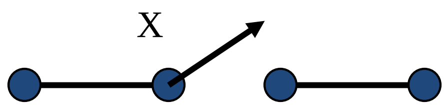
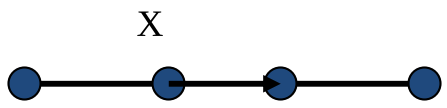
Preliminaries

- Before we get into the design of computers and their components, we must understand the basics of Boolean algebra.
- The knowledge of Boolean algebra will enable you to specify and optimize your computer design(s)

Switching Theory



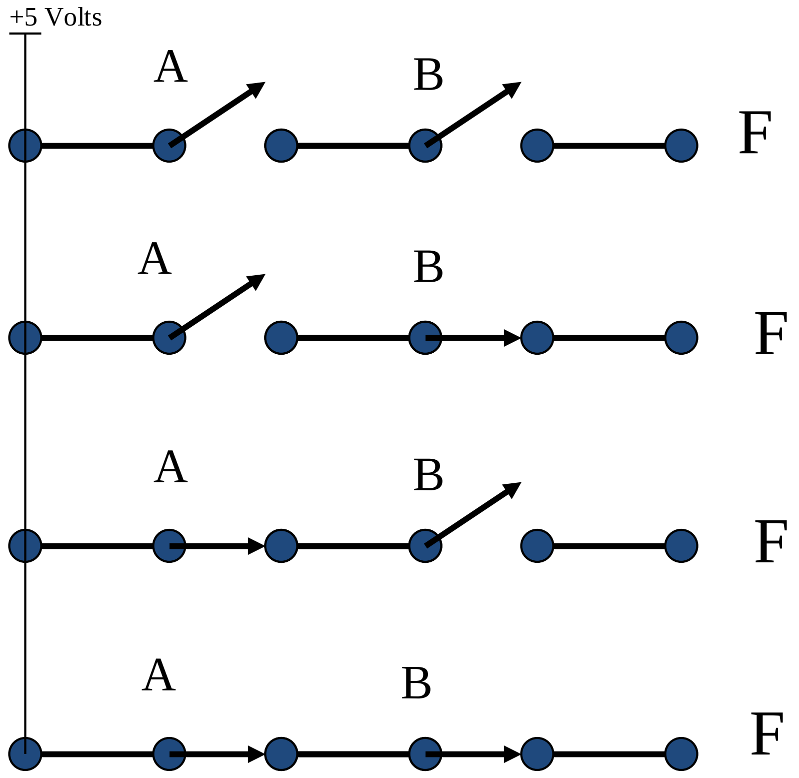
X = 0 → Switch open



X = 1 → Switch closed

Switching Theory

(The AND Gate)



A = 0, B = 0, F = 0

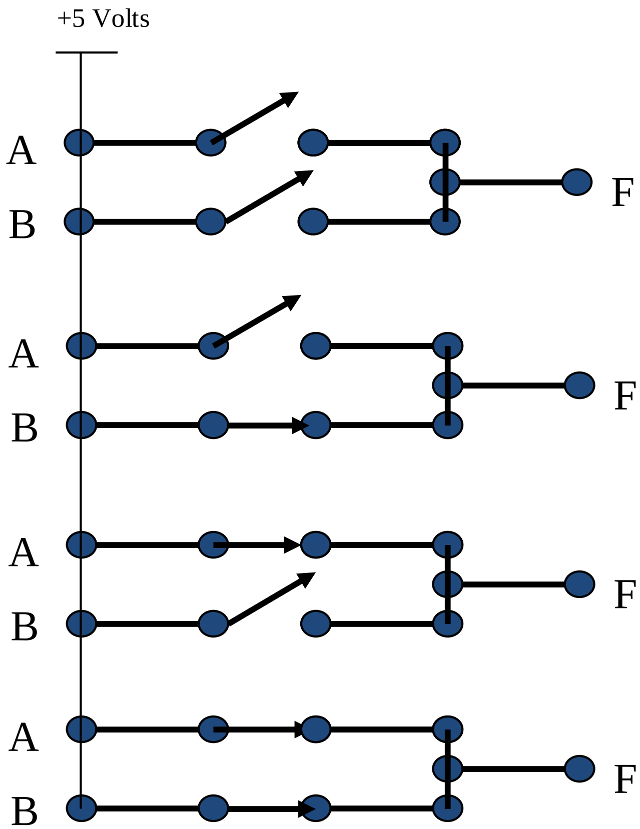
A = 0, B = 1, F = 0

A = 1, B = 0, F = 0

A = 1, B = 1, F = 1

Switching Theory

(The OR Gate)



A = 0, B = 0, F = 0

A = 0, B = 1, F = 1

A = 1, B = 0, F = 1

A = 1, B = 1, F = 1

Symbolic Representations of Logic Functions

- There are more intuitive ways to represent logic functions than by the use of switches
- Each of the basic logic functions have symbolic representations that are universally understood
- These basic representations are referred to as logic gates

Boolean Functions

- $+$ → The OR function
- $'$ → The NOT function
- $\bullet$ → The AND function
- $\oplus$ →The XOR function

Logic Gates

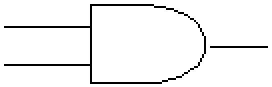
- Each Boolean function has a corresponding symbolic representation.
- These symbolic representations are commonly referred to as logic gates

Truth Tables

- Truth tables specify values of a Boolean expression for all possible combinations of variables in the expression
- Each of the basic logic gates, for instance has a set of inputs. The respective truth tables for each gate shows the behavior of the gate's output for all possible input combinations.

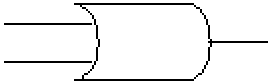
Truth Tables for Logic Gates

AND



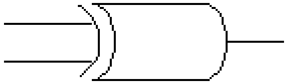
A	B	A • B
0	0	0
0	1	0
1	0	0
1	1	1

OR



A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

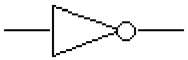
XOR



A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Truth Tables for Logic Gates

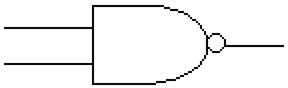
NOT



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

Truth Tables for Logic Gates

NAND

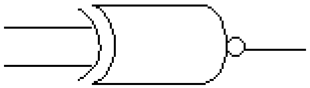


A	B	$\overline{(A \bullet B)}$
0	0	1
0	1	1
1	0	1
1	1	0

NOR

A	B	$\overline{(A + B)}$
0	0	1
0	1	0
1	0	0
1	1	0

XNOR



A	B	$\overline{(A \oplus B)}$
0	0	1
0	1	0
1	0	0
1	1	1

Exercise I

Draw the logic gates and generate the truth tables for the following expressions:

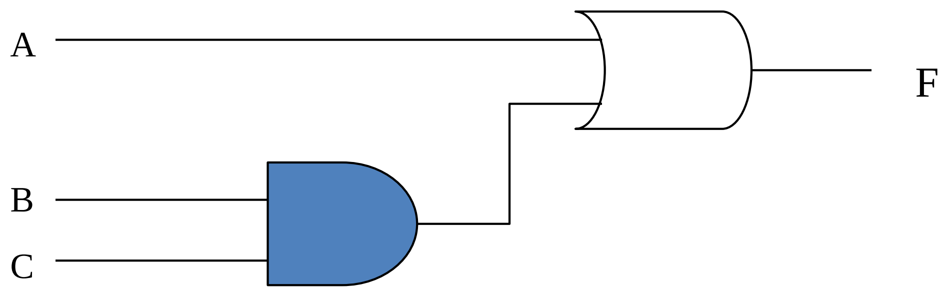
1. F = A + BC

2. F = AB’ + A’B

3. F = A+B+AC

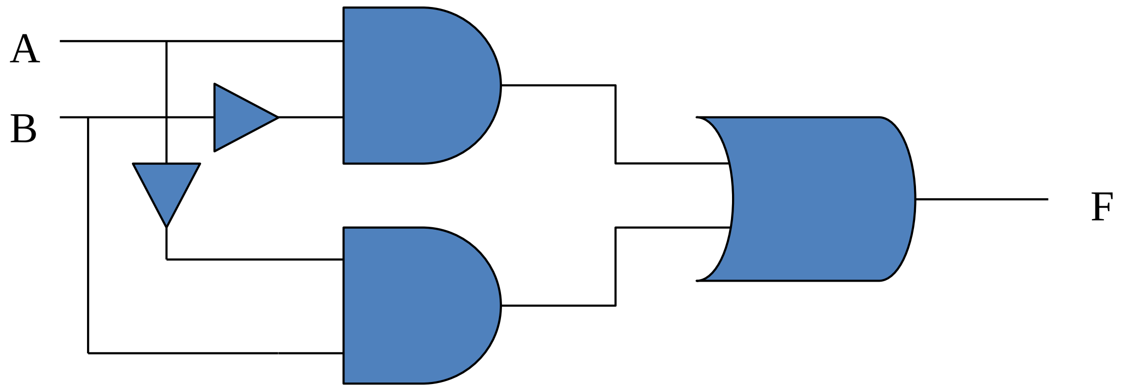
Exercise 1: Solution 1

$$F = A + BC$$



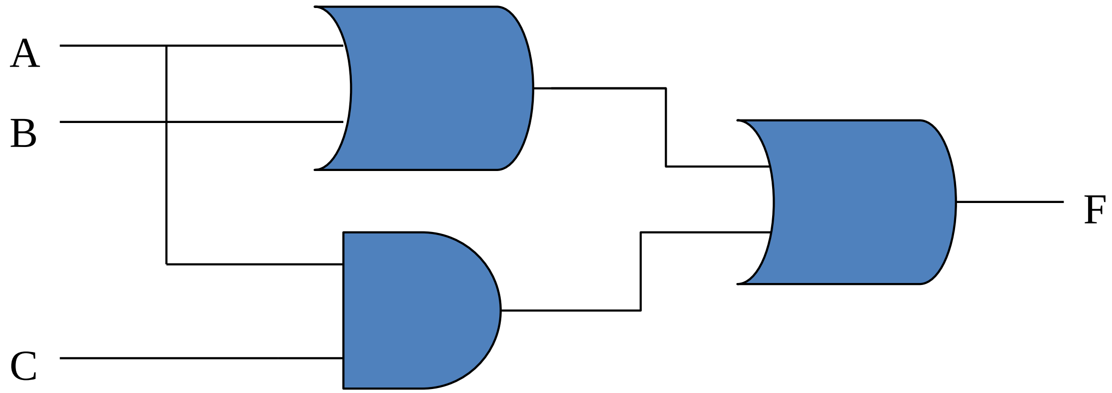
Exercise 1: Solution 2

$$F = AB' + A'B$$



Exercise 1: Solution 3

F = A+B+AC



Basic Theorems of Boolean Algebra

$X + 0 = X$	$X . 1 = X$
$X + 1 = 1$	$X . 0 = 0$
Idempotent Laws	
$X + X = X$	$X . X = X$
Involution Law	
$(X')' = X$	
Laws of Complementarity	
$X + X' = 1$	$X . X' = 0$

Commutative, Associative and Distributive Laws

A lot of the laws of ordinary algebra are valid for Boolean algebra:

Commutative Law: $A+B = B+A$ $AB = BA$

Associative Law: $(A+B)+C = A+(B+C)$ $(AB)C = A(BC)$

Distributive Law: $A(B+C) = AB + AC$

Second Distributive Law: $A + BC = (A+B)(A+C)$

Note: This law does not hold for ordinary algebra.

Simplification Theorems

These theorems are very helpful in simplifying Boolean expressions:

1. AB +AB’ = A
2. (A+B)(A+B’) = A
3. A + AB = A
4. A(A+B) = A
5. (A+B’)B = AB
6. AB’+B = A+B

Simplification Theorems
Proof of Theorem 1

1. $AB + AB' = A$

   $A(B + B') = A(1) = A$

Simplification Theorems Proof of Theorem 2

2. $(A+B)(A+B') = A$

   $AA + AB' + AB + BB'$

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

   $A + A = A$

Simplification Theorems
Proof of Theorem 3

3. $A + AB = A$
   $A(1 + B)$
   $A(1) = A$

Simplification Theorems
Proof of Theorem 4

4. $A(A+B) = A$

   $AA + AB$

   $A + AB$

   $A(1+B)$

   $A(1) = A$

Simplification Theorems Proof of Theorem 5

5. $(A+B')B = AB$

   $AB + BB'$

   $AB + 0 = AB$

Simplification Theorems Proof of Theorem 6

6. $AB' + B = A + B$

$B + AB' = A + B$

$(B+A)(B+B')$

$(B+A)(1) = B+A = A+B$