INTEL UNNATI INDUSTRIAL TRAINING

SUMMER 2023.

DESIGN AND IMPLEMENTATION OF AUTOMATED TELLER MACHINE (FSM) CONTROLLER.

TEAM NAME : InnovateX

AUTHORS:

K.N.V.TANUSRI

E.NEHA NARMADA

DATE OF SUBMISSION: JULY 15 2023.

ABSTRACT

The Automated Teller Machine (ATM) is a device that allows customers to perform various banking transactions, such as withdrawing cash, depositing funds, and checking account balances, without the need for human assistance. The controller of an ATM is responsible for managing and controlling the interactions between the user and the machine.

To represent the behavior of the ATM controller, a Finite State Machine (FSM) can be used. An FSM is a mathematical model that consists of a set of states, transitions between those states, and actions associated with those transitions. It provides a way to describe the different states the ATM controller can be in and how it transitions from one state to another based on user input and system events.

INTRODUCTION

An Automated Teller Machine ATM is a safety-critical and real-time system that is highly complicated in design and implementation. An Automated Teller Machine ATM is a safety-critical and real-time system that is highly complicated in design and implementation. An ATM system is a real-time front terminal of automatic teller services with the support of a central bank server and a centralized account database.

An ATM that provides money withdraw and account balance management services. The architecture of the ATM system,encompasses an ATM processor, a system clock, a remote account database, and a set of peripheral devices such as the card reader,monitor, keypad, bills storage, and bills disburser.

An Automated Teller Machine is an electronic device that allows bank customers to perform various financial transactions without the need for a human teller. It typically provides services such as cash withdrawals, balance inquiries, funds transfers, deposits, bill payments, and more.

Designing and implementing an entire ATM system is a complex task that involves multiple components, including hardware, software, and network integration.

The implementation of an ATM system requires collaboration between hardware engineers, software developers, network specialists, and security experts. It involves designing the user interface, developing the software components, integrating with backend systems, ensuring security measures, and testing the system thoroughly. ATMs are convenient, allowing consumers to perform quick self-service transactions such as deposits, cash withdrawals, bill payments, and transfers between accounts.

ATM stands for Automated Teller Machine which is a self-service banking outlet. You can withdraw money, check your balance, or even transfer funds at an ATM.

ATM transactions are either free or bear a nominal charge depending upon the banks. Banks usually do not charge for the first 3-5 ATM transactions in a month. Once you cross the limit of free transactions, you may have to pay a nominal charge. Also, some banks levy charges if you withdraw money from the ATM of another bank of which you are not an account holder.

An ATM (Automated Teller Machine) is an electronic machine used for financial transactions. As the term implies, it is an ‘automated’ banking platform that does not require any banking representative/teller or a human cashier. Automated Teller Machines have revolutionized the banking sector by providing easy access to customers and loading off the burden from bank officials.

ATMs have multiplied the accessibility and convenience of the people. It helps people to have cash in times of urgent need.

MOTIVATION

The motivation behind solving the problem of designing and implementing an Automated Teller Machine (ATM) FSM controller lies in the numerous benefits and conveniences that ATMs offer to both banks and their customers. Some key motivations for solving this problem such as Accessibility Convenience, Efficiency, Time Savings, Extended Banking Services, Cost Savings for Bank, Increased Customer Satisfaction and more.

ATMs provide customers with round-the-clock access to banking services. With an ATM controller, customers can perform various transactions at their convenience, such as withdrawing cash, checking balances, transferring funds, and making deposits, without the need to visit a physical bank branch during limited operating hours.

By automating routine banking transactions, ATMs help reduce the time and effort required for both customers and bank staff. Customers can quickly perform transactions without waiting in long queues, while banks can handle a higher volume of transactions efficiently.

ATMs can offer a wide range of banking services beyond basic cash withdrawals. With an FSM controller, additional features like balance inquiries, bill payments, account transfers, cardless transactions, and personalized services can be implemented, enhancing the overall customer experience.

ATMs reduce the need for banks to maintain a large network of physical branches and employ a large number of tellers. This can result in significant cost savings for banks, allowing them to allocate resources more efficiently.

