Digital Engineering

Assoc. Prof. Osama Elnahas, Dr. Dina Abdelhafiz

Dr. Bassant Tolba, Dr. Radwa Rady

Second Year –Information Technology Program
Fall 2025

Lecture (2)

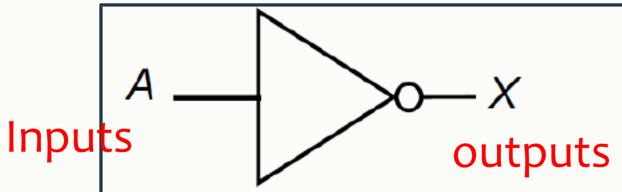
Part A

Basic Gates

Basic Gates

- Unary gate: A **NOT** gate is sometimes referred to as an *inverter* because it inverts the input value

NOT (Inverter)



Logic diagram symbol

$X = \overline{A}$

$X = A'$

Boolean expression

A NOT gate is sometimes referred to as an *inverter* because it inverts the input value

A	X
0	**1**
1	**0**

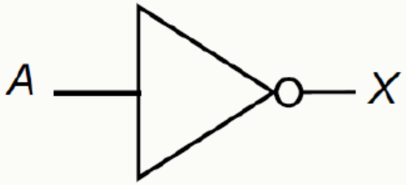
Truth table

Basic Gates

- Unary and binary gates:

The *Output* signal from an **AND** gate is 1 (ON) if and only if **both** *Input* signals are 1.

NOT (Inverter)

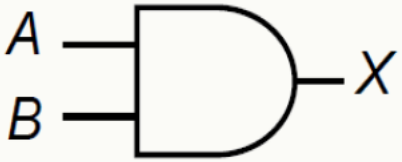


$X = \bar{A}$

$X = A'$

A	X
0	**1**
1	**0**

AND



Logic diagram symbol

$X = A \cdot B$

Boolean expression

A	B	X
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

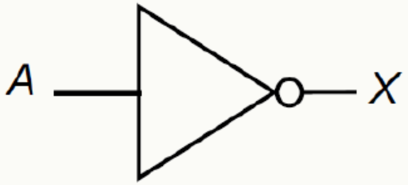
Truth table

Basic Gates

The *Output* signal from an **OR** gate is on, 1 if either, or both, *Input* signals are on, 1.

- Unary and binary gates:

NOT (Inverter)

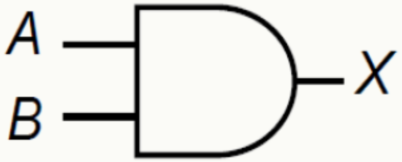


$X = \bar{A}$

$X = A'$

A	X
0	**1**
1	**0**

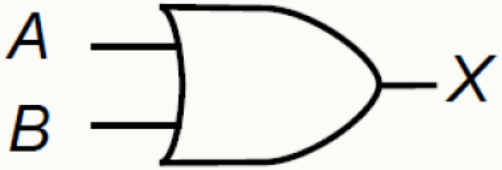
AND



$X = A \cdot B$

A	B	X
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

OR



Logic diagram symbol

$X = A + B$

Boolean expression

A	B	X
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

Truth table

Basic Gates

- Unary and binary gates:

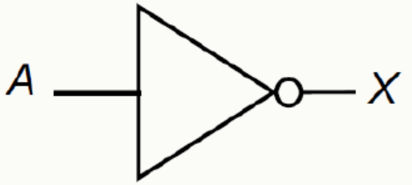
- The *Output* from an **XOR** (exclusive or) is *True* (on, 1) if and only if the *Input* signals are different..

Logic diagram symbol

Boolean expression

Truth table

NOT (Inverter)

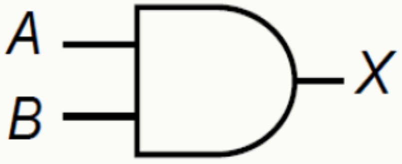


$X = \bar{A}$

$X = A'$

A	X
0	**1**
1	**0**

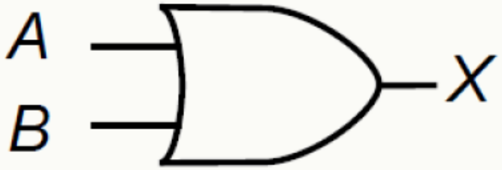
AND



$X = A \cdot B$

A	B	X
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

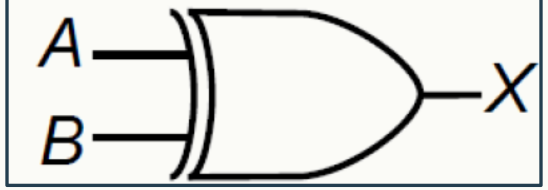
OR



$X = A + B$

A	B	X
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

XOR



Exclusive OR

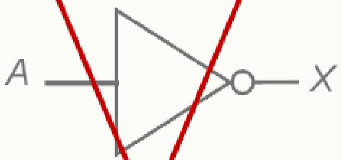
$X = A \oplus B$

A	B	X
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

Basic Gates

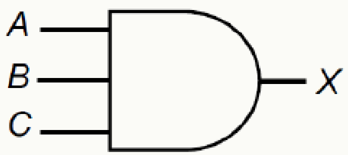
- With 3 inputs: Guess the outputs ??

NOT?



A	X
0	1
1	0

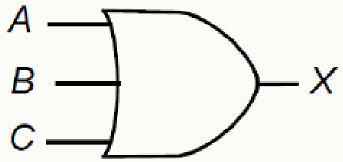
AND



A	B	C	X
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

$$X = A \cdot B \cdot C$$

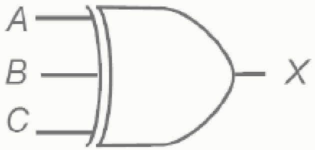
OR



A	B	C	X
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

$$X = A + B + C$$

XOR



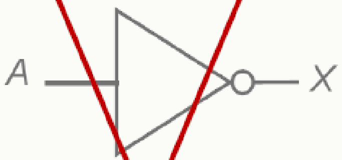
A	B	C	X
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

$$X = A \oplus B \oplus C$$

Basic Gates

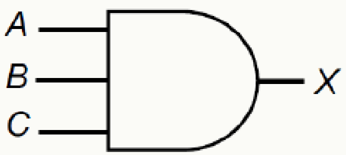
- With 3 inputs: Guess the outputs ??

NOT?



A	X
0	1
1	0

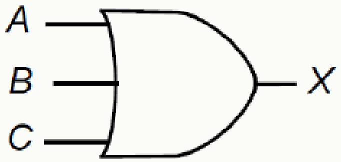
AND



A	B	C	X
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**0**
1	0	1	**0**
1	1	0	**0**
1	1	1	**1**

$X = A \cdot B \cdot C$

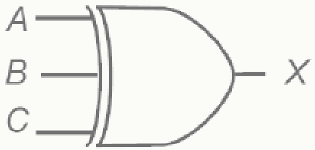
OR



A	B	C	X
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

$X = A + B + C$

XOR



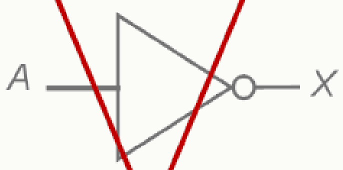
A	B	C	X
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

$X = A \oplus B \oplus C$

Basic Gates

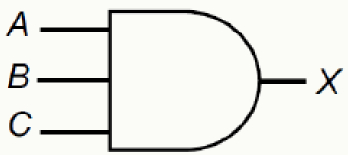
- With 3 inputs: Guess the outputs ??

NOT?



A	X
0	1
1	0

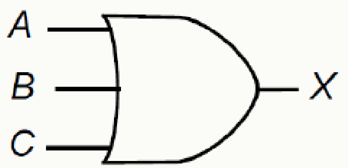
AND



A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$$X = A \cdot B \cdot C$$

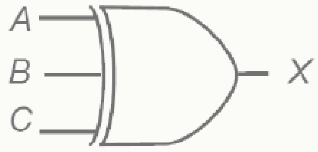
OR



A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$$X = A + B + C$$

XOR



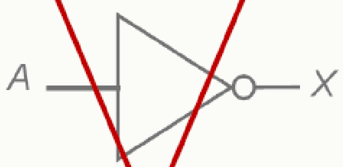
A	B	C	X
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

$$X = A \oplus B \oplus C$$

Basic Gates

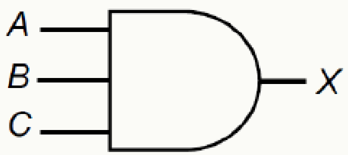
- With 3 inputs:

NOT?



A	X
0	1
1	0

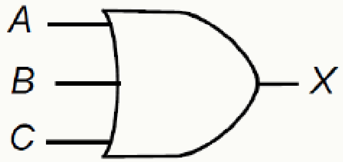
AND



A	B	C	X
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**0**
1	0	1	**0**
1	1	0	**0**
1	1	1	**1**

$X = A \cdot B \cdot C$

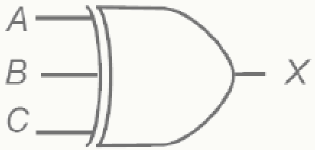
OR



A	B	C	X
0	0	0	**0**
0	0	1	**1**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

$X = A + B + C$

XOR



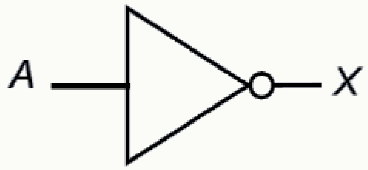
A	B	C	X
0	0	0	**0**
0	0	1	**1**
0	1	0	**1**
0	1	1	**0**
1	0	0	**1**
1	0	1	**0**
1	1	0	**0**
1	1	1	**1**

$X = A \oplus B \oplus C$

Inverted Basic Gates

- Buffer: Inverted NOT:

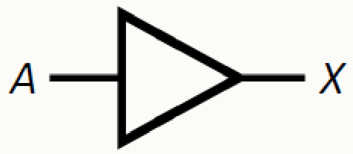
NOT

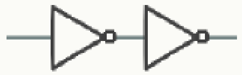


A	X
0	**1**
1	**0**

$$X = \overline{A}$$

"Buffer"





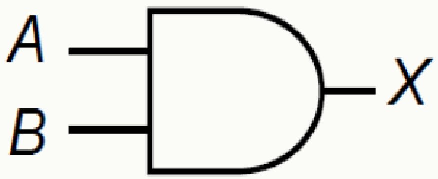
A	X
0	**0**
1	**1**

$$X = A$$

Inverted Basic Gates

- NAND: Inverted AND:

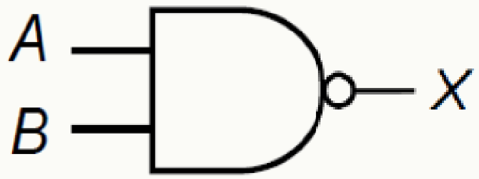
AND



A	B	X
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

$$X = A \cdot B$$

NAND



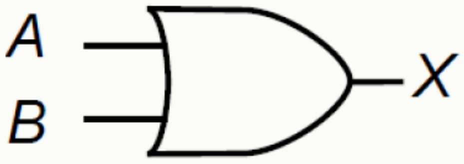
A	B	X
0	0	**1**
0	1	**1**
1	0	**1**
1	1	**0**

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

Inverted Basic Gates

- NOR: Inverted OR:

OR



A	B	X
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

$$X = A + B$$

NOR

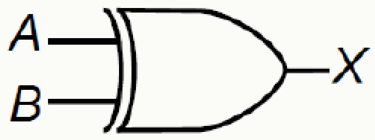
A	B	X
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**0**

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

Inverted Basic Gates

- Exclusive NOR: Inverted XOR
  - Are they different? Are they similar?

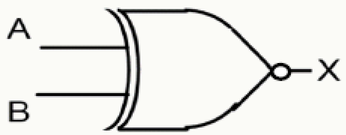
XOR



A	B	X
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

$$X = A \oplus B$$

XNOR

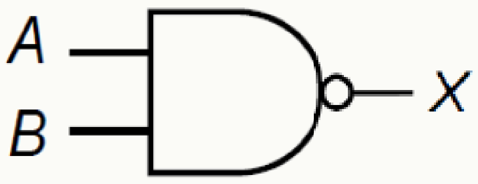


A	B	X
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**1**

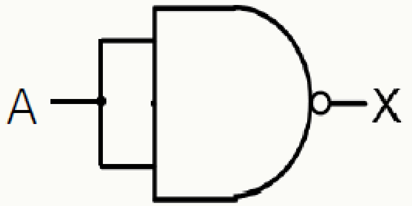
$$X = \overline{A \oplus B}$$

Inverted Basic Gates

- NAND: What if we connected the same input?
- Truth table?



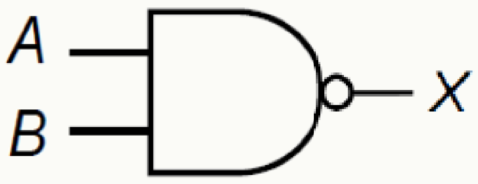
A	B	X
0	0	**1**
0	1	**1**
1	0	**1**
1	1	**0**



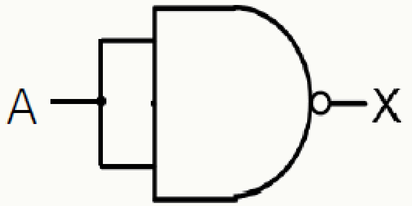
A	A	X

Inverted basic gates

- NAND: What if we connected the same input?
- Truth table?



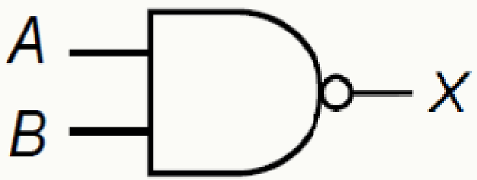
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



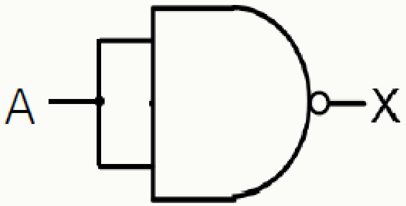
A	A	X
0	0	1

Inverted Basic Gates

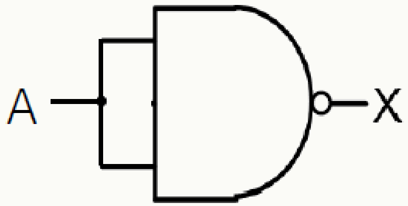
- NAND: What if we connected the same input?
- Truth table?
- It will act as NOT gate (NAND and NOR gates are cheaper).



A	B	X
0	0	**1**
0	1	**1**
1	0	**1**
1	1	**0**



A	A	X
0	0	**1**
1	1	**0**



A	X
0	**1**
1	**0**

Summary

1. AND Gate

Expression: $Y = A \cdot B \cdot C$

(Outputs true only when all inputs are true.)

2. OR Gate

Expression: $Y = A + B + C$

(Outputs true if at least one input is true.)

3. NOT Gate

Expression: $Y = \overline{A}$

(Outputs the inverse of the single input.)

4. NAND Gate

Expression: $Y = \overline{A \cdot B \cdot C}$

(Outputs false only when all inputs are true.)

5. NOR Gate

Expression: $Y = \overline{A + B + C}$

(Outputs true only when all inputs are false.)

6. XOR Gate

Expression: $Y = A \oplus B \oplus C$

(Outputs true if an odd number of inputs are true.)

7. XNOR Gate

Expression: $Y = \overline{A \oplus B \oplus C}$

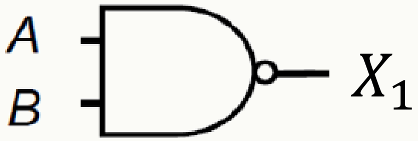
(Outputs true if an even number of inputs are true.)

Question ??

- Which logic gate is "(A NAND B) AND (A OR B)" equivalent to?
- Step (1): Convert into notation: Give the output a symbol X

$$X = (\overline{A \cdot B}) \cdot (A + B)$$

- Step (2): Draw circuit diagram:
  - Start with each ( ).
  - Draw the NAND gate.
  - $X_1 = (\overline{A.B})$

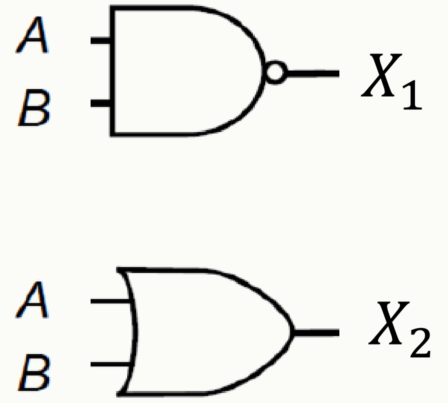


Question ??

- Which logic gate is "(A NAND B) AND (A OR B)" equivalent to?
- Step (1): Convert into notation: Give the output a symbol X

$$X = (\overline{A \cdot B}) \cdot (A + B)$$

- Step (2): Draw circuit diagram:
  - Start with each ( ).
  - Draw the OR gate.
  - $X_2 = (A + B)$

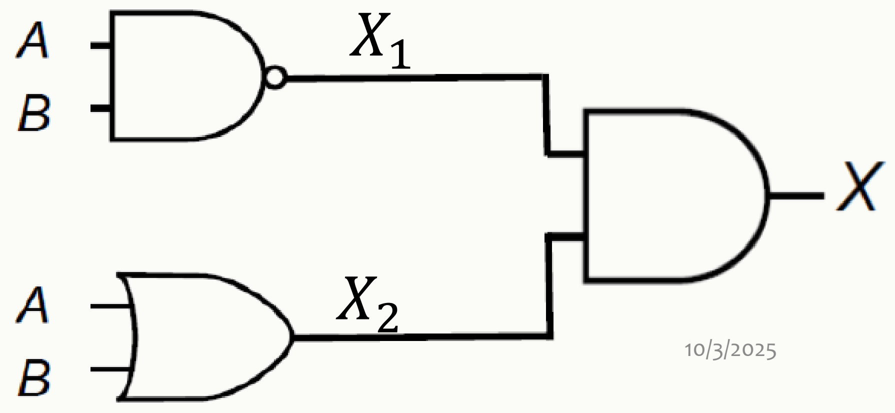


Question ??

- Which logic gate is "(A NAND B) AND (A OR B)" equivalent to?
- Step (1): Convert into notation: Give the output a symbol X

$$X = (\overline{A \cdot B}) \cdot (A + B)$$

- Step (2): Draw circuit diagram:
  - AND between X_1, X_2
  - $X = X_1 . X_2$
  - $X = (\overline{A . B}) . (A + B)$

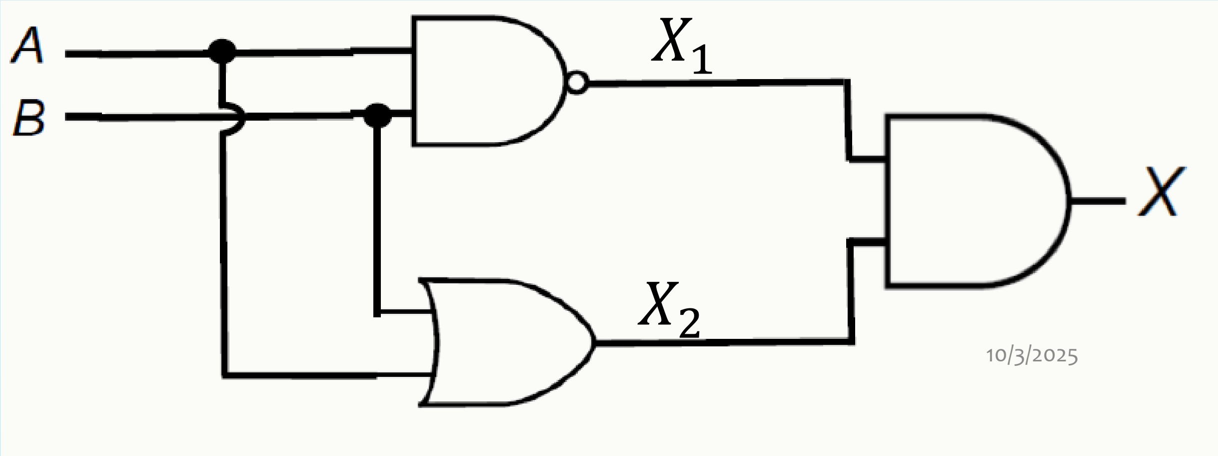


Question ??

- Which logic gate is "(A NAND B) AND (A OR B)" equivalent to?
- Step (1): Convert into notation: Give the output a symbol X

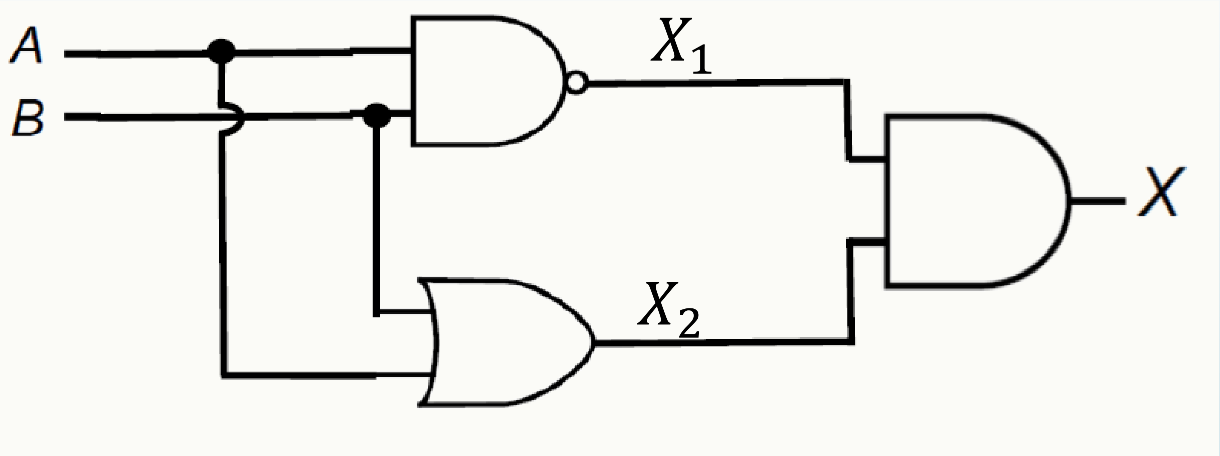
$$X = (\overline{A \cdot B}) \cdot (A + B)$$

- Step (2): Draw circuit diagram:
  - Optimize figure.



Question ??

- Which logic gate is "(A NAND B) AND (A OR B)" equivalent to?



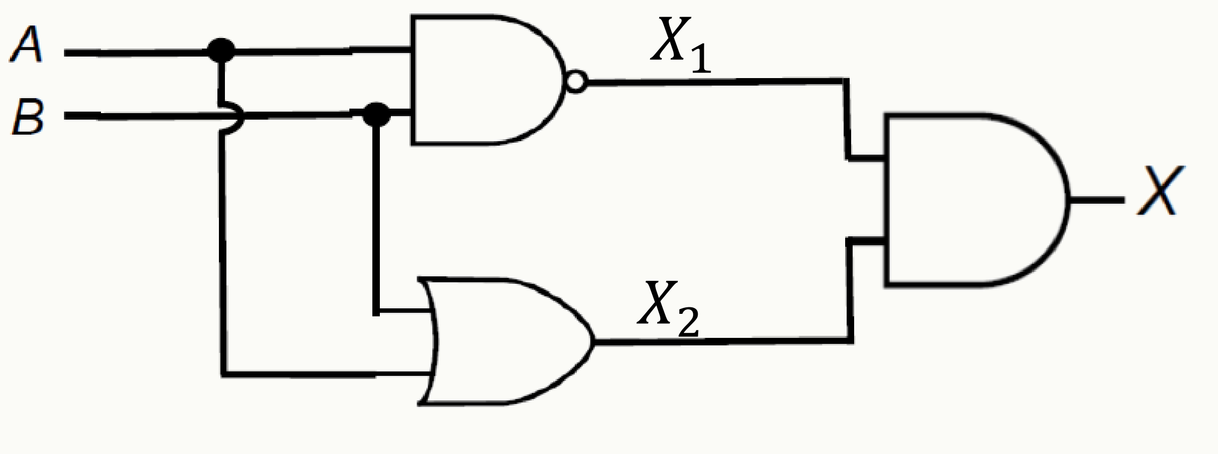
- Step (3): Truth table of X:
  - Truth table of X_1=??
  - $X_1 = (\overline{A.B})$

A NAND B

A	B	X_1
0	0	
0	1	
1	0	
1	1	

Question ??

- Which logic gate is "(A NAND B) AND (A OR B)" equivalent to?
- Step (3): Truth table of X:
  - Truth table of X_1=??
  - $X_1 = (\overline{A.B})$

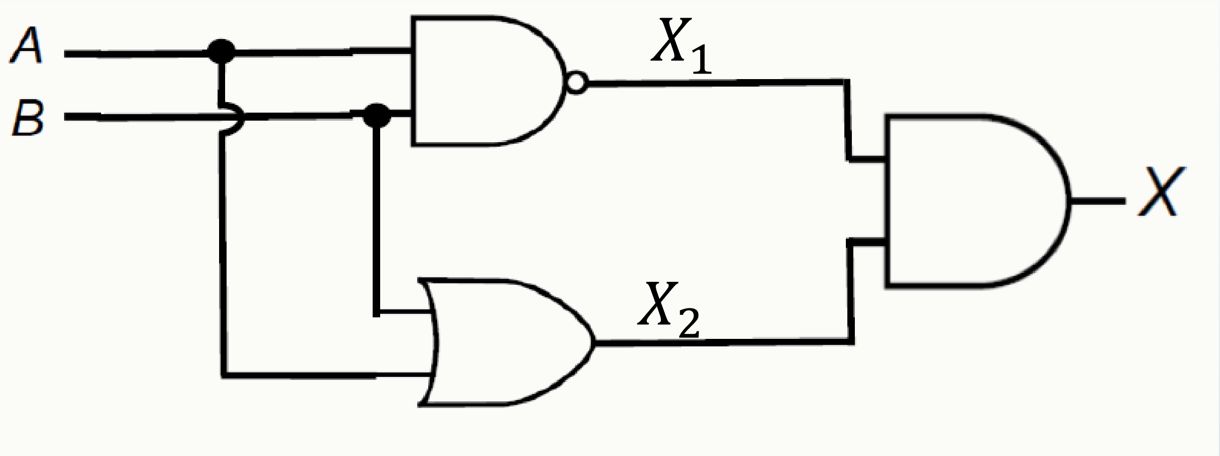


A NAND B

A	B	X_1
0	0	1
0	1	1
1	0	1
1	1	0

Question ??

- Which logic gate is "(A NAND B) AND (A OR B)" equivalent to?
- Step (3): Truth table of X:
  - Truth table of X_2=??
  - $X_2 = A + B$



A NAND B

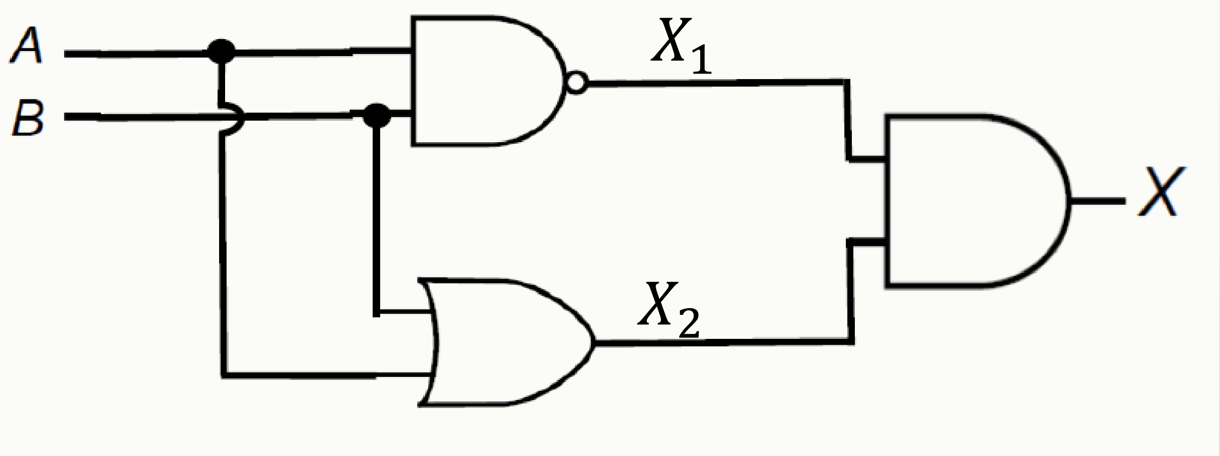
A	B	X_1
0	0	1
0	1	1
1	0	1
1	1	0

A OR B

A	B	X_2
0	0	
0	1	
1	0	
1	1	

Question ??

- Which logic gate is "(A NAND B) AND (A OR B)" equivalent to?
- Step (3): Truth table of X:
  - Truth table of X_2=??
  - $X_2 = A + B$



A NAND B

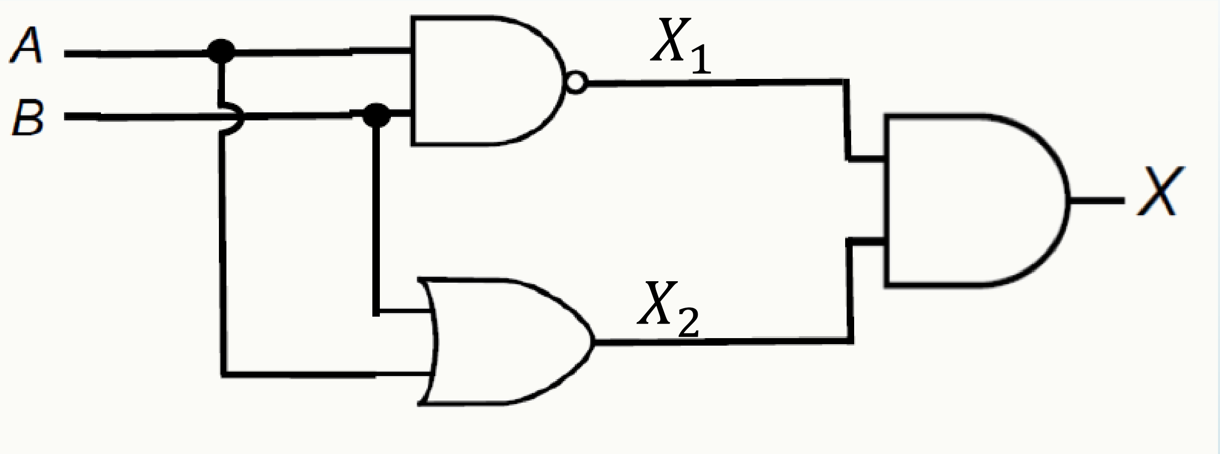
A	B	X_1
0	0	1
0	1	1
1	0	1
1	1	0

A OR B

A	B	X_2
0	0	0
0	1	1
1	0	1
1	1	1

Question ??

- Which logic gate is "(A NAND B) AND (A OR B)" equivalent to?
- Step (3): Truth table of X:
  - Truth table of X=??
  - $X = X_1 \cdot X_2$



A NAND B

A	B	X_1
0	0	**1**
0	1	**1**
1	0	**1**
1	1	**0**

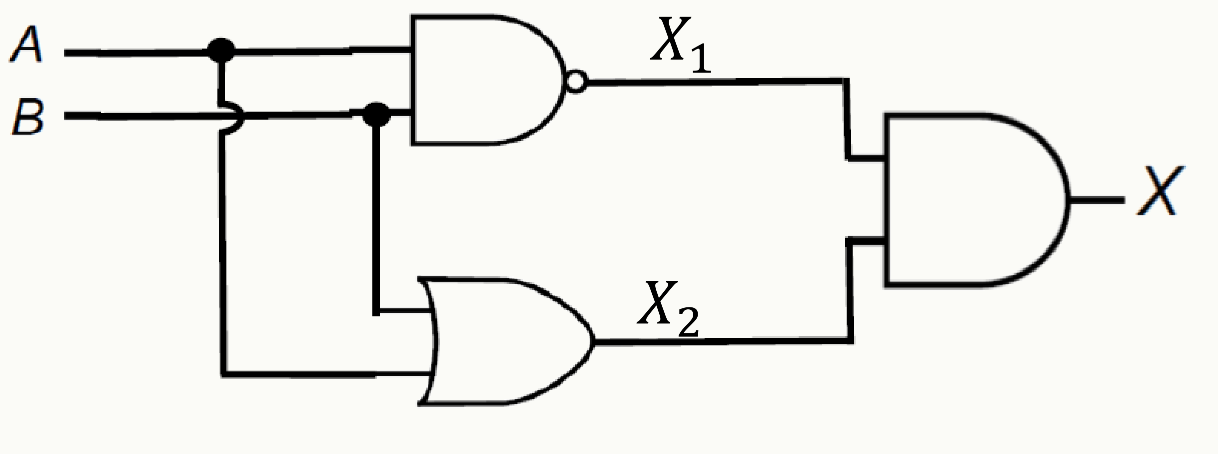
A OR B

A	B	X_2
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

A	B	X_1	X_2	X
0	0	**1**	**0**	
0	1	**1**	**1**	
1	0	**1**	**1**	
1	1	**0**	**1**	

Question ??

- Which logic gate is "(A NAND B) AND (A OR B)" equivalent to?
- Step (3): Truth table of X:
  - Truth table of X=??
  - $X = X_1 \cdot X_2$



A NAND B

A	B	X_1
0	0	1
0	1	1
1	0	1
1	1	0

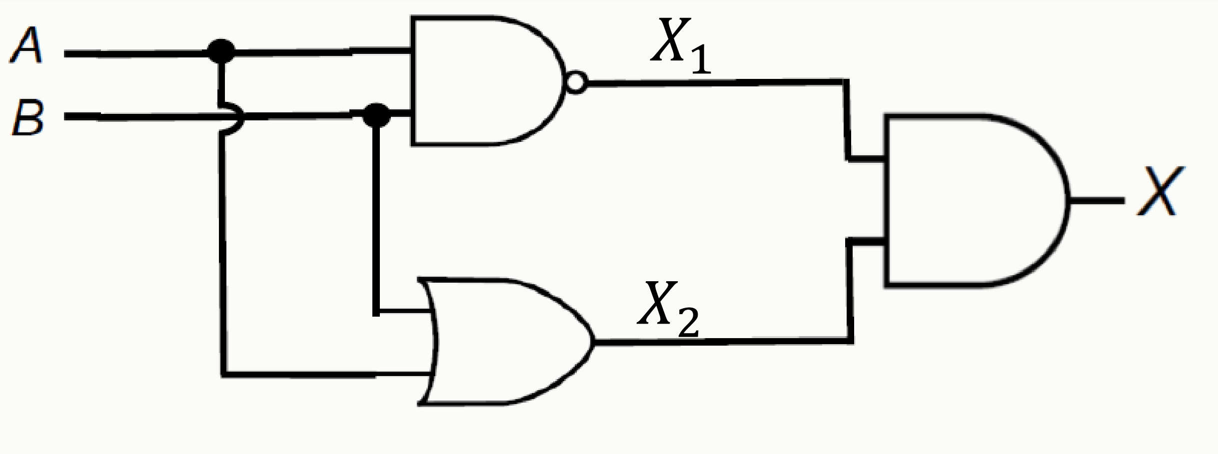
A OR B

A	B	X_2
0	0	0
0	1	1
1	0	1
1	1	1

A	B	X_1	X_2	X
0	0	1	0	0
0	1	1	1	1
1	0	1	1	1
1	1	0	1	0

Question ??

- Which logic gate is "(A NAND B) AND (A OR B)" equivalent to?
- Step (3): Truth table of X:
  - Summary:



A NAND B

A	B	X_1
0	0	1
0	1	1
1	0	1
1	1	0

A OR B

A	B	X_2
0	0	0
0	1	1
1	0	1
1	1	1

A	B	X_1	X_2	X
0	0	1	0	0
0	1	1	1	1
1	0	1	1	1
1	1	0	1	0

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Question ??

- Which logic gate is "(A NAND B) AND (A OR B)" equivalent to?
- Step (4): Compare the known truth tables of AND, OR, NOR, NAND.
  - It is the truth table of XOR.

$$X = A \oplus B \text{ (XOR)}$$

A	*B*	***X***
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

Part B

Boolean Algebra (Switching Algebra)

Operations with 0

- From the AND truth table:
  - Any AND with zero is zero.
  - $\boxed{0 \cdot X = X \cdot 0 = 0}$

A	B	C
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

- From the OR truth table:
  - The OR of any value with zero is the same value.
  - $\boxed{0 + X = X + 0 = X}$

A	B	C
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

Operations with 1

- From the AND truth table:
  - The AND of any value with 1 is the same value
  - $\boxed{1 \cdot X = X \cdot 1 = X}$

A	B	C
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

- From the OR truth table:
  - Any OR with 1 is 1
  - $\boxed{1 + X = X + 1 = 1}$

A	B	C
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

Idempotent

- The operation of a variable with itself:
- From the AND truth table:
  - $\boxed{X \cdot X = X}$

A	B	C
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

- From the OR truth table:
  - $\boxed{X + X = X}$

A	B	C
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

Complementarity

- The operation of a variable with its complement:
- From the AND truth table:
  - $X \cdot \bar{X} = 0$

A	B	C
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

- From the OR truth table:
  - $X + \bar{X} = 1$

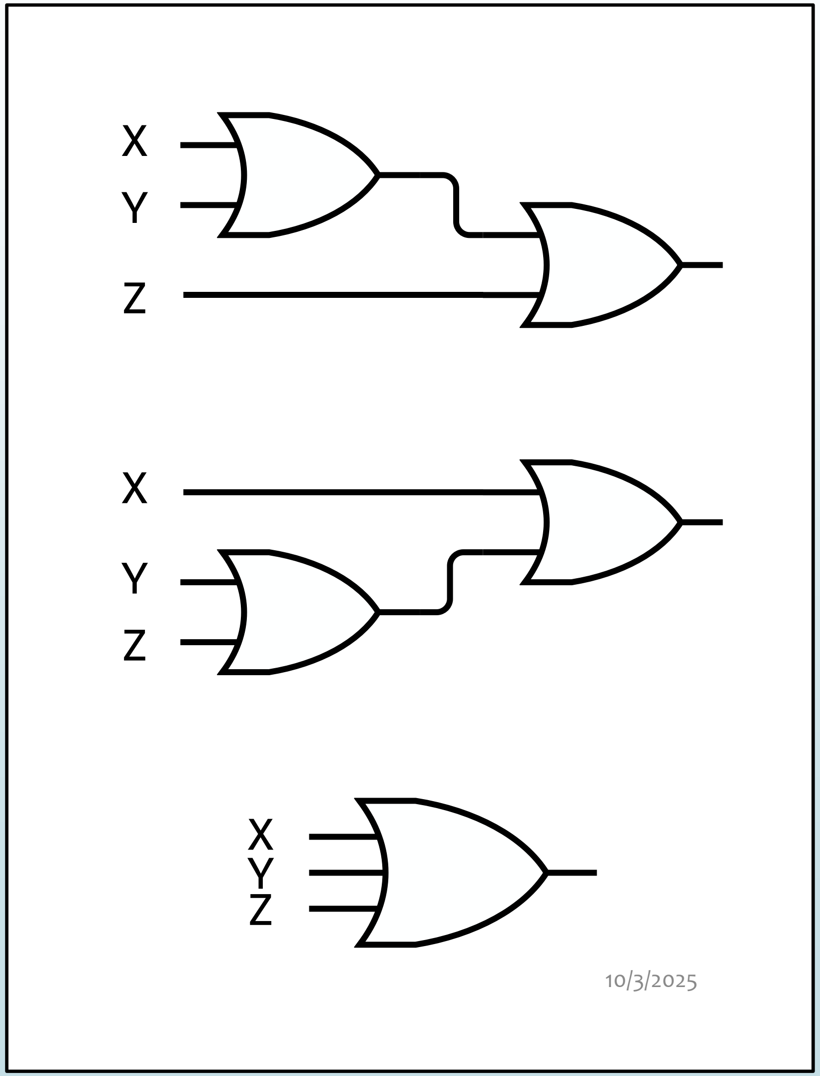
A	B	C
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

Simple Laws

- Involution Law:
  - $(X')' = X$
- Commutative Laws:
  - $X + Y = Y + X$
  - $XY = YX$
- Distributive laws
  - $X(Y + Z) = XY + XZ$
  - $X + YZ = (X + Y) \cdot (X + Z)$

Associative Laws

$$(X + Y) + Z = X + (Y + Z)$$
$$= X + Y + Z$$



Associative Laws

$$(X + Y) + Z = X + (Y + Z)$$
$$= X + Y + Z$$

$$(XY)Z = X(YZ)$$
$$= XYZ$$