

## Основы математической логики

### 1. Основные понятия формальной логики

Слово логика означает совокупность правил, которым подчиняется процесс мышления. Сам термин "логика" происходит от древнегреческого *logos*, означающего "слово, мысль, понятие, рассуждение, закон". Формальная логика - наука о формах и законах мышления. Законы логики отражают в сознании человека свойства, связи и отношения объектов окружающего мира. Логика как наука позволяет строить формальные модели окружающего мира, отвлекаясь от содержательной стороны. Основными формами мышления являются понятия, суждения и умозаключения. Понятие - это форма мышления, которая выделяет существенные признаки предмета или класса предметов, отличающие его от других. Например, компьютер, человек, ученики.

Суждения - это форма мышления, в которой утверждается или отрицается связь между предметом и его признаком, отношения между предметами или факт существования предмета и которая может быть либо истинной, либо ложной. Языковой формой выражения суждения является повествовательное предложение. Вопросительные и побудительные предложения суждениями не являются.

Суждения рассматриваются не с точки зрения их смысла и содержания, а только с точки зрения их истинности или ложности. Истинным будет суждение, в котором связь понятий правильно отражает свойства и отношения реальных объектов.

"Дважды два равно четырем" - истинное суждение, а вот "Процессор предназначен для печати" - ложное. Суждения могут быть простыми и сложными. "Весна наступила, и грачи прилетели" - сложное суждение, состоящее из двух простых.

Простые суждения (высказывания) выражают связь двух понятий. Сложные - состоят из нескольких простых суждений.

Умозаключение - прием мышления, позволяющий на основе одного или нескольких суждений-посылок получить новое суждение (знание или вывод).

Примерами умозаключений являются доказательства теорем в геометрии.

Посылками умозаключения по правилам формальной логики могут быть только истинные суждения. Тогда и умозаключение будет истинным. Иначе можно прийти к ложному умозаключению.

Математическая логика изучает вопросы применения математических методов для решения логических задач и построения логических схем, которые лежат в основе работы любого компьютера. Суждения в математической логике называют высказываниями или логическими выражениями. Подобно тому, как для описания

действий над переменными был разработан раздел математики алгебра, так и для обработки логических выражений в математической логике была создана алгебра высказываний, или алгебра логики.

## 2. Логические выражения и логические операции

Логическое выражение - это символическая запись, состоящая из логических величин (констант или переменных), объединенных логическими операциями (связками).

В булевой алгебре простым высказываниям ставятся в соответствие логические переменные, значение которых равно 1, если высказывание истинно, и 0, если высказывание ложно. Обозначаются логические переменные буквами латинского алфавита.

Существуют разные варианты обозначения истинности и ложности переменных:

Истина И True T 1

Ложь Л False F 0

Связки "НЕ", "И", "ИЛИ" заменяются логическими операциями инверсия, конъюнкция, дизъюнкция. Это основные логические операции, при помощи которых можно записать любое логическое выражение.

Логическое отрицание (инверсия).

В обыденной речи мы часто пользуемся словом "НЕ", или словами "НЕВЕРНО, ЧТО", когда хотим что-то отрицать. Пусть, например, кто-то сказал: "Тоска зеленая." (Обозначим это высказывание А). Если Вы не согласны, