



# ОСНОВЫ ЛОГИКИ И ЛОГИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРА

# ФОРМЫ МЫШЛЕНИЯ

- **ЛОГИКА** — это наука о формах и законах человеческого мышления и, в частности, о законах доказательных рассуждений.
- Логика изучает мышление как средство познания объективного мира. Законы логики отражают в сознании человека свойства, связи и отношения объектов окружающего мира.
- Формальная логика связана с анализом наших обычных содержательных умозаключений, выражаемых разговорным языком. Математическая логика изучает только умозаключения со строго определенными объектами и суждениями, для которых можно однозначно решить, истинны они или ложны.
- Идеи и аппарат логики используется в кибернетике, вычислительной технике и электротехнике (построение компьютеров основано на законах математической логики).
- В основе логических схем и устройств ПК лежит специальный математический аппарат, использующий законы логики. Математическая логика изучает вопросы применения математических методов для решения логических задач и построения логических схем. Знание логики необходимо при разработке алгоритмов и программ, так как в большинстве языков программирования есть логические операции.

# Основные формы мышления

Основными формами мышления являются: *ПОНЯТИЯ, СУЖДЕНИЯ, УМОЗАКЛЮЧЕНИЯ.*

***ПОНЯТИЕ*** - форма мышления, в которой отражаются существенные признаки отдельного объекта или класса однородных объектов.

Примеры: *портфель, трапеция, ураганный ветер.*

Понятие имеет две стороны: *содержание* и *объем.*

Содержание понятия составляет совокупность существенных признаков объекта. Чтобы раскрыть содержание понятия, следует найти признаки, необходимые и достаточные для выделения данного объекта из множества других объектов. Например, содержание понятия «персональный компьютер» можно раскрыть следующим образом: «Персональный компьютер — это универсальное электронное устройство для автоматической обработки информации, предназначенное для одного пользователя».

Объем понятия определяется совокупностью предметов, на которую оно распространяется. Объем понятия «персональный компьютер» выражает всю совокупность (сотни миллионов) существующих в настоящее время в мире персональных компьютеров.

# Основные формы мышления

**СУЖДЕНИЕ** – это форма мышления, в которой что-либо утверждается или отрицается об объектах, их свойствах и отношениях.

Суждениями обычно являются повествовательными предложениями, которые могут быть или истинными или ложными.

*«Берн — столица Франции»,  
«Река Кубань впадает в Азовское море»,  
« $2 > 9$ »,  
« $3 \times 5 = 10$ »*

**УМОЗАКЛЮЧЕНИЕ** – это форма мышления, посредством которой из одного или нескольких истинных суждений, называемых посылками, мы по определенным правилам вывода получаем новое суждение (заключение).

*Все металлы - простые вещества. Литий - металл. → Литий - простое вещество.*

*Один из углов треугольника равен  $90^\circ$ . → Этот треугольник прямоугольный.*

# АЛГЕБРА ВЫСКАЗЫВАНИЙ

- В основе работы логических схем и устройств персонального компьютера лежит специальный математический аппарат - математическая логика. Математическая логика изучает вопросы применения математических методов для решения логических задач и построения логических схем. Знание логики необходимо при разработке алгоритмов и программ, так как в большинстве языков программирования есть логические операции.
- Английский математик **Джордж Буль (1815 — 1864 г.)** создал логическую алгебру, в которой высказывания обозначены буквами. Сочинение Джорджа Буля, в котором подробно исследовалась эта алгебра, было опубликовано в 1854 г. Оно называлось «Исследование законов мысли» («Investigation of the Laws of Thought»). Отсюда ясно, что Буль рассматривал свою алгебру как инструмент изучения законов человеческого мышления, то есть законов логики. Алгебру логики иначе называют алгеброй высказываний. В математической логике суждения называются высказываниями.

**ВЫСКАЗЫВАНИЕ** - это повествовательное предложение, о котором можно сказать, что оно истинно или ложно.

Например:

*Земля - планета Солнечной системы.* (Истинно)

$2+8<5$  (Ложно)

$5 \cdot 5=25$  (Истинно)

*Всякий квадрат есть параллелограмм* (Истинно)

*Каждый параллелограмм есть квадрат* (Ложно)

$2 \cdot 2 =5$  (Ложно)

Не всякое предложение является высказыванием:

- 1) Восклицательные и вопросительные предложения высказываниями не являются.
  - “Какого цвета этот дом?”
  - “Пейте томатный сок!”
  - “Стоп!”
  
- 2) Не являются высказываниями и определения.

“Назовем медианой отрезок, соединяющий вершину треугольника с серединой противоположной стороны”.

Определения не бывают истинными или ложными, они лишь фиксируют принятое использование терминов.

3) Не являются высказываниями и предложения типа “Он сероглаз” или “ $x - 4x + 3 = 0$ ” - в них не указано о каком человеке идет речь или для какого числа  $x$  верно равенство. Такие предложения называются **высказывательными формами**.

**Высказывательная форма** — это повествовательное предложение, которое прямо или косвенно содержит хотя бы одну переменную и становится высказыванием, когда все переменные замещаются своими значениями.

Высказывания могут быть **простыми и сложными**.

**Высказывание считается простым, если никакую его часть нельзя рассматривать как отдельное высказывание**

Некоторые высказывания можно разложить на отдельные части, при этом каждая такая часть будет самостоятельным высказыванием. Например, высказывание “Сегодня в 4 часа дня я был в школе, а к 6 часам вечера пошел на каток” состоит из 2 частей. Высказывание может состоять и из большего количества частей.

**Высказывание, которое можно разложить на части, будем называть сложным, а неразложимое далее высказывание - простым.**

Сложное высказывание получается путем объединения простых высказываний **логическими связками** — **НЕ, И, ИЛИ**.  
Значение истинности сложных высказываний зависит от истинности входящих в них простых высказываний и объединяющих их связок.

Например, даны простые высказывания:

*На улице идет дождь.*

*На улице светит солнце.*

*На улице пасмурная погода.*

Составим из них сложные высказывания:

*На улице идет дождь **И** на улице светит солнце.*

*На улице светит солнце **ИЛИ** на улице пасмурная погода.*

***Неверно что** на улице идет дождь.*

- В математической логике не рассматривается конкретное содержание высказывания, важно только, истинно оно или ложно. Поэтому **высказывание можно представить некоторой переменной величиной, значением которой может быть только 0 или 1. Если высказывание истинно, то его значение равно 1, если ложно - 0.**
- Простые высказывания называли **логическими переменными** и для простоты записи их обозначают латинскими буквами: *A, B, C...*
  - Луна является спутником Земли.  $A = 1$*
  - Москва – столица Германии.  $B = 0$*
- Сложные высказывания называются **логическими функциями**. Значения логической функции также может принимать значения только 0 или 1.

# БАЗОВЫЕ ЛОГИЧЕСКИЕ ОПЕРАЦИИ

В алгебре высказываний, как и в обычной алгебре, вводится ряд операций. Логические связки И, ИЛИ и НЕ заменяются логическими операциями: ***конъюнкцией, дизъюнкцией и инверсией***. Это основные логические операции, при помощи которых можно записать любую логическую функцию.

# 1. Логическая операция ИНВЕРСИЯ (ОТРИЦАНИЕ)

- соответствует частице НЕ
- обозначается черточкой над именем переменной или знаком  $\neg$  перед переменной
- **Инверсия логической переменной истинна, если сама переменная ложна, и, наоборот, инверсия ложна, если переменная истинна.**

Таблица истинности инверсии имеет вид:

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

## 2. Логическая операция ДИЗЪЮНКЦИЯ (ЛОГИЧЕСКОЕ СЛОЖЕНИЕ)

- соответствует союзу ИЛИ
- обозначается знаком  $\vee$  или  $\oplus$  или  $|$
- **Дизъюнкция двух логических переменных ложна тогда и только тогда, когда оба высказывания ложны.**  
Это определение можно обобщить для любого количества логических переменных, объединенных дизъюнкцией.

$A \vee B \vee C = 0$ , только если  $A=0$ ,  $B=0$ ,  $C=0$ .

Таблица истинности дизъюнкции имеет следующий вид:

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

### 3. Логическая операция КОНЪЮНКЦИЯ (ЛОГИЧЕСКОЕ УМНОЖЕНИЕ)

- соответствует союзу И
- обозначается знаком  $\&$  или  $\wedge$ , или  $\cdot$ .
- **Конъюнкция двух логических переменных истинна тогда и только тогда, когда оба высказывания истинны.**  
Это определение можно обобщить для любого количества логических переменных, объединенных конъюнкцией.  
 $A \& B \& C = 1$ , только если  $A = 1, B = 1, C = 1$ .  
Таблица истинности конъюнкции имеет следующий вид:

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

# ЛОГИЧЕСКИЕ ВЫРАЖЕНИЯ И ТАБЛИЦЫ ИСТИННОСТИ

- Сложные высказывания можно записывать в виде формул. Для этого простые логические высказывания нужно обозначить как логические переменные буквами и связать их с помощью знаков логических операций. Такие формулы называются **логическими выражениями**. Например:

$$\frac{(A \vee B) \& (\bar{A} \vee \bar{B})}{(A \vee B \& C)}$$

- Чтобы определить значение логического выражения необходимо подставить значения логических переменных в выражение и выполнить логические операции. Операции в логическом выражении выполняются слева направо с учетом скобок в следующем порядке:
  1. инверсия;
  2. конъюнкция;
  3. дизъюнкция;
  4. импликация и эквивалентность.Для изменения указанного порядка выполнения логических операций используются круглые скобки.

# Таблицы истинности

- Для каждого составного высказывания (логического выражения) можно построить *таблицу истинности*, которая определяет истинность или ложность логического выражения при всех возможных комбинациях исходных значений простых высказываний (логических переменных).
- При построении таблиц истинности целесообразно руководствоваться определенной последовательностью действий:
  - 1) записать выражение и определить порядок выполнения операций
  - 2) определить количество строк в таблице истинности. Оно равно количеству возможных комбинаций значений логических переменных, входящих в логическое выражение (определяется по формуле  $Q = 2^n$ , где  $n$  - количество входных переменных)
  - 3) определить количество столбцов в таблице истинности (*= количество логических переменных + количество логических операций*)
  - 4) построить таблицу истинности, обозначить столбцы (имена переменных и обозначения логических операций в порядке их выполнения) и внести в таблицу возможные наборы значений исходных логических переменных.
  - 5) заполнить таблицу истинности, выполняя базовые логические операции в необходимой последовательности и в соответствии с их таблицами истинности

Теперь мы можем определить значение логической функции для любого набора значений логических переменных.

Например, построим таблицу истинности для логической функции:

$$F(A, B, C) = \bar{A} \& (B \vee C)$$

1. Количество входных переменных в заданном выражении равно трем ( $A, B, C$ ). Значит, количество входных наборов, а значит и строк  $Q=2^3=8$ .
2. Количество столбцов равно 6 (3 переменные + 3 операции). Столбцы таблицы истинности соответствуют значениям исходных выражений  $A, B, C$ , промежуточных результатов  $\bar{A}$  и  $(B \vee C)$ , а также искомого окончательного значения сложного арифметического выражения  $\bar{A} \& (B \vee C)$



<b>A</b>	<b>B</b>	<b>C</b>	$\bar{A}$	<b>B V C</b>	$\bar{A} \& (B \vee C)$
<b>0</b>	<b>0</b>	<b>0</b>			
<b>0</b>	<b>0</b>	<b>1</b>			
<b>0</b>	<b>1</b>	<b>0</b>			
<b>0</b>	<b>1</b>	<b>1</b>			
<b>1</b>	<b>0</b>	<b>0</b>			
<b>1</b>	<b>0</b>	<b>1</b>			
<b>1</b>	<b>1</b>	<b>0</b>			
<b>1</b>	<b>1</b>	<b>1</b>			

A	B	C	$\bar{A}$	$B \vee C$	$\bar{A} \& (B \vee C)$
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	0	1	0

Задание. Постройте таблицу истинности для данного логического выражения:

$$(A \vee B) \& (\bar{A} \vee B)$$

$$(A \vee B) \& (\bar{A} \vee B)$$

A	B	$A \vee B$	$\bar{A}$	$\bar{A} \vee B$	$(A \vee B) \& (\bar{A} \vee B)$
0	0	0	1	1	0
0	1	1	1	1	1
1	0	1	0	0	0
1	1	1	0	1	1

**Равносильные логические выражения.** Логические выражения, у которых последние столбцы таблиц истинности совпадают, называются *равносильными*. Для обозначения равносильных логических выражений используется знак  $=$ .

Например:  $\bar{A} \& \bar{B} = \overline{A \vee B}$

# ЗАПИСЬ ЛОГИЧЕСКОГО ВЫРАЖЕНИЯ ПО ТАБЛИЦЕ ИСТИННОСТИ

Правила построения логического выражения:

1. Для каждой строки таблицы истинности с единичным значением функции построить **минтерм**.  
**Минтермом** называется произведение, в котором каждая переменная встречается только один раз — либо с отрицанием, либо без него. Переменные, имеющие нулевые значения в строке, входят в минтерм с отрицанием, а переменные со значением 1 — без отрицания.
2. Объединить все минтермы операцией дизъюнкция (логическое сложение), что даст стандартную сумму произведений для заданной таблицы истинности.

**Пример.** Дана таблица истинности:

Построим логическое выражение для F.

Найдем строки, в которых F=1.

Это вторая, третья и шестая.

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Для второй строки A=0, B=0, C=1. Эту строку описывает минтерм  $\bar{A} \& \bar{B} \& C$

Для третьей строки A=0, B=1, C=0. Эту строку описывает минтерм  $\bar{A} \& B \& \bar{C}$

Для шестой строки A=1, B=0, C=1. Эту строку описывает минтерм  $A \& \bar{B} \& C$

Объединяя термы, получим булево выражение

$$F = \bar{A} \& \bar{B} \& C \vee \bar{A} \& B \& \bar{C} \vee A \& \bar{B} \& C$$

В это выражение вошли термы-произведения для строк с единичным значением функции F, а вся сумма соответствует совокупности из трех строк. Для остальных пяти наборов значений входных переменных это выражение равно нулю.

# Логические функции

- Любое логическое выражение (составное высказывание) можно рассматривать как логическую функцию  $F(X_1, X_2, \dots, X_n)$  аргументами которой являются логические переменные  $X_1, X_2, \dots, X_n$  (простые высказывания). Сама функция как и аргументы могут принимать только два различных значения: «истина» (1) и «ложь» (0).
- Выше были рассмотрены функции двух аргументов: логическое умножение  $F(A, B) = A \& B$ , логическое сложение  $F(A, B) = A \vee B$ , а также логическое отрицание  $F(A) = \neg A$ , в котором значение второго аргумента можно считать равным нулю.
- Каждая логическая функция двух аргументов имеет четыре возможных набора значений аргументов. Может существовать  $N = 2^4 = 16$  различных логических функций двух аргументов.
- Таким образом, существует 16 различных логических функций двух аргументов, каждая из которых задается своей таблицей истинности :

Аргументы		Логические функции															
<i>A</i>	<i>B</i>	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$	$F_9$	$F_{10}$	$F_{11}$	$F_{12}$	$F_{13}$	$F_{14}$	$F_{15}$	$F_{16}$
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Легко заметить, что здесь логическая функция  **$F_2$**  является функцией логического умножения,  **$F_8$**  — функцией логического сложения,  **$F_{13}$**  — функцией логического отрицания для аргумента *A* и  **$F_{11}$**  — функцией логического отрицания для аргумента *B*. В обыденной и научной речи кроме базовых логических связок «и», «или», «не» используются и некоторые другие: «если... то...», «... тогда и только тогда, когда...» и др. Некоторые из них имеют свое название и свой символ, и им соответствуют определенные логические функции.

