INTEL UNNATI INDUSTRIAL TRAINING - SUMMER 2023

Problem Statement: Design and Implementation of Automated Teller Machine (FSM) Controller

College Name: PARUL UNIVERSITY

Team Name: Tech Girls
Team Member 1:
Name: Ayushi Jignesh Desai
Email: 200305105099@paruluniversity.ac.in
Mobile No.: 9409188369

Team Member 2:
Name: Siddhi AjitSingh Chaudhary
Email: 200305105089@paruluniversity.ac.in
Mobile No.: 7046486407

INDEX

LIST OF FIGURES

1. BLOCK DIAGRAM

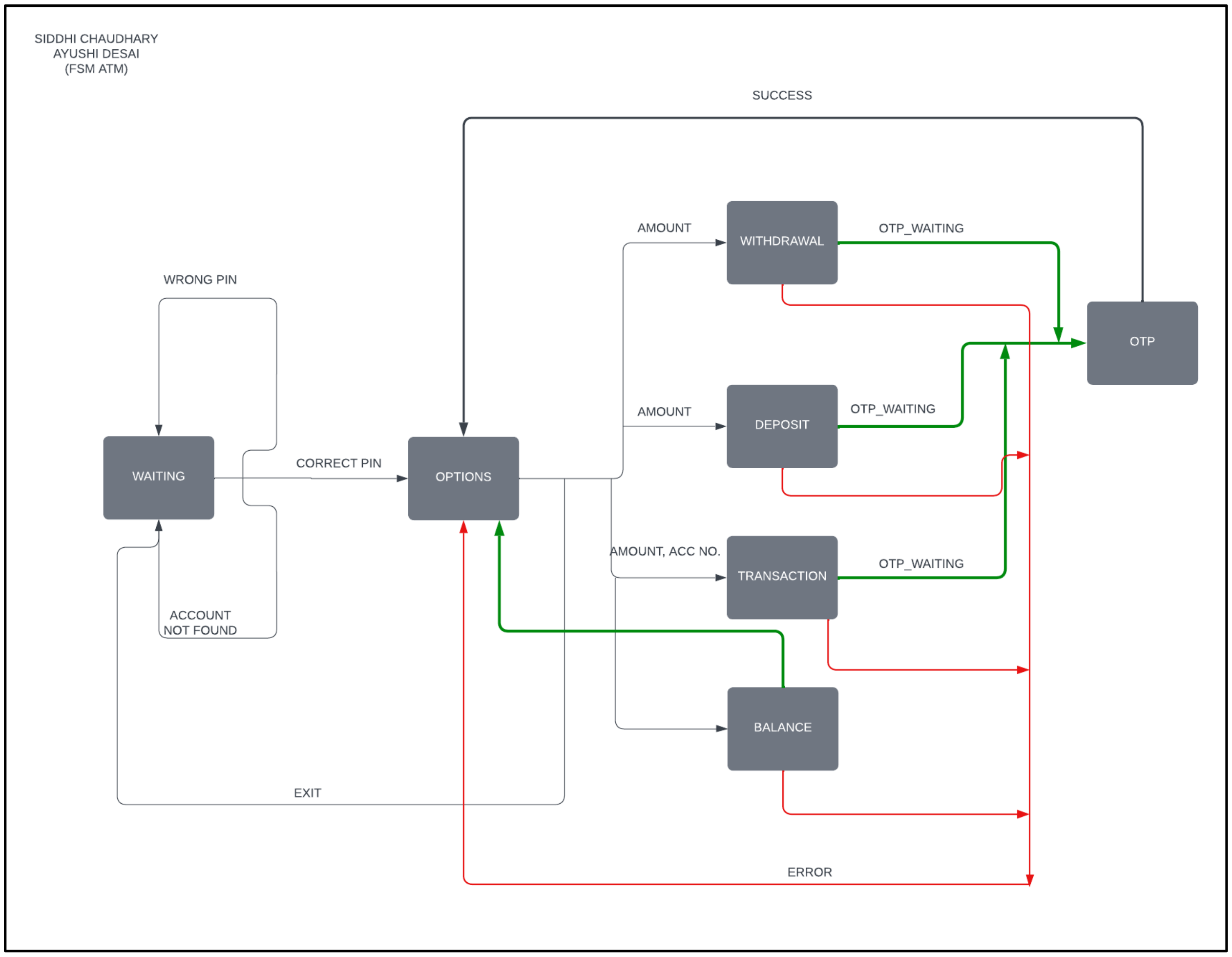


Fig. 1.1 Block Diagram

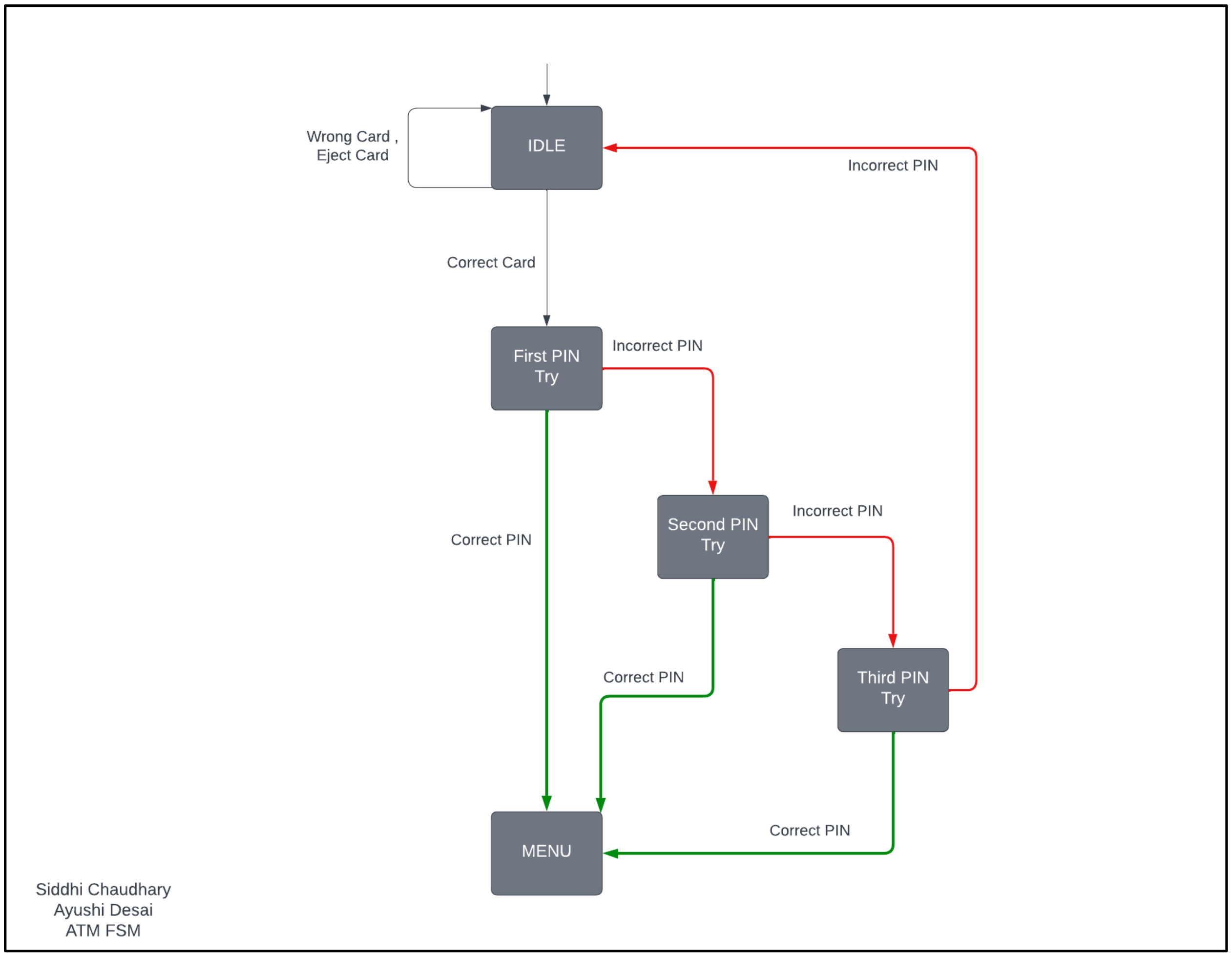


Fig. 1.2 PIN Try Block Diagram

3. APPROACH TO SOLVE THE PROBLEM

Our problem statement is: Design and Implementation of Automated Teller Machine (FSM) Controller in which we had to design a virtual Automated Teller Machine(ATM) based on two FSM(Finite State Machine): Mealy State Machine or Moore State Machine.

WEEK 1:

For the implementation of the problem statement we started with Block Diagram and its specification based on which our ATM would be created. And for the solution of our problem statement we have used the Mealy State Machine (FSM).
While developing the Block Diagram the specifications and the correct flow for the project was challenging through the First week.

WEEK 2:

For the second week of our training period we have started with the Verilog coding for our given FSM with the help of the Block Diagram. After we are finished with the Verilog coding with successful compilation of Analysis and Elaboration (No error) then we started to work on the Test Bench for our Verilog code to verify its functionalities.
