

# Логические основы компьютера

---

## Базовые логические элементы

# Базовые логические элементы

---

Компьютер выполняет арифметические и логические операции при помощи т.н. **базовых логических элементов**, которые также еще называют вентилями.

- Вентиль «И» – конъюнктор.  
Реализует конъюнкцию.
  - Вентиль «ИЛИ» – дизъюнктор.  
Реализует дизъюнкцию.
  - Вентиль «НЕ» – инвертор.  
Реализует инверсию
-

# Составные элементы

---

- Любая логическая операция может быть представлена через конъюнкцию, дизъюнкцию и инверсию
  - Любой сколь угодно сложный элемент компьютера может быть сконструирован из элементарных вентиляей
-

# Сигналы-аргументы и сигналы-функции

---

Вентили оперируют с электрическими импульсами:

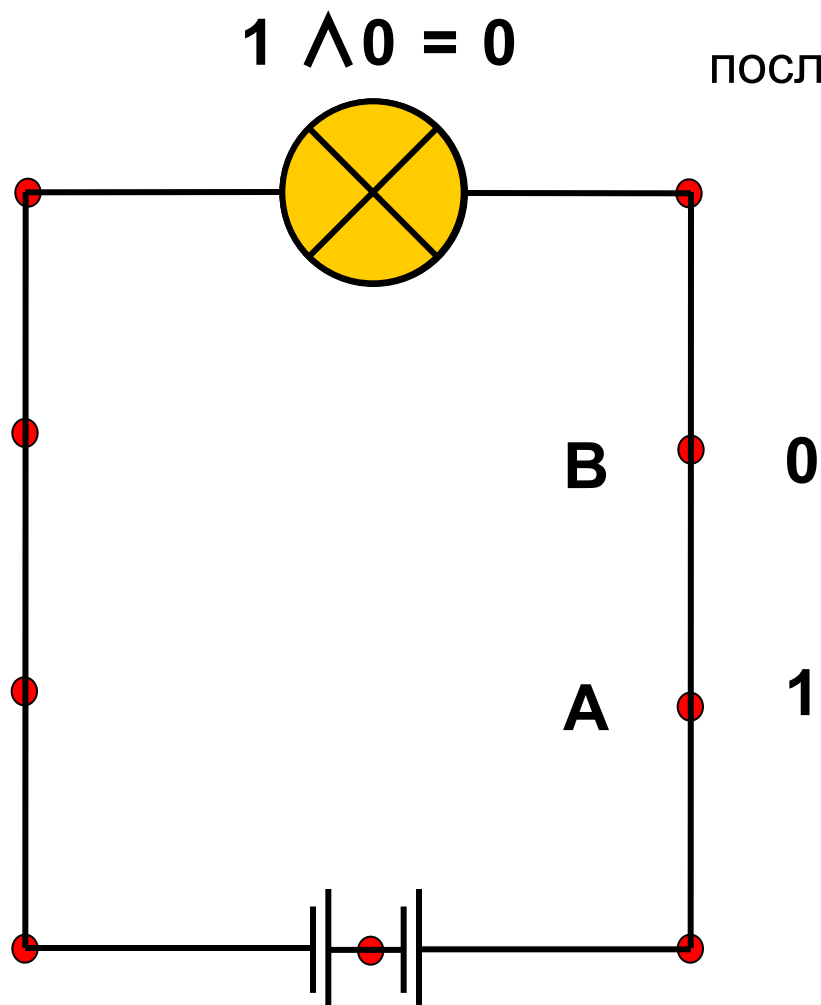
- Импульс имеется –  
логический смысл сигнала «1»
- Импульса нет –  
логический смысл сигнала «0»

На входы вентиля подаются импульсы – значения аргументов, на выходе вентиля появляется сигнал – значение функции

---

# Логическая схема типа «И» (конъюнктор)

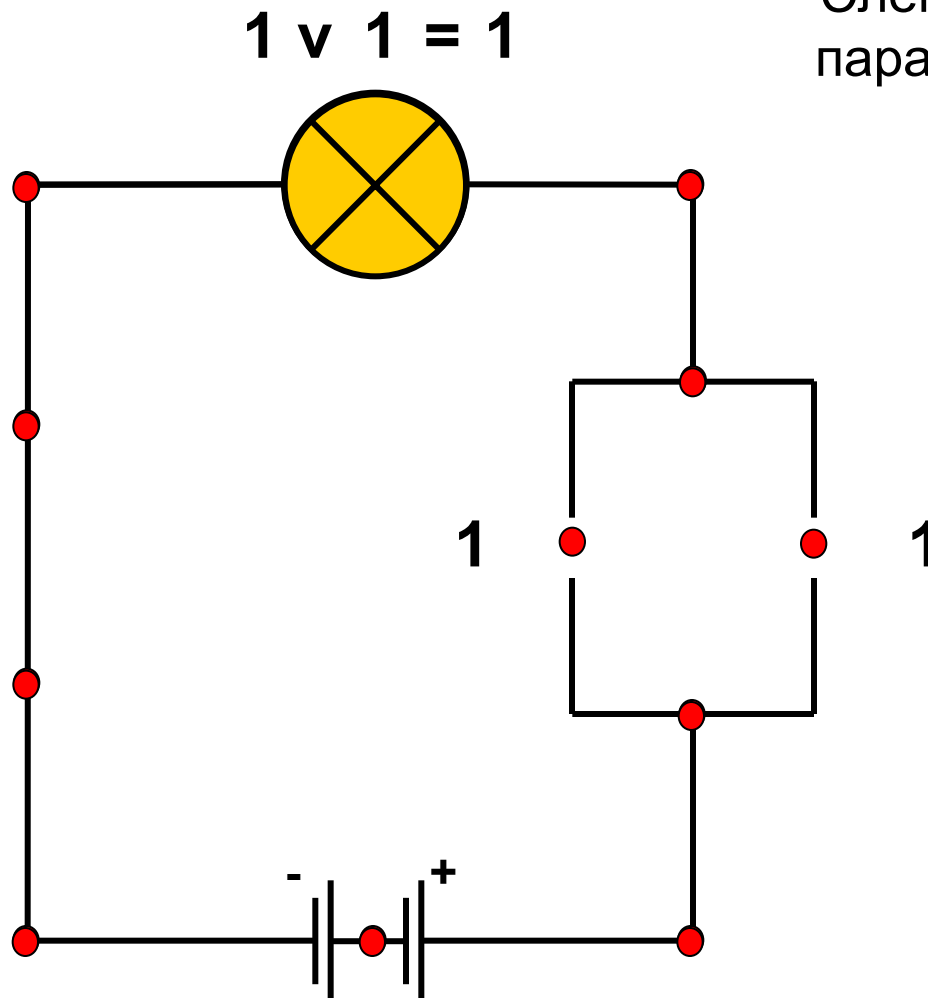
Электрическая цепь из двух  
последовательно подключенных  
выключателей



A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

# Логическая схема типа «ИЛИ» (дизъюнктор)

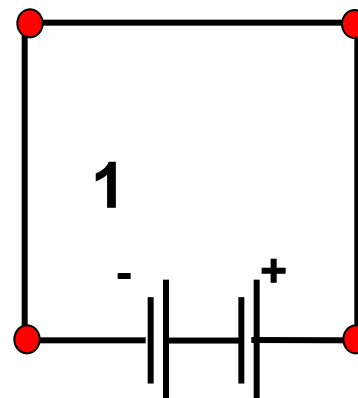
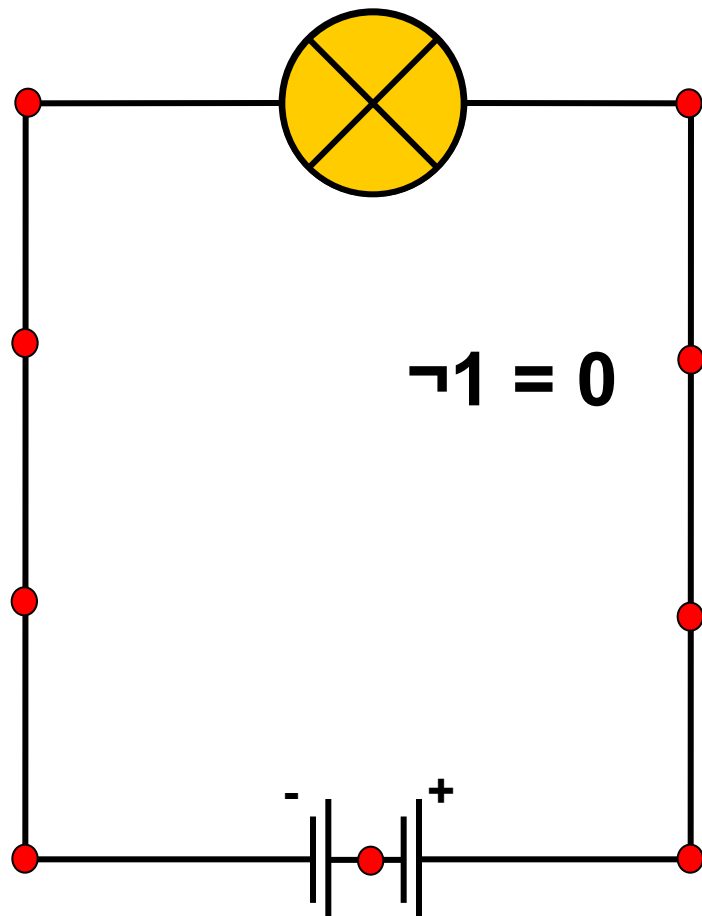
Электрическая цепь из двух  
параллельно подключенных  
выключателей



A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

# Логическая схема типа «НЕ» (инвертор)

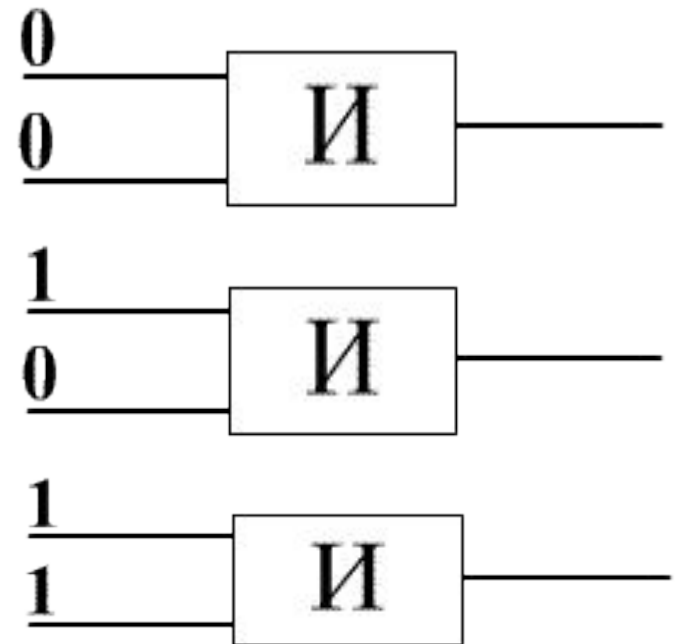
Электрическая цепь с одним  
автоматическим выключателем



A	$\neg A$
0	1
1	0

# Конъюнктор

- На входы конъюнктора подаются сигналы **0** или **1**
- На выходе конъюнктора появляются сигналы **0** или **1** в соответствии с таблицей истинности





# Дизъюнктор

- На входы дизъюнктора подаются сигналы **0** или **1**
- На выходе дизъюнктора появляются сигналы **0** или **1** в соответствии с таблицей истинности



# Инвертор

- На входы инвертора подаются сигналы **0** или **1**
- На выходе инвертора появляются сигналы **1** или **0** в соответствии с таблицей истинности



# Сумматор двоичных чисел

---

- Любое математическое сколь угодно сложное выражение может быть представлено в виде последовательности элементарных математических операций
  - Все математические действия в компьютере сводятся к сложению двоичных чисел
  - Основу микропроцессора составляют *сумматоры двоичных чисел*
-

# Полусумматор. Арифметическое сложение двоичных чисел

В каждом разряде образуется сумма цифр в соответствующих разрядах слагаемых, при этом возможен перенос единицы в старший разряд

**Без переноса**

$$\begin{array}{r} + 0000 \ 0001 \\ 0000 \ 0010 \\ \hline 0000 \ 0011 \end{array}$$

**С переносом**

$$\begin{array}{r} + 0000 \ 0011 \\ 0000 \ 0010 \\ \hline 0000 \ 0101 \end{array}$$

# Обозначим слагаемые через $A$ и $B$ , перенос – через $P$ , а сумму – через $S$

Таблица сложения одноразрядных двоичных чисел:

Слагаемые		Перенос	Сумма
$A$	$B$	$P$	$S$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Очевидно, что  $P = A \wedge B$

# Получаем формулу для вычисления S

Если сравнить  $A \vee B$  с S:

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

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

то очевидно, что они практически идентичны.

Чтобы равенство оказалось полным нужно  
выражение  $A \vee B$  умножить на  $\neg P$

# Получаем формулу для вычисления S

$$S = (A \vee B) \wedge \neg P \Rightarrow (A \vee B) \wedge \neg(A \wedge B)$$

A	B	$A \vee B$	$A \wedge B$	$\neg(A \wedge B)$	$(A \vee B) \wedge \neg(A \wedge B)$
0	0	0	0	1	0
0	1	1	0	1	1
1	0	1	0	1	1
1	1	1	1	0	0

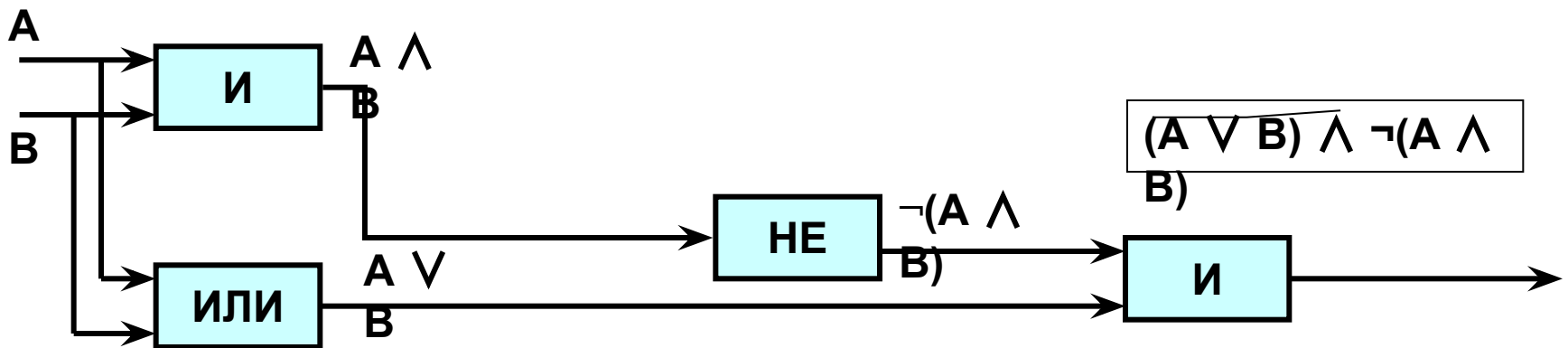
Теперь, имея элементарные логические выражения, можно построить логическую схему устройства для сложения одноразрядных двоичных чисел (полусумматора)

# Логическая схема

## ДВОИЧНОГО ПОЛУСУММАТОРА

Полусумматор называется так, потому, что здесь не учитывается перенос единицы из младшего разряда

$$(A \vee B) \wedge \neg(A \wedge B)$$





# Полный одноразрядный сумматор

Должен иметь три входа (A, B и  $P_0$ ) и два выхода (S и P)

Слагаемые		Переносы		Сумма
A	B	$P_0$	P	S
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1

# Формула полного одноразрядного сумматора

- **P** принимает значение **1** когда хотя бы две из трех переменных равны **1**:

$$P = (A \wedge B) \vee (A \wedge P_0) \vee (B \wedge P_0)$$

- Сумма равна произведению логического сложения (**A**, **B** и **P<sub>0</sub>**) на инвертированный перенос  $\neg P$ :

$$S = (A \vee B \vee P_0) \wedge \neg P$$

- Это выражение справедливо во всех случаях, кроме одного, когда **A**, **B** и **P<sub>0</sub>** равны **1**:

# Формула полного одноразрядного сумматора

Правильное значение суммы – 1. Для ее получения необходимо полученное выражение сложить с произведением этих же переменных:

$$S = (A \vee B \vee P_0) \wedge \neg P \vee (A \wedge B \wedge P_0)$$

# Многоразрядный сумматор

---

Построен на основе полных одноразрядных сумматоров (по одному на каждый разряд), причем таким образом, чтобы выход (перенос) младшего сумматора был подключен ко входу старшего сумматора

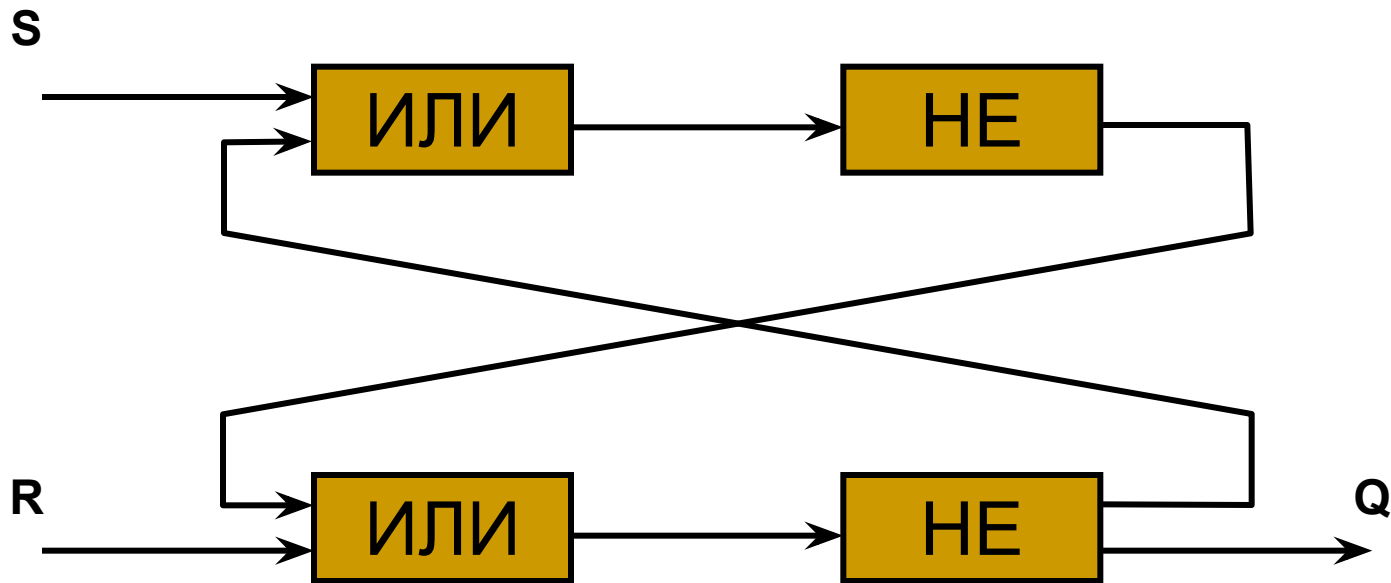
---

# Триггер

---

- Важнейшая структурная единица оперативной памяти и регистров процессора
  - Состоит из двух логических элементов «ИЛИ» и двух логических элементов «НЕ»
-

# Логическая схема триггера



# Работа триггера

---

- В обычном состоянии на входы триггера **S** и **R** подан сигнал «**0**» и триггер хранит «**0**».
  - При подаче сигнала «**1**» на вход **S** триггер принимает значение на выходе **Q** значение «**1**»
  - При подаче сигнала «**1**» на вход **R** триггер возвращается в свое исходное состояние – хранит «**0**»
-