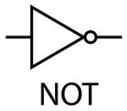
Higher Logic

NOT Gate



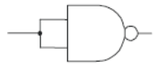
The output Z is the opposite of the input A. If the input is 0 (low) then the output will be 1 (high).

A	Z
0	1
1	0

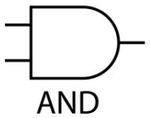
Boolean expression

$\overline{A} = Z$

NAND Equivalent



AND Gate



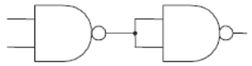
The output Z will only be high (1) when both input A and B are high (1).

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

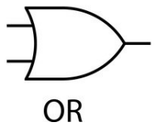
Boolean expression

$A \cdot B = Z$

NAND Equivalent



OR Gate



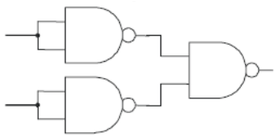
The output Z will be high (1) when either input A or B or both inputs are high (1).

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

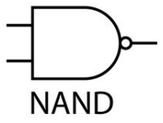
Boolean expression

$A + B = Z$

NAND Equivalent



NAND



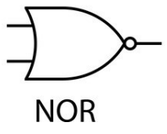
The NAND is the opposite of the AND. The output Z will only be low (0) when both input A and B are high (1).

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

Boolean expression

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

NOR Gate



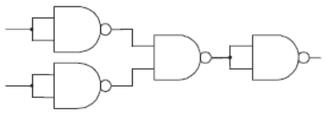
The NOR is the opposite of the OR. The output Z will only be high (1) when input A and B are low (0).

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

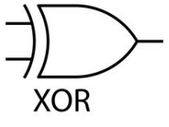
Boolean expression

$\overline{A + B} = Z$

NAND Equivalent



XOR Gate



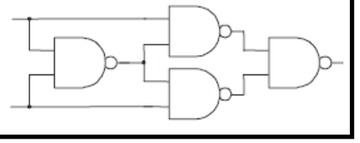
The output Z will only be high (1) when only one input is high (1).

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

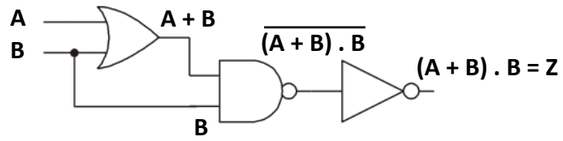
Boolean expression

$A \oplus B = Z$

NAND Equivalent



Developing Boolean expressions from circuits



Developing Boolean expressions from truth tables

A	B	C	Z	
0	0	0	0	
0	0	1	1	$\overline{A} \cdot \overline{B} \cdot C$
0	1	0	0	
0	1	1	1	$\overline{A} \cdot B \cdot C$
1	0	0	0	
1	0	1	1	$A \cdot \overline{B} \cdot C$
1	1	0	0	
1	1	1	1	$A \cdot B \cdot C$

1 - Look for the rows of the truth table where output Z is high (1).

2 - Create an expression for that row.

3 - Create an overall expression for all of the rows using the OR expression in between.

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

Developing Boolean expressions from word problems

Input/Output	Operation
alarm sounds	Z=1
gates open	A=0
maximum loading exceeded	B=1
button C pressed	C=1
button D pressed	D=1

Start by figuring out what each of the inputs need to be doing for the alarm to sound. If it must happen then there needs to be . between the inputs in the expression.

A proposed design has the following specification (all conditions must be met).

The alarm (Z):

- will not sound unless the gate (A) is closed **From the table A = 1**
- will not sound when the weight on the platform exceeds the maximum loading (B) **From the table B = 0**
- will sound when either button (C) or (D) is pressed, but not when both are pressed at the same time. **C⊕D**

(a) Complete the Boolean equation for the alarm system in operation.

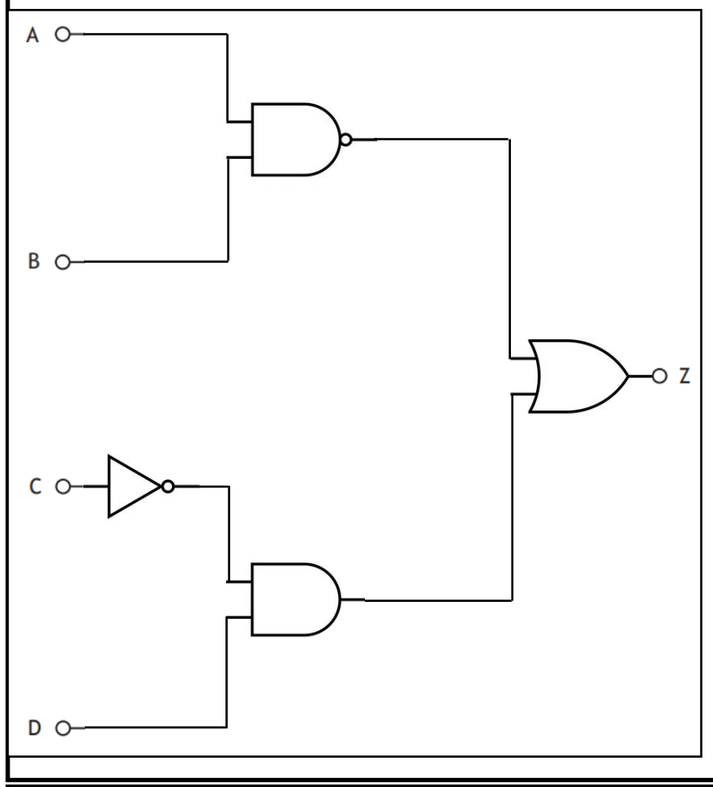
$Z =$ **$A \cdot \overline{B} \cdot (C + D)$**

Developing a circuit from a Boolean expression

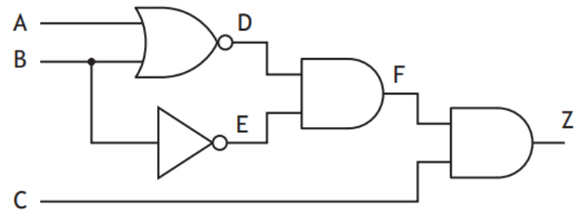
$$Z = (\overline{A \cdot B}) + (\overline{C} \cdot D)$$

1 - Start by figuring out how many logic gates you will need and what type they are.

- In this example there are 4 inputs.
- 2 inputs are going into a NAND gate and the other 2 inputs into an AND gate. We know this from the . between the expressions in the brackets.
- The output from those gates are going into an OR gate. We know this from the + in between the 2 expressions.
- C is inverted so there needs to be a NOT gate too.



Completing a truth table from a Logic circuit



Complete the truth table for the logic diagram.

Include the intermediate logic values for D, E and F.

A	B	C	D	E	F	Z
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

A	B	C	D	E	F	Z
0	0	0	1	1	1	0
0	0	1	1	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	1	0	0
1	0	1	0	1	0	0
1	1	0	0	0	0	0
1	1	1	0	0	0	0

To complete this type of question you will need to have a sound knowledge of the different logic gates and their truth tables.

Column D is dependant on A and B going through a NOR gate.

A	B	OR	NOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Column E is the opposite of B.

Column F is dependant on D and E going through an AND gate.

Column Z is dependant on F and C.

A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

NAND Equivalent advantages

Fewer IC chips required to make circuits meaning a simpler construction.

Fewer ICs meaning smaller product size.

Fewer ICs means reduced cost.

Buying NAND gates in bulk would lower the cost rather than buying different types of gates to perform the same function.