Developing the Verilog code was the most difficult part of this project as it was very new to us for coding in Intel Quartus with the Verilog Language initially and it took us much more time than expected but we were able to come through it.

WEEK 3:

Successfully developed the Test Bench, integrated it with the Verilog code and Verification of the Specification for the given FSM. After the completion of the Verilog code and its Test Bench we started with the RTL Simulation and prosperously got the Waveform for our ATM and the RTL Viewer for the same.
Designing the Test Bench with effective and accurate test cases was a task to be done.

WEEK 4:

Developed the Pin Planner with our required pin for the project then successfully passed through the fitter and developed the Chip Planner. After developing the Chip planner we started with the Power Analyzer Tool and got the results.
Assigning the accurate pins with the pin planner took some time to work properly.

WEEK 5:

Started analyzing the code coverage, the area and power, the timing and proper timing constraints reports. Implemented the given design on the targeted FPGA. Developing the project report and uploading the pieces of the project on the Github repository. Finished with the Video Demo.

4. COMPLETE FLOW OF SOLUTION

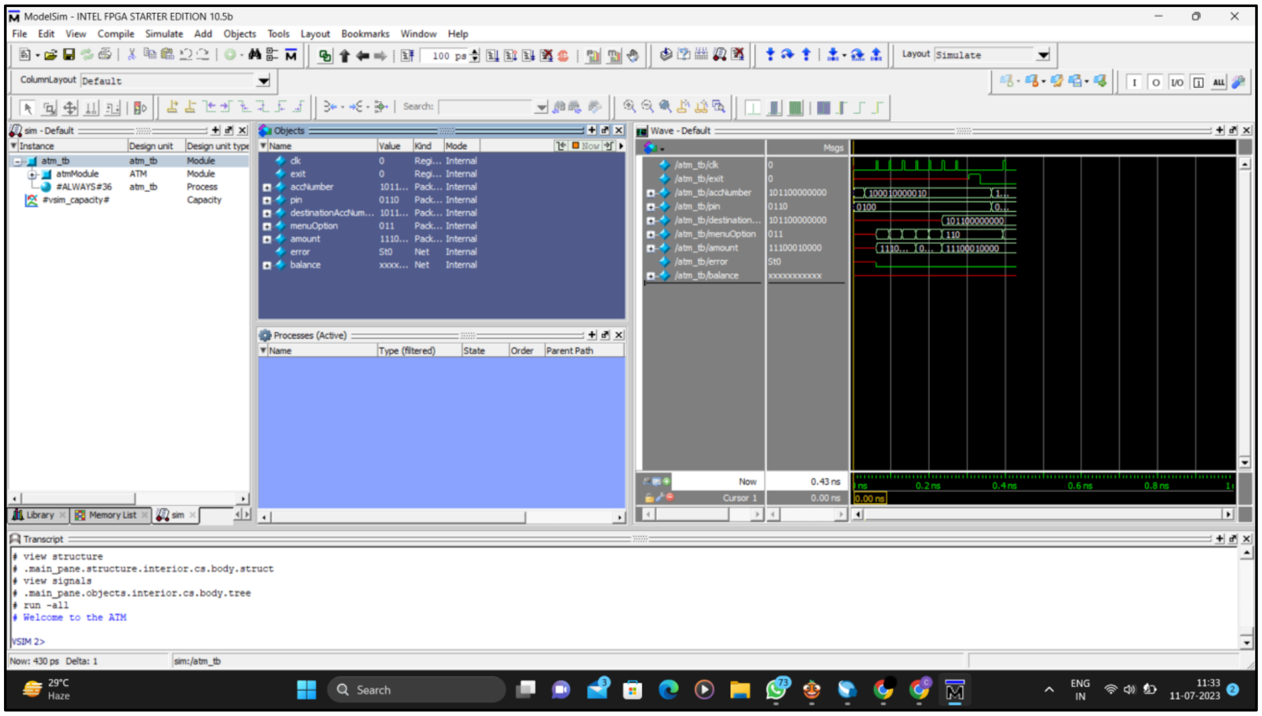
We started to solve our problems by coding the given statement. We considered the given criteria before beginning the coding portion. We coded an ATM using the FSM (mealy) we created. Our main concerns for the code other than the standard ATM was to exit if the pin was entered wrong 3 times and also if the amount entered to deposit or withdraw was greater than 10,000 then the code will switch to using OTP for more security.

