**DEVELOP THE TESTBENCH AND VERIFY FOR ITS FUNCTIONALITY**

```verilog

module ElectricityBillPaymentFSM\_Testbench;

reg clk;

reg reset;

reg amount\_entered;

reg payment\_processed;

wire payment\_complete;

// Instantiate the DUT (Device Under Test)

ElectricityBillPaymentFSM dut (

.clk(clk),

.reset(reset),

.amount\_entered(amount\_entered),

.payment\_processed(payment\_processed),

.payment\_complete(payment\_complete)

);

// Clock generator

always #5 clk = ~clk;

// Stimulus generation

initial begin

clk = 0;

reset = 1;

amount\_entered = 0;

payment\_processed = 0;

// Reset FSM

#10 reset = 0;

// Transition from IDLE to ENTER\_AMOUNT

#20 amount\_entered = 1;

// Transition from ENTER\_AMOUNT to PROCESS\_PAYMENT

#20 payment\_processed = 1;

// Transition from PROCESS\_PAYMENT back to ENTER\_AMOUNT

#20 payment\_processed = 0;

// End simulation

#10 $finish;

end

// Assertion

always @(posedge clk) begin

if (payment\_complete) begin

$display("Payment successfully processed.");

end

end

endmodule

```

In this testbench, the `clk` signal is generated using a clock generator, which alternates between 0 and 1 with a delay of 5 units of time. The `reset` signal is initially set high and then brought low after 10 units of time to reset the FSM.

After the reset, the testbench provides stimulus to the FSM. It sets `amount\_entered` high after 20 units of time to transition from the IDLE state to the ENTER\_AMOUNT state. Then, it sets `payment\_processed` high after another 20 units of time to transition from the ENTER\_AMOUNT state to the PROCESS\_PAYMENT state. Finally, it sets `payment\_processed` low after another 20 units of time to transition back from the PROCESS\_PAYMENT state to the ENTER\_AMOUNT state.

The `payment\_complete` signal is monitored in the assertion block, and when it becomes high, the testbench displays a message indicating that the payment has been successfully processed.

To run this testbench, you can use a Verilog simulator such as ModelSim or Icarus Verilog. The simulation will show the desired message "Payment successfully processed." if the FSM transitions correctly and the `payment\_complete` signal is set high as expected.