Chapter 3

Fundamentals of Digital Logic Circuits

3.1. Boolean Variables

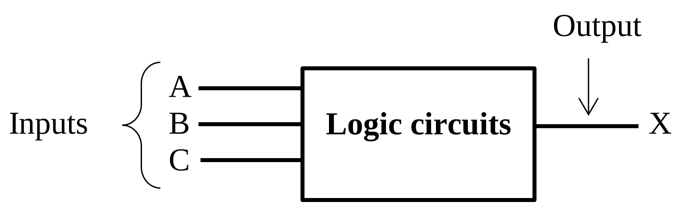
A logic gate is a building block of a digital circuit. Most logic gates have two inputs and one output and are based on Boolean algebra. At any given moment, every terminal is in one of the two binary conditions *false* (high) or *true* (low). False represents 0, and true represents 1. Depending on the type of logic gate being used and the combination of inputs, the binary output will differ. A logic gate can be thought of like a light switch, wherein one position the output is OFF -➔0, and in another, it is ON ➔ 1. (Rouse, 2019)

Boolean variables are often used to represent the voltage level present on a wire or at the input/output terminals of a circuit. For example, in a certain digital system, the Boolean value of 0 might be assigned to any voltage in the range from 0 V to 0.8 V, while the Boolean value of 1 might be assigned to any voltage in the range 2 V to 5 V. On the other hand, voltage between 0.8 V and 2 V are undefined (neither 0 nor 1) and should not occur under normal circumstances. Thus. Boolean 0 and 1 represent the state of a voltage variable, or what is called its logical level. Some of the more common way to represent logic 0 level or logic 1 level are: (Describing Logic Circuit, 2001)

Logic 0	**Logic 1**
False	True
Off	On
Low	High
No	Yes
Open switch	Closed switch

3.2. Truth Tables

A truth table is a means for describing how a logic circuit's output depends on the logic levels present at the circuit's inputs.

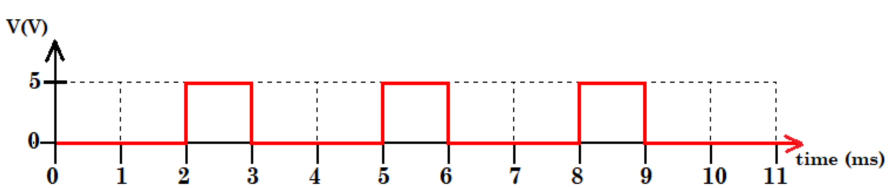


Truth Table - Logic Circuit			
Inputs			**Output**
C (MSB)	**B**	**A (LSB)**	**X**
0	0	0	**0**
0	0	1	**0**
0	1	0	**1**
0	1	1	**0**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**

The number of input combinations or all possible outputs is calculated by 2^N , where *N* is the number of inputs, and the logic level for output will depend on the type of logic circuits. Therefore, if three inputs, A, B, and C, are connected to a logic circuit, then the input combinations is $2^3 = 8$ possible outputs.

3.3. Timing Diagram

A timing diagram is the graphical representation of input and output signals as functions of time. Since the inputs and outputs can only take the values 0 (0 V) or 1 (5 V), their graphical representations are series of square pulses with a variety of time lengths. The first signal in the timing diagram is the LSB and the last signal is the MSB.



The inputs and outputs are drawn on the same diagram to show the input-output behavior of the digital system. A timing diagram is usually generated by an oscilloscope or logic analyzer. Computer-aided design tools have software simulator that generate timing diagrams. A timing diagram shows all possible input and output patterns, not necessarily in an order similar to that of a truth table. (Ferdjallah, 2019)

3.4. Basic logic gates

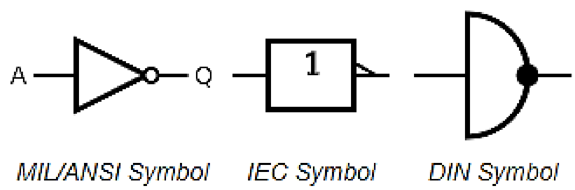
There are seven basic logic gates: NOT, AND, OR, NAND, NOR, XOR, and XNOR.

3.4.1. NOT gate

A **NOT** logic gate, also often called **Inverter**, has only one input and one output signal. Logically with NOT gates, the input and the output swap, so if the input is 1, the output is logically inverted to 0; likewise if the input is 0, its output will 1.

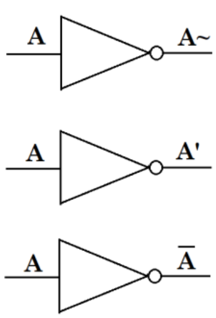
NOT Gate – Truth Table	
Input	**Output**
A	Q
0	1
1	0

There are three digital symbols for the NOT gate, but the most common NOT gate symbol uses is the MIL/ANSI symbol:

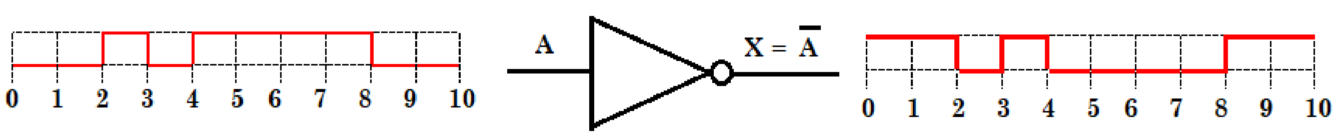


Digital symbol of a NOT gate

In logic variable, an inverted variable is expressed with the symbol tilde ~ or single prime ′ next to the inverted variable or a horizontal bar on top of the inverted variable:



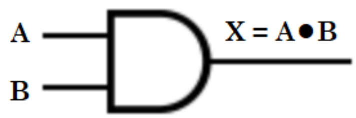
Example) Find the output X of the following signal



3.4.2. AND Gate

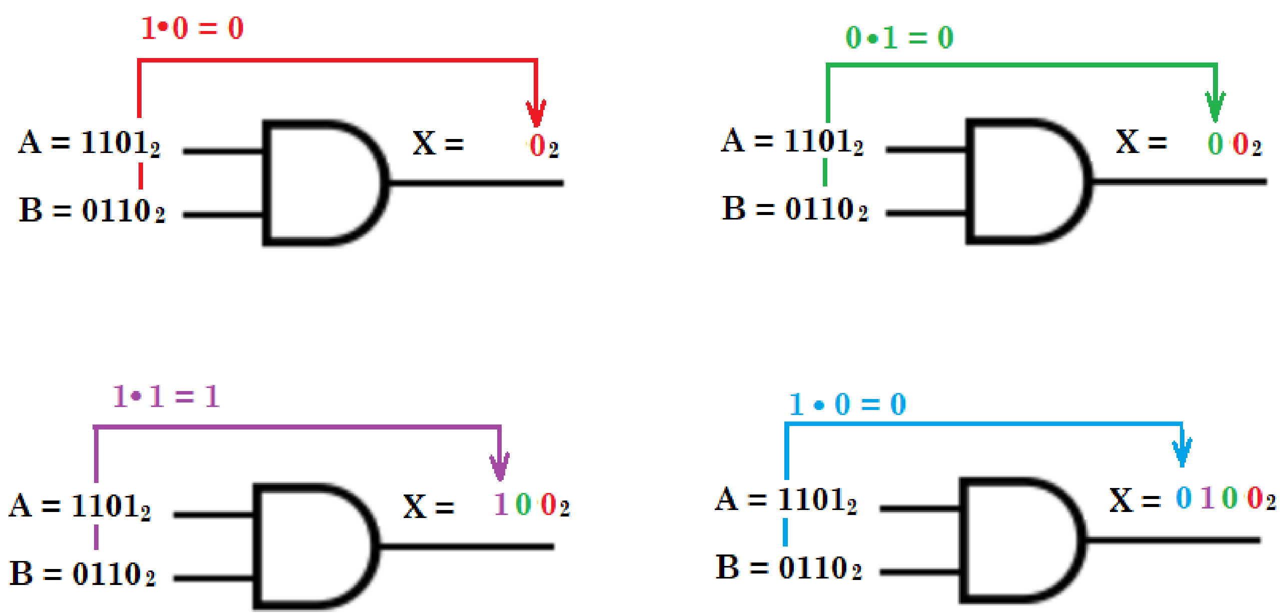
The AND gate is a basic digital logic gate, with two or more inputs and one output, that implements logical conjunction or logic multiplication. Therefore, a logic gate AND returns an output HIGH (1) only if ALL the inputs to the AND gate are HIGH (1).

2-input AND Gate – Truth Table		
Input		**Output**
A	**B**	**X = A•B**
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**



Digital symbol of an AND gate

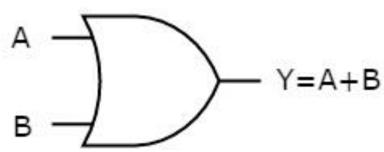
Example) Find the output X given input A and B



3.4.3. OR Gate

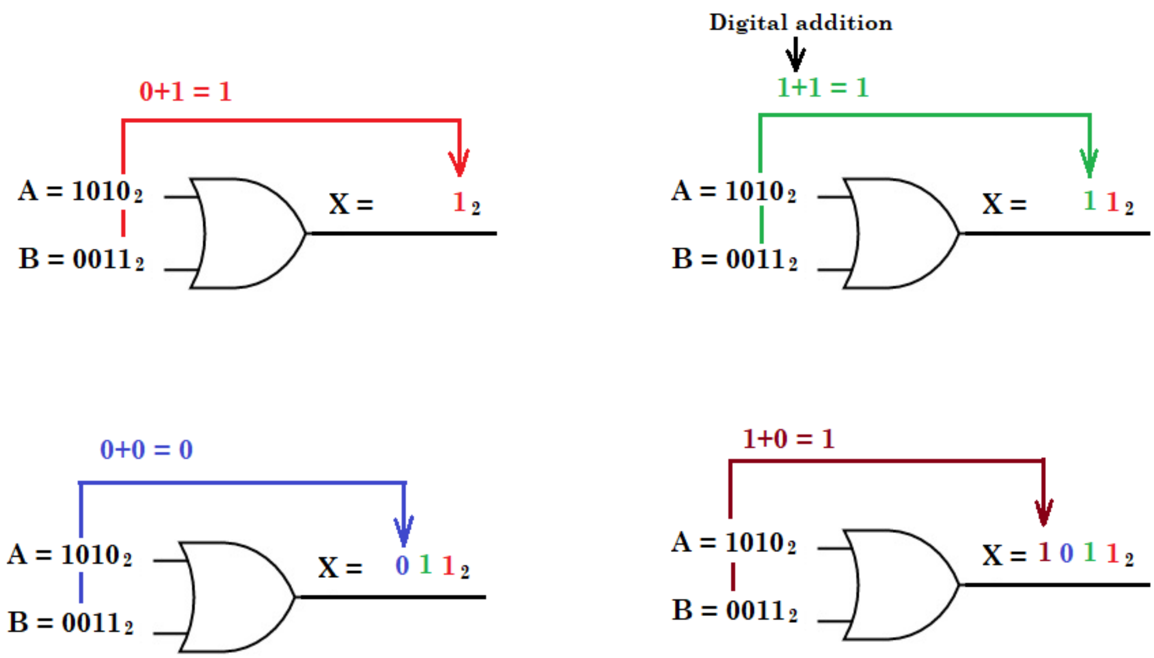
An **OR** gate is a digital gate, with also two or more inputs and one output, that performs logical disjunction, or logical addition. The output of an **OR gate** is HIGH (1) when one or more of its inputs are HIGH (1). If all of an **OR gate's** inputs are LOW (0), then the output of the **OR gate** is LOW (0).

2-input OR Gate – Truth Table		
Input		**Output**
A	**B**	**Y = A+B**
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**



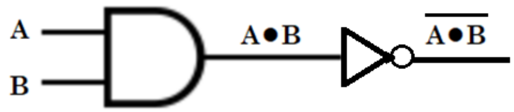
Digital symbol of an OR gate

Example) Find the output X given input A and B



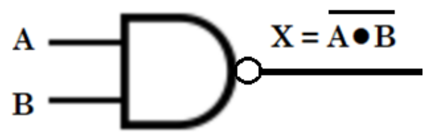
3.4.4. NAND Gate

The output of a NAND (NOT-AND) gate is the inversion of the output of an AND gate. In other words, its output is complement to that of an AND gate.



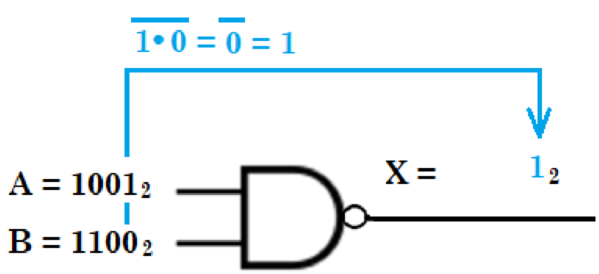
Therefore, a NAND output is LOW (0) only if all its inputs are HIGH (1)

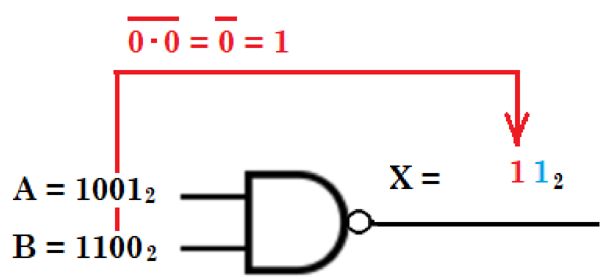
2-input NAND Gate – Truth Table			
Input		**Output**	
A	**B**	**AND gate** $A \circ B$	**NAND gate** $Y = \overline{A \circ B}$
0	0	0	**1**
0	1	0	**1**
1	0	0	**1**
1	1	1	**0**

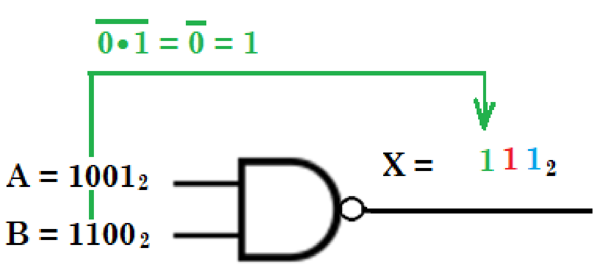


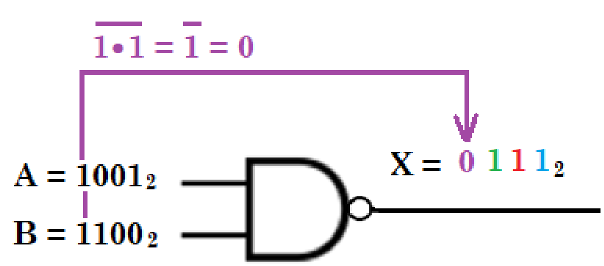
Digital symbol of a NAND gate

Example) Find the output X given input A and B



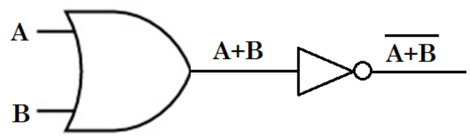






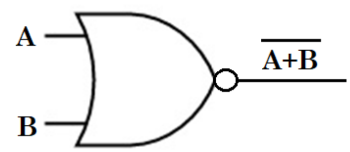
3.4.5. NOR gate

The output of a NOR (NOT-OR) gate is the inversion of the output of an OR gate. In other words, its output is complement to that of an OR gate.



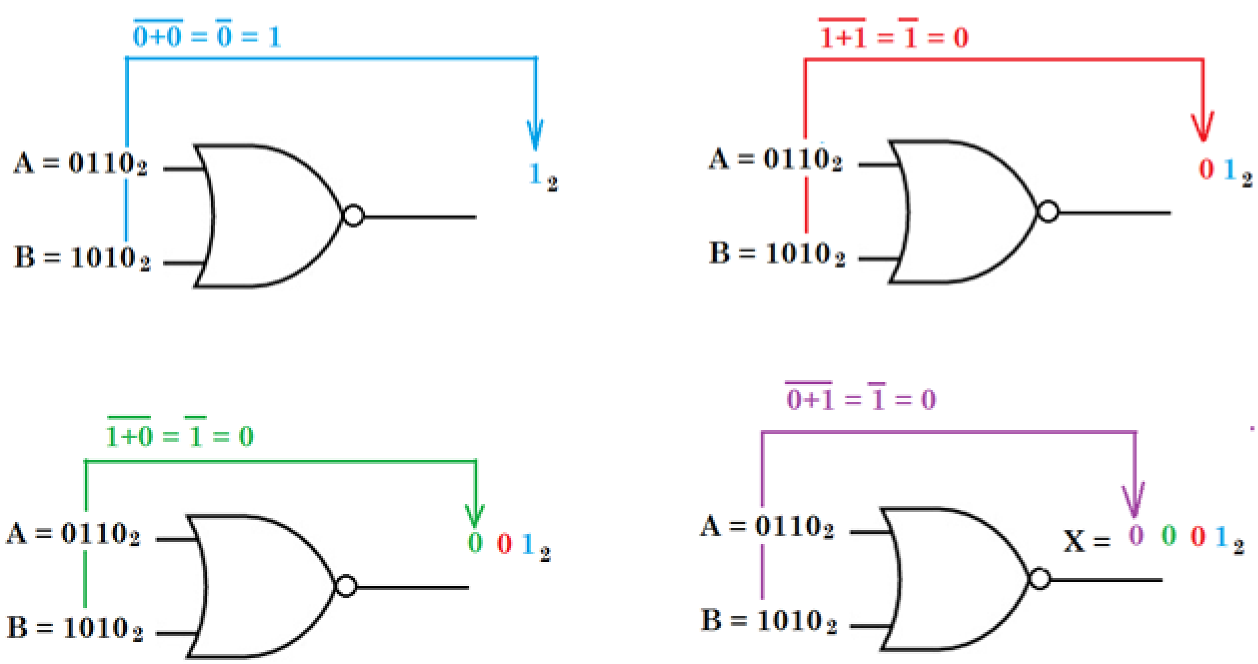
Therefore, a NOR output is HIGH (1) only if all its inputs are LOW (0)

