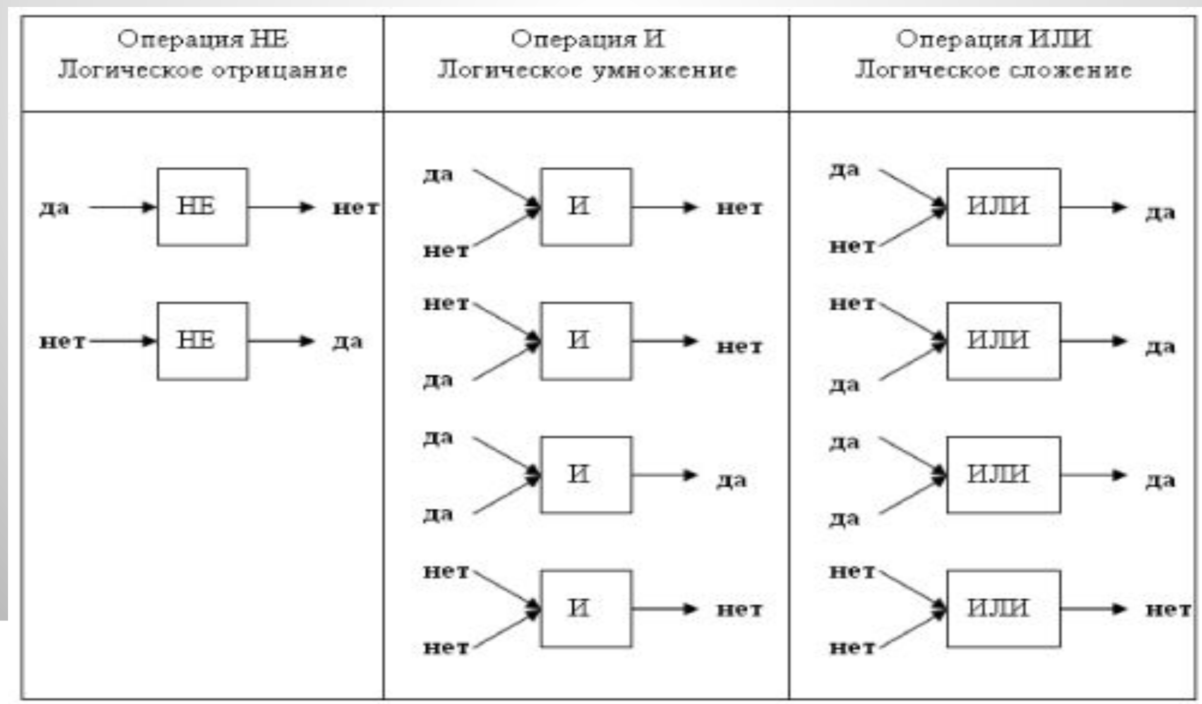
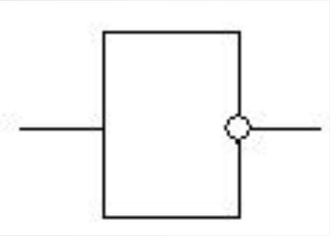
Логические схемы

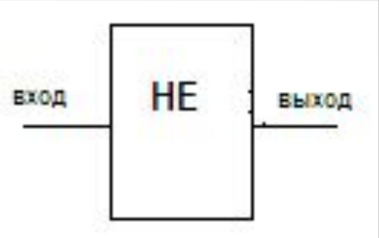
Логические схемы нужны для того чтобы в наглядной графической форме отобразить последовательность выполнения операций при вычислении логических формул.
Входящие слева линии и цифры около них обозначают значения операндов, линия справа и соответствующая цифра - результат операции (значение на выходе логических элементов). 1 - это логическая единица (истина), 0 - логический ноль (ложь).
Таблицы истинности в форме логических схем будут выглядеть т.о.



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:



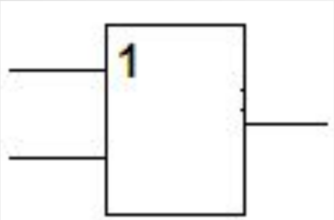


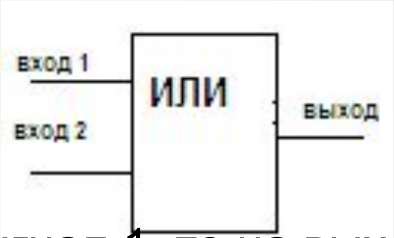
Если на вход поступает сигнал, соответствующий 1, то на выходе будет 0. И наоборот.

вход	*выход*
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:



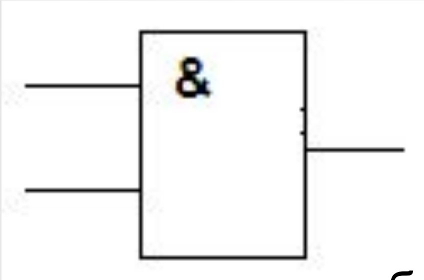


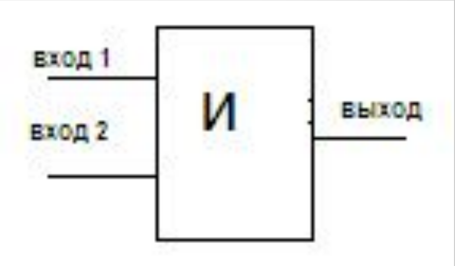
Если хотя бы на один вход поступает сигнал 1, то на выходе будет сигнал 1.

вход 1	*вход 2*	*выход*
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.

вход 1	*вход 2*	*выход*
0	**0**	**0**
0	**1**	**0**
1	**0**	**0**
1	**1**	**1**

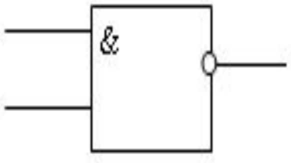
Другие логические элементы построены из трех простейших базовых элементов и выполняют более сложные логические преобразования информации.

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:

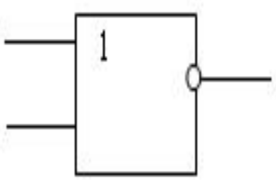
вход 1	*вход 2*	*выход*
0	**0**	**1**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**

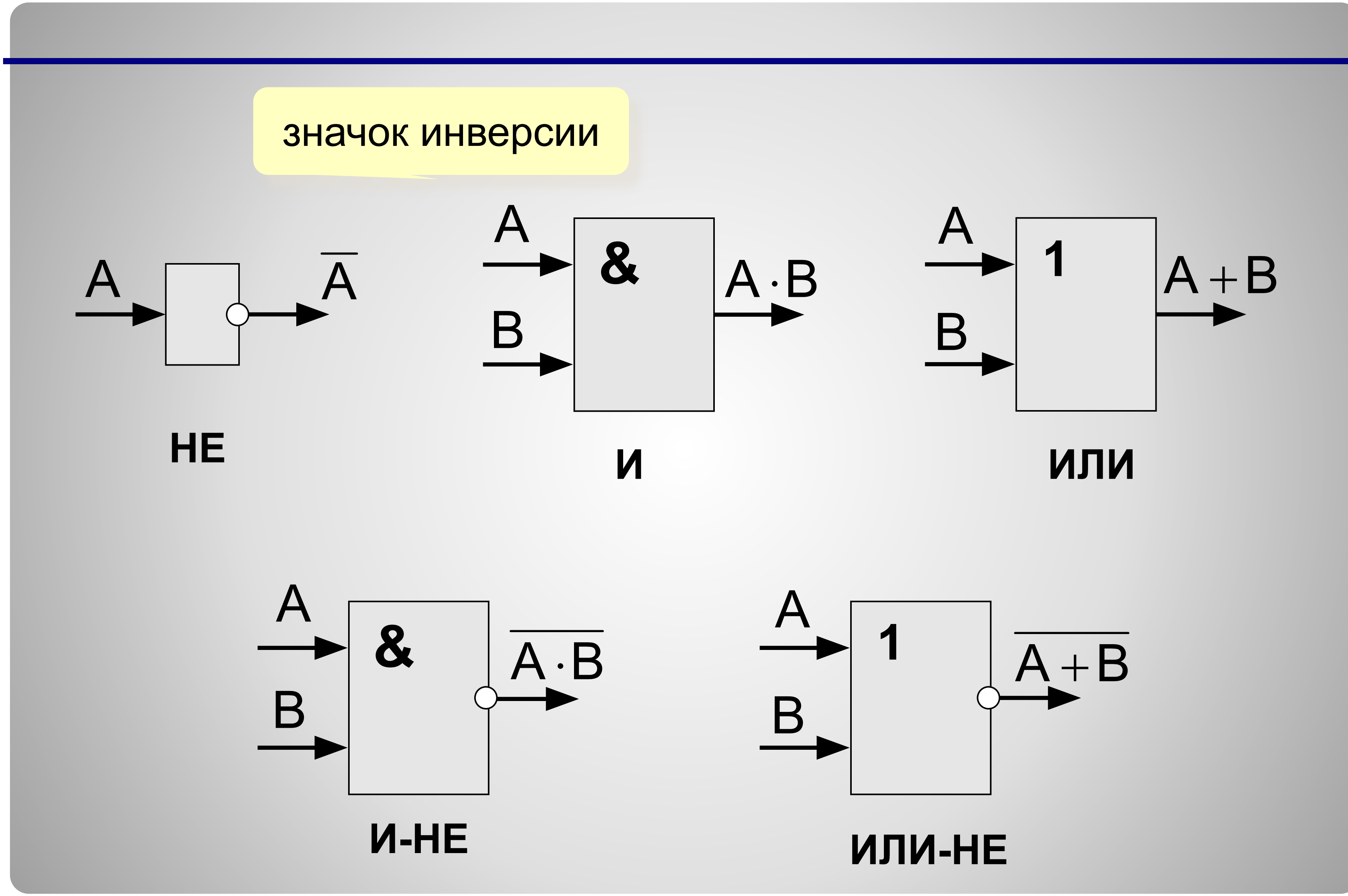


Логический элемент ИЛИ-НЕ

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

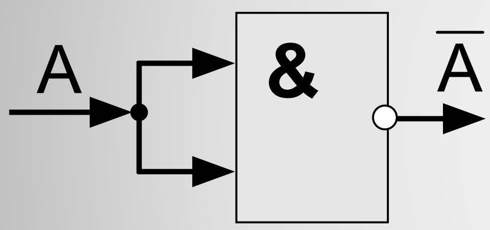
вход 1	*вход 2*	*выход*
0	**0**	**1**
0	**1**	**0**
1	**0**	**0**
1	**1**	**0**



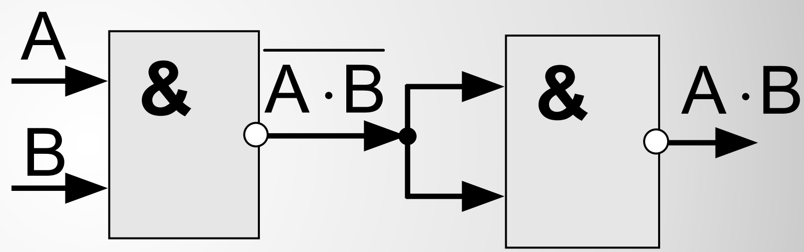

значок инверсии
A
A
НЕ
A
B
&
A·B
И
A
B
1
A+B
ИЛИ
A
B
&
A·B
И-НЕ
A
B
1
A+B
ИЛИ-НЕ

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

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

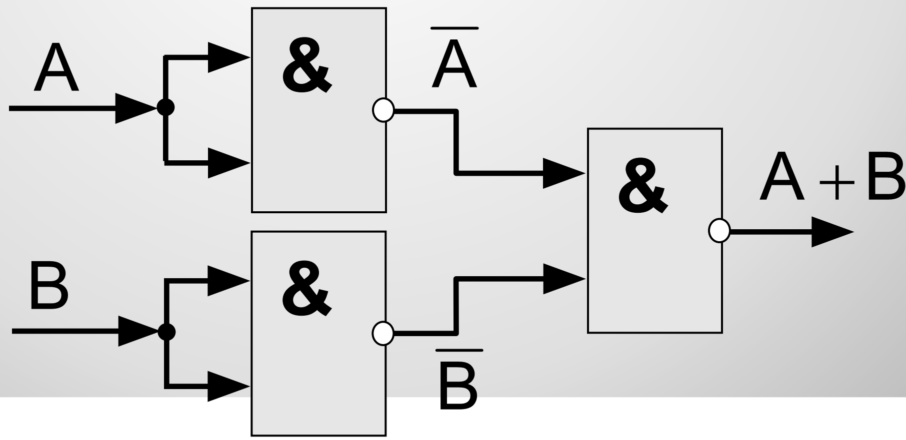


И: $A \cdot B = \overline{\overline{A \cdot B}}$



ИЛИ:

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

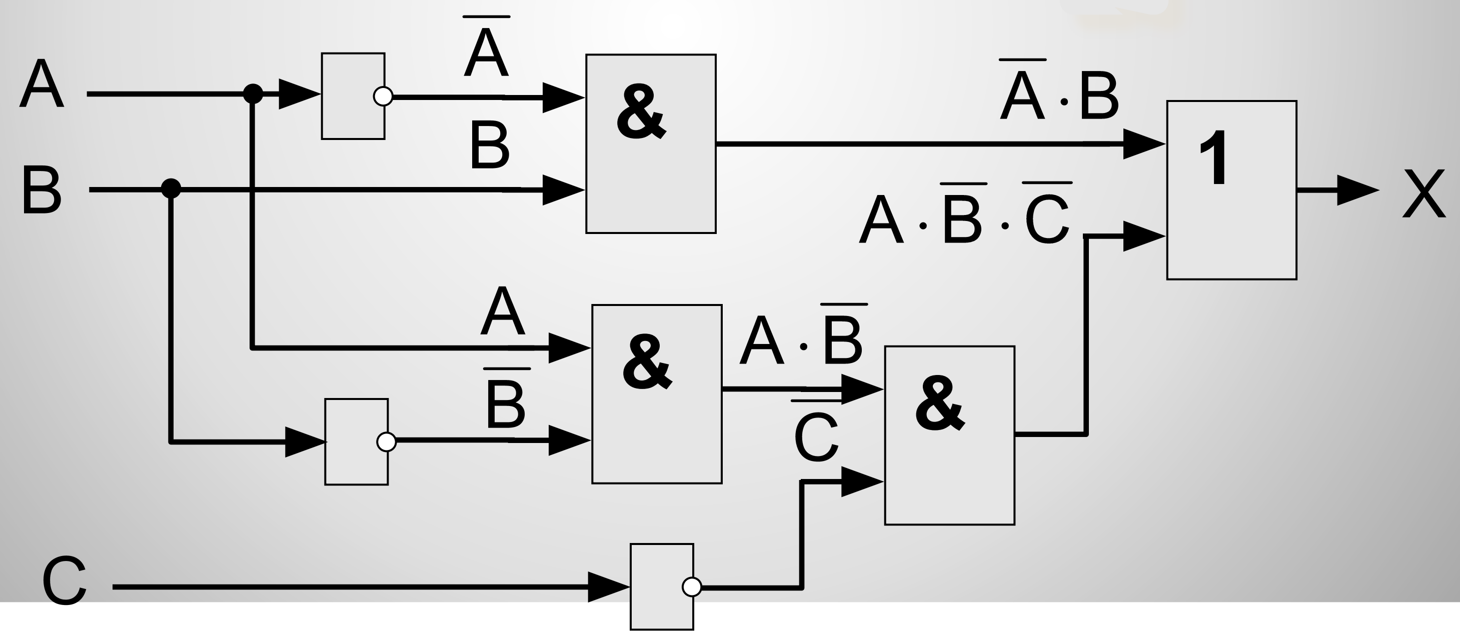


Составление схем

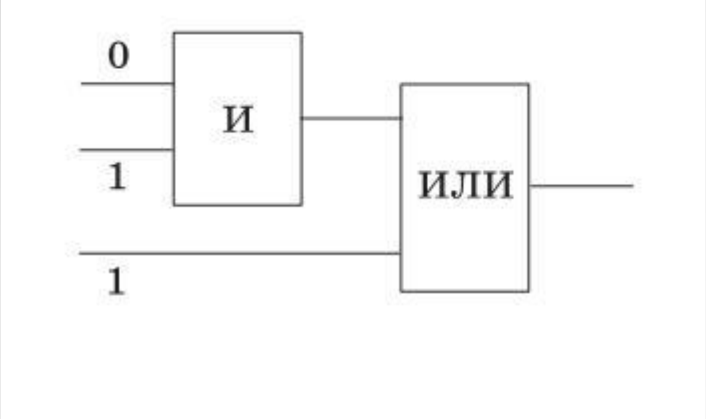
последняя операция - ИЛИ

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

И



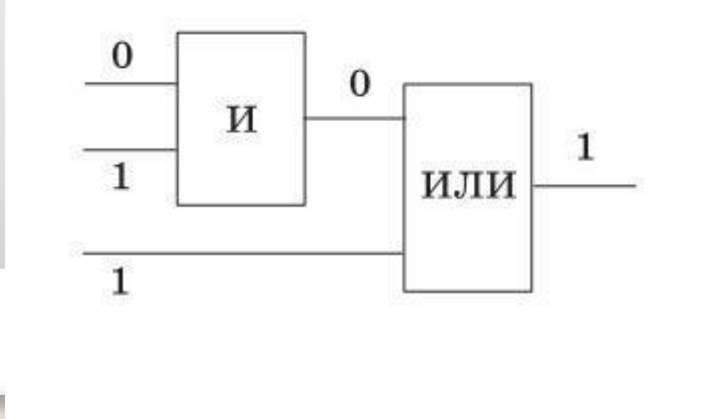
Пример 1. *Нарисовать схему для логического выражения:* **1 ИЛИ 0 и 1.**



Читать эту схему надо слева направо. Первой выполняется операция И (что наглядно видно на схеме), затем ИЛИ.

