**Design and Implementation of Automated Teller Machine (FSM) Controller**

**Introduction:**

ATMs, as known today, are based on a computer system with microprocessor or other programmatic components, with data storage such as Hard disks or semiconductor memories, as well as special peripherals, such as monitors, cash boxes, card readers, etc. which are operated by industry-specific software. ATMs today are regularly networked with other computers and may also be connected to additional hardware such as local monitoring units such as sensors and cameras, or to server-based network-connected control systems. The designed ATM controller FSM should perform the following checks: Invalid PIN entry (3 times allowed and later it should lock the account for next 24 hours), Withdraw, Deposit, Old balance and new balance display, Mini statement for the recent transactions.

**Approach to solve the problem:**

****

**Step 1**: Design the FSM Architecture

Using the Intel Quartus Prime tool, the first step is to design the Finite State Machine (FSM) architecture for the ATM controller. This involves creating a block diagram that represents the states, transitions, and actions of the FSM. Each state and transition should be clearly defined and connected according to the specified requirements.

Outcome: A block diagram representing the FSM architecture is created in the Intel Quartus Prime tool.

**Step 2**: Define State Register and Output Signals

In this step, the state register and output signals of the FSM are defined. The state register is a memory element that stores the current state of the FSM, while the output signals represent the actions or checks performed by the FSM. Each state and transition in the block diagram should be associated with the corresponding state register value and output signal.

Outcome: The state register and output signals are defined in the Intel Quartus Prime tool.

**Step 3**: Implement the FSM in Verilog HDL

Using the VHDL (VHSIC Hardware Description Language) editor in Intel Quartus Prime, the FSM is implemented by writing the VHDL code that represents the states, transitions, and actions. The code should define the state register, output signals, and the logic for state transitions and actions based on the specified requirements.

Outcome: The FSM is implemented in VHDL code within the Intel Quartus Prime tool.

**Step 4**: Simulate and Verify the FSM

To ensure the correctness of the implemented FSM, it is necessary to simulate and verify its behaviour. This involves creating testbench files in Intel Quartus Prime to provide inputs and observe the outputs of the FSM. The testbench files generate different input scenarios, including valid and invalid PIN entries, withdrawals, deposits, and requests for balance and mini statement. The outputs are then compared against the expected behaviour to verify the correctness of the FSM.

Outcome: The FSM is simulated and verified using testbench files in Intel Quartus Prime.

**Step 5**: Generate and Program the FPGA

Once the FSM design is verified, the next step is to generate the programming file for the FPGA (Field-Programmable Gate Array) using the Intel Quartus Prime tool. This file contains the configuration data for the FPGA to implement the FSM design. The programming file is then programmed onto the FPGA device to enable its operation as an ATM controller.

Outcome: The programming file for the FPGA is generated using Intel Quartus Prime and programmed onto the FPGA device.

**Step 6**: Test the FPGA Implementation

After programming the FPGA, the final step is to test the implemented FSM on the FPGA device. This involves connecting the necessary peripherals such as keypad, display, and face recognition module to the FPGA, and performing real-time tests to verify the behaviour and functionality of the ATM controller FSM.

Outcome: The FPGA implementation of the ATM controller FSM is tested and validated for real-time operation.

**ATM Project Report:**

Conclusion of Tasks Performed:

The ATM project involved the development of an ATM controller FSM (Finite State Machine) to handle various operations such as PIN validation, withdrawals, deposits, balance display, and mini statement generation. The primary goal of the project was to ensure the proper functioning of the ATM system while adhering to specific requirements and performing necessary checks.

During the implementation phase, several tests were conducted to validate the functionality and accuracy of the ATM controller FSM. The following tests were carried out:

**1. Invalid PIN Entry Test:** The system was tested to allow a maximum of three invalid PIN entries. After three unsuccessful attempts, the account would be locked for the next 24 hours. This test was performed successfully, ensuring that the account locking feature was activated after the specified number of failed attempts.

**2. Withdrawal Test:** Withdrawal functionality was tested to ensure that it allowed users to withdraw funds from their accounts. Additionally, a specific condition was set that if the withdrawal amount exceeded 10,000 INR, a Face Recognition Test would be required. This condition was successfully implemented, and the withdrawal operation worked as expected, triggering the face recognition test when necessary.

**3. Deposit Test:** The deposit feature was tested to allow users to deposit funds into their accounts. The system accurately recorded the deposited amount and updated the account balance accordingly. This test was also successful.

**4. Balance Display Test:** The balance display functionality was thoroughly tested to ensure that it correctly presented both the old balance and the new balance after any transaction. The system accurately retrieved and displayed the account balance, reflecting the impact of various operations such as withdrawals and deposits.

**5. Mini Statement Test:** The mini statement generation feature was tested to provide users with a summary of their recent transactions. The system successfully compiled and displayed the necessary information, including transaction dates, types, and amounts.

**Conclusion:**

The use of Intel Quartus Prime tool facilitates the design, implementation, verification, and testing of the ATM controller FSM. By following the steps outlined above, a reliable and functional FPGA-based ATM controller can be developed, incorporating the specified checks and actions such as PIN entry validation, withdrawal limits, deposits, balance display, and mini statement functionality.

Overall, the ATM controller FSM successfully performed the required checks and functionalities. The tests conducted verified that the system was robust and accurate in handling PIN validation, withdrawals, deposits, balance display, and mini statement generation. The final accuracy of the ATM project was confirmed to meet the desired standards, ensuring a reliable and secure user experience.