To begin, we coded the basic ATM with states of Withdrawal, Deposit and Withdrawal with showing of balance. We first declared the basic states and params which were globally used throughout the code, and moreover created databases for Accounts and Pins consisting of account numbers and pins to corresponding account numbers. We added different inputs and output registers and wires to store the results and clock. And, further in coding we added OTP and Exiting of code if the pin was entered wrong three times. For OTP we added extra states to the existing ones and a condition if the amount entered was greater than 10,000 then the OTP state will be activated. And, for pin we took approach of for loop with condition if wrong pin was entered 3 times the code will move back to menu option.

Once the code was done, we started to code the test bench for the code with various cases which the code should successfully pass through. Some of the test cases were if the pin entered was wrong, withdrawing too much money, showing the balance etc. After successfully coding the test bench we performed a compiling of design which consisted of various steps. Afterwards, we moved on to the different steps of the process of solving the problem statement.

Firstly, we configured RTL simulation where we got a graph as output of the performance with various states on the y-axis. Then, we performed different netlist viewers (technology map viewer (post-mapping), RTL viewer, technology map viewer (post-fitting)). We faced difficulty in performing a state machine viewer. Proceeding, we assigned different pins in the pin planner to the outputs and inputs. On the basis of it a chip planner was formed. After performing the power analyzer tool, we got the temperature at 0 C for lower end and 85 C for higher end, also, 25 virtual pins were calculated.

5. NECESSARY FILES


ModelSim - INTEL FPGA STARTER EDITION 10.5b

Instance	Design unit	Design unit type
atm_tb	atm_tb	Module
atmModule	ATM	Module
#ALWAYS#36	atm_tb	Process
#vsim_capacity#		Capacity

Name	Value	Kind	Mode
clk	0	Regi...	Internal
exit	0	Regi...	Internal
accNumber	1011...	Pack...	Internal
pin	0110	Pack...	Internal
destinationAccNum...	1011...	Pack...	Internal
menuOption	011	Pack...	Internal
amount	1110...	Pack...	Internal
error	St0	Net	Internal
balance	xxxx...	Net	Internal

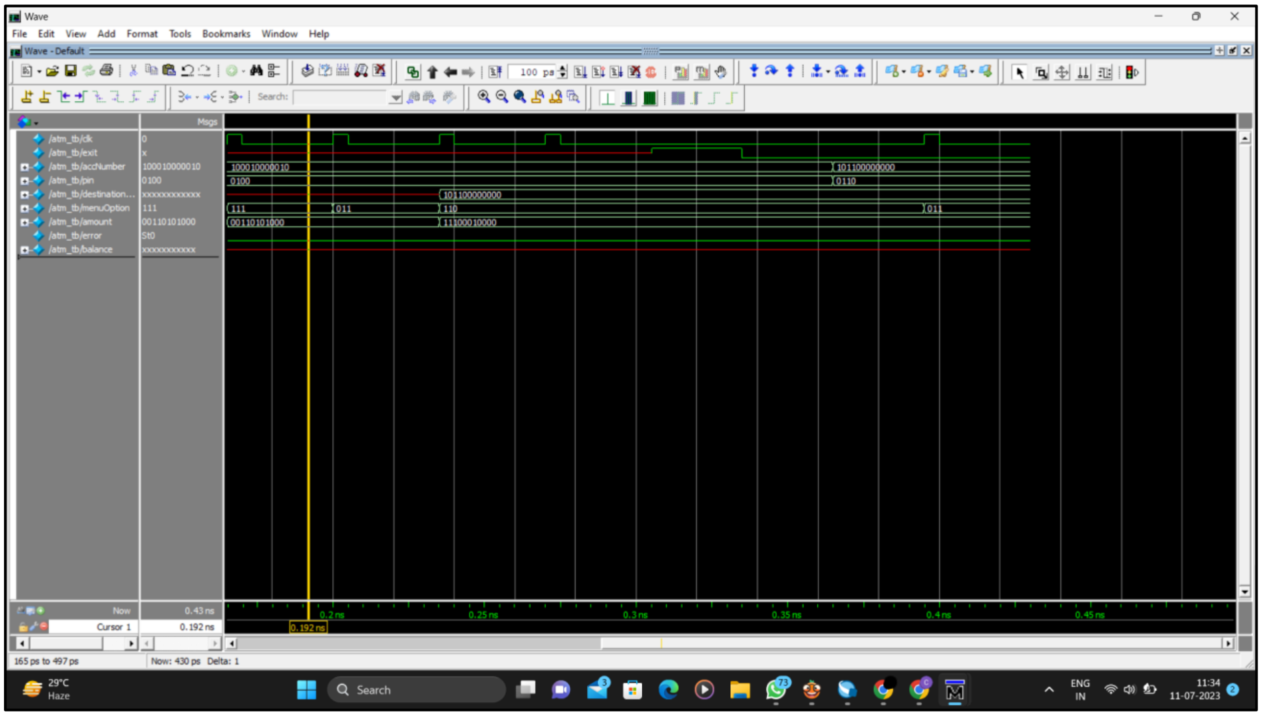
	Msgs
/atm_tb/clk	0
/atm_tb/exit	0
/atm_tb/accNumber	101100000000
/atm_tb/pin	0110
/atm_tb/destination...	101100000000
/atm_tb/menuOption	011
/atm_tb/amount	11100010000
/atm_tb/error	St0
/atm_tb/balance	xxxxxxxxxxxx

```
# view structure
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run -all
# Welcome to the ATM

VSIM 2>
```

