Code:

`define true 1'b1

`define false 1'b0

`define FIND 1'b0

`define AUTHENTICATE 1'b1

`define WAITING 3'b000

`define MENU 3'b010

`define BALANCE 3'b011

`define WITHDRAW 3'b100

`define WITHDRAW\_SHOW\_BALANCE 3'b101

`define TRANSACTION 3'b110

module authentication(

input [11:0] accNumber,

input [3:0] pin,

input action,

input deAuth,

output reg wasSuccessful,

output reg [3:0] accIndex

);

reg [11:0] acc\_database [0:9];

reg [3:0] pin\_database [0:9];

//initializing the database with arbitrary accounts

initial begin

acc\_database[0] = 12'd2749; pin\_database[0] = 4'b0000;

acc\_database[1] = 12'd2175; pin\_database[1] = 4'b0001;

acc\_database[2] = 12'd2429; pin\_database[2] = 4'b0010;

acc\_database[3] = 12'd2125; pin\_database[3] = 4'b0011;

acc\_database[4] = 12'd2178; pin\_database[4] = 4'b0100;

acc\_database[5] = 12'd2647; pin\_database[5] = 4'b0101;

acc\_database[6] = 12'd2816; pin\_database[6] = 4'b0110;

acc\_database[7] = 12'd2910; pin\_database[7] = 4'b0111;

acc\_database[8] = 12'd2299; pin\_database[8] = 4'b1000;

acc\_database[9] = 12'd2689; pin\_database[9] = 4'b1001;

end

always @ (deAuth) begin

if(deAuth == `true)

wasSuccessful = 1'bx;

end

//looping through the database, trying to find a match for the given accNumber and pin

// if action is set to find then it'll simply ry to find a match for the given accNumber and returns its index

integer i;

always @(accNumber or pin) begin

wasSuccessful = `false;

accIndex = 0;

//loop through the data base

for(i = 0; i < 10; i = i+1) begin

//found a match for accNumber

if(accNumber == acc\_database[i]) begin

if(action == `FIND) begin

wasSuccessful = `true;

accIndex = i;

end

if(action == `AUTHENTICATE) begin

if(pin == pin\_database[i]) begin

wasSuccessful = `true;

accIndex = i;

end

end

end

end

end

endmodule

//

module ATM(

input clk,

input exit,

input [11:0] accNumber,

input [3:0] pin,

input [11:0] destinationAcc,

input [2:0]menuOption,

input [10:0] amount,

output reg error,

output reg [10:0] balance

);

//initializing the balance database with an arbitrary amount of money

reg [15:0] balance\_database [0:9];

initial begin

$display("Welcome to the ATM");

balance\_database[0] = 16'd500;

balance\_database[1] = 16'd500;

balance\_database[2] = 16'd500;

balance\_database[3] = 16'd500;

balance\_database[4] = 16'd500;

balance\_database[5] = 16'd500;

balance\_database[6] = 16'd500;

balance\_database[7] = 16'd500;

balance\_database[8] = 16'd500;

balance\_database[9] = 16'd500;

end

reg [2:0] currState = `WAITING;

wire [3:0] accIndex;

wire [3:0] destinationAccIndex;

wire isAuthenticated;

wire wasFound;

reg deAuth = `false;

authentication authAccNumberModule(accNumber, pin, `AUTHENTICATE, deAuth, isAuthenticated, accIndex);

authentication findAccNumberModule(destinationAcc, 0, `FIND, deAuth, wasFound, destinationAccIndex);

//main block of module with asynchronous exit

always @(posedge clk or isAuthenticated or menuOption or exit) begin

//restart the error

error = `false;

if(exit == `true) begin

//transition to the waiting state

currState = `WAITING;

//deathenticate the current user

deAuth = `true;

#20;

end

if(currState == `MENU) begin

//set the selected option as the current state

if((menuOption >= 0) & (menuOption <= 7))begin

currState = menuOption;

end else

currState = menuOption;

end

//switch case for the menu options

//the rest is pretty straight forward

case (currState)

`WAITING: begin

if (isAuthenticated == `true) begin

currState = `MENU;

$display("Logged In.");

end

else if(isAuthenticated == `false) begin

$display("Account number or password was incorrect");

currState = `WAITING;

end

end

`BALANCE: begin

balance = balance\_database[accIndex];

$display("Account %d has balance %d", accNumber, balance\_database[accIndex]);

currState = `MENU;

end

`WITHDRAW: begin

if (amount <= balance\_database[accIndex]) begin

balance\_database[accIndex] = balance\_database[accIndex] - amount;

balance = balance\_database[accIndex];

currState = `MENU;

error = `false;

end

else begin

currState = `MENU;

error = `true;

end

end

`WITHDRAW\_SHOW\_BALANCE: begin

if (amount <= balance\_database[accIndex]) begin

balance\_database[accIndex] = balance\_database[accIndex] - amount;

balance = balance\_database[accIndex];

currState = `MENU;

error = `false;

$display("Account %d has balance %d after withdrawing %d", accNumber, balance\_database[accIndex], amount);

end

else begin

currState = `MENU;

error = `true;

end

end

`TRANSACTION: begin

if ((amount <= balance\_database[accIndex]) & (wasFound == `true) & (balance\_database[accIndex] + amount < 2048)) begin

currState = `MENU;

error = `false;

balance\_database[destinationAccIndex] = balance\_database[destinationAccIndex] + amount;

balance\_database[accIndex] = balance\_database[accIndex] - amount;

$display("Destination account %d after transaction has a total balance of %d", destinationAcc, balance\_database[destinationAccIndex]);

end

else begin

currState = `MENU;

error = `true;

end

end

endcase

end

endmodule