Circuit Simplification using Boolean Algebra

Aims

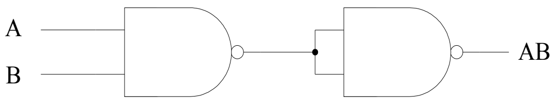
- to show that AND, OR and NOT gates can be constructed from NAND or NOR gates;
- to show why circuit simplification is necessary;
- to introduce the boolean identities used in circuit simplification;
- to apply the boolean identities to boolean expressions in order to simplify them;
- to show how to represent the result of a simplification as a logic circuit.

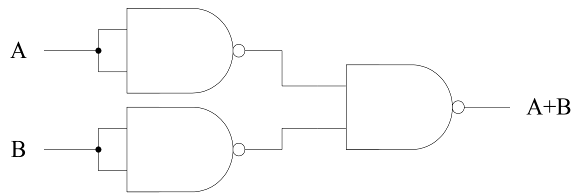
5.1 Complete Circuits

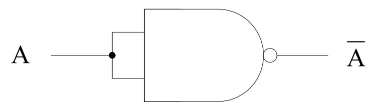
- Boolean expressions are usually implemented using NOT, OR and AND gates. It is often convenient to implement circuits using only a single type of gate.

- Both NAND and NOR gates are said to be *complete*, because any boolean expression can be implemented using either of them. No other gates have this property, which is why they are often the preferred building blocks of circuits. However, circuits constructed using only NAND and NOR gates are not optimal in the sense of the minimum number of gates.

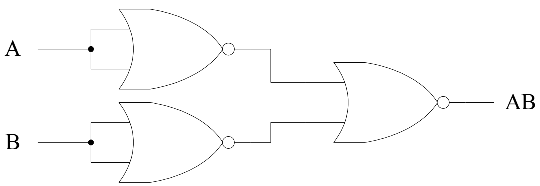
NAND Gates

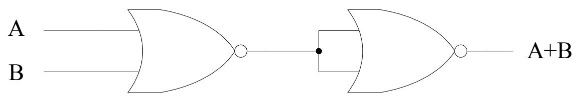


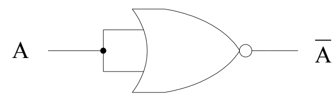




NOR Gates







5.2 Circuit Equivalence

- Circuit designers strive to reduce the number of gates in their products for a number of reasons:

  1. Cost
  2. Size
  3. Power consumption

- Equally, circuit designers will attempt to use simpler gates where possible, such as two-input rather than four-input gates. Boolean algebra can be used to simplify logic circuits, with the aim being to produce an equivalent circuit that uses fewer or simpler gates.

- There are a number of *laws* of boolean algebra that can be used to help us in the search for circuit equivalence. Often these laws are the same as those in ordinary algebra (e.g. distributive, commutative, associative etc.).

5.3 Boolean Algebra

- When designing a circuit the designer generally starts with a boolean expression and then applies the laws of boolean algebra in order to find a simpler but equivalent expression. From this simplified expression, a circuit can be constructed.

- In order to use such an approach, some *identities* are needed from boolean algebra.

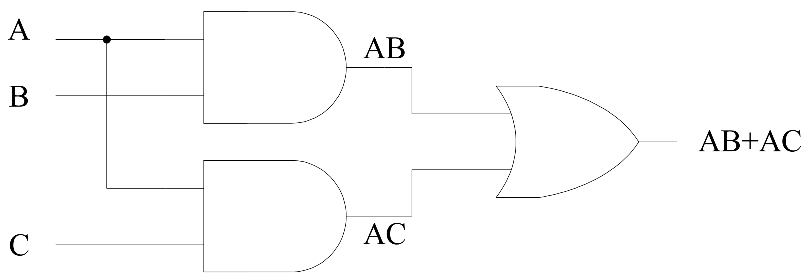
Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
	$\overline{\overline{A}} = A$	
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$ $A + \overline{A}B = A + B$
De-Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}.\overline{B}$

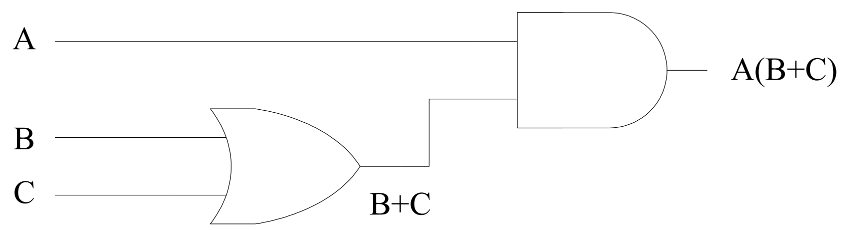
Example 1

- The OR form of the distributive law states that A(B+C) = AB+AC. This can be proven by producing truth tables for the expression on each side of the equals sign.