Now: 430 ps Delta: 1 sim:/atm_tb

Fig. 5.1 RTL Simulation


Wave - Default

	Msgs
/atm_tb/clk	0
/atm_tb/exit	x
/atm_tb/accNumber	100010000010
/atm_tb/pin	0100
/atm_tb/destination...	xxxxxxxxxxxx
/atm_tb/menuOption	111
/atm_tb/amount	00110101000
/atm_tb/error	St0
/atm_tb/balance	xxxxxxxxxxxx

Now: 0.43 ns Cursor 1: 0.192 ns

165 ps to 497 ps Now: 430 ps Delta: 1

Fig. 5.2 WaveForm

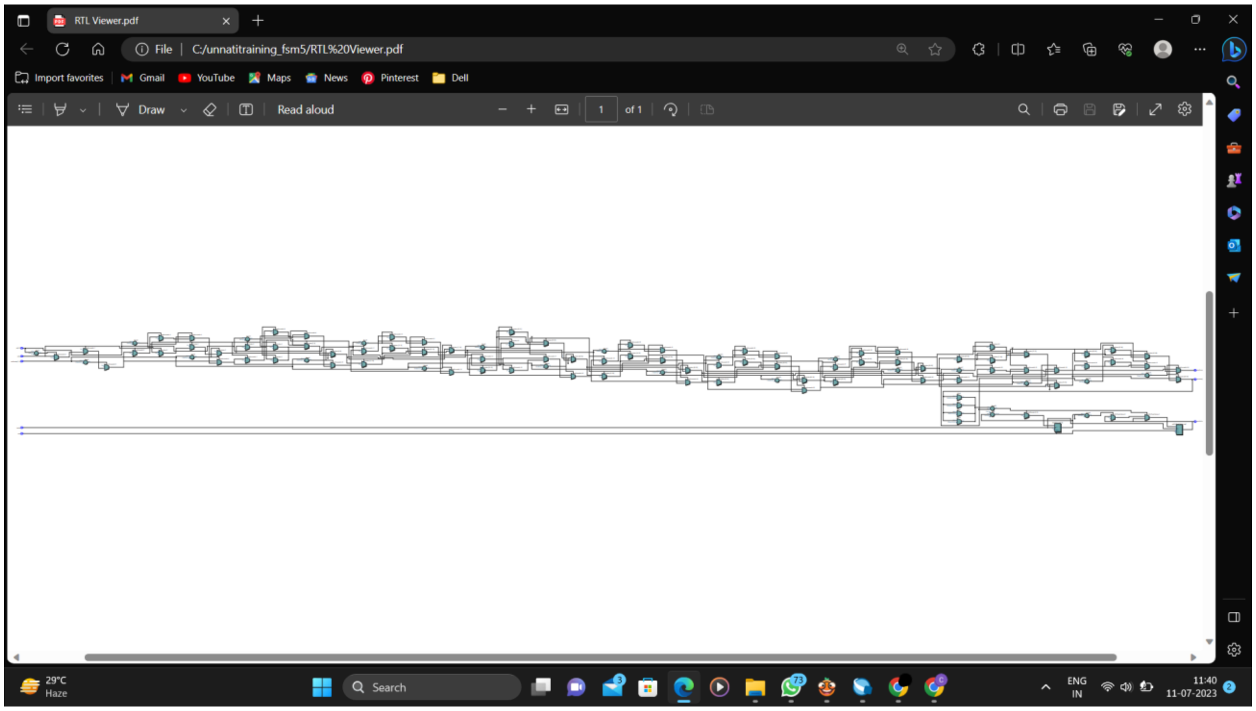