FINITE STATE MACHINE (FSM) MODEL

In a Automated Teller Machine (ATM) implemented using a Mealy Machine, the behaviour and transitions of the ATM system are determined by the current state and the input received at the moment.

The Mealy machine is a type of Finite State Machine(FSM) where outputs are associated with the transitions between the states.

The FSM model used in Verilog code of the ATM controller consists of different states, transitions and outputs. The explanation for FSM model :

States:

Idle: The initial state where the ATM is waiting for user interaction.

Card_Inserted: The state when a valid card is inserted.

PIN_Entered: The state when the user enters a PIN.

Transaction_Selected: The state when the user selects a transaction type.

Transaction_Details_Entered: The state when the user enters transaction details.

Transaction_Processing: The state when the transaction is being processed.

Transaction_Completed: The state when the transaction is successfully completed.

Transaction_Failed: The state when the transaction encounters an error or fails.

Define the inputs:

clk: The clock signal for synchronous operation.

reset: The reset signal to initialize the ATM.

card_insert: A signal indicating if a card is inserted.

pin_enter: A signal indicating if the user enters a PIN.

txn_select: A signal indicating if the user selects a transaction type.

txn_confirm: A signal indicating if the user confirms the transaction.

Define the outputs:

display_message: An output indicating messages or instructions for the user.

cash_dispense: An output indicating if cash is dispensed.

receipt_print: An output indicating if a receipt is printed.

transaction_completed: An output indicating if the transaction is successfully completed.

transaction_failed: An output indicating if the transaction encounters an error or fails.

Determine the state transitions and outputs:

Based on the inputs and current state, define the next state and the corresponding outputs using conditional statements.

Implement the Mealy FSM controller:

Use VHDL or Verilog to write the code for the Mealy FSM controller based on the defined states, inputs, and outputs.

Use sequential logic to update the current state based on the clock and reset signals.

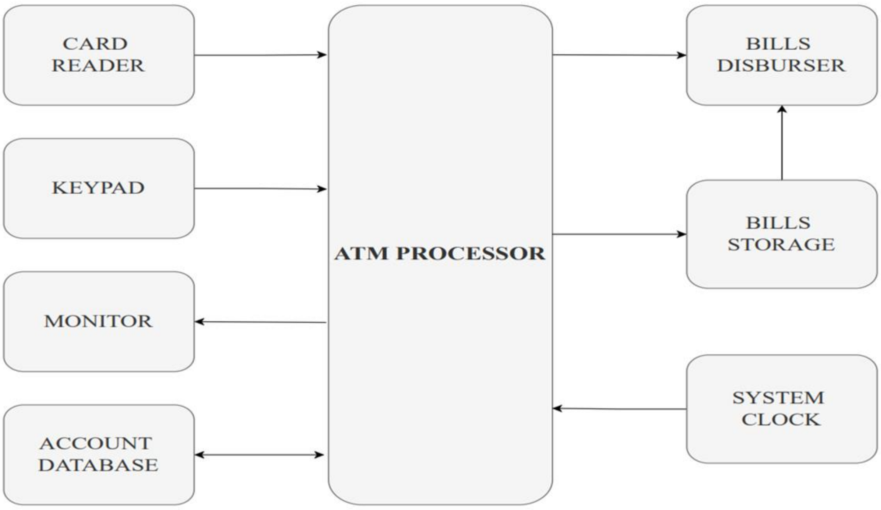
Use combinational logic to determine the next state and outputs based on the current state and inputs.

Connect the Mealy FSM controller to the ATM components:

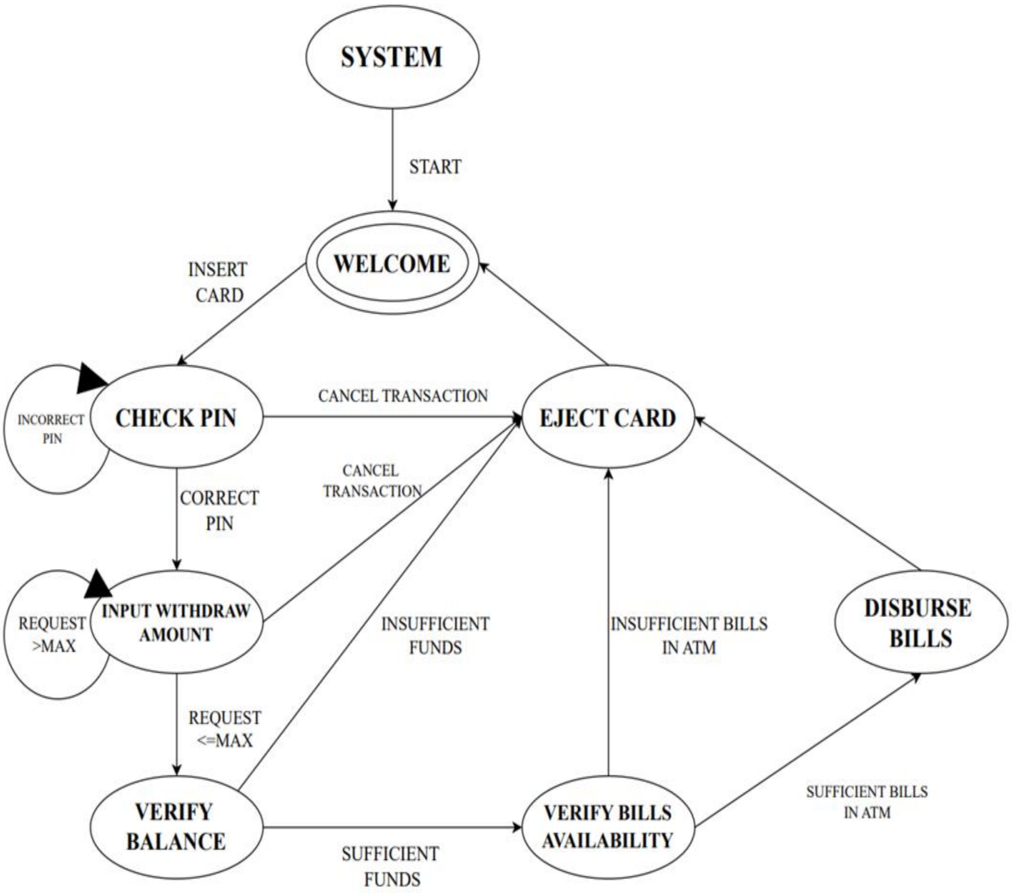
Connect the outputs of the Mealy FSM controller to the respective components in the ATM, such as the display, cash dispenser, receipt printer, etc.

Implement the necessary logic in the ATM components to respond to the Mealy FSM outputs accordingly.

BLOCK DIAGRAM



STATE MODEL

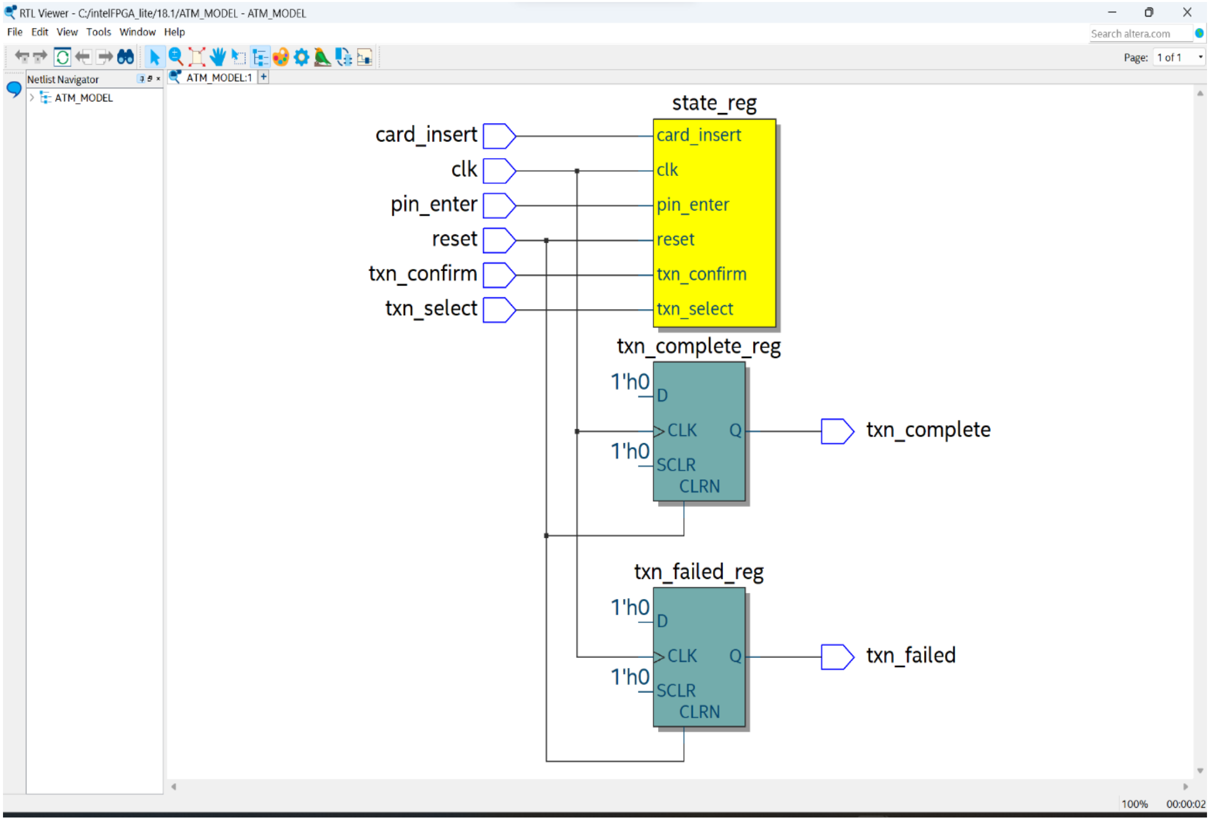


APPROACH

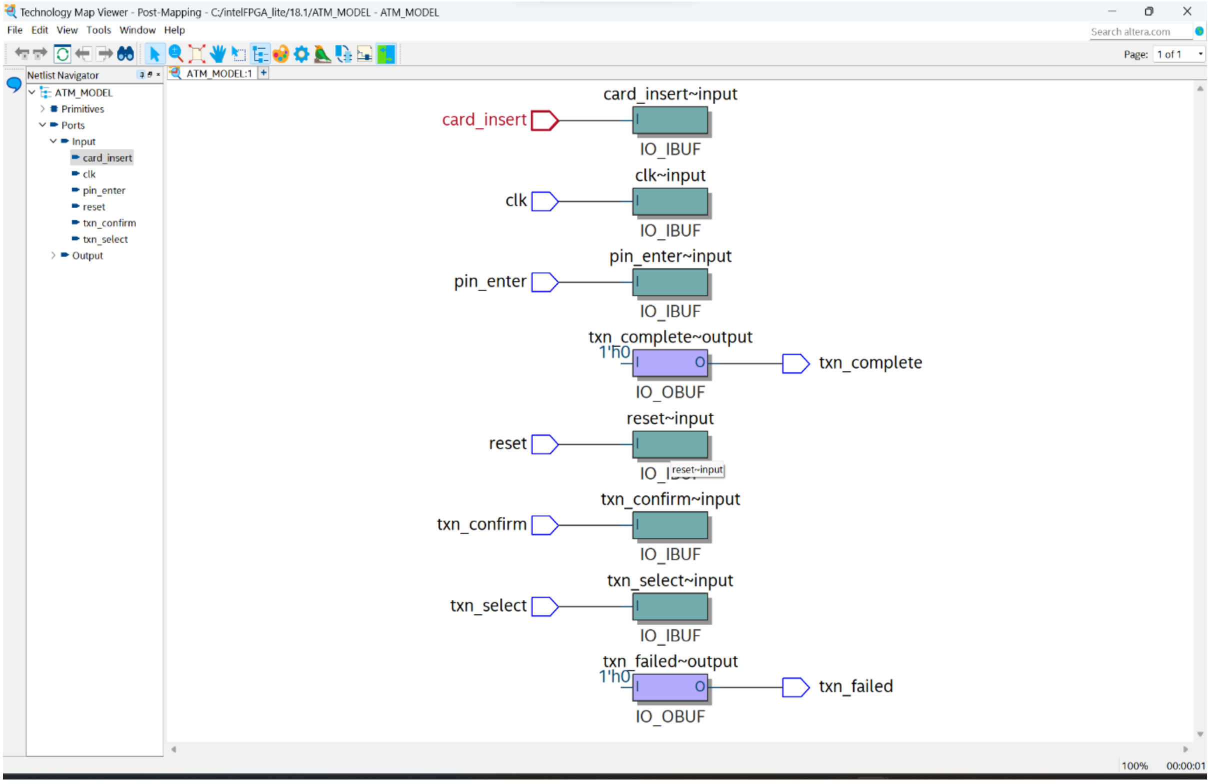
- **Requirements Analysis:** Gather and analyze the requirements for the ATM system. Here, we need to install the "Quartus Prime Lite Edition" from intel.com
- **Set-up:** After installation a file is downloaded. Open the downloaded file and select setup. "Quartus Prime Lite Edition" will be successfully installed in your system.
- **Implementation:** Open "Quartus Prime Lite Version" and select "New project wizard". Give a directory name and select your Project name. Select the type of project. Here, we have selected an "Empty Project". Give a family name(Cyclone V) and add a filter (5CSEMA5F31C6).Check the directory and filter that you've added. Now, Click on Finish. A directory with the required filter is created successfully.
- **Transaction Processing:** Implement the logic for processing different types of transactions. Go to File , select new , click on "Verilog HDL File". Write the required code into the Verilog. Save the Verilog code with a required file name.
- **Error Handling:** Analysis and synthesis has to be done. Check if there are any errors, if there is any error in the code then rewrite the required code. We must write a Test Bench Code for the previous code. You are supposed to save the code with .v extension.
- **Testing:** Go to new file , select new , click on "Verilog HDL File". Verilog 2 is created. Now, add test bench code for the previous code and perform analysis and synthesis on test bench code.
- **Deployment:** Deploy he FSM controller to the target environment, which may involve integrating with hardware components, network interfaces and backend systems. Plan for regular maintenance to ensure the continued smooth operation of the ATM system.

RESULTS

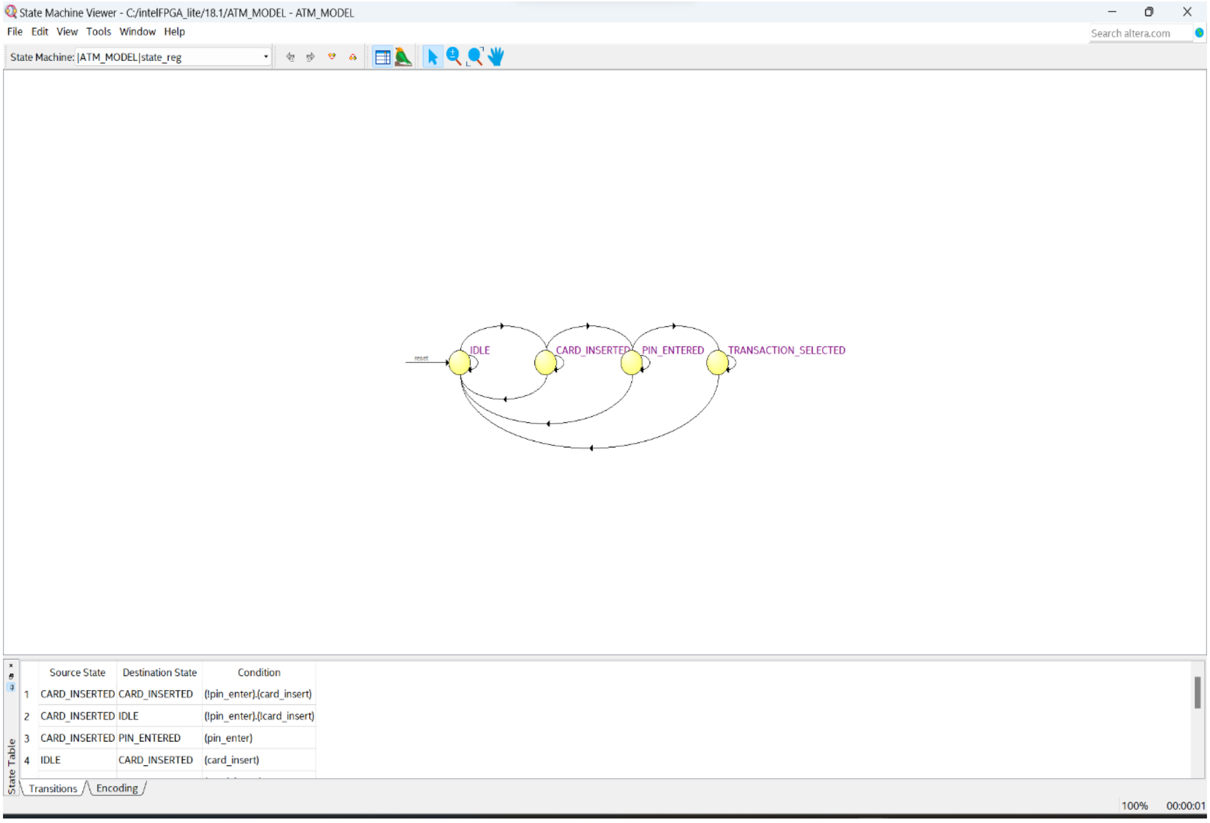
RTL Viewer:



Post Fitting View:

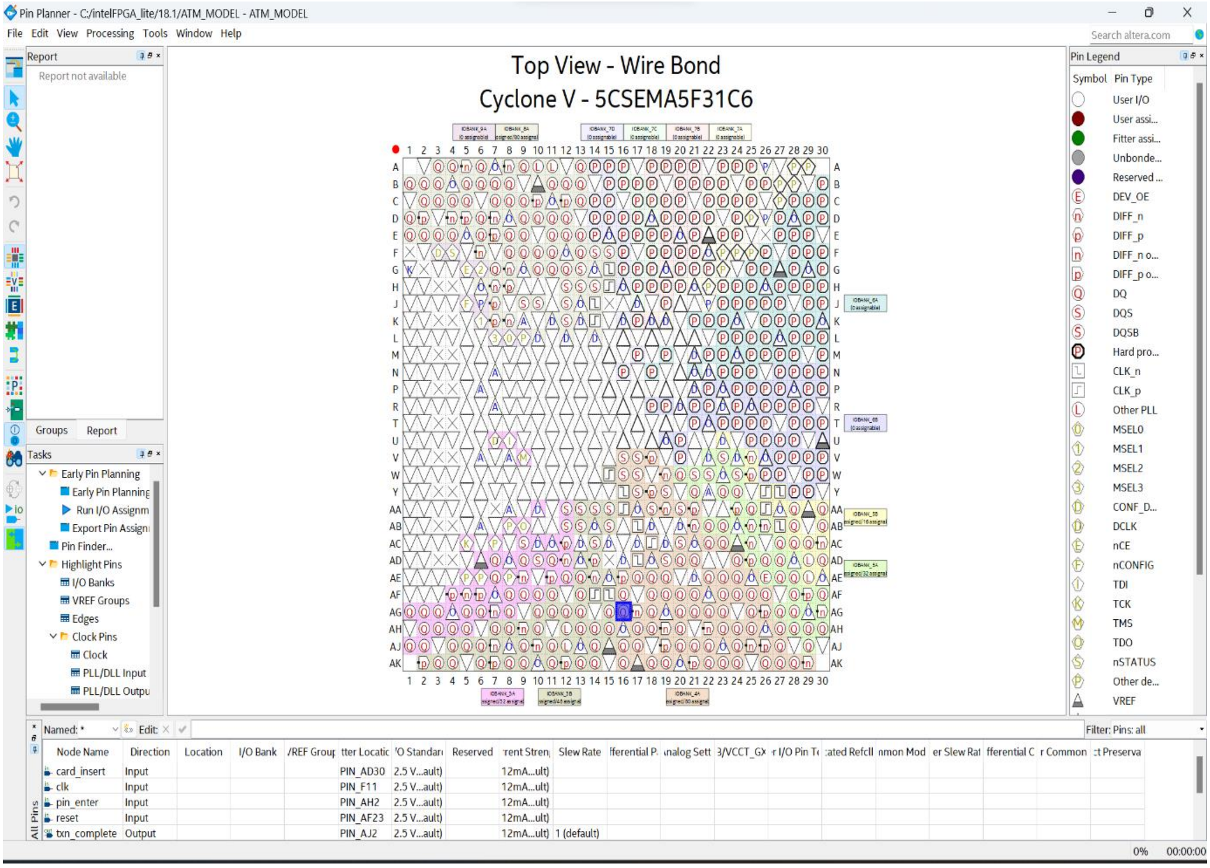


State Transition Diagram:


State Machine Viewer - C:/intelFPGA_lite/18.1/ATM_MODEL - ATM_MODEL
File Edit View Tools Window Help
Search altera.com
State Machine: |ATM_MODEL|state_reg
reset
IDLE
CARD_INSERTED
PIN_ENTERED
TRANSACTION_SELECTED
State Table
Transitions
Encoding
100%
00:00:01

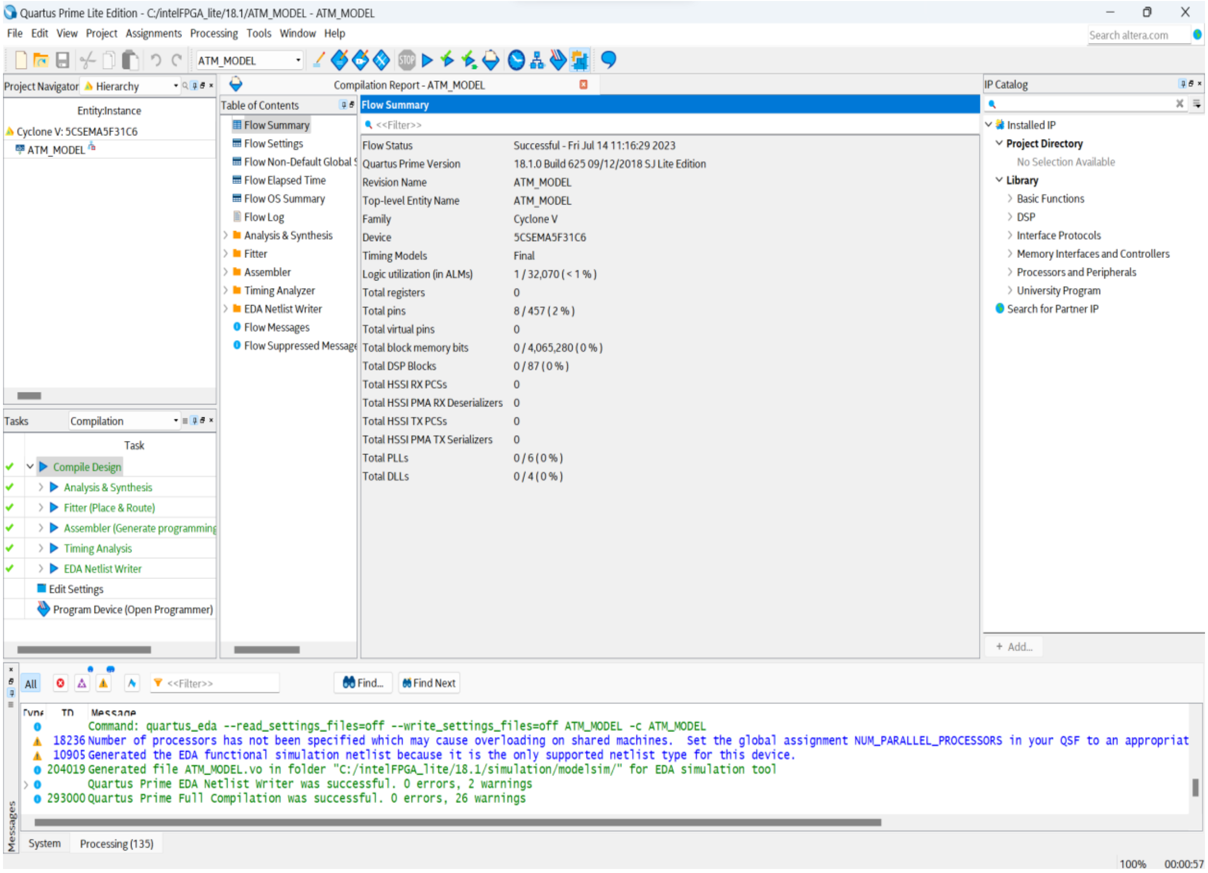
	Source State	Destination State	Condition
1	CARD_INSERTED	CARD_INSERTED	(!pin_enter).(card_insert)
2	CARD_INSERTED	IDLE	(!pin_enter).(!card_insert)
3	CARD_INSERTED	PIN_ENTERED	(pin_enter)
4	IDLE	CARD_INSERTED	(card_insert)

Pin Planner:



Node Name	Direction	Location	I/O Bank	/REF Grouр	tter Locatic	'O Standar	Reserved	rent Stren	Slew Rate	fferential P.	nalog Sett	3/VCCT_GX	r I/O Pin T	ated Refcll	nmon Mod	er Slew Rat	fferential C	r Common	t Preserva
card_insert	Input				PIN_AD30	2.5 V...ault)		12mA...ult)											
clk	Input				PIN_F11	2.5 V...ault)		12mA...ult)											
pin_enter	Input				PIN_AH2	2.5 V...ault)		12mA...ult)											
reset	Input				PIN_AF23	2.5 V...ault)		12mA...ult)											
txn_complete	Output				PIN_AJ2	2.5 V...ault)		12mA...ult)	1 (default)										

Compilation Report:



Flow Summary

Flow Status	Successful - Fri Jul 14 11:16:29 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	ATM_MODEL
Top-level Entity Name	ATM_MODEL
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	1 / 32,070 (< 1 %)
Total registers	0
Total pins	8 / 457 (2 %)
Total virtual pins	0
Total block memory bits	0 / 4,065,280 (0 %)
Total DSP Blocks	0 / 87 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 4 (0 %)

```
Type  ID     Message
            Command: quartus_eda --read_settings_files=off --write_settings_files=off ATM_MODEL -c ATM_MODEL
      18236 Number of processors has not been specified which may cause overloading on shared machines.  Set the global assignment NUM_PARALLEL_PROCESSORS in your QSF to an appropriat
      10905 Generated the EDA functional simulation netlist because it is the only supported netlist type for this device.
     204019 Generated file ATM_MODEL.vo in folder "C:/intelFPGA_lite/18.1/simulation/modelsim/" for EDA simulation tool
            Quartus Prime EDA Netlist Writer was successful. 0 errors, 2 warnings
     293000 Quartus Prime Full Compilation was successful. 0 errors, 26 warnings
```

As a result the Automated Teller Machine FSM controller depends on the specific implementation and the actions associated with each state transition. In general, the FSM controller manages the flow of the ATM interactions and ensures that the user can perform banking transactions accurately and securely. ATM is the easiest way of depositing and withdrawing money.

If ATM machines are connected to the internet then it is possible to do transaction from any where, 24 hours a day and 365 days an year. With the security of ATM improving it has now become a safe mode of transaction. Hence it can be concluded that ATM is a safe, fast, reliable, convenient, excisable.

We have utilized the Verilog code and Intel Quartus Prime Lite software and performed various ATM operations such as card insertion, PIN verification and transaction processing.

REFERENCES

- G. C. Sacket and C. Y. Metz, "ATM and Multiprotocol Networking," McGraw-Hill, 1996.
- H. Dutton and Peter Lenhard, "Asynchronous Transfer Mode (ATM) Technical Oveview," 2nd Ed., Prentice Hall, 1995.
- B. Dorling, D. Freedman, C. Metz, and J. Burger, "Internetworking over ATM: An Introduction," Prentice Hall, 1996.

GitHub Link:

https://github.com/TanusriK/intelunnati_InnovateX