

Логика

Уроки

Логика – наука о законах и формах мышления.

Слово «логика» произошло от древнегреческого logos, что означает «слово, мысль, понятие, рассуждение, закон».

Основные формы мышления: понятие, высказывание, умозаключение.

Понятие – это форма мышления, которая выделяет существенные признаки предмета или класса предметов, позволяющие отличать их от других («Вот стул на нем сидят, вот стол – за ним едят.»).

Высказывание – это повествовательное предложение, о котором можно сказать, истинно оно или ложно.

Земля – планета солнечной системы. $3+5=19$

Умозаключение – это форма мышления, с помощью которой из одного или несколько суждений может быть получено новое суждение (вывод). (Доказательство теорем в алгебре).

Английский математик **Джордж Буль** (1815-1864) изобрел своеобразную алгебру – систему обозначений и правил, применимую к всевозможным объектам, в частности к высказываниям. Эта алгебра называется его именем – алгебра Буля или булева алгебра.

Логическая переменная – это простое высказывание (содержит одну мысль), значение которой 0 (ложь, false) или 1 (истина, true), обозначается латинскими буквами.

Логическая функция – составное высказывание, полученное из простых высказываний, соединенных между собой с помощью логических операций.

Логические операции.

Логические операции над высказываниями выполняются на основе законов и правил булевой алгебры.

Логическое умножение КОНЪЮНКЦИЯ (\wedge , &) определяет соединение логических выражений с помощью союза И.

Например: 1) А – “ $X > 10$ ”; В – “ $X < 30$ ”; А и В - “ $10 < X < 30$ ”.

2) А – “Компьютер предназначен для обработки информации”

В – “Компьютер предназначен для хранения информации”

А и В – “Компьютер предназначен для обработки и хранения информации”

3) Иванов посещает факультатив по математике и по физике.

Сложное высказывание **А и В** принимает значение - истина, когда оба высказывания принимают значение - истина.

Логическое сложение ДИЗЪЮНКЦИЯ (\vee) определяет соединение логических выражений с помощью союза ИЛИ.

Например: В библиотеке есть книги Беляева А.Р. или книги других фантастов. $X \geq 0$.

Иванов участвует в олимпиаде по математике или по физике.

Сложное высказывание **А или В** принимает значение - истина, когда хотя бы одно из высказываний принимает значение - истина.

Отрицание ИНВЕРСИЯ - НЕ (\neg) изменяет значение логического высказывания на противоположное.

Например: 1) А – “а – простое число”; не А – “неверно, что а – простое число.»

Логические операции.

Импликация (логическое следование) (\rightarrow) соответствует связке «если ..., то ...»

Если А, то В, В необходимо для А, А достаточно для В, В тогда, когда А

Импликация истинна всегда, за исключением случая, когда А истинно, а В ложно.

Пример: Если идет дождь, то земля сухая. (ложь)

А - “х делится на 9”

В - “х делится на 3”

А \rightarrow В - “если число х делится на 9, то оно делится на 3” (истина)

Эквивалентность(равнозначность) ($\Leftrightarrow, =$) соответствует связке «тогда и только

тогда, когда..». А тогда и только тогда, когда В

Результат эквивалентности будет истинным тогда и только тогда, когда оба исходных выражения одновременно истинны или ложны.

Исключающая дизъюнкция «либо..., либо» \oplus , (неэквивалентность) А \oplus В.

Результат будет истинным, когда одно из исходных выражений истинно, а другое ложно.

Пример: Пан или пропал.

Построение таблиц истинности.

Таблица истинности – таблица, определяющая значение сложного высказывания при всех возможных значениях простых высказываниях.

Для составления таблицы надо:

1. определить число переменных;
2. выяснить количество строк в таблице (вычисляется как 2^n , где n – количество переменных);
3. выяснить количество столбцов = количество переменных + количество логических операций;
4. установить последовательность выполнения логических операций;
5. записать логические операции в таблицу истинности и определить для каждого значение;

Порядок выполнения логических операций:

1. операция в скобках;
2. отрицание;
3. логическое умножение;
4. логическое сложение;
5. импликация;
6. эквивалентность.

Составьте по булевому выражению $F=(A \vee B) \wedge \neg C$ таблицу истинности:

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

Самостоятельно: 1) $F=A \vee (\overline{B \wedge C})$; 2) $F=A \wedge (\overline{B \vee C})$; 3) $F=\overline{(A \wedge B)} \vee C$.

Законы алгебры логики.

Если логическое выражение содержит большое число операций, то составлять для него таблицу истинности достаточно сложно. В таких случаях формулы надо привести к нормальной форме. Формула имеет нормальную форму, если в ней отсутствуют знаки эквивалентности, импликации, двойного отрицания, знаки отрицания находятся только при логических переменных. Для приведения формулы к нормальной форме используют законы логики и правила преобразований.

Тождества.

$A + 0 = A$	$A \& 0 = 0$	
$A + 1 = 1$	$A \& 1 = A$	$=$ $A = A$ двойное отрицание
$A + A = A$	$A \& A = A$	
$A + \bar{A} = 1$	$A \& \bar{A} = 0$	

Законы алгебры логики.

Переместительный закон

$$A + B = B + A \qquad A \& B = B \& A$$

Сочетательный закон

$$(A + B) + C = A + (B + C) \qquad (A \& B) \& C = A \& (B \& C)$$

Распределительный закон

$(A + B) \& C = A \& C + B \& C$ дистрибутивность конъюнкции относительно дизъюнкции – аналог раскрытия скобок в алгебре

$A \& (B + C) = (A \& B) + (A \& C)$ дистрибутивность дизъюнкции относительно конъюнкции – «чудо» логики, в алгебре аналога нет.

Закон де Моргана (закон отрицания)

$$\overline{A + B} = \bar{A} \& \bar{B} \qquad \overline{A \& B} = \bar{A} + \bar{B}$$

Доказать на уроке, используя таблицы истинности.

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

$$A \leftrightarrow B = A \& B + \bar{A} \& \bar{B} = (\bar{A} + B) \& (A + \bar{B})$$

Задания:

1) Записать формулу, упростить:

1. A или (не A и B)
2. A и (не A или B)
3. (A или B) и (не B или A) и (не C или B)
4. (1 или (A или B)) или ((A или C) и 1)

Решение:

$$A + (\overline{A} \& B) = (A + \overline{A}) \& (A + B) = 1 \& (A + B) = A + B;$$

$$A \& (\overline{A} + B) = A \& \overline{A} + A \& B = 0 + A \& B = A \& B;$$

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

$$(A \& \overline{B} + A + 0 + B \& A) \& (\overline{C} + B) = A \& (\overline{B} + 1 + B) \& (\overline{C} + B) = A \& (\overline{C} + B);$$

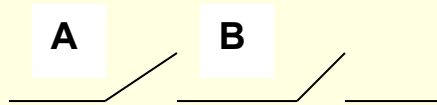
$$(1 + (A + B)) + ((A + C) \& 1) = 1 + (A + C) = 1.$$

2) Доказать, используя таблицы истинности, второй распределительный закон
 $A \& B + C = (A + C) \& (B + C)$.

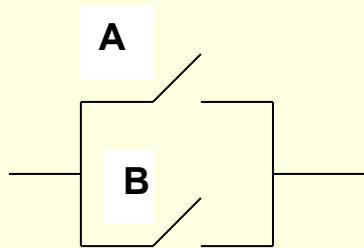
Логические схемы.