2-input NOR Gate – Truth Table			
Input		**Output**	
A	**B**	**OR gate** $A+B$	**NOR gate** $Y = \overline{A+B}$
0	0	0	**1**
0	1	1	**0**
1	0	1	**0**
1	1	1	**0**



Digital symbol of a NOR gate

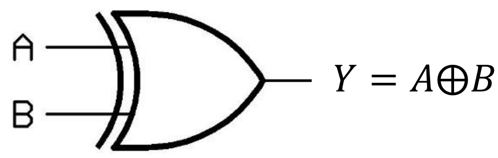
Example) Find the output X given input A and B



3.4.6. Exclusive-OR, XOR, Ex-OR, Gate

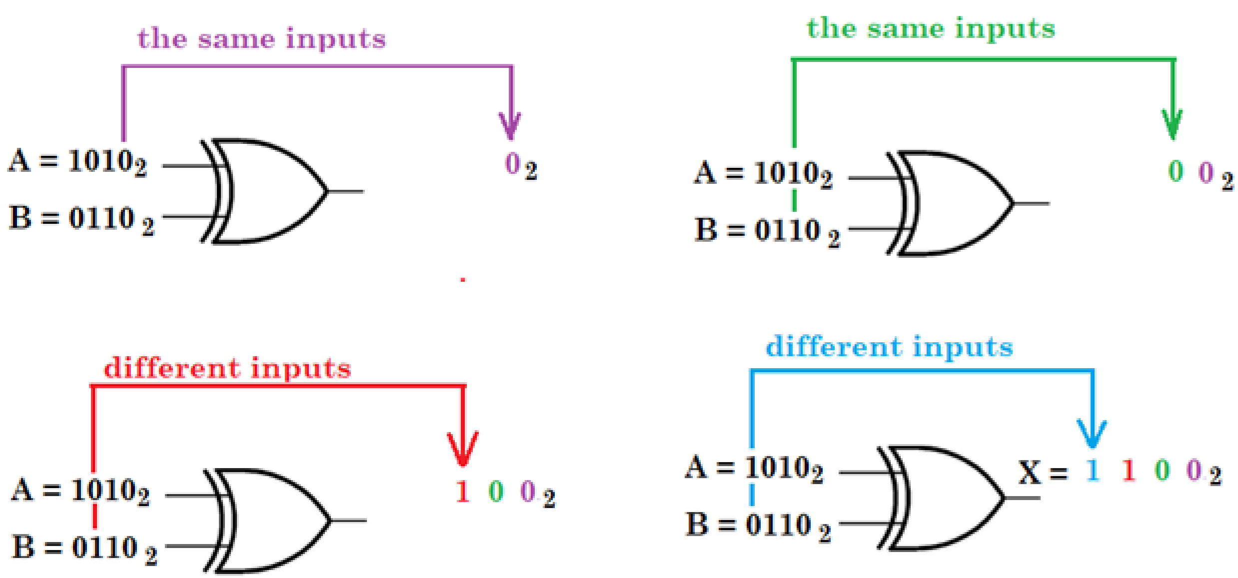
Exclusive-OR and Exclusive-NOR are formed by a combination of other gates, which we will cover in the next chapter. The output of an X-OR gate is HIGH (1) only when the two inputs are at opposite logic levels. Otherwise, if the two inputs are at the same logic level, it will produce an output of LOW (0).

