AIM: Implement using Gate Level Modelling.

1. F1= Σm(1,2,4,6,7)
2. F2= πm(0,2,3,5,7)

Tool Used: Xilinx

Verilog Code:

module Assignment1(a, b, c, y);

input a;

input b;

input c;

output y;

wire p1;

wire p2;

wire p3;

wire p4;

and a4(p1,a,~c);

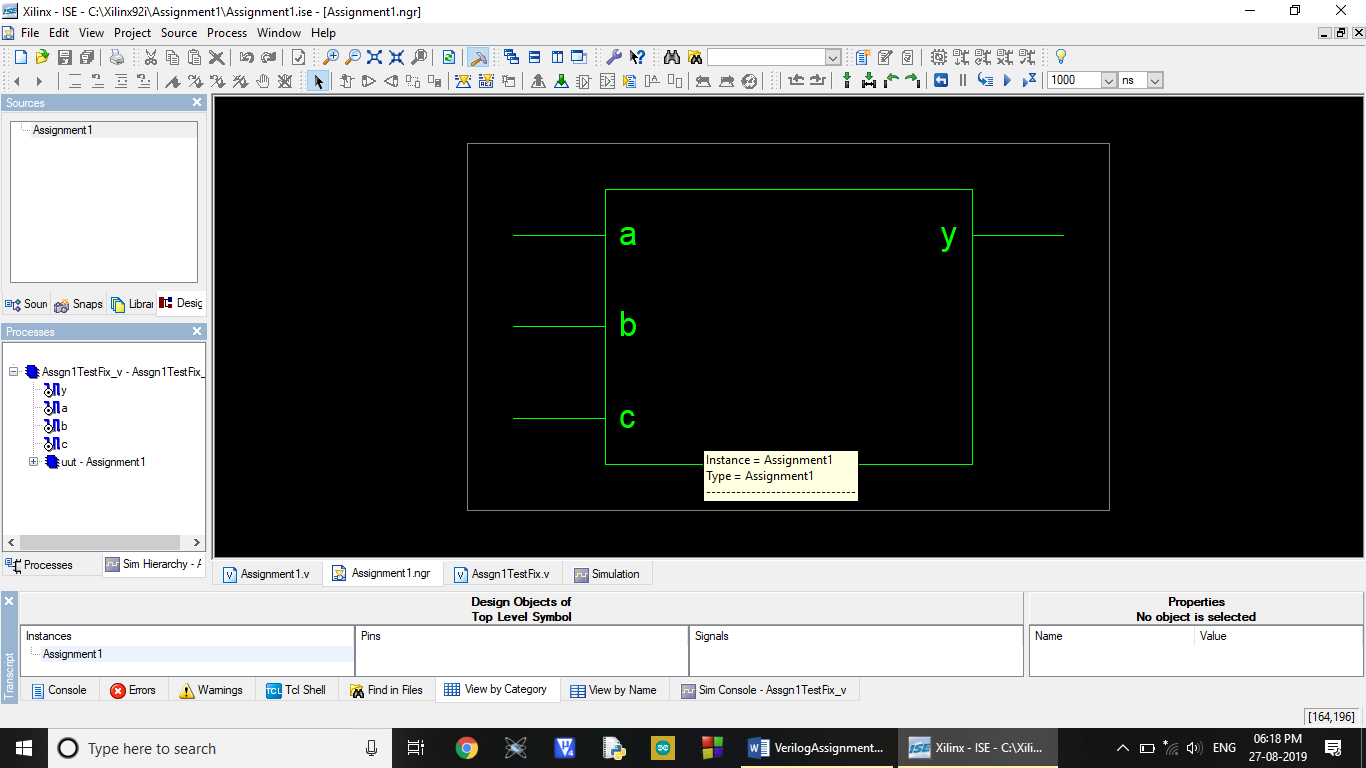
and a5(p2,~a,~b,c);

and a6(p3,a,b);

and a7(p4,b,~c);

or a8(y,p1,p2,p3,p4);

endmodule



Text Fixture:

module Assgn1TestFix\_v;

