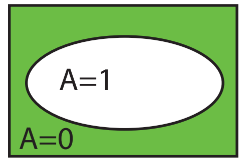
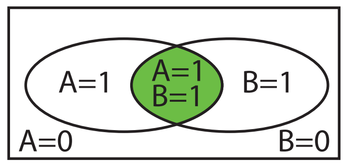
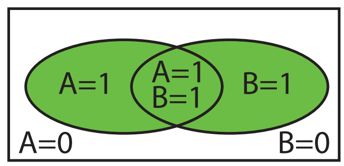
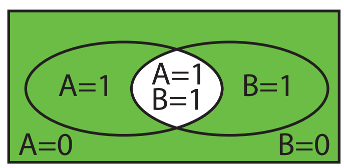
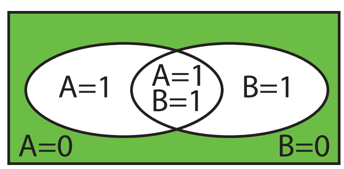
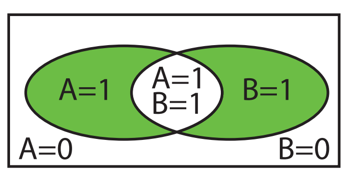
Boolean Algebra Primer

James Cook University

Operator	**Symbols**	**Truth table**	**Diagram**	**Description**
NOT	$\bar{A}$, $!A$, $\sim A$	$\begin{array}{c\|c} A & \bar{A} \\ \hline 0 & 1 \\ 1 & 0 \end{array}$		Invert the value
AND	AB, $A.B$, $A \wedge B$	$\begin{array}{cc\|c} A & B & A.B \\ \hline 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{array}$		Are all values 1?
OR	$A+B$, $A \vee B$	$\begin{array}{cc\|c} A & B & A+B \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{array}$		Is any value 1?
NAND	$\overline{AB}$, $\overline{A.B}$, $\overline{A \wedge B}$	$\begin{array}{cc\|c} A & B & \overline{A.B} \\ \hline 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{array}$		Are they not all 1?
NOR	$\overline{A+B}$, $\overline{A \vee B}$	$\begin{array}{cc\|c} A & B & \overline{A+B} \\ \hline 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{array}$		Is neither of them 1?
XOR	$A \oplus B$, $A \veebar B$	$\begin{array}{cc\|c} A & B & A \oplus B \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{array}$		Is exactly one value 1?

Fundamentals of Boolean algebra

Law	**Definition**	**Comments**
Commutative law	$AB = BA$ $A + B = B + A$	
Associative law	$(AB)C = A(BC)$ $(A + B) + C = A + (B + C)$	Typical algebraic manipulations apply.
Distributive law	$A(B + C) = AB + AC$	
Inversion law	$\overline{\overline{A}} = A$	The NOT operator is its own inverse.
AND identities	$A.0 = 0$ $A.A = A$ $A.1 = A$ $A.\overline{A} = 0$	The AND operator behaves like multiplication of values 0 and 1.
OR identities	$A + 0 = A$ $A + A = A$ $A + 1 = 1$ $A + \overline{A} = 1$	The OR operator behaves like addition of 0 and 1, where answers saturate at 1.
Identities for simplifying	$A + AB = A$ $A + \overline{A}B = A + B$	These are helpful in algebraic simplifications.
De Morgan's Theorem	$\overline{AB} = \overline{A} + \overline{B}$ $\overline{A + B} = \overline{A}.\overline{B}$	In words: "to bring a NOT into or out of brackets, swap AND and OR."

From truth tables to algebraic expression

Example:

Given the following truth table, write down an algebraic expression for x.

In truth tables, an "x" means "don't care". The variable indicated by an "x" has no impact on the result of that row.

a	b	c	x
x	1	0	1
x	0	0	1
1	0	0	1
0	0	1	0

The **minterm canonical form** is formed as follows:

1. Locate all rows of the truth table where the output variable is 1. For each such row, write down the AND of all columns. Ignore cells with "don't care" values:

a	b	c	x	Working
x	1	0	1	$b.\overline{c}$
x	0	0	1	$\overline{b}.\overline{c}$
1	0	0	1	$a.\overline{b}.\overline{c}$
0	0	1	0	—

2. Write down the OR of all values in the working column.

$$x = b.\overline{c} + \overline{b}.\overline{c} + a.\overline{b}.\overline{c}$$

3. Simplify (e.g. factorise)

$$\begin{aligned} x &= b.\overline{c} + \overline{b}.\overline{c} + a.\overline{b}.\overline{c} \\ &= (b + \overline{b} + a.\overline{b})\overline{c} \\ &= (1 + a.\overline{b})\overline{c} \\ &= 1\overline{c} \\ &= \overline{c} \end{aligned}$$

Implementing expressions using NAND or NOR gates

NAND and NOR gates are universal, i.e. any Boolean expression can be implemented using them.

Strategies for converting expressions into NAND or NOR:

1. Use De Morgan's theorems to turn AND into OR and vice versa.
2. Add double NOT terms ($\overline{\overline{A}} = A$) if any logic gates are in the form of AND instead of NAND, or OR instead of NOR.
3. Implement any NOT operators using the identities: $\overline{A} = \overline{AA} = A \text{ NAND } A$ and $\overline{A} = \overline{A + A} = A \text{ NOR } A$.

Example:

Implement $x = ab + \bar{c}$ using NAND gates.

Thinking: "turn the OR into an AND using De Morgan's Theorem"

$$\begin{aligned} x &= ab + \bar{c} \\ &= \overline{\overline{ab + \bar{c}}} \\ &= \overline{\overline{ab}.c} \\ &= \overline{(a \text{ NAND } b).c} \\ &= (a \text{ NAND } b) \text{ NAND } c. \end{aligned}$$

Implement $x = ab + \bar{c}$ using NOR gates.

Thinking: "turn the AND into an OR using De Morgan's Theorem"

$$\begin{aligned} x &= ab + \bar{c} \\ &= \overline{\overline{ab}} + \bar{c} \\ &= \overline{\bar{a} + \bar{b}} + \bar{c} \\ &= (\bar{a} \text{ NOR } \bar{b}) + \bar{c} \\ &= \overline{\overline{(\bar{a} \text{ NOR } \bar{b}) + \bar{c}}} \\ &= \overline{(\bar{a} \text{ NOR } \bar{b}) \text{ NOR } \bar{c}}. \end{aligned}$$

If only NOR gates are available then the NOT operators would be implemented using NORs.

$$\begin{aligned} x &= \overline{([a \text{ NOR } a] \;\text{ NOR }\; [b \text{ NOR b}]) \;\text{ NOR }\; [c \text{ NOR c}])} \\ &= ([a \text{ NOR } a] \;\text{ NOR }\; [b \text{ NOR b}]) \;\text{ NOR }\; [c \text{ NOR c}]) \text{ NOR } ([a \text{ NOR } a] \;\text{ NOR }\; [b \text{ NOR b}]) \;\text{ NOR }\; [c \text{ NOR c}]). \end{aligned}$$

In this case the NAND implementation would clearly be preferred!

Practice questions

1. Convert the following truth table into a Boolean algebra expression and then simplify.

a	b	x
0	0	0
1	0	1
0	1	1
1	1	0

2. Convert the following truth table into a Boolean algebra expression and then simplify.

a	b	x
0	0	0
1	0	1
0	1	1
1	1	1

3. Convert the following truth table into a Boolean algebra expression and then simplify.

a	b	c	x
1	X	X	0
0	0	1	1
0	1	0	1
0	1	1	1
X	0	0	0

4. Convert the following truth table into a Boolean algebra expression and then simplify.

a	b	c	d	x
1	1	X	X	0
0	0	1	0	1
0	0	0	1	1
0	0	0	0	0

5. Simplify the following expression: $Q = A + B(A + C) + AC$.

6. Prove the identity $A + AB = A$. (Hint: Factorise out A on the LHS.)

7. Prove the identity $A + \overline{A}B = A + B$. (Hint: Expand $A \rightarrow A + AB$ on the LHS.)

8. Implement $y = ab$ using only NOR and NOT gates.

Answers

1. $x = a\overline{b} + \overline{a}b$

2. $x = a + b$

3. $x = \overline{a}\,(b + c)$

4. $x = \overline{a}\,\overline{b}\,(c\,\overline{d} + \overline{c}\,d)$

5. $Q = A + BC$

6. $LHS = A + AB = A(1 + B) = A = RHS$

7. $LHS = A + \overline{A}B = A + AB + \overline{A}B = A + (A + \overline{A})B = A + B = RHS$

8. $y = \overline{\overline{a} + \overline{b}} = \overline{a}\,\mathrm{NOR}\,\overline{b}$.