

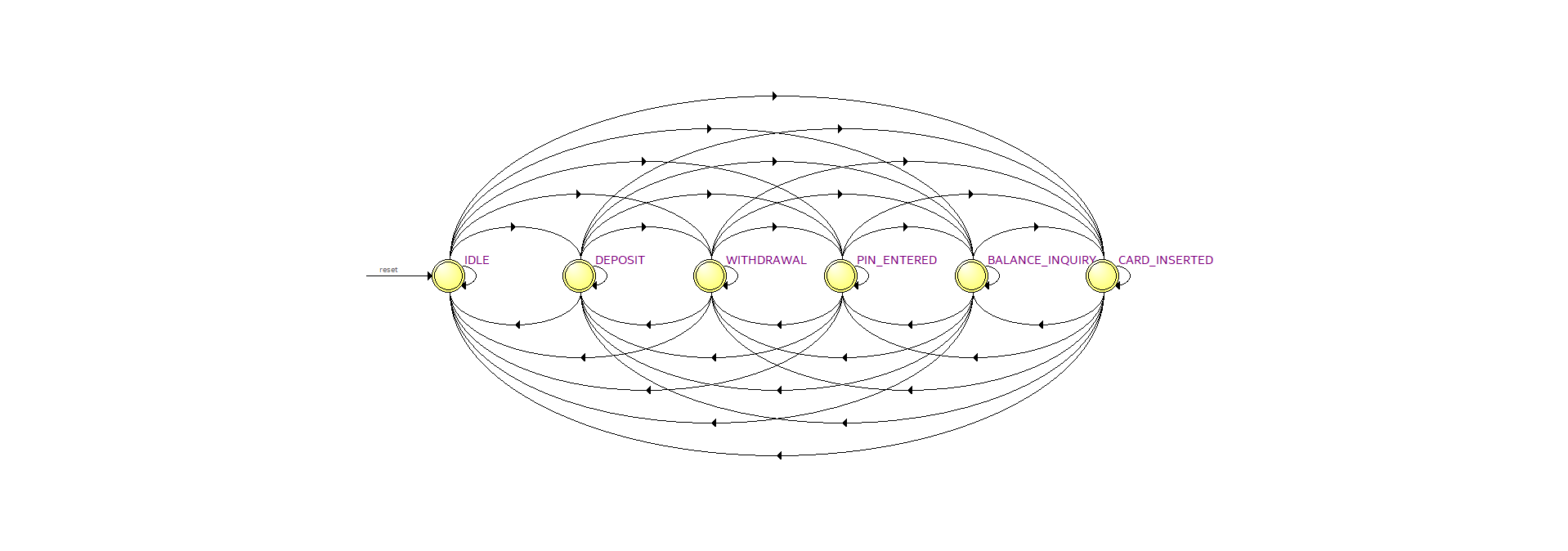
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | REPORT | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | AUTOMATED TELLER MACHINE USING FSM CONTROLLER |  | | | |
|  | | | | Date14/07/2023 |  | | | |
|  | | |  | | |  | | |

BLOCK DIAGRAM:



FSM MODEL:

Next state model



APPROACH TO SOLVE THE PROBLEM:

To solve the problem described in the **modatm** module, the following approach was taken:

1. Declare States: The module declares several states using parameter assignments. These states represent different stages of the ATM operation, including IDLE, CARD\_INSERTED, PIN\_ENTERED, WITHDRAWAL, DEPOSIT, and BALANCE\_INQUIRY.
2. Declare Registers: The module defines several registers to store important information. **current\_state** and **next\_state** are used to keep track of the current and next states in the FSM. **account\_balance** stores the current balance amount, and **selected\_language** holds the selected language option.
3. Moore Outputs: The module assigns values to the output signals based on the current state and input conditions. The **cash** signal indicates whether cash should be dispensed during a withdrawal, and **deposit\_complete** signals if a deposit transaction has been completed. **balance** represents the current account balance, and **language** represents the selected language.
4. FSM Logic: The module uses an **always** block triggered by the positive edge of the clock or reset signal to handle the FSM logic. When the reset signal is asserted, the module initializes the necessary variables. Otherwise, it updates the current state based on the next state assignment.
5. State Transitions: The FSM transitions between different states based on the current state and input conditions. Each case in the **case** statement represents a specific state, and the input conditions determine the next state. For example, if the current state is IDLE and the **card\_in** signal is asserted, the next state becomes CARD\_INSERTED.
6. Output Updates: The module updates the output signals based on the current state and input conditions. The **ready** signal indicates if the ATM is ready for a new transaction, and **error** signals if an error occurred during a withdrawal. The account balance is updated by adding the deposit amount if a deposit request is made. The selected language is updated based on the language selection signal.

