MOD 10 COUNTER

module mod\_10\_counter(

input wire clk,

input wire reset,

output reg [3:0] count

);

reg toggle;

always @(posedge clk or posedge reset) begin

if (reset)

toggle <= 1'b0; // Reset to 0

else

toggle <= ~toggle;

end

always @(posedge clk or posedge reset) begin

if (reset)

count <= 4'b0000; // Reset to 0

else if (toggle)

count <= count + 1;

else if (count == 9)

count <= 0;

end

endmodule

MOD 10 COUNTER TESTBENCH

module MOD10\_TB\_v;

reg clk;

reg reset;

wire [3:0] count;

// Instantiate the Unit Under Test (UUT)

mod\_10\_counter uut (

.clk(clk),

.reset(reset),

.count(count)

);

initial begin

clk = 0;

reset = 1;

#100 reset = 0;

#500 reset = 1;

end

always #10 clk = ~clk;

endmodule

OUTPUT

