**Verilog code:**

module atm(

input clk,

input rst,

input [4:0] pin,

input [7:0] amount,

input face,

output reg [7:0] old\_balance,

output reg [7:0] new\_balance,

output reg mini\_statement

);

reg [7:0] balance = 1000;

reg [3:0] pin\_count = 0;

reg locked = 0;

always @(posedge clk) begin

if (rst) begin

pin\_count <= 0;

locked <= 0;

old\_balance <= balance;

new\_balance <= balance;

mini\_statement <= 0;

end else begin

if (locked) begin

new\_balance <= balance;

mini\_statement <= 0;

end else begin

if (pin == 1234 && pin\_count < 3) begin

pin\_count <= pin\_count + 1;

new\_balance <= balance - amount;

mini\_statement <= 1;

end else if (pin != 1234 && pin\_count < 3) begin

pin\_count <= pin\_count + 1;

end else if (pin != 1234 && pin\_count == 3) begin

locked <= 1;

end

end

end

   end

endmodule

**testbench.v:**

module atm\_tb;

reg clk = 0;

reg rst = 0;

reg [4:0] pin = 1234;

reg [7:0] amount = 1000;

reg face = 1;

wire [7:0] old\_balance;

wire [7:0] new\_balance;

wire mini\_statement;

atm atm(

.clk(clk),

.rst(rst),

.pin(pin),

.amount(amount),

.face(face),

.old\_balance(old\_balance),

.new\_balance(new\_balance),

.mini\_statement(mini\_statement)

);

initial begin

#10 rst = 1;

#10 rst = 0;

#1000 $finish;

end

always #1 clk = ~clk;

endmodule