Digital Logic Part 1

Boolean Logic Gates

G. Snider Glenforest Secondary School

Among the nineteenth century figures who contributed to the development of computers in the twentieth century, George Boole has a unique standing. It is not difficult to see the mechanical calculating machines which Charles Babbage attempted to build, or the tabulating machines manufactured by Herman Hollerith to process data for the U. S. Census Bureau as predecessors of modern computers. George Boole, however, was a mathematician who developed a rather abstract system of rules for logical thought. His system which used equations with variables and operators to represent statements, came to be known as Boolean Algebra.

George Boole

Boole's achievements were praised by his contemporaries, and the mathematician Augustus De Morgan added a corollary known as "De Morgan's Theorem" (De Morgan also corresponded with Charles Babbage). The relevance of Boolean Algebra to the design of electrical calculating machines was not recognized until Claude Shannon, a graduate student at the Massachusetts Institute of Technology wrote a Master's thesis on the topic in 1937. Shannon had been employed by another pioneer in the field of computers, Vannevar Bush, to work on the Differential Analyser, a modern version of Babbage's Analytical Engine. The electrical relays used in the automated machines of the day were always in one of two states, on or off, and Shannon recognized that the Boolean system, where 0 and 1 represent False and True statements, could be used to systematize and simplify the design of complex electrical circuits.

AND

Let's begin by looking at how the conjunction "and" is treated in Boolean logic. Take the statement:

"If Jack is nimble and if Jack is quick then Jack can jump over the candlestick."

This sentence can be broken down into three statements, each of which can be represented by a variable.

A "Jack is nimble"

B "Jack is quick"

x "Jack can jump over the candlestick"

In the English language, conditional statements are those which be gin with the word "if" (of possibly "when", "given that", "assuming that" or other similar words and phrases). In this example, the conditions are that Jack must be nimble and Jack must be quick. If these two conditions are met, then there is a result or outcome; in this case, the result is that Jack is able to jump over the candlestick. If either, or both conditions are unfulfilled or false, then Jack cannot perform the gymnastic feat.

The statements used in this system can only be true or false. When we have two conditions, therefore, there are four possible combinations of conditions and results which must be considered:

A is false	and	B is false	then	x is false
A is false	and	B is true	then	x is false
A is true	and	B is false	then	x is false
A is true	and	B is true	then	x is true

We can use a table to chart the various possible combinations.

A	B	x
False	False	False
False	True	False
True	False	False
True	True	True

This type of table is called a truth table. Each statement has only two possible values, true and false, which can be represented by the numbers 1(true) and 0 (false). Therefore the truth table can be simplified by using ones and zeros to represent true and false. The resulting truth table for this statement looks like this:

AND

A	B	x
0	0	0
0	1	0
1	0	0
1	1	1

Since both A *and* B must be true for x to be true, this is known as the AND statement.

The relationship between the conditions (in electronic circuits the conditions are often referred to as inputs) and the result (output) can also be expressed algebraically. In Boolean algebra, the conditions and the result are separated by an equals sign.

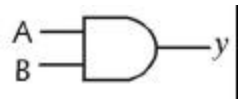
The AND statement is represented by the multiplication sign, and the whole statement can thus be written:

$$A \cdot B = x$$

Intuitively, it might seem that the AND connection should be represented by the plus sign rather than multiplication In fact, when performing a Boolean search on an internet search engine or a computer data base, the plus sign is sometimes used for AND. However, if we substitute the values from the truth table for the variables, it should be obvious why Boole chose the multiplication sign to represent AND:

$$0 \cdot 0 = 0$$
$$0 \cdot 1 = 0$$
$$1 \cdot 0 = 0$$
$$1 \cdot 1 = 1$$

There is another way of representing Boolean statements which is very commonly used in designing digital circuits. Each function has a graphical representation known as a gate. The AND gate looks like this:



AND

In the case of the AND gate, *both* conditions (inputs) must be true if the outcome is to be true. If there were more than two conditions connected by the word "and" then they would *all* have to be true for the result to be true. Another way to express the meaning of the AND function therefore is to say:

"All are true"

For example, a university might have the following entrance requirements:

"The applicant must (A) have completed 6 grade 12 courses, (B) have an average mark of 75%, and (C) pass an entrance exam."