# ИМПЛИКАЦИЯ (ЛОГИЧЕСКОЕ СЛЕДОВАНИЕ).

- Импликация двух высказываний  $A$  и  $B$  соответствует союзу «ЕСЛИ...ТО». Она обозначается символом  $\rightarrow$
- Запись  $A \rightarrow B$  читается как «из  $A$  следует  $B$ »
- **Импликация двух высказываний истинна всегда, кроме случая, если первое высказывание истинно, а второе ложно.**
- Таблица истинности импликации двух суждений  $A$  и  $B$  такова:

$A$	$B$	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

В программировании эту операцию обозначают «IMP».

$$A \rightarrow B = \bar{A} \vee B$$

# ЭКВИВАЛЕНТНОСТЬ (ЛОГИЧЕСКОЕ РАВЕНСТВО, ФУНКЦИЯ ТОЖДЕСТВА)

- Она обозначается символами  $\equiv$  или  $\Leftrightarrow$ . («тогда и только тогда»).
- Запись  $A \equiv B$  читается как «А эквивалентно В».
- **Эквивалентность двух высказываний истинна только в тех случаях, когда оба высказывания ложны или оба истинны.**
- **Таблица истинности эквивалентности двух суждений А и В такова:**

A	B	$A \equiv B$
0	0	1
0	1	0
1	0	0
1	1	1

В программировании эту операцию обозначают «EQV».

В алгебре высказываний все логические функции могут быть сведены путём логических преобразований к трём базовым логическим операциям: инверсии, дизъюнкции и конъюнкции

# Логические законы и правила преобразования логических выражений

Равносильности формул логики высказываний часто называют *законами логики*. Законы логики отражают наиболее важные закономерности логического мышления.

В алгебре высказываний законы логики записываются в виде формул, которые позволяют проводить эквивалентные преобразования логических выражений в соответствии с законами логики.

Знание законов логики позволяет проверять правильность рассуждений и доказательств. Нарушения этих законов приводят к логическим ошибкам и вытекающим из них противоречиям.

Перечислим наиболее важные из них:

**1. Закон тождества.** Всякое высказывание тождественно самому себе:

$$A = A$$

Этот закон сформулирован древнегреческим философом Аристотелем. **Закон тождества** утверждает, что мысль, заключенная в некотором высказывании, остается неизменной на протяжении всего рассуждения, в котором это высказывание фигурирует.

**2. Закон непротиворечия.** Высказывание не может быть одновременно истинным и ложным. Если высказывание **A** — истинно, то его отрицание **не A** должно быть ложным. Следовательно, логическое произведение высказывания и его отрицания должно быть ложно:

$$\bar{A} \& A = 0$$

Закон непротиворечия говорит о том, что никакое предложение не может быть истинно одновременно со своим отрицанием.

*“Это яблоко спелое” и “Это яблоко не спелое”*

**3. Закон исключенного третьего.** Высказывание может быть либо истинным, либо ложным, третьего не дано. Это означает, что результат логического сложения высказывания и его отрицания всегда принимает значение истина:

$$A \vee \bar{A} = 1$$

Закон исключенного третьего говорит о том, что для каждого высказывания имеются лишь две возможности: это высказывание либо истинно, либо ложно. Третьего не дано.

*“Сегодня я получу 5 либо не получу”*. Истинно либо суждение, либо его отрицание.

**4. Закон двойного отрицания.** Если дважды отрицать некоторое высказывание, то в результате мы получим исходное высказывание:

$$\bar{\bar{A}} = A$$

Закон двойного отрицания. Отрицать отрицание какого-нибудь высказывания - то же, что утверждать это высказывание.

*“Неверно, что 2х 2<sup>1</sup> 4”*

**5. Законы идемпотентности.** В алгебре логики нет показателей степеней и коэффициентов.

Конъюнкция одинаковых «сомножителей» равносильна одному из них:

$$A \& A = A$$

Дизъюнкция одинаковых «слагаемых» равносильна одному:

$$A \vee A = A$$

**6. Законы де Моргана:**

$$\overline{A \vee B} = \bar{A} \& \bar{B}$$

$$\overline{A \& B} = \bar{A} \vee \bar{B}$$

Смысл законов де Моргана (Август де Морган (1806-1871) - шотландский математик и логик) можно выразить в кратких словесных формулировках:

отрицание логической суммы эквивалентно логическому произведению отрицаний слагаемых;

отрицание логического произведения эквивалентно логической сумме отрицаний множителей.

**7. Правило коммутативности.** В обычной алгебре слагаемые и множители можно менять местами. В алгебре высказываний можно менять местами логические переменные при операциях логического умножения и логического сложения:

Логическое умножение:

$$A \& B = B \& A$$

Логическое сложение:

$$A \vee B = B \vee A$$

**8. Правило ассоциативности.** Если в логическом выражении используются только операция логического умножения или только операция логического сложения, то можно пренебрегать скобками или произвольно их расставлять:

Логическое умножение:  $(A \& B) \& C = A \& (B \& C)$

Логическое сложение:  $(A \vee B) \vee C = A \vee (B \vee C)$

**9. Правило дистрибутивности.** В отличие от обычной алгебры, где за скобки можно выносить только общие множители, в алгебре высказываний можно выносить за скобки, как общие множители, так и общие слагаемые:

Дистрибутивность умножения относительно сложения:

$$(A \& B) \vee (A \& C) = A \& (B \vee C)$$

Дистрибутивность сложения относительно умножения:

$$(A \vee B) \& (A \vee C) = A \vee (B \& C)$$

10.  $A \& 1 = A$   $A \& 0 = 0$

11.  $A \vee 1 = 1$   $A \vee 0 = A$

**12. Законы поглощения:**

$$A \& (A \vee B) = A \quad A \vee (A \& B) = A \& B$$



# РЕШЕНИЕ ЛОГИЧЕСКИХ ЗАДАЧ

### ЗАДАЧА 1.

Разбирается дело Лёнчика, Пончика и Батончика. Кто-то из них нашел и утаил клад. На следствии каждый из них сделал по два заявления.

Батончик: «Я не делал этого. Пончик сделал это»

Лёнчик: «Пончик не виновен. Батончик сделал это»

Пончик: «Я не делал этого. Лёнчик не делал этого»

Суд установил, что один из них дважды солгал, другой — дважды сказал правду, третий — один раз солгал, один раз сказал правду. Кто утаил клад?

#### Решение:

Введём обозначения: Б –клад утаил Батончик, П - клад утаил Пончик, Л - клад утаил Лёнчик. Рассмотрим три возможных варианта – виноват Батончик, виноват Пончик, виноват Лёнчик. При таких вариантах получаем следующие значения высказываний трёх обвиняемых.

Возможные варианты			Высказывания Батончика		Высказывания Лёнчика		Высказывания Пончика		Соответствие условию задачи
Б	Л	П	¬Б	П	¬П	Б	¬П	¬Л	
1	0	0	0	0	1	1	1	1	-
0	0	1	1	1	0	0	0	1	+
0	1	0	1	0	1	0	1	0	-

В первом варианте один солгал дважды, а двое сказали правду дважды, что не соответствует условию задачи. В третьем варианте все один раз сказали правду и один раз солгали, что также не соответствует условию задачи. Во втором варианте один дважды солгал, другой дважды сказал правду, а третий один раз сказал правду, а один раз солгал, что соответствует условию задачи. Следовательно клад утаил Пончик.

## Задача 2.

В школьном первенстве по настольному теннису в четверку лучших вошли девушки: Наташа, Маша, Люда и Рита. Самые горячие болельщики высказали свои предположения о распределении мест в дальнейших состязаниях. Один считает, что первой будет Наташа, а Маша будет второй. Другой болельщик на второе место прочит Люду, а Рита, по его мнению, займет четвертое место. Третий любитель тенниса с ними не согласился. Он считает, что Рита займет третье место, а Наташа будет второй. Когда соревнования закончились, оказалось, что каждый из болельщиков был прав только в одном из своих прогнозов. Какое место на чемпионате заняли Наташа, Маша, Люда, Рита?

### Решение:

Введём обозначения: Н1 – Наташа на 1 месте, М2 – Маша на 2 месте, Л2 – Люда на 2 месте, Р4 – Рита на 4 месте, Р3 – Рита на 3 месте, Н2 – Наташа на 2 месте. Занесём возможные варианты высказываний трёх болельщиков в таблицу с учётом того, что каждый из болельщиков оказался прав только в одном из своих прогнозов:

Высказывания 1-ого болельщика		Высказывания 2-ого болельщика		Высказывания 2-ого болельщика		Соответствие условию задачи
Н1	М2	Л2	Р4	Р3	Н2	
0	1	0	1	0	1	-
0	1	0	1	1	0	-
0	1	1	0	1	0	-
0	1	1	0	0	1	-
1	0	0	1	0	1	-
1	0	0	1	1	0	-
1	0	1	0	0	1	-
1	0	1	0	1	0	+

Из анализа таблицы видно, что условию задачи соответствует только последняя строка, значит первое место заняла Наташа, второе – Люда, третье – Рита, а Маша – четвертое.

### Задача 3.

Вадим, Сергей и Михаил изучают различные иностранные языки: китайский, японский и арабский. На вопрос, какой язык изучает каждый из них, один ответил: "Вадим изучает китайский, Сергей не изучает китайский, а Михаил не изучает арабский". Впоследствии выяснилось, что в этом ответе только одно утверждение верно, а два других ложны. Какой язык изучает каждый из молодых людей?

#### Решение:

Введём обозначения: ВК – Вадим изучает китайский язык, СК – Сергей изучает китайский язык, МА - Михаил изучает арабский язык. Занесём в таблицу возможные варианты значений высказываний с учётом условия задачи, что одно из утверждений верно, а два - ложны:

Возможные варианты высказываний						Соответствие условию задачи
ВК	¬СК	¬МА	ВК	СК	МА	
1	0	0	<b>1</b>	<b>1</b>	1	-
0	0	1	0	1	0	+
0	1	0	<b>0</b>	<b>0</b>	1	-

Проанализируем строки в трёх последних столбцах. Условию задачи соответствует только вторая строка, значит Сергей изучает китайский язык, Михаил – японский (так как он не изучает арабский), тогда Вадим изучает арабский язык.

**Задача 4.** Три одноклассника — Влад, Тимур и Юра, встретились спустя 10 лет после окончания школы. Выяснилось, что один из них стал врачом, другой физиком, а третий юристом. Один полюбил туризм, другой бег, страсть третьего — регби.

•Юра сказал, что на туризм ему не хватает времени, хотя его сестра — единственный врач в семье, заядлый турист. Врач сказал, что он разделяет увлечение коллеги.

•Забавно, но у двоих из друзей в названиях их профессий и увлечений не встречается ни одна буква их имен.

Определите, кто чем любит заниматься в свободное время и у кого какая профессия.

*Решение:* Здесь исходные данные разбиваются на тройки (имя — профессия — увлечение).

Из слов Юры ясно, что он не увлекается туризмом и он не врач. Из слов врача следует, что он турист.

Имя	Юра		
Профессия		врач	
Увлечение		туризм	

Буква "а", присутствующая в слове "врач", указывает на то, что Влад тоже не врач, следовательно врач — Тимур. В его имени есть буквы "т" и "р", встречающиеся в слове "туризм", следовательно второй из друзей, в названиях профессии и увлечения которого не встречается ни одна буква его имени — Юра. Юра не юрист и не регбист, так как в его имени содержатся буквы "ю" и "р".

Следовательно, окончательно имеем:

Имя	Юра	Тимур	Влад
Профессия	физик	врач	юрист
Увлечение	бег	туризм	регби

*Ответ.* Влад — юрист и регбист, Тимур — врач и турист, Юра — физик и бегун.

# Задачи для самостоятельного решения

**Задача 1.** Трое друзей, болельщиков автогонок "Формула-1", спорили о результатах предстоящего этапа гонок.  
— Вот увидишь, Шумахер не придет первым, — сказал Джон. Первым будет Хилл.  
— Да нет же, победителем будет, как всегда, Шумахер, — воскликнул Ник. — А об Алезе и говорить нечего, ему не быть первым.

Питер, к которому обратился Ник, возмутился:

— Хиллу не видать первого места, а вот Алезе пилотирует самую мощную машину.

По завершении этапа гонок оказалось, что каждое из двух предположений двоих друзей подтвердилось, а оба предположения третьего из друзей оказались неверны. Кто выиграл этап гонки?

**Задача 2.** В спортивных соревнованиях принимали участие пять команд: "Вымпел", "Метеор", "Нептун", "Старт" и "Чайка". Об их итогах соревнования имеет пять высказываний:

- 1) Второе место занял "Вымпел", а "Старт" оказался на третьем.
- 2) Хорошо выступала команда "Нептун", она стала победителем, а "Чайка" вышла на второе место.
- 3) Да нет же, "Чайка" заняла только третье место, а "Нептун"- был последним.
- 4) Первое место по праву завоевал "Старт", а "Метеор" был 4-м.
- 5) Да, "Метеор", действительно, был четвертым, а "Вымпел" был 2-м.

Известно, что команды не делили места между собой и что в каждом высказывании одно утверждение правильное, а другое нет.

Как распределились места между командами?

**Задача 3** Три дочери писательницы Дорис Кей — Джуди, Айрис и Линда, тоже очень талантливы. Они приобрели известность в разных видах искусств — пении, балете и кино. Все они живут в разных городах, поэтому Дорис часто звонит им в Париж, Рим и Чикаго.

Известно, что:

Джуди живет не в Париже, а Линда — не в Риме;

парижанка не снимается в кино;

та, кто живет в Риме, певица;

Линда равнодушна к балету.

Где живет Айрис, и какова ее профессия?



# **ЛОГИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРА**

# Логические элементы

В основе обработки компьютером информации лежит алгебра логики, разработанная Дж. Булем. Знания из области математической логики можно использовать для конструирования различных электронных устройств.

Нам известно, что 0 и 1 в логике не просто цифры, а обозначение состояний какого-то предмета нашего мира, условно называемых "ложь" и "истина". Таким предметом, имеющим два фиксированных состояния, может быть электрический ток. Были созданы устройства управления электричеством - электронные схемы, состоящие из набора полупроводниковых элементов. Такие электронные схемы, которые преобразовывают сигналы только двух фиксированных напряжений электрического тока стали называть *логическими элементами*.

***Логические элементы*** — это электронные устройства, которые преобразуют проходящие через них двоичные электрические сигналы по определенному закону.

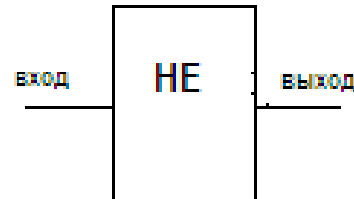
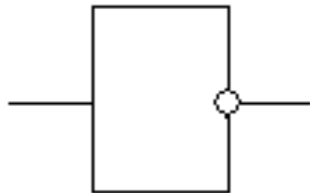
Логические элементы имеют один или несколько входов, на которые подаются электрические сигналы, обозначаемые условно **0**, если отсутствует электрический сигнал, и **1**, если имеется электрический сигнал.

Также логические элементы имеют один выход, с которого снимается преобразованный электрический сигнал.

Было доказано, что все электронные схемы компьютера могут быть реализованы с помощью трёх базовых логических элементов **И**, **ИЛИ**, **НЕ**.

# Логический элемент НЕ (инвертор)

Простейшим логическим элементом является *инвертор*, выполняющий функцию отрицания (инверсию). У этого элемента один вход и один выход. На функциональных схемах он обозначается:



Если на вход поступает сигнал, соответствующий 1, то на выходе будет 0. И наоборот.

<i>вход</i>	<i>выход</i>
1	0
0	1

# Логический элемент ИЛИ (дизъюнктор)

Логический элемент, выполняющий логическое сложение, называется *дизъюнктор*. Он имеет, как минимум, два входа. На функциональных схемах он обозначается:



Если хотя бы на один вход поступает сигнал 1, то на выходе будет сигнал 1.

<i>вход 1</i>	<i>вход 2</i>	<i>выход</i>
0	0	0
0	1	1
1	0	1
1	1	1

# Логический элемент И (конъюнктор)

Логический элемент, выполняющий логическое умножение, называется **конъюнктор**. Он имеет, как минимум, два входа. На функциональных схемах он обозначается:



На выходе этого элемента будет сигнал 1 только в том случае, когда на все входы поступает сигнал 1. Когда хотя бы на одном входе будет ноль, на выходе также будет ноль.

<i>вход 1</i>	<i>вход 2</i>	<i>выход</i>
0	0	0
0	1	0
1	0	0
1	1	1

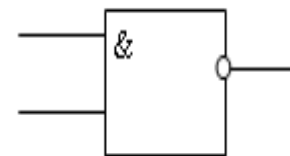
Другие логические элементы построены из трех простейших базовых элементов и выполняют более сложные логические преобразования информации.

Рассмотрим еще два логических элемента, которые играют роль базовых при создании более сложных элементов и схем.

### Логический элемент И-НЕ

Логический элемент И-НЕ выполняет логическую функцию штрих Шеффера (И-НЕ), он имеет, как минимум, два входа. На функциональных схемах он обозначается:

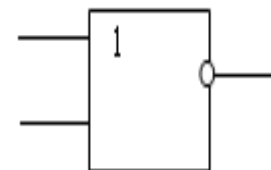
<i>вход 1</i>	<i>вход 2</i>	<i>выход</i>
0	0	1
0	1	1
1	0	1
1	1	0



### Логический элемент ИЛИ-НЕ

Логический элемент ИЛИ-НЕ выполняет логическую функцию стрелка Пирса (И-НЕ), он имеет, как минимум, два входа. На функциональных схемах он обозначается:

<i>вход 1</i>	<i>вход 2</i>	<i>выход</i>
0	0	1
0	1	0
1	0	0
1	1	0

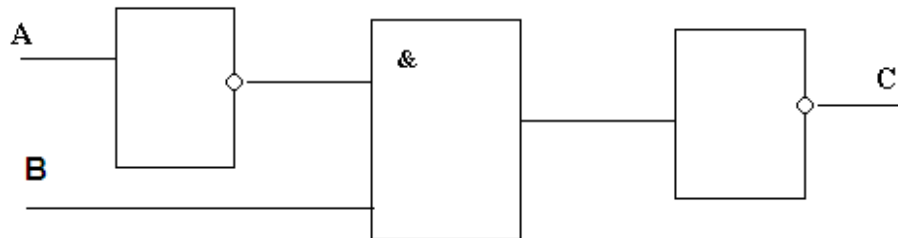


# Функциональные схемы

Сигнал, выработанный одним логическим элементом, можно подавать на вход другого элемента, это дает возможность образовывать цепочки из отдельных логических элементов — *функциональные схемы*.

**Функциональная (логическая) схема** – это схема, состоящая из логических элементов, которая выполняет определённую функцию. Анализируя функциональную схему, можно понять, как работает логическое устройство, т.е. дать ответ на вопрос: какую функцию она выполняет.

Важной формой описания функциональных схем является структурная формула. Покажем на примере, как выписывают формулу по заданной функциональной схеме.



Ясно, что элемент “И” осуществляет логическое умножение значений  $\neg A$  и  $B$ . Над результатом в элементе “НЕ” осуществляется операция отрицания, т.е. вычисляется значение выражения:

$$\overline{\overline{A} \& B}$$

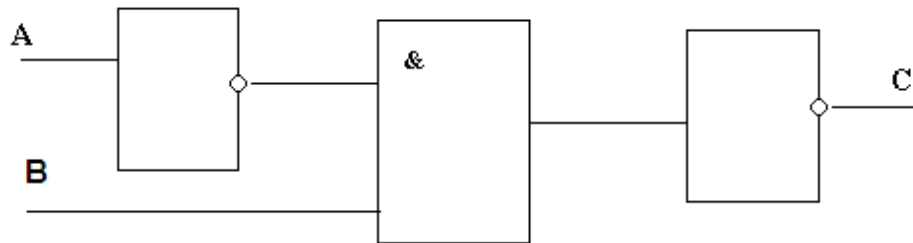
Таким образом структурной формулой данной функциональной схемы является формула:

$$C = \overline{\overline{A} \& B}$$

# Таблица истинности функциональной схемы

Для функциональной схемы можно составить таблицу истинности, то есть таблицу значений сигналов на входах и выходах схемы, по которой можно понять какую функцию выполняет данная схема. **Таблица истинности** - это табличное представление логической (функциональной) схемы в котором перечислены все возможные сочетания значений входных сигналов вместе со значением выходного сигнала для каждого из этих сочетаний.

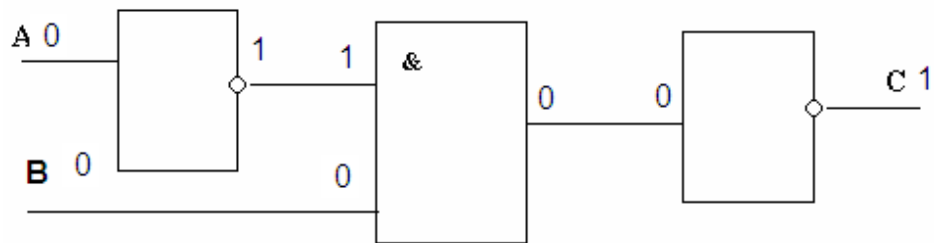
Составим таблицу истинности для данной логической схемы:



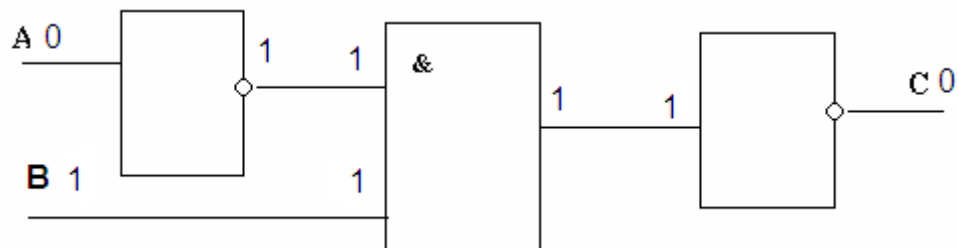
Начертим таблицу: количество столбцов = количество входов + количество выходов, количество строк =  $2^{\text{количество входов}}$ . В данной таблице 3 столбца и 4 строки. Заполним первые столбцы всеми возможными вариантами входных сигналов

A (вход 1)	B (вход 2)	C (выход)
0	0	
0	1	
1	0	
1	1	

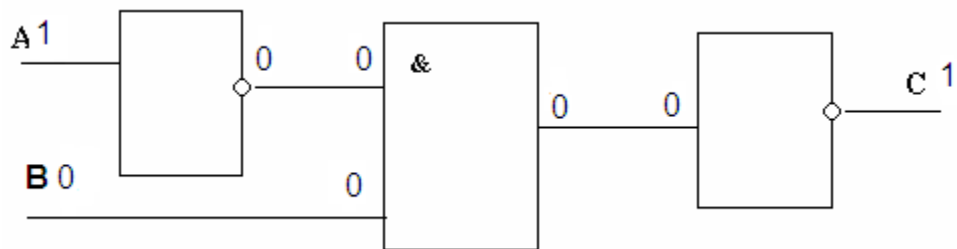
Рассмотрим первый вариант входных сигналов:  $A=0, B=0$ . Проследим по схеме, как проходят и преобразуются входные сигналы. Результат, полученный на выходе ( $C=1$ ), запишем в таблицу.



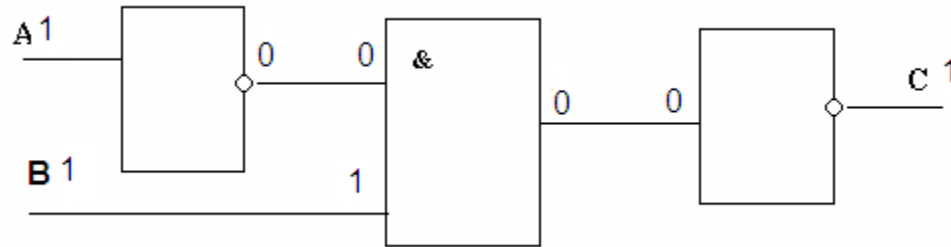
Рассмотрим второй вариант входных сигналов:  $A=0, B=1$ . Проследим по схеме, как проходят и преобразуются входные сигналы. Результат, полученный на выходе ( $C=0$ ), запишем в таблицу.



Рассмотрим третий вариант входных сигналов:  $A=1, B=0$ . Проследим по схеме, как проходят и преобразуются входные сигналы. Результат, полученный на выходе ( $C=1$ ), запишем в таблицу.



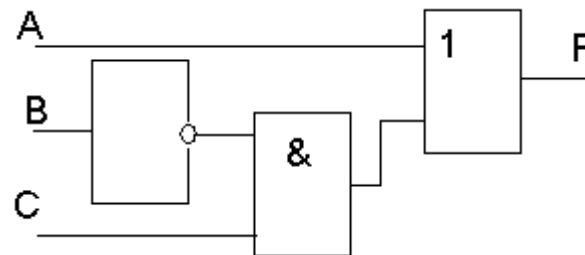
Рассмотрим четвёртый вариант входных сигналов:  $A=1$ ,  $B=1$ . Проследим по схеме, как проходят и преобразуются входные сигналы. Результат, полученный на выходе ( $C=1$ ), запишем в таблицу.



В результате получаем таблицу истинности данной логической схемы:

A (вход 1)	B (вход 2)	C (выход)
0	0	1
0	1	0
1	0	1
1	1	1

**Задание.** Построить таблицу истинности для данной логической схемы и записать формулу для данной схемы:



# Логическая реализация типовых устройств компьютера

Обработка любой информации на компьютере сводится к выполнению процессором различных арифметических и логических операций. Для этого в состав процессора входит так называемое арифметико-логическое устройство (АЛУ). Оно состоит из ряда устройств, построенных на рассмотренных выше логических элементах. Важнейшими из таких устройств являются ***триггеры, полусумматоры, сумматоры, шифраторы, дешифраторы, счетчики, регистры.***

Выясним , как из логических элементов разрабатываются логические устройства.

# Этапы конструирования логического устройства.

Конструирование логического устройства состоит из следующих этапов:

1. Построение таблицы истинности по заданным условиям работы проектируемого узла (т.е. по соответствию его входных и выходных сигналов).
2. Конструирование логической функции данного узла по таблице истинности, ее преобразование (упрощение), если это возможно и необходимо.
3. Составление функциональной схемы проектируемого узла по формуле логической функции.

После этого остается только реализовать полученную схему.

**Задание.** Построить логическую схему для заданной таблицы истинности:

Запишем логическую функцию по данной таблице истинности:

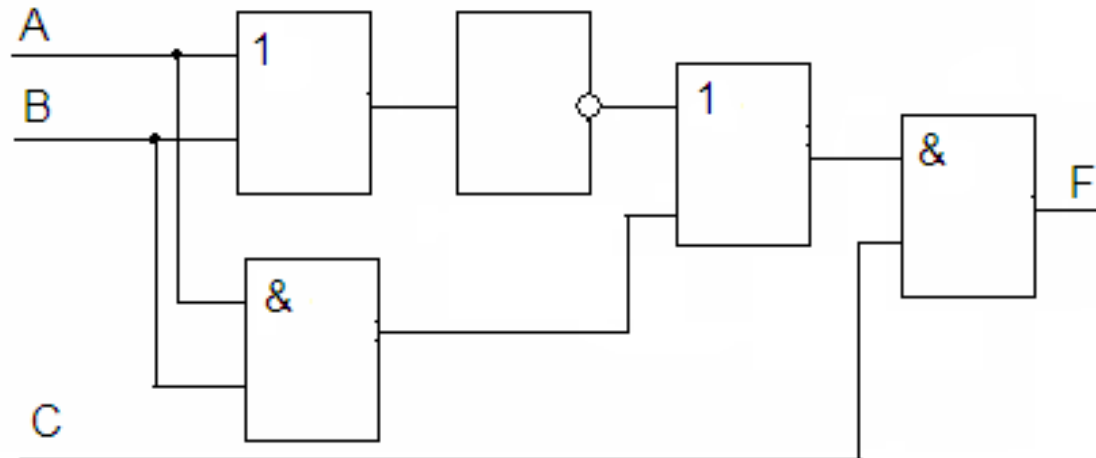
$$F = \bar{A} \& \bar{B} \& C \vee A \& B \& C$$

Упростим полученное логическое выражение:

$$F = C \& (\bar{A} \& \bar{B} \vee A \& B) = C \& ((\overline{A \vee B}) \vee A \& B)$$

Построим логическую схему для данного выражения:

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



Попробуем, действуя по этому плану, сконструировать устройство для сложения двух двоичных чисел (*одноразрядный полусумматор*).

Пусть нам необходимо сложить двоичные числа **A** и **B**. Через **P** и **S** обозначим первую и вторую цифру суммы: **A + B = PS**. Вспомните таблицу сложения двоичных чисел.

1. Таблица истинности, определяющая результат сложения, имеет вид:

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

2. Сконструируем функции  $P(A,B)$  и  $S(A,B)$  по этой таблице:

$$P(A, B) = A \& B$$

$$S(A, B) = \bar{A} \& B \vee A \& \bar{B}$$

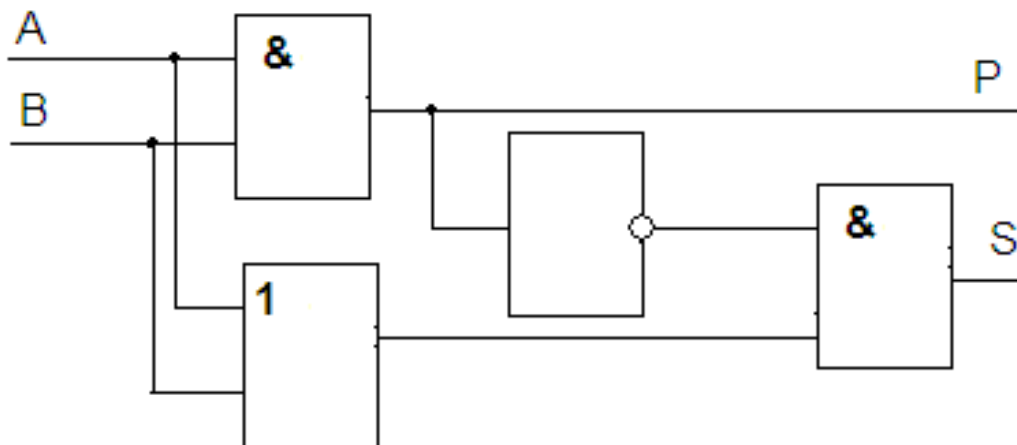
Преобразуем вторую формулу, пользуясь законами логики:

$$\begin{aligned} S(A, B) &= \bar{A} \& B \vee A \& \bar{B} = \bar{A} \& B \vee A \& \bar{B} \vee A \& \bar{A} \vee B \& \bar{B} = (\bar{A} \& A \vee \bar{A} \& B) \vee (A \& \bar{B} \vee B \& \bar{B}) = \\ &= \bar{A} \& (A \vee B) \vee \bar{B} \& (A \vee B) = (A \vee B) \& (\bar{A} \& \bar{B}) = (A \vee B) \& \overline{(A \& B)} \end{aligned}$$

3. Теперь можно построить функциональную схему одnorазрядного полусумматора:

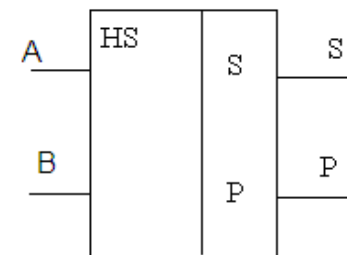
$$P(A, B) = A \& B$$

$$S(A, B) = (A \vee B) \& \overline{(A \& B)}$$



Чтобы убедиться в том, как работает схема, проследите за прохождением сигналов в каждом из четырёх случаев и составьте таблицу истинности данной логической схемы.

**Условное обозначение одnorазрядного сумматора:**



# Полный одноразрядный сумматор.

Одноразрядный двоичный сумматор на три входа и два выхода называется *полным одноразрядным сумматором*.

Логика работы одноразрядного сумматора на три входа или полного сумматора приведена в таблице, где **A**, **B** - суммируемые двоичные цифры, **P<sub>0</sub>** - перенос из младшего разряда, **S** - образующаяся сумма данного разряда и осуществляет перенос **P** в следующий старший разряд.

Слагаемые		Перенос из младшего разряда	Сумма	Перенос
A	B	P <sub>0</sub>	S	P
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Формула переноса:  $P = A \& B \& \bar{P}_0 \vee \bar{A} \& B \& P_0 \vee A \& \bar{B} \& P_0 \vee A \& B \& P_0$

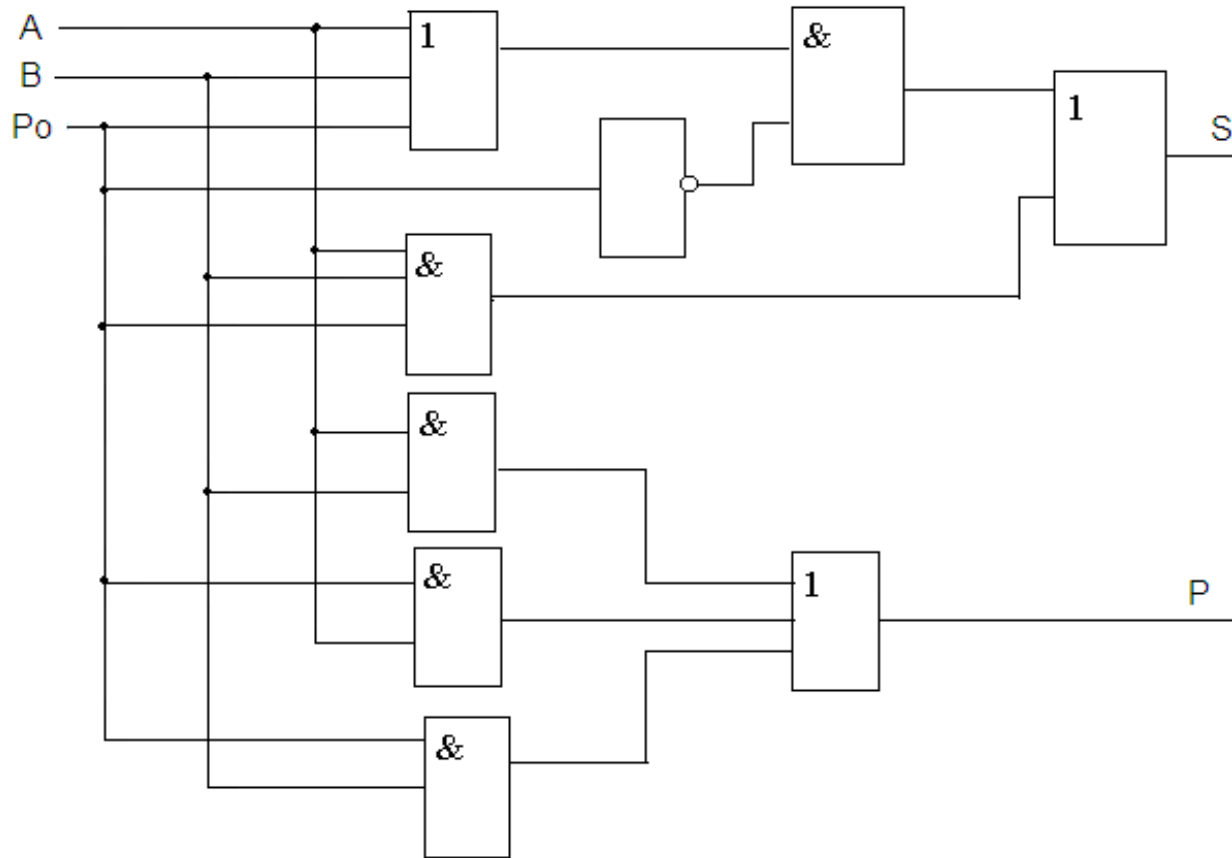
Формула для вычисления суммы  $S = \bar{A} \& B \& \bar{P}_0 \vee A \& \bar{B} \& \bar{P}_0 \vee \bar{A} \& \bar{B} \& P_0 \vee A \& B \& P_0$

После преобразования формулы переноса и суммы принимают вид:

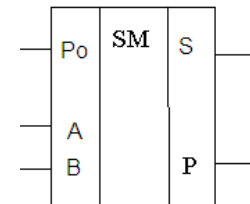
$$P = A \& B \vee A \& P_0 \vee B \& P_0$$

$$S = (A \vee B \vee P_0) \& \bar{P}_0 \vee (A \& B \& P_0)$$

Теперь можно построить схему полного одноразрядного сумматора с учётом переноса из младшего разряда.

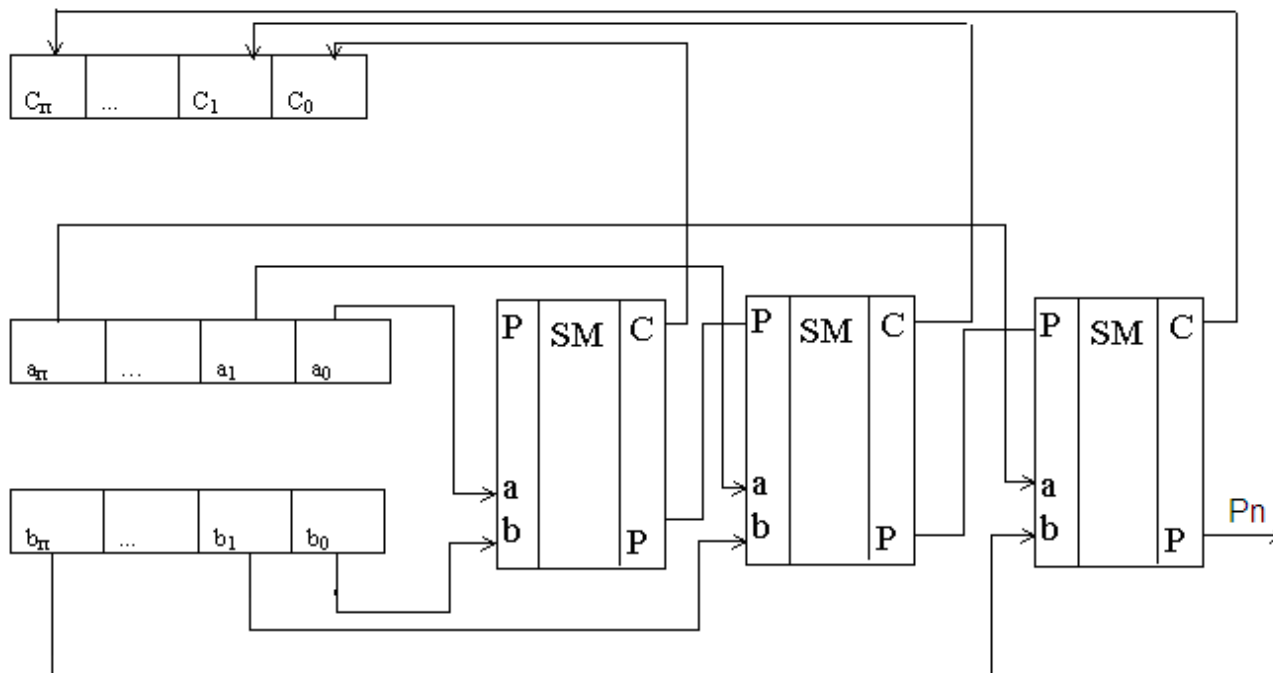


**Сумматор** - это электронная логическая схема, выполняющая суммирование двоичных чисел поразрядным сложением. Сумматор является центральным узлом арифметико-логического устройства процессора. Находит он применение и в других устройствах компьютера. В реальных электронных схемах сумматор изображается так:



Сумматор выполняет сложение *многозначных двоичных чисел*. Он представляет собой последовательное соединение *одноразрядных двоичных сумматоров*, каждый из которых осуществляет сложение в одном разряде. Если при этом возникает переполнение разряда, то перенос суммируется с содержимым старшего соседнего разряда.

На рисунке показано, как из N сумматоров можно составить устройство для сложения двух N-разрядных двоичных кодов, это схема многоразрядного сумматора.



# ТРИГГЕР

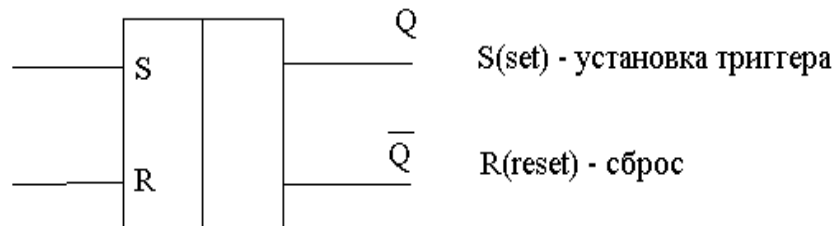
**Триггер** - электронная схема, применяемая для хранения значения одноразрядного двоичного кода.

Воздействуя на входы триггера, его переводят в одно из двух возможных состояний (0 или 1). С поступлением сигналов на входы триггера в зависимости от его состояния либо происходит переключение, либо исходное состояние сохраняется. При отсутствии входных сигналов триггер сохраняет свое состояние сколь угодно долго.

Термин *триггер* происходит от английского слова *trigger* - защёлка, спусковой крючок. Для обозначения этой схемы в английском языке чаще употребляется термин *flip-flop*, что в переводе означает "хлопанье". Это звукоподражательное название электронной схемы указывает на её способность почти мгновенно переходить ("перебрасываться") из одного электрического состояния в другое.

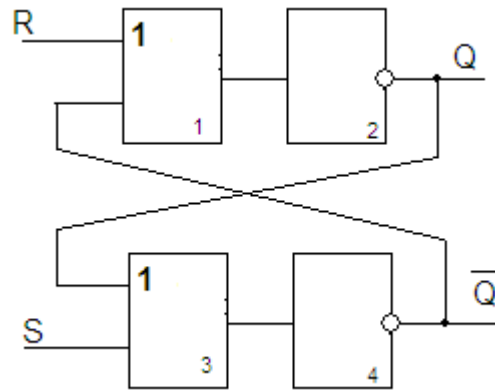
Существуют разные варианты исполнения триггеров в зависимости от элементной базы (И-НЕ, ИЛИ-НЕ) и функциональных связей между сигналами на входах и выходах (*RS*, *JK*, *T*, *D* и другие).

Самый распространённый тип триггера - это *RS*-триггер (*S* и *R* соответственно от английских *set* - установка, и *reset* - сброс). Условное обозначение *RS*-триггера:



# RS-триггер

RS-триггер построен на 2-х логических элементах: ИЛИ - НЕ либо И – НЕ.  
Как, правило, триггер имеет 2 выхода: прямой и инверсный ( $\bar{Q}$ )



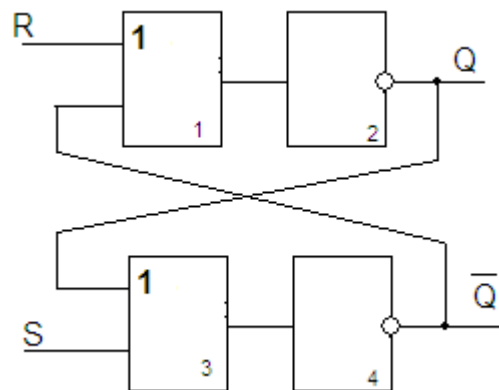
## Как он работает?

Пусть на вход элемента №1 подан сигнал 1, а на вход элемента №3 - 0. На выходе элемента №1 независимо от того, какой второй сигнал поступит на вход, будет 1, т.к. это элемент ИЛИ (по свойствам дизъюнкции). Пройдя через элемент №2 сигнал примет значение 0 ( $Q=0$ ).

Следовательно, и на втором входе элемента №3 установится сигнал 0. На выходе элемента №3 - 0. Пройдя через элемент №4 сигнал изменится на 1. Следовательно,  $\bar{Q}=1$ .

Убедимся, что данное устройство сохраняет информацию. Запомним, что  $S=0, R=1, Q=0, \bar{Q}=1$ . В момент прекращения входных сигналов ( $S=0, R=0$ ) на выходе =1. Это напряжение подается на вход элемента №1. На выходе элемента №1 сохраняется 1, и на Q - сигнал 0. На входах элемента №3 - 0, следовательно  $\bar{Q}=1$ . Таким образом, при отсутствии на внешних входах сигналов 1 триггер поддерживает постоянное напряжение на своих выходах. Чтобы изменить напряжение на выходах триггера, надо подать сигнал 1 на вход элемента №3. Тогда  $Q=1, \bar{Q}=0$ .

# RS-триггер



Вход		Выход		Режим работы
S	R	Q	$\bar{Q}$	
0	0	0	0	Хранение
1	0	1	0	Запись 1
0	1	0	1	Запись 0
1	1	X	X	Запрещение ( $Q \neq \bar{Q}$ )

# РЕГИСТРЫ

Функциональная схема компьютера, состоящая из триггеров, предназначенная для запоминания многоразрядных кодов и выполнения над ними некоторых логических преобразований называется *регистром*.

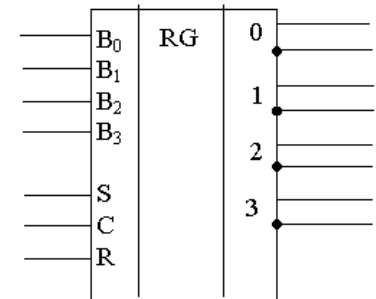
Упрощенно регистр можно представить как совокупность ячеек, в каждой из которых может быть записано одно из двух значений: 0 или 1, то есть один разряд двоичного числа.

С помощью регистров можно выполнять следующие операции: установку, сдвиг, преобразование. Основными типами регистров являются параллельные и последовательные (сдвигающие).

Совокупность регистров, используемых ЭВМ для запоминания программы работы, исходных и промежуточных результатов называется оперативной памятью (ОП).

Регистры содержатся в различных вычислительных узлах компьютера - процессоре, периферийных устройствах и т.д.

**Регистр** - это устройство, предназначенное для хранения многоразрядного двоичного числового кода, которым можно представлять и адрес, и команду, и данные.



# РЕГИСТРЫ

Существует несколько типов регистров, отличающихся видом выполняемых операций.

Некоторые важные регистры имеют свои названия, например:

***сдвиговый регистр*** - предназначен для выполнения операции сдвига;

***счетчики*** - схемы, способные считать поступающие на вход импульсы. К ним относятся *T*-триггеры (название от англ. *tumble* - опрокидываться). Этот триггер имеет один счетный вход и два выхода. Под действием сигналов триггер меняет свое состояние с нулевого на единичное и наоборот. Число перебрасываний соответствует числу поступивших сигналов;

***счетчик команд*** - регистр устройства управления процессора (УУ), содержимое которого соответствует адресу очередной выполняемой команды; служит для автоматической выборки программы из последовательных ячеек памяти;

***регистр команд*** - регистр УУ для хранения кода команды на период времени, необходимый для ее выполнения. Часть его разрядов используется для хранения кода операции, остальные - для хранения кодов адресов операндов.

В ЭВМ применяются регистры 8, 16, 32, 48 и 64 разрядов.

# ШИФРАТОРЫ И ДЕШИФРАТОРЫ

Шифратор и дешифратор являются типовыми узлами ЭВМ.

**Шифратор (кодер)** - это логическое устройство, которое преобразует единичный сигнал на одном из входов в  $n$ -разрядный двоичный код. Наибольшее применение он находит в устройствах ввода информации (например в клавиатуре), для преобразования десятичных чисел в двоичную систему счисления.

**Дешифратор (декодер)** - это логическое устройство, преобразующее двоичный код, поступающий на его входы, в сигнал только на одном из его выходов. Дешифраторы широко применяются в устройствах управления, в системах цифровой индикации с газоразрядными индикаторами, для построения распределителей импульсов по различным цепям и т.д. Схема используется для перевода двоичных цифр в десятичные. Дешифратор двоичного  $n$ -разрядного кода имеет  $2^n$  выходов, т.к. каждому из  $2^n$  значений входного кода должен соответствовать единичный сигнал на одном из выходов дешифратора.