Project 1: Basic Circuits written in Verilog HDL implemeted on the FPGA

1. Design

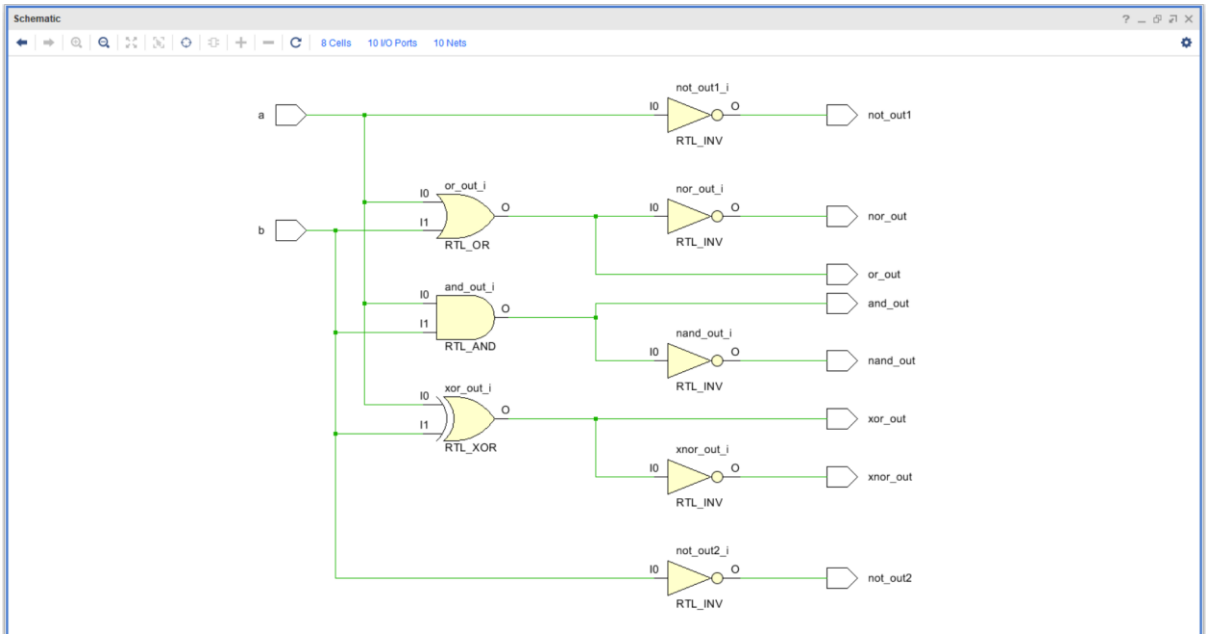
```
module all_gate(
    input a,
    input b,
    output not_out1,
    output not_out2,
    output or_out,
    output and_out,
    output xor_out,
    output nor_out,
    output nand_out,
    output xnor_out
    );
    //GATE LEVEL modeling
    not(not_out1, a);
    not(not_out2, b);
    or(or_out, a, b);
    and(and_out, a, b);
    xor(xor_out, a ,b);
    nor(nor_out, a, b);
    nand(nand_out, a, b);
    xnor(xnor_out, a, b);
endmodule
```

2. Test bench:-

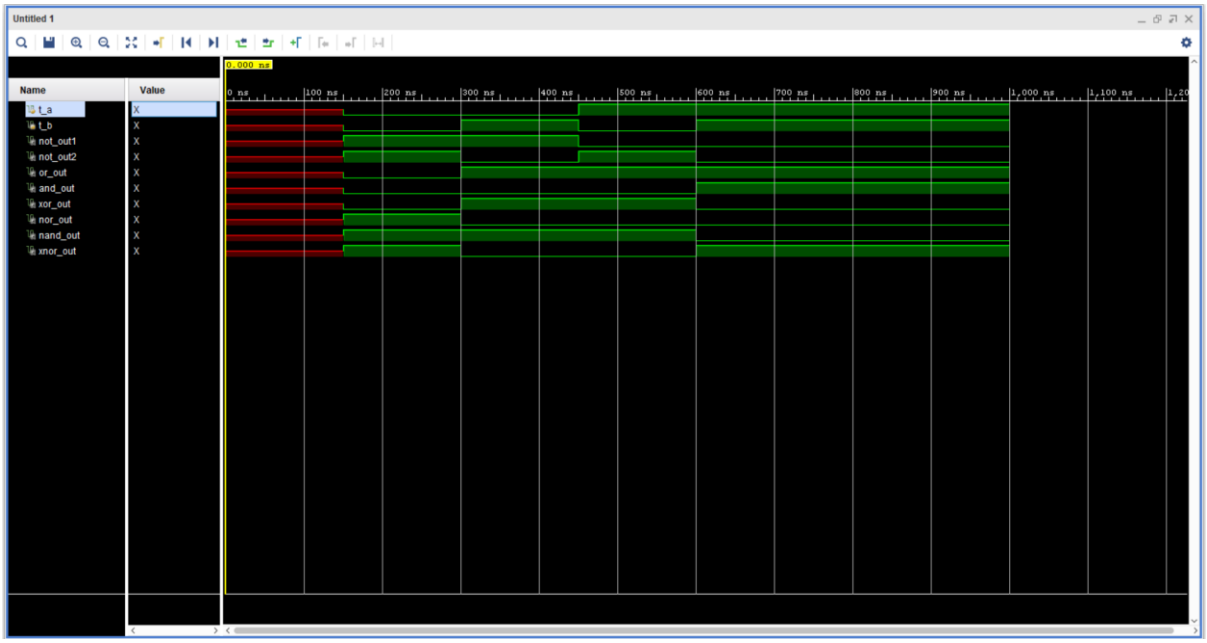
```
module allgate_tb;
reg t_a, t_b;
wire not_out1,not_out2,or_out,and_out,xor_out,nor_out,nand_out,xnor_out;
all_gate  my_gate( .a(t_a), .b(t_b), .not_out1(not_out1),.not_out2(not_out2),.or_out(or_out),
.and_out(and_out),.xor_out(xor_out),.nor_out(nor_out),.nand_out(nand_out),.xnor_out(xnor_out) );
initial
```

```
begin
$monitor(t_a, t_b, not_out1, not_out2, or_out, and_out, xor_out, nor_out, nand_out, xnor_out);
#150
t_a = 1'b0;
t_b = 1'b0;
#150
t_a = 1'b0;
t_b = 1'b1;
#150
t_a = 1'b1;
t_b = 1'b0;
#150
t_a = 1'b1;
t_b = 1'b1;
end
endmodule
```

3. Output:-



4. Waveforms


Untitled 1
Name
Value
t_a
t_b
not_out1
not_out2
or_out
and_out
xor_out
nor_out
nand_out
xnor_out
X
0.000 ns
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
1,000 ns
1,100 ns