Над возможностями применения логики в технике ученые и инженеры задумывались давно. Американский логик Чарльз Сандерс Пирс первым осознал, что бинарная логика имеет сходства с работой электрических переключательных (контактных) схем. Компьютер работает на электричестве, любая информация представлена в компьютере в виде электрических импульсов. Рассмотрим реализацию логических элементов через электрические контактные схемы. Контакты обозначены латинскими буквами.

Последовательное соединение контактов. Цепь с последовательным соединением соответствует логической операции И (конъюнкции).



Параллельное соединение контактов. Цепь с параллельным соединением соответствует логической операции ИЛИ (дизъюнкции).

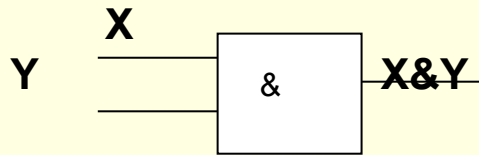


Логическая операция НЕ (инверсия) реализуется через контактную схему электромагнитного реле.

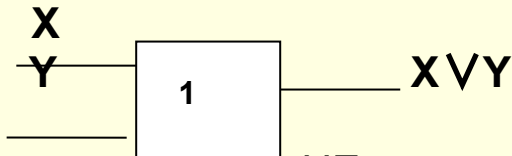
&

Логические схемы – схемы, выполняющие операции преобразования: запоминания, пересылки двоичных битов информации в компьютере
Логический элемент (вентиль) – часть электронной логической схемы, которая выполняет элементарную логическую операцию. К элементарным логическим операциям реализуемым на логических микросхемах относятся операции: и, или, не, и-или, и-не, или-не, и-или-не и др.

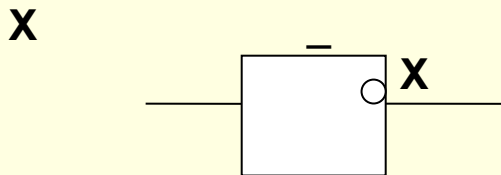
Логический элемент И - конъюнктор.&



Логический элемент ИЛИ – дизъюнктор.



Логический элемент НЕ – инвертор.



Задания:

Упростить логическое выражение.

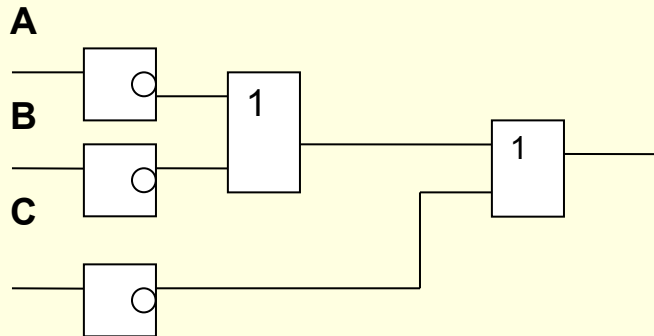
Построить таблицу истинности.

Построить логическую схему.

Показать. Что схема работает в соответствии с таблицей истинности.

Выражение: $F = \overline{A \& B} + \overline{B \& C}$

Ответ: $F = \overline{A \& B} + \overline{B \& C} = \overline{A} + \overline{B} + \overline{B} + \overline{C} = \overline{A} + \overline{B} + \overline{C}$.



