

Boolean Algebra

Logical Operations

- Now we have known how to represent numbers and other types of information using bits.
- Suppose we ask a computer to do a simple math $1 + 2$. We can imagine that the computer will convert 1 and 2 into binary first and then do the addition.
- How does a computer do an addition? Or, how is the addition implemented on hardware?
 - Interestingly, [hardware function implementation](#), such as executing mathematical operations on two numbers, moving information from one place to another, controlling processes, etc., [is based on boolean algebra](#).
- Boolean algebra is a [mathematical framework](#) we use to describe digital circuits at an abstract level.
- In boolean algebra, a variable either has the value **1 (TRUE)** or **0 (FALSE)**.
 - These values consistent with the two logical states a bit can have, and are also easy to be represented using [electronic signal](#) which is an electrical [voltage](#) that [switches between low and high values at regular intervals](#). High values are 1s, and low values are 0s. We also call this electronic signal “[clock](#)”.
- The [basic logical operations](#) are **AND**, **OR**, and **NOT**, which are symbolically represented by dot, plus sign, and overbar.
 - **AND**: $A \text{ AND } B = AB$ (or $A \cdot B$)
 - **OR**: $A \text{ OR } B = A + B$
 - **NOT**: $\text{NOT } A = \bar{A}$ (or A')
- We evaluate a boolean expression from left to right. The [precedence](#) is: **NOT > AND > OR**.
- We will use these logical operations to implement the addition on hardware.

Truth Tables

- To understand the meaning of these logical operation, we need to know truth tables.
- Truth tables are a way to define the logical operations.
- A truth table lists the values of an operation for every possible combination of values of operands.

NOT	
A	$\bar{A}$
0	1
1	0

OR		
A	B	$A+B$
0	0	0
0	1	1
1	0	1
1	1	1

AND		
A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

- Note: $A + B$ is not arithmetic addition here, so $1 + 1 = 1$, not 2

- Example1: Generate a truth table for $(A+B)(A+C)(A'+B')$

A	B	C	A+B	A+C	A'+B'	$(A+B)(A+C)(A'+B')$
0	0	0	0	0	1	0
0	0	1	0	1	1	0
0	1	0	1	0	1	0
0	1	1	1	1	1	1
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	0	0
1	1	1	1	1	0	0

- Exercise: Generate a truth table for $ABC + A'B'C'$ (not covered in class but for your practice)

A	B	C	ABC	A'B'C'	$ABC + A'B'C'$
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	1	1	0	1