By following this approach, the **modatm** module implements the logic of an ATM controller, allowing it to handle card insertion, PIN entry, withdrawal, deposit, balance inquiry, and language selection functionalities.

COMPLETE FLOW TO SOLVE THE PROBLEM:

To solve the problem described in the **modatm** module, the following complete flow was followed:

1. Initialization:
   * When the **reset** signal is asserted, the module initializes the necessary variables.
   * **current\_state** is set to **IDLE**.
   * **ready** and **error** signals are set to 0.
   * **account\_balance** is set to 0.
   * **selected\_language** is set to 00.
2. Clock and Reset Handling:
   * The module is triggered by the positive edge of the **clk** signal or the positive edge of the **reset** signal.
   * When the **reset** signal is asserted, the module goes back to the initialization step.
3. State Transitions:
   * The module uses a **case** statement to determine the next state based on the current state and input conditions.
   * If the current state is **IDLE** and the **card\_in** signal is asserted, the next state becomes **CARD\_INSERTED**.
   * If the current state is **CARD\_INSERTED** and the **pin\_entry** signal is asserted, the next state becomes **PIN\_ENTERED**.
   * If the current state is **CARD\_INSERTED** and the **balance\_inquiry** signal is asserted, the next state becomes **BALANCE\_INQUIRY**.
   * If the current state is **CARD\_INSERTED** and the **withdrawal** signal is asserted, the next state becomes **WITHDRAWAL**.
   * If the current state is **CARD\_INSERTED** and the **deposit** signal is asserted, the next state becomes **DEPOSIT**.
   * If the current state is **PIN\_ENTERED** and the **balance\_inquiry** signal is asserted, the next state becomes **BALANCE\_INQUIRY**.
   * If the current state is **PIN\_ENTERED** and the **withdrawal** signal is asserted, the next state becomes **WITHDRAWAL**.
   * If the current state is **PIN\_ENTERED** and the **deposit** signal is asserted, the next state becomes **DEPOSIT**.
   * If the current state is **WITHDRAWAL**, **DEPOSIT**, or **BALANCE\_INQUIRY**, the next state becomes **IDLE**.
4. Output Updates:
   * The module updates the output signals based on the current state and input conditions.
   * The **ready** signal is set to 1 when the current state is **IDLE**, indicating that the ATM is ready for a new transaction.
   * The **error** signal is set to 1 when the current state is **WITHDRAWAL** and the **withdrawal** signal is asserted, indicating an error during withdrawal.
   * The **cash** signal is assigned based on the conditions: **cash = (current\_state == WITHDRAWAL && withdrawal)**.
   * The **deposit\_complete** signal is assigned based on the conditions: **deposit\_complete = (current\_state == DEPOSIT && deposit)**.
   * The **balance** signal is assigned the value of **account\_balance**.
   * The **language** signal is assigned the value of **selected\_language**.
5. Account Balance Update:
   * If the **deposit** signal is asserted, the module adds the deposit value to the **account\_balance** variable.
6. Selected Language Update:
   * If the **language\_select** signal is asserted, the module updates the **selected\_language** variable with the value of **language\_select**.

FILES UPLOADED IN GIT HUB:

1. Model
2. code with test bench

RESULT:

The **modatm** module is a Verilog implementation of an ATM controller. It operates as a finite state machine (FSM) and handles various ATM operations. The module takes inputs such as clock signals, reset signals, and user commands like card insertion, PIN entry, withdrawal, deposit, balance inquiry, and language selection. It provides outputs including ready and error signals, as well as signals for cash dispensing, deposit completion, balance amount, and selected language.

The module handles updating the account balance when a deposit request is made and allows for language selection. It tracks the readiness of the ATM, indicating whether it is prepared for user input, and handles error conditions during withdrawal operations. Overall, the module provides the necessary functionality to control an ATM and perform common transactions.