Logic Gates

Course Code: CSC 1101 Course Title: Introduction to Computer Studies

Dept. of Computer Science
Faculty of Science and Technology

Lecturer No:	**09**	**Week No:**	**09**	**Semester:**	
Lecturer:	*Name & email*				

Lecture Outline

- Understanding Logic Gate and the use of these.

Specific Objectives

- Logic gates
- Logic circuits and Boolean expressions
- Combinational circuits and design

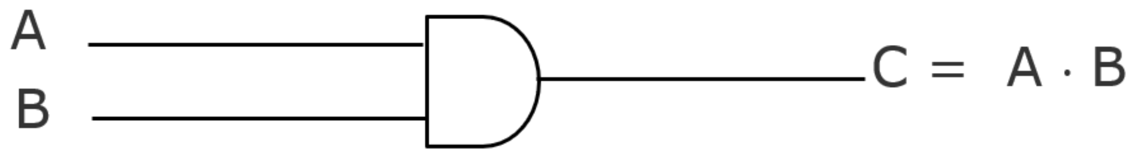
Logic Gate

- Logic gates are electronic circuits that operate on one or more input signals to produce standard output signal
- Are the building blocks of all the circuits in a computer
- Some of the most basic and useful logic gates are AND, OR, NOT, NAND and NOR gates

AND Gate

- Physical realization of logical multiplication (AND) operation.
- Generates output signal of 1 only if all input signals are also 1

AND Gate (Block Diagram Symbol and Truth Table)

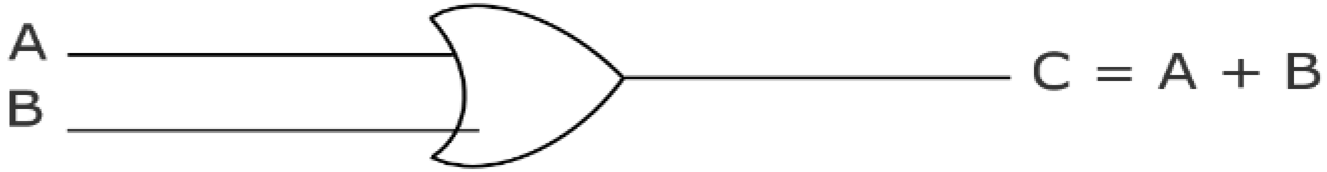


Inputs		Output
A	B	$C = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

- Physical realization of logical addition (OR) operation
- Generates an output signal of 1 if at least one of the input signals is also 1

OR Gate (Block Diagram Symbol and Truth Table)

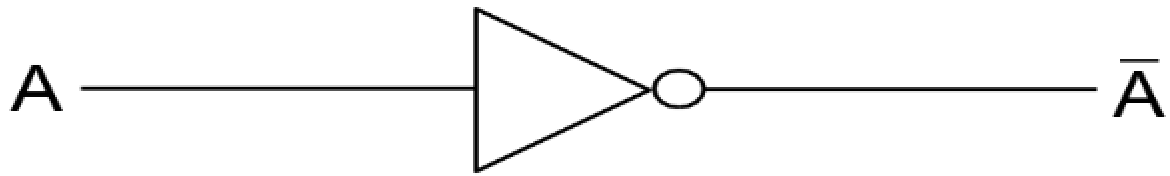


Inputs		Output
A	B	C = A + B
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate

- Physical realization of complementation operation
- Generates an output signal, which is the reverse of the input signal

NOT Gate (Block Diagram Symbol and Truth Table)

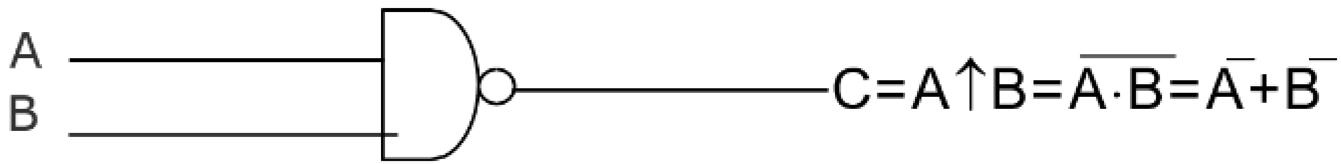


Input	Output
A	A
0	1
1	0

NAND Gate

- Complemented AND gate
- Generates an output signal of:
  - 1 if any one of the inputs is a 0
  - 0 when all the inputs are 1

NAND Gate (Block Diagram Symbol and Truth Table)

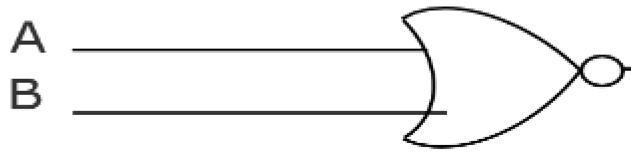


Inputs		Output
A	B	C =A+B
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate

- Complemented OR gate
- Generates an output signal of:
  - 1 only when all inputs are 0
  - 0 if any one of inputs is a 1

NOR Gate (Block Diagram Symbol and Truth Table)



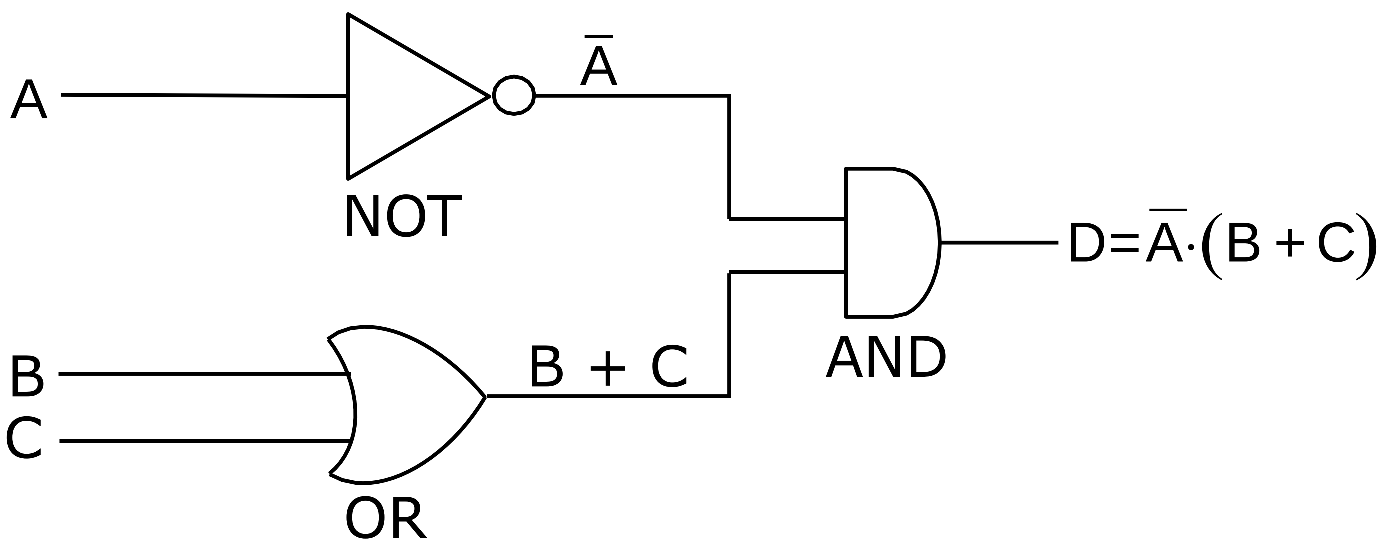
$C = A \downarrow B = \overline{A+B} = \bar{A} \cdot \bar{B}$

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

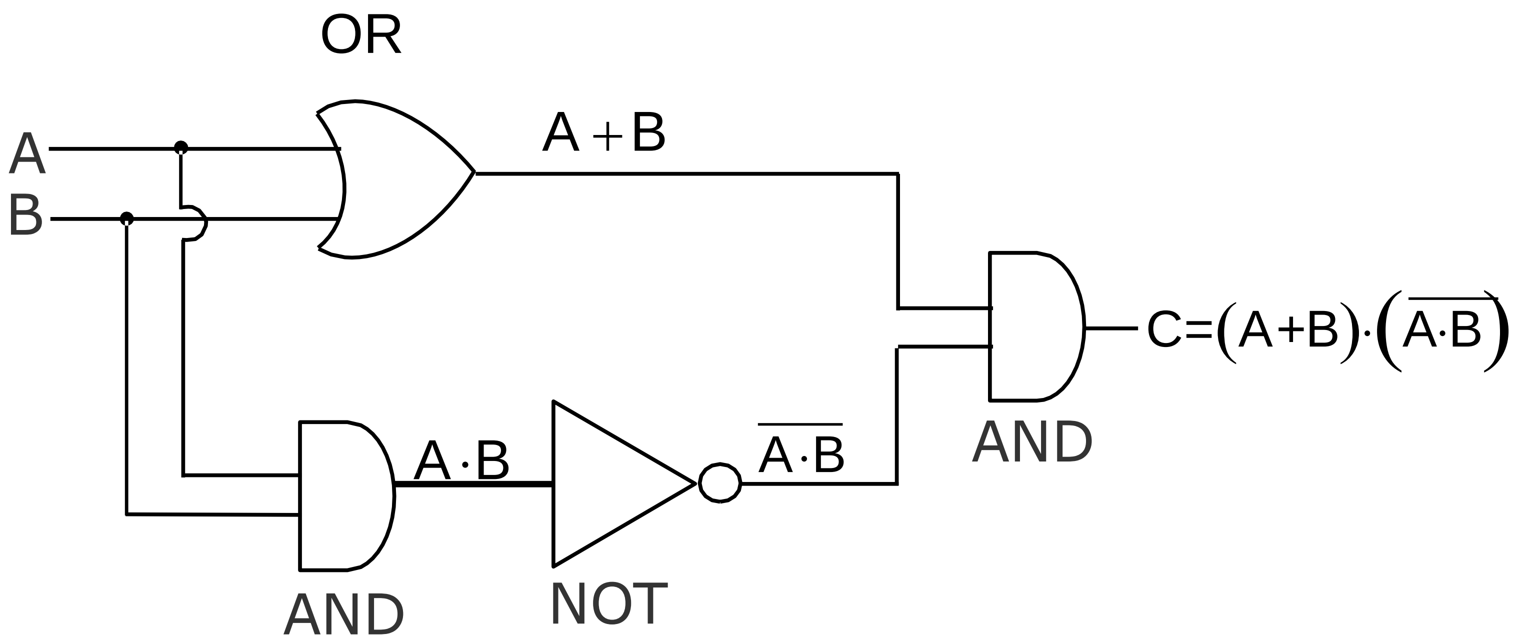
Logic Circuits

- When logic gates are interconnected to form a gating / logic network, it is known as a *combinational logic circuit*
- The Boolean algebra expression for a given logic circuit can be derived by systematically progressing from input to output on the gates
- The three logic gates (AND, OR, and NOT) are logically complete because any Boolean expression can be realized as a logic circuit using only these three gates

Finding Boolean Expression of a Logic Circuit(Example 1)

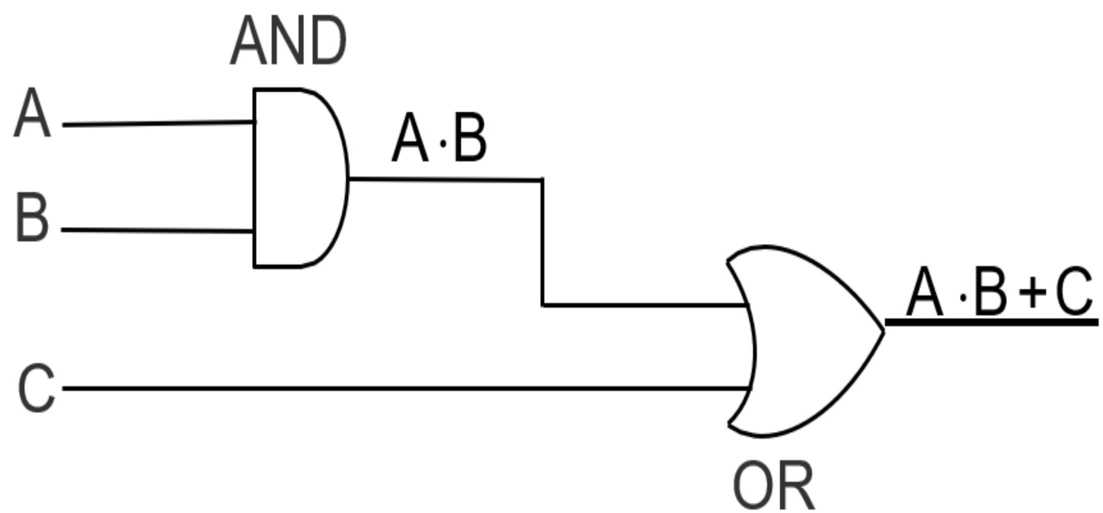


Finding Boolean Expression of a Logic Circuit(Example 2)



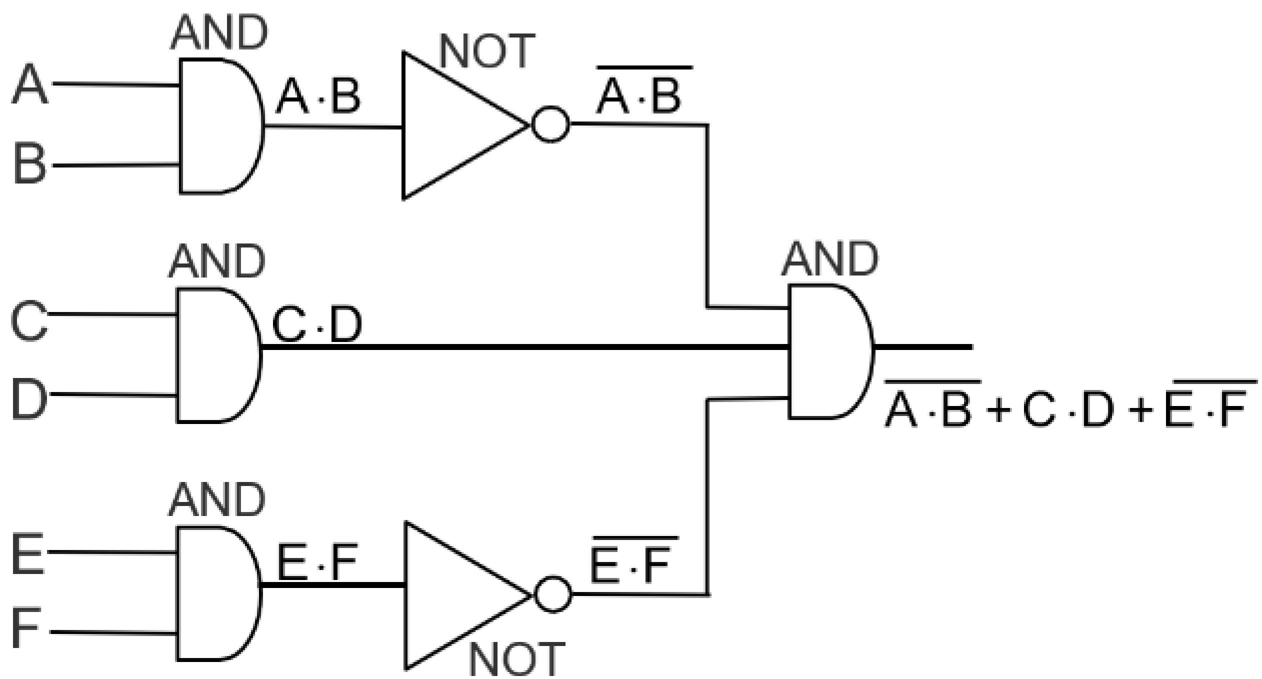
Constructing a Logic Circuit from a Boolean Expression (Example 1)

Boolean Expression = $A \cdot B + C$



Constructing a Logic Circuit from a Boolean Expression (Example 2)