This statement can be written as an algebraic expression like this:

$$A \cdot B \cdot C = y$$

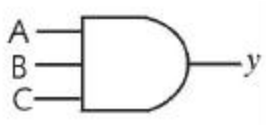
Since there are three conditions, A, B, and C, each of which may be true or false, there are (2x2x2) = 8 possible combinations, so the truth table in this case will have 8 lines. (Note that the possibilities are ordered from 0 to 7 in binary.)

AND

A	B	C	*y*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

There is only one combination of the conditions which satisfies the entrance requirements: *all* the conditions must be true.

A three input AND gate would be drawn as shown below.



3-Input AND

OR

The second term in the English language which has a specific meaning in digital logic is the word "or". Take ,for example, the statement:

> "If the pitcher throws four balls or if the batter is hit by a pitch, then the batter may advance to first base."

As we did in the previous example, we can divide this sentence into three statements:

A "the pitcher throw four balls"

B "the batter is hit by a pitch"

y "the batter may advance to first base"

If either of the conditions is true, then the outcome will be that the batter gets a free base. Notice that this includes the possibility that with a count of three balls, the pitcher throws a ball which hits the batter; if both conditions are true, the batter advances.

The truth table for the OR function has the same combination of inputs as the AND truth table, but the results are different.

OR

A	B	x
0	0	0
0	1	1
1	0	1
1	1	1

If *any* of the inputs to the OR gate are true, then the outcome is true.

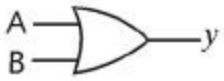
In Boolean algebra, the OR function is represented by the + sign. Thus the OR statement is written as an expression in this way:

$$A + B = x$$

The reasons for this may not be immediately clear, but if we substitute the values from the truth table into the expression, and also substitute "greater than or equal to" for the equals sign, we can see the logic at work here:

$$A + B \geq x$$
$$0 + 0 \geq 0$$
$$0 + 1 \geq 1$$
$$1 + 0 \geq 1$$
$$1 + 1 \geq 1$$

The OR gate looks like this:

OR 

In the case of the OR gate, *either or both* of the conditions (inputs) must be true if the outcome is to be true. If there were more than two conditions connected by the word "or" then the result will be true if any of the conditions are true. Another way to express the meaning of the OR function therefore is to say:

"Any are true"

To take another example from baseball rules,

"If a batted ball (A) lands in fair territory beyond the infield, or if (B) after hitting the ground it is in or above fair territory as it passes first or third base, or if (C) while still on the infield, it rolls or bounces into fair territory, then it is a fair ball."

When there are more than two conditions, as in this example, only one of them need be true for the result to be true. Although it is physically impossible for one ball to satisfy all three conditions at once, this does not invalidate the logic of the statement.

This statement can be written as an algebraic expression like this:

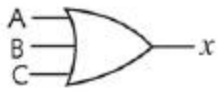
$$A + B + C = x$$

The truth table for a three input OR gate would look like this:

OR

A	B	C	*x*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

The graphic symbol for the three-input OR gate looks like this:



NOT

The simplest logical statement recognized by Boole is the negation, which is merely the statement that something is not true. Algebraically this is expressed by placing an overbar above the variable or expression which is being negated. For example, the expression $\overline{A}$ means simply "A is not true", while $\overline{A+B}$ means "neither A nor B is true".

For an example, consider the statement:

"If it is not daytime, then it is nighttime."

If we use variables to represent these statements, then we have:

A "It is daytime"

$\overline{A}$ "It is not daytime"

x "It is nighttime"

Thus the Boolean expression for the sentence is:

$$\overline{A} = x$$

The NOT gate, or Inverter, is the graphic representation of this equation. Unlike the AND and the OR gates, it has only one input and one output.

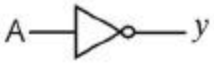
Notice, however, that as with the other gates we discussed, the input can be either true or false. So the statement "if it is not daytime, then it is nighttime" also implies that if it *is* daytime, then it is *not* nighttime. But in each case when one statement is true, then the other is false.

The truth table for the Inverter is very simple, but not trivial. There are only two possible states for the input—true or false, 1 or 0—, and the output will always be the opposite, or inverted.

