



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.

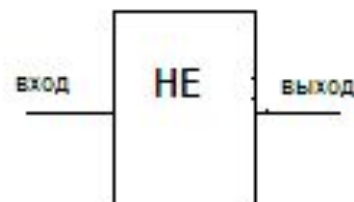
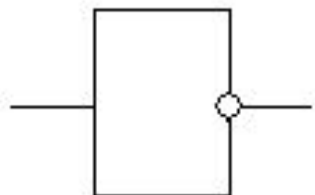
Логические элементы имеют один или несколько входов, на которые подаются электрические сигналы, обозначаемые условно **0**, если отсутствует электрический сигнал, и **1**, если имеется электрический сигнал.

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

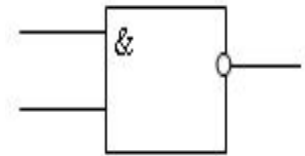
Другие логические элементы построены из трех простейших базовых элементов и выполняют более сложные логические преобразования информации.

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:

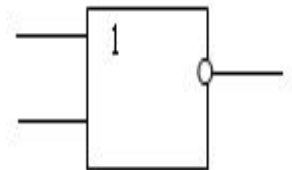
<i>вход 1</i>	<i>вход 2</i>	<i>выход</i>
0	0	1
0	1	1
1	0	1
1	1	0



Логический элемент ИЛИ-НЕ

Логический элемент ИЛИ-НЕ выполняет логическую функцию стрелка Пирса (ИЛИ-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

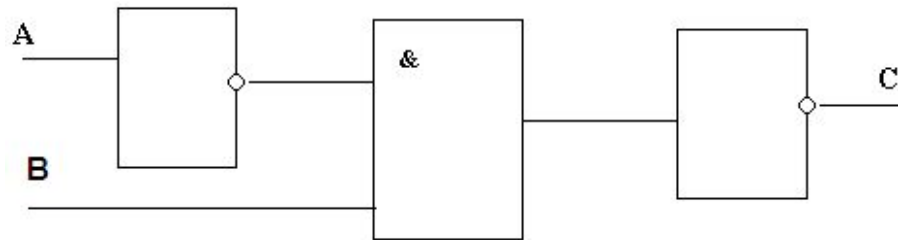


Функциональные схемы

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:

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

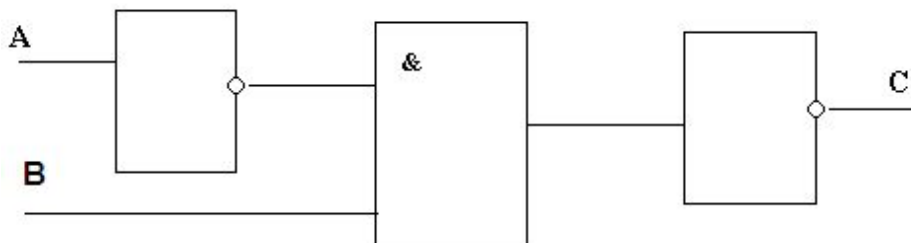
Таким образом структурной формулой данной функциональной схемы является формула:

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

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:

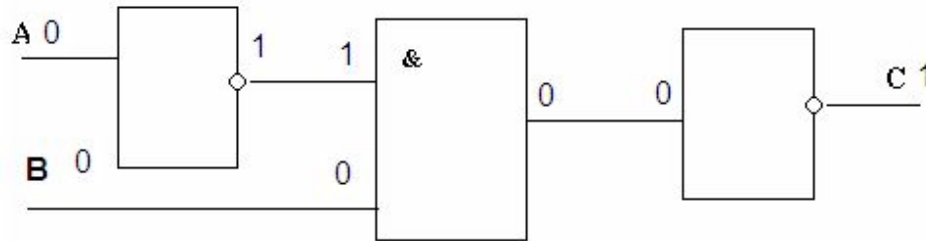


Начертим таблицу: количество столбцов = количество входов + количество выходов, количество строк = $2^{\text{количество входов}}$. В данной таблице 3 столбца и 4 строки.

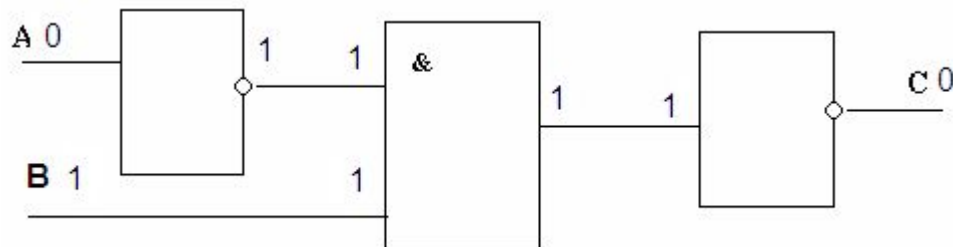
Заполним первые столбцы всеми возможными вариантами входных сигналов

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

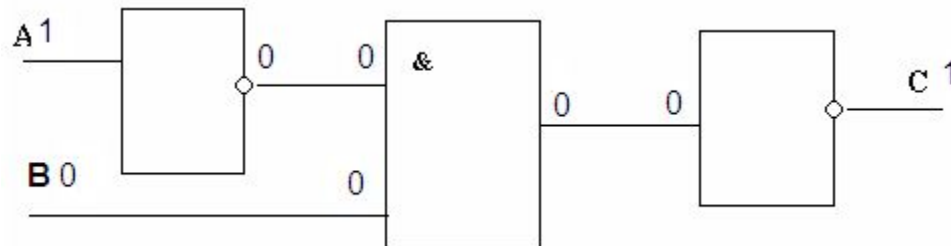
Рассмотрим первый вариант входных сигналов: $A=0$, $B=0$. Проследим по схеме, как проходят и преобразуются входные сигналы. Результат, полученный на выходе ($C=1$), запишем в таблицу.



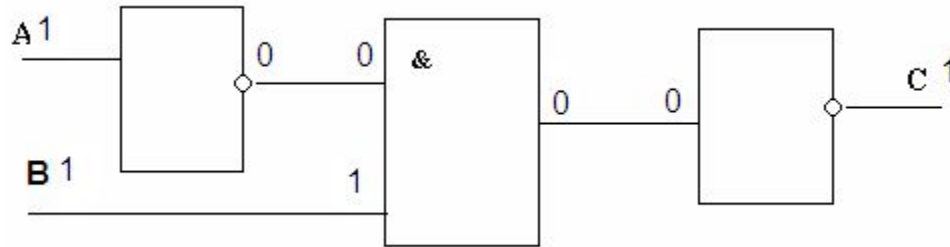
Рассмотрим второй вариант входных сигналов: $A=0$, $B=1$. Проследим по схеме, как проходят и преобразуются входные сигналы. Результат, полученный на выходе ($C=0$), запишем в таблицу.



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.



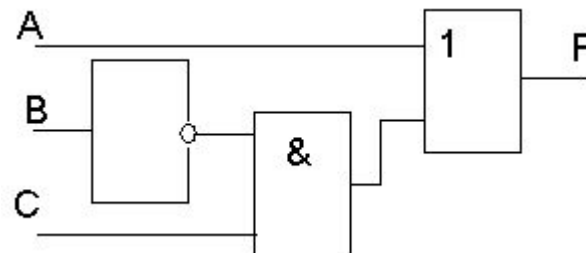
Рассмотрим четвёртый вариант входных сигналов: A=1, B=1. Проследим по схеме, как проходят и преобразуются входные сигналы. Результат, полученный на выходе (C=1), запишем в таблицу.



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, её 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:

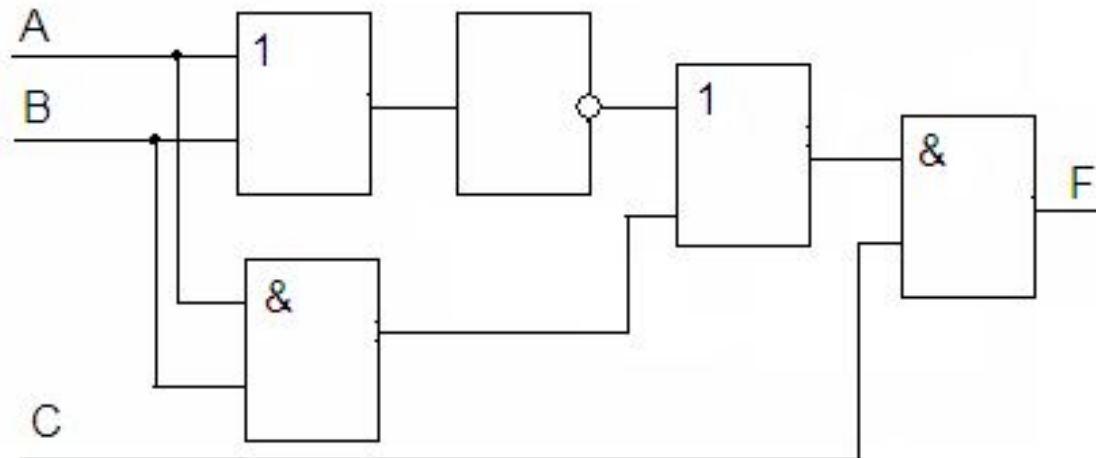
$$F = \bar{A} \& \bar{B} \& C \vee A \& B \& C$$

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:

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

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

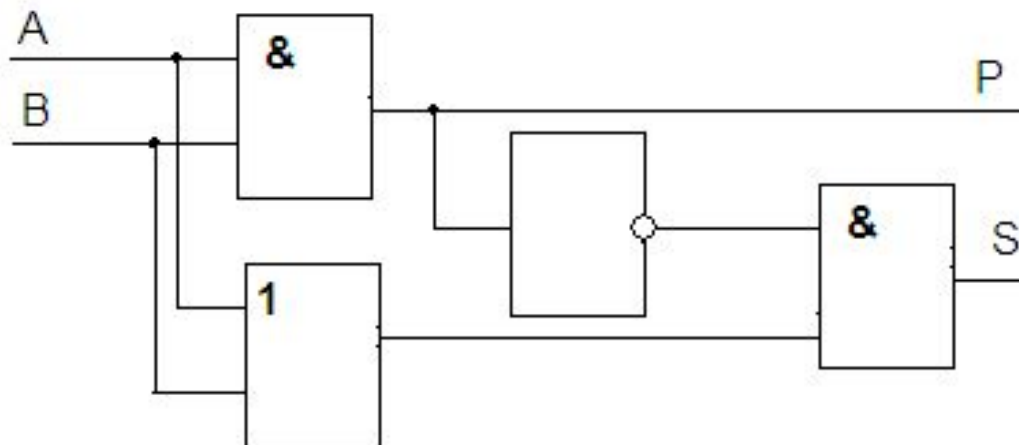
Преобразуем вторую формулу, пользуясь законами логики:

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

3. Теперь можно построить функциональную схему одnorазрядного полусумматора:

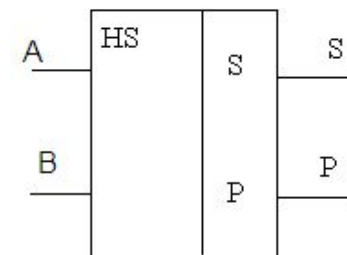
$$P(A, B) = A \& B$$

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



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, **P₀** - 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 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: $P = A \& B \& \bar{P}_0 \vee \bar{A} \& B \& P_0 \vee A \& \bar{B} \& P_0 \vee A \& B \& P_0$

Формула для вычисления суммы:

