

# Logical gates

---

# NOT gate

---

Вентиль NOT (инвертор) возвращает сигнал, **противоположный** полученному. Это реализация логического НЕ.

Таблица истинности:

a	$\neg a$
0	1
1	0

# AND gate

---

- Вентиль AND (логическое И) возвращает 1, если на **всех** входах **1**.

a	b	$a \wedge b$
0	0	0
1	0	0
0	1	0
1	1	1

# XOR gate

---

- Вентиль XOR (исключающее ИЛИ) возвращает 1, если **только на одном** из входов 1. То есть, если входные значения были **разные**.

a	b	$a \neq b$
0	0	0
1	0	1
0	1	1
1	1	0

# NAND gate

---

- Он выдаст 1 если **хотя бы** на одном из входов **0**.

a	b	$a \bar{b}$
0	0	1
1	0	1
0	1	1
1	1	0

# OR gate

---

- Вентиль OR (логическое ИЛИ) возвращает 1, если **хотя бы на одном из входов 1**.

a	b	$a \vee b$
0	0	0
1	0	1
0	1	1
1	1	1

# XNOR gate

---

- Вентиль XNOR вернет 1, если на входах сигналы **одинаковые**. Этот вентиль получается из XOR добавлением инвертора на выходе.
- Таблица истинности:

a	b	$a \equiv b$
0	0	1
1	0	0
0	1	0
1	1	1

# NOR

---

- Он возвратит 1, если **на всех** входах **0**.

a	b	$a \nabla b$
0	0	1
1	0	0
0	1	0
1	1	0



```
#include<iostream>
using namespace std;
int a,b;
void function()
{
a = a+b;
b = a*2;
}
void main()
{
a=1; b=3;
cout <<a<<" "<<b<<endl;
function();
cout <<a<<" "<<b<<endl;
}
```

```
#include<iostream>
using namespace std;
int function(int x, int y)
{
return x+y;
}
int main()
{
int a=2, b=3;
a = function(a, b);
b = function(function(a, b), b);
cout <<a<<" "<<b<<endl;
return 0;
}
```

# Theory

---

- +
- -
- \*
- /
- %

# Theory

---

- Integer variable
- Floating-point variables
- Logical variable
- Character variable

# Theory

Data	Variable Type
Age of a person	<b>unsigned short</b>
Name of a person	
Gender of a person ('F' or 'M')	
State of a electrical switch (ON or OFF)	
Number of the passengers in an airplane	
Area of a circle	
Temperature	

