Логические основы работы ЭВМ

ЛОГИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРА.

ФОРМЫ МЫШЛЕНИЯ

АЛГЕБРА ЛОГИКИ

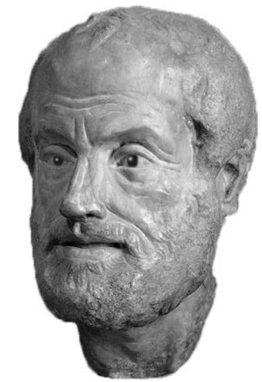
ПОСТРОЕНИЕ ТАБЛИЦ ИСТИННОСТИ.
ЗАКОНЫ ЛОГИКИ

БАЗОВЫЕ ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ
КОМПЬЮТЕРА

Логика. Высказывания.

Логика, высказывания

Логика (др.греч. *λογικος*) – это наука о том, как правильно рассуждать, делать выводы, доказывать утверждения.



Аристотель
(384-322 до н.э.)

Формальная логика отвлекается от конкретного содержания, изучает только истинность и ложность высказываний.

Логическое высказывание – это повествовательное предложение, относительно которого можно однозначно сказать, истинно оно или ложно.

Высказывание или нет?

Сейчас идет дождь.

Жирафы летят на север.

~~История – интересный предмет~~.

У квадрата – 10 сторон и все разные.

Красиво!

В городе N живут 2 миллиона человек.

Который час?

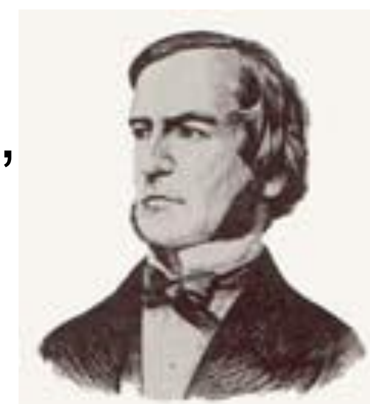
Логика и компьютер

Двоичное кодирование – все виды информации кодируются с помощью 0 и 1.

Задача – разработать оптимальные правила обработки таких данных.

Почему «логика»?
Результат выполнения операции можно представить как истинность (1) или ложность (0) некоторого высказывания.

Джордж Буль разработал основы алгебры, в которой используются только 0 и 1 (алгебра логики, булева алгебра).



- Для описания функционирования аппаратных и программных средств ЭВМ используется *алгебра логики* или, как ее часто называют, *булева алгебра*. Булева алгебра оперирует с логическими переменными, которые могут принимать только два значения: истина и ложь.
- Совокупность значений логических переменных называется *набором переменных*.
- *Логической функцией* от набора логических переменных (аргументов) называется функция, которая может принимать только два значения: истина и ложь. Любая логическая функция может быть задана с помощью таблицы истинности.

- ***Логический элемент компьютера*** — это часть электронной логической схемы, которая реализует элементарную логическую функцию.
- Логическими элементами компьютеров являются электронные схемы И, ИЛИ, НЕ, И—НЕ, ИЛИ—НЕ и другие (называемые также вентилями), а также триггер.
- С помощью этих схем можно реализовать любую логическую функцию, описывающую работу устройств компьютера.
- Высокий уровень обычно соответствует значению "истина" ("1"), а низкий — значению "ложь" ("0").
- Каждый логический элемент имеет свое условное обозначение, которое выражает его логическую функцию

- ***Работу логических элементов описывают с помощью таблиц истинности.***
- ***Таблица истинности*** это табличное представление логической схемы (операции), в котором перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.

Основные понятия

- Под ***высказыванием*** понимается повествовательное предложение, о котором можно сказать, что оно истинно или ложно. Высказывания будем обозначать заглавными латинскими буквами.
- Каждое высказывание кроме своего смыслового значения имеет и ***истинностное значение***, которое есть истина (сокращенно И) или ложь (сокращенно Л).

Обозначение высказываний

A – Сейчас идет дождь.
B – Форточка открыта.

простые высказывания (элементарные)

! **Любое высказывание может быть ложно (0) или истинно (1).**

Составные высказывания строятся из простых с помощью логических связок (операций) «**и**», «**или**», «**не**», «**если … то**», «**тогда и только тогда**» и др.

A и B	Сейчас идет дождь и открыта форточка.
A или не B	Сейчас идет дождь или форточка закрыта.
если A, то B	Если сейчас идет дождь, то форточка открыта.
A тогда и только тогда, когда B	Дождь идет тогда и только тогда, когда открыта форточка.

Примеры

- ***Пример 1.***
- D : 5 — простое число,
- E : 12 = 22 • 3,
- F : Земля — спутник Луны,
- G : Функция y = cos x является четной

- Высказывания D, E, F, G являются примерами ***простых высказываний***, высказывание H — пример сложного высказывания.
- ***Сложные высказывания*** образуются из простых с помощью логических операций (связок).
- Определим следующие ***логические операции***: отрицание, конъюнкцию, дизъюнкцию, импликацию и эквиваленцию.

Базовый набор операций

С помощью операций **И, ИЛИ** и **НЕ** можно реализовать любую логическую операцию.

И **ИЛИ** **НЕ**

базовый набор операций

Логические операции

- ***Отрицанием*** высказывания А называется высказывание ¬А(Ā) (читается «не А»), которое истинно тогда и только тогда, когда высказывание Л ложно.

И	Л
Л	И

Операция НЕ (инверсия)

Если высказывание **A** истинно, то «**не A**» ложно, и наоборот.

A	не A
0	1
1	0

также $\overline{\mathbf{A}}$, $\neg\mathbf{A}$, **not A** (Паскаль), **! A** (Си)

таблица истинности операции НЕ

Таблица истинности логического выражения X – это таблица, где в левой части записываются все возможные комбинации значений исходных данных, а в правой – значение выражения X для каждой комбинации.

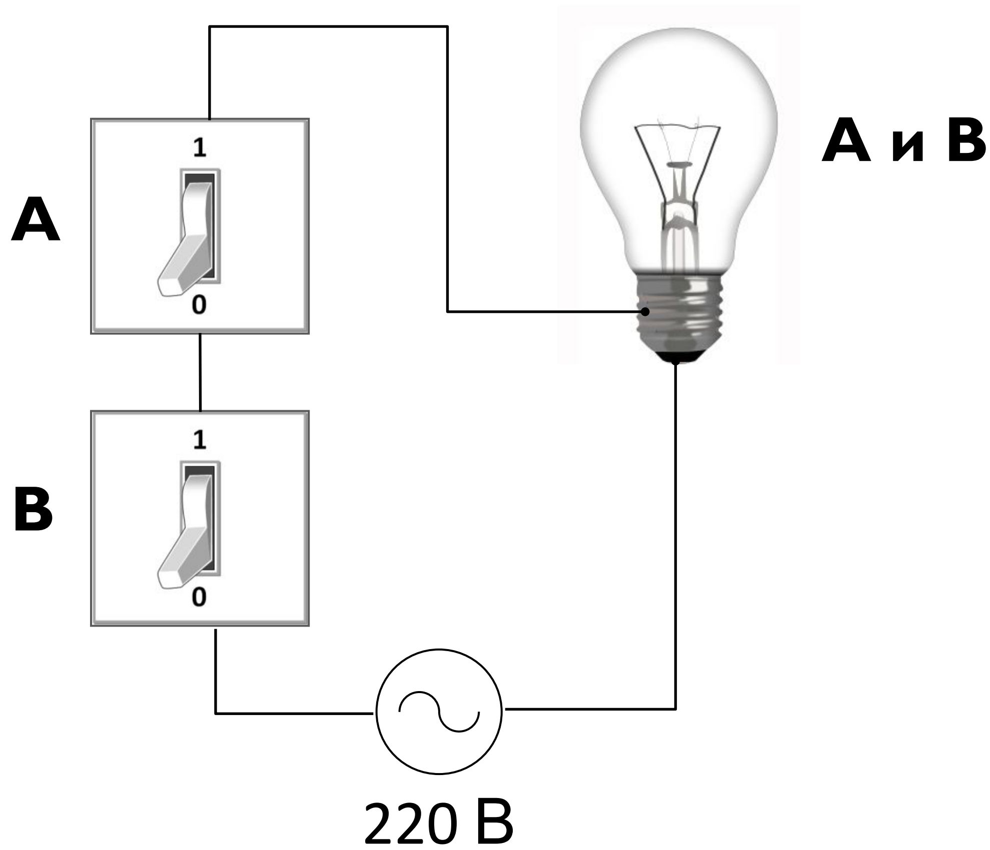
Логические операции

- ***Конъюнкцией*** высказываний A и B называется высказывание $A \wedge B$ (читается «A и B»), которое истинно тогда и только тогда, когда оба высказывания A и B истинны.

<table>
<tr><td colspan="2" rowspan="2">$A \wedge B$</td><td colspan="2">B</td></tr>
<tr><td>И</td><td>Л</td></tr>
<tr><td rowspan="2">A</td><td>И</td><td>И</td><td>Л</td></tr>
<tr><td>Л</td><td>Л</td><td>Л</td></tr>
</table>

Операция И

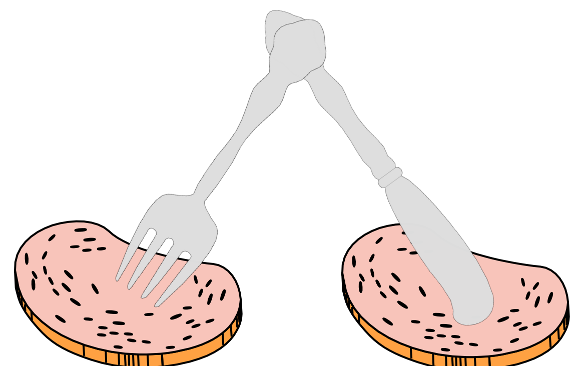
Высказывание «**А и В**» истинно тогда и только тогда, когда **А** и **В** истинны одновременно.



Операция И (логическое умножение, конъюнкция)

	A	B	A и B
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

также: **A · B**, **A ∧ B**, **A and B** (Паскаль), **A && B** (Си)



A ∧ B

конъюнкция – от лат. *conjunctio* — соединение

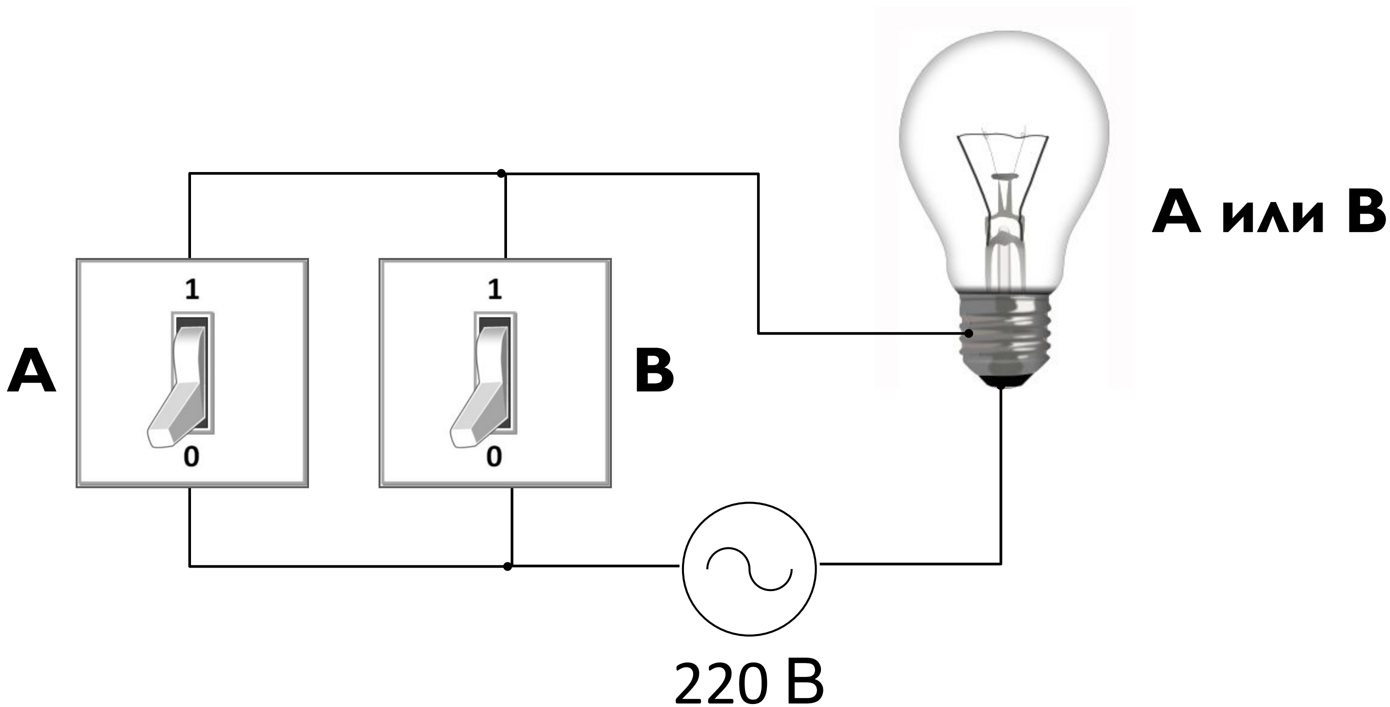
Логические операции

- ***Дизъюнкцией*** высказываний А и В называется высказывание $A \vee B$ (читается «А или В»), которое ложно тогда и только тогда, когда оба высказывания А и В ложны.

$A \vee B$		B	
		И	Л
A	И	И	И
	Л	И	Л

Операция ИЛИ (логическое сложение, дизъюнкция)

Высказывание «**А или В**» истинно тогда, когда истинно **А** или **В**, или оба вместе.



Операция ИЛИ (логическое сложение, дизъюнкция)

A	B	A или B
0	0	0
0	1	1
1	0	1
1	1	1

также: **A+B**, **A ∨ B**,
A or B (Паскаль),
A || B (Си)

дизъюнкция – от лат. *disjunctio* — разъединение

Логические операции

- ***Импликацией*** высказываний A и B называется высказывание $A \to B$ (читается «если A, то B»), которое ложно тогда и только тогда, когда высказывание A истинно, а высказывание B ложно.
- Высказывания A и B в импликации имеют специальные названия: высказывание A называется ***посылкой*** или ***условием***, а высказывание B называется ***следствием*** или ***заключением***.

Логические операции

A→B		B	
		И	Л
A	И	И	Л
	Л	И	И

Импликация («если …, то …»)

Высказывание «**A** → **B**» истинно, если не исключено, что из **A** следует **B**.

A – «Работник хорошо работает».

B – «У работника хорошая зарплата».

A	B	A → B
0	0	1
0	1	1
1	0	0
1	1	1

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

Импликация («если …, то …»)

«Если Вася идет гулять, то Маша сидит дома».

A – «Вася идет гулять».

B – «Маша сидит дома».

$$A \rightarrow B = 1$$

A	B	A → B
0	0	1
0	1	1
1	0	0
1	1	1

? А если Вася не идет гулять?

Маша может пойти гулять (B=0), а может и не пойти (B=1)!

Логические операции

- ***Эквиваленцией*** высказываний A и B называется высказывание A↔ B (читается «A тогда и только тогда, когда B»), которое истинно тогда и только тогда, когда оба высказывания A и B одновременно истинны или ложны.

<table>
<tr><td colspan="2" rowspan="2">A↔B</td><td colspan="2">B</td></tr>
<tr><td>И</td><td>Л</td></tr>
<tr><td rowspan="2">A</td><td>И</td><td>И</td><td>Л</td></tr>
<tr><td>Л</td><td>Л</td><td>И</td></tr>
</table>

Эквивалентность («тогда и только тогда, …»)

Высказывание «**A** ↔ **B**» истинно тогда и только тогда, когда **A** и **B** равны.

A	B	A ↔ B
0	0	1
0	1	0
1	0	0
1	1	1

$$A \leftrightarrow B = \overline{A \oplus B} = A \cdot B + \overline{A} \cdot \overline{B}$$

Операция «исключающее ИЛИ»

Высказывание «**A ⊕ B**» истинно тогда, когда истинно **A** или **B**, но ***не оба одновременно*** (то есть **A ≠ B**).

«Либо пан, либо пропал».

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

также:
A xor B (Паскаль),
A ^ B (Си)

арифметическое сложение, 1+1=2

остаток

сложение по модулю 2: A ⊕ B = (A + B) **mod** 2

Свойства операции «исключающее ИЛИ»

$A \oplus 0 = A$

$A \oplus 1 = \overline{A}$

$A \oplus A = 0$

$(A \oplus B) \oplus B = ?$

$$A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B$$

A	B	$A \cdot \overline{B}$	$\overline{A} \cdot B$	$A \cdot \overline{B} + \overline{A} \cdot B$	$A \oplus B$
0	0	0	0	0	0
0	1	0	1	1	1
1	0	1	0	1	1
1	1	0	0	0	0

Штрих Шеффера, «И-НЕ»

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

A	B	A | B
0	0	1
0	1	1
1	0	1
1	1	0

Базовые операции через «И-НЕ»:

$$\overline{A} = A \mid A \qquad A \cdot B = \overline{A \mid B} = (A \mid B) \mid (A \mid B)$$

$$A + B = \overline{A} \mid \overline{B} = (A \mid A) \mid (B \mid B)$$

Стрелка Пирса, «ИЛИ-НЕ»

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

A	B	$A \downarrow B$
0	0	1
0	1	0
1	0	0
1	1	0

Формализация

Прибор имеет три датчика и может работать, если два из них исправны. Записать в виде формулы ситуацию «авария».

A – «Датчик № 1 неисправен».

B – «Датчик № 2 неисправен».

C – «Датчик № 3 неисправен».

Аварийный сигнал:

X – «Неисправны два датчика».

X – «Неисправны датчики № 1 **и** № 2» **или**
«Неисправны датчики № 1 **и** № 3» **или**
«Неисправны датчики № 2 **и** № 3».

Формализация – это переход к записи на формальном языке!

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

логическая формула

Формулы логики высказываний

- 1) Логические переменные, буквы И и Л являются формулами логики высказываний;
- 2) если A и B — формулы логики высказываний, то формулами логики высказываний являются также выражения: $\bar{A}$, $A \wedge B$, $A \vee B$, $A \rightarrow B$, $A \leftrightarrow B$;
- 3) выражение является формулой логики высказываний тогда и только тогда, когда оно удовлетворяет первому или второму пункту данного определения.

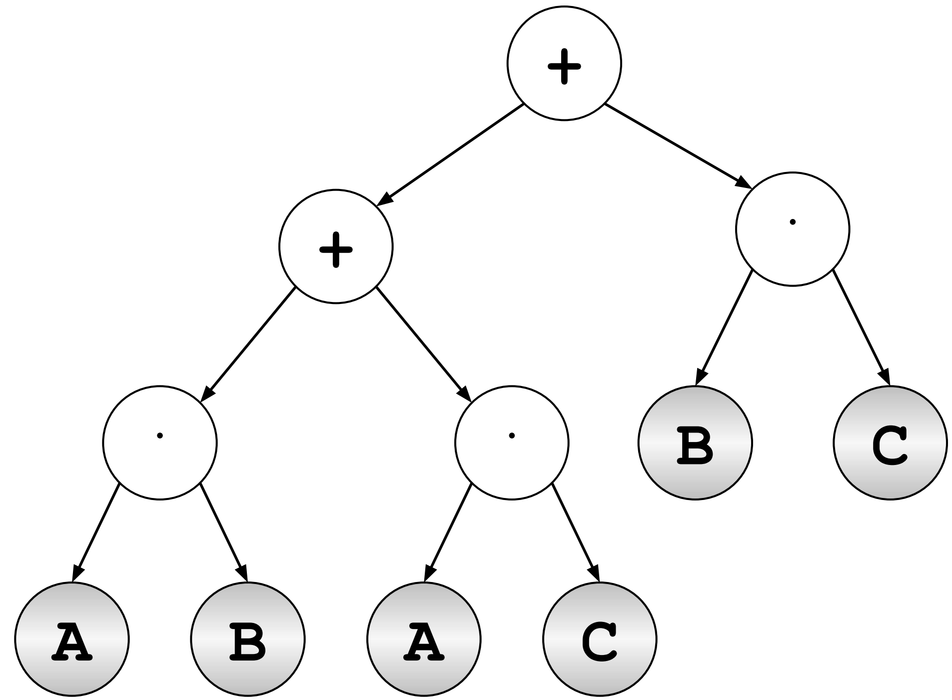
- Формулу можно упростить за счет уменьшения числа скобок в ней, приняв следующие соглашения:
- *1) внешние скобки в формуле можно опускать;*
- *2) внутренние скобки в формуле можно опускать с учетом следующего порядка выполнения действий: отрицание, конъюнкция, дизъюнкция, импликация, эквиваленция.*

Вычисление логических выражений

$$\begin{array}{c} \color{blue}{1\quad 4\quad\quad 2\quad\quad 5\quad\quad 3} \\ \boxed{X = A\cdot B + A\cdot C + B\cdot C} \end{array}$$

Порядок вычислений:

- скобки
- НЕ
- И
- ИЛИ, исключающее ИЛИ
- импликация
- эквивалентность



- Подставляя в формулу вместо переменных их допустимые значения и выполняя указанные в ней действия, находим **истинностное значение формулы**.
- Зависимость истинностных значений формулы от значений входящих в нее переменных наглядно иллюстрируют **истинностные таблицы**.

Составление таблиц истинности

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

	A	B	$A \cdot B$	$\overline{A} \cdot B$	$\overline{B}$	X
0	0	0	0	0	1	1
1	0	1	0	1	0	1
2	1	0	0	0	1	1
3	1	1	1	0	0	1

Логические выражения могут быть:

- **тождественно истинными** (всегда 1, тавтология)
- **тождественно ложными** (всегда 0, противоречие)
- **вычислимыми** (зависят от исходных данных)

- Формула логики высказываний называется ***тождественно истинной (тождественно ложной)***, если при любых значениях входящих в нее переменных ее истинностное значение равно истине (лжи).
- Тождественно истинные формулы называют также ***тавтологиями***, а тождественно ложные — ***противоречиями***.
- Для того, чтобы убедиться в этом, достаточно составить для каждой из формул таблицу истинности.

Составление таблиц истинности

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

	A	B	C	A·B	A·C	B·C	X
0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
2	0	1	0	0	0	0	0
3	0	1	1	0	0	1	1
4	1	0	0	0	0	0	0
5	1	0	1	0	1	0	1
6	1	1	0	1	0	0	1
7	1	1	1	1	1	1	1

Примеры

- ***Пример 1.*** Определить тип формулы

$p \wedge (p \to q) \to q$

		$p \to q$	$(p \to q) \to q$	$p \wedge (p \to q) \to q$
И	И	И	И	И
И	Л	Л	Л	И
Л	И	И	Л	И
Л	Л	И	Л	И

Формула является тавтологией

- Две формулы логики высказываний А и В называются ***равносильными***, если при любом наборе значений переменных, входящих в эти формулы, истинностные значения формул А и В равны.
- То, что формулы А и В равносильны, будем записывать так: $A \equiv B$.

Законы логики высказываний

- *Пусть буквы A, B, C обозначают произвольные формулы логики высказываний. Тогда истинны следующие утверждения:*

$$\left.\begin{array}{l} 1.\ A \wedge A \equiv A, \\ 2.\ A \vee A \equiv A, \end{array}\right\} \text{ — } \textit{свойства идемпотентности}$$

$$\left.\begin{array}{l} 3.\ A \wedge B \equiv B \wedge A, \\ 4.\ A \vee B \equiv B \vee A, \end{array}\right\} \text{ — } \textit{свойства коммутативности}$$

$$\left.\begin{array}{l} 5.\ A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C, \\ 6.\ A \vee (B \vee C) \equiv (A \vee B) \vee C, \end{array}\right\} \text{ — } \textit{свойства ассоциативности}$$

$$\left.\begin{array}{l} 7.\ A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C), \\ 8.\ A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C), \end{array}\right\}$$ — *свойства дистрибутивности*

$$\left.\begin{array}{l} 9.\ \neg(A \wedge B) \equiv \neg A \vee \neg B, \\ 10.\ \neg(A \vee B) \equiv \neg A \wedge \neg B, \end{array}\right\}$$ — *свойства де Моргана*

$$\left.\begin{array}{l} 11.\ A \wedge (A \vee B) \equiv A, \\ 12.\ A \vee (A \wedge B) \equiv A, \end{array}\right\}$$ — *свойства поглощения*

13. $\neg(\neg A) \equiv A,$ — *свойство двойного отрицания*

14. $A \Rightarrow B \equiv \neg B \Rightarrow \neg A,$ — *свойство контрапозиции*

15. $\neg(A \Rightarrow B) \equiv A \wedge \neg B,$ — *свойство отрицания импликаци*

16. $A \wedge \neg A \equiv Л,$ — *свойство противоречия*

17. $A \vee \neg A \equiv И,$ — *свойство исключенного третьего*

18. $A \Rightarrow B \equiv \neg A \vee B,$

19. $A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A),$

20. $A \wedge И \equiv A$,

21. $A \vee И \equiv И$,

22. $A \wedge Л \equiv Л$,

23. $A \vee Л \equiv A$.

Упрощение логических выражений

Шаг 1. Заменить операции $\oplus \rightarrow \leftrightarrow$ на их выражения через **И**, **ИЛИ** и **НЕ**:

$$A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B$$

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

$$A \leftrightarrow B = A \cdot B + \overline{A} \cdot \overline{B}$$

Шаг 2. Раскрыть инверсию сложных выражений по формулам де Моргана:

$$\overline{A \cdot B} = \overline{A} + \overline{B} \qquad \overline{A + B} = \overline{A} \cdot \overline{B}$$

Шаг 3. Используя законы логики, упрощать выражение, стараясь применять закон исключения третьего.

Примеры

- ***Пример 1.*** Доказать, что формула p ^ (p → q) → q является тождественно истинной, используя свойства.

$$p \wedge (p \Rightarrow q) \Rightarrow q \overset{18}{\equiv} p \wedge (\neg p \vee q) \overset{7}{\equiv}$$

$$(p \wedge \neg p) \vee (p \wedge q) \Rightarrow q \overset{16}{\equiv} \text{Л} \vee (p \wedge q) \Rightarrow q \overset{3}{\equiv}$$

$$(p \wedge q) \vee \text{Л} \Rightarrow q \overset{23}{\equiv} p \wedge q \Rightarrow q \overset{18}{\equiv}$$

$$\neg (p \wedge q) \vee q \overset{9}{\equiv} (\neg p \vee \neg q) \vee q \overset{6}{\equiv} \neg p \vee (\neg q \vee q) \overset{17}{\equiv}$$

21

$$\neg p \vee И \equiv И$$

формула является тождественно истинной

Примеры

- ***Пример 2.*** Доказать равносильность формул $p \to \neg(q \vee p) \vee \neg(r \vee q)$ и $\neg(p \wedge (q \vee r))$.

$$p \Rightarrow \neg(q \vee p) \vee \neg(r \vee q) \overset{9}{\equiv}$$

$$p \Rightarrow \neg((q \vee p) \wedge (r \vee q)) \overset{4}{\equiv}$$

$$p \Rightarrow \neg((q \vee p) \wedge (q \vee r)) \overset{8}{\equiv} p \Rightarrow \neg(q \vee (p \wedge r)) \overset{18}{\equiv}$$

$$\neg p \vee \neg(q \vee (p \wedge r)) \overset{9}{\equiv} \neg(p \wedge (q \vee (p \wedge r))) \overset{7}{\equiv}$$

$$\neg((p \wedge q) \vee (p \wedge (p \wedge r))) \overset{5}{\equiv}$$

$$\neg((p \wedge q) \vee ((p \wedge p) \wedge r))) \overset{1}{\equiv}$$

$$\neg((p \wedge q) \vee (p \wedge r)) \overset{7}{\equiv}$$

$$\neg(p \wedge (q \vee r)).$$

Примеры

- ***Пример 3.*** С помощью таблиц истинности доказать равносильность формул $\overline{(\bar{p} \wedge q)} \wedge (r \to p)$ и $p \vee (\overline{q \vee r})$.

Сначала составим таблицу истинности для первой формулы, а затем для второй. После этого сравним, полученный результат.

Проставим порядок действий

$$\overset{3}{\overline{(\overset{1}{\bar{p}} \overset{2}{\wedge} q)}} \overset{5}{\wedge} (r \overset{4}{\rightarrow} p)$$

p	q	r	1	2	3	4	5
И	И	И	Л	Л	И	И	И
И	И	Л	Л	Л	И	И	И
И	Л	И	Л	Л	И	И	И
Л	И	И	И	И	Л	Л	Л
И	Л	Л	Л	Л	И	И	И
Л	И	Л	И	И	Л	И	Л
Л	Л	И	И	Л	И	Л	Л
Л	Л	Л	И	Л	И	И	И

Проставим порядок действий

$$p \overset{3}{\vee} \left(\overset{2}{\overline{q \vee r}}\right)$$

p	q	r	1	2	3
И	И	И	И	Л	И
И	И	Л	И	Л	И
И	Л	И	И	Л	И
Л	И	И	И	Л	Л
И	Л	Л	Л	И	И
Л	И	Л	И	Л	Л
Л	Л	И	И	Л	Л
Л	Л	Л	Л	И	И

$\overline{(\bar{p} \wedge q)} \wedge (r \to p)$	$p \vee (\overline{q \vee r})$
5	3
И	И
И	И
И	И
Л	Л
И	И
Л	Л
Л	Л
И	И

По результатам таблиц истинности делаем вывод, что формулы равносильны.

Логические уравнения

$$\overline{A}\cdot B + A\cdot\overline{B}\cdot C = 1$$

A=1, B=0, C=1

$$\overline{A}\cdot B = 1$$ **или** $$A\cdot\overline{B}\cdot C = 1$$

A=0, B=1, C – **любое**
2 решения: (0, 1, 0), (0, 1, 1)

! Всего 3 решения!

$$K\cdot L + M\cdot L\cdot N + K\cdot L\cdot\overline{M} = 1$$

K=1, L=1,
M и N – **любые**
4 решения

M=1, L=1, N=1,
K – **любое**
2 решения

K=1, L=1, M=0,
N – **любое**
2 решения

$$L\cdot(K + M\cdot N) = 1$$

! Всего 5 решений!

Синтез логических выражений

Синтез логических выражений

A	B	X	
0	0	1 •	$\overline{A}\cdot\overline{B}$
0	1	1 •	$\overline{A}\cdot B$
1	0	0	
1	1	1 •	$A\cdot B$

Шаг 1. Отметить строки в таблице, где **X = 1**.

Шаг 2. Для каждой из них записать логическое выражение, которое истинно только для этой строки.

Шаг 3. Сложить эти выражения и упростить результат.

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

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

распределительный

исключения третьего

распределительный

исключения третьего

Синтез логических выражений (2 способ)

A	B	X	
0	0	1	
0	1	1	
1	0	0	$A \cdot \overline{B}$
1	1	1	

Шаг 1. Отметить строки в таблице, где **X = 0**.

Шаг 2. Для каждой из них записать логическое выражение, которое истинно только для этой строки.

Шаг 3. Сложить эти выражения и упростить результат, который равен $\overline{X}$.

Шаг 4. Сделать инверсию.

$$\overline{X} = A \cdot \overline{B} \;\Rightarrow\; X = \overline{A \cdot \overline{B}} = \overline{A} + B$$

Когда удобнее применять 2-ой способ?

Синтез логических выражений (3 способ)

A	B	X	
0	0	0 •	$A+B$
0	1	1	
1	0	0 •	$\overline{A}+B$
1	1	1	

Шаг 1. Отметить строки в таблице, где **X = 0**.

Шаг 2. Для каждой из них записать логическое выражение, которое **ложно** только для этой строки.

Шаг 3. **Перемножить** эти выражения и упростить результат.

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

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

Синтез логических выражений

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

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

Синтез логических выражений (2 способ)

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

$A \cdot \overline{B} \cdot \overline{C}$

$A \cdot B \cdot \overline{C}$

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

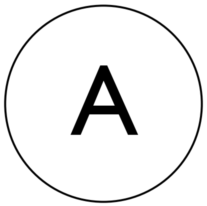
$$= A \cdot \overline{C} \cdot (\overline{B} + B)$$

$$= A \cdot \overline{C}$$

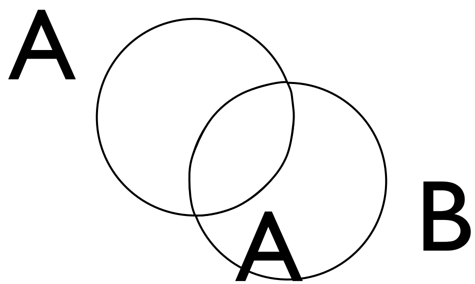
$$X = \overline{A \cdot \overline{C}} = \boxed{\overline{A} + C}$$

Диаграммы Венна

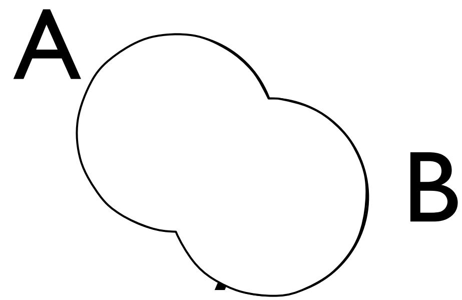
Диаграммы Венна (круги Эйлера)



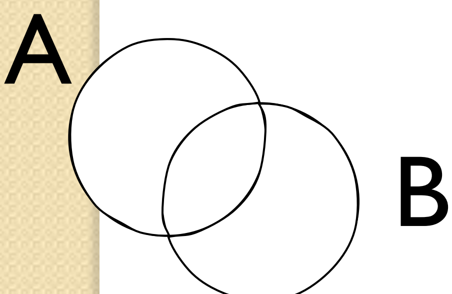
$\overline{A}$



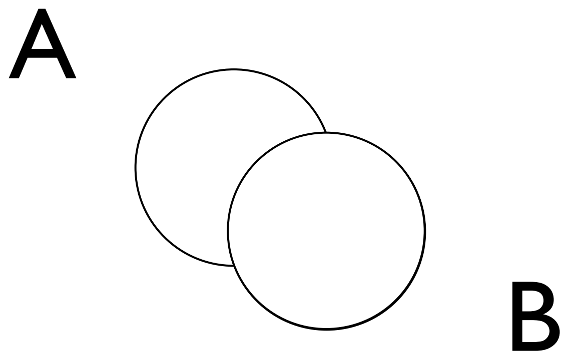
$A \cdot B$



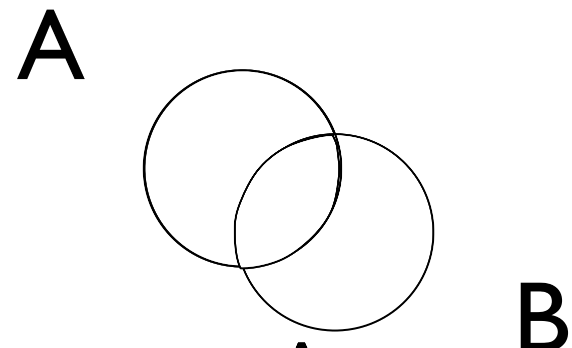
$A + B$



$A \oplus B$

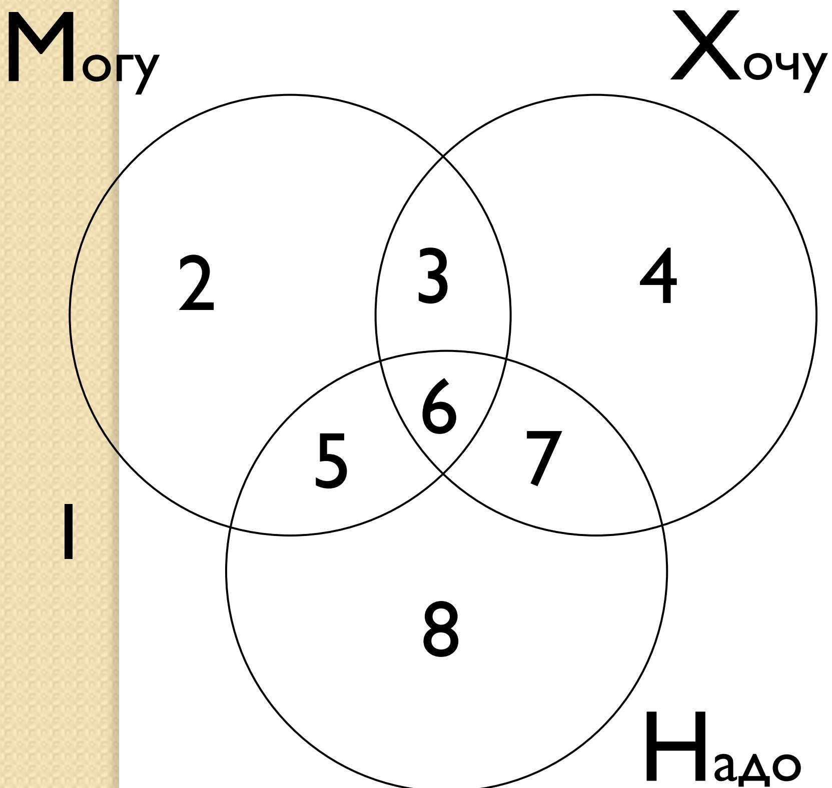


$A \to B$



$A \leftrightarrow B$

Диаграмма с тремя переменными



$$1 = \overline{М} \cdot \overline{Х} \cdot \overline{Н} \qquad 5 = М \cdot \overline{Х} \cdot Н$$

$$2 = М \cdot \overline{Х} \cdot \overline{Н} \qquad 6 = М \cdot Х \cdot Н$$

$$3 = М \cdot Х \cdot \overline{Н} \qquad 7 = \overline{М} \cdot Х \cdot Н$$

$$4 = \overline{М} \cdot Х \cdot \overline{Н} \qquad 8 = \overline{М} \cdot \overline{Х} \cdot Н$$

$$3 + 4 = М \cdot Х \cdot \overline{Н} + \overline{М} \cdot Х \cdot \overline{Н} \qquad 3 + 4 = Х \cdot \overline{Н}$$

! Логические выражения можно упрощать!

Задачи

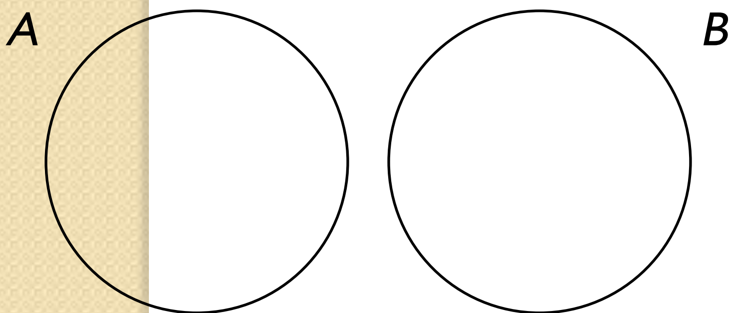
Известно количество сайтов, которых находит поисковый сервер по следующим запросам :

Запрос	***Количество сайтов***
огурцы	*100*
помидоры	*200*
огурцы & помидоры	*50*

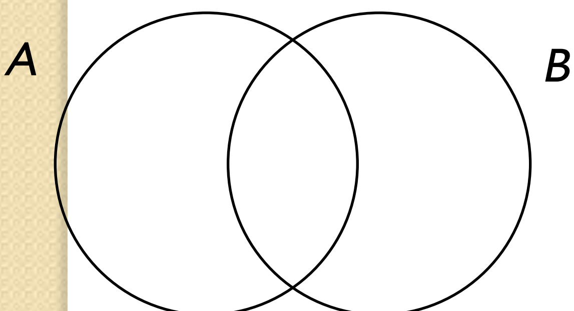
Сколько сайтов будет найдено по запросу

огурцы | помидоры

Задачи



$N_{A|B} =$



50

огурцы & помидоры

$$N_{A|B} = N_A + N_B - N_{A\&B}$$

огурцы | помидоры

250

огурцы

100

помидоры

200

Задачи

Известно количество сайтов, которых находит поисковый сервер по следующим запросам :

Запрос	**Количество сайтов**
Динамо & Рубин	*320*
Спартак & Рубин	*280*
(Динамо | Спартак) & Рубин	*430*

Сколько сайтов будет найдено по запросу

`Динамо & Спартак & Рубин`

! Общее условие с & можно отбросить !

Задачи

Известно количество сайтов, которых находит поисковый сервер по следующим запросам :

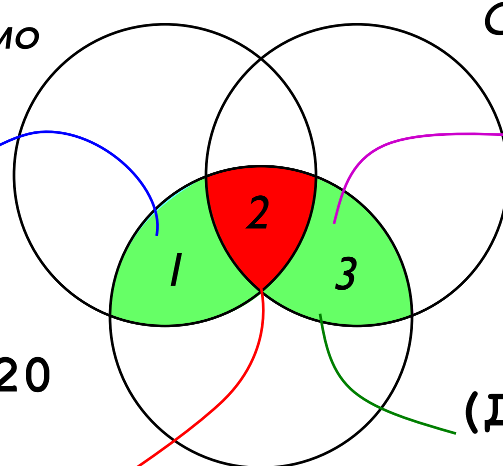
Запрос	***Количество сайтов***
Динамо	*320*
Спартак	*280*
Динамо | Спартак	*430*

Сколько сайтов будет найдено по запросу

`Динамо & Спартак`

Ответ: 320 + 280 – 430 = **170**

Задачи




```
Спартак & Рубин
    = 2 + 3 = 280

Динамо & Рубин
    = 1 + 2 = 320

(Динамо | Спартак) & Рубин
    = 1 + 2 + 3 = 430

Динамо & Спартак & Рубин
    = 2
    = (320 + 280) – 430 = 170
```

Задачи

Некоторый сегмент сети Интернет состоит из 1000 сайтов. Поисковый сервер в автоматическом режиме составил таблицу ключевых слов для сайтов этого сегмента. Вот ее фрагмент:

Ключевое слово	**Количество сайтов, для которых данное слово является ключевым**
сканер	*200*
принтер	*250*
монитор	*450*

Сколько сайтов будет найдено по запросу

(принтер | сканер) & монитор

если по трем следующим запросам найдено:

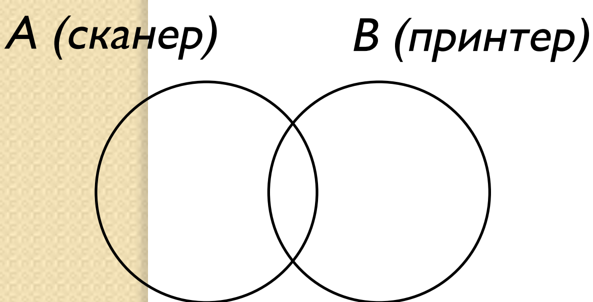
принтер | сканер – *450 сайтов,*

принтер & монитор – *40 сайтов*

сканер & монитор – *50 сайтов.*

Задачи

(принтер | сканер) & монитор = ?

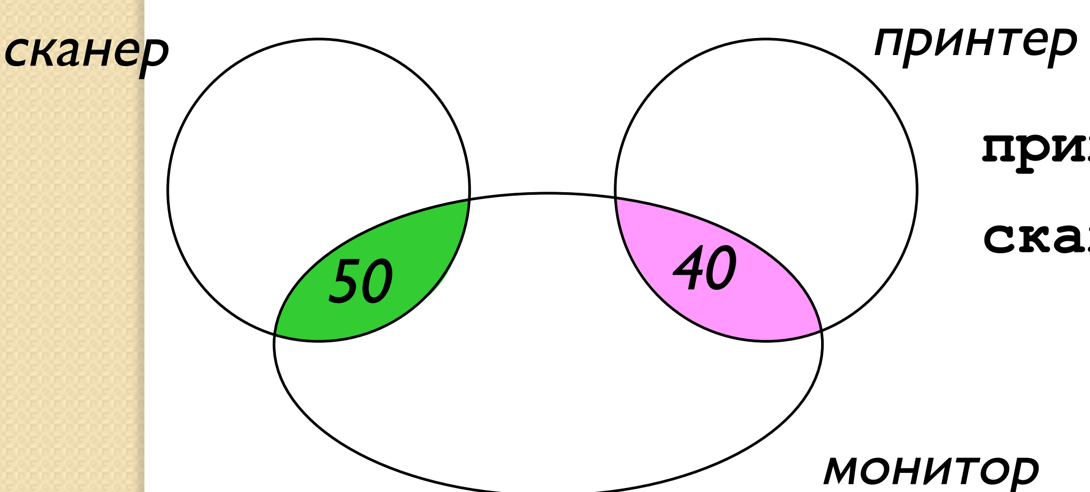


450 **принтер | сканер**

$$N_{A|B} = N_A + N_B - N_{A\&B}$$

0

сканер 200 **принтер** 250



принтер & монитор = 40

сканер & монитор = 50

$40 + 50 =$ **90**

Предикаты и кванторы

Предикаты

Предикат (логическая функция) – это утверждение, содержащее переменные.

Предикат-свойство – от одной переменной:

P(*N*) = «В городе *N* живут более 2 млн человек»

P(Москва) = 1

P(Якутск) = 0

Простое(x) = «x – простое число»

Спит(x) = «x всегда спит на уроке»

Предикат-отношение – от нескольких переменных:

Больше(x, y) = «$x > y$»

Живет(x, y) = «x живет в городе y»

Любит(x, y) = «x любит y»

Предикаты и кванторы

Предикаты задают **множества:**

$$P(x) = (x > 0)$$

$$P(x, y) = (x + y = 1)$$

Предикаты, которые **всегда истинны**:

$P(x) = (x^2 \geq 0)$ для всех вещественных чисел

«Для любого допустимого x утверждение P(x) истинно»:

квантор

$$\forall x \, P(x)$$

высказывание

Квантор – знак, обозначающий количество.

$\forall$ = A (*all* – все) $\exists$ = E (*exists* – существует)

Кванторы

Какой квантор использовать?

« … $\forall$ моря соленые».

« … $\exists$ кошки серые».

« … $\exists$ числа чётные».

« … $\forall$ окуни – рыбы».

« … $\exists$ прямоугольники – квадраты».

« … $\forall$ квадраты – прямоугольники».

Истинно ли высказывание?

$\cancel{\forall x\, \mathrm{P}(x) \text{ при } \mathrm{P}(x) = (x > 0)}$

✓ $\exists x\, \mathrm{P}(x) \text{ при } \mathrm{P}(x) = (x > 0)$

✓ $\forall x\, \mathrm{P}(x) \text{ при } \mathrm{P}(x) = (x^2 \geq 0)$

✓ $\exists x\, \mathrm{P}(x) \text{ при } \mathrm{P}(x) = (x^2 \geq 0)$

Кванторы

Дано:
A = «Все люди смертны» = 1.
B = «Сократ – человек» = 1.
Доказать:
C = «Сократ смертен» = 1.
Доказательство:

A	B	$A \to B$
0	0	1
0	1	1
1	0	0
1	1	1

P(x) = «x – человек» Q(x) = «x – смертен»

A = 1: $\forall x\, (P(x) \to Q(x))$

при «x =Сократ» $P(\text{Сократ}) \to Q(\text{Сократ}) = 1$

B = 1: $P(\text{Сократ}) = 1$

по свойствам импликации $Q(\text{Сократ}) = 1$

Несколько кванторов

Квантор **связывает** одну переменную:

$\forall x\, \mathrm{P}(x, y)$ – предикат от переменной y

$\exists y\, \mathrm{P}(x, y)$ – предикат от переменной x

Два квантора связывают две переменных:

$\forall x\, \exists y\, \mathrm{P}(x, y)$ – высказывание «для любого x существует y, при котором $\mathrm{P}(x,y)=1$»

$\exists x\, \forall y\, \mathrm{P}(x, y)$ – высказывание «существует x, такой что при любом y верно $\mathrm{P}(x,y)=1$»

Сравните два последних высказывания при:

$\mathrm{P}(x, y) = (x + y = 0)$ $\quad\quad$ $\mathrm{P}(x, y) = (x \cdot y = 0)$

Отрицание

НЕ «для любого x выполняется Р(x)» ⇔ «существует x, при котором не выполняется Р(x)»

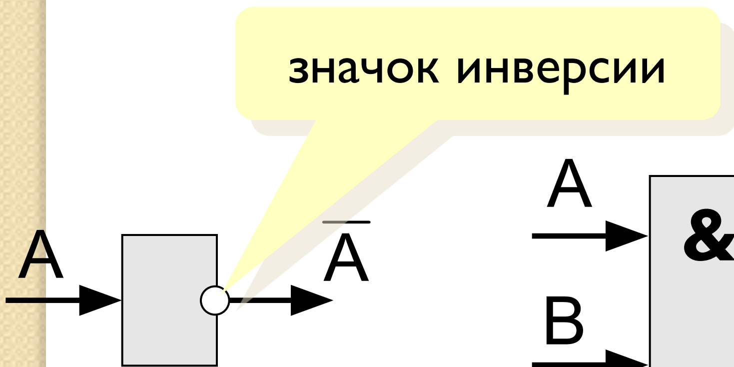
$$\overline{\forall x\, \mathrm{P}(x)} = \exists x\, \overline{\mathrm{P}(x)}$$

НЕ «существует x, при котором выполняется Р(x)» ⇔ «для любого x не выполняется Р(x)»

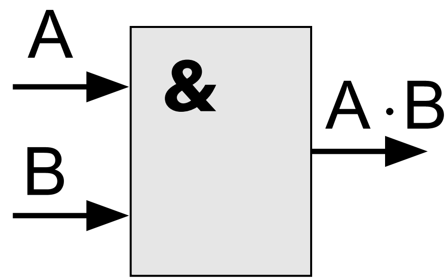
$$\overline{\exists x\, \mathrm{P}(x)} = \forall x\, \overline{\mathrm{P}(x)}$$

Логические элементы компьютера

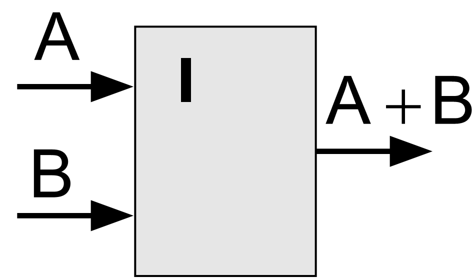
Логические элементы компьютера



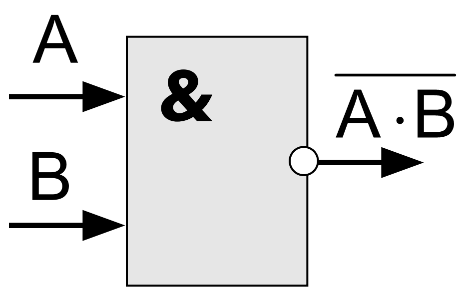
НЕ



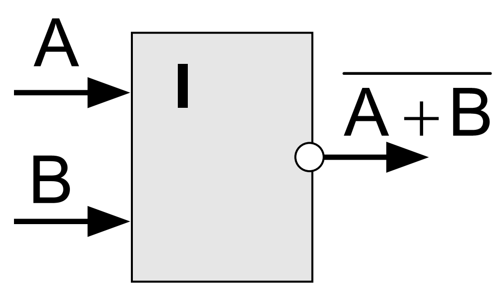
И



ИЛИ



И-НЕ

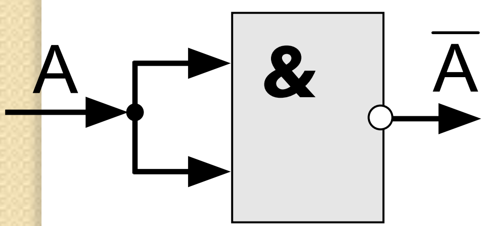


ИЛИ-НЕ

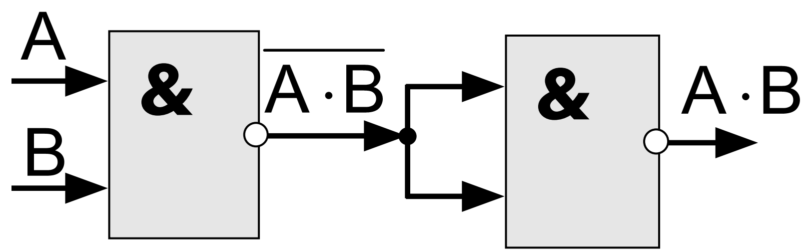
Логические элементы компьютера

Любое логическое выражение можно реализовать на элементах **И-НЕ** или **ИЛИ-НЕ.**

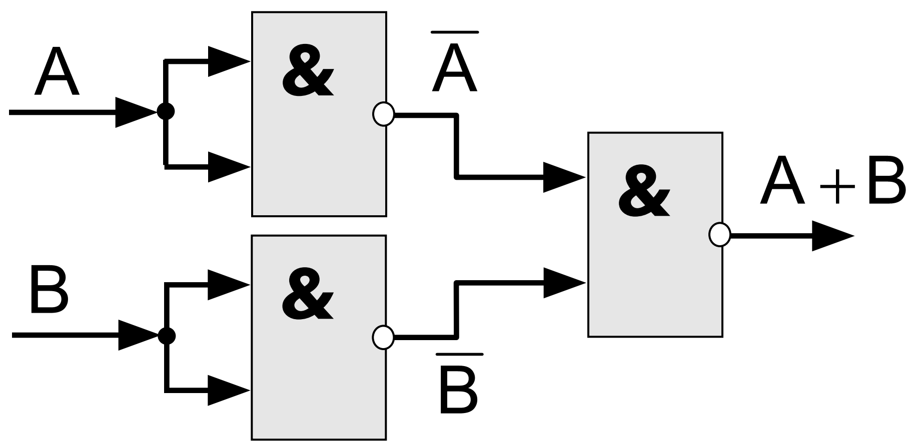
НЕ: $\overline{A} = \overline{A} + \overline{A} = \overline{A \cdot A}$



И: $A \cdot B = \overline{\overline{A \cdot B}}$



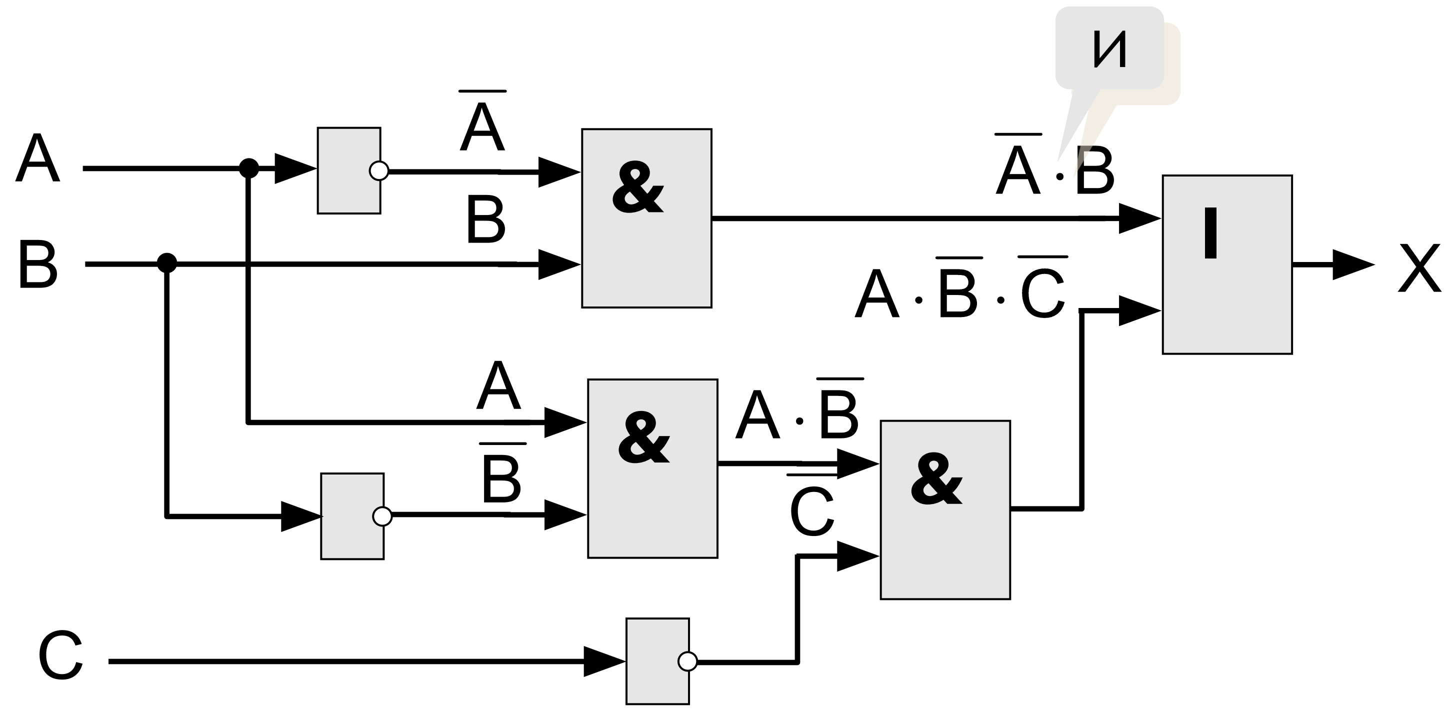
ИЛИ: $A + B = \overline{\overline{A} \cdot \overline{B}}$



Составление схем

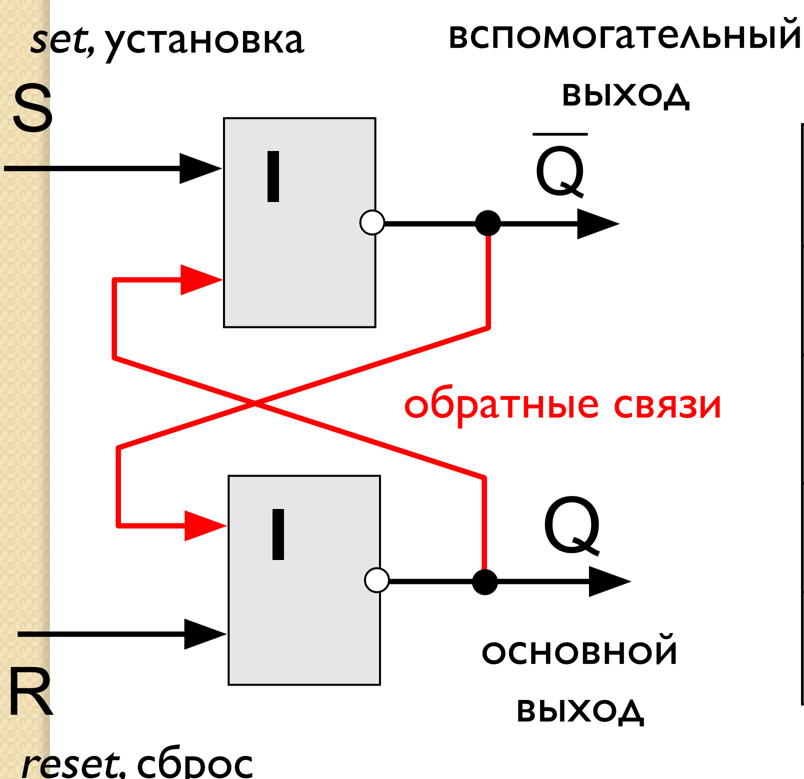
последняя операция - ИЛИ

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



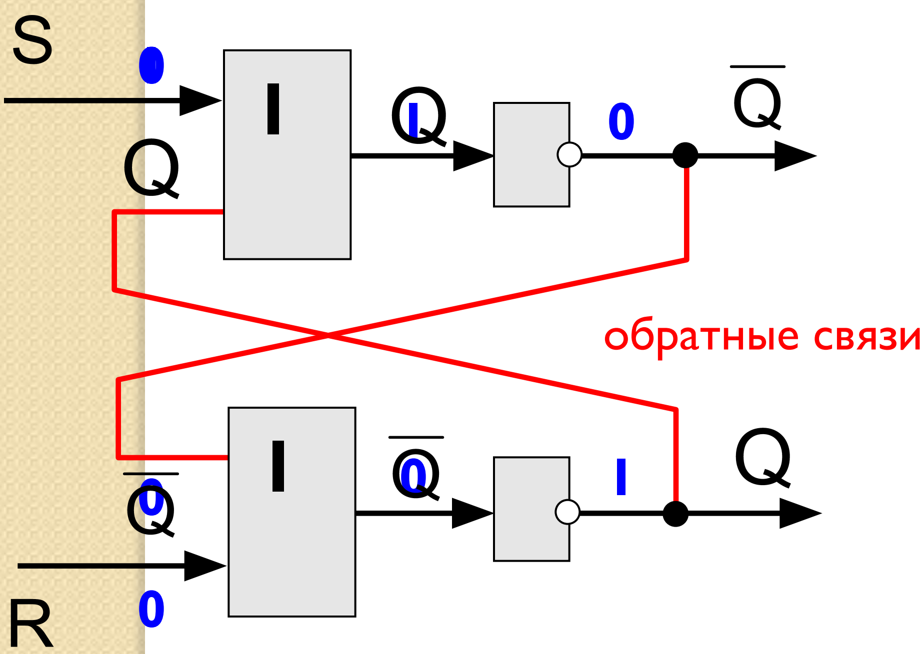
Триггер (англ. *trigger* – защёлка)

Триггер – это логическая схема, способная хранить 1 бит информации (1 или 0). Строится на 2-х элементах **ИЛИ-НЕ** или на 2-х элементах **И-НЕ.**



S	R	Q	$\overline{Q}$	режим
0	0	Q	$\overline{Q}$	хранение
0	1	0	1	сброс
1	0	1	0	установка 1
1	1	0	0	запрещен

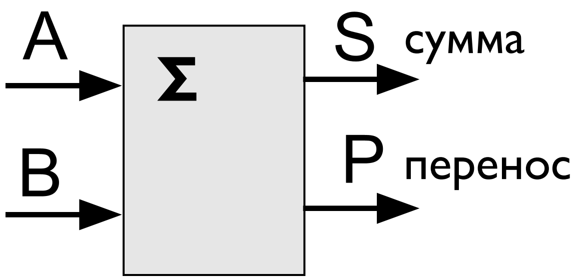
Триггер – таблица истинности



S	R	Q	$\overline{Q}$	режим
0	0	Q	$\overline{Q}$	хранение
0	1	0	1	сброс
1	0	1	0	установка 1
1	1	0	0	запрещен

Полусумматор

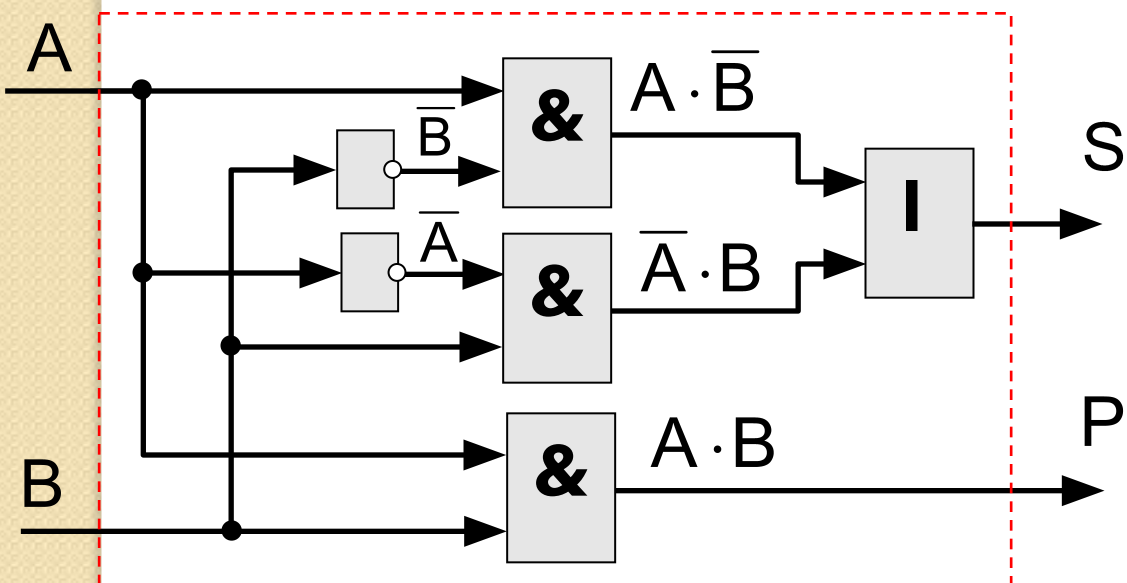
Полусумматор – это логическая схема, способная складывать два одноразрядных двоичных числа.



A	B	P	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$P = A \cdot B$

$S = A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B$

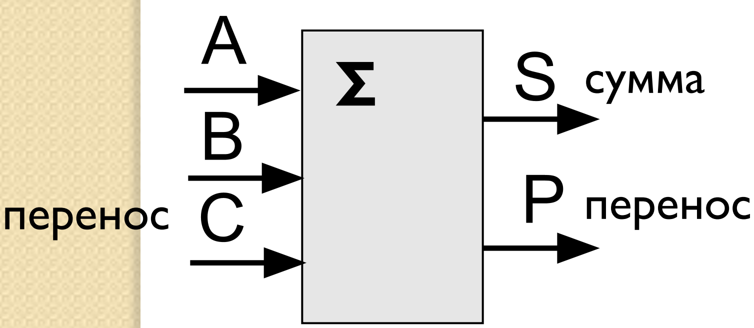


$S = A \cdot \overline{B} + \overline{A} \cdot B$

Схема на 4-х элементах?

Сумматор

Сумматор – это логическая схема, способная складывать два одноразрядных двоичных числа с переносом из предыдущего разряда.



A	B	C	P	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Многоразрядный сумматор

это логическая схема, способная складывать два n-разрядных двоичных числа.

$$
\begin{array}{rl}
 & A = \quad a_n \ a_{n-1} \ \ldots \ a_1 \\
+ & B = \quad b_n \ b_{n-1} \ \ldots \ b_1 \\
\hline
 & C = \boxed{p} \ c_n \ c_{n-1} \ \ldots \ c_1
\end{array}
$$

перенос