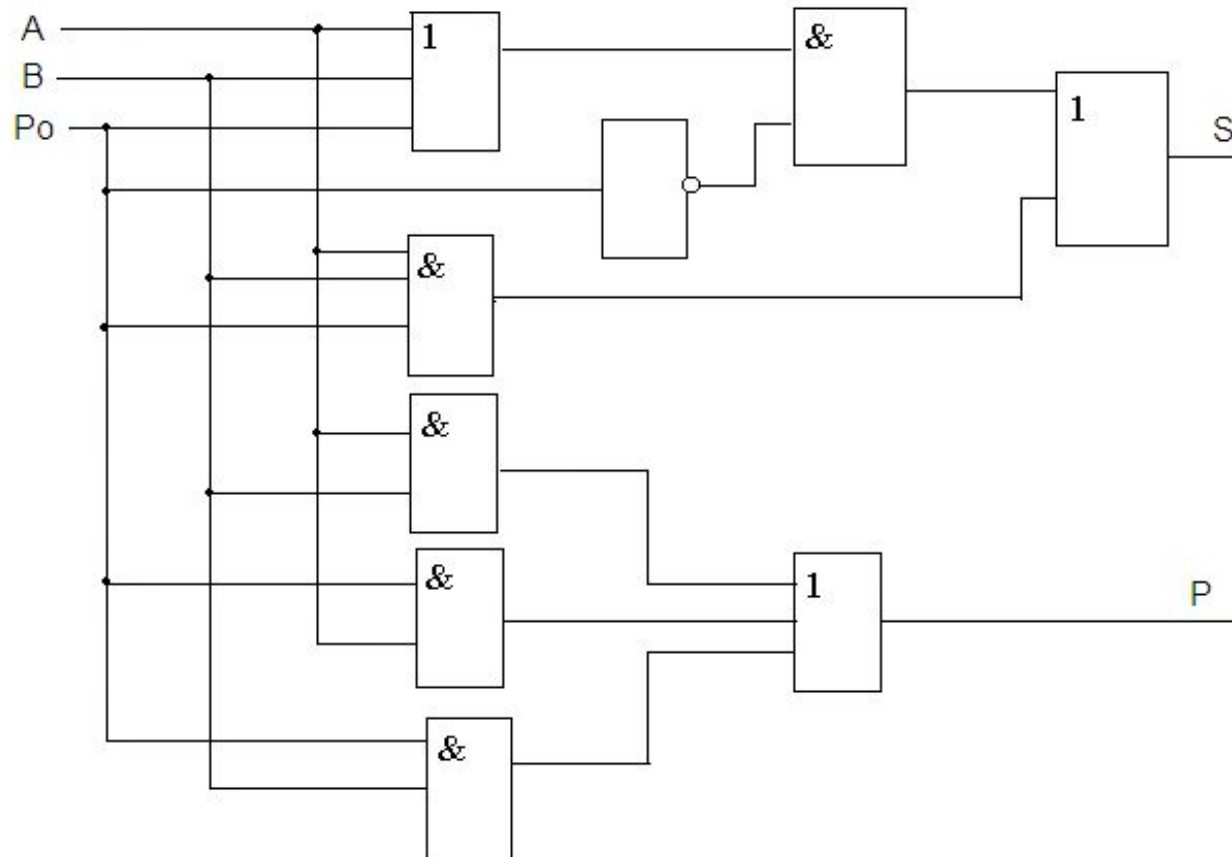
$$S = \bar{A} \& B \& \bar{P}_0 \vee A \& \bar{B} \& \bar{P}_0 \vee \bar{A} \& \bar{B} \& P_0 \vee A \& B \& P_0$$

После преобразования формулы переноса и суммы принимают вид:

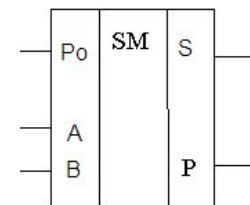
$$P = A \& B \vee A \& P_0 \vee B \& P_0$$

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

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

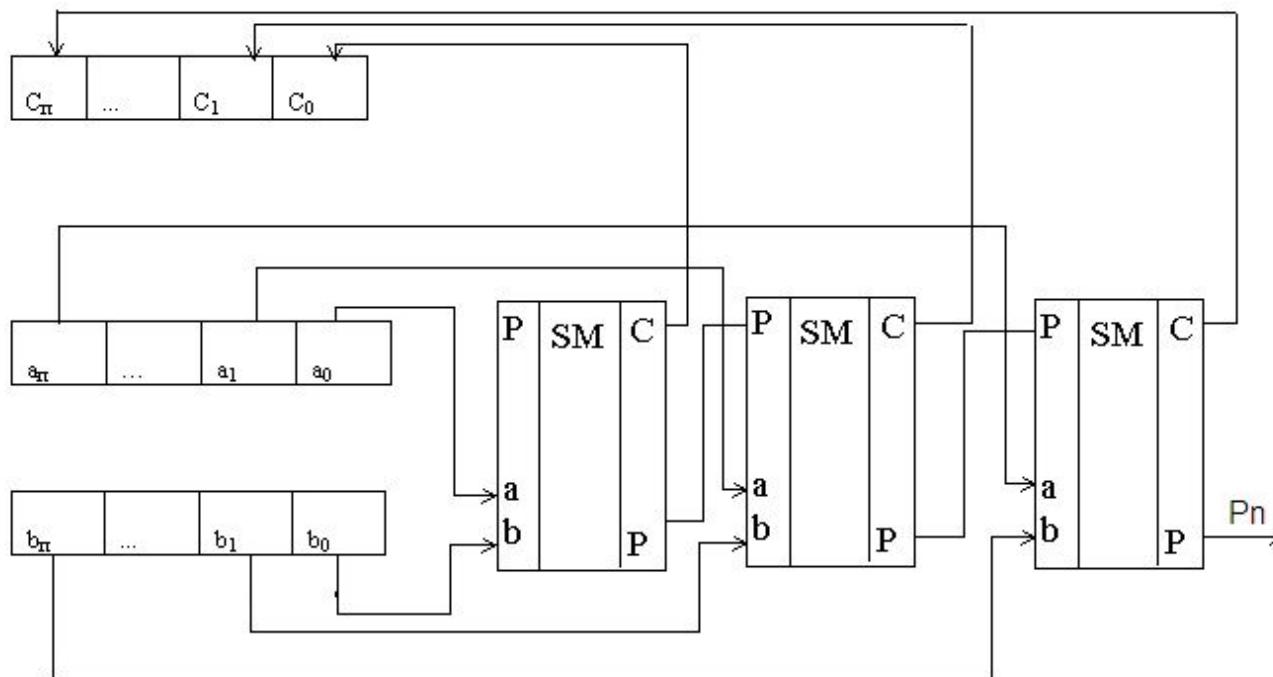


Сумматор - это электронная логическая схема, выполняющая суммирование двоичных чисел поразрядным сложением. Сумматор является центральным узлом арифметико-логического устройства процессора. Находит он применение и в других устройствах компьютера. В реальных электронных схемах сумматор изображается так:



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

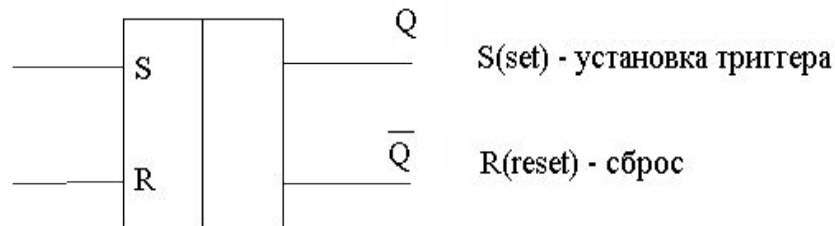
Триггер - электронная схема, применяемая для хранения значения одноразрядного двоичного кода.

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 сколько 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.

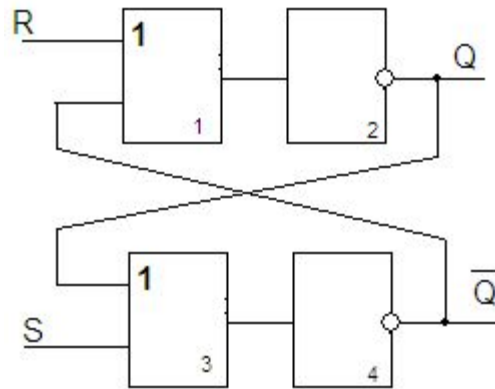
Существуют разные варианты исполнения триггеров в зависимости от элементной базы (И-НЕ, ИЛИ-НЕ) и функциональных связей между сигналами на входах и выходах (*RS*, *JK*, *T*, *D* и другие).

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 ($Q=0$).

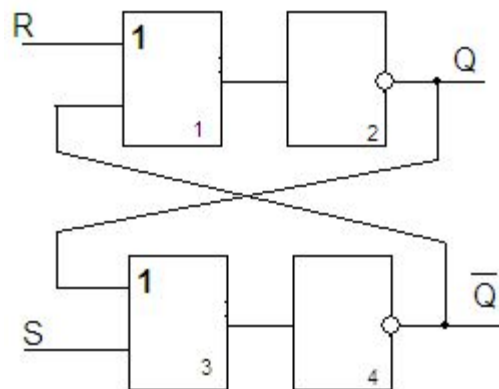
Следовательно, и на втором входе элемента №3 установится сигнал 0. На выходе элемента №3 - 0. Пройдя через элемент №4 сигнал изменится на 1. Следовательно, $\bar{Q}=1$.

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*.

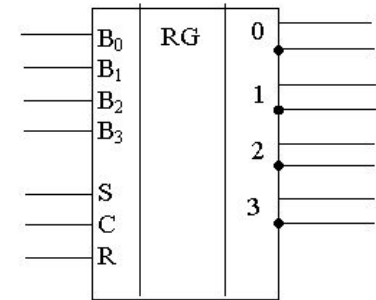
Упрощенно регистр можно представить как совокупность ячеек, в каждой из которых может быть записано одно из двух значений: 0 или 1, то есть один разряд двоичного числа.

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.