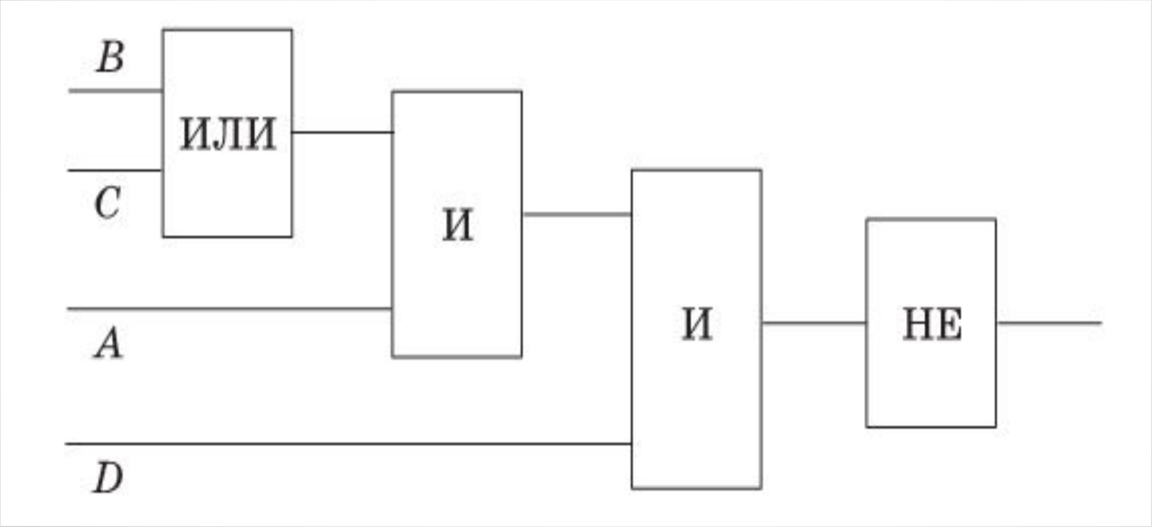
Теперь в порядке слева направо припишем к выходящим линиям результаты операций:

В результате получилась 1, т.е. "истина".



Пример 2. *Представить в виде логической схемы логическую формулу:* **НЕ (А И (В ИЛИ С) И D)**

Логическая схема будет выглядеть так:



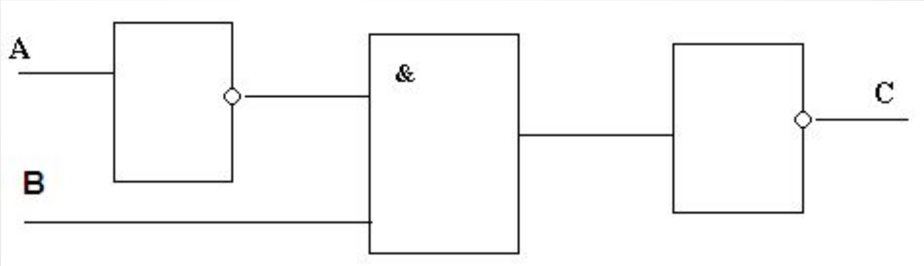
Теперь с помощью схемы рассчитаем значение формулы при A=C=D=1, B=0

В результате получится логический ноль, т.е. "ложно".

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.



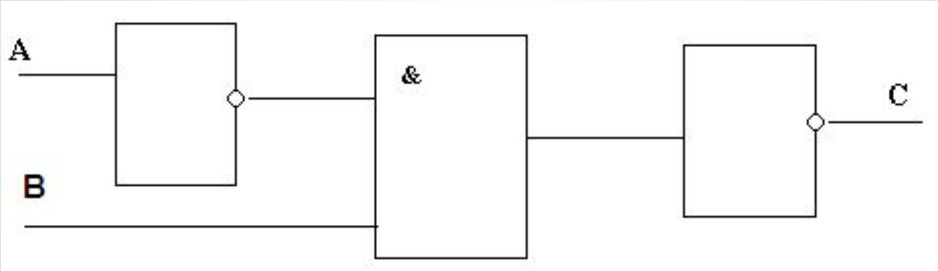
Ясно, что элемент "И" осуществляет логическое умножение значений ¬A и B. Над результатом в элементе "НЕ" осуществляется операция отрицания, т.е. вычисляется значение выражения: $\overline{\bar{A} \,\&\, B}$

Таким образом структурной формулой данной функциональной схемы является формула:

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

Таблица истинности функциональной схемы

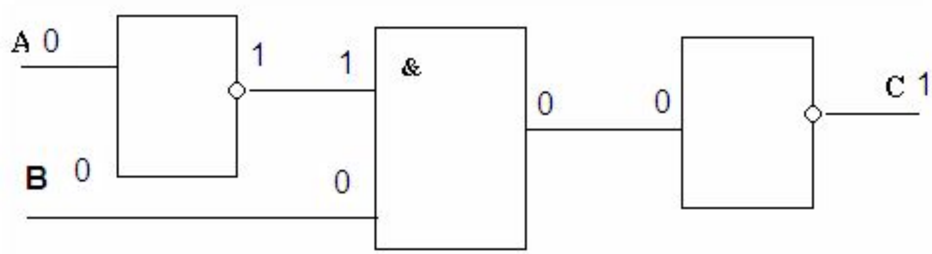
Для функциональной схемы можно составить таблицу истинности, то есть таблицу значений сигналов на входах и выходах схемы, по которой можно понять какую функцию выполняет данная схема. ***<u>Таблица истинности -</u>*** это табличное представление логической (функциональной) схемы в котором перечислены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



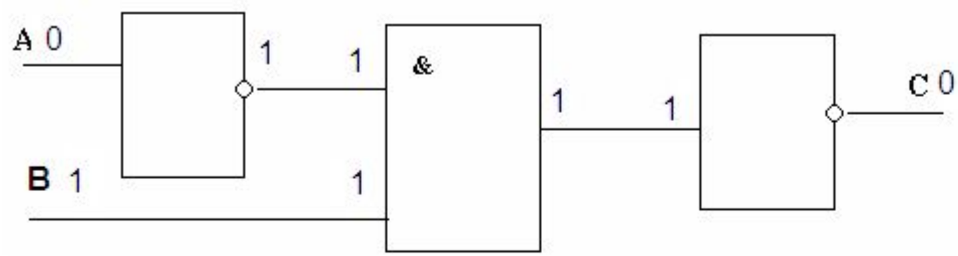
Начертим таблицу: количество столбцов = количество входов + количество выходов, количество строк = 2 $^{\text{количество входов}}$. В данной таблице 3 столбца и 4 строки. Заполним первые столбцы всеми возможными вариантами входных сигналов

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

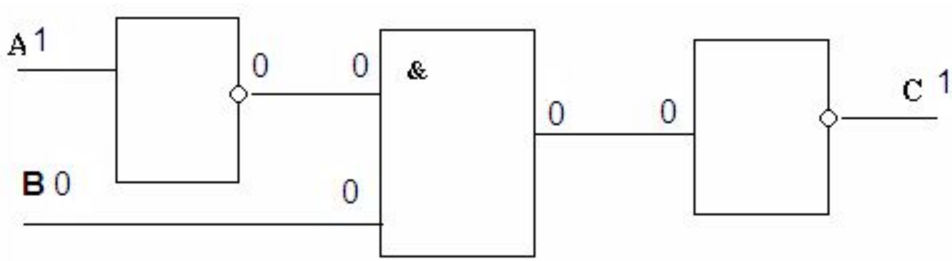
Рассмотрим первый вариант входных сигналов: A=0, B=0. Проследим по схеме, как проходят и преобразуются входные сигналы. Результат, полученный на выходе (C=1), запишем в таблицу.



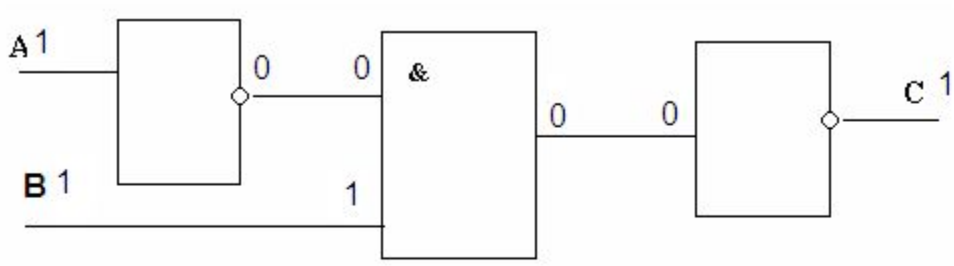
Рассмотрим второй вариант входных сигналов: A=0, B=1. Проследим по схеме, как проходят и преобразуются входные сигналы. Результат, полученный на выходе (C=0), запишем в таблицу.



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.



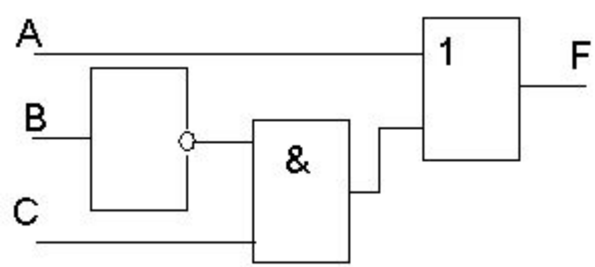
Рассмотрим четвёртый вариант входных сигналов: A=1, B=1. Проследим по схеме, как проходят и преобразуются входные сигналы. Результат, полученный на выходе (C=1), запишем в таблицу.



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:



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.

Логическая реализация типовых устройств компьютера

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:

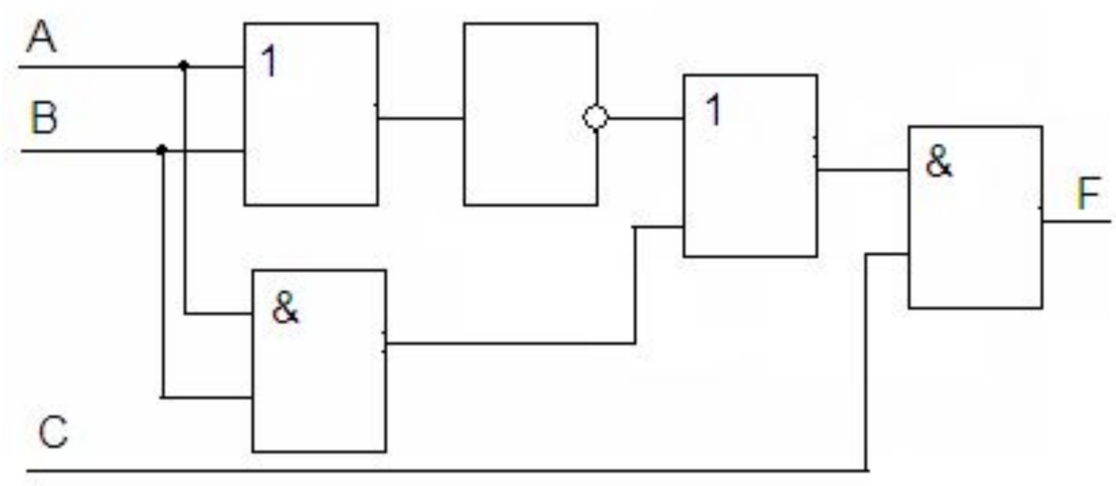
$$F = \overline{A} \,\&\, \overline{B} \,\&\, C \vee A \,\&\, B \,\&\, C$$

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

Упростим полученное логическое выражение:

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

Построим логическую схему для данного выражения:



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:

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

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

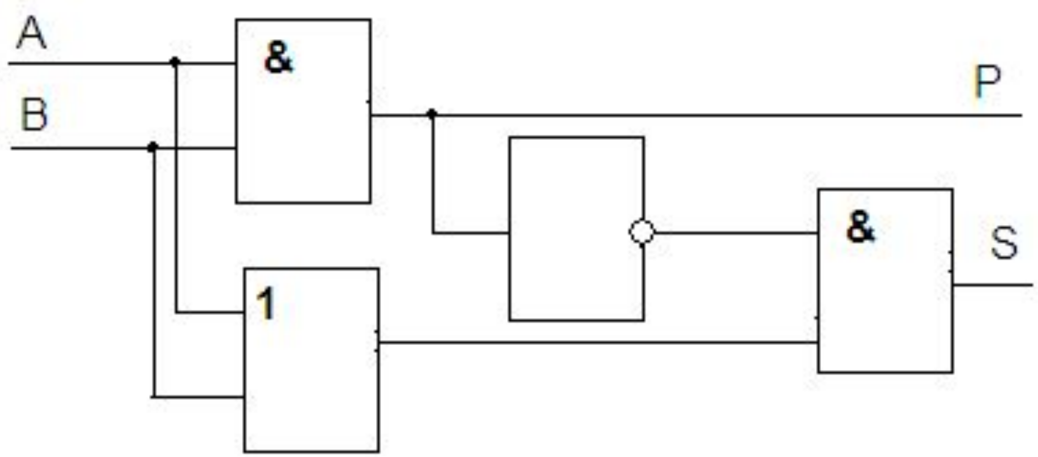
Преобразуем вторую формулу, пользуясь законами логики:

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

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

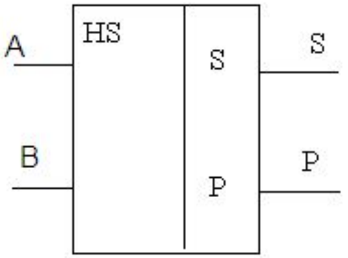
3. Теперь можно построить функциональную схему одноразрядного полусумматора:

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



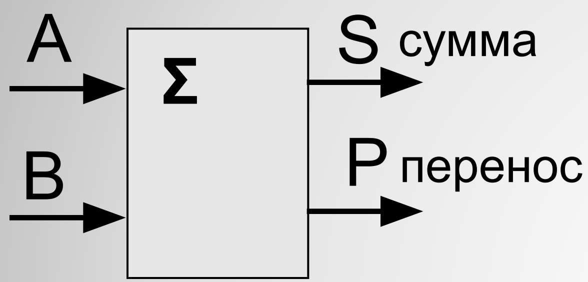
Чтобы убедиться в том, как работает схема, проследите за прохождением сигналов в каждом из четырёх случаев и составьте таблицу истинности данной логической схемы.

Условное обозначение одноразрядного сумматора:



Полусумматор

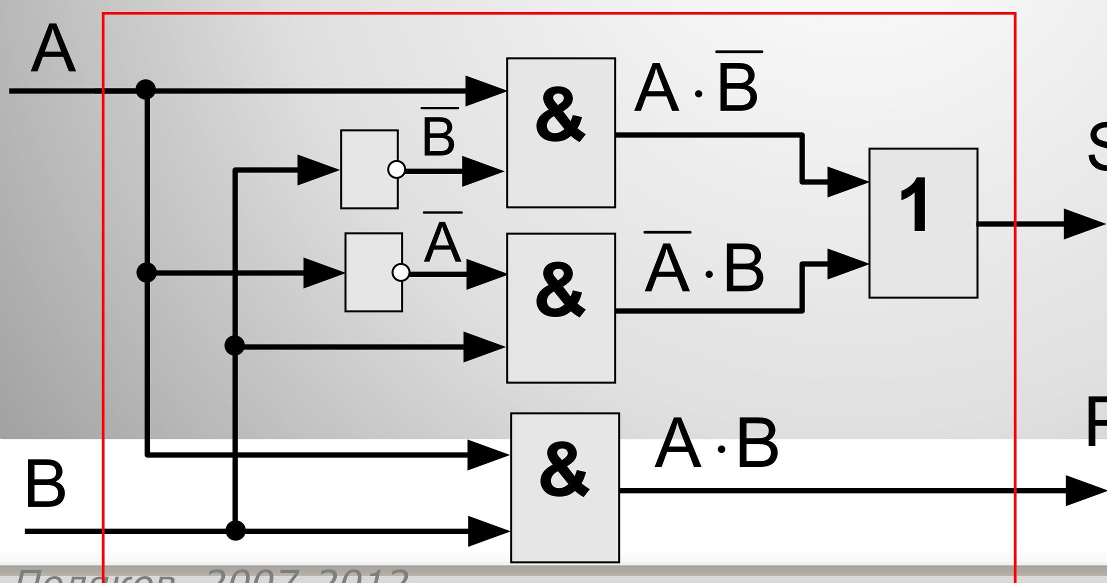
Полусумматор – это логическая схема, способная складывать два одноразрядных двоичных числа.



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?**

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 **А, В** - суммируемые двоичные цифры , **Ро** - перенос из младшего разряда, ***S*** - образующаяся сумма данного разряда и осуществляет перенос **Р** в следующий старший разряд.

Слагаемые		**Перенос из младшего разряда**	**Сумма**	**Перенос**
A	**B**	**P_0**	**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

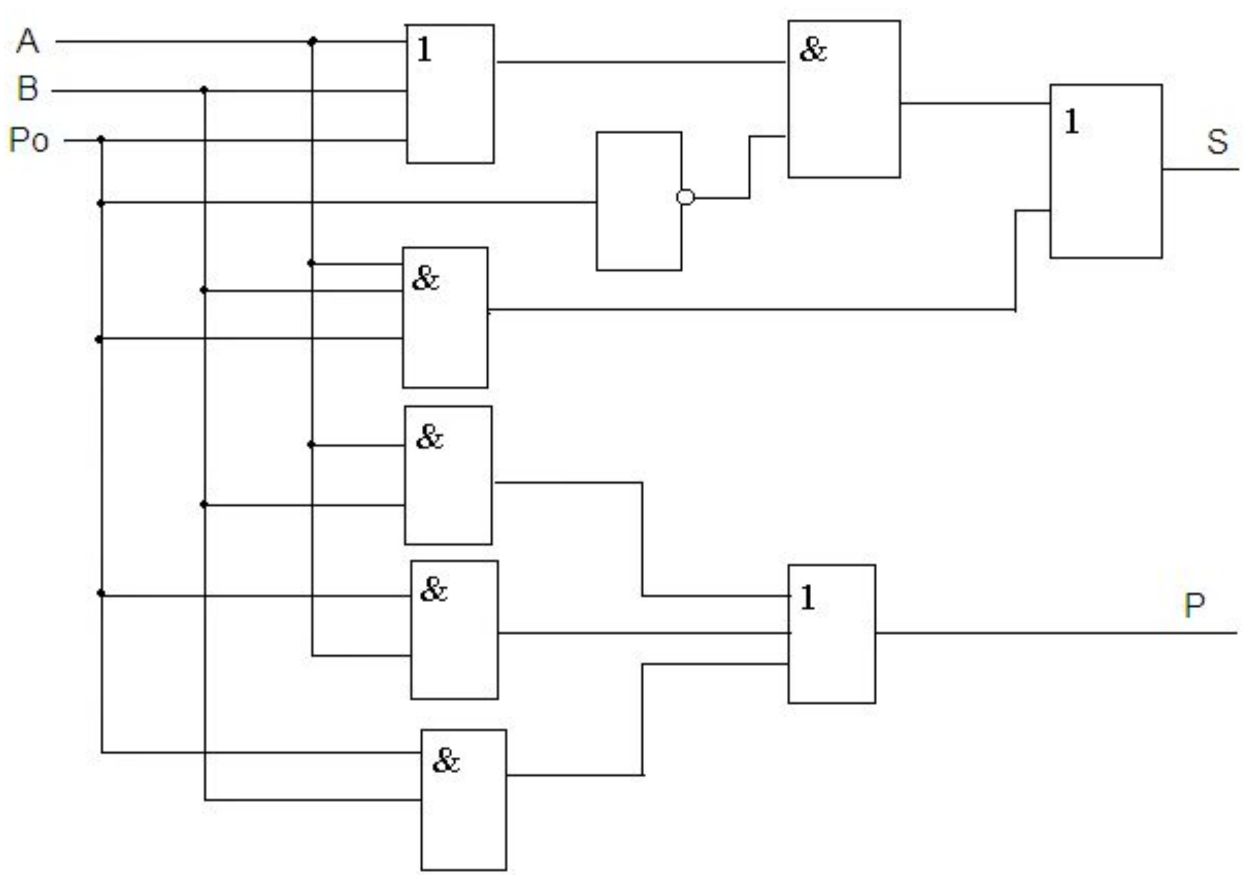
Формула переноса: $P = A \& B \& \overline{P}_0 \vee \overline{A} \& B \& P_0 \vee A \& \overline{B} \& P_0 \vee A \& B \& P_0$

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

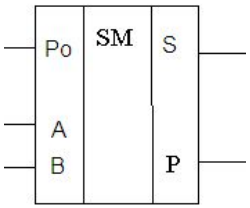
После преобразования формулы переноса и суммы принимают вид:

$$P = A \& B \vee A \& P_0 \vee B \& P_0 \qquad S = (A \vee B \vee P_0) \& \overline{P_0} \vee (A \& B \& P_0)$$

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

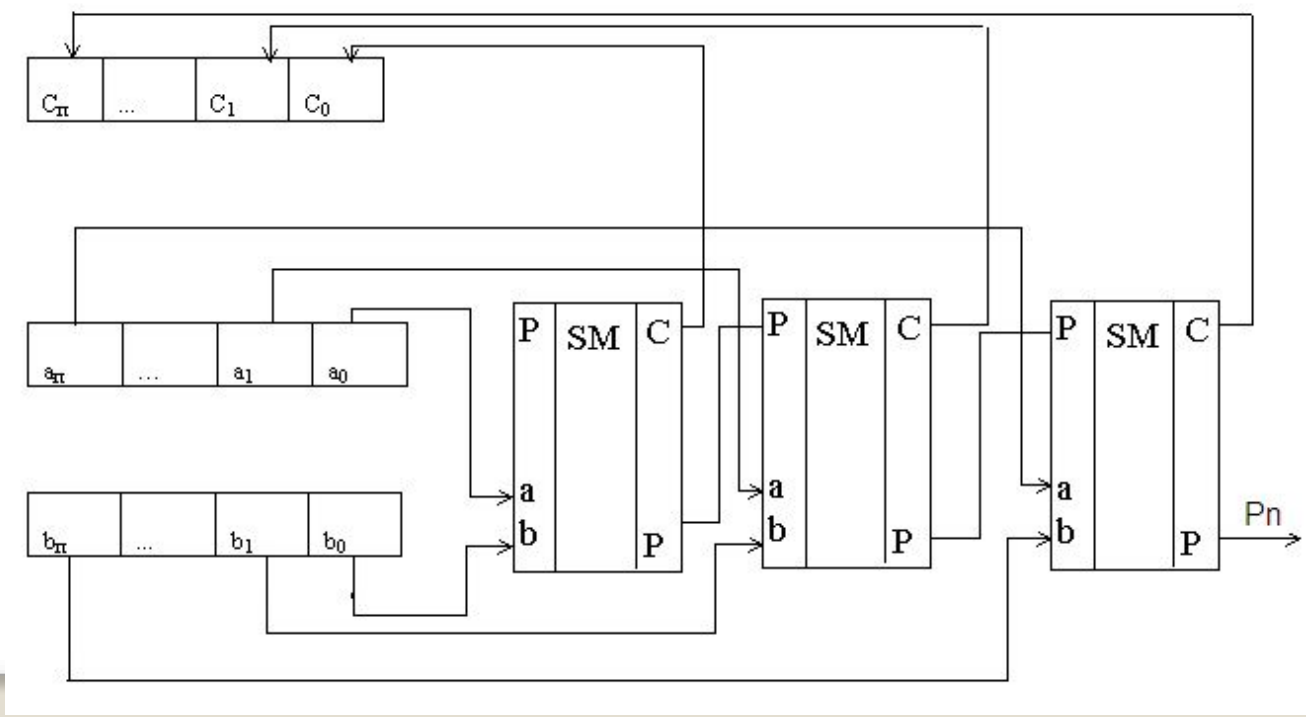


Сумматор - это электронная логическая схема, выполняющая суммирование двоичных чисел поразрядным сложением. Сумматор является центральным узлом арифметико-логического устройства процессора. Находит он применение и в других устройствах компьютера. В реальных электронных схемах сумматор изображается так:



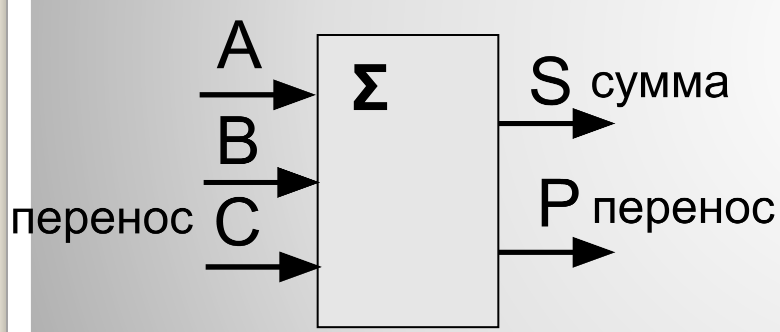
Сумматор выполняет сложение *многозначных двоичных чисел*. Он представляет собой последовательное соединение *одноразрядных двоичных сумматоров*, каждый из которых осуществляет сложение в одном разряде. Если при этом возникает переполнение разряда, то перенос суммируется с содержимым старшего соседнего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

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



Сумматор

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



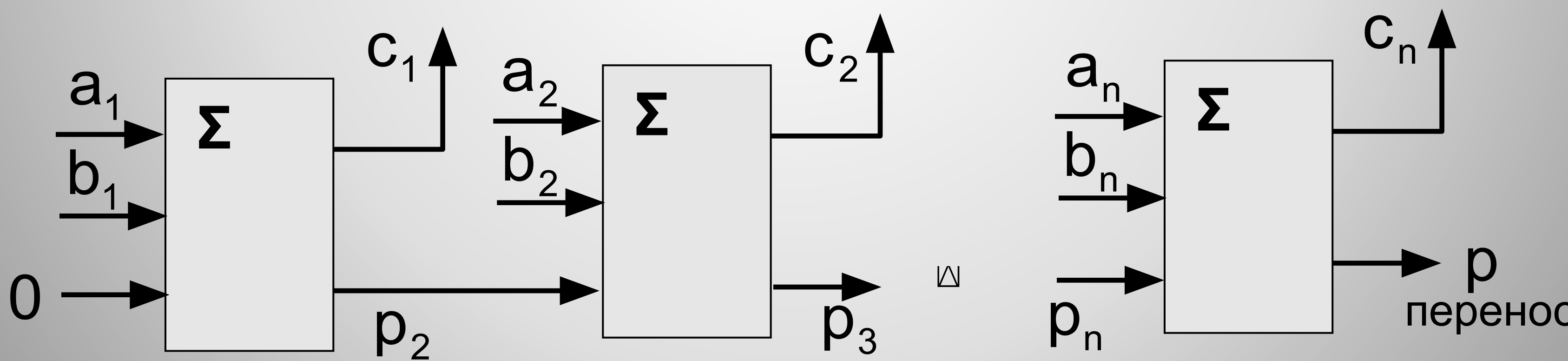
A	B	C	P	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

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



ТРИГГЕР

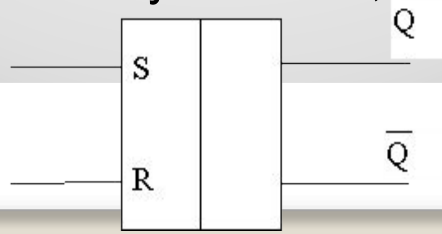
Триггер - электронная схема, применяемая для хранения значения одноразрядного двоичного кода.

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:

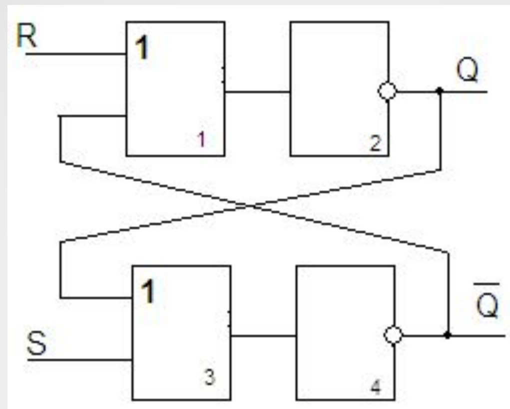


S(set) - установка триггера

R(reset) - сброс

RS-триггер

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



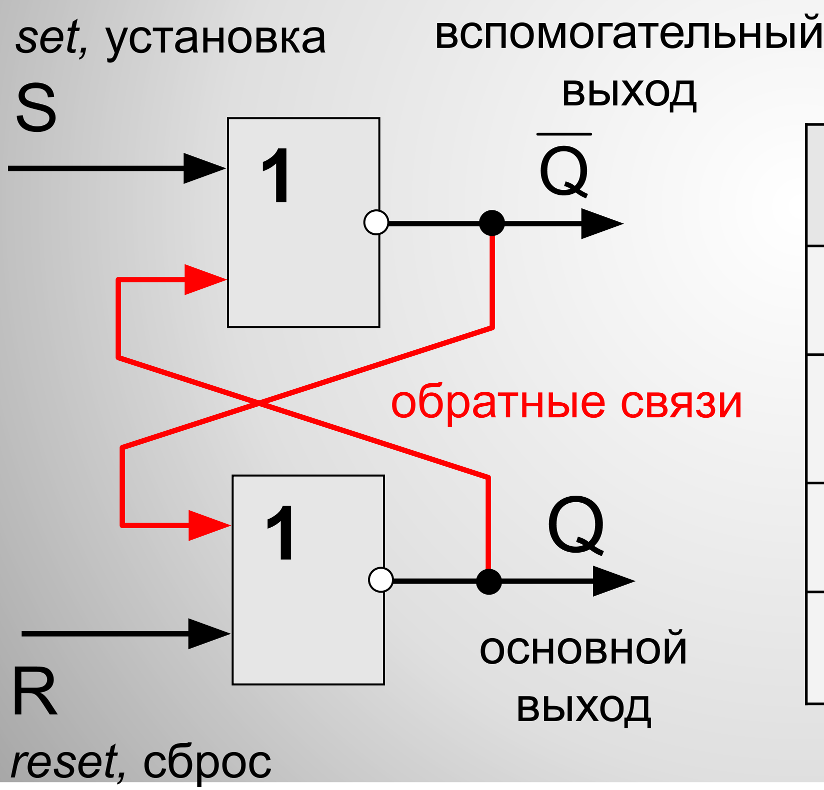
Как он работает?

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, $\overline{Q}$= 1.

Убедимся, что данное устройство сохраняет информацию. Запомните, что S=0, R=1, Q=0, $\overline{Q}$=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 $\overline{Q}$ =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, $\overline{Q}$ =0.

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