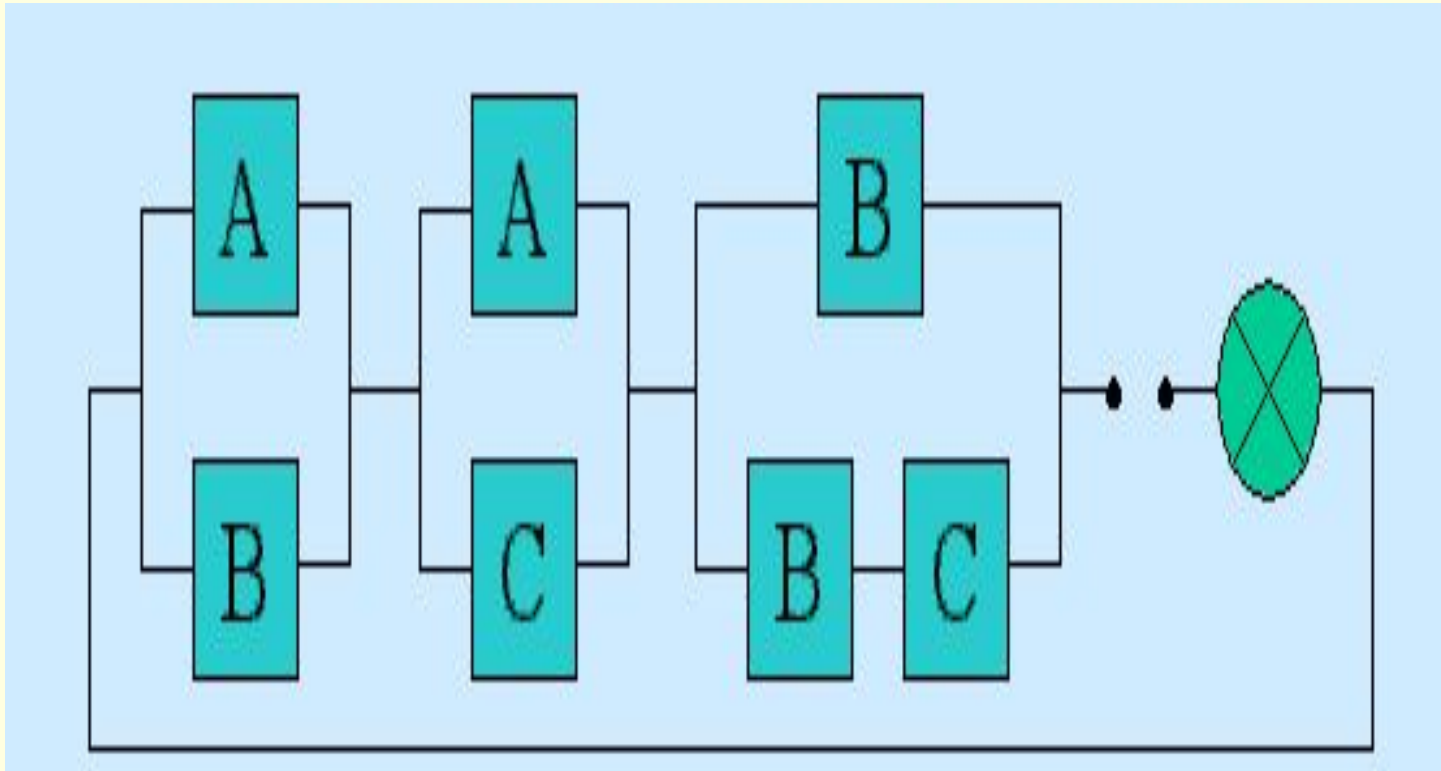
Задание для самостоятельной работы:

$$1) F = \overline{X \& Y} + \overline{Y \& Z} + \overline{Z \& X} = \overline{Y \& (X + Z)} + \overline{Z \& X} = \overline{Y} + \overline{X + Z} + \overline{Z \& X} =$$

$$\overline{Y} + \overline{X} \& \overline{Z} + \overline{Z \& X} = \overline{Y} + \overline{X} \& (\overline{Z} + Z) = \overline{Y} + \overline{X}$$

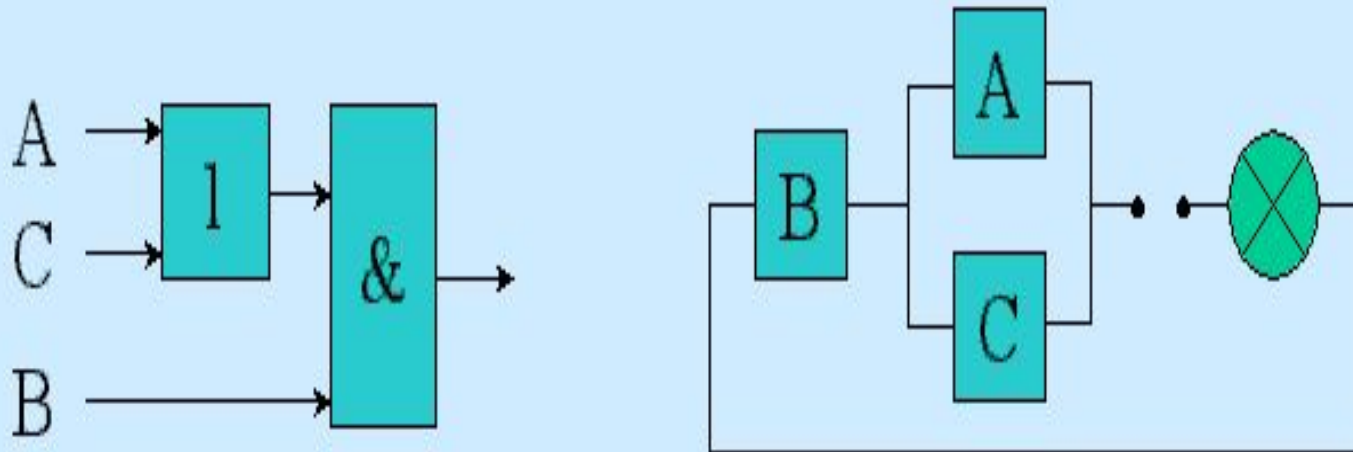
$$2) F = A \& B + D + \neg(A + \neg B) = A \& B + D + \neg A \& B = B \& (A + \neg A) + D = B + D.$$

Упростить электрическую схему и построить электрическую и логическую схемы результата.



Решение

$$P = (A+B)(A+C)(B+BC) = B(A+C)$$



Алгебра логики дала в руки конструктора мощное средство разработки, анализа и совершенствования логических схем. Гораздо проще, быстрее и дешевле изучать свойства и доказывать правильность работы схемы с помощью выражающей ее формулы, чем создавать реальное техническое устройство. Именно в этом состоит смысл математического моделирования.

Сумматор –

это электронная логическая схема, выполняющая суммирование двоичных чисел.

В целях максимального упрощения работы компьютера все многообразие математических операций в процессоре сводится к сложению двоичных чисел. Поэтому главной частью процессора является сумматор, который обеспечивает такое сложение. При сложении двоичных чисел образуется сумма в данном разряде, при этом возможен перенос в старший разряд. Обозначим слагаемые A и B , сумму S и перенос P . Построим таблицу сложения одноразрядных двоичных чисел с учетом переноса в старший разряд.

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

Составим булево выражение по этой таблице:

$$S = \bar{A}\bar{B} + A\bar{B}; \quad P = A\bar{B}$$

Упростим формулу для S :

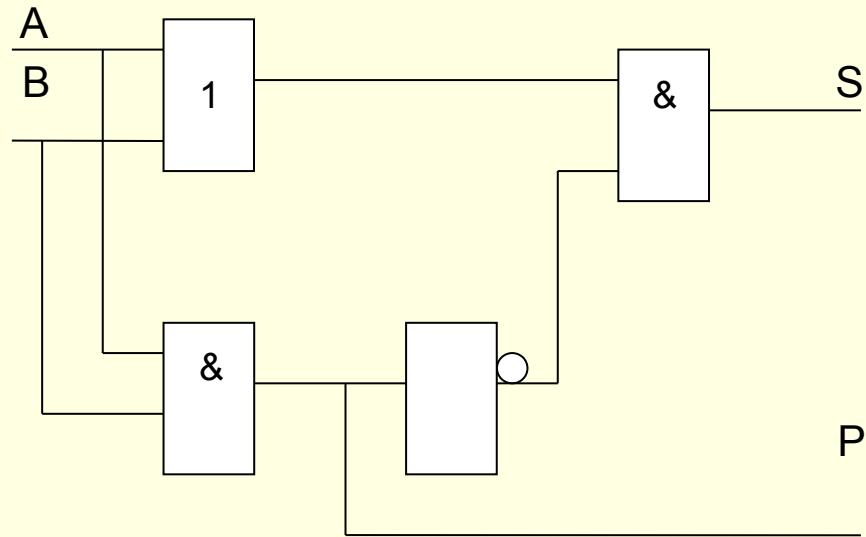
$$\bar{A}\bar{B} = \bar{A}\bar{A} + \bar{A}\bar{B} = \bar{A}(A + B),$$

$$A\bar{B} = A\bar{B} + B\bar{B} = \bar{B}(A + B).$$

$$S = \bar{A}\bar{B} + A\bar{B} = \bar{A}(A + B) + \bar{B}(A + B) =$$

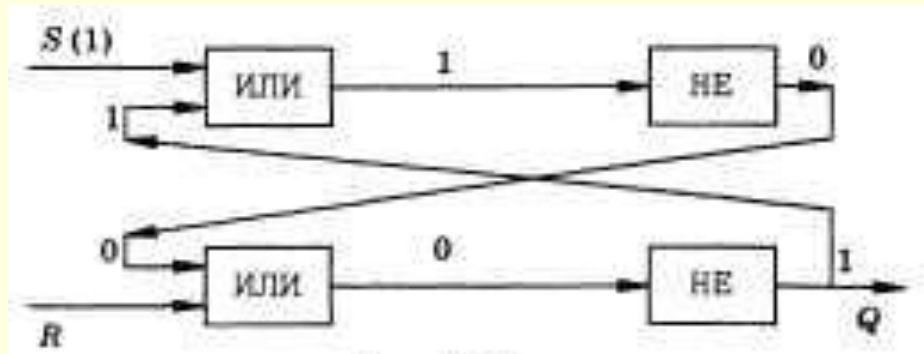
$$(A + B) \&(\bar{A} + \bar{B}) = (A + B)\overline{A\&B}.$$

Схема сумматора



Триггер. Регистры.

Триггер – устройство памяти компьютера для хранения одного бита информации. Это устройство позволяет запоминать, хранить и считывать информацию. Триггер может находиться в одном из двух устойчивых состояний, которые соответствуют логической «1» и логическому «0». Триггер способен почти мгновенно переходить из одного электрического состояния в другое и наоборот. Самый распространенный триггер – SR-триггер (S и R от английских слов set – установка, reset – сброс). Он имеет два входа S и R, два выхода Q и $\neg Q$. На каждый из входов подаются входные сигналы в виде кратковременных импульсов «1», отсутствие импульса – «0». Для построения триггера достаточно двух логических элементов «ИЛИ» и двух элементов «НЕ».



Запрещено

При подаче сигнала на вход S триггер переходит в устойчивое единичное состояние.

При подаче сигнала на вход R триггер сбрасывается в устойчивое нулевое состояние.

При отсутствии входных сигналов триггер сохраняет тот сигнал, который был установлен входным импульсом.

Если на два входа подан сигнал, то появляется неоднозначный результат, поэтому такая комбинация запрещена.

S	R	Q	\bar{Q}	Режим триггера
0	0	Последнее значение		Хранение 1 бит
0	1	0	1	Установка «0»
1	0	1	0	Установка «1»
1	1	Запрещено		

Регистр

- это устройство, предназначенное для хранения многобитового двоичного числового кода, которым можно представлять и адрес, и команду, и данные. Если в регистр входит N триггеров, то можно запомнить N бит информации. Регистры содержатся в различных вычислительных узлах компьютера – процессоре, периферийных устройствах и т.д.