

[T]EE2026 DIGITAL DESIGN

---

# TUTORIAL 3: LOGIC GATES

GU JING

ELEGUJI@NUS.EDU.SG

Example:  $F(x,y,z) = \bar{x} + y\bar{z}$ , find canonical form

(Method 1) via truth table

Canonical minterm and maxterm:

$$F(x,y,z) = \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z + \bar{x}y\bar{z} + \bar{x}yz + xy\bar{z}$$

$$\bar{F}(x,y,z) = (\bar{x}+y+z)(\bar{x}+y+\bar{z})(\bar{x}+\bar{y}+\bar{z})$$

(Method 2) via Postulates and Theorems

$$F(x,y,z) = \bar{x} + y\bar{z} = \bar{x} \cdot 1 \cdot 1 + 1 \cdot y \cdot \bar{z}$$

$$= \bar{x}(y+\bar{y})(z+\bar{z}) + (x+\bar{x})y\bar{z}$$

$$= \bar{x}(yz + y\bar{z} + \bar{y}z + \bar{y}\bar{z}) + xy\bar{z} + \bar{x}y\bar{z}$$

$$= \bar{x}yz + \bar{x}y\bar{z} + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z} + \underline{\bar{x}y\bar{z}} + \underline{\bar{x}y\bar{z}}$$

repeat

$$F(x,y,z) = \overline{\overline{\bar{x} + y\bar{z}}} = \overline{\bar{x} \cdot \bar{y}\bar{z}} = \overline{\bar{x}(\bar{y} + \bar{z})} = \overline{\bar{x}\bar{y} + \bar{x}\bar{z}}$$

$$= (\bar{x} + y) \cdot (\bar{x} + \bar{z}) = (\bar{x} + y + \cancel{z})(\bar{x} + y + \cancel{\bar{z}})(\bar{x} + \bar{y} + \bar{z})$$

repeat

**Truth table:**

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

## Six Huntington postulates:

There are 6 Huntington Postulates that define the Boolean Algebra:

1. Closure - For all elements  $x$  and  $y$  in the set  $\mathcal{B}$

- i.  $x + y$  is an element of  $\mathcal{B}$  and
- ii.  $x \cdot y$  is an element of  $\mathcal{B}$

2. There exists a 0 and 1 element in  $\mathcal{B}$ , such that

- i.  $x + 0 = x$
- ii.  $x \cdot 1 = x$

3. Commutative Law

- i.  $x + y = y + x$
- ii.  $x \cdot y = y \cdot x$

4. Distributive Law

- i.  $x \cdot (y + z) = x \cdot y + x \cdot z$  ( $\cdot$  over  $+$ )
- ii.  $x + (y \cdot z) = (x + y) \cdot (x + z)$  ( $+$  over  $\cdot$ )

5. For every element  $x$  in the set  $\mathcal{B}$ , there exists an element  $\bar{x}$  in the set  $\mathcal{B}$ , such that

- i.  $x + \bar{x} = 1$
- ii.  $x \cdot \bar{x} = 0$

( $\bar{x}$  is called the **complement** of  $x$ )

6. There exist at least two distinct elements in the set  $\mathcal{B}$

# The Three Operators in Two-Valued Boolean Algebra (B={0,1})

**OR:**  $A + B$ 

$A$	$B$	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

**AND:**  $A \cdot B$ 

$A$	$B$	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

**NOT:**  $\bar{A}$ 

$A$	$\bar{A}$
0	1
1	0

$$A = 0 \rightarrow \bar{A} = 1$$

$$A = 1 \rightarrow \bar{A} = 0$$

$$0 + 0 = 0$$

$$0 \cdot 0 = 0$$

$$0 + 1 = 1$$

$$0 \cdot 1 = 0$$

$$1 + 0 = 1$$

$$1 \cdot 0 = 0$$

$$1 + 1 = 1$$

$$1 \cdot 1 = 1$$

Priority: NOT has highest precedence, followed by AND and OR  
 $\text{NOT}(A \cdot B + C) = \text{NOT}((A \cdot B) + C)$

