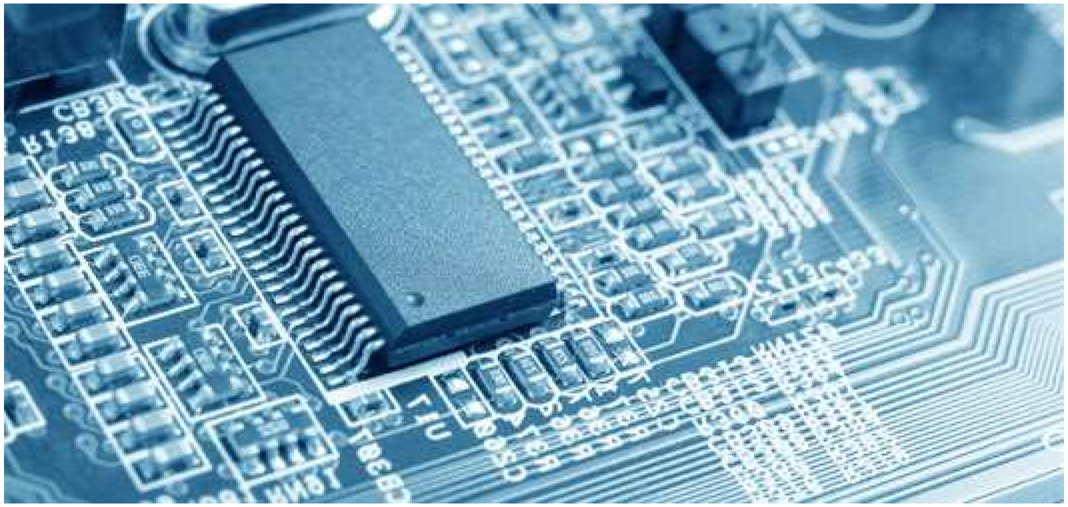
NAZARBAYEV UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY



ROBT206 – Microcontrollers with Lab

Lecture 4 – Boolean Algebra

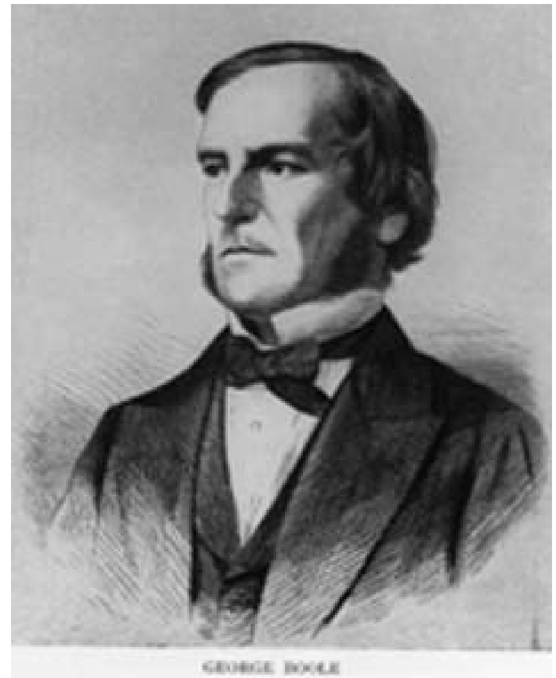
18 January, 2018

Topics

Today's Topics
Binary Logic and Gates
Boolean Algebra

George Boole, 1815-1864

- Born to working class parents
- Taught himself mathematics and joined the faculty of Queen's College in Ireland.
- Wrote *An Investigation of the Laws of Thought* (1854)
- Introduced binary variables
- Introduced the three fundamental logic operations: AND, OR, and NOT.



Scanned at the American Institute of Physics

Binary Logic and Gates

- **Binary variables** take on one of two values.
- **Logical operators** operate on binary values and binary variables.
- Basic logical operators are the **logic functions** AND, OR and NOT.
- **Logic gates** implement logic functions.
- **Boolean Algebra**: a useful mathematical system for specifying and transforming logic functions.
- We study Boolean algebra as a foundation for designing and analyzing digital systems!

Binary Variables

- Recall that the two binary values have different names:
  - True/False
  - On/Off
  - Yes/No
  - 1/0
- We use 1 and 0 to denote the two values.

Logical Operations

- The three basic logical operations are:
  - AND
  - OR
  - NOT
- AND is denoted by a dot (·).
- OR is denoted by a plus (+).
- NOT is denoted by an overbar (¯), a single quote mark (') after, or (~) before the variable.

Notation Examples

- Examples:
  - $\mathbf{Y = A \cdot B}$ is read "Y is equal to A AND B."
  - $\mathbf{z = x + y}$ is read "z is equal to x OR y."
  - $\mathbf{X = \overline{A}}$ is read "X is equal to NOT A."

- Note: The statement:

  1 + 1 = 2 (read "one plus one equals two")

  is not the same as

  1 + 1 = 1 (read "1 or 1 equals 1").

Operator Definitions

- Operations are defined on the values "0" and "1" for each operator:

AND	OR	NOT
$0 \cdot 0 = 0$	$0 + 0 = 0$	$\overline{0} = 1$
$0 \cdot 1 = 0$	$0 + 1 = 1$	$\overline{1} = 0$
$1 \cdot 0 = 0$	$1 + 0 = 1$	
$1 \cdot 1 = 1$	$1 + 1 = 1$	

Truth Tables

- *Truth table* – a tabular listing of the values of a function for all possible combinations of values on its arguments

- Example: Truth tables for the basic logic operations:

AND		
X	Y	$Z = X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

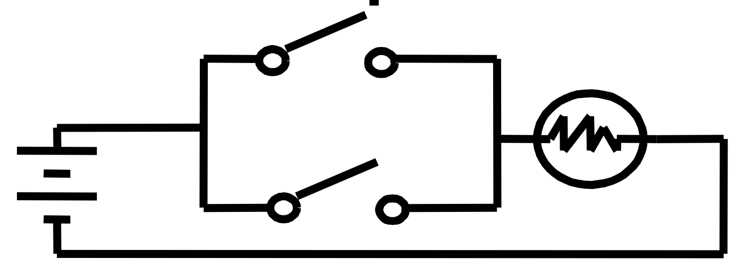
OR		
X	Y	$Z = X+Y$
0	0	0
0	1	1
1	0	1
1	1	1

NOT	
X	$Z = \overline{X}$
0	1
1	0

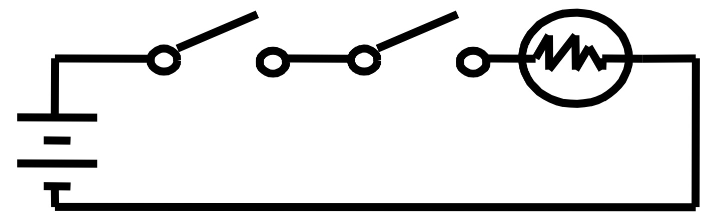
Logic Function Implementation

- Using Switches
  - For inputs:
    - logic 1 is switch closed
    - logic 0 is switch open
  - For outputs:
    - logic 1 is light on
    - logic 0 is light off.
  - NOT uses a switch such that:
    - logic 1 is switch open
    - logic 0 is switch closed

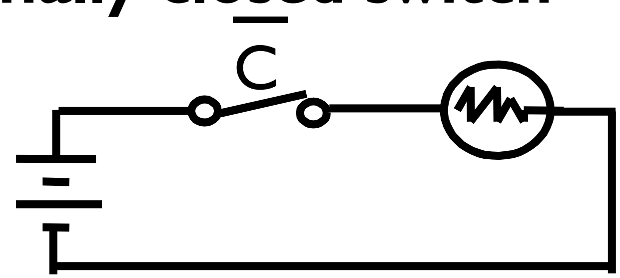
Switches in parallel => OR



Switches in series => AND

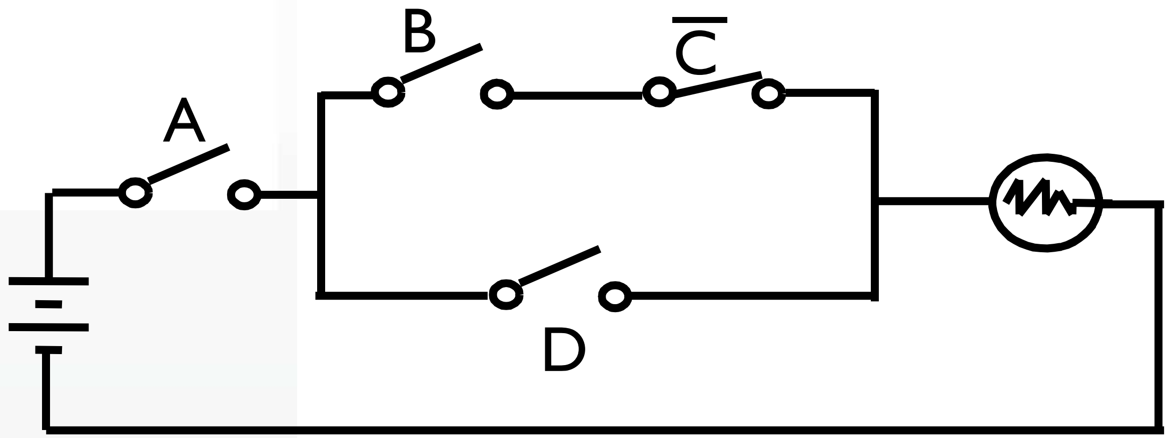


Normally-closed switch => NOT



Logic Function Implementation

- Example: Logic Using Switches



- Light is on (L = 1) for

  L(A, B, C, D) =

  and off (L = 0), otherwise.
- Useful model for relay circuits and for CMOS gate circuits, the foundation of current digital logic technology

Logic Gates

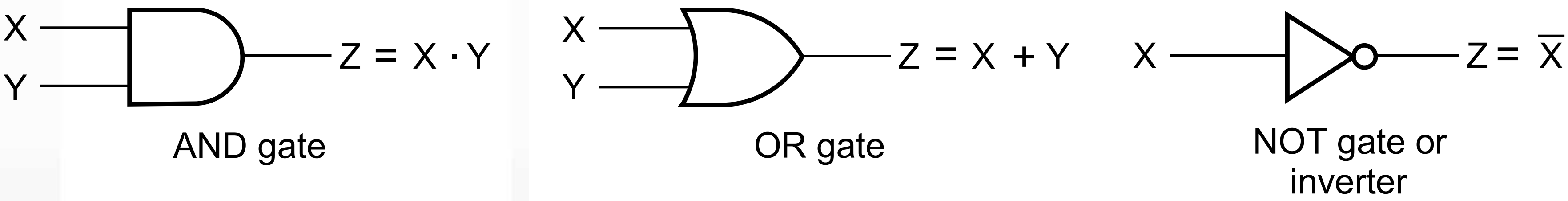
- In the earliest computers, switches were opened and closed by magnetic fields produced by energizing coils in ***relays***. The switches in turn opened and closed the current paths.

- Later, ***vacuum tubes*** that open and close current paths electronically replaced relays.

- Today, ***transistors*** are used as electronic switches that open and close current paths.

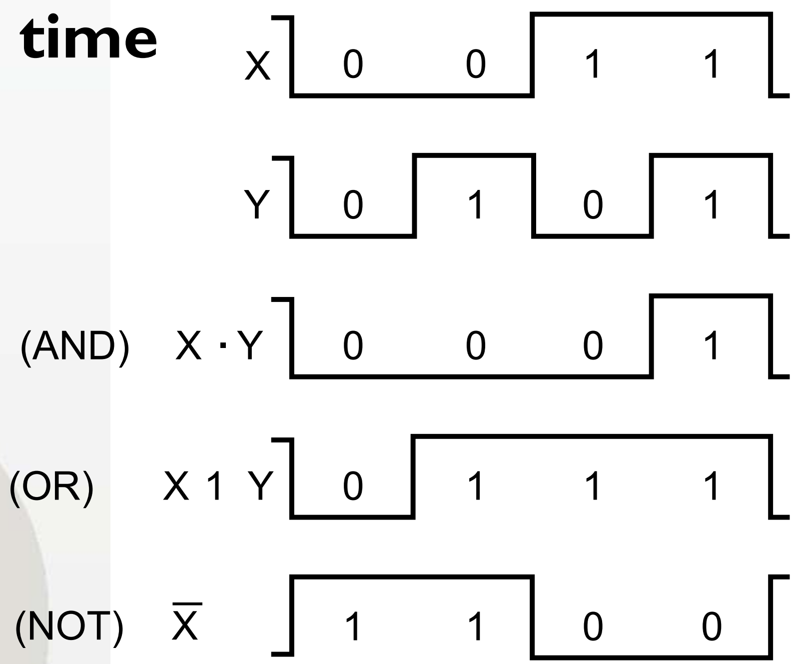
Logic Gate Symbols and Behavior

- **Logic gates have special symbols:**



(a) Graphic symbols

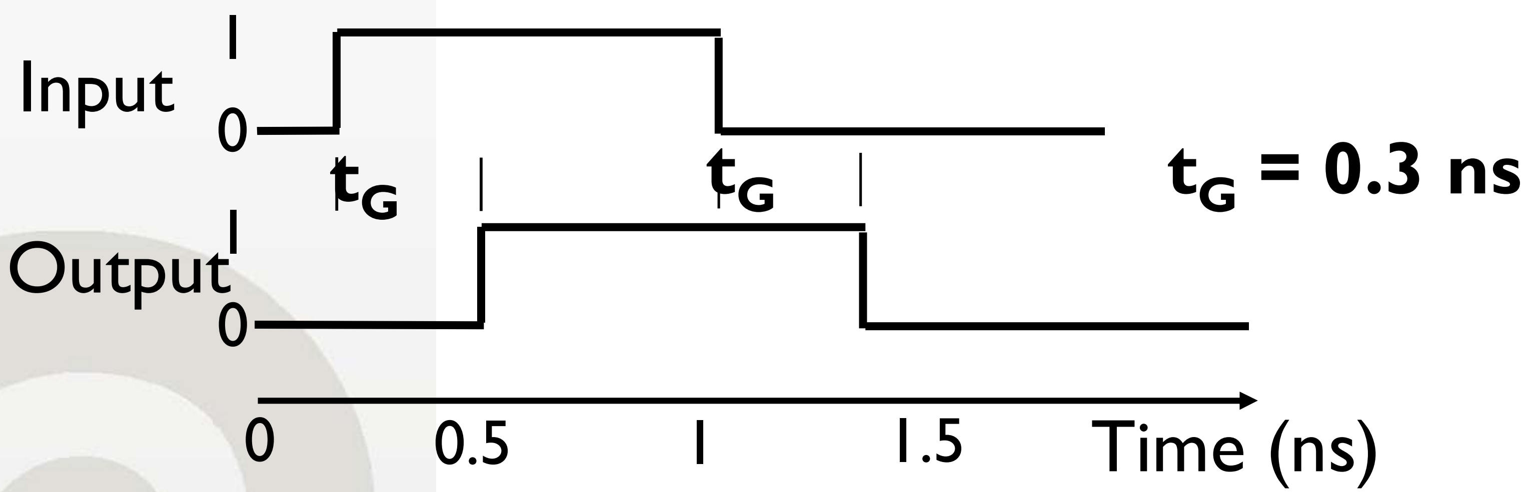
- **And waveform behavior in time as follows:**



(b) Timing diagram

Gate Delay

- In actual physical gates, if one or more input changes causes the output to change, the output change does not occur instantaneously.
- The delay between an input change(s) and the resulting output change is the ***gate delay*** denoted by t_G:



Logic Diagrams and Expressions

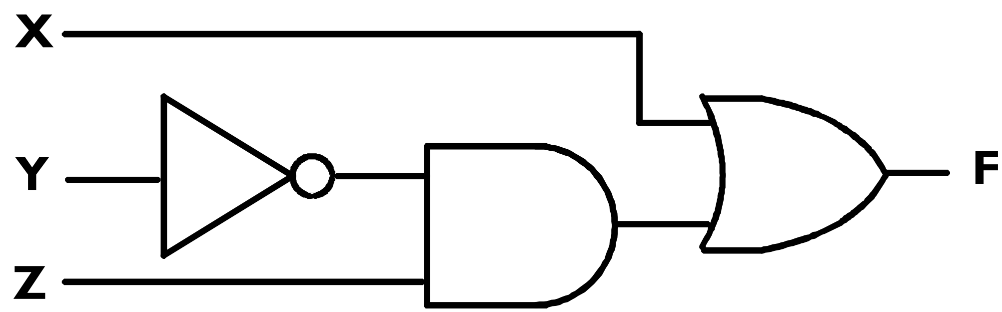
Truth Table

XYZ	$F = X + \overline{Y} \cdot Z$
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

Equation

$$F = X + \overline{Y}\, Z$$

Logic Diagram



- Boolean equations, truth tables and logic diagrams describe the same function!
- Truth tables are unique; expressions and logic diagrams are not. This gives flexibility in implementing functions.

Boolean Algebra

- An algebraic structure defined on a set of binary variables, together with three binary operators (denoted +, $\cdot$ and $\overline{}$) that satisfies the following basic identities:

1.	$X + 0 = X$	2.	$X \cdot 1 = X$	
3.	$X + 1 = 1$	4.	$X \cdot 0 = 0$	
5.	$X + X = X$	6.	$X \cdot X = X$	
7.	$X + \overline{X} = 1$	8.	$X \cdot \overline{X} = 0$	
9.	$\overline{\overline{X}} = X$			
10.	$X + Y = Y + X$	11.	$XY = YX$	**Commutative**
12.	$(X + Y) + Z = X + (Y + Z)$	13.	$(XY)Z = X(YZ)$	**Associative**
14.	$X(Y + Z) = XY + XZ$	15.	$X + YZ = (X + Y)(X + Z)$	**Distributive**
16.	$\overline{X + Y} = \overline{X} \cdot \overline{Y}$	17.	$\overline{X \cdot Y} = \overline{X} + \overline{Y}$	**DeMorgan's**

Boolean Operator Precedence

- The order of evaluation in a Boolean expression is:

1. Parentheses
2. NOT
3. AND
4. OR

Example 1: Boolean Algebraic Proof

- $A + A \cdot B = A$ (Absorption Theorem)

Proof Steps Justification using Identities

$A + A \cdot B$

$= A \cdot 1 + A \cdot B$ $X = X \cdot 1$

$= A \cdot (1 + B)$ $X \cdot Y + X \cdot Z = X \cdot (Y + Z)$ (Distributive Law)

$= A \cdot 1$ $1 + X = 1$

$= A$ $X \cdot 1 = X$

- **Our primary reason for doing proofs is to learn:**
  - Careful and efficient use of the identities and theorems of Boolean algebra, and
  - How to choose the appropriate identity or theorem to apply to make forward progress, irrespective of the application.

Example 2: Boolean Algebraic Proofs

- $AB + \overline{A}C + BC = AB + \overline{A}C$ (Consensus Theorem)

Proof Steps	Justification (identity or theorem)
$AB + \overline{A}C + BC$	
$= AB + \overline{A}C + 1 \cdot BC$	?
$= AB + \overline{A}C + (A + \overline{A}) \cdot BC$	?
$=$	

Example 3: Boolean Algebraic Proofs

- $(\overline{X+Y})Z + X\overline{Y} = \overline{Y}(X+Z)$

Proof Steps

$= (\overline{X+Y})Z + X\overline{Y}$

Proof of DeMorgan's Laws

$$\overline{x + y} = \overline{x} \cdot \overline{y} \qquad \overline{x \cdot y} = \overline{x} + \overline{y}$$

Boolean Function Evaluation

$$F1 = xy\bar{z}$$
$$F2 = x + \bar{y}z$$
$$F3 = \bar{x}\bar{y}\bar{z} + \bar{x}\,y\,z + x\bar{y}$$
$$F4 = x\bar{y} + \bar{x}\,z$$

x	y	z	F1	F2	F3	F4
0	0	0	0	0		
0	0	1	0	1		
0	1	0	0	0		
0	1	1	0	0		
1	0	0	0	1		
1	0	1	0	1		
1	1	0	1	1		
1	1	1	0	1		

Expression Simplification

- An application of Boolean algebra
- Simplify to contain the smallest number of **<u>literals</u>** (complemented and uncomplemented variables):

$$AB + \overline{A}CD + \overline{A}BD + \overline{A}C\overline{D} + ABCD$$

$$= AB + ABCD + \overline{A}\,C\,D + \overline{A}\,C\,\overline{D} + \overline{A}\,B\,D$$

$$= AB + AB(CD) + \overline{A}\,C\,(D + \overline{D}) + \overline{A}\,B\,D$$

$$= AB + \overline{A}\,C + \overline{A}\,B\,D = B(A + \overline{A}D) + \overline{A}C$$

$$= B\,(A + D) + \overline{A}\,C \quad \rightarrow \quad 5 \text{ literals}$$

Complementing Functions

- Use DeMorgan's Theorem to complement a function:
  1. Interchange AND and OR operators
  2. Complement each constant value and literal

- Example: Complement $F = \overline{x}\, y\, \overline{z} + x\, \overline{y}\, \overline{z}$

  $\overline{F} = (x + \overline{y} + z)(\overline{x} + y + z)$
- Example: Complement $G = (\overline{a} + bc)\overline{d} + e$

  $\overline{G} =$

Any Questions?