// Inputs

reg a;

reg b;

reg c;

// Outputs

wire y;

// Instantiate the Unit Under Test (UUT)

Assignment1 uut (

.a(a),

.b(b),

.c(c),

.y(y)

);

initial begin

// Initialize Inputs

a = 0;b = 0;c = 0;#100;

a = 0;b = 0;c = 1;#100;

a = 0;b = 1;c = 0;#100;

a = 0;b = 1;c = 1;#100;

a = 1;b = 0;c = 0;#100;

a = 1;b = 0;c = 1;#100;

a = 1;b = 1;c = 0;#100;

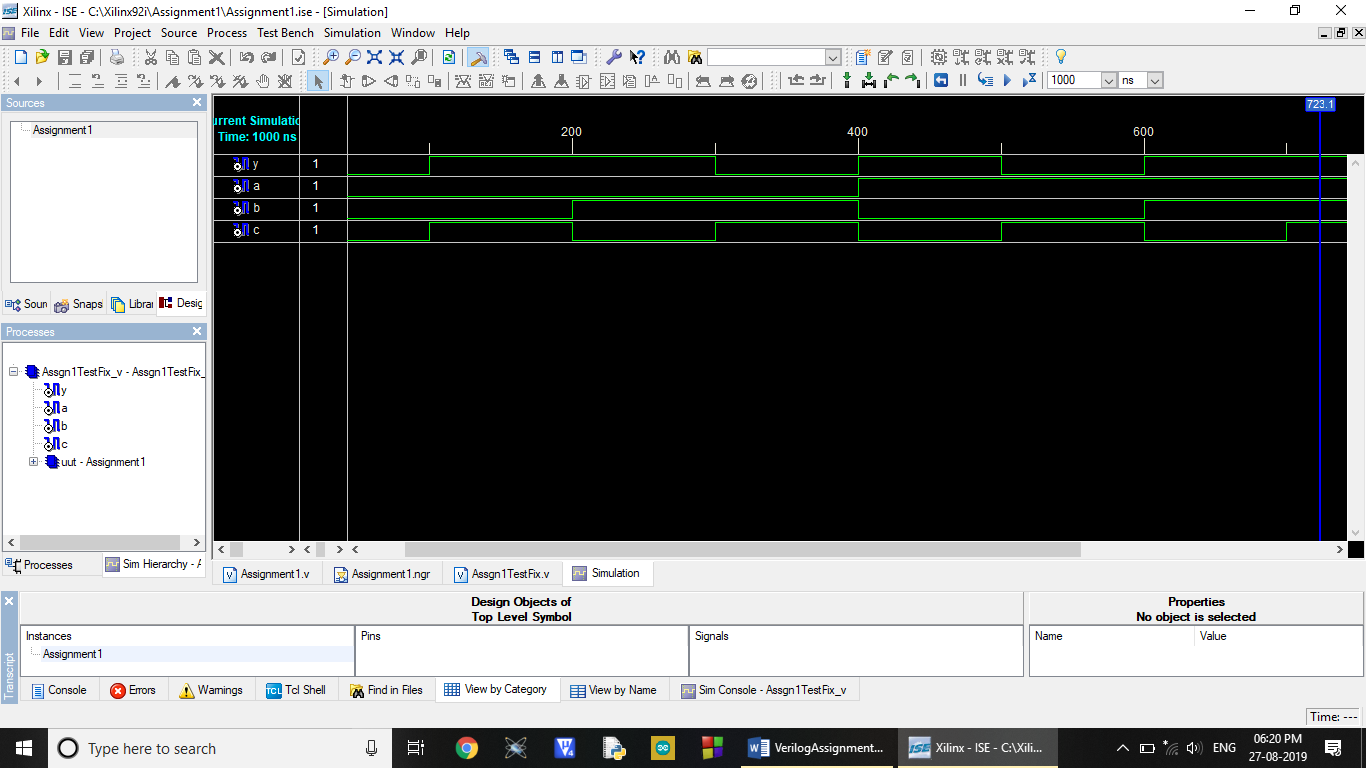
a = 1;b = 1;c = 1;#100;

// Add stimulus here

end

endmodule

Behavioral Simution/Output Waveform:



POS:

module Assignment2(a, b, c, y);

input a;

input b;

input c;

output y;

wire p1;

wire p2;

wire p3;

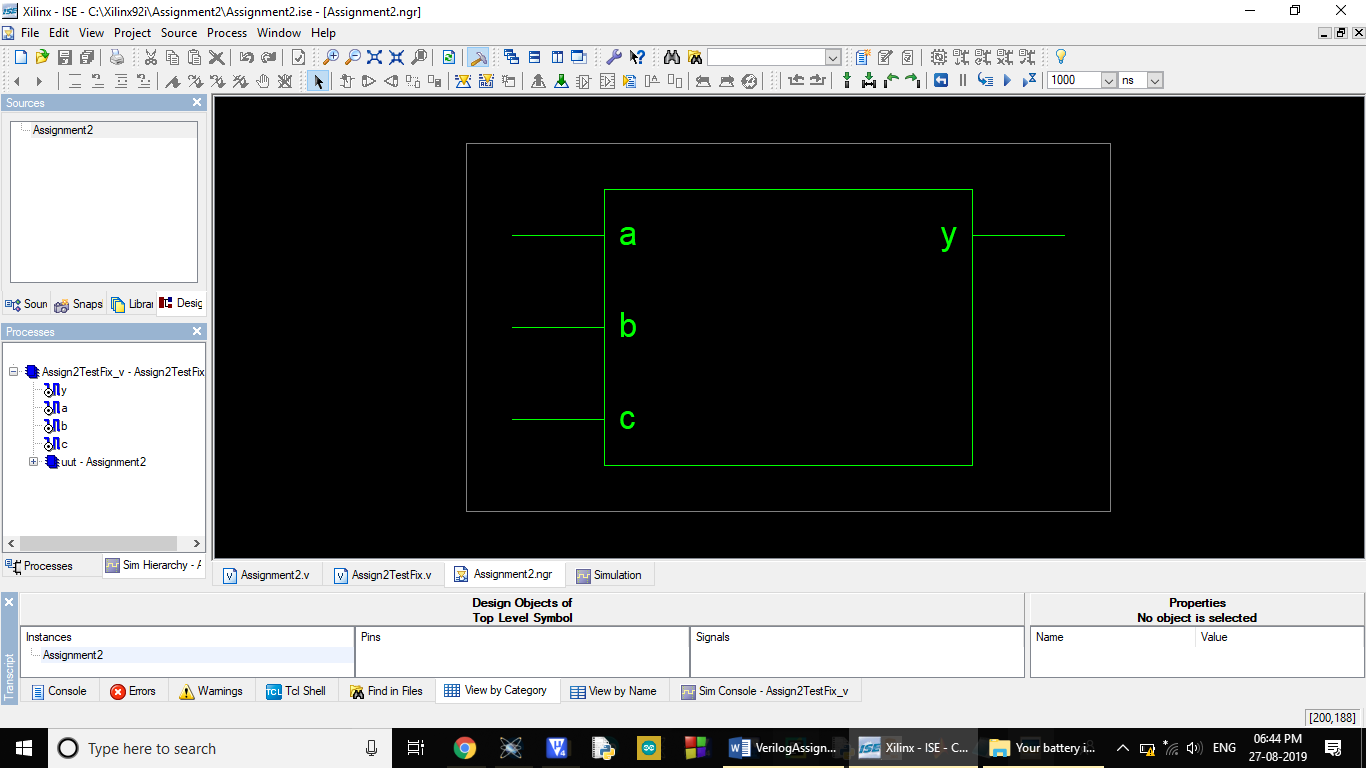
or a4(p1,~a,~c);

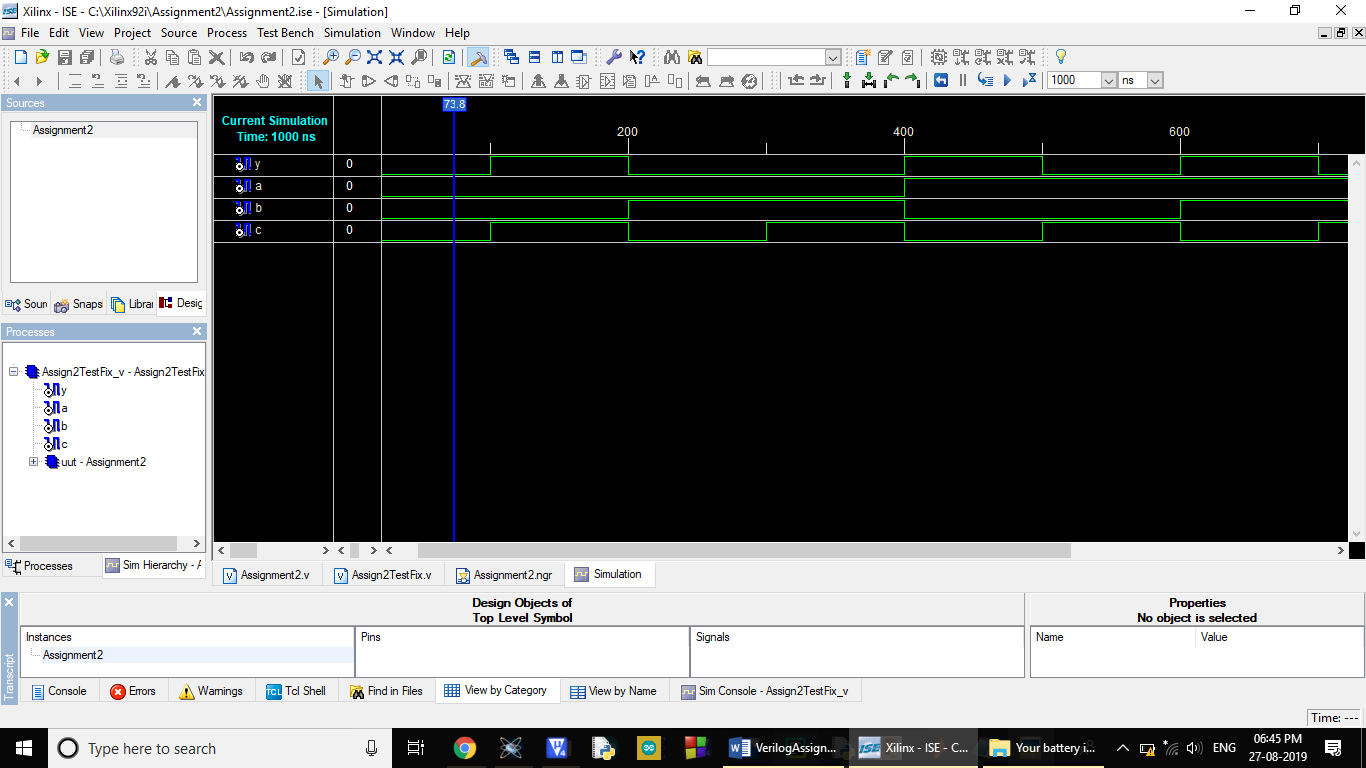
or a5(p2,a,~b);

or a6(p3,a,c);

and a8(y,p1,p2,p3);

endmodule





Text Fixture:

module Assign2TestFix\_v;

// Inputs

reg a;

reg b;

reg c;

// Outputs

wire y;

// Instantiate the Unit Under Test (UUT)

Assignment2 uut (

.a(a),

.b(b),

.c(c),

.y(y)

);

initial begin

// Initialize Inputs

a = 0;b = 0;c = 0;#100;

a = 0;b = 0;c = 1;#100;

a = 0;b = 1;c = 0;#100;

a = 0;b = 1;c = 1;#100;

a = 1;b = 0;c = 0;#100;

a = 1;b = 0;c = 1;#100;

a = 1;b = 1;c = 0;#100;

a = 1;b = 1;c = 1;#100;

// Add stimulus here

end

endmodule