Boolean Expression = $\overline{A\cdot B}+C\cdot D+\overline{E\cdot F}$

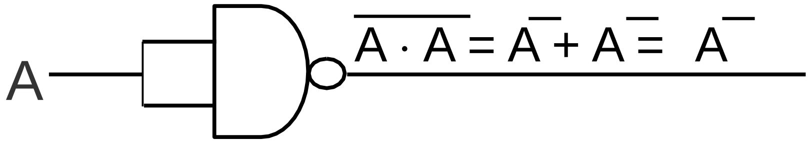


Universal NAND Gate

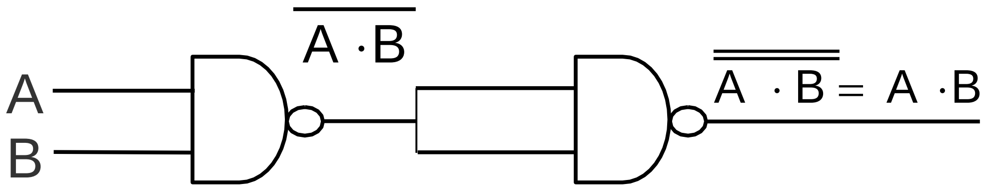
- Nand gate is an universal gate, I is alone sufficient to implement any Boolean expression

- **To understand this, consider:**
  - Basic logic gates (AND, OR, and NOT) are logically complete
  - Sufficient to show that AND, OR, and NOT gates can be implemented with NAND gates

Implementation of NOT, AND and OR Gates by NAND Gates



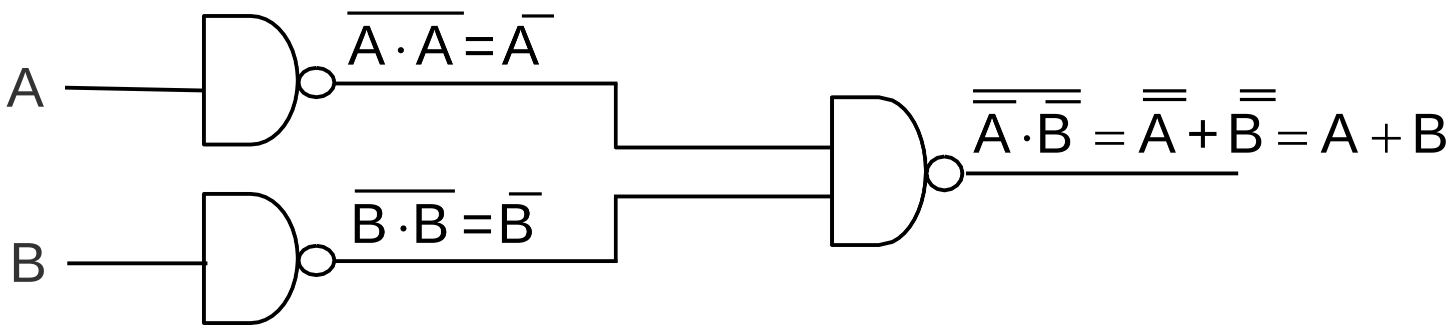
(a) NOT gate implementation.



(b) AND gate implementation.

(Continued on next slide)

Implementation of NOT, AND and OR Gates by NAND Gates (*Continued from previous slide..)*



(c) OR gate implementation.

Method of Implementing a Boolean Expression with Only NAND Gates

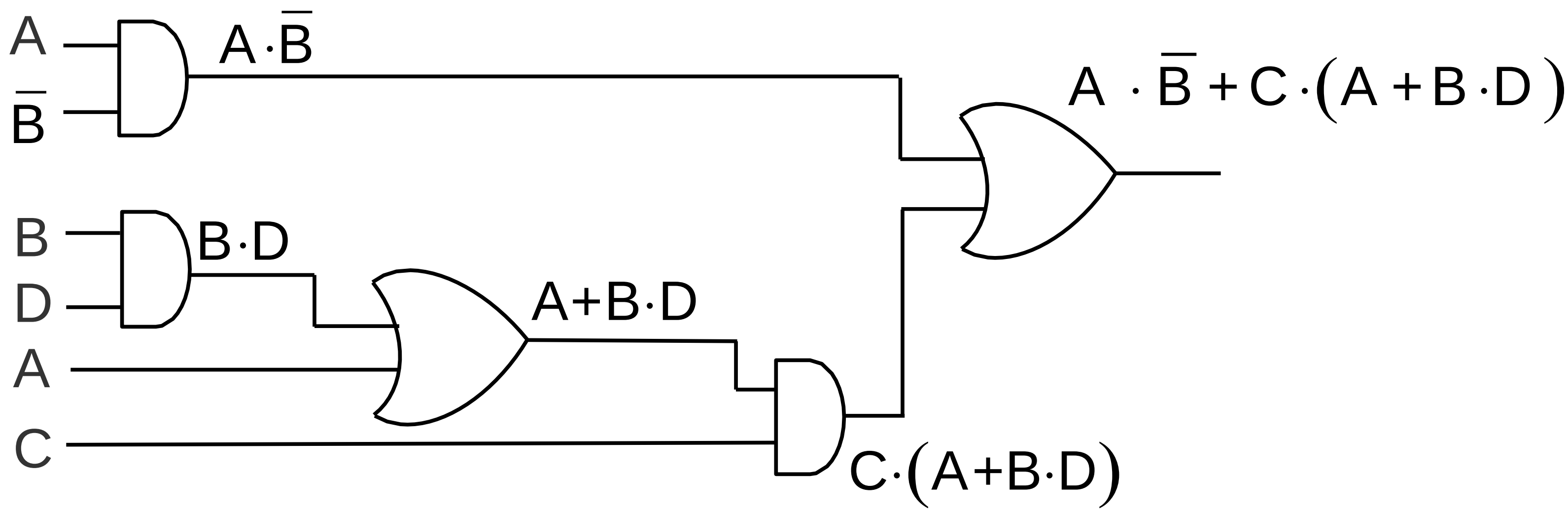
Step 1: From the given algebraic expression, draw the logic diagram with AND, OR, and NOT gates. Assume that both the normal (A) and complement ($\overline{A}$) inputs are available

Step 2: Draw a second logic diagram with the equivalent NAND logic substituted for each AND, OR, and NOT gate

Step 3: Remove all pairs of cascaded inverters from the diagram as double inversion does not perform any logical function. Also remove inverters connected to single external inputs and complement the corresponding input variable

Implementing a Boolean Expression with Only NAND Gates (Example)

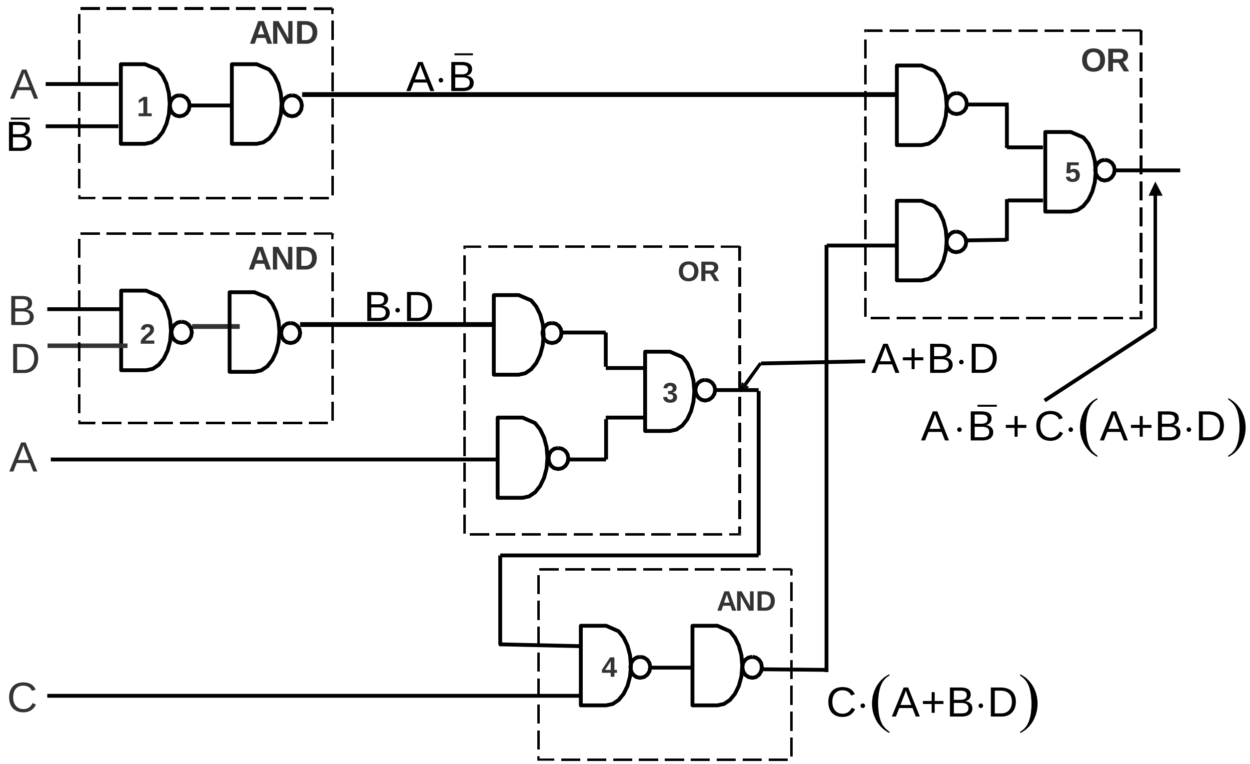
Boolean Expression = $A \cdot \overline{B} + C \cdot (A + B \cdot D)$



(a) **Step 1:** AND/OR implementation

(Continued on next slide)

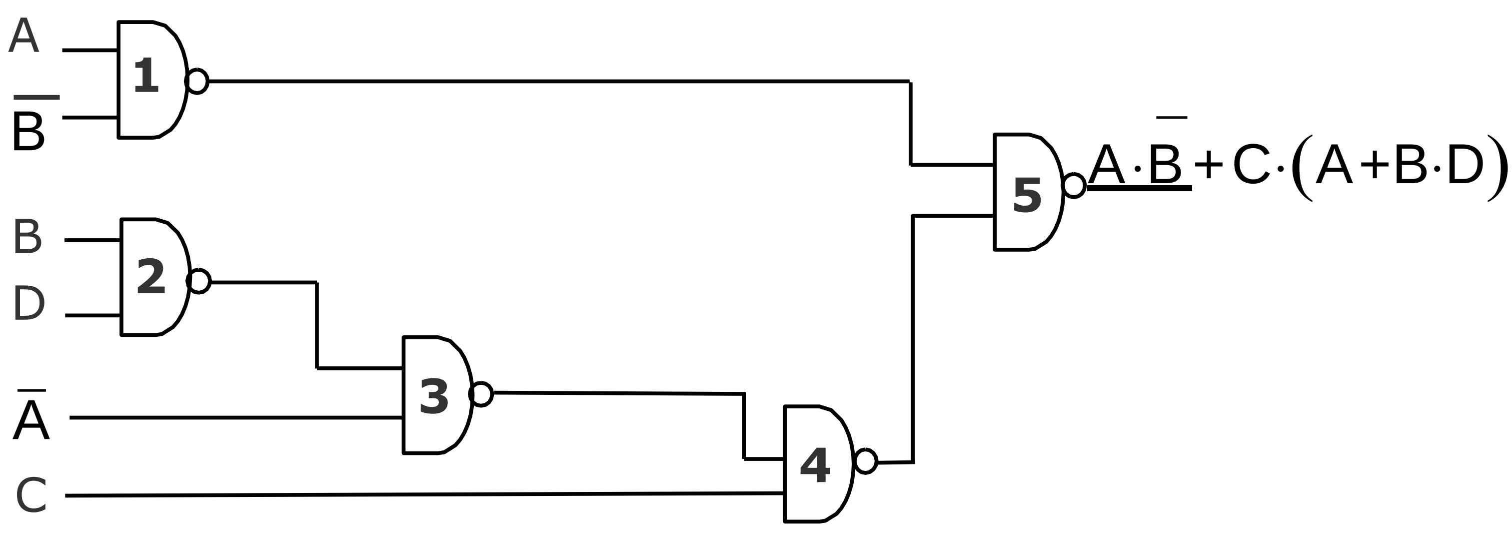
Implementing a Boolean Expression with Only NAND Gates (Example) *(Continued from previous slide..)*



(b) **Step 2:** Substituting equivalent NAND functions

(Continued on next slide)

Implementing a Boolean Expression with Only NAND Gates (Example) *(Continued from previous slide..)*

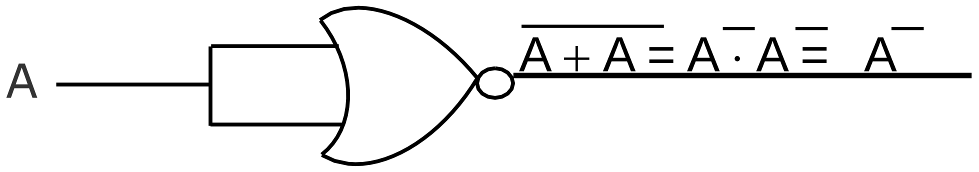


(c) **Step 3:** NAND implementation.

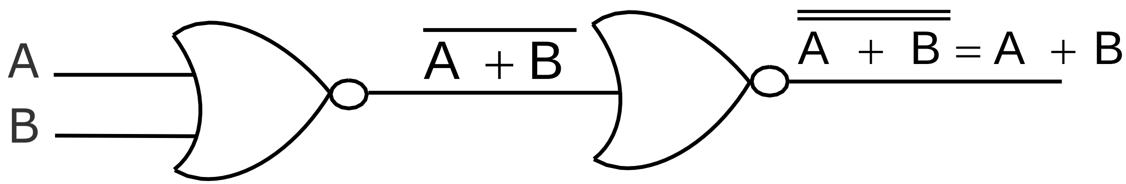
Universal NOR Gate

- NOR gate is an universal gate, it is alone sufficient to implement any Boolean expression
- To understand this, consider:
  - Basic logic gates (AND, OR, and NOT) are logically complete
  - Sufficient to show that AND, OR, and NOT gates can be implemented with NOR gates

Implementation of NOT, OR and AND Gates by NOR Gates



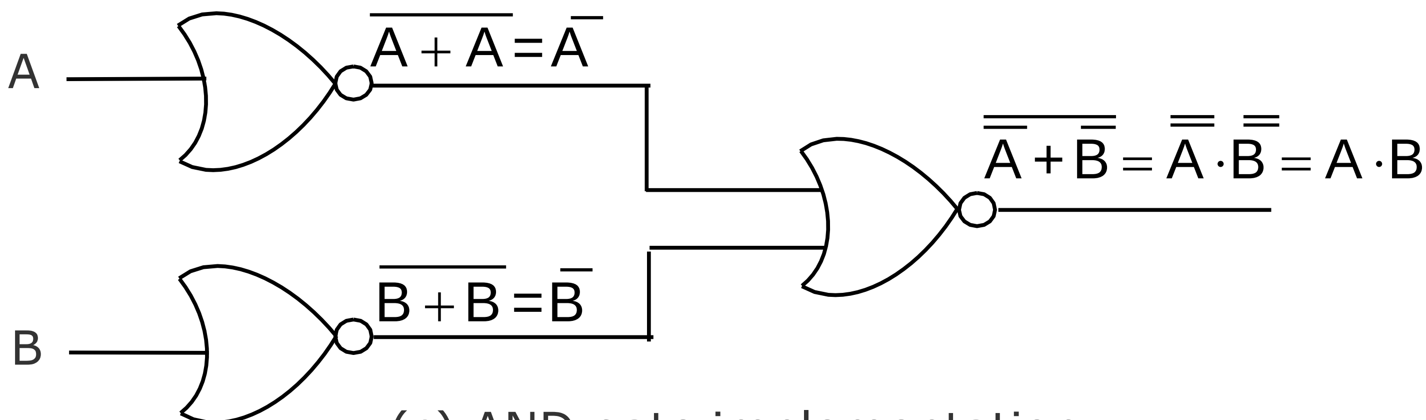
(a) NOT gate implementation.



(b) OR gate implementation.

(Continued on next slide)

Implementation of NOT, OR and AND Gates by NOR Gates *(Continued from previous slide..)*



(c) AND gate implementation.

Method of Implementing a Boolean Expression with Only NOR Gates

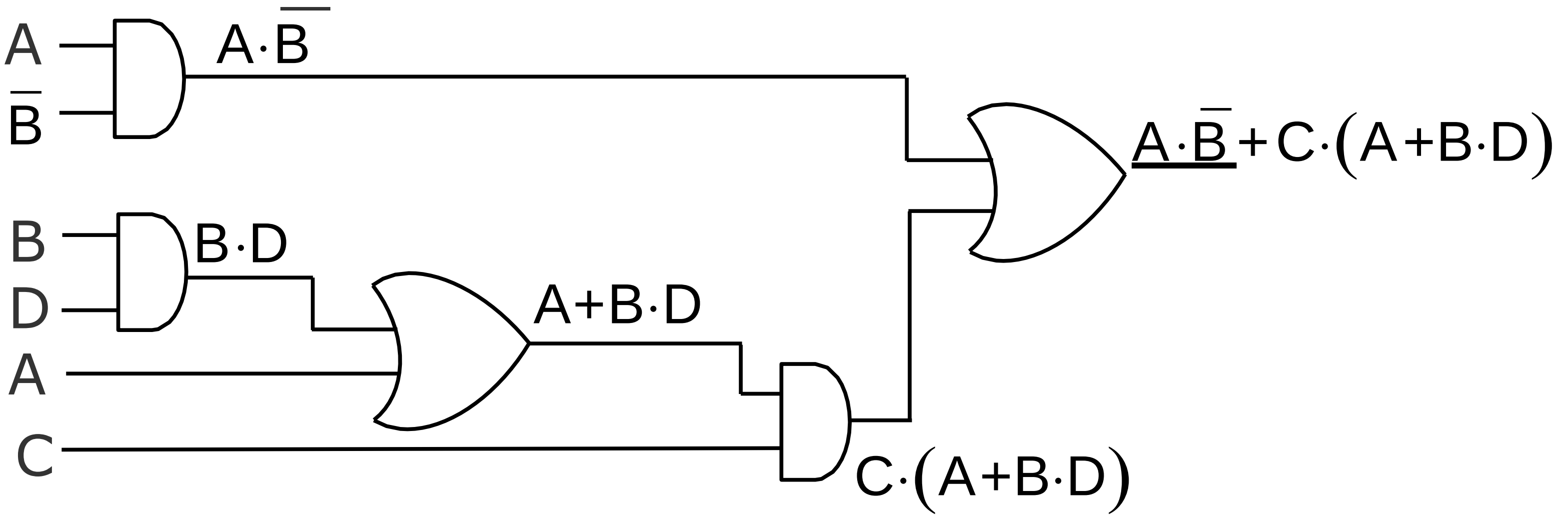
Step 1: For the given algebraic expression, draw the logic diagram with AND, OR, and NOT gates. Assume that both the normal (A) and complement $(\overline{A})$ inputs are available

Step 2: Draw a second logic diagram with equivalent NOR logic substituted for each AND, OR, and NOT gate

Step 3: Remove all parts of cascaded inverters from the diagram as double inversion does not perform any logical function. Also remove inverters connected to single external inputs and complement the corresponding input variable

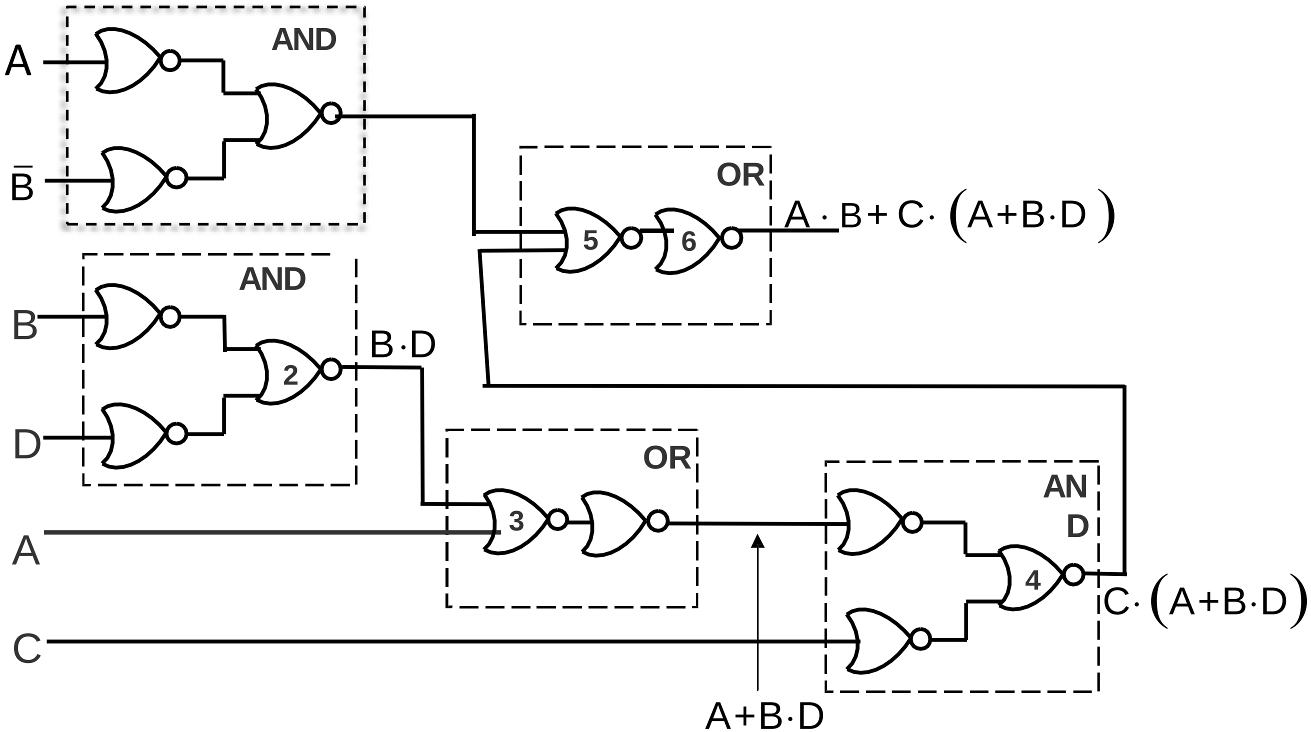
Implementing a Boolean Expression with Only NOR Gates (Examples)*(Continued from previous slide..)*

Boolean Expression = $A \cdot \overline{B} + C \cdot (A + B \cdot D)$



(a) **Step 1:** AND/OR implementation.

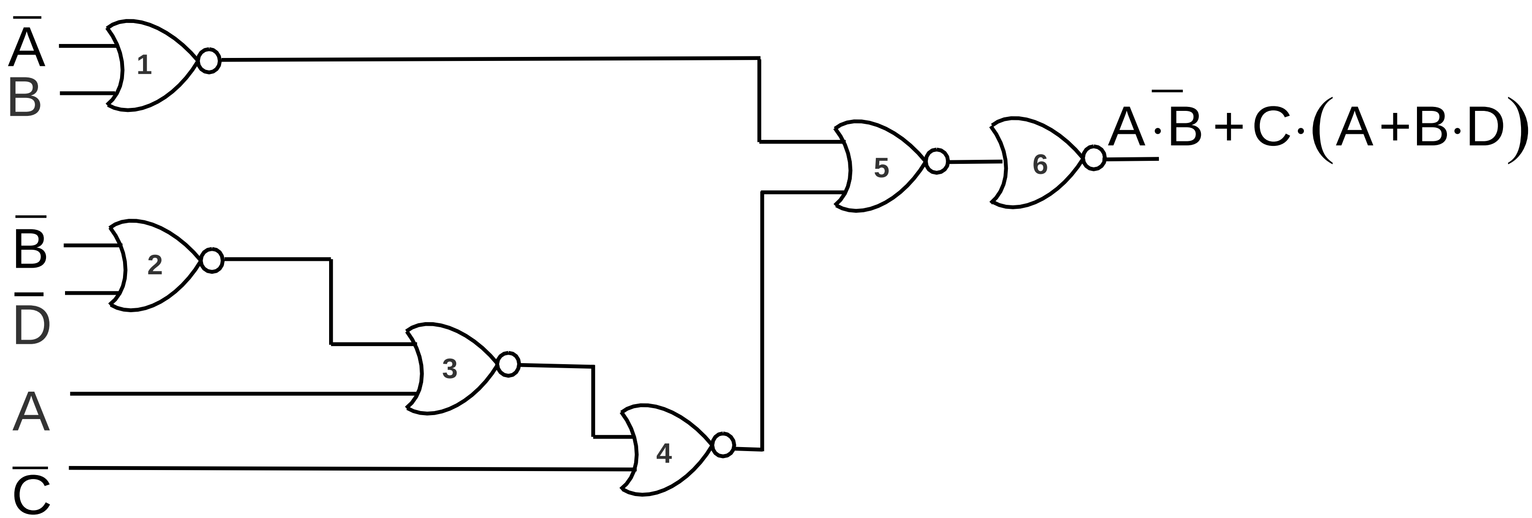
(Continued on next slide)



(b) **Step 2:** Substituting equivalent NOR functions.

(Continued on next slide)

Implementing a Boolean Expression with Only NOR Gates (Examples)*(Continued from previous slide..)*



(c) **Step 3:** NOR implementation.

Key Words/Phrases

- Absorption law
- AND gate
- Associative law
- Boolean algebra
- Boolean expression
- Boolean functions
- Boolean identities
- Canonical forms for Boolean functions
- Combination logic circuits
- Cumulative law
- Complement of a function
- Complementation
- De Morgan's law
- Distributive law
- Dual identities
- Equivalence function
- Exclusive-OR function
- Exhaustive enumeration method
- Half-adder
- Idempotent law
- Involution law
- Literal
- Logic circuits
- Logic gates
- Logical addition
- Logical multiplication
- Maxterms
- Minimization of Boolean functions
- Minterms
- NAND gate
- NOT gate
- Operator precedence
- OR gate
- Parallel Binary Adder
- Perfect induction method
- Postulates of Boolean algebra
- Principle of duality
- Product-of-Sums expression
- Standard forms
- Sum-of Products expression
- Truth table
- Universal NAND gate
- Universal NOR gate

References

- Computer Fundamentals: Pradeep K. Sinha & Priti Sinha

Thank you