A	B	C	AB	AC	AB+AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

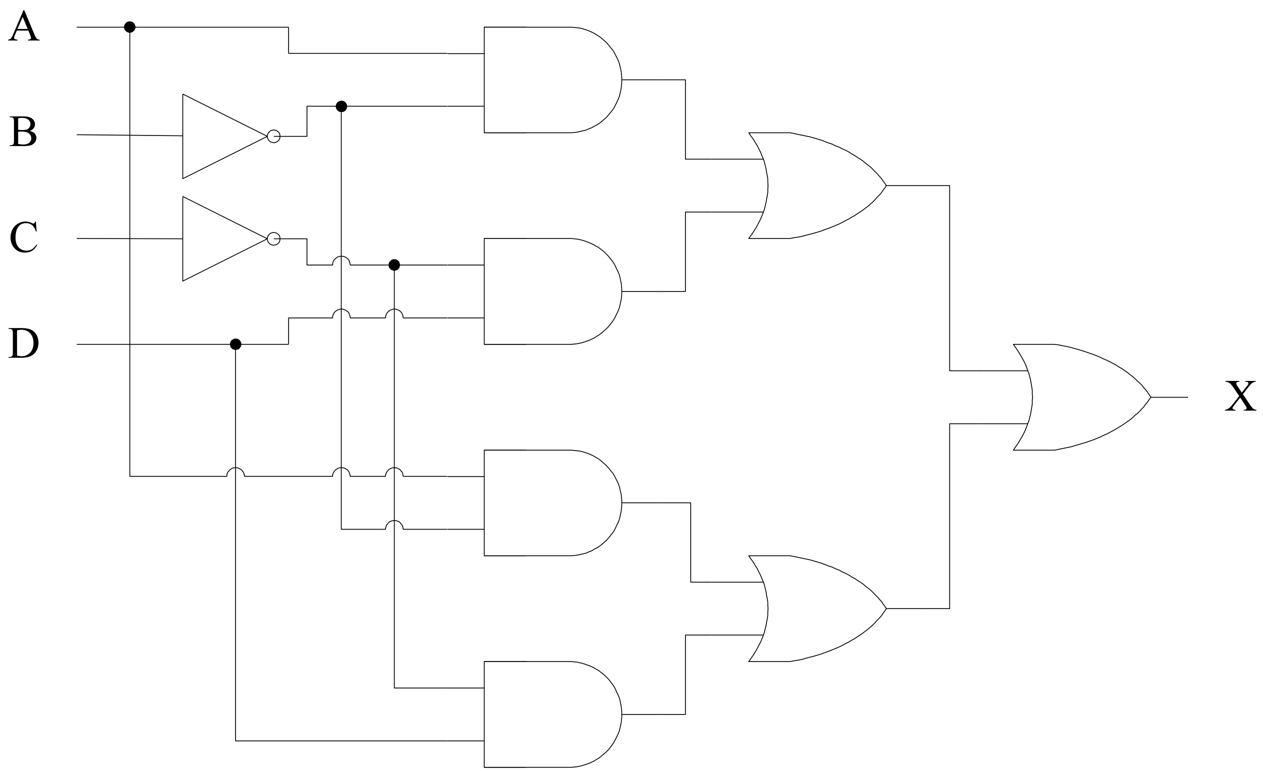
A	B	C	B+C	A(B+C)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1





- Two circuits are equivalent if and only if they have the same output for all possible inputs.

Example 2



- The boolean algebra expression for the circuit is:

$$X = A\overline{B} + \overline{C}D + A\overline{B} + \overline{C}D$$

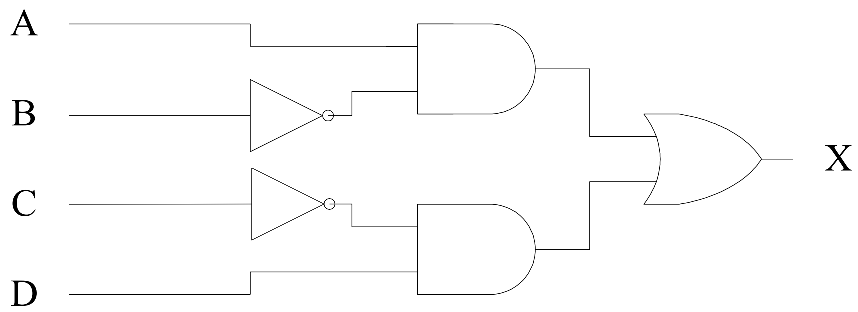
- The OR form of the idempotent law states that A+A = A. This can be proven by constructing a truth table.

