VERILOG CODE

module ElectricityBillPaymentFSM(

input wire clk,

input wire reset,

input wire amount\_entered,

input wire payment\_processed,

output reg payment\_complete

);

// Define the FSM states

parameter IDLE = 2'b00;

parameter ENTER\_AMOUNT = 2'b01;

parameter PROCESS\_PAYMENT = 2'b10;

reg [1:0] current\_state, next\_state;

always @(posedge clk or posedge reset) begin

if (reset) begin

current\_state <= IDLE;

end else begin

current\_state <= next\_state;

end

end

always @(current\_state or amount\_entered or payment\_processed) begin

next\_state = current\_state;

case (current\_state)

IDLE:

begin

if (amount\_entered) begin

next\_state = ENTER\_AMOUNT;

end

end

ENTER\_AMOUNT:

begin

if (payment\_processed) begin

next\_state = PROCESS\_PAYMENT;

end

end

PROCESS\_PAYMENT:

begin

if (!payment\_processed) begin

next\_state = ENTER\_AMOUNT;

end

end

endcase

end

always @(current\_state) begin

case (current\_state)

IDLE:

begin

// Perform any IDLE state actions here

end

ENTER\_AMOUNT:

begin

// Perform any ENTER\_AMOUNT state actions here

end

PROCESS\_PAYMENT:

begin

// Perform any PROCESS\_PAYMENT state actions here

payment\_complete = 1; // Set the payment\_complete signal high

end

endcase

end

endmodule