ENGG 125.03 Laboratory Activity 1A: Logic Gates

Activity Sheet; Submitted: 27 April 2021

Mallari, Kevin Luis T.
BS Computer Engineering
Electronics, Computer, and Communications Engineering
Ateneo de Manila University
Philippines

Abstract—The laboratory activity focuses on making the student familiar with the Verilog programming language and the Quartus environment. This was done in two (2) parts. The first part of the activity involves the use of simple logic gates and operations. The specific operations used were NOT, AND, OR, NOR, and XOR. For the second part of the activity, a combination of gates were used. The principles were the same with the first part. However, now, multiple gates are used in one line of code. The outputs of all the equations in the activity were simulated using Quartus and verified using truth tables.

Keywords - Arduino, Touch Sensor, Ultrasonic Sensor

I. Computations/Solutions

To make sure that the simulation is in line with the equations, the solutions for the different equations will be computed. The solutions will be shown using truth tables. The truth tables are evaluated based on the equations given.

Firstly, the truth table for the logic gates required in part 1 will be shown.

A	f (~A)
0	1
1	0

Fig. 1. NOT Gate Truth Table

A	B	f (AB)
0	0	0
0	1	0
1	0	0
1	1	1

Fig. 2. 2-input AND Gate Truth Table

A	B	f (A + B)
0	0	0
0	1	1
1	0	1
1	1	1

Fig. 3. 2-input OR Gate Truth Table

A	B	f ~(A + B)
0	0	1
0	1	0
1	0	0
1	1	0

Fig. 4. 2-input NOR Gate Truth Table

A	B	f (A ⊕ B)
0	0	0
0	1	1
1	0	1
1	1	0

Fig. 5. 2-input XOR Gate Truth Table

Second, the truth table for the logic gates required in part 2 will be shown. There are two outputs for this part. The equation of the first output is:

Out1 = [A AND NOT(B)] OR (B AND C)

while the equation of the second output is:

Out2 = (A AND B) OR (B AND C) OR (A AND C)

A	B	C	f ((A ~B) + (BC))

0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Fig. 6. Truth Table for (A ~B) + (BC)

A	B	C	f (AB + BC + AC)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Fig. 7. Truth Table for AB + BC + AC

II. HDL Code

Since there are two parts to the activity, there are also two sets of code. Shown below is the code for part 1.

```
module logicgates1(notOutput, andOutput,
orOutput, norOutput, xorOutput, A, B);

input A, B;
output notOutput, andOutput, orOutput,
norOutput, xorOutput;

// Code for a. NOT
assign notOutput = ~A;

// Code for b. 2-input AND
assign andOutput = A & B;

// Code for c. 2-input OR
assign orOutput = A | B;

// Code for d. 2-input NOR
assign norOutput = ~(A | B);

// Code for e. 2-input XOR
xor(xorOutput, A, B);

endmodule
```

Fig. 8. Code for Part 1

In the code, only two (2) inputs were used. For each gate/operation (NOR, AND, OR, NOR, and XOR) used, the appropriate syntax was coded. The answers to the operations were assigned to the respective output variable.

```
module logicgates2(Out1, Out2, A, B, C);

input A, B, C;
output Out1, Out2;

assign Out1 = (A & ~B) | (B & C);
assign Out2 = (A & B) | (B & C) | (A & C);

endmodule
```

Fig. 9. Code for Part 2

The same procedure that was used to make the code in part 1 was used in part 2. The equation was first taken into consideration, then, the code was made using the equation.

III. Output Circuit

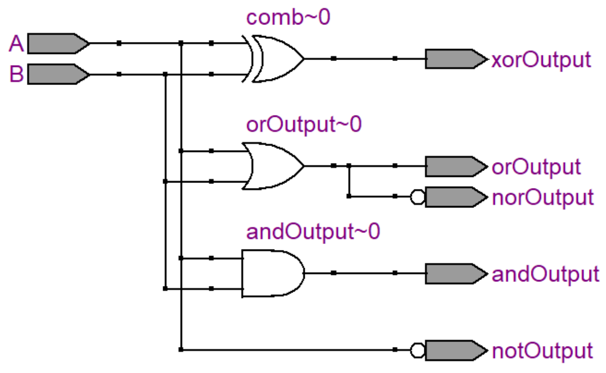

Fig. 10. Circuit for Part 1

The circuit for part 1, based on the code and the truth table can be seen in figure 10. The outputs of the circuit are shown in the simulations part of the paper.

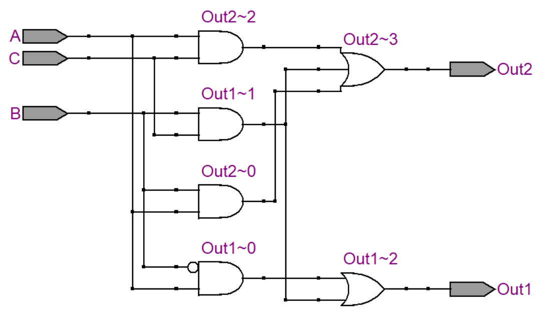

Fig. 11. Circuit for Part 2

The circuit for part 2 is seen on figure 11. Only one circuit was produced because only one set of code was used for both the equations. Because the two (2) equation for part 2 has a common factor (BC), the software was able to simplify the circuit.

IV. Screenshot(s) of Simulations and VWF file

The screenshots of the simulations are presented below.

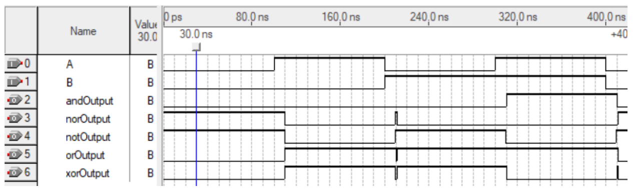

Fig. 12. Simulation for Part 1

The screenshot above shows the simulation for part 1 of the activity.

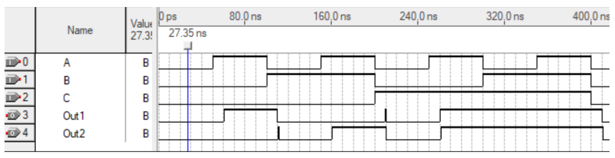

Fig. 13. Simulation for Part 2

The screenshot above shows the simulation for part 2 of the activity.

As seen in the screenshots, the simulations follow the truth tables for the most part. There are a few differences, however. For example, there is a delay in the result of the outputs by about 10 ns. This is because of the propagation delay. Another noticeable flaw in the simulation is the random spike up/down on the outputs that lasts for an extremely short time. This is most probably due to the inputs not instantaneously changing when transitioning from a HIGH value to a LOW value, and vice versa.