Fig. 5.3 RTL Viewer

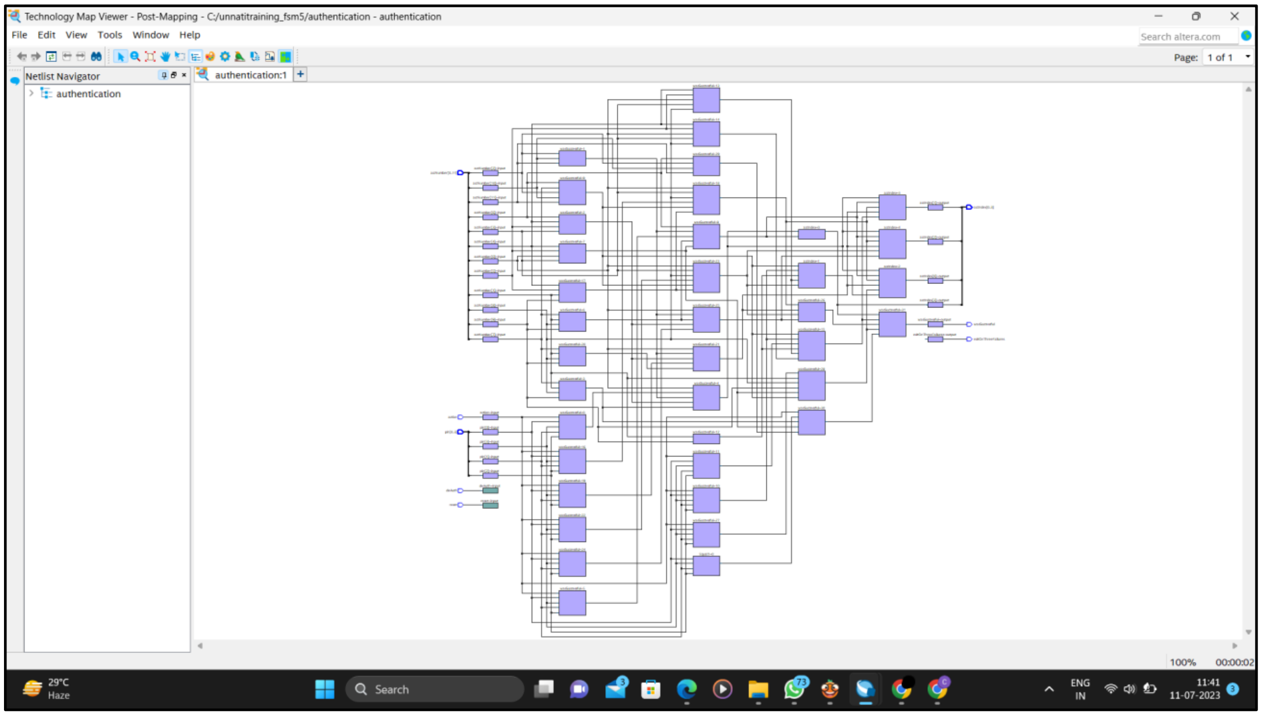

Technology Map Viewer - Post-Mapping - C:/unnatitraining_fsm5/authentication - authentication
File Edit View Tools Window Help
Netlist Navigator
authentication
authentication:1
Page: 1 of 1
100%
00:00:02

Fig. 5.4 Technology Map Viewer (Post Mapping)

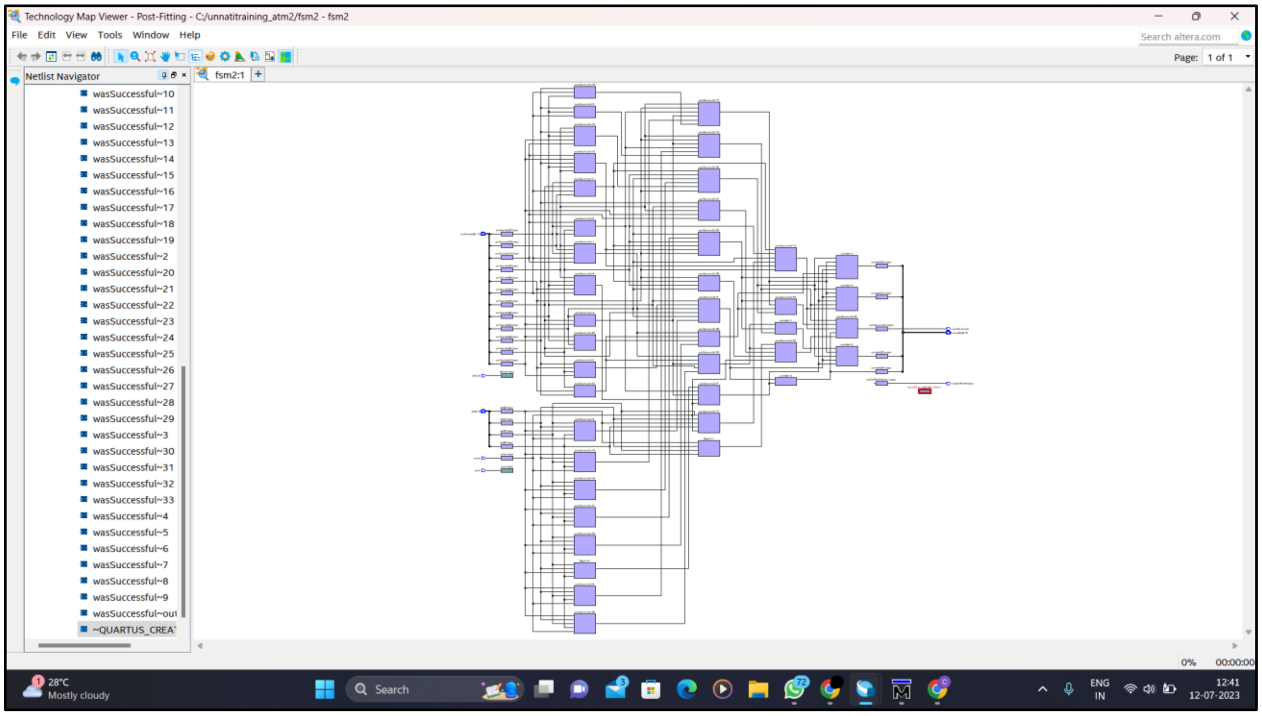
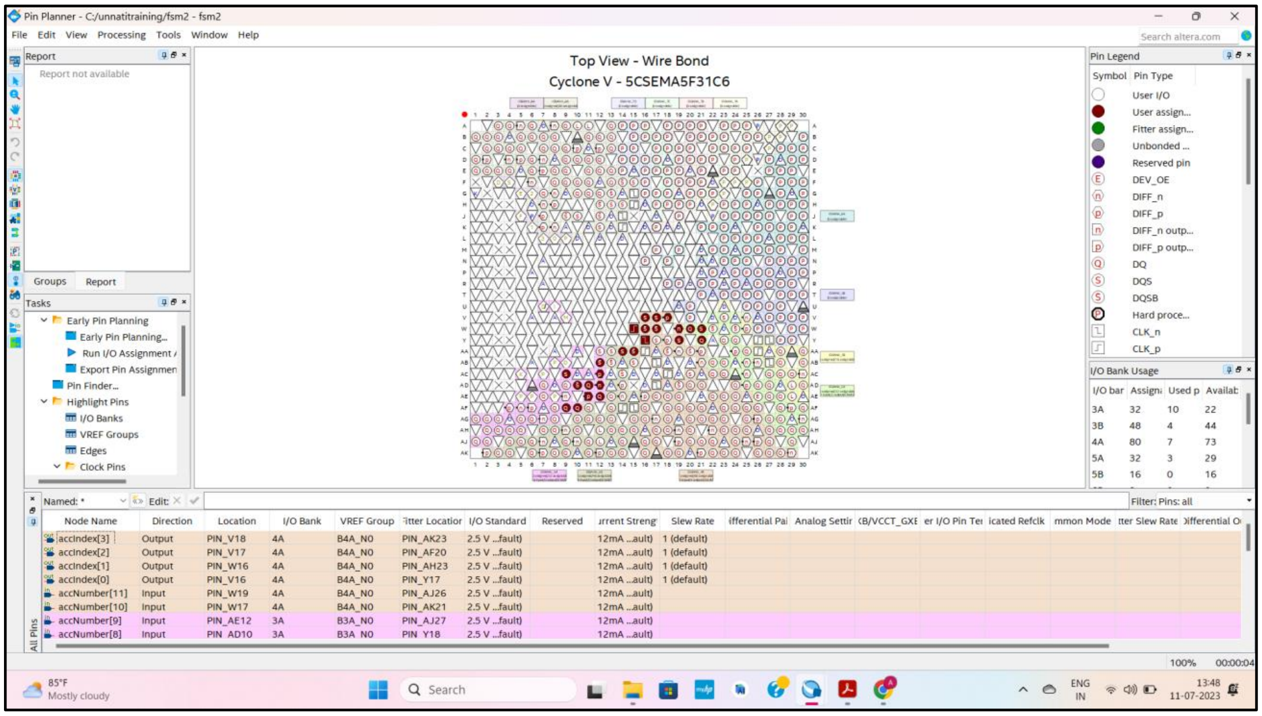


