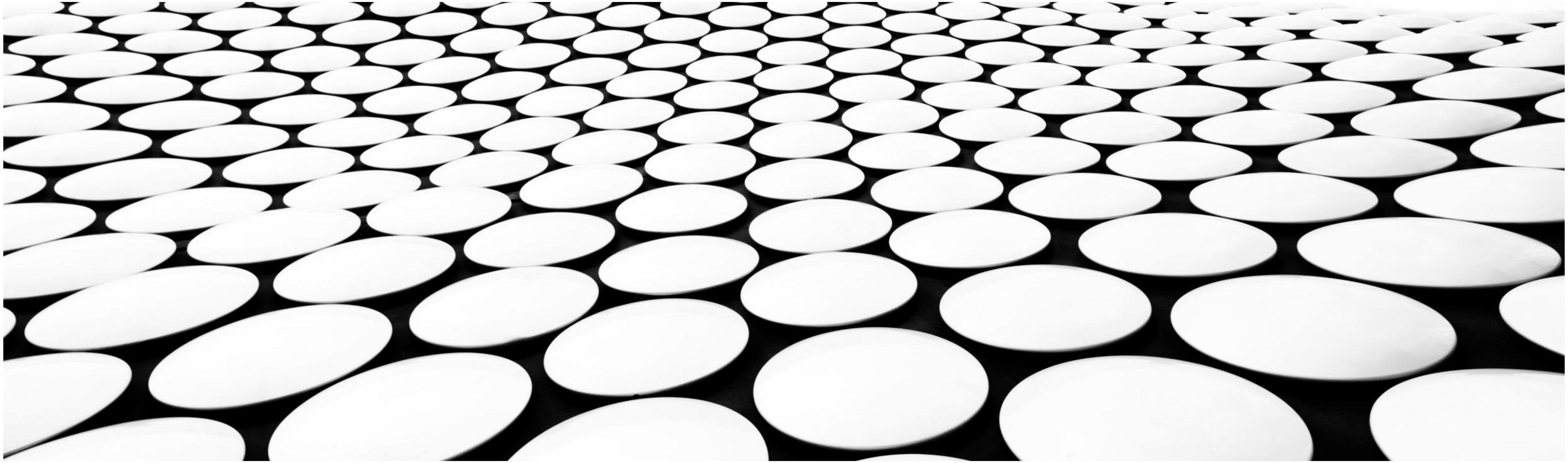
CS5102: FOUNDATIONS OF COMPUTER SYSTEMS

TOPIC-3: BASICS OF LOGIC GATE

DR. ARIJIT ROY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY PATNA



These slides are based on the book: David Money Harris and Sarah L. Harris, Computer Organization and Design

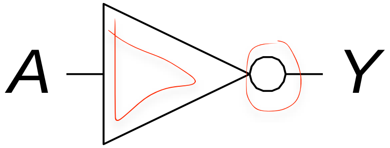
LOGIC GATES

0 1
1 0

- Perform logic functions:
  - inversion (NOT), AND, OR, NAND, NOR, etc.
- Single-input:
  - NOT gate, buffer
- Two-input:
  - AND, OR, XOR, NAND, NOR, XNOR
- Multiple-input

SINGLE-INPUT LOGIC GATES

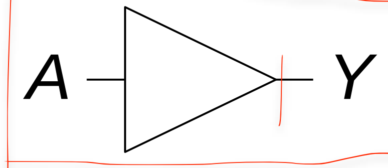
NOT



$$Y = \overline{A}$$

A	Y
0	1
1	0

BUF

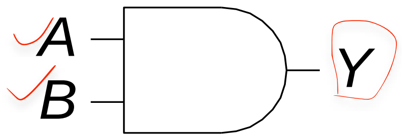


$$Y = A$$

A	Y
0	0
1	1

TWO-INPUT LOGIC GATES

AND



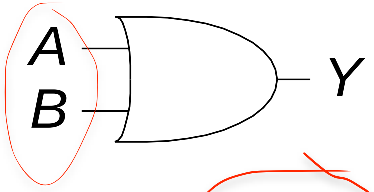
$Y = AB$

A	B	Y
0	0	
0	1	
1	0	
1	1	

A.B

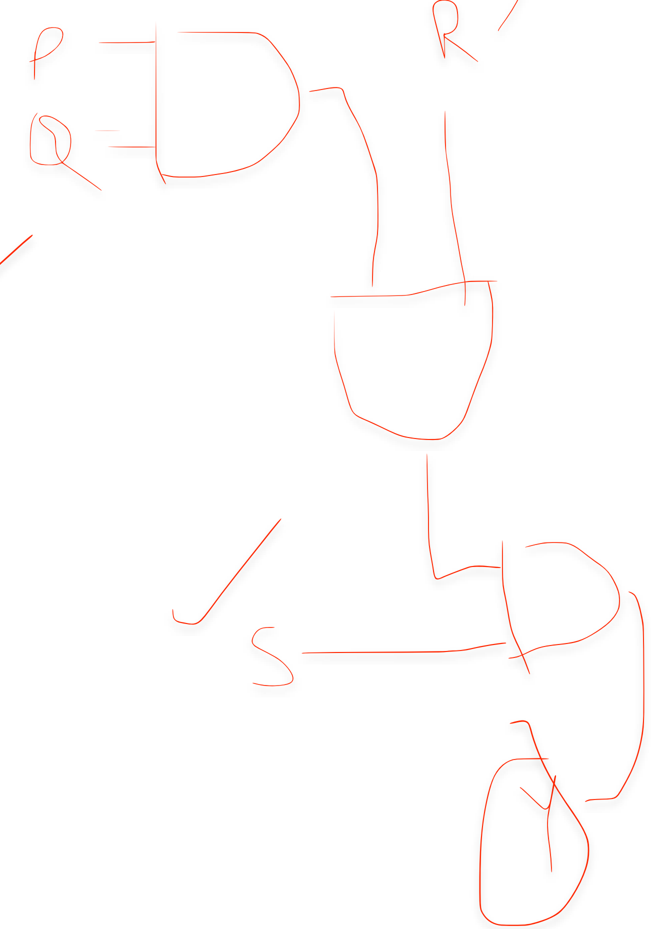
(PQRS

OR



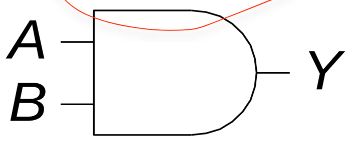
$Y = A + B$

A	B	Y
0	0	
0	1	
1	0	
1	1	



TWO-INPUT LOGIC GATES

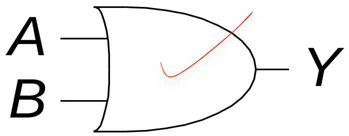
AND



$$Y = AB$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

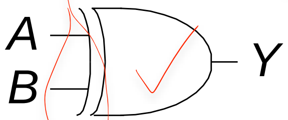


$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

MORE TWO-INPUT LOGIC GATES

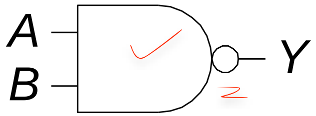
XOR



$Y = A \oplus B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

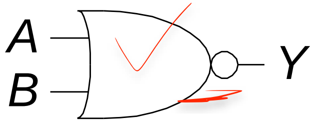
NAND



$Y = \overline{AB}$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

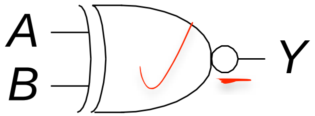
NOR



$Y = \overline{A + B}$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

XNOR

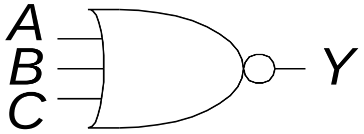


$Y = \overline{A \oplus B}$

A	B	Y
0	0	
0	1	
1	0	
1	1	

MULTIPLE-INPUT LOGIC GATES

NOR3

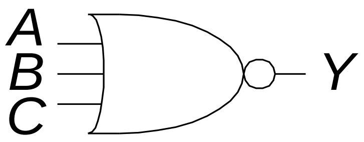


$$Y = \overline{A+B+C}$$

A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

MULTIPLE-INPUT LOGIC GATES

NOR3

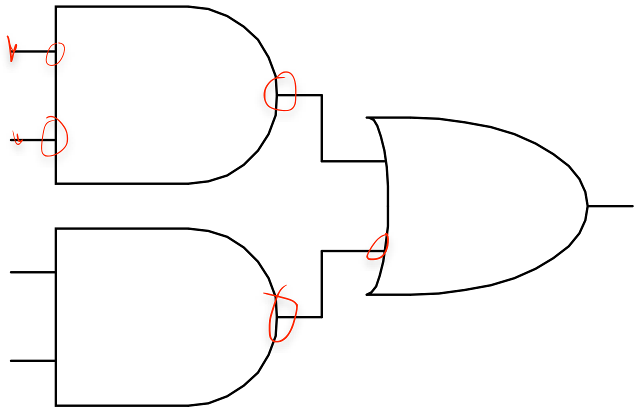


$$Y = \overline{A+B+C}$$

A	B	C	Y
0	0	0	1
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	0

RULES OF COMBINATIONAL COMPOSITION

- Every circuit element is itself combinational
- Every node of the circuit is either designated as an input to the circuit or connects to exactly one output terminal of a circuit element
- The circuit contains no cyclic paths: every path through the circuit visits each circuit node at most once
- Example:



SOME DEFINITIONS

- Complement: variable with a bar over it

  $\bar{A}, \bar{B}, \bar{C}$

- Literal: variable or its complement

  $A, \bar{A}, B, \bar{B}, C, \bar{C}$

- Implicant: product of literals

  $AB\bar{C}, \bar{A}C, BC$

- Minterm: product that includes all input variables

  $AB\bar{C}, A\bar{B}\bar{C}, ABC$

- Maxterm: sum that includes all input variables

  $(A+\bar{B}+C), (\bar{A}+\bar{B}+\bar{C}), (\bar{A}+B+C)$

A — $\bar{A}$

S — $\bar{S}$

PQR

ADC

SUM-OF-PRODUCTS (SOP) FORM

- All Boolean equations can be written in SOP form
- Each row in a truth table has a *minterm*
- A minterm is a product (AND) of literals
- Each minterm is TRUE for that row (and only that row)
- The function is formed by ORing the minterms for which the output is TRUE
- Thus, a sum (OR) of products (AND terms)

A	B	Y	minterm
0	0	0	$\overline{A}\ \overline{B}$
0	1	1	$\overline{A}\ B$
1	0	0	$A\ \overline{B}$
1	1	1	$A\ B$

$Y = \mathrm{F}(A, B) =$

SUM-OF-PRODUCTS (SOP) FORM

- All Boolean equations can be written in SOP form
- Each row in a truth table has a *minterm*
- A minterm is a product (AND) of literals
- Each minterm is TRUE for that row (and only that row)
- The function is formed by ORing the minterms for which the output is TRUE
- Thus, a sum (OR) of products (AND terms)

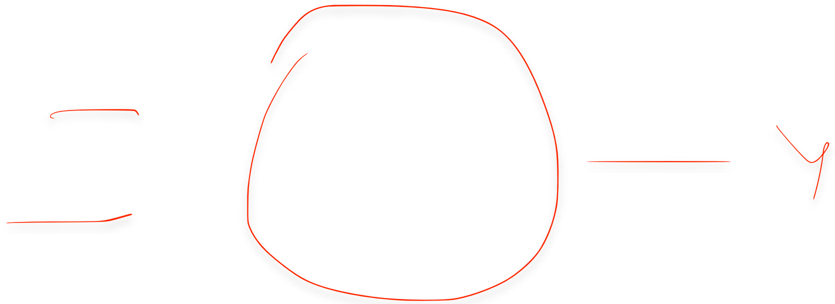
A	B	Y	minterm
0	0	0	$\overline{A}\ \overline{B}$
0	1	1	$\overline{A}\ B$
1	0	0	$A\ \overline{B}$
1	1	1	$A\ B$

$Y = \mathrm{F}(A, B) =$

SUM-OF-PRODUCTS (SOP) FORM

- All Boolean equations can be written in SOP form
- Each row in a truth table has a *minterm*
- A minterm is a product (AND) of literals
- Each minterm is TRUE for that row (and only that row)
- The function is formed by ORing the minterms for which the output is TRUE
- Thus, a sum (OR) of products (AND terms)

A	B	Y	minterm
0	0	0	$\overline{A}\,\overline{B}$
0	1	1	$\overline{A}\,B$
1	0	0	$A\,\overline{B}$
1	1	1	$A\,B$



$$Y = \mathrm{F}(A, B) = \overline{A}B + AB$$

PRODUCT-OF-SUM (POS) FORM

- All Boolean equations can be written in POS form
- Each row in a truth table has a *maxterm*
- A maxterm is a sum (OR) of literals
- Each maxterm is FALSE for that row (and only that row)
- The function is formed by ANDing the maxterms for which the output is FALSE
- Thus, a product (AND) of sums (OR terms)

A	B	Y	maxterm
0	0	0	$A + B$
0	1	1	$A + \overline{B}$
1	0	0	$\overline{A} + B$
1	1	1	$\overline{A} + \overline{B}$

$$Y = \mathrm{F}(A, B) = (A + B)(\overline{A} + B)$$

BOOLEAN EQUATIONS EXAMPLE

- You are going to the cafeteria for lunch and you won't eat corndogs (C)
  - You won't eat lunch ($\overline{E}$)
  - If it's not open ($\overline{O}$) or
  - If they only serve corndogs (C)
- Write a truth table for determining if you will eat lunch (E).

O	C	E
0	0	
0	1	
1	0	
1	1	

BOOLEAN EQUATIONS EXAMPLE

- You are going to the cafeteria for lunch
  - You won't eat lunch ($\overline{E}$)
  - If it's not open ($\overline{O}$) or
  - If they only serve corndogs (C)
- Write a truth table for determining if you will eat lunch (E).

O	C	E
0	0	0
0	1	0
1	0	1
1	1	0

SOP & POS FORM

- SOP – sum-of-products

O	C	E	minterm
0	0		$\overline{O}\ \overline{C}$
0	1		$\overline{O}\ C$
1	0		$O\ \overline{C}$
1	1		$O\ C$

- POS – product-of-sums

O	C	Y	maxterm
0	0		$O + C$
0	1		$O + \overline{C}$
1	0		$\overline{O} + C$
1	1		$\overline{O} + \overline{C}$

SOP & POS FORM

- SOP – sum-of-products

O	C	E	minterm
0	0	0	$\overline{O}\ \overline{C}$
0	1	0	$\overline{O}\ C$
1	0	1	$O\ \overline{C}$
1	1	0	$O\ C$

$$Y = O\overline{C}$$

- POS – product-of-sums

O	C	E	maxterm
0	0	0	$O + C$
0	1	0	$O + \overline{C}$
1	0	1	$\overline{O} + C$
1	1	0	$\overline{O} + \overline{C}$

$$Y = (O + C)(O + \overline{C})(\overline{O} + \overline{C})$$

BOOLEAN ALGEBRA

- Set of axioms and theorems to simplify Boolean equations
- Like regular algebra, but in some cases simpler because variables can have only two values (1 or 0)
- Axioms and theorems obey the principles of duality:
  - ANDs and ORs interchanged, 0’s and 1’s interchanged

BOOLEAN AXIOMS

	Axiom		Dual	Name
A1	$B = 0$ if $B \neq 1$	A1′	$B = 1$ if $B \neq 0$	Binary field
A2	$\overline{0} = 1$	A2′	$\overline{1} = 0$	NOT
A3	$0 \bullet 0 = 0$	A3′	$1 + 1 = 1$	AND/OR
A4	$1 \bullet 1 = 1$	A4′	$0 + 0 = 0$	AND/OR
A5	$0 \bullet 1 = 1 \bullet 0 = 0$	A5′	$1 + 0 = 0 + 1 = 1$	AND/OR

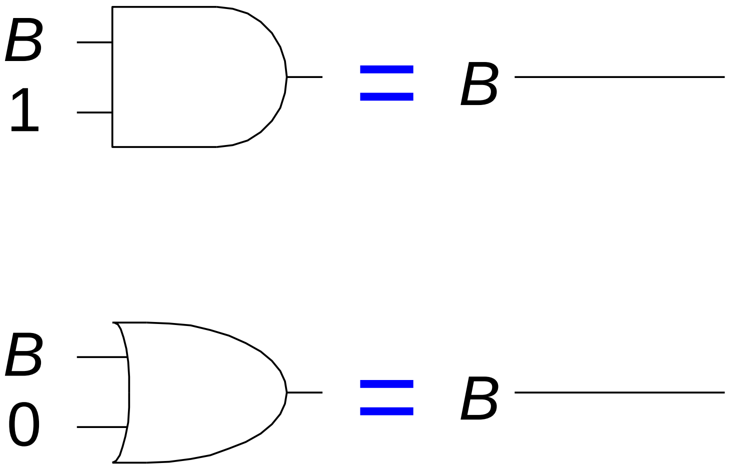
	Theorem		Dual	Name
T1	$B \bullet 1 = B$	T1′	$B + 0 = B$	Identity
T2	$B \bullet 0 = 0$	T2′	$B + 1 = 1$	Null Element
T3	$B \bullet B = B$	T3′	$B + B = B$	Idempotency
T4	$\overline{\overline{B}} = B$			Involution
T5	$B \bullet \overline{B} = 0$	T5′	$B + \overline{B} = 1$	Complements

T1: IDENTITY THEOREM

- $B \bullet 1 =$
- $B + 0 =$

T1: IDENTITY THEOREM

- $B \bullet 1 = B$
- $B + 0 = B$

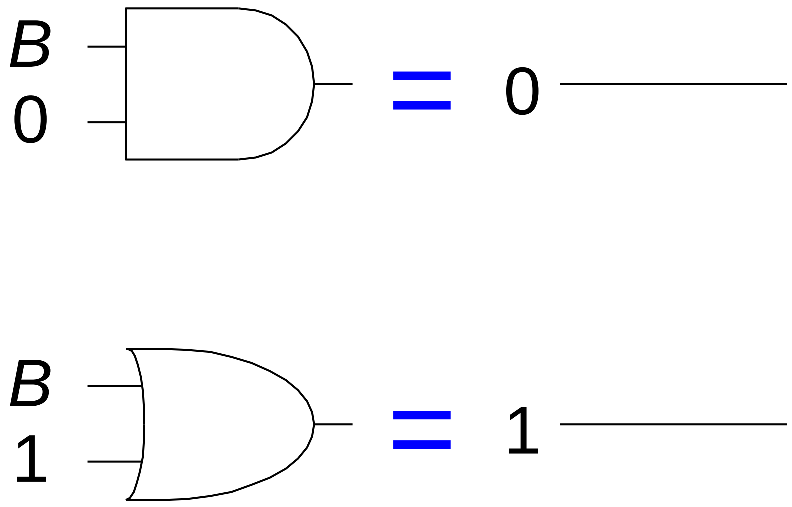


T2: NULL ELEMENT THEOREM

- $B \bullet 0 =$
- $B + 1 =$

T2: NULL ELEMENT THEOREM

- $B \bullet 0 = 0$
- $B + 1 = 1$

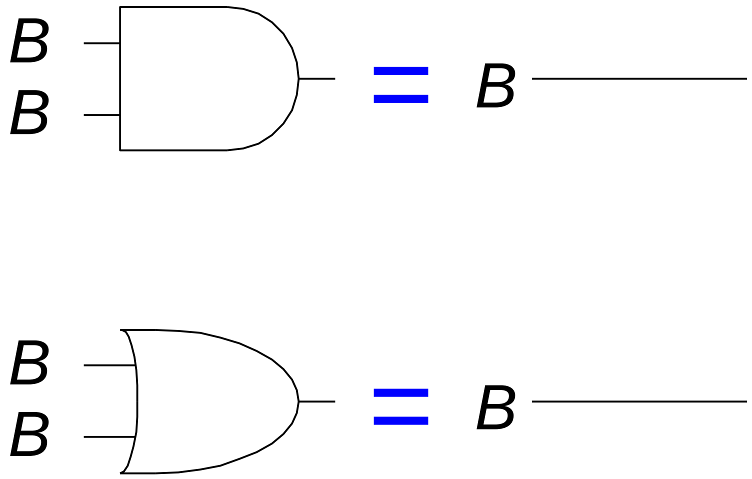


T3: IDEMPOTENCY THEOREM

- $B \bullet B =$
- $B + B =$

T3: IDEMPOTENCY THEOREM

- $B \bullet B = B$
- $B + B = B$

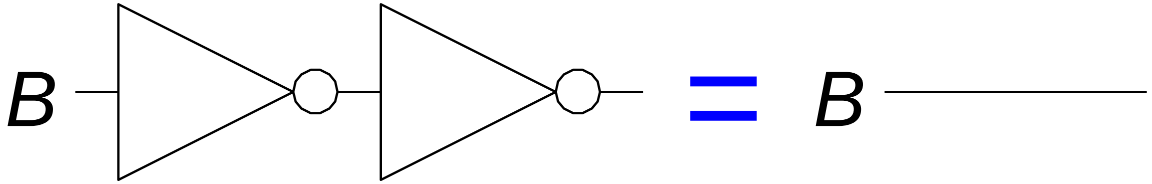


T4: IDENTITY THEOREM

- $\overline{\overline{B}} =$

T4: IDENTITY THEOREM

- $\overline{\overline{B}} = B$

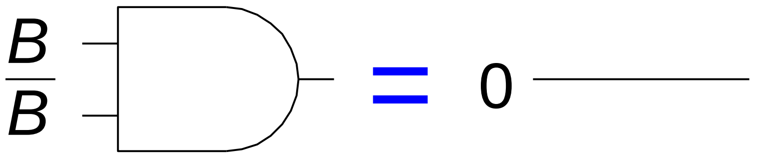


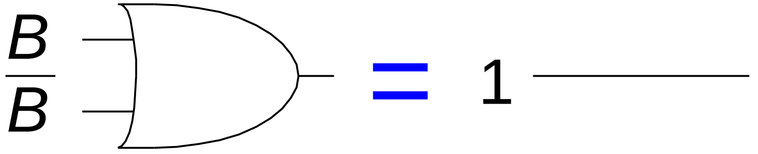
T5: COMPLEMENT THEOREM

- $B \bullet \overline{B} =$
- $B + \overline{B} =$

T5: COMPLEMENT THEOREM

- $B \bullet \overline{B} = 0$
- $B + \overline{B} = 1$





BOOLEAN THEOREMS: SUMMARY

	Theorem		Dual	Name
T1	$B \bullet 1 = B$	T1′	$B + 0 = B$	Identity
T2	$B \bullet 0 = 0$	T2′	$B + 1 = 1$	Null Element
T3	$B \bullet B = B$	T3′	$B + B = B$	Idempotency
T4		$\bar{\bar{B}} = B$		Involution
T5	$B \bullet \overline{B} = 0$	T5′	$B + \overline{B} = 1$	Complements

BOOLEAN THEOREMS OF SEVERAL VARIABLES

	Theorem		Dual	Name
T6	$B \bullet C = C \bullet B$	T6′	$B + C = C + B$	Commutativity
T7	$(B \bullet C) \bullet D = B \bullet (C \bullet D)$	T7′	$(B + C) + D = B + (C + D)$	Associativity
T8	$(B \bullet C) + B \bullet D = B \bullet (C + D)$	T8′	$(B + C) \bullet (B + D) = B + (C \bullet D)$	Distributivity
T9	$B \bullet (B + C) = B$	T9′	$B + (B \bullet C) = B$	Covering
T10	$(B \bullet C) + (B \bullet \overline{C}) = B$	T10′	$(B + C) \bullet (B + \overline{C}) = B$	Combining
T11	$(B \bullet C) + (\overline{B} \bullet D) + (C \bullet D)$ $= B \bullet C + \overline{B} \bullet D$	T11′	$(B + C) \bullet (\overline{B} + D) \bullet (C + D)$ $= (B + C) \bullet (\overline{B} + D)$	Consensus
T12	$\overline{B_0 \bullet B_1 \bullet B_2 \ldots}$ $= (\overline{B_0} + \overline{B_1} + \overline{B_2} \ldots)$	T12′	$\overline{B_0 + B_1 + B_2 \ldots}$ $= (\overline{B_0} \bullet \overline{B_1} \bullet \overline{B_2})$	De Morgan's Theorem

SIMPLIFYING BOOLEAN EXPRESSIONS: EXAMPLE 1

$$Y = \overline{A}B + AB$$

SIMPLIFYING BOOLEAN EXPRESSIONS: EXAMPLE 1

$$Y = \bar{A}B + AB$$

$$= B(\bar{A} + A) \qquad \text{T8}$$

$$= B(1) \qquad \text{T5'}$$

$$= B \qquad \text{T1}$$

SIMPLIFYING BOOLEAN EXPRESSIONS: EXAMPLE 2

$$Y = A(AB + ABC)$$

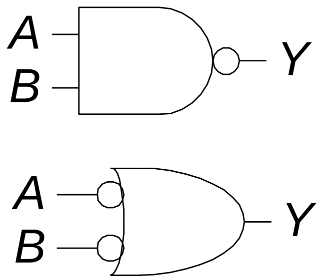
SIMPLIFYING BOOLEAN EXPRESSIONS: EXAMPLE 2

$Y = A(AB + ABC)$

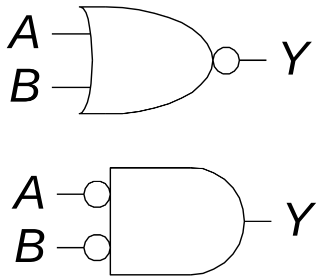
$= A(AB(1 + C))$	T8
$= A(AB(1))$	T2’
$= A(AB)$	T1
$= (AA)B$	T7
$= AB$	T3

DEMORGAN’S THEOREM

$$Y = \overline{AB} = \overline{A} + \overline{B}$$



$$Y = \overline{A + B} = \overline{A} \bullet \overline{B}$$



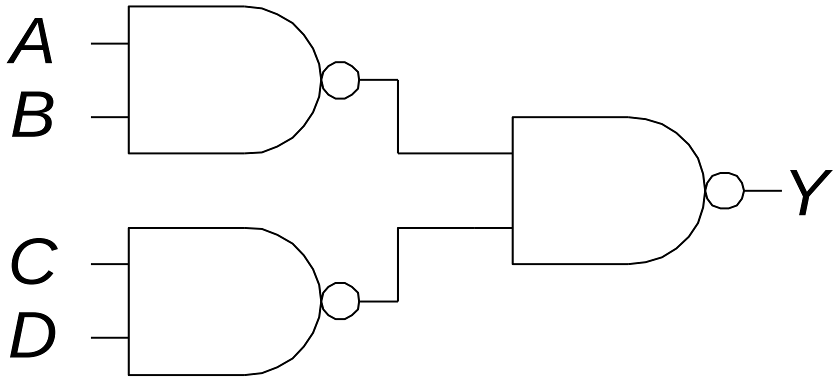
BUBBLE PUSHING

- Pushing bubbles backward (from the output) or forward (from the inputs) changes the body of the gate from AND to OR or vice versa.
- Pushing a bubble from the output back to the inputs puts bubbles on all gate inputs.

- Pushing bubbles on *all* gate inputs forward toward the output puts a bubble on the output and changes the gate body.

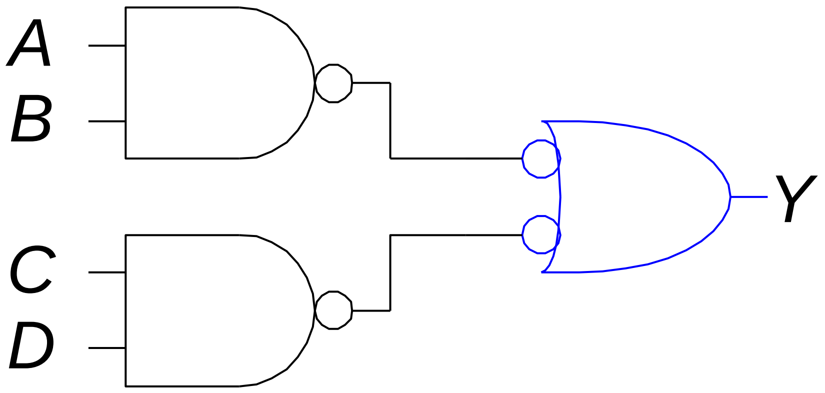
BUBBLE PUSHING

- What is the Boolean expression for this circuit?



BUBBLE PUSHING

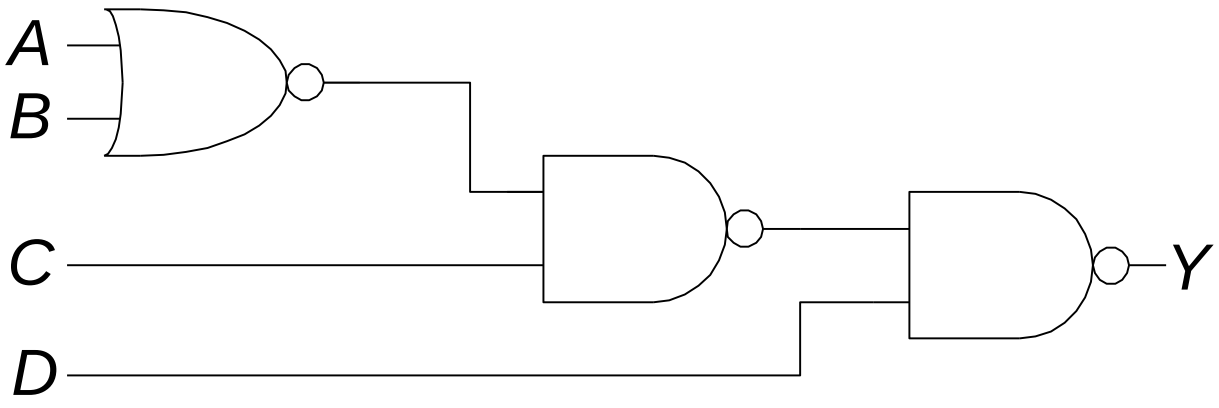
➢ What is the Boolean expression for this circuit?



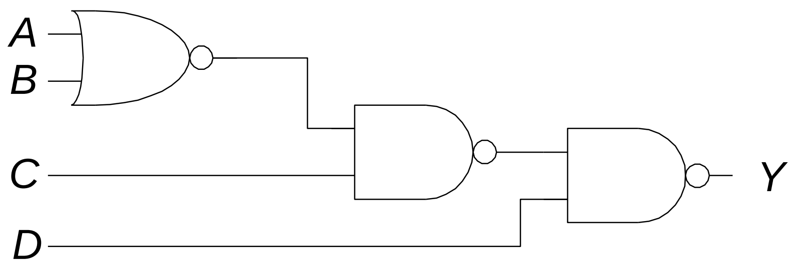
$$Y = AB + CD$$

BUBBLE PUSHING RULES

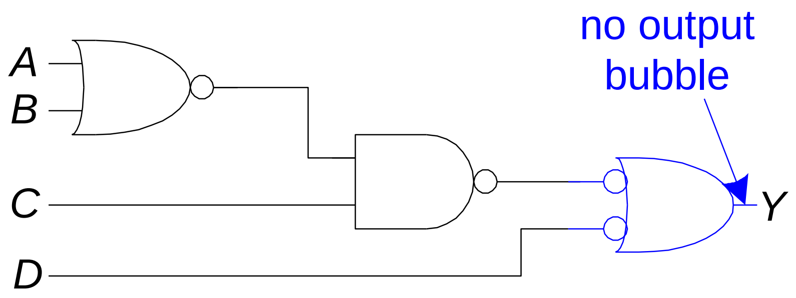
- Begin at the output of the circuit and work toward the inputs.
- Push any bubbles on the final output back toward the inputs.
- Draw each gate in a form so that bubbles cancel.



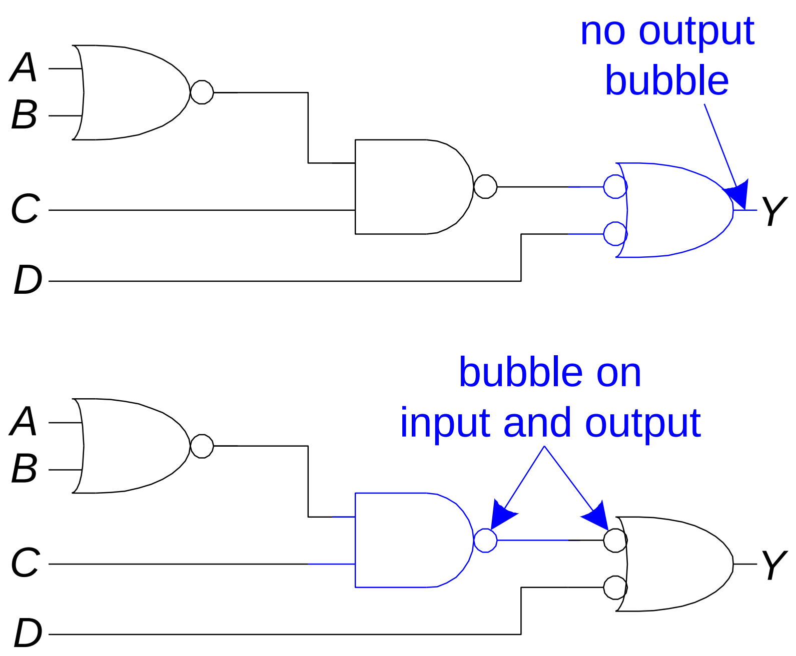
BUBBLE PUSHING EXAMPLE



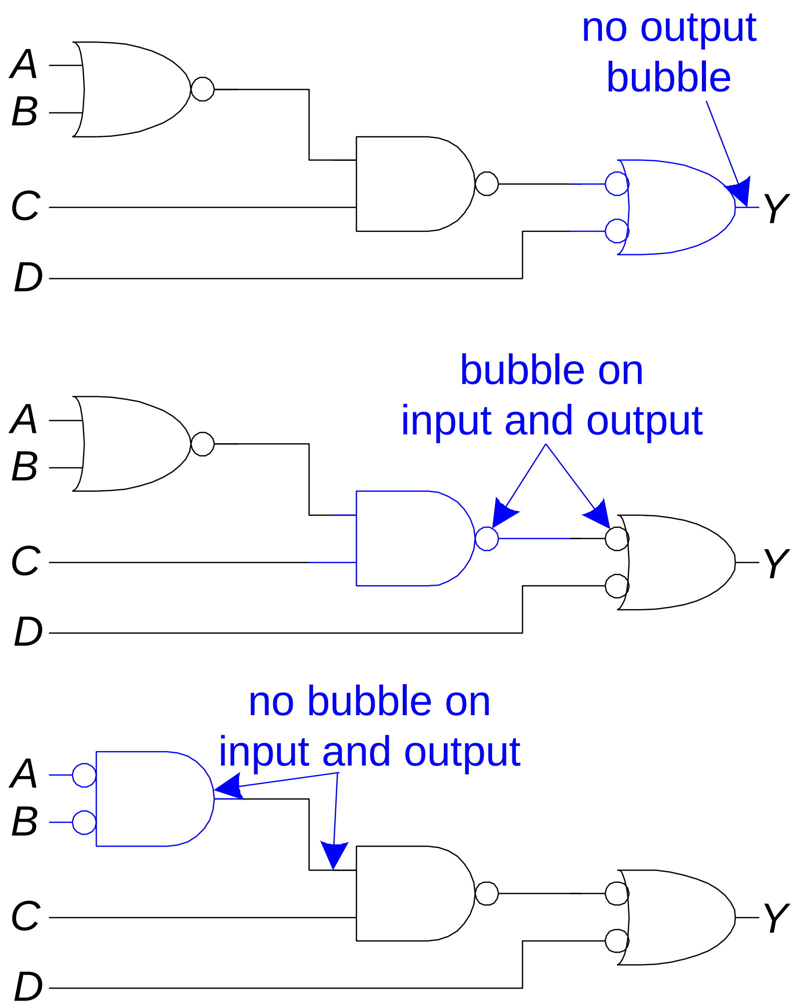
BUBBLE PUSHING EXAMPLE



BUBBLE PUSHING EXAMPLE



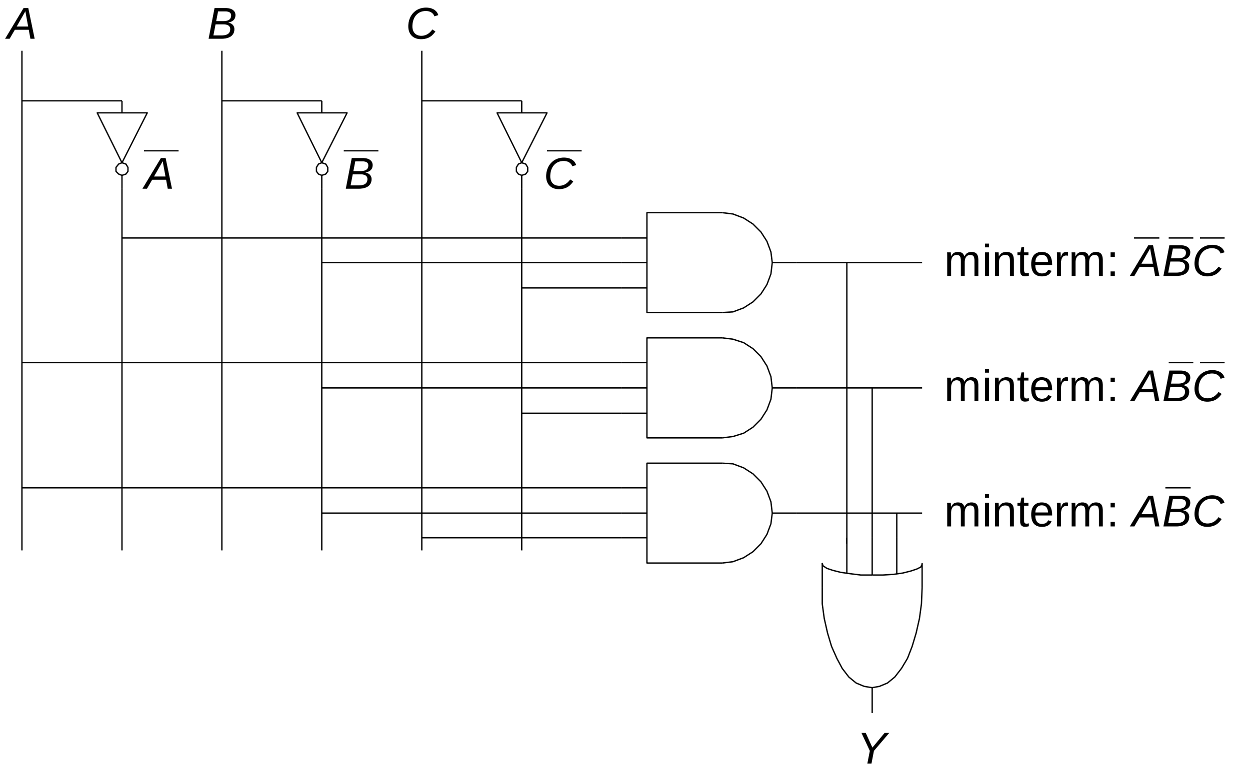
BUBBLE PUSHING EXAMPLE



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

FROM LOGIC TO GATES

- Two-level logic: ANDs followed by ORs
- Example: $Y = \overline{A}\overline{B}\overline{C} + A\overline{B}\overline{C} + A\overline{B}C$



THANK YOU!