NOT

A	x
0	1
1	0

The negation or inversion is actually indicated on the NOT gate by the circle or "bubble" on the output of the gate. Without the bubble, the triangular gate is called the Identity gate.

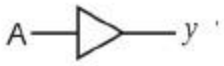


INVERTER

IDENTITY

"A rose is a rose is a rose."

When Gertrude Stein wrote these famous words in her poem "Sacred Emily", she most



IDENTITY

certainly did not have digital logic in mind. But the sentiment expressed sums up the function of the Identity gate very nicely: the output is always the same as the input.

In electronics, this symbol represents a buffer or an amplifier. Both these devices are designed so that the output is the same as the input.

IDENTITY

A	x
0	0
1	1

NAND

The defendant testified under oath: "I did not kill her and bury the body." He may be making a true statement, and yet still be guilty of a crime! How is this possible? What is he actually denying? If we parse this sentence, it can be written "I did not [kill her and bury the body]." In other words, the defendant is denying that he committed both crimes. While it is possible that he committed neither crime, he has not categorically said "I did not kill her and I did not bury the body".

The difference between these statements can be seen more clearly if we use Boolean algebra to analyse it. There are two acts involved:

A He killed her

B He buried her

The first statement asserts "I did not kill her and bury the body" This can be written:

$$\overline{A \cdot B} = y$$

The second statement, "I did not kill her and I did not bury the body" is fundamentally different. It can be written:

$\overline{A} \cdot \overline{B} = y$

Let's look at the truth tables for these two expressions. In the first case, we have an AND statement with an overbar above it, asserting that the statement taken as a whole is false. In other words, the accused is making a statement of the form NOT AND, which is normally abbreviated NAND. The output of the NAND gate truth table is the inverse of the AND gate output. In other words, wherever there is a zero in the output column of the AND truth table there is a one in the NAND column, and vise versa.

A	B	AND	NAND
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

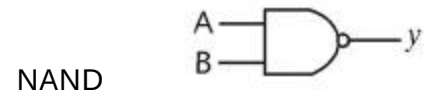
If we look at this truth table we can see that there are three conditions under which the NAND statement is true: When A is false, when B is false, and when both are false. When both A and B are true, then the NAND statement is false. Going back to our defendant's testimony, his statement is true as long as he did not commit both crimes, but he may still be guilty of one.

The expression $\overline{A} \cdot \overline{B} = y$ is quite different. To construct the truth table, we must first find the inverse of A and B for all possible combinations, then apply the criteria for the AND gate to the inverted values (all must be true). The result looks like this:

A	B	$\overline{A}$	$\overline{B}$	$\overline{A} \cdot \overline{B}$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

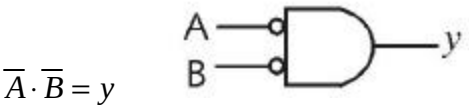
Thus we can see that the statement "I did not kill her and I did not bury her", is an unequivocal statement of innocence, because it is true only when both denials are true.

The NAND gate looks like an AND gate with the negation bubble on the output.



NAND

It is also possible to place the bubbles on the inputs to the AND gate, which corresponds to the expression $\overline{A} \cdot \overline{B} = y$



$\overline{A} \cdot \overline{B} = y$

The meaning of the NAND gate can thus be expressed as *"not both"* or if there are more than two inputs:

"not all"

NOR

As it happens, there is another way that someone accused of misdoing can state his or her innocence. A literate eight-year-old faced with an allegation might say:

"I neither chewed gum in class, nor stuck used gum under the desk!"

If we parse this sentence, we can see that the form "neither … nor" can be written:

"I did not [chew gum in class or stick used gum under the desk]."

Again we have two different acts which are being denied:

A chew gum in class

B stick used gum under the desk

Written as a Boolean expression, we have the equation:

$$\overline{A + B} = y$$

The truth table for this expression is derived by inverting the outputs for the OR gate:

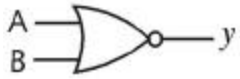
A	B	OR	NOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Notice that the NOR statement is true only when both the inputs are false. Another way of

saying this is *"neither is true"* or if there are more than two inputs:

"none is true"

The NOR gate is drawn like the OR gate with the output inverted:



NOR

EXCLUSIVE OR

In discussing the OR gate, it was noted that the result is true when either condition is true and also when both are true. The OR function is sometimes described as the inclusive OR, because the case where both are true is included.

