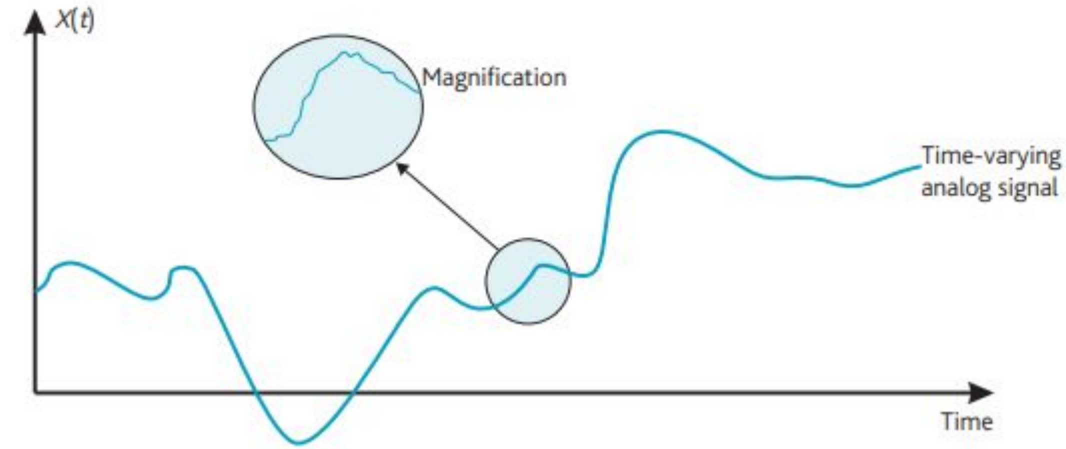
HNDIT1032 Computer and Network Systems

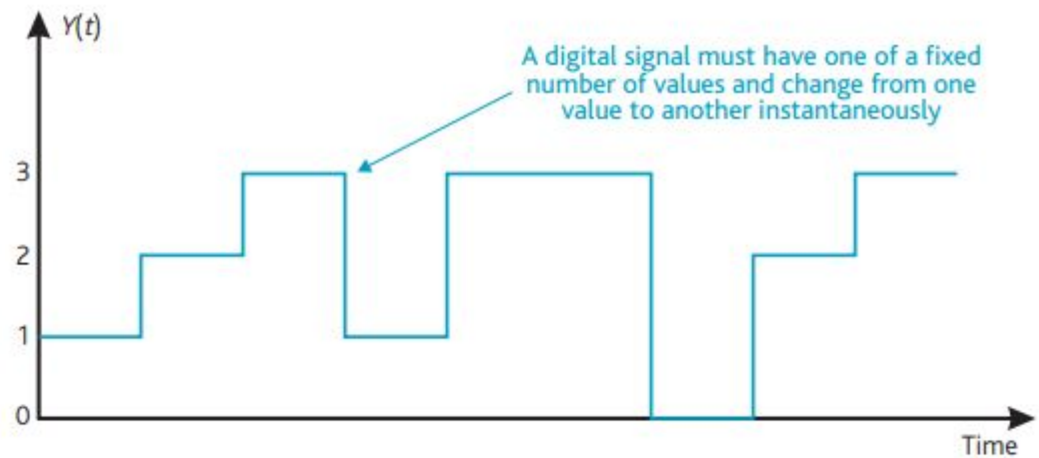
Week 03- Basic Logic Gates & Boolean Algebra

Analog & Digital System

- Analog variable can have any value between its maximum and minimum limits.
- Information inside a computer is represented in digital form.
- A digital variable is discrete in both value and in time.

Analog & Digital System





Logic Values

- Every logic input or output must assume one of two discrete states. You cannot have a state that is neither 1 nor 0.
- Each logic input or output can exist in only one state at any one time.
- Each logic state has an inverse or complement that is the opposite of its current state.

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.
- Variable identifier examples:
  - A, B, y, z, or X1 for now
  - RESET, START_IT, or ADD1 later

Binary Logic and Gates

- Digital circuits are hardware components (based on transistors) that manipulate binary information
- We model the transistor-based electronic circuits as logic gates.
  - Designer can ignore the internal electronics of a gate

Basic Logic Gates

- 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.

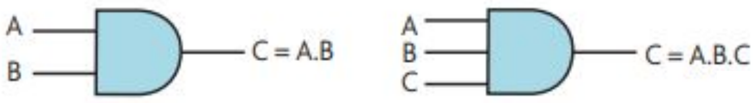
Truth Tables

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

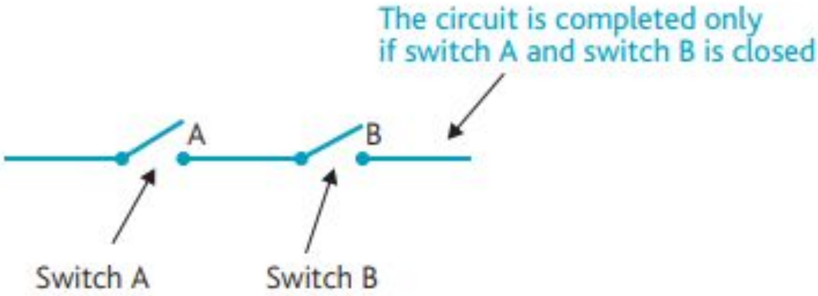
The AND Gate

- The AND gate is a circuit with two or more inputs and a single output.
- The output of an AND gate is true if and only if each of its inputs is also in a true state.
- Conversely, if one or more of the inputs to the AND gate is false, the output will also be false.

The AND Gate...



(a) Two-input AND gate (b) Three-input AND gate



Inputs		Output
A	B	$F = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

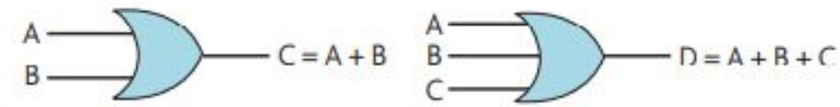
False because one or more inputs is false

True because both inputs are true

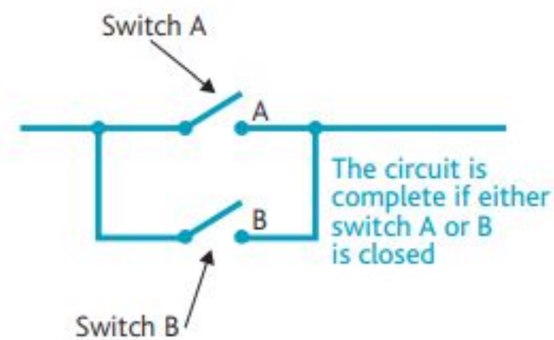
The OR Gate

- The output of an OR gate is true if any one (or more than one) of its inputs is true.
- The logical symbol for an OR operation is an addition sign, so that the logical operation A OR B is written as A +B.

The OR Gate…



(a) Two-input OR gate. (b) Three-input OR gate.



Inputs A	Inputs B	Output F=A+B
0	0	0
0	1	1
1	0	1
1	1	1

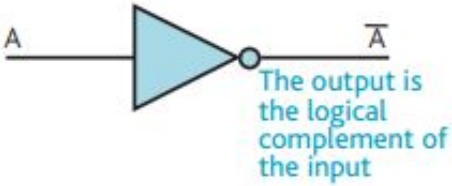
False because no input is true

True because at least one input is true

The Not Gate

- The NOT gate is also called an inverter or a complemented and is a two-terminal device with a single input and a single output.
- If the input of an inverter is X, its output is NOT X which is written X – or X’ .

The Not Gate

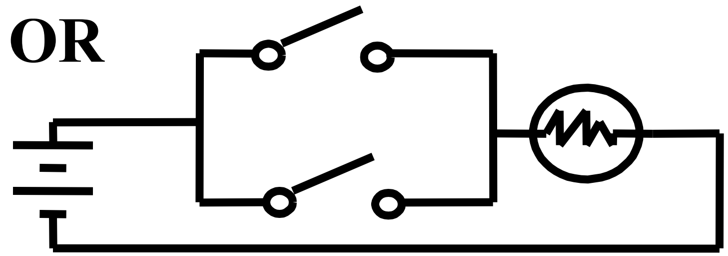


Input A	Output $F=\overline{A}$
0	1
1	0

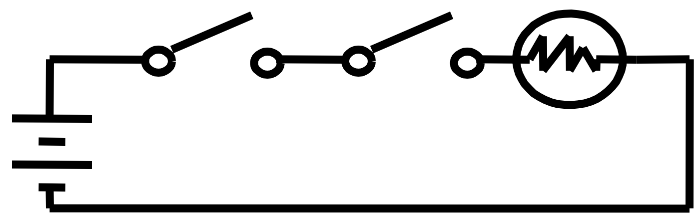
Logic Function Implementation

- 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

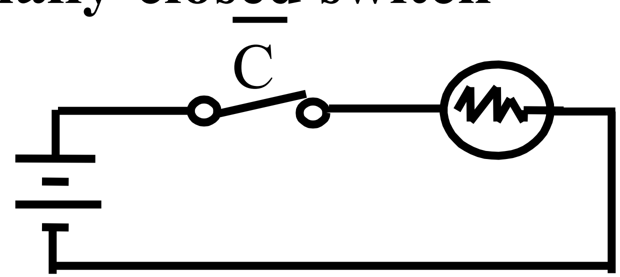
Switches in parallel => OR



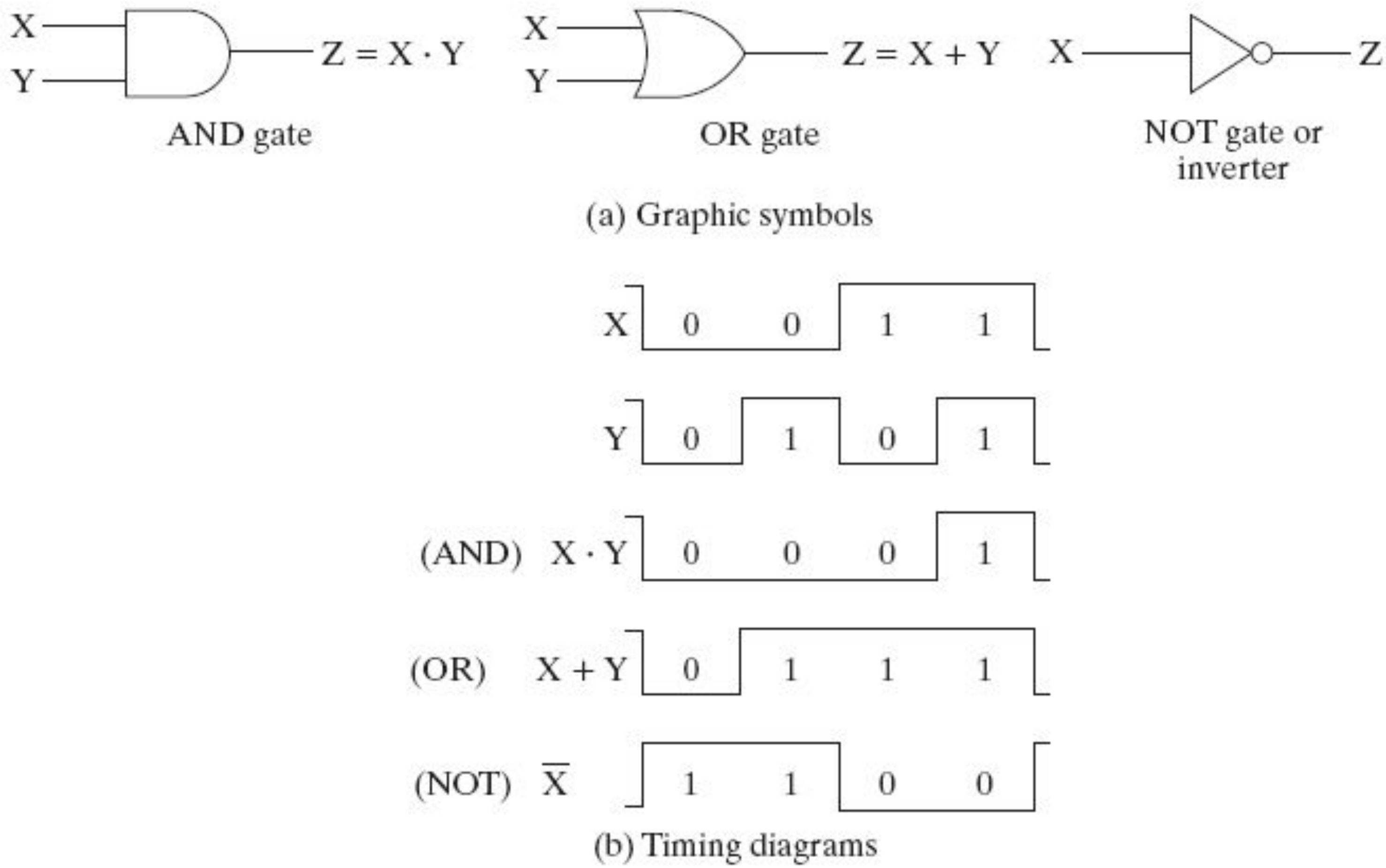
Switches in series => AND



Normally-closed switch => NOT



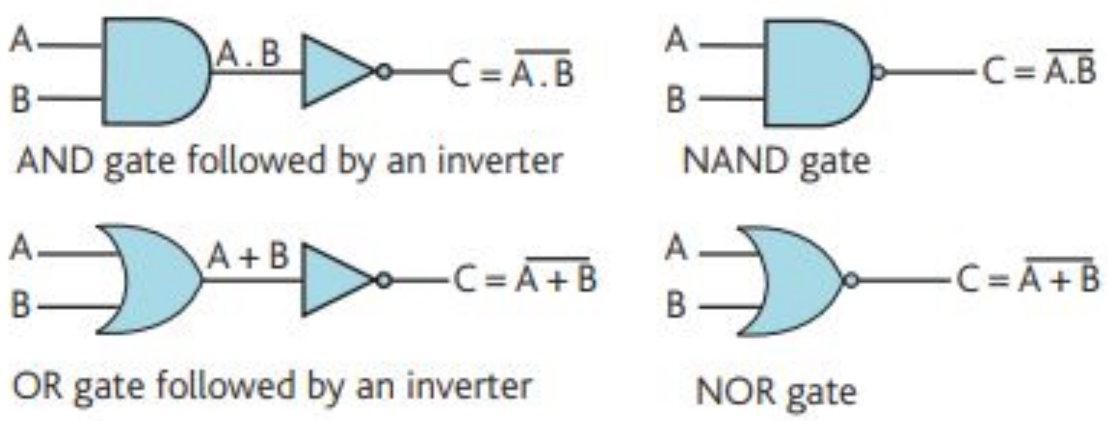
Logic Gate Behavior



The NAND & NOR Gates

- The two most widely used gates in real circuits are the NAND and NOR gates.
- These aren’t fundamental gates because the NAND gate is derived from an AND gate followed by an inverter (Not AND).
- NOR gate is derived from an OR gate followed by an inverter (Not OR), respectively. T

The NAND & NOR Gates…



The NAND & NOR Gates…

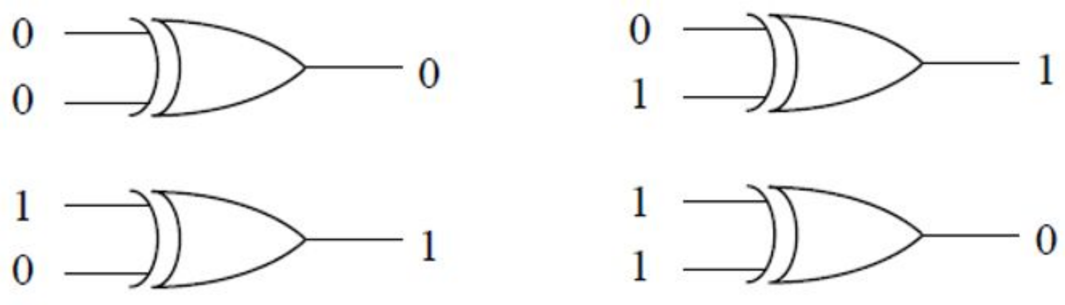
A	B	NAND $C = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

A	B	NOR $C = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

The XOR Gate

- An ***Exclusive-OR gate*** is sometimes called a parity checker.
- Parity checkers count the number of ones being input to a circuit and output a logic 1 or 0 based on whether the number of ones is odd or even.

The XOR Gate



All 06 Logic Gates

Inputs		Output					
A	B	AND $A \cdot B$	OR $A+B$	NAND $\overline{A.B}$	NOR $\overline{A + B}$	EOR $A \oplus B$	EXNOR $\overline{A \oplus B}$
0	0	0	0	1	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	1	0	0	0	1

Boolean Algebra, Law and Circuit simplification

Boolean Algebra

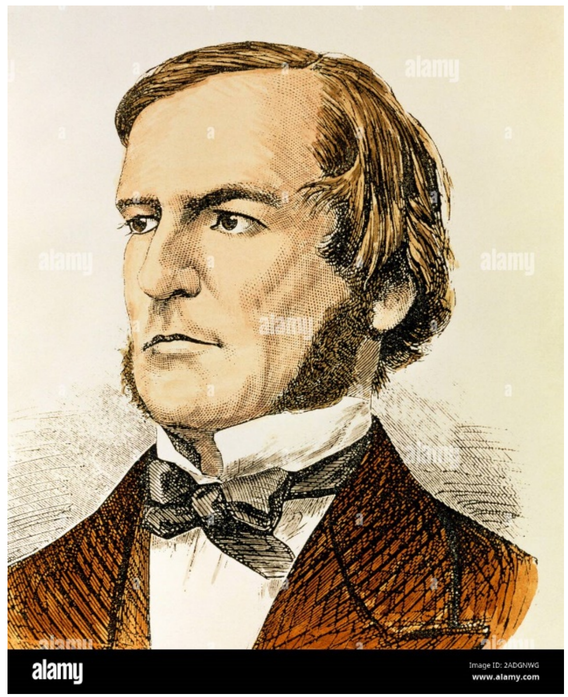
- Boolean expression: a expression formed by binary variables, for example,
- Boolean function: a binary variable identifying the function followed by an equal sign and a Boolean expression for example

$$L(D, X, A) = D\overline{X} + A$$

Introduction to Boolean Algebra

- We can write the Boolean equation $C = A \cdot B$ which uses variables A, B, and C and the AND operator.
- George Boole was an English mathematician (1815–1864) who developed a mathematical analysis of logic and published it in his book *An Investigation of the Laws* of Thought in 1854.

George Boole



Axioms and theorems of Boolean algebra

- An axiom or postulate is a fundamental rule that has to be taken for granted (i.e. the axioms of Boolean algebra define the framework of Boolean algebra from which everything else can be derived).
- Boolean variables obey the same commutative, distributive, and associative laws as the variables of conventional algebra.

Commutative, distributive, and associative laws of Boolean algebra

$A + B = B + A$ $A \cdot B = B \cdot A$	The AND and OR operators are **commutative** so that the order of the variables in a sum or product group does not matter.
$A \cdot (B \cdot C) = (A \cdot B) \cdot C$ $A + (B + C) = (A + B) + C$	The AND and OR operators are **associative** so that the order in which sub-expressions are evaluated does not matter.
$A \cdot (B + C) = A \cdot B + A \cdot C$ $A + B \cdot C = (A + B)(A + C)$	The AND operator behaves like multiplication and the OR operator like addition. The first **distributive** property states that in an expression containing both AND and OR operators the AND operator takes precedence over the OR. The second distributive law, $A + B \cdot C = (A + B)(A + C)$, is not valid in conventional algebra.

Basic Axioms of Boolean Algebra

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

Boolean operations on a constant and a variable

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

DeMorgan's Theorem

- The purpose of DeMorgan's Theorem is to allow us to distribute an inverter from the output of an AND or OR gate to the gate's inputs.
- In doing so, an AND gate is switched to an OR gate and an OR gate is switched to an AND gate

DeMorgan's Theorem

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

$$\overline{A} + \overline{B} = \overline{A \cdot B}$$

Next Week Discussion

- How to draw circuits?