A	A	A+A
0	0	0
1	1	1

- This law can be used to simplify the original expression:

$$X = \underbrace{A\overline{B} + \overline{C}D}_{A} + \underbrace{A\overline{B} + \overline{C}D}_{A}$$

$$X = A\overline{B} + \overline{C}D$$



- Therefore the original circuit has been reduced from nine gates to five gates.

Example 3

- Simplify the following expression:

$$X = ABC + AB\overline{C} + \overline{A}C$$

$$X = AB\left(C + \overline{C}\right) + \overline{A}C \qquad \text{(distributive, OR)}$$

$$X = AB.1 + \overline{A}C \qquad \text{(inverse, OR)}$$

$$X = AB + \overline{A}C \qquad \text{(identity, AND)}$$

Example 4

- Simplify the following expression:

$$Z = AC + ABC + A\overline{B}C$$

$$= AC + AC\left(B + \overline{B}\right) \qquad \text{(distributive, OR)}$$

$$= AC + AC.1 \qquad \text{(inverse, OR)}$$

$$= AC + AC \qquad \text{(identity, AND)}$$

$$= AC \qquad \text{(idempotence, OR)}$$

Example 5

- Simplify the following expression:

$$Z = \overline{A}B + AB\overline{C} + ABC$$

$$= \overline{A}B + AB\left(C + \overline{C}\right) \qquad \text{(distributive, OR)}$$

$$= \overline{A}B + AB.1 \qquad \text{(inverse, OR)}$$

$$= \overline{A}B + AB \qquad \text{(identity, AND)}$$

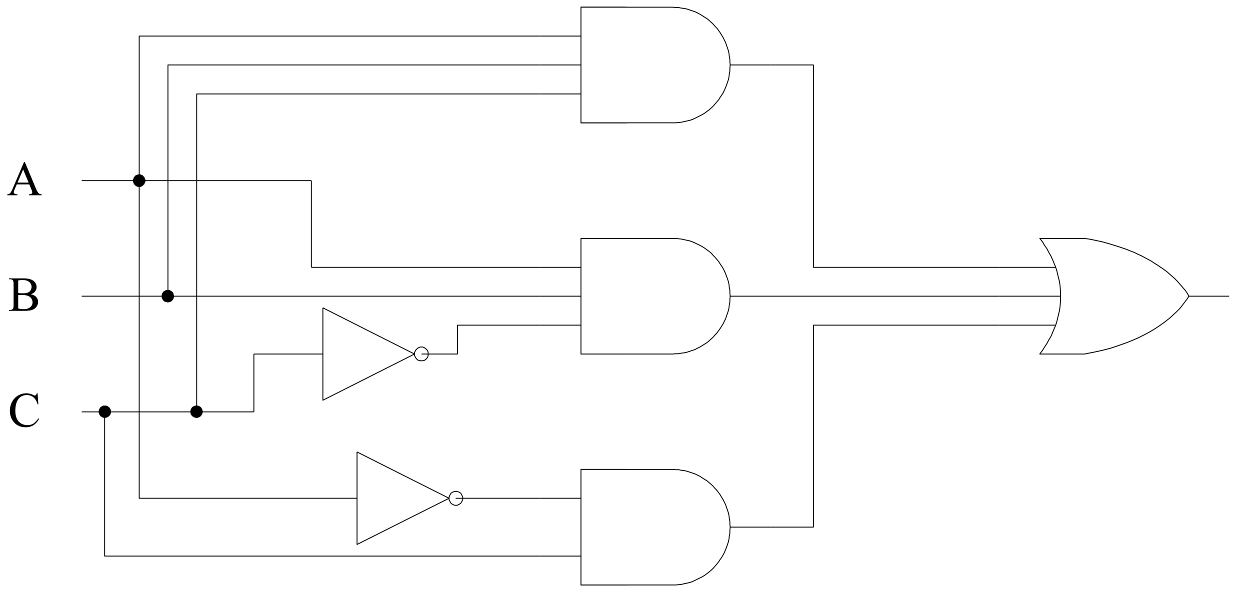
$$= B\left(\overline{A} + A\right) \qquad \text{(distributive, OR)}$$

$$= B.1 \qquad \text{(inverse, OR)}$$

$$= B \qquad \text{(identity, AND)}$$

6.1 Boolean Algebra Simplifications

Example 1

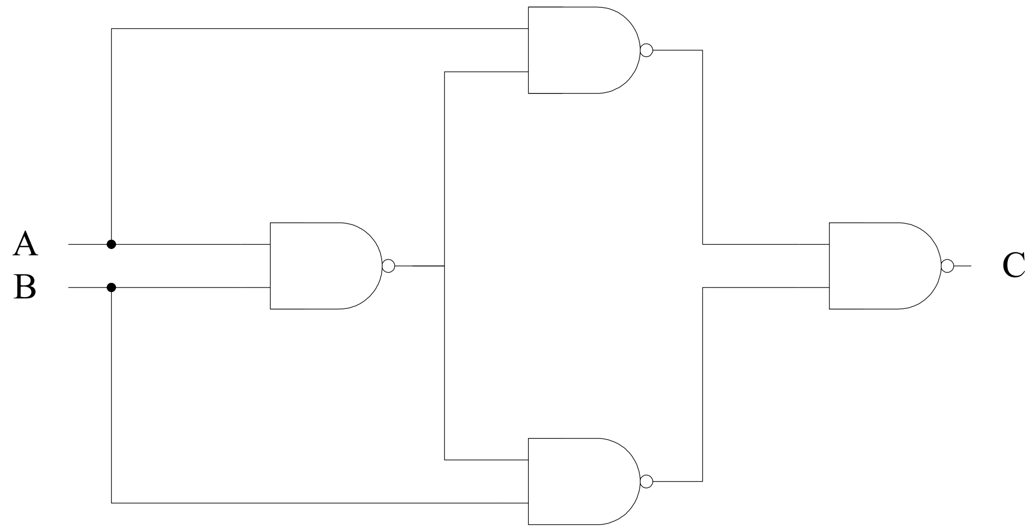


$$ABC + AB\overline{C} + \overline{A}C$$

$AB\left(C + \overline{C}\right) + \overline{A}C$ (distributive, OR)

$AB + \overline{A}C$ (inverse, OR, identity, AND)

Example 2



$$= \overline{\overline{A.\overline{AB}}.\overline{B.\overline{AB}}}$$

$$= \overline{\overline{A\left(\overline{A}+\overline{B}\right)}.\overline{B\left(\overline{A}+\overline{B}\right)}} \qquad \text{(de-morgan's, AND)}$$

$$= \overline{\left(\overline{A\overline{A}+A\overline{B}}\right).\left(\overline{B\overline{A}+B\overline{B}}\right)} \qquad \text{(distributive, OR)}$$

$$= \overline{\left(\overline{0+A\overline{B}}\right).\left(\overline{B\overline{A}+0}\right)} \qquad \text{(inverse, AND)}$$

$$= \overline{\overline{A\overline{B}}.\overline{B\overline{A}}} \qquad \text{(identity, OR)}$$

$$= \overline{\overline{A\overline{B}}}+\overline{\overline{B\overline{A}}} \qquad \text{(de-morgan's, AND)}$$

$$= A\overline{B}+B\overline{A} \qquad \text{(inverse)}$$

Example 3

- Simplify the following expression.

$$X = A\overline{B}C + \overline{A}.\overline{B}C + \overline{A}B\overline{C} + AB$$

$$= \overline{B}C\left(A + \overline{A}\right) + B\left(\overline{A}.\overline{C} + A\right) \qquad \text{(distributive, OR)}$$

$$= \overline{B}C1 + B\left(\overline{A}.\overline{C} + A\right) \qquad \text{(inverse, OR)}$$

$$= \overline{B}C + B\left(\overline{A}.\overline{C} + A\right) \qquad \text{(identity, AND)}$$

$$= \overline{B}C + B\left(\overline{C} + A\right) \qquad \text{(absorption, OR)}$$

$$= \overline{B}C + B\overline{C} + BA \qquad \text{(distributive, OR)}$$

Example 4

- Simplify the following expression.

$$
\begin{aligned}
Z &= \overline{\overline{AB}C}.\overline{B}.\overline{C}D \\
&= \left(\overline{\overline{AB}} + \overline{C}\right).\overline{B}.\overline{C}D && \text{(de-morgan, AND)} \\
&= \left(AB + \overline{C}\right).\overline{B}.\overline{C}D && \text{(inverse)} \\
&= AB\overline{B}.\overline{C}D + \overline{C}.\overline{B}.\overline{C}D && \text{(distributive, OR)} \\
&= AB\overline{B}.\overline{C}D + \overline{B}.\overline{C}.\overline{C}D && \text{(commutative, AND)} \\
&= AB\overline{B}.\overline{C}D + \overline{B}.\overline{C}D && \text{(idempotent, AND)} \\
&= A0\overline{C}D + \overline{B}.\overline{C}D && \text{(inverse, AND)} \\
&= 0A\overline{C}D + \overline{B}.\overline{C}D && \text{(commutative, AND)} \\
&= 0\overline{C}D + \overline{B}.\overline{C}D && \text{(null, AND)} \\
&= 0D + \overline{B}.\overline{C}D && \text{(null, AND)} \\
&= 0 + \overline{B}.\overline{C}D && \text{(null, AND)} \\
&= \overline{B}.\overline{C}D && \text{(identity, OR)}
\end{aligned}
$$

Name	**AND form**	**OR form**
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
	$\overline{\overline{A}} = A$	
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$ $A + \overline{A}B = A + B$
De-Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}.\overline{B}$