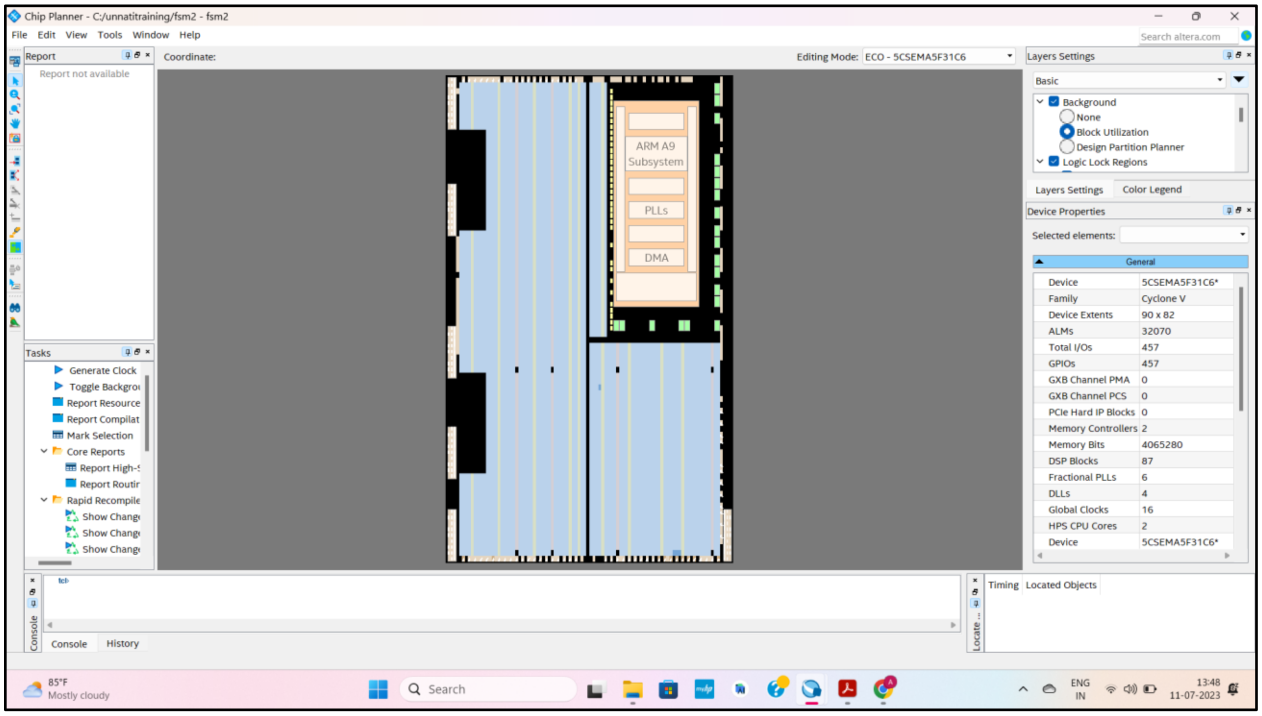
Fig. 5.5 Technology Map Viewer (Post Fitting)



I/O bank	Assigned	Used p	Available
3A	32	10	22
3B	48	4	44
4A	80	7	73
5A	32	3	29
5B	16	0	16

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate
accIndex[3]	Output	PIN_V18	4A	B4A_N0	PIN_AK23	2.5 V ...fault)		12mA ...ault)	1 (default)
accIndex[2]	Output	PIN_V17	4A	B4A_N0	PIN_AF20	2.5 V ...fault)		12mA ...ault)	1 (default)
accIndex[1]	Output	PIN_W16	4A	B4A_N0	PIN_AH23	2.5 V ...fault)		12mA ...ault)	1 (default)
accIndex[0]	Output	PIN_V16	4A	B4A_N0	PIN_Y17	2.5 V ...fault)		12mA ...ault)	1 (default)
accNumber[11]	Input	PIN_W19	4A	B4A_N0	PIN_AJ26	2.5 V ...fault)		12mA ...ault)	
accNumber[10]	Input	PIN_W17	4A	B4A_N0	PIN_AK21	2.5 V ...fault)		12mA ...ault)	
accNumber[9]	Input	PIN_AE12	3A	B3A_N0	PIN_AJ27	2.5 V ...fault)		12mA ...ault)	
accNumber[8]	Input	PIN_AD10	3A	B3A_N0	PIN_Y18	2.5 V ...fault)		12mA ...ault)	

Fig. 5.6 Pin Planner



Coordinate:

Editing Mode: ECO - 5CSEMA5F31C6

Layers Settings: Basic

- [✓] Background
  - None
  - [● Block Utilization]
  - Design Partition Planner
- [✓] Logic Lock Regions