2-input X-OR Gate – Truth Table		
Input		Output
A	B	$Y = A \oplus B$
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**



Digital symbol of a XOR gate

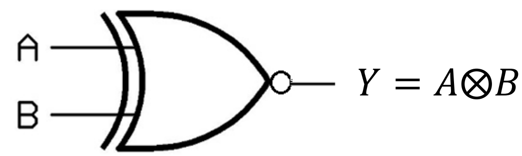
Example) Find the output X given input A and B



3.4.6. Exclusive-NOR, XNOR, Ex-NOR, Gate

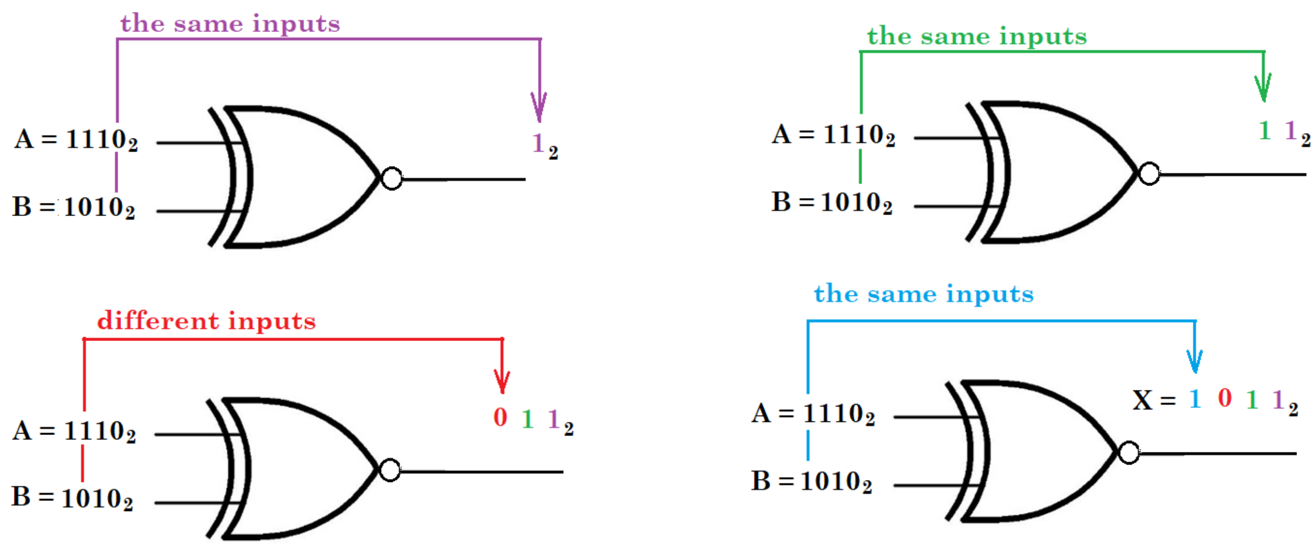
The output of an X-NOR gate is HIGH (1) only when the two inputs are at the same logic levels. Otherwise, if the two inputs are at the different logic level, it will produce an output of LOW (0).

2-input X-NOR Gate – Truth Table		
Input		**Output**
A	**B**	$Y = A \otimes B$
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**1**



Digital symbol of a XNOR gate

Example) Find the output X given input A and B



References

Describing Logic Circuit. (2001). In N. s. Ronald J. Tocci, *Digital Systems* (pp. 58-59). Pearson Prentice Hall.

Engineering and Technology History Wiki. (n.d.). *Engineering and Technology History Wiki*. Retrieved from Binary Numbers and Binary Math: ethw.org/Binary_Numbers_and_Binary_Math

Ferdjallah, M. (2019). *Introduction to Digital Systems: Modeling, Synthesis, and Simulation Using VHDL .* Retrieved from O'Reilly Safari Books Online: https://www.oreilly.com/library/view/introduction-to-digital/9780470900550/chap3-sec019.html

Rouse, M. (2019, July). *Logic Gate (AND, OR, XOR, NOT, NAND, NOR and XNOR)*. Retrieved from WhatIs: https://whatis.techtarget.com/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR