Help

Boolean Algebra is the mathematical framework for digital logic. Some fundamental laws of Boolean Algebra are listed in Table 4.2. With these laws, we consider A, B, C either as Booleans or as individual bits of a logic operation.

A & B = B & A	Commutative Law
A | B = B | A	Commutative Law
(A & B) & C = A & (B & C)	Associative Law
(A | B) | C = A | (B | C)	Associative Law
(A | B) & C = (A & C) | (B & C)	Distributive Law
(A & B) | C = (A | C) & (B | C)	Distributive Law
A & 0 = 0	Identity of 0
A | 0 = A	Identity of 0
A & 1 = A	Identity of 1
A | 1 = 1	Identity of 1
A | A = A	Property of OR
A | (~A) = 1	Property of OR
A & A = A	Property of AND
A & (~A) = 0	Property of AND
~(~A) = A	Inverse
~(A | B) = (~A) & (~B)	De Morgan's Theorem
~(A & B) = (~A) | (~B)	De Morgan's Theorem

Table 4.2. Fundamental laws of Boolean Algebra.

CHECKPOINT 4.5

Let A be an 8-bit number, and consider the operation B=A&0x20, where A&0x20 is performed bit by bit. Now, if we consider B as a Boolean value, what is the relationship between A and B?

Hide Answer

B will be 0x20 if bit 5 of A is 1, and will be 0x00 if bit 5 of A is 0. Since B is Boolean, B will be true if and only of bit 5 of A is 1.

CHECKPOINT 4.6

Let C be an 8-bit number, and consider the operation C=C&0xDF. How does this operation affect C?

Hide Answer

The value 0xDF has bit 5 low, and other bits high. The AND with 0xDF will clear bit 5 of C.

CHECKPOINT 4.7

Let D be an 8-bit number, and consider the operation D=D|0x20. How does this operation affect D?

Hide Answer

The OR with 0x20 will set bit 5 of D.

When multiple operations occur in a single expression, **precedence** is used to determine the order of operation. Usually NOT is evaluated first, then AND, and then OR. This order can be altered using parentheses.

There are multiple ways to symbolically represent the digital logic functions. For example, $\sim A$ A' $\overline{A}$ $!A$ and $\neg A$ are five ways to represent NOT(A). One can use the pipe symbol (|) or the plus sign to represent logical OR: $A|B$ $A{+}B$. In this class we will not use the plus sign to represent OR to avoid confusion with arithmetic addition. One can use the ampersand symbol (&) or a multiplication sign (* •×) to represent logical AND: $A\&B$ $A{\bullet}B$. In this class we will not use the multiplication sign to represent AND to avoid confusion with arithmetic multiplication. Another symbolic rule is adding a special character (* n \) to a name to signify the signal is negative logic (0 means true and 1 means false). These symbols do not signify an operation, but rather are part of the name used to clarify its meaning. E.g., Enable* is a signal than means enable when the signal is zero.

Help

CHECKPOINT 4.8

Let C bit an 8-bit number. Are these two operations the same or different? C=C&0xDF C=C&(~0x20)

Hide Answer

Since ~0x20 is equal to 0xDF, they are the same. The second is a clearer way to signify clear bit 5.

Help