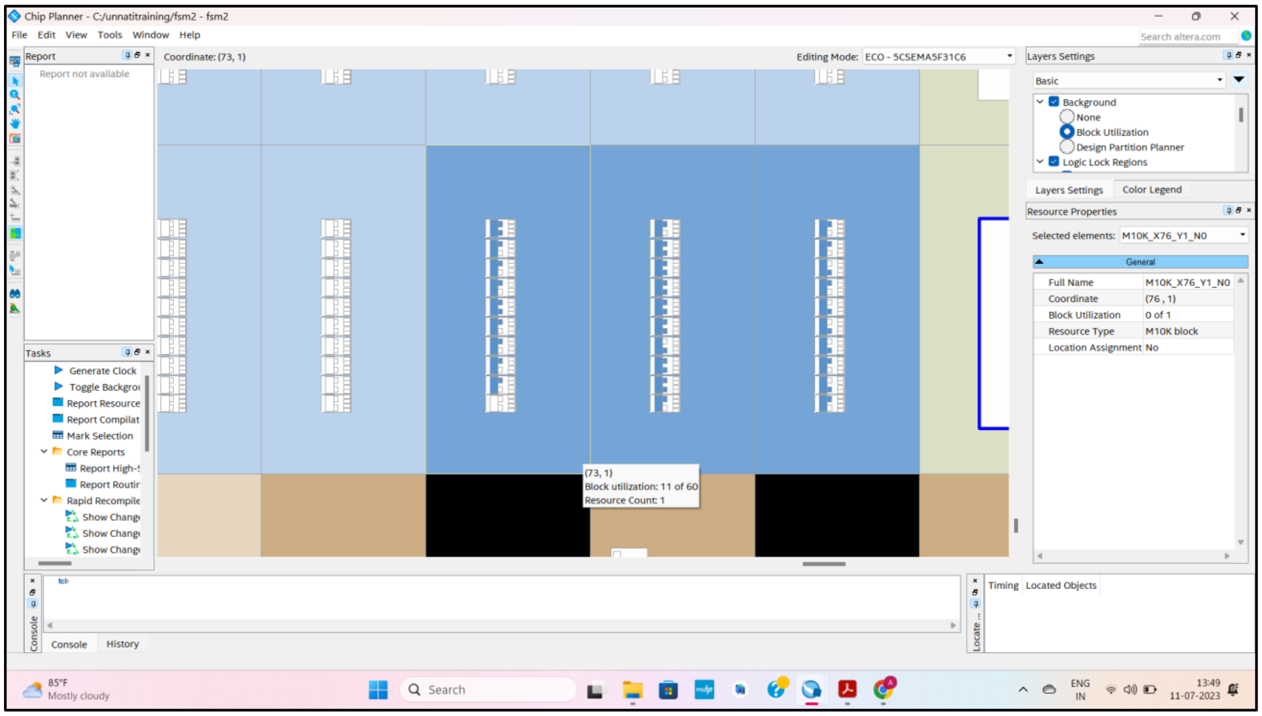
Device Properties

Selected elements:

General

Device	5CSEMA5F31C6*
Family	Cyclone V
Device Extents	90 x 82
ALMs	32070
Total I/Os	457
GPIOs	457
GXB Channel PMA	0
GXB Channel PCS	0
PCIe Hard IP Blocks	0
Memory Controllers	2
Memory Bits	4065280
DSP Blocks	87
Fractional PLLs	6
DLLs	4
Global Clocks	16
HPS CPU Cores	2
Device	5CSEMA5F31C6*

Fig. 5.7 Chip Planner



Coordinate: (73, 1)

Editing Mode: ECO - 5CSEMA5F31C6

(73, 1)
Block utilization: 11 of 60
Resource Count: 1

Layers Settings: Basic

- [✓] Background
  - None
  - [● Block Utilization]
  - Design Partition Planner
- [✓] Logic Lock Regions

Resource Properties

Selected elements: M10K_X76_Y1_N0

General

Full Name	M10K_X76_Y1_N0
Coordinate	(76 , 1)
Block Utilization	0 of 1
Resource Type	M10K block
Location Assignment	No

Fig. 5.7(a) Chip Planner

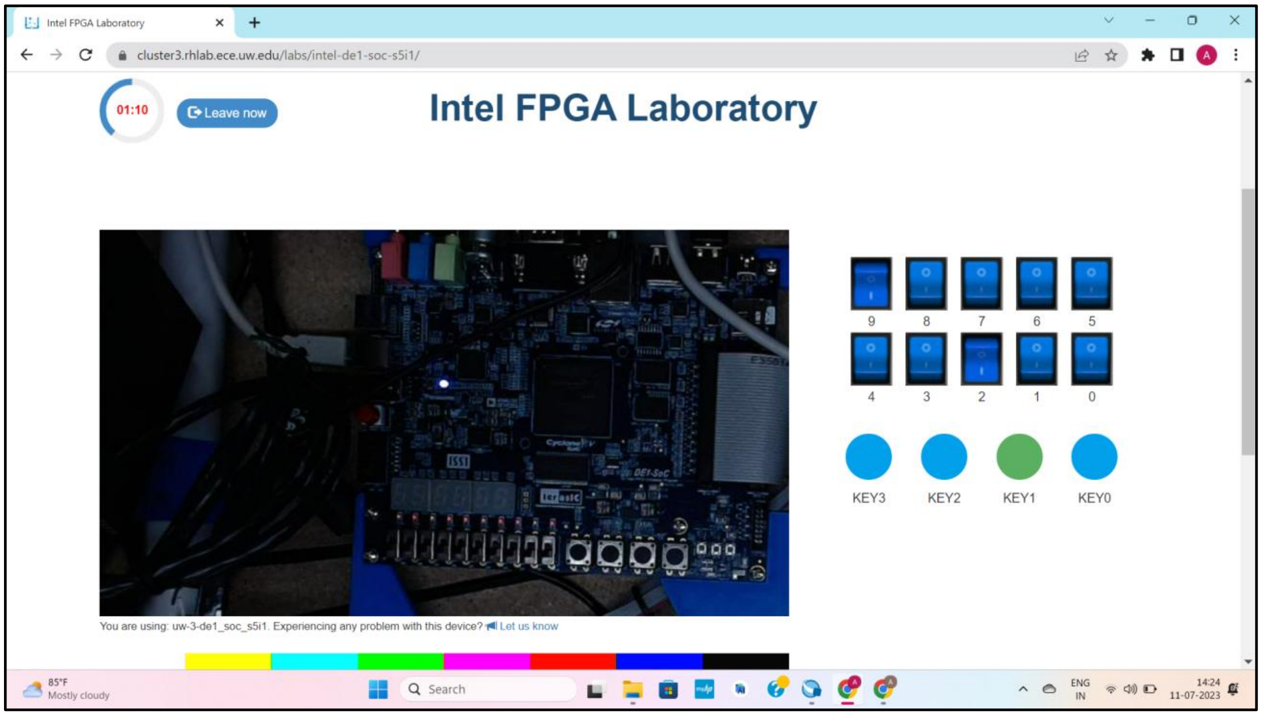
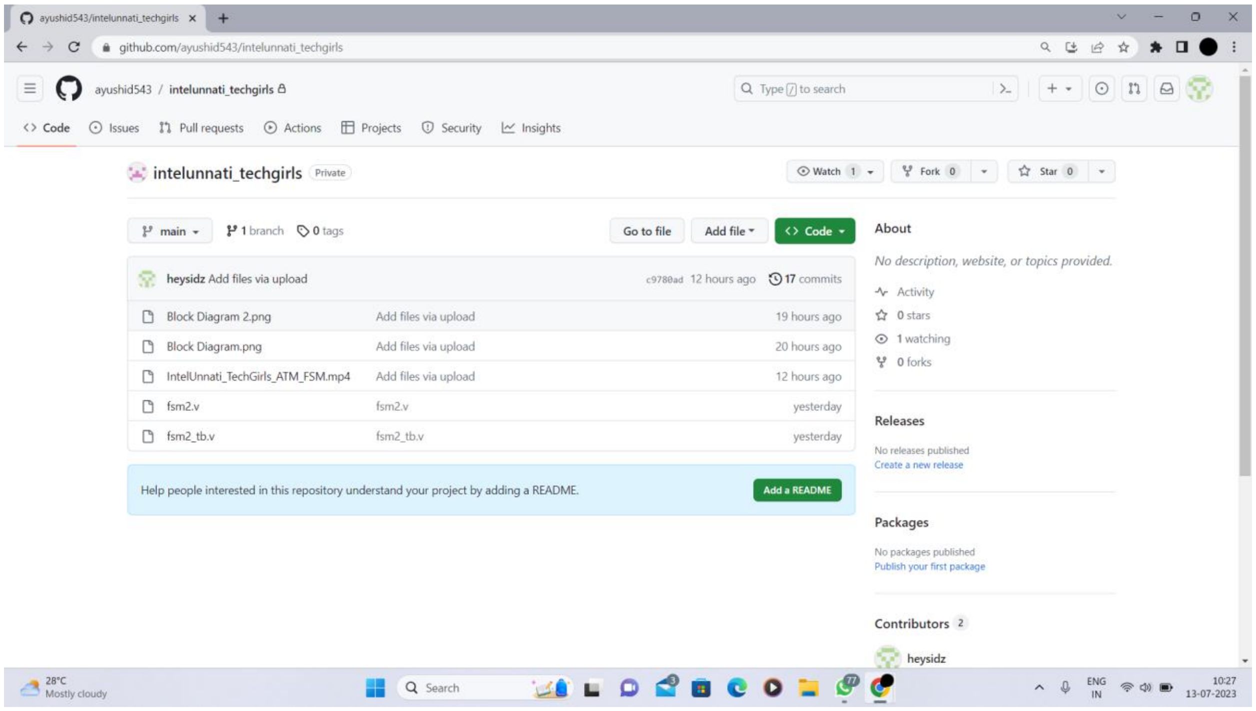

Intel FPGA Laboratory
cluster3.rhlab.ece.uw.edu/labs/intel-de1-soc-s5i1/
01:10
Leave now
Intel FPGA Laboratory
9
8
7
6
5
4
3
2
1
0
KEY3
KEY2
KEY1
KEY0
You are using: uw-3-de1_soc_s5i1. Experiencing any problem with this device? Let us know

Fig. 5.8 Labsland



ayushid543 / intelunnati_techgirls

intelunnati_techgirls Private

main · 1 branch · 0 tags

heysidz Add files via upload · c9780ad 12 hours ago · 17 commits

Block Diagram 2.png	Add files via upload	19 hours ago
Block Diagram.png	Add files via upload	20 hours ago
IntelUnnati_TechGirls_ATM_FSM.mp4	Add files via upload	12 hours ago
fsm2.v	fsm2.v	yesterday
fsm2_tb.v	fsm2_tb.v	yesterday

Help people interested in this repository understand your project by adding a README. Add a README

About

No description, website, or topics provided.

Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 2

heysidz

Fig. 5.9 GitHub Repository

6. VIDEO DEMONSTRATION

https://github.com/ayushid543/intelunnati_techgirls/blob/main/IntelUnnati_TechGirls_ATM_FSM.mp4

7. RESULTS

According to our problem statement: Design and Implementation of Automated Teller Machine (FSM) Controller we have designed a Automated Teller Machine (FSM) Controller using Intel Quartus with Verilog coding language. Specification for this ATM (FSM) Controller are the options for Withdrawal, Deposit, View Balance and View Balance after Withdrawal or Deposition. For the PIN entry we kept 3 trials in case of incorrect PIN insertion and after 3 attempts the account will be locked. In case of withdrawal of more than Rs 10,000 then there is an OTP sent to the user's registered mobile number for authentication along with the PIN entry. And same goes in case of Deposition.

8. SUMMARY

We have completed the given problem statement to the best of our abilities from making ourselves familiar with the topic to executing our ideas. We started with making a block diagram and continuing to code and then test our written code and analyzing it. The project at hand has encountered several challenges throughout its execution. These challenges have presented themselves in various forms, posing obstacles and requiring adaptive strategies to overcome them. Some of them are us being complete novices at working with this technology and others just a part of the process. Despite these constraints we have successfully designed an ATM using FSM with functionalities like OTP and exiting the code after three unsuccessful attempts at entering the pin. We have learnt immensely from this project and are grateful for the opportunity that was provided (and will continue to work and make it optimal code).