Experiment 1

Xilinix FPGA Tools

Digital Systems

EEE3342C-20FALL 0011

Spenser Tacinelli
PID: sp498810

Equipment:

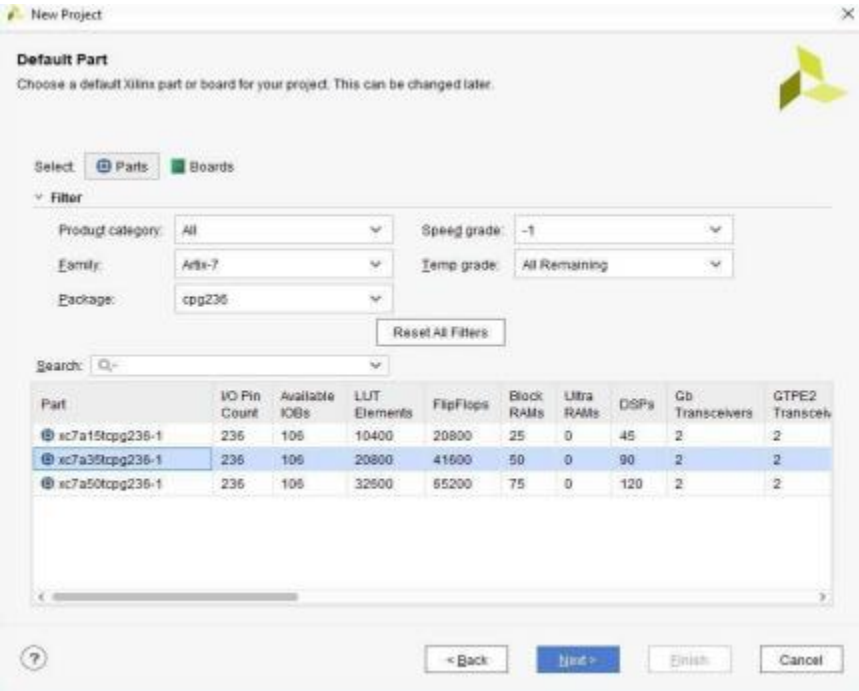
The Xilinx's FPGA VIVADO HLx Editions design tools are available in the laboratory. These tools can also be downloaded from Xilinx's web site at www.xilinx.com. The WebPack version of this tool that we use for the laboratory experiments are located under the support download section at the related website (https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloa dNav/vivado-design-tools/2017-4.html). Please take note that the file download size exceeds 1 GB and also during the installation process updates may have to be installed. It can take you up to a few hours to download and install the software on your computer. The user does not need to have the BASYS development board interface to the computer to design and simulate an FPGA.

Objective:

The objective of experiment 1 is to give an introduction to the Verilog software and to implement the most basic logic gates (and, or, xor, nand, iverter, two and five output) and test them in a simulation.

Design Steps:

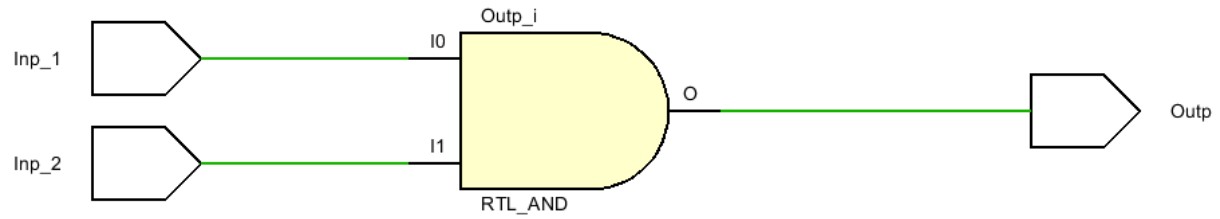
I began by starting a new project, names Test_1 which uses a cpg236, which is in the Arix-7 family and has a speed grade of -1.



I then defined the inputs and outputs of my part. I then programmed the logic of the part, connecting the inputs to the outputs. I then “built” the project.

```
// Engineer:
//
// Create Date: 09/14/2020 01:32:07 PM
// Design Name:
// Module Name: part_1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module part_1(
    input Inp_1,
    input Inp_2,
    output Outp
    );

assign Outp = Inp_1 & Inp_2;
endmodule
```

I then produced the schematic for the part.

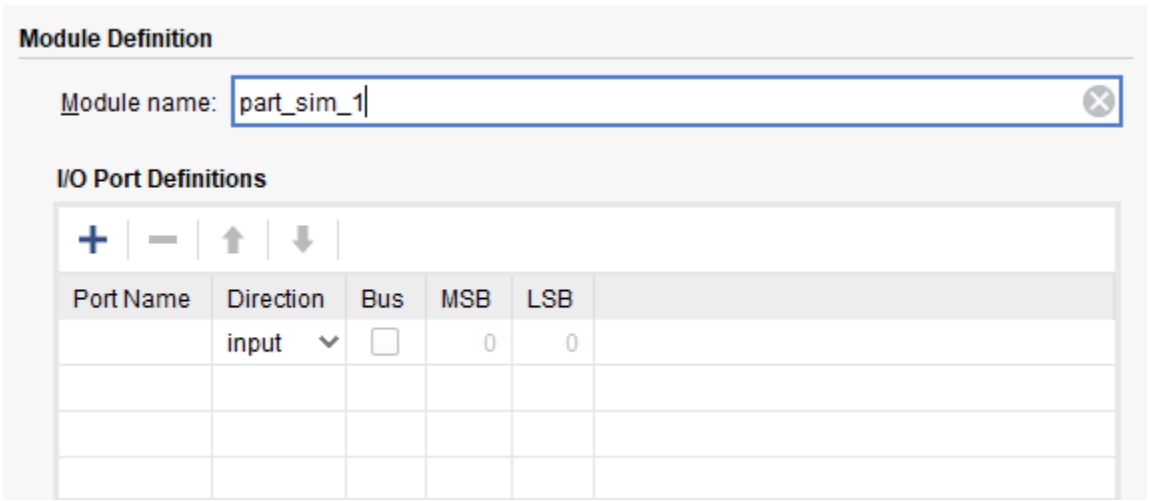


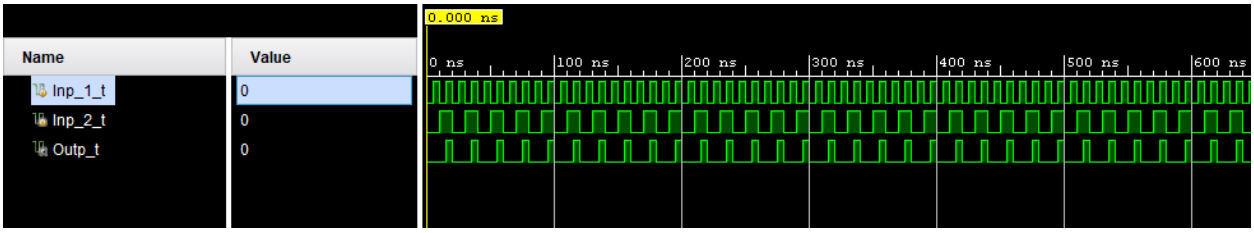
Then I registered the package pins within the package view.

Name	Direction	Neg Diff Pair	Package Pin
All ports (3)			
Scalar ports (3)			
Inp_1	IN		V17
Inp_2	IN		V16
Outp	OUT		U16

I then created a simulation file for the newly designed part. Then programmed the logic for the new simulation, allowing for every possible input. Then a behavioral simulation was run to test to logical behavior of my code.

```
22
23  module Part_1_Sim(
24      );
25      reg Inp_1_t;
26      reg Inp_2_t;
27      wire Outp_t;
28
29      Part_1 UUT(
30      .Inp_1(Inp_1_t),
31      .Inp_2(Inp_2_t),
32      .Outp(Outp_t)
33      );
34
35      initial
36      begin
37          Inp_1_t=1'b0;
38          Inp_2_t=1'b0;
39      end
40
41      always #5 Inp_1_t=~Inp_1_t;
42      always #10 Inp_2_t=~Inp_2_t;
43  endmodule
44
45
```

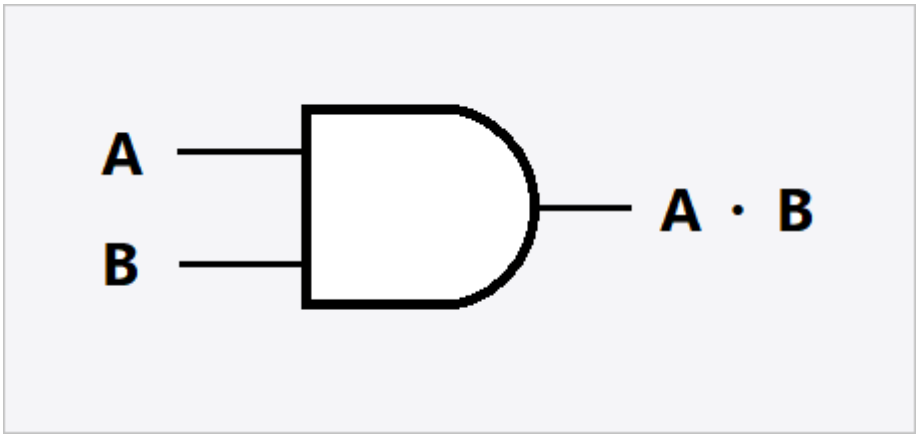




AND Gate Schematic

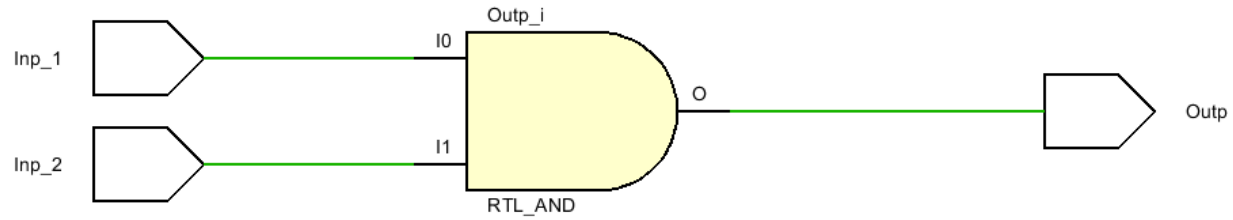
Logic Diagram:

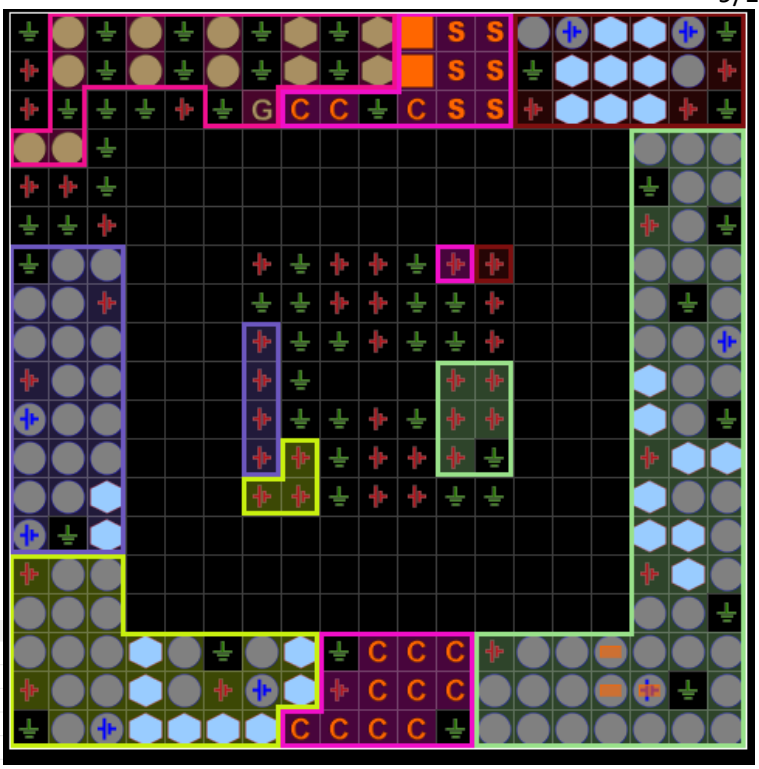
The resulting schematic diagram for an AND gate.



Design Specification Plan:

This is the design specification for an AND gate. It has two inputs (Inp_1 and Inp_2) and one output (Outp).

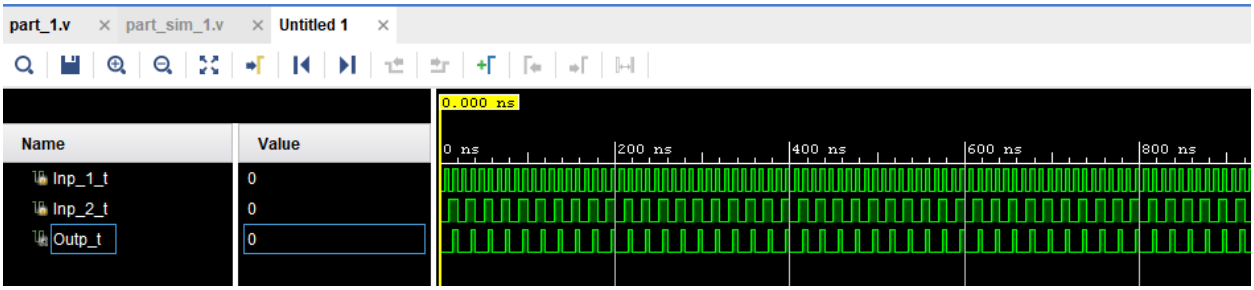




Name	Direction	Neg Diff Pair	Package Pin
All ports (3)			
Scalar ports (3)			
Inp_1	IN		V17
Inp_2	IN		V16
Outp	OUT		U16

Results Statement:

Here is the simulation results for the AND gate.

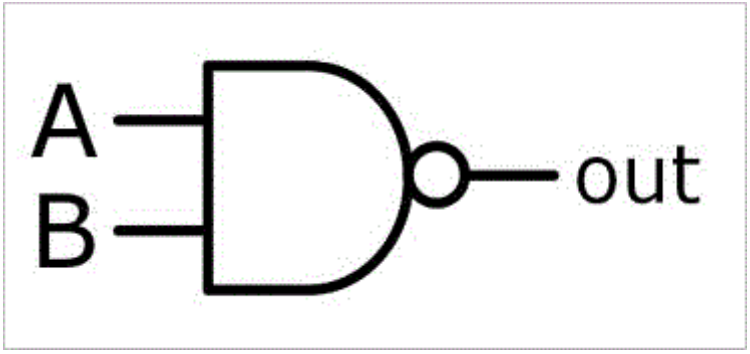


SW0	SW1	LED0
On	On	On
Off	On	Off
On	Off	Off
Off	Off	off

NAND Gate Schematic

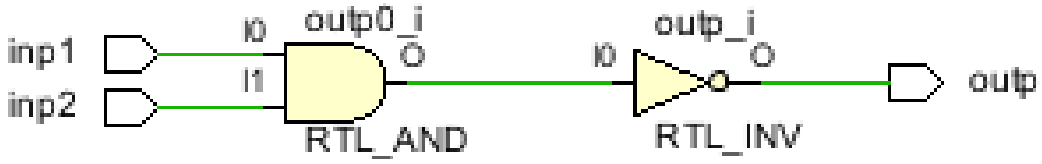
Logic Diagram:

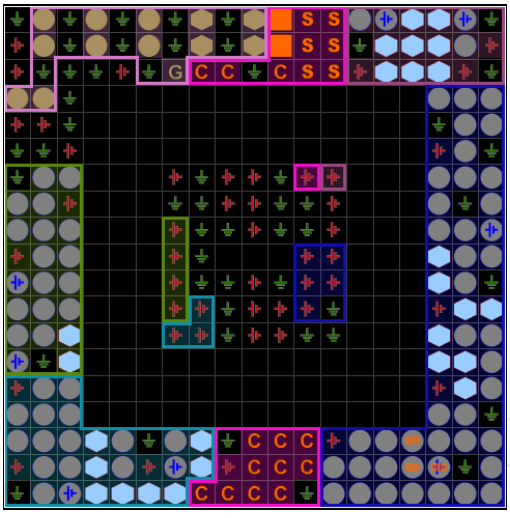
The resulting schematic diagram for an NAND gate.



Design Specification Plan:

This is the design specification for an NAND gate. It has two inputs (inp1 and inp2) and one output (outp).

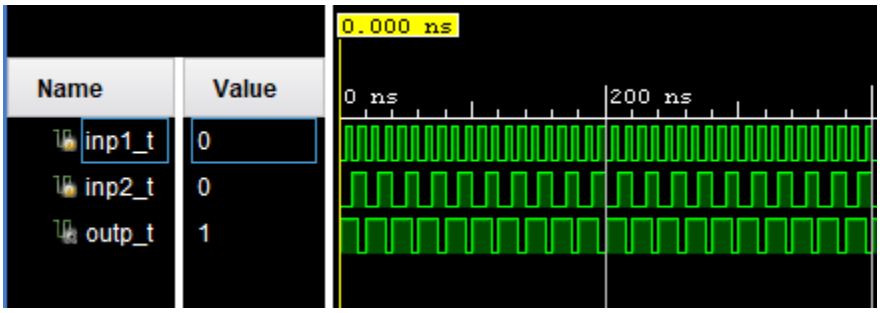




inp1	IN		V17
inp2	IN		V16
outp	OUT		U16

Results Statement:

Here is the simulation results for the NAND gate.



SW0	SW1	LED0
On	On	Off
Off	On	On
On	Off	On
Off	Off	On

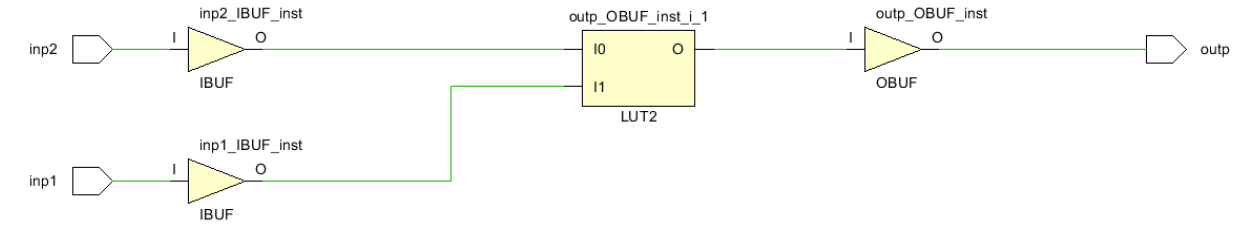
OR Gate Schematic

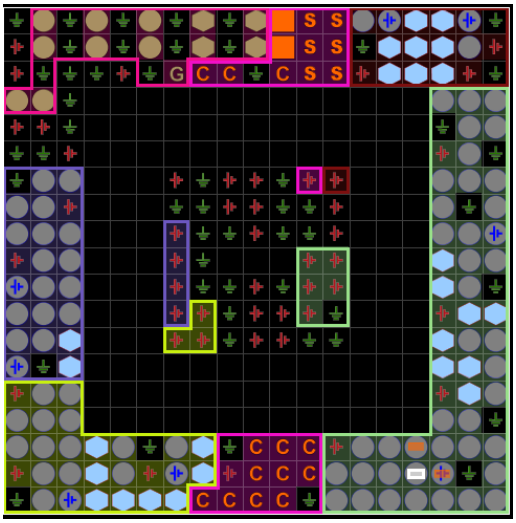
Logic Diagram:

The resulting schematic diagram for an OR gate.

Design Specification Plan:

This is the design specification for an OR gate. It has two inputs (Inp1 and Inp2) and one output (outp).

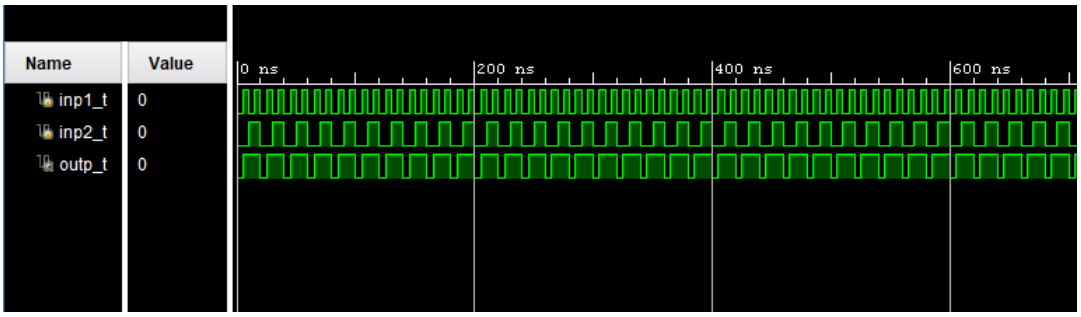




inp1	IN		V16	14	LVCMOS33*
inp2	IN		V17	14	LVCMOS33*
outp	OUT		U16	14	LVCMOS33*

Results Statement:

Here is the simulation results for the OR gate.

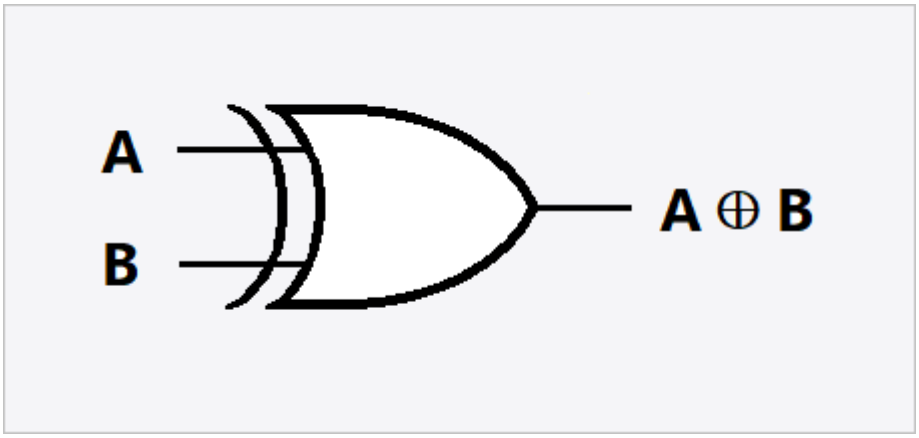


SW0	SW1	LED0
On	On	On
Off	On	On
On	Off	On
Off	Off	Off

XOR Gate Schematic

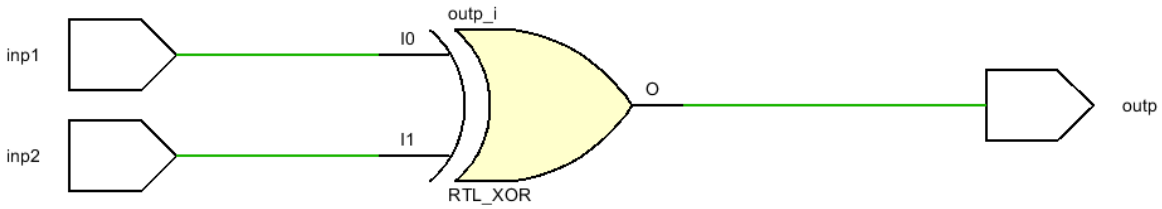
Logic Diagram:

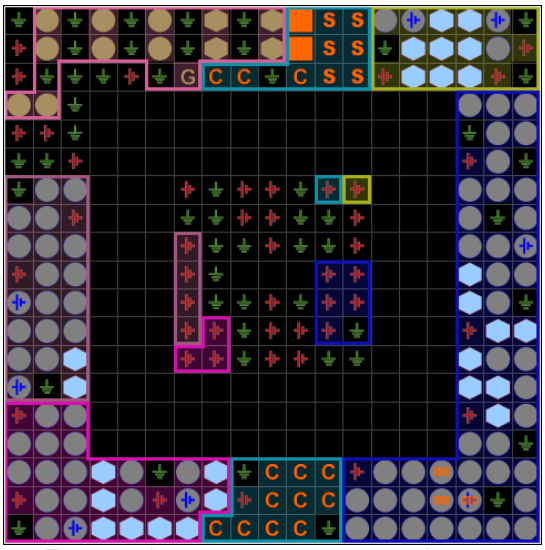
The resulting schematic diagram for an XOR gate.



Design Specification Plan:

This is the design specification for an XOR gate. It has two inputs (Inp1 and Inp2) and one output (outp).

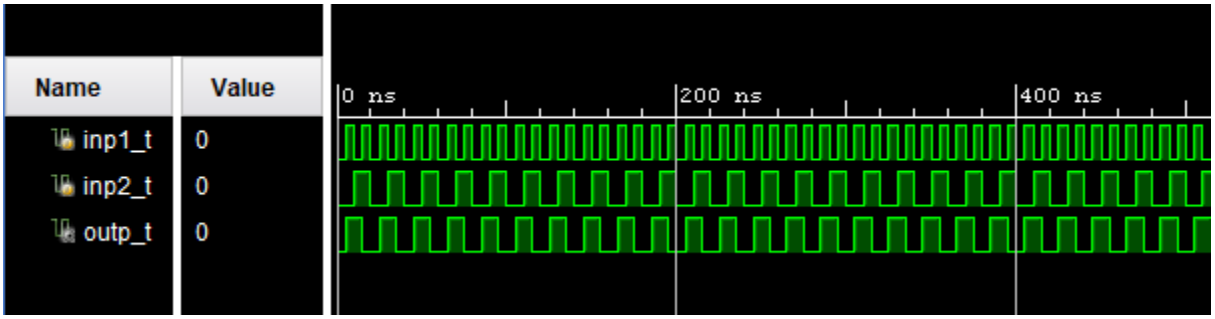




inp1	IN		V17	✓	14	LVCMOS33*
inp2	IN		V16	✓	14	LVCMOS33*
outp	OUT		U16	✓	14	LVCMOS33*

Results Statement:

Here is the simulation results for the XOR gate.

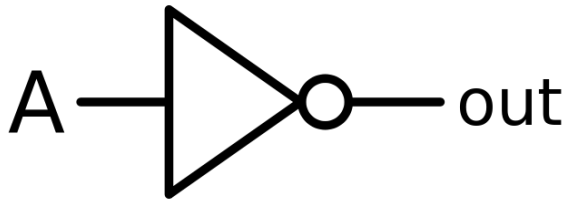


SW0	SW1	LED0
On	On	Off
Off	On	On
On	Off	On
Off	Off	Off

Inverter Gate Schematic

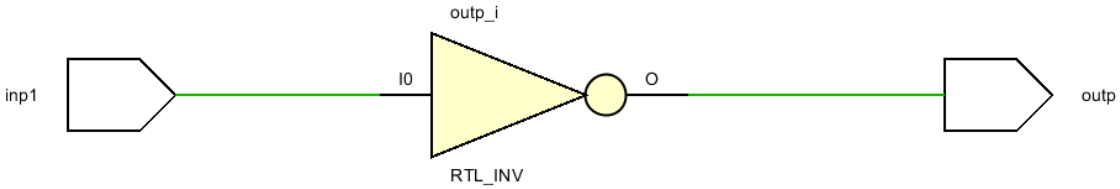
Logic Diagram:

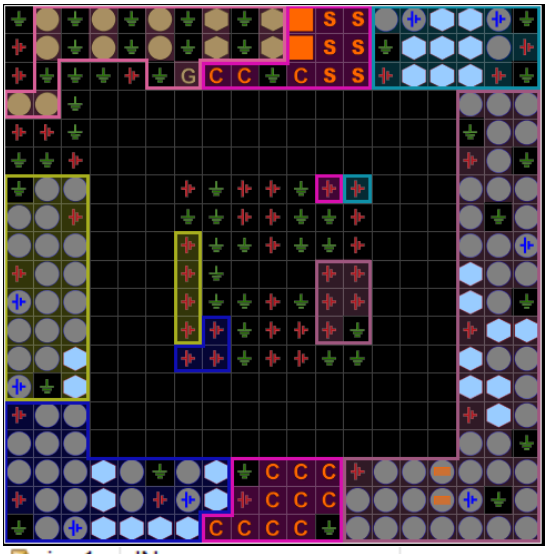
The resulting schematic diagram for an INVERTER gate.



Design Specification Plan:

This is the design specification for an INVERTER gate. It has one input (Inp1) and one output (outp).

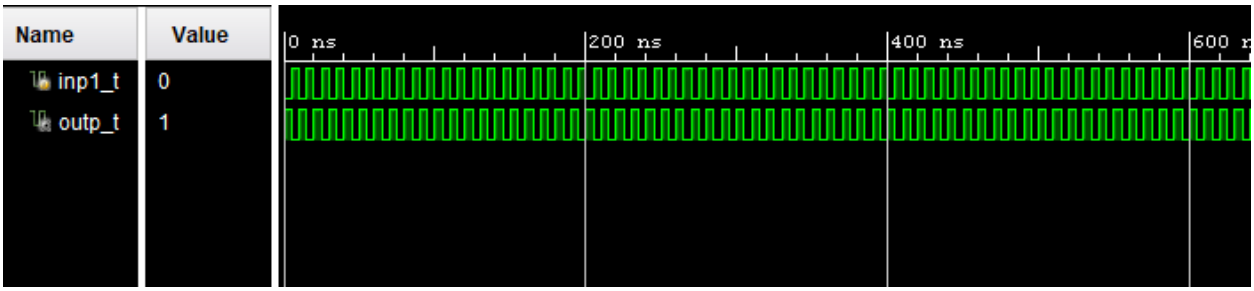




inp1	IN		V16	✓	14	LVCMOS33*
outp	OUT		U16	✓	14	LVCMOS33*

Results Statement:

Here is the simulation results for the INVERTER gate.

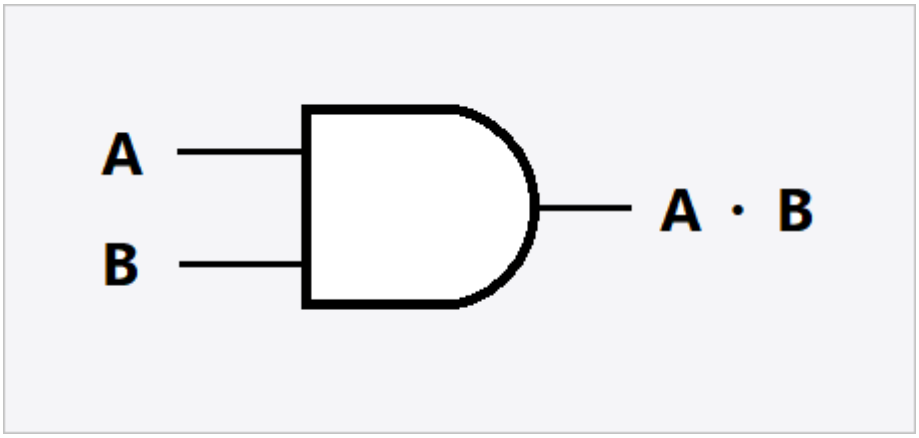


SW0	LED
On	Off
Off	On

AND Gate Verilog

Logic Diagram

The results from the simulation testing for the simulation for a AND gate.



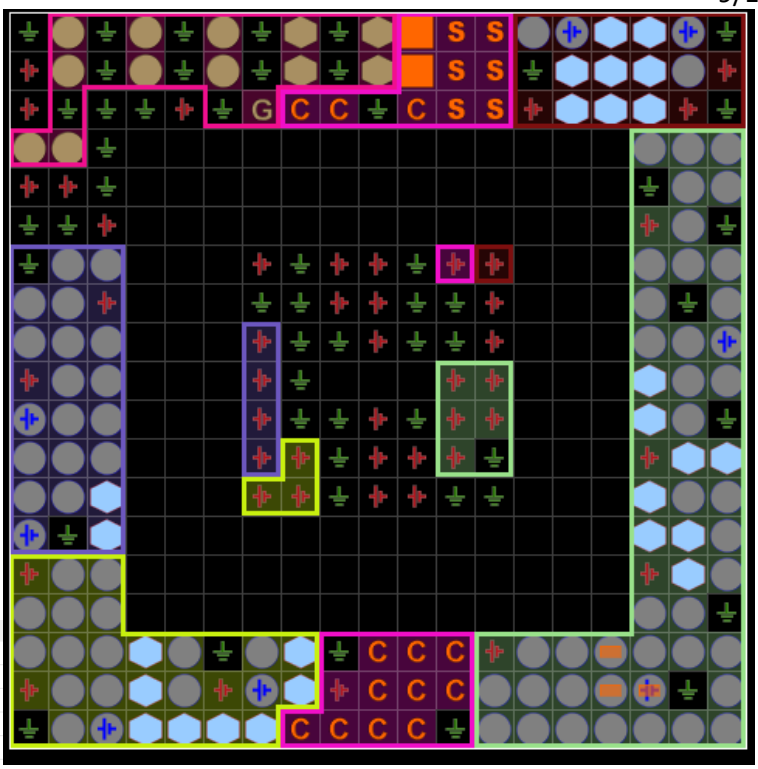
Design Specification Plan

The AND logic gate, with two inputs (Inp_1 and Inp_2) and one output (Outp). As shown in the results and screen shots, this was designed and implemented as instructed.

Project Summary | part_1.v

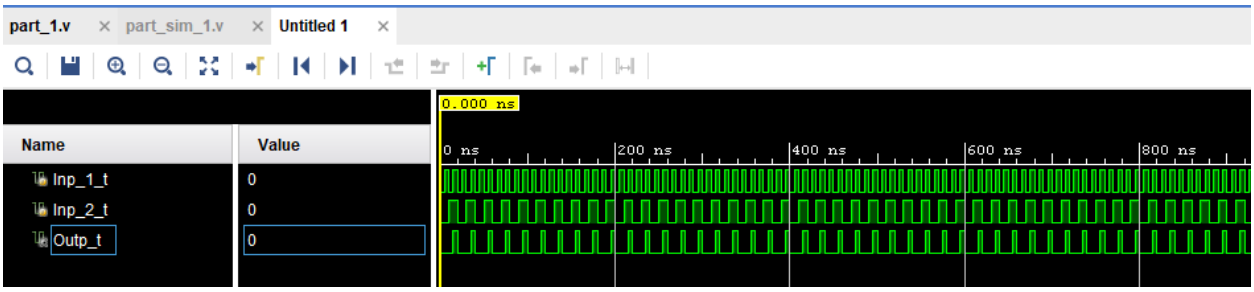
C:/Users/spens/Desktop/digital_systems_hdl/test_1/test_1.srcs/sources_1/new/part_1.v

```
2  //////////////////////////////////////////////////////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date: 09/14/2020 01:32:07 PM
7  // Design Name:
8  // Module Name: part_1
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////
21
22
23 module part_1(
24     input Inp_1,
25     input Inp_2,
26     output Outp
27     );
28
29 assign Outp = Inp_1 & Inp_2;
30 endmodule
```



Name	Direction	Neg Diff Pair	Package Pin
All ports (3)			
Scalar ports (3)			
Inp_1	IN		V17
Inp_2	IN		V16
Outp	OUT		U16

Results Statement

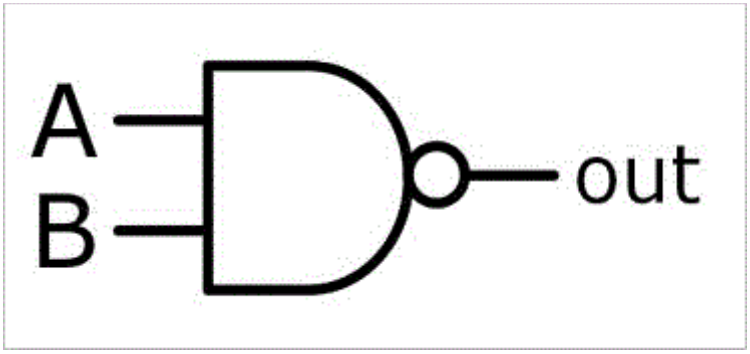


SW0	SW1	LED0
On	On	On
Off	On	Off
On	Off	Off
Off	Off	off

NAND Gate Verilog

Logic Diagram

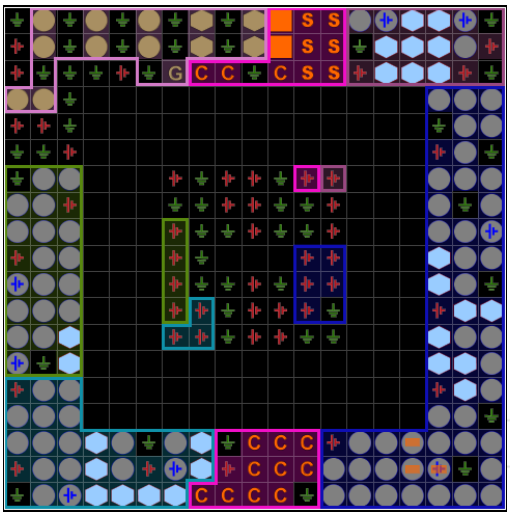
The results from the simulation testing for the simulation for a NAND gate.



Design Specification Plan

The AND logic gate, with two inputs (inp1 and inp2) and one output (outp). As shown in the results and screen shots, this was designed and implemented as instructed.

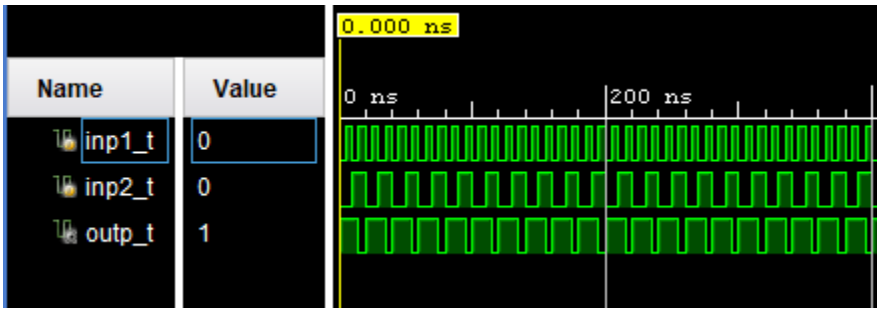
```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date: 09/19/2020 02:51:20 PM
7  // Design Name:
8  // Module Name: nand_gate
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////
21
22
23 module nand_gate(
24     input inp1,
25     input inp2,
26     output outp
27     );
28 assign outp = ~(inp1 & inp2);
29 endmodule
```



inp1	IN		V17
inp2	IN		V16
outp	OUT		U16

Results Statement

Explanation of pictures and tables.



SW0	SW1	LED0
On	On	Off
Off	On	On
On	Off	On
Off	Off	On

OR Gate Verilog

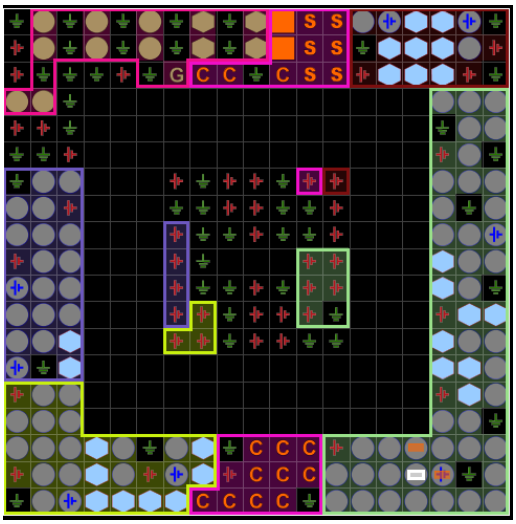
Logic Diagram

The results from the simulation testing for the simulation for a OR gate.

Design Specification Plan

The OR logic gate, with two inputs (inp1 and inp2) and one output (outp). As shown in the results and screen shots, this was designed and implemented as instructed.

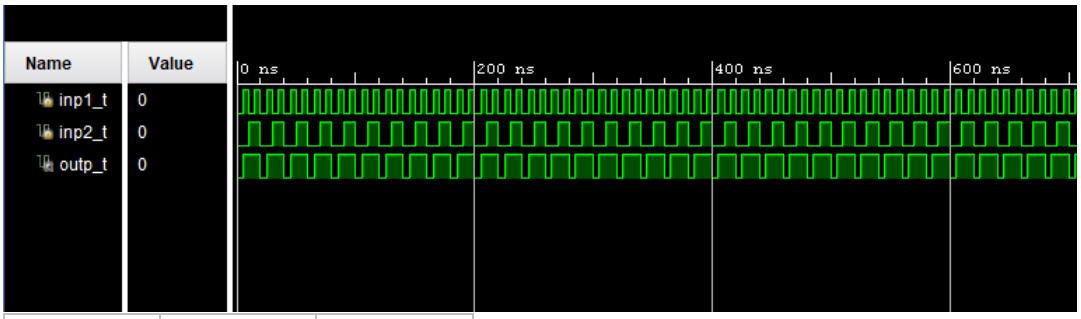
```
1  `timescale 1ns / 1ps
2  //////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date: 09/20/2020 08:18:1
7  // Design Name:
8  // Module Name: or_gate
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////
21
22
23 module or_gate(
24     input inp1,
25     input inp2,
26     output outp
27     );
28 assign outp = (inp1 | inp2);
29 endmodule
```



inp1	IN		V16	✓	14	LVCMOS33*
inp2	IN		V17	✓	14	LVCMOS33*
outp	OUT		U16	✓	14	LVCMOS33*

Results Statement

Explanation of pictures and tables.

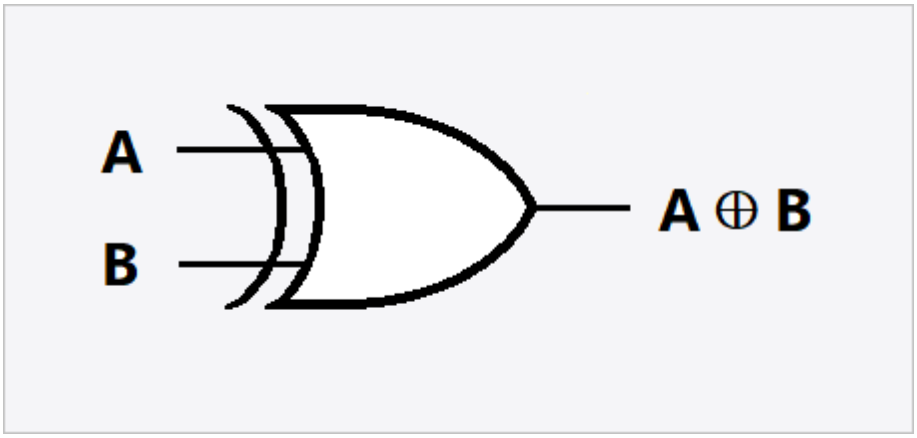


SW0	SW1	LED0
On	On	On
Off	On	On
On	Off	On
Off	Off	Off

XOR Gate Verilog

Logic Diagram

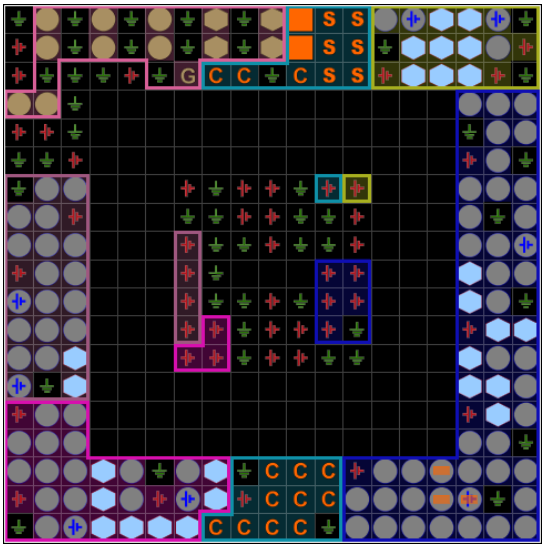
The results from the simulation testing for the simulation for a XOR gate.



Design Specification Plan

The XOR logic gate, with two inputs (inp1 and inp2) and one output (outp). As shown in the results and screen shots, this was designed and implemented as instructed.

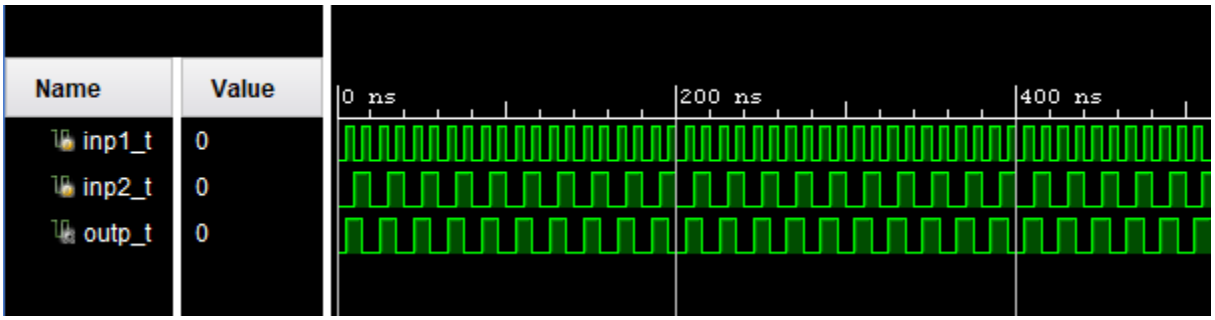
```
`timescale 1ns / 1ps
//////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 09/20/2020 08:39:04 PM
// Design Name:
// Module Name: xor_gate
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////


module xor_gate(
    input inp1,
    input inp2,
    output outp
    );
assign outp = inp1 ^ inp2;
endmodule
```



inp1	IN		V17	✓	14	LVCMOS33*
inp2	IN		V16	✓	14	LVCMOS33*
outp	OUT		U16	✓	14	LVCMOS33*

Results Statement

Explanation of pictures and tables.

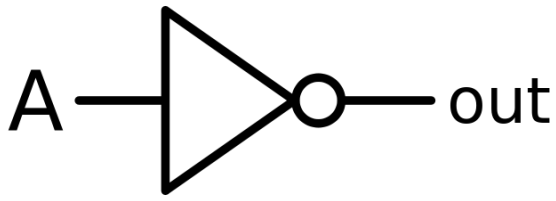


SW0	SW1	LED0
On	On	Off
Off	On	On
On	Off	On
Off	Off	Off

<u>INVERTER Gate Verilog</u>

Logic Diagram

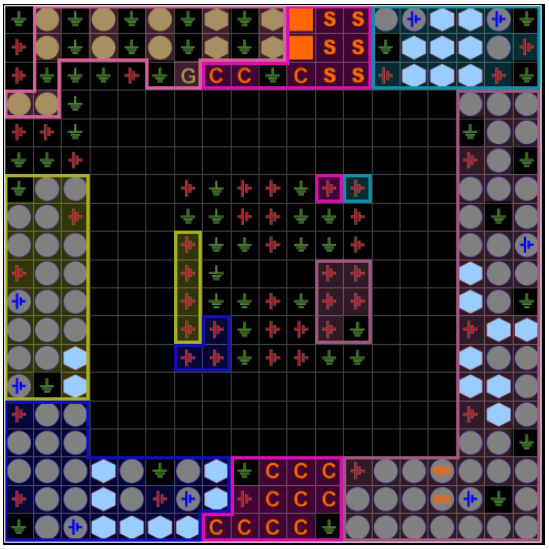
The results from simulation testing for the simulation for a INVERTER gate.



Design Specification Plan

The XOR logic gate, with two inputs (inp1) and one output (outp). As shown in the results and screen shots, this was designed and implemented as instructed.

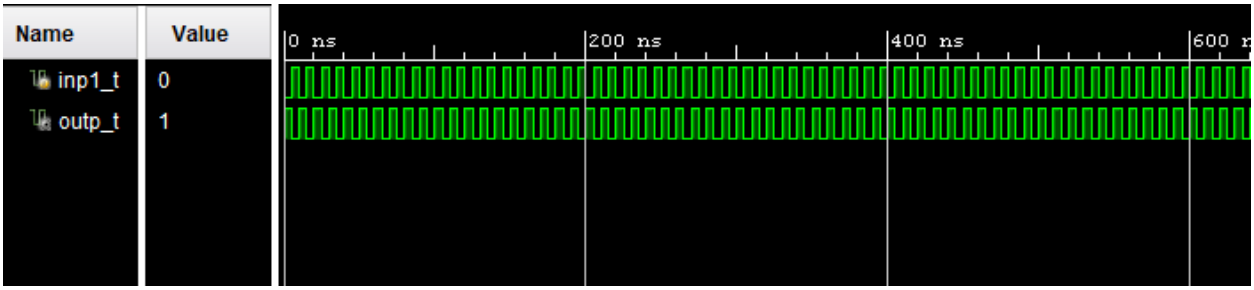
```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date: 09/20/2020 08:59:11 PM
7  // Design Name:
8  // Module Name: not_gate
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////
21
22
23 module not_gate(
24     input inp1,
25     output outp
26     );
27 assign outp = ~inp1;
28 endmodule
29
```



inp1	IN		V16	✓	14	LVCMOS33*
outp	OUT		U16	✓	14	LVCMOS33*

Results Statement

Explanation of pictures and tables.

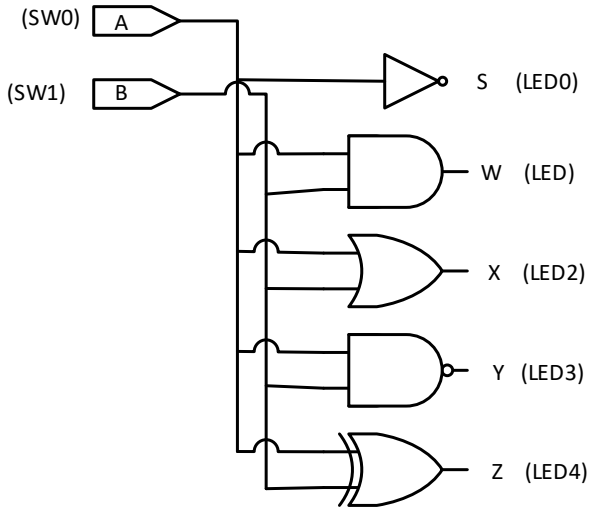


SW0	LED
On	Off
Off	On

2-input 5-output Gate Schematic

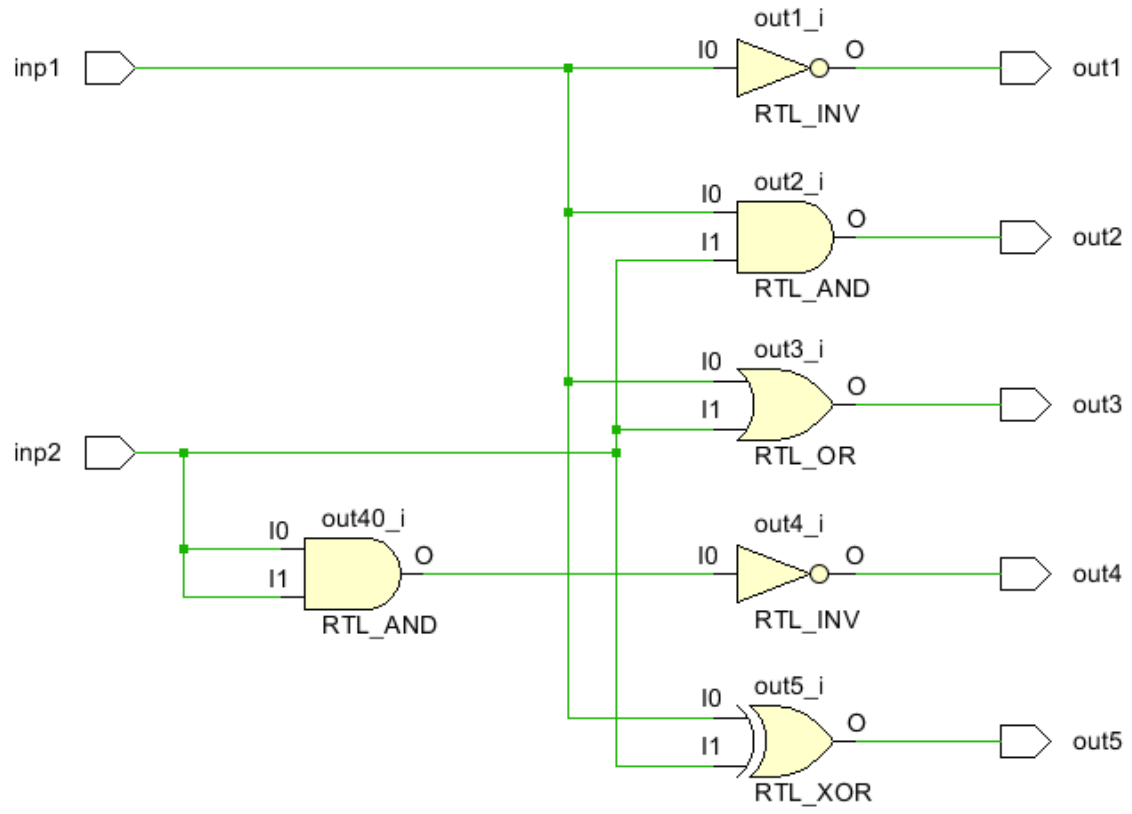
Logic Diagram

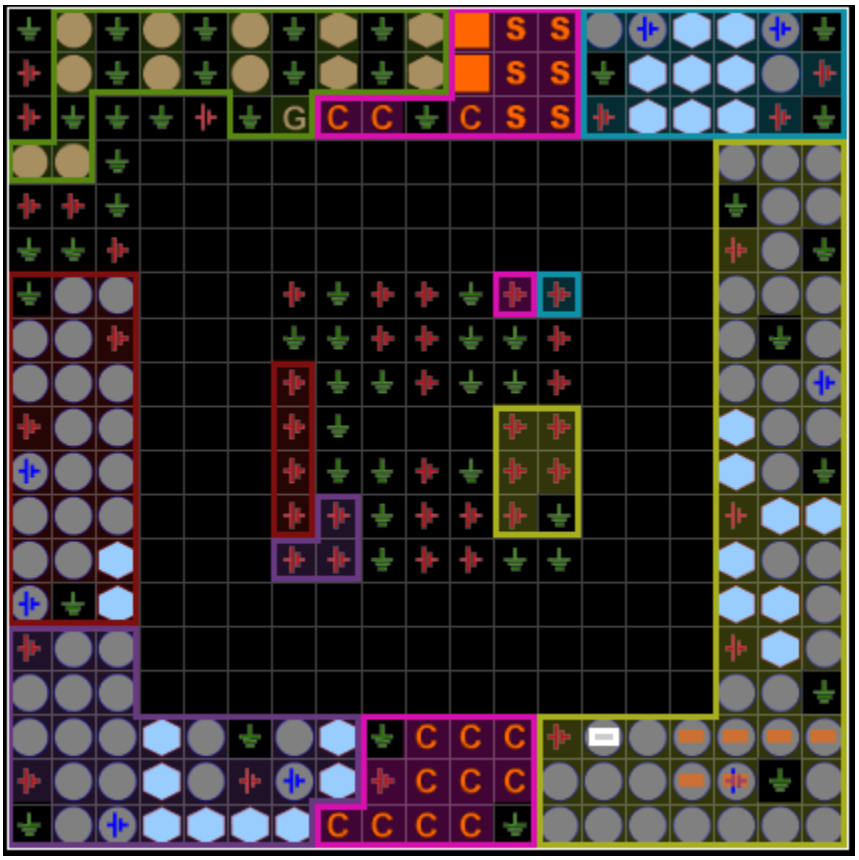
The results from simulation testing for the simulation for a 2-input 5-output gate.



Design Specification Plan

The 2-input 5-output logic gate, with two inputs (inp1 and inp2) and 5 outputs (out1, out2, out3, out4, out5). As shown in the results and screen shots, this was designed and implemented as instructed.

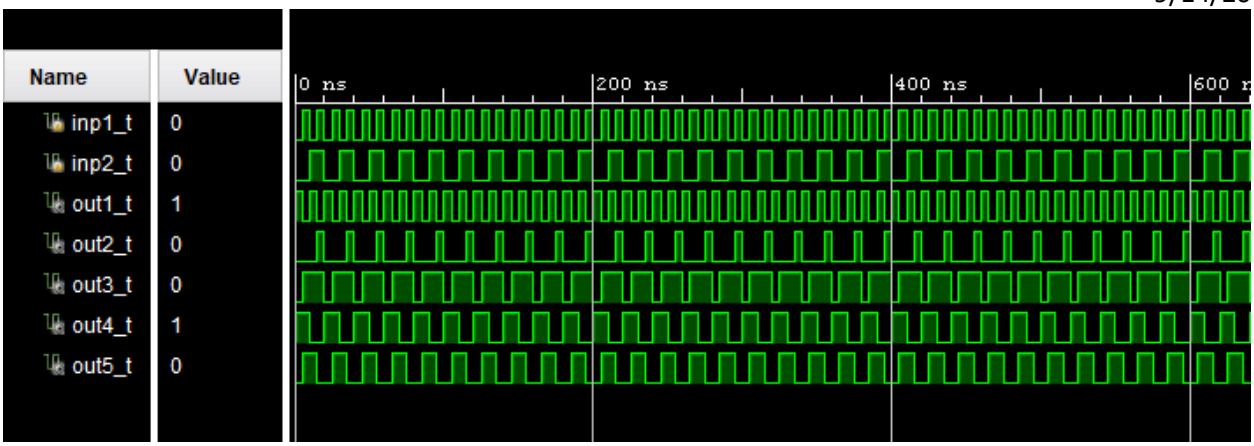




inp1	IN		V16	✓	14	LVCMOS33*
inp2	IN		V17	✓	14	LVCMOS33*
out1	OUT		U16	✓	14	LVCMOS33*
out2	OUT		U17	✓	14	LVCMOS33*
out3	OUT		U18	✓	14	LVCMOS33*
out4	OUT		U19	✓	14	LVCMOS33*
out5	OUT		U14	✓	14	LVCMOS33*

Results Statement

Explanation of pictures and tables.

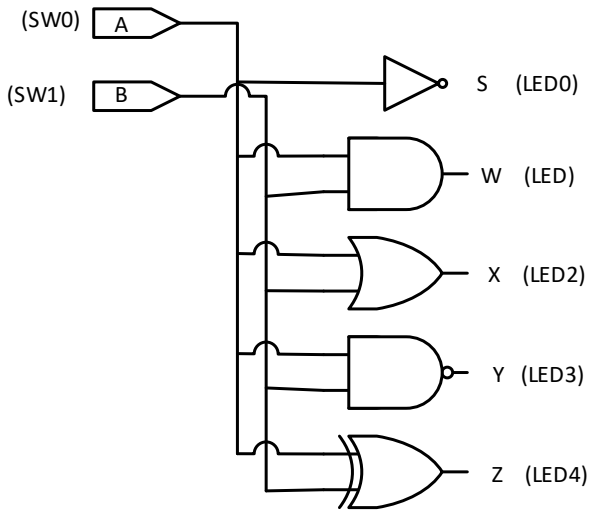


SW0	SW1	LED0	LED1	LED2	LED3	LED4
On	On	Off	On	On	Off	Off
Off	On	On	Off	On	On	On
On	Off	Off	Off	On	On	On
Off	Off	On	Off	Off	On	Off

2-input 5-output Gate Schematic

Logic Diagram

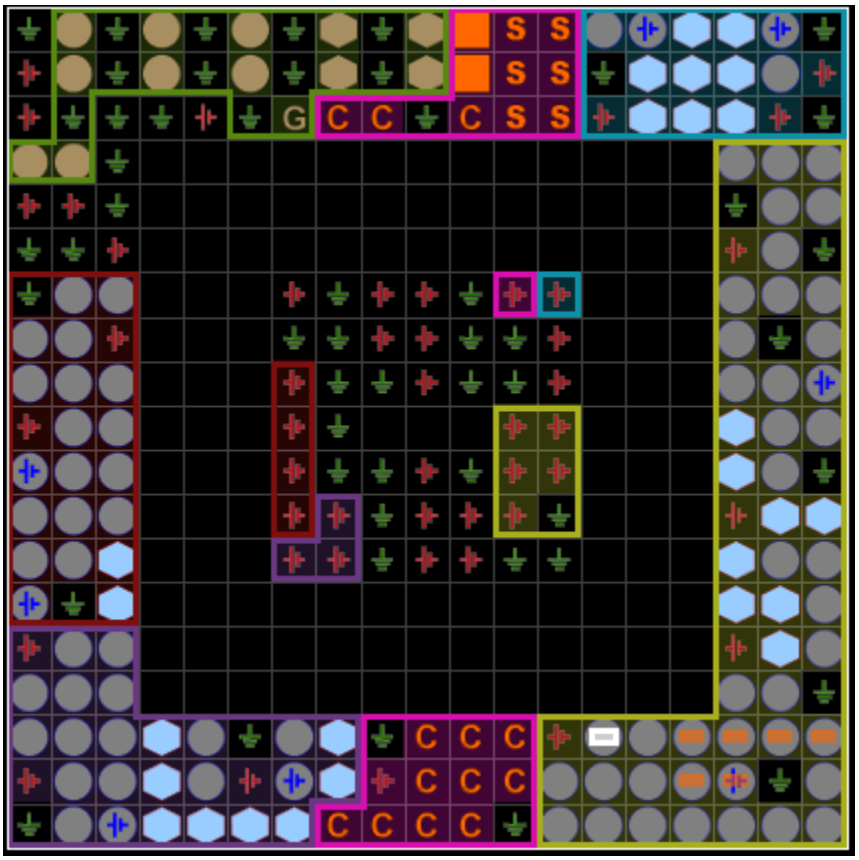
The results from the simulation testing for the simulation for a 2-input 5-output gate.



Design Specification Plan

The 2-input 5-output logic gate, with two inputs (inp1 and inp2) and 5 outputs (out1, out2, out3, out4, out5). As shown in the results and screen shots, this was designed and implemented as instructed.

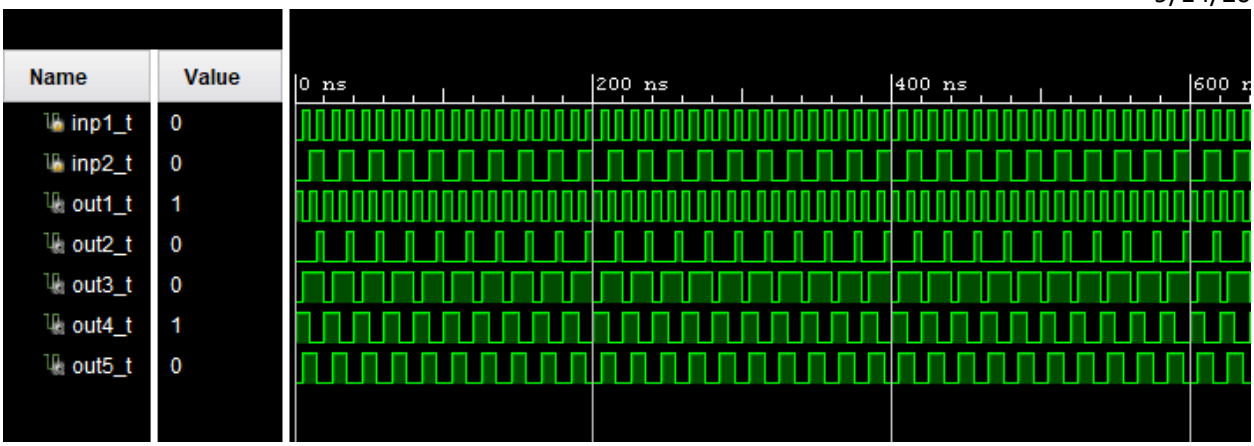
```
1   `timescale 1ns / 1ps
2   //////////////////////////////////////////////////
3   // Company:
4   // Engineer:
5   //
6   // Create Date: 09/21/2020 01:09:21 PM
7   // Design Name:
8   // Module Name: 2i5o
9   // Project Name:
10  // Target Devices:
11  // Tool Versions:
12  // Description:
13  //
14  // Dependencies:
15  //
16  // Revision:
17  // Revision 0.01 - File Created
18  // Additional Comments:
19  //
20  //////////////////////////////////////////////////
21
22
23  module io_gate(
24      input inp1,
25      input inp2,
26      output out1,
27      output out2,
28      output out3,
29      output out4,
30      output out5
31      );
32  assign out5 = inp1 ^ inp2;
33  assign out4 = ~(inp2 & inp2);
34  assign out3 = inp1 | inp2;
35  assign out2 = inp1 & inp2;
36  assign out1 = ~inp1;
37  endmodule
```



inp1	IN		V16	✓	14	LVCMOS33*
inp2	IN		V17	✓	14	LVCMOS33*
out1	OUT		U16	✓	14	LVCMOS33*
out2	OUT		U17	✓	14	LVCMOS33*
out3	OUT		U18	✓	14	LVCMOS33*
out4	OUT		U19	✓	14	LVCMOS33*
out5	OUT		U14	✓	14	LVCMOS33*

Results Statement

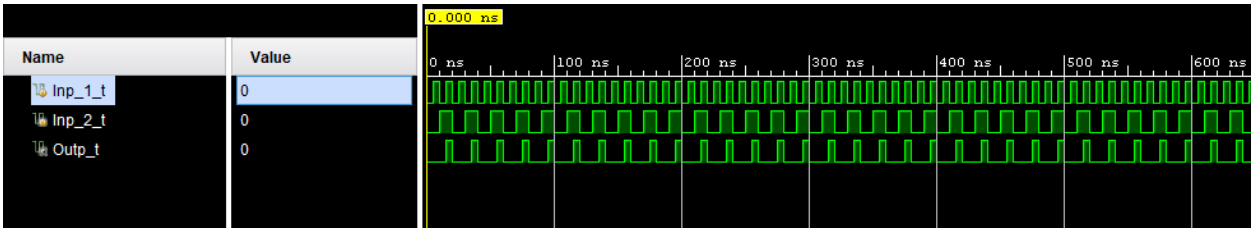
Explanation of pictures and tables.

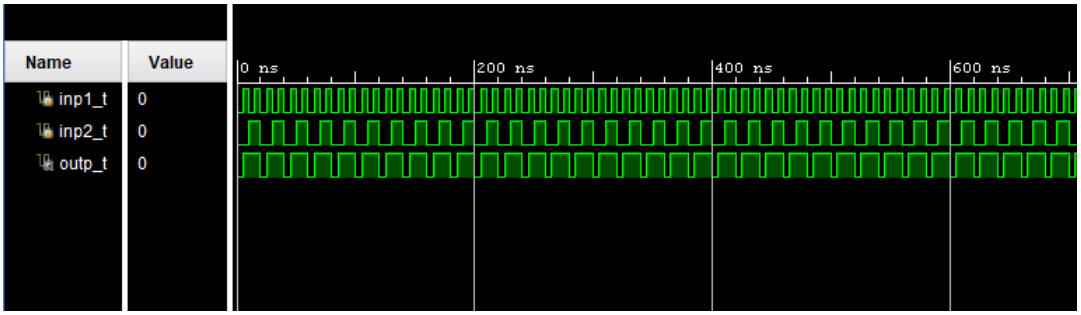


SW0	SW1	LED0	LED1	LED2	LED3	LED4
On	On	Off	On	On	Off	Off
Off	On	On	Off	On	On	On
On	Off	Off	Off	On	On	On
Off	Off	On	Off	Off	On	Off

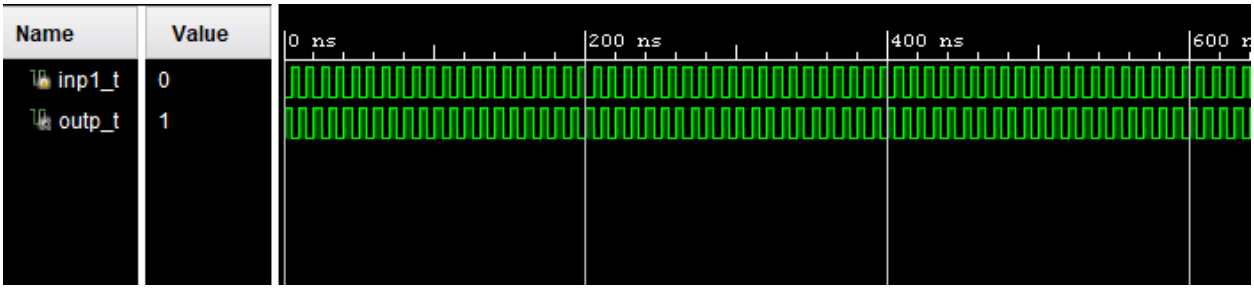
Test Plan

The initial design was tested through a simulation based off of the test bench, which was programmed for every possible input, which would give every output.

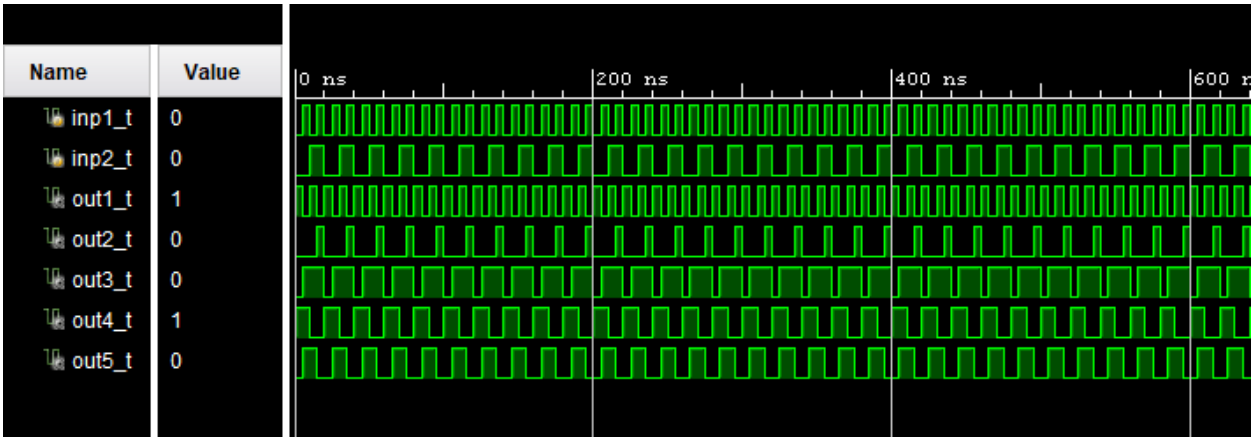




Test Bench for AND, OR, XOR, and NAND



Inverter Test Bench



2-input 5-output test bench

Conclusion

Sd My results were what I expected, and all the test and simulation results were correct and agreed with each other. By running each gate through a simulation and then analyzing said simulation, was very interesting, and lead to further understanding of logic gates functionality and how they work together in a more complex gate like the 2-input 5-output gate.