module LogicCircuit (

input wire clk, // Global clock

input wire rst, // Reset signal

input wire [3:0] in1, // Input 1

input wire [3:0] in2, // Input 2

output wire [3:0] out // Output

);

wire [3:0] and\_out, or\_out, xor\_out, mux\_out, ff\_out;

// AND Gate

assign and\_out = in1 & in2;

// OR Gate

assign or\_out = in1 | in2;

// XOR Gate

assign xor\_out = in1 ^ in2;

// 2-to-1 MUX (selecting between AND and OR output)

assign mux\_out = (in1[0]) ? and\_out : or\_out;

// D Flip-Flop (Sequential logic)

reg [3:0] dff\_reg;

always @(posedge clk or posedge rst) begin

if (rst)

dff\_reg <= 4'b0000;

else

dff\_reg <= mux\_out;

end

assign ff\_out = dff\_reg;

// Final Output (can be changed based on logic depth requirement)

assign out = ff\_out;

endmodule