Often, however, we use the word OR in an exclusive fashion. A mother in the video store might tell her children that they can rent a movie or a video game. If the spoiled brats clamor to rent a game and a movie, Mom will likely respond: "No, I meant *one or the other but not both*!"

The truth table for the exclusive OR, also known as the XOR gate, looks like this:

XOR

A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

Notice that when both conditions are true, the outcome is false. The XOR gate is drawn like an OR gate with the addition of an extra curved line on the input side:

XOR

To indicate the similarities and differences between the inclusive and exclusive OR gate, the expression for the XOR gate uses the plus sign inside a circle:

$$A \oplus B = y$$

EXCLUSIVE NOR

The Exclusive NOR or XNOR is simply an XOR gate with the output inverted. The expression is written like this:

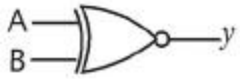
$$\overline{A \oplus B} = y$$

The truth table for the Exclusive NOR, resembles that of the XOR in that there are two zeros and two ones in the output column, unlike the other gates which always have one unique output. The outputs, however are the inverse of the XOR outputs. Thus the table looks like this:

XNOR

A	B	x
0	0	1
0	1	0
1	0	0
1	1	1

Following the pattern already established, the symbol for the XNOR is drawn by adding a bubble to the output of the XOR gate.



XNOR

The function of the XNOR gate can be summed up in the phrase:

"neither or both are true"

This is not a phrase one often hears spoken, but this gate is very useful in digital electronics. Notice that the output is true whenever A = B. This gate can therefore be very useful in making comparisons between two binary numbers.

Review Questions

Part 1

Fill in the blanks

1. In Boolean Logic the numbers 1 and 0 represent _____ and _____.
2. When there are two inputs to a logic gate, the truth table will have _____ lines.
3. The output of the AND gate is 1when _____ of the inputs are true.
4. The output of the OR gate is 1when _____ of the inputs are true.
5. The output of the NOT gate is 1when the input is _____.
6. The output of the NAND gate is 0 when _____ of the inputs are _____.
7. The output of the NOR gate is 1when _____ of the inputs are _____.
8. "One or the other but not both" describes the operation of the _____ gate.
9. The truth table below corresponds to the _____ gate.

A	B	x
0	0	1
0	1	0
1	0	0
1	1	0

10. The truth table below corresponds to the _____ gate.

A	B	x
0	0	0
0	1	1
1	0	1
1	1	1

11. The truth table below corresponds to the _____ gate.

A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

12. The truth table below corresponds to the _____ gate.

A	B	x
0	0	0
0	1	0
1	0	0
1	1	1

Part 2

Answer the following questions in your notebook.

13. Why is the multiplication sign used to represent the AND function in Boolean algebra?
14. Which comparator would be more appropriate than the equals sign in the expression $A + B = x$? Why?
15. Give an example of an English language statement which could be represented by the Boolean expression $\overline{A + B} = y$. (Do not copy from the text, write your own question.)
16. Give an example of an English language statement which could be represented by the Boolean expression $A \cdot B \cdot C = y$.
17. What use is a gate like the Identity function where the output is the same as the input?
18. Draw the graphical symbol for the NOR gate.
19. Draw the graphical symbol for the AND gate.
20. Draw the graphical symbol for the NOT gate.
21. Draw the graphical symbol for the XNOR gate.
22. Draw the graphical symbol for the OR gate.
23. Explain the difference between the Inclusive OR gate and the Exclusive OR gate.
24. What symbol is used in algebraic expressions to represent negation or inversion?
25. How is inversion represented on the graphical symbols of the logic gates?

Part 3

On the next page complete the chart by filling in the symbol, expression and truth table for each of the eight logic gates.

Digital Logic Gates

NAME	SYMBOL	EXPRESSION	TRUTH TABLE
IDENTITY		$x = A$	A | y 0 | 0 1 | 1
NOT			A | y 0 | 1 |
AND			A | B | y 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
OR			A | B | y 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
NAND			A | B | y 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
NOR			A | B | y 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
XOR			A | B | y 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
XNOR			A | B | y 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |