**Code:**

* **Design Module**

// Code your design here

module Logic\_operation( input in1,

input in2,

input reset,

input [3:0] op\_cntrl,

output reg out);

always @(posedge op\_cntrl , negedge reset)

begin

if (reset)

out <= 0;

else

begin

if (op\_cntrl == 3'b000) // not gate

out = ~in1;

else if( op\_cntrl == 3'b001) //or gate

out = (in1 || in2);

else if( op\_cntrl == 3'b010) // and gate

out = (in1 & in2);

else if( op\_cntrl == 3'b011) // nand gate

out = ~(in1 & in2);

else if( op\_cntrl == 3'b100) // nor gate

out = ~(in1 || in2);

else if( op\_cntrl == 3'b101) // xor gate

out = (in1 ^ in2);

else

out = 0;

end

end

endmodule

* **TestBench**

// Code your testbench here

// or browse Examples

module tb();

reg in1,in2,reset;

reg [3:0] op\_ctrl;

wire out;

Logic\_operation dut(in1,in2,reset,op\_ctrl,out);

initial

begin

in1 = $random;

in2 = $random;

reset <= 1;

#5

op\_ctrl = 3'b000;

#10

op\_ctrl = 3'b001;

reset <= 0;

#10

op\_ctrl = 3'b010;

reset <= 1;

#10

op\_ctrl = 3'b001;

#10

op\_ctrl = 3'b010;

#10

op\_ctrl = 3'b001;

#10

op\_ctrl = 3'b110;

#10;

end

// FOR WAVEFORM ON EDA PLAYGROUND ONLY

// initial

// begin

// $dumpfile("dump.vcd");

// $dumpvars();

// #100;

// $finish;

// end

endmodule

**Output:**

Graphical user interface

Description automatically generated