#	Theorem		
1	$A + A = A$	$A \cdot A = A$	Tautology Law
2	$A + 1 = 1$	$A \cdot 0 = 0$	Union Law
3	$\overline{(\bar{A})} = A$		Involution Law
4	$\begin{aligned} A + (B + C) \\ = (A + B) + C \end{aligned}$	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	Associative Law
5	$\overline{A + B} = \bar{A} \cdot \bar{B}$	$\overline{A \cdot B} = \bar{A} + \bar{B}$	De Morgan's Law
6	$A + A \cdot B = A$	$A \cdot (A + B) = A$	Absorption Law
7	$A + \bar{A} \cdot B = A + B$	$A \cdot (\bar{A} + B) = A \cdot B$	
8	$AB + A\bar{B} = A$	$(A + B)(A + \bar{B}) = A$	Logical adjacency
9	$\begin{aligned} AB + \bar{A}C + BC \\ = AB + \bar{A}C \end{aligned}$	$\begin{aligned} (A + B)(\bar{A} + C)(B + C) \\ = (A + B)(\bar{A} + C) \end{aligned}$	Consensus Law

**SOP:** Sum of Products, e.g.  $F = ABC + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}$

**POS:** Product of Sums, e.g.  $F = (A + B + C)(\bar{A} + \bar{B} + C)(\bar{A} + B + \bar{C})$

**Minterm** is a product term that contains **all variables** in the function

**Maxterm** is a sum term that contains **all variables** in the function

A	B	C	F	Minterm	Maxterm
0	0	0	0	$\bar{A} \cdot \bar{B} \cdot \bar{C}$	$A + B + C$
0	0	1	0	$\bar{A} \cdot \bar{B} \cdot C$	$A + B + \bar{C}$
0	1	0	1	$\bar{A} \cdot B \cdot \bar{C}$	$A + \bar{B} + C$
0	1	1	0	$\bar{A} \cdot B \cdot C$	$A + \bar{B} + \bar{C}$
1	0	0	1	$A \cdot \bar{B} \cdot \bar{C}$	$\bar{A} + B + C$
1	0	1	1	$A \cdot \bar{B} \cdot C$	$\bar{A} + B + \bar{C}$
1	1	0	1	$A \cdot B \cdot \bar{C}$	$\bar{A} + \bar{B} + C$
1	1	1	0	$A \cdot B \cdot C$	$\bar{A} + \bar{B} + \bar{C}$

minterms  
(maxterms) that are  
equal to 1 (0) only  
for a given input

- Minterm
  - AND all the variables
  - If the variable in truth table is “0”, take its complement in the minterm
- Maxterm
  - OR all the variables
  - If the variable in truth table is “1”, take its complement in the maxterm

**Canonical Form:** if the boolean function is expressed as a **sum of minterms (CSOP)**, or a **product of maxterms (CPOS)**.

**Any Boolean function can be obtained from a given truth table and expressed in either CSOP or CPOS** (choose CSOP if truth table has fewer 1's, choose CPOS if truth table has fewer 0's).

Q1. Use algebraic manipulation to find the MSOP for

$$f = X_1X_3 + X_1\overline{X}_2 + \overline{X}_1X_2X_3 + \overline{X}_1\overline{X}_2\overline{X}_3$$

Write the Verilog code that describes the above function as a module  
(use dataflow description style).

Q1 Ans:

$$\begin{aligned}
 F &= X_1X_3 + X_1\overline{X}_2 + \overline{X}_1X_2X_3 + \overline{X}_1\overline{X}_2\overline{X}_3 \\
 &= X_3(X_1 + \overline{X}_1X_2) + \overline{X}_2(X_1 + \overline{X}_1X_3) \\
 &= X_3(X_1 + X_2) + \overline{X}_2(X_1 + \overline{X}_3) \quad \{\text{using } A + \overline{A}B = A+B\} \\
 &= X_1X_3 + X_2X_3 + X_1\overline{X}_2 + \overline{X}_2\overline{X}_3 \\
 &= X_1X_3 + X_2X_3 + \overline{X}_2\overline{X}_3 \quad \{\text{using } AB + \overline{A}C + BC = AB + \overline{A}C: A=X_3, B=X_1, C=X_2\} \\
 \text{Or } &X_1\overline{X}_2 + X_2X_3 + \overline{X}_2\overline{X}_3 \quad \{\text{using } AB + \overline{A}C + BC = AB + \overline{A}C: A=X_2, B=X_1, C=X_3\}
 \end{aligned}$$

```

module func (x1, x2, x3, F);
    input x1, x2, x3;
    output F;
    assign F = x1&x3 | x1&~x2 | ~x1&x2&x3 | ~x1&~x2&~x3;
endmodule

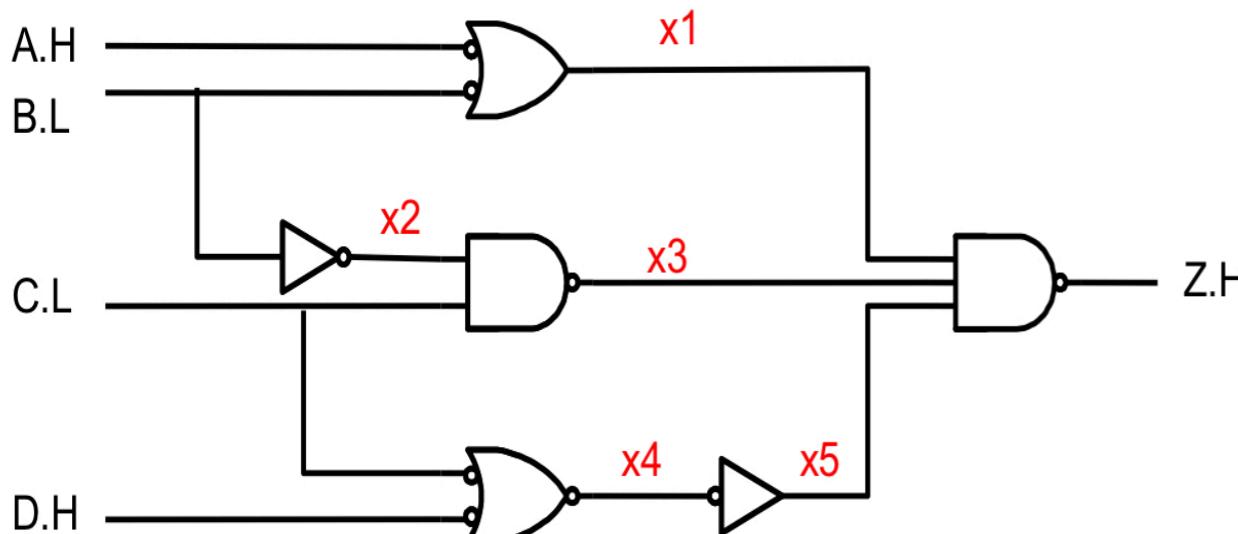
```

Q2. For circuits 1 and 2 below:

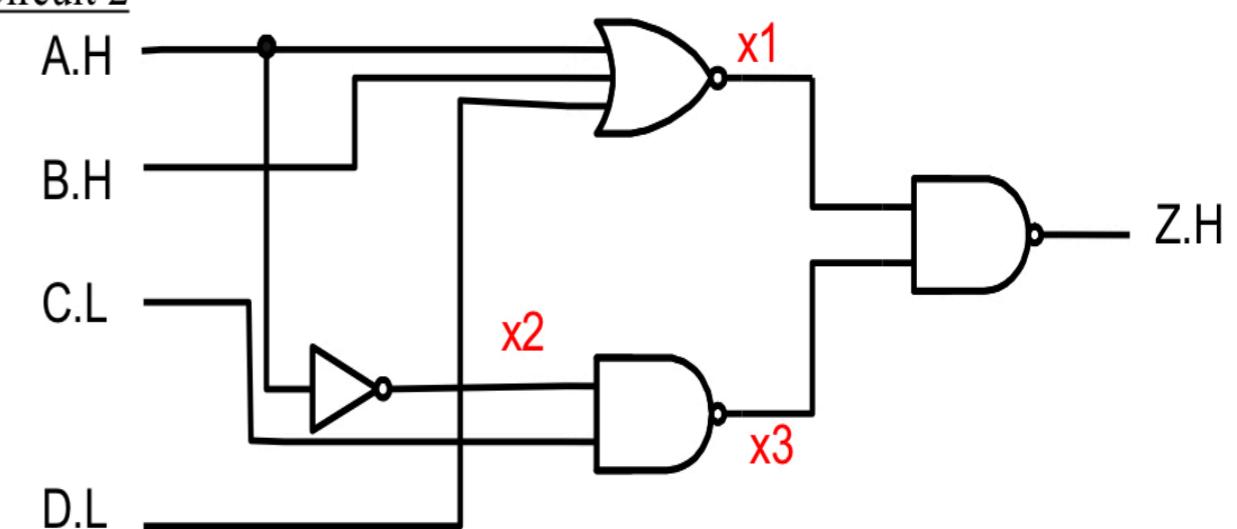
- (a) Fill in all the intermediate signal names using the positive logic convention
- (b) Find the logic expression for Z for the two cases
- (c) Write the Verilog code that describes the above function as a module (use structural description style). In the description, assume that only the Verilog primitives are available (no need to describe):

`not(out,in)`, `nand (out,in1,in2,...)`, `nor(out,in1,in2,...)`.

Circuit 1



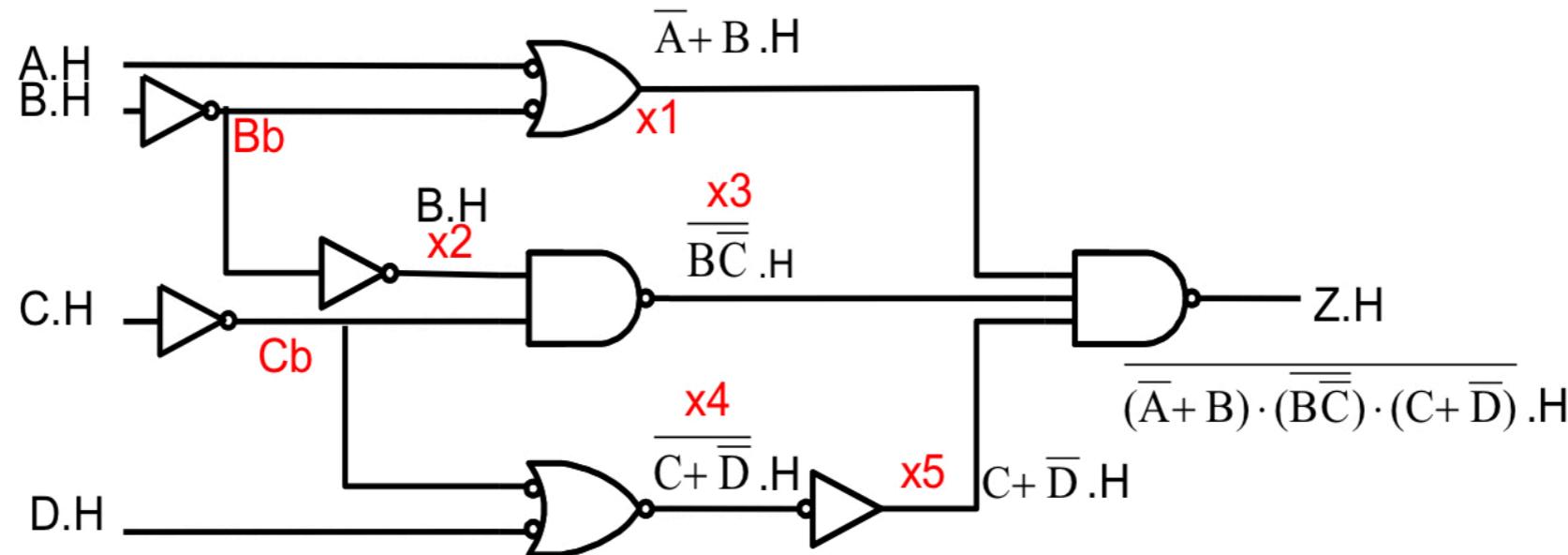
Circuit 2



## [T]EE2026 Tutorial 3: Logic Gates

Q2 Ans.

Circuit 1

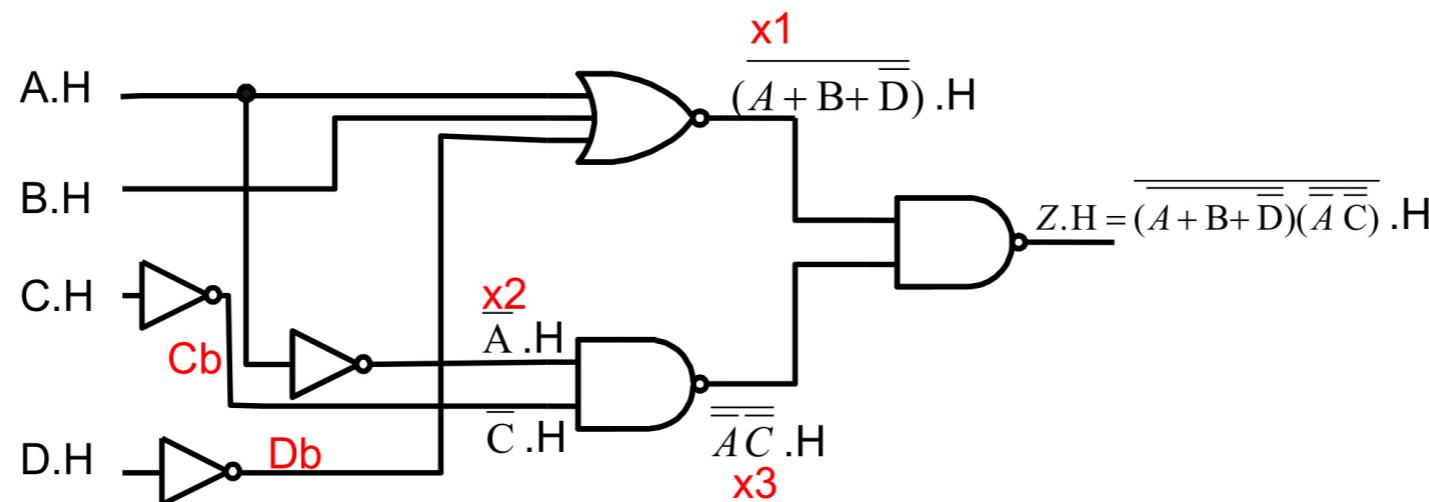


```
module func (Z, A, B, C, D);
    input A, B, C, D;
    output Z;
    wire x1, x2, x3, x4, x5, Bb, Cb;
    not u1(Bb,B); // inverter gate to generate Bb=NOT(B) (omitted inverter)
    not u2(Cb,C); // inverter gate to generate Cb=NOT(C) (omitted inverter)
    nand u3(x1,A,Bb); // NAND2 gate, instance u3 (push bubbles in gate @top-left)
    not u4(x2,Bb); // inverter gate
    nand u5(x3,x2,Cb); // NAND2 gate
    and u6(x4,Cb,d); // AND2 gate (push bubbles in gate at bottom-left)
    not u7(x5,x4); // inverter gate
    nand u8(Z,x1, x3,x5); // NAND3 gate
endmodule
```

## [T]EE2026 Tutorial 3: Logic Gates

Q2 Ans.

Circuit 2



```
module func(Z,A,B,C,D);
    input A, B, C, D;
    output Z;
    wire x1, x2, x3, Cb, Db;
    not u1(Cb,C); // inverter gate to generate Cb=NOT(C) (see omitted inverter)
    not u2(Db,D); // inverter gate to generate Db=NOT(D) (see omitted inverter)
    nor u3(x1,A,B,Db); // NOR3 gate, instance u3
    not u4(x2,A); // inverter gate
    nand u5(x3,x2,Cb); // NAND2 gate
    nand u6(Z,x1,x3); // NAND2 gate
    nand u6(Z,x1,x3); // NAND2 gate
endmodule
```

Q3. Design a logic circuit implementing the following Boolean function, using only NOR gates:

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

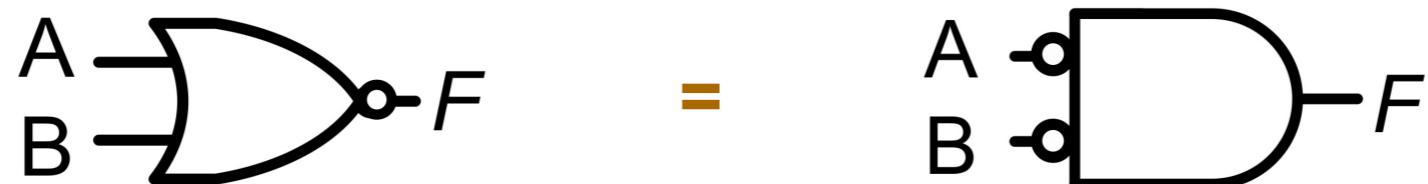
where  $\oplus$  represents the 2-operand XOR operation. Use alternate gate representation if necessary for clear circuit diagrams.

Write the Verilog code that describes the above function as a module (use dataflow description style).

Q3 Ans.

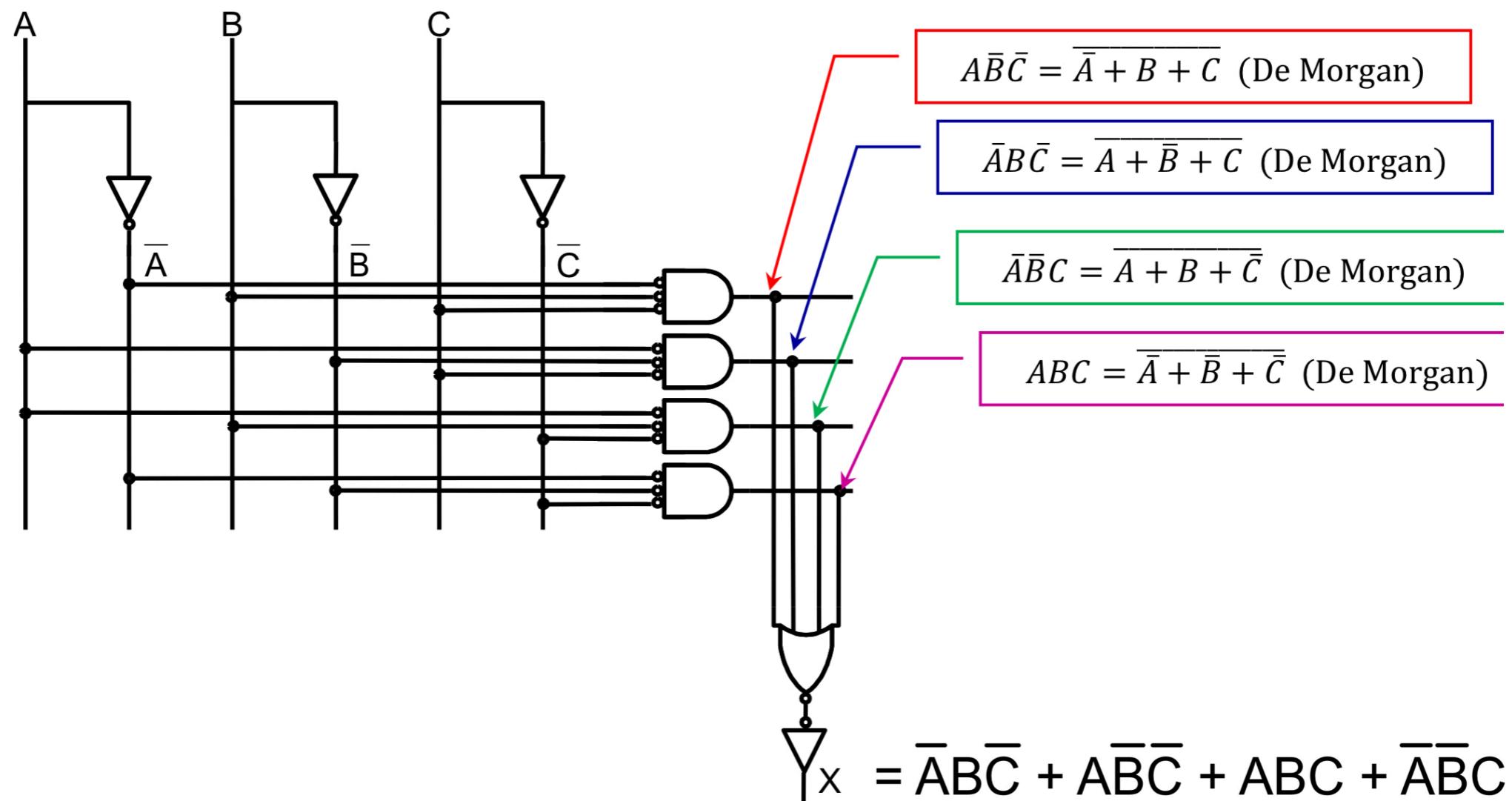
$$\begin{aligned}
 X &= A \oplus B \oplus C \\
 &= (A \oplus B) \oplus C \\
 &= (\bar{A}B + A\bar{B}) \oplus C \\
 &= (\bar{A}B + A\bar{B}) \cdot \bar{C} + (\bar{A}B + A\bar{B}) \cdot C \\
 &= \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + (A + \bar{B}) \cdot (\bar{A} + B) \cdot C \\
 &= \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + (A\bar{A} + AB + \bar{A}\bar{B} + B\bar{B}) \cdot C \\
 &= \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + ABC + \bar{A}\bar{B}C
 \end{aligned}$$

NOR gate:  $F = \overline{\overline{A} + \overline{B}} = \overline{\overline{A}\overline{B}}$



## [T]EE2026 Tutorial 3: Logic Gates

Q3 Ans.



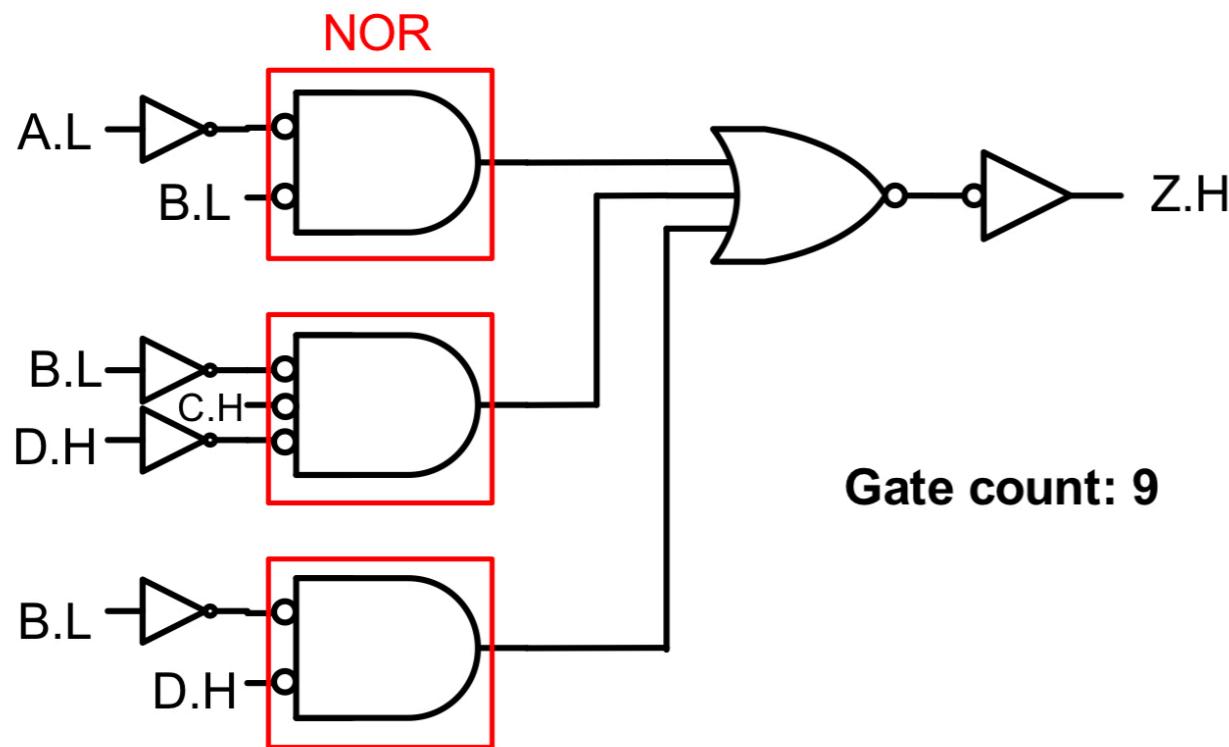
```
module func(A,B,C,X);
    input A, B, C;
    output X;
    assign X = A ^ B ^ C;
    // or X = ~A & B & ~C | A & ~B & ~C | A & B & C | ~A & ~B & C;
endmodule
```

Q4. Design a circuit to realize  $Z = \bar{A}B + \bar{B}\bar{C}D + \bar{B}\bar{D}$  in the positive logic convention. A and B are active low signals while C, D and Z are active high. Use a minimum number of gates, and draw clear circuit diagrams with alternate gate representations as required.

Write the Verilog code that describes the above function as a module (use dataflow description style).

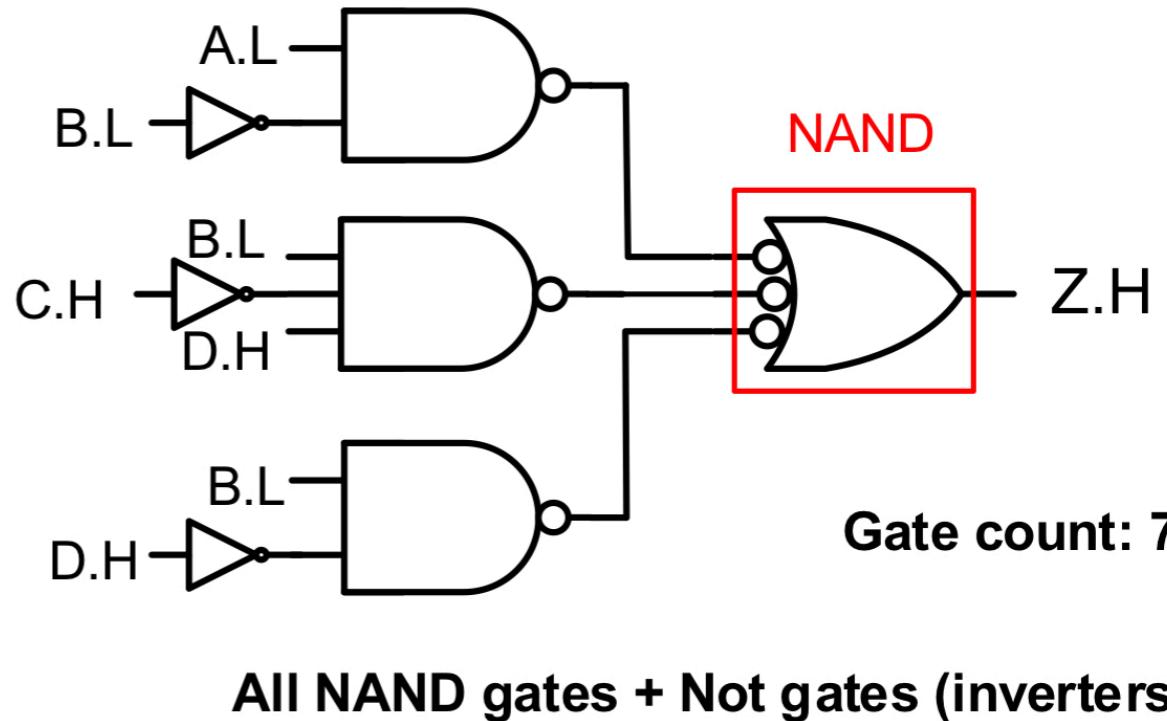
Q4 Ans.  $Z = \overline{A}\overline{B} + \overline{B}\overline{C}\overline{D} + \overline{B}\overline{D}$

Using NOR gate:



All NOR gates + Not gates (inverters)

Using NAND gate:



All NAND gates + Not gates (inverters)

```
module func(Z,A,B,C,D);
    input A, B, C, D;
    output Z;
    assign Z = A & ~B | B & ~C & ~D | B & ~D;
endmodule
```

**THE END**

**For Consultation: [eleguji@nus.edu.sg](mailto:eleguji@nus.edu.sg)**  
**Office: E4-03-10**