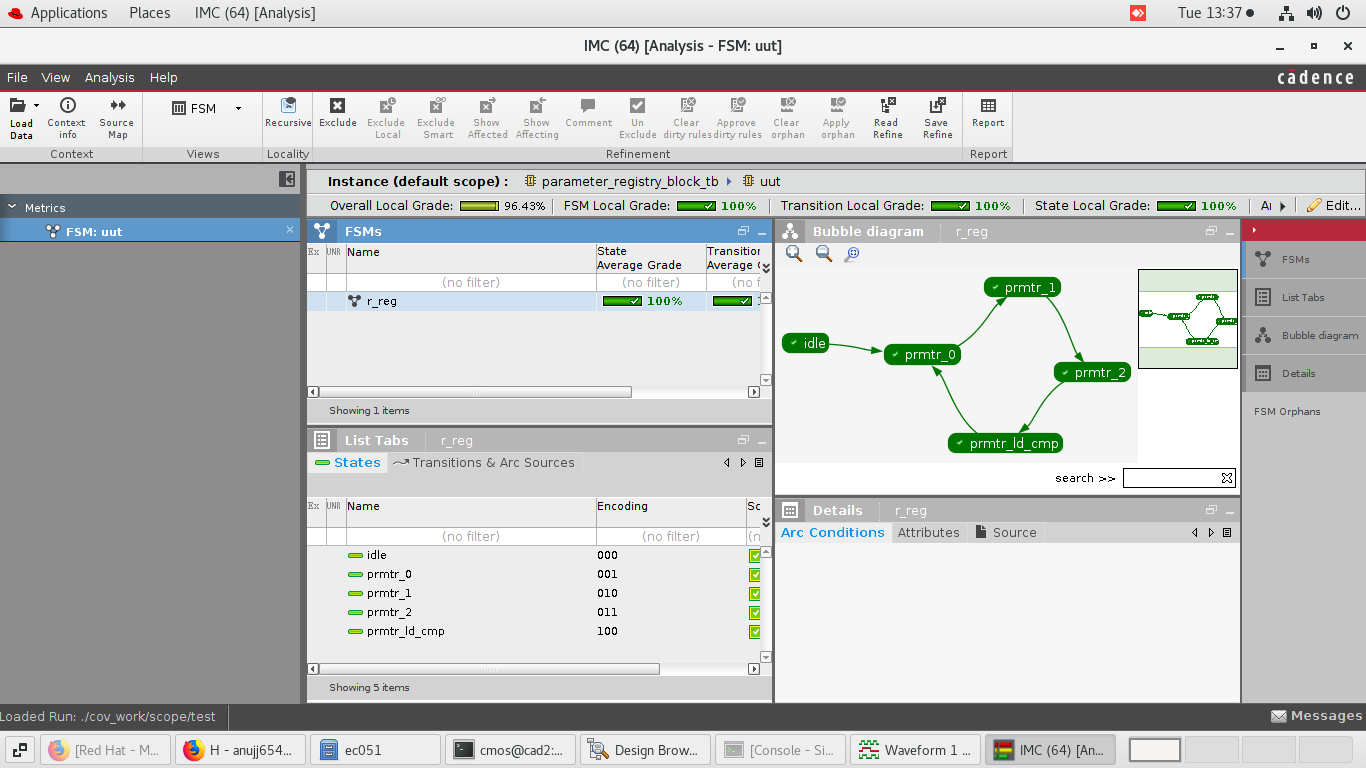




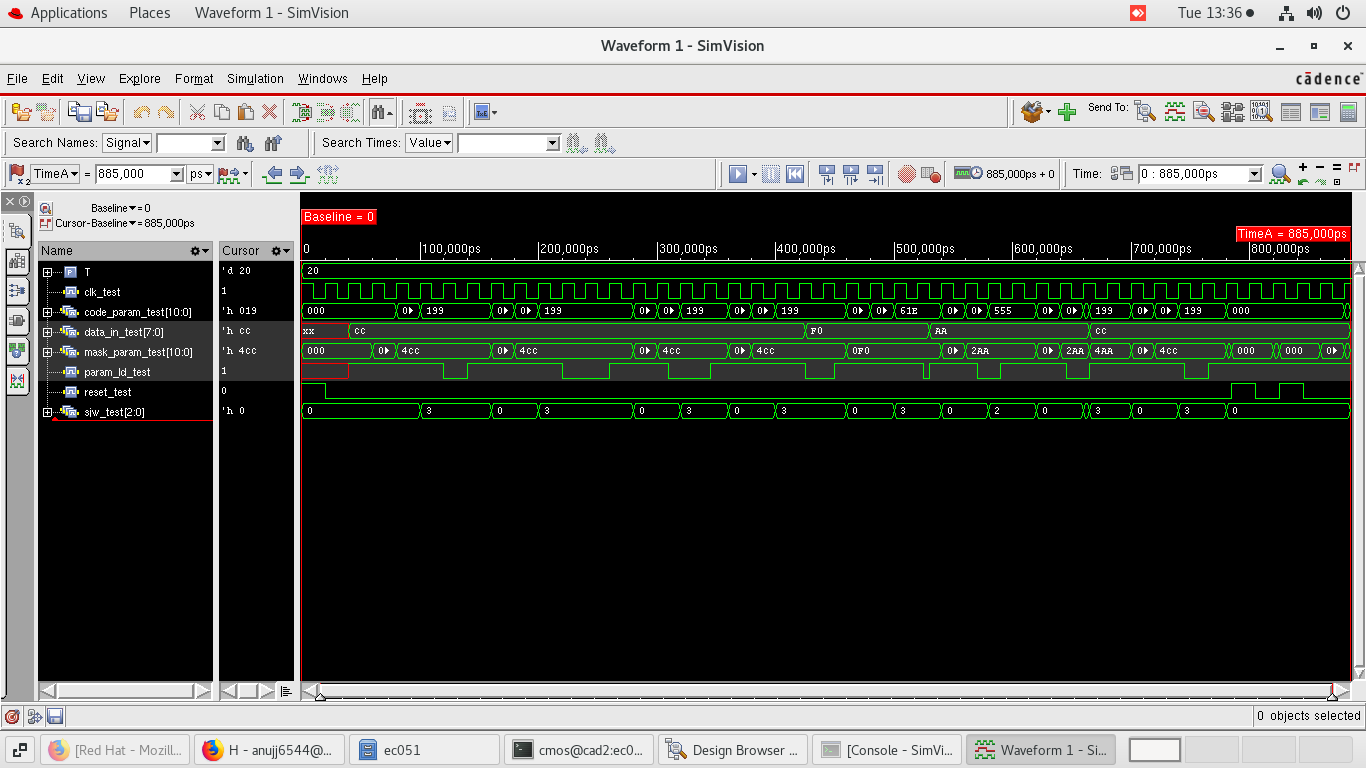


**Code Coverage**

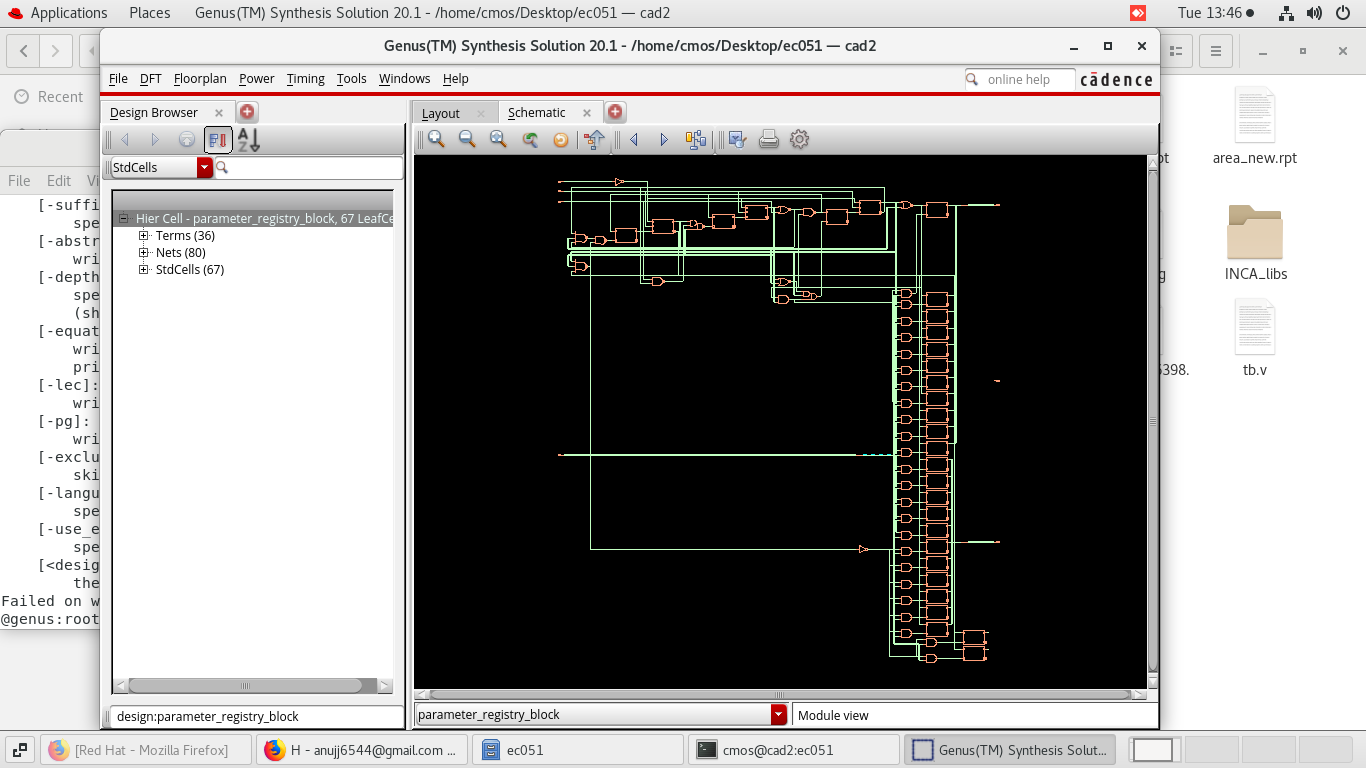
****

****

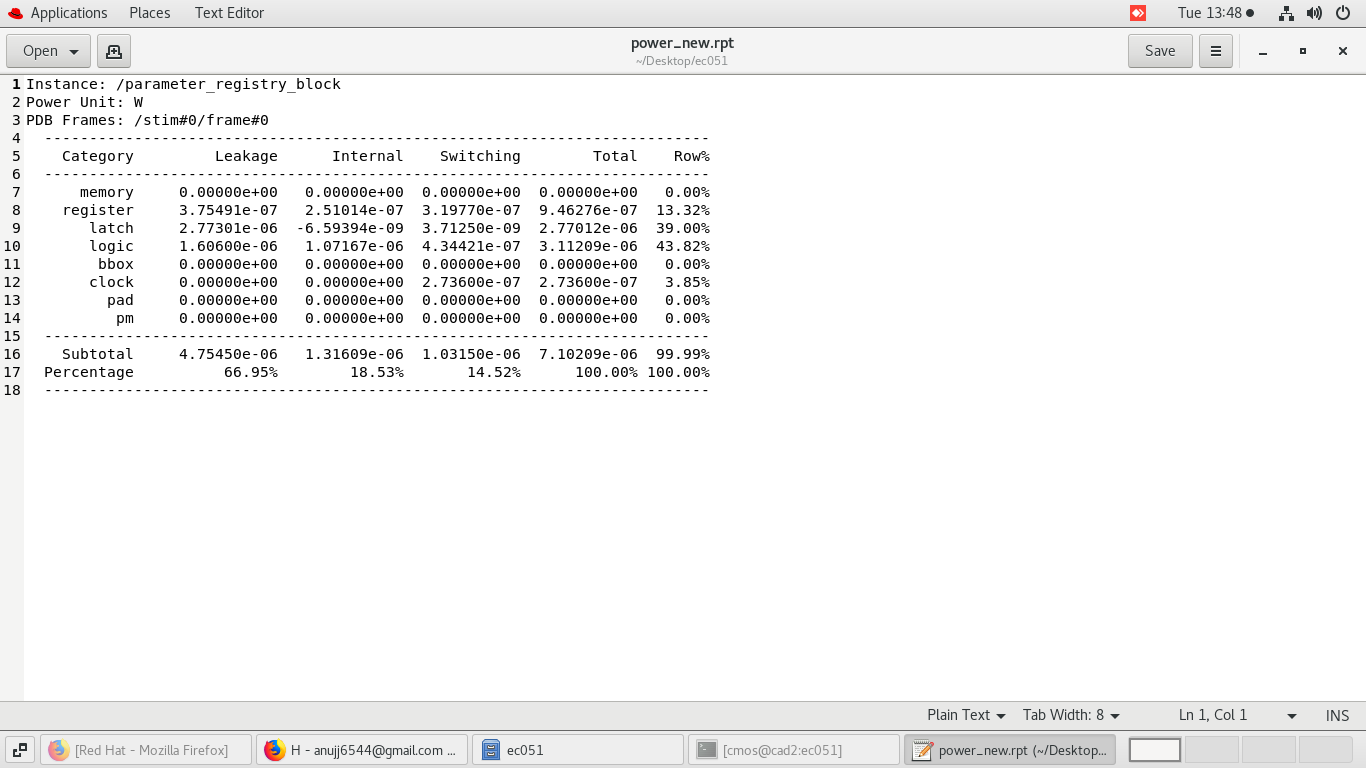
**Simulation Results**

****

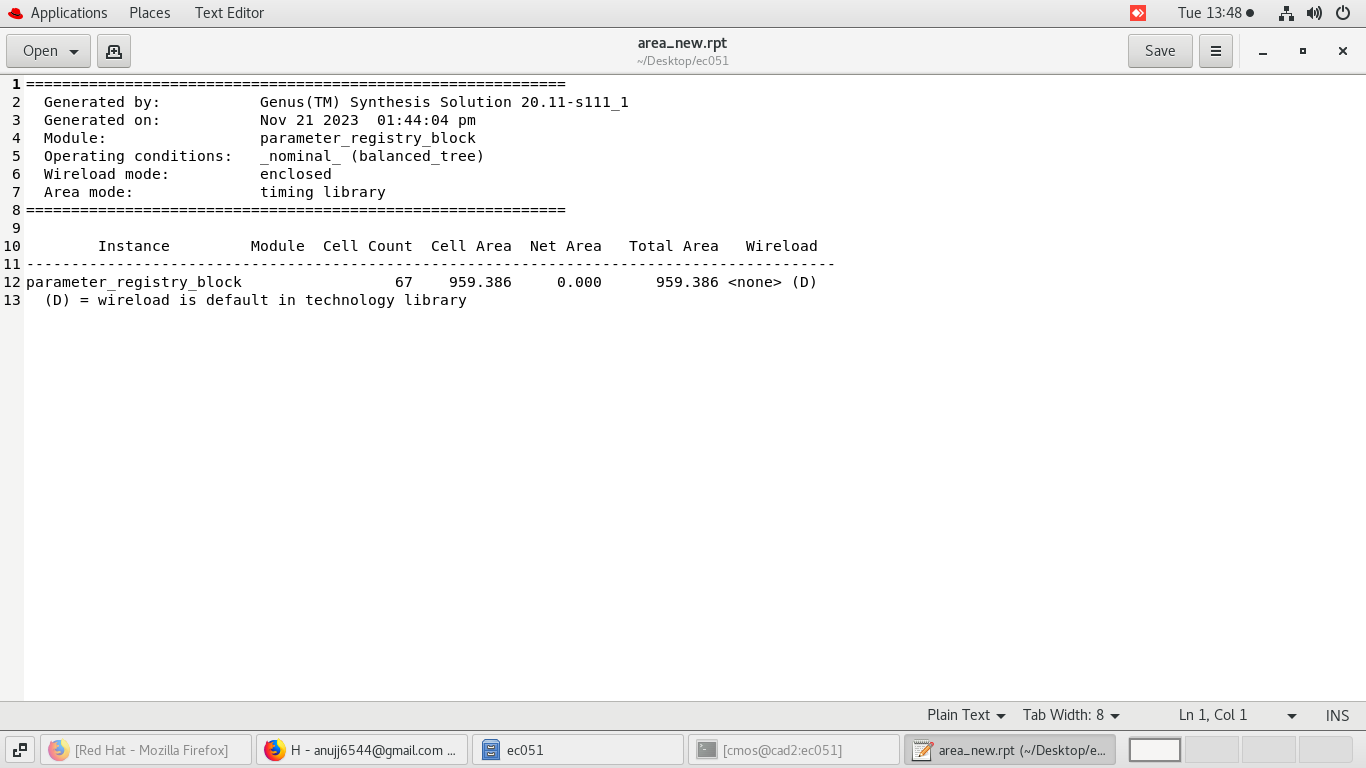
**Synthesis Schematic**

****

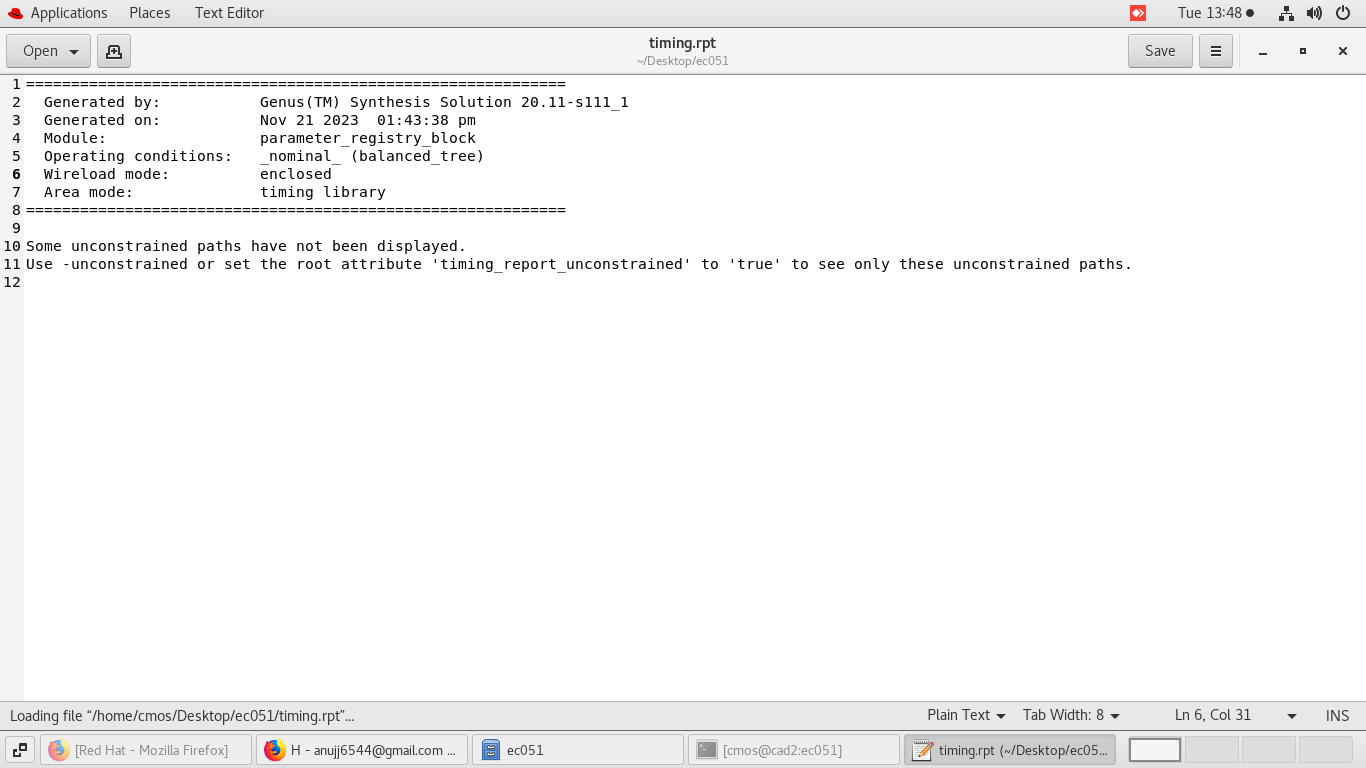
**Power Report**

****

**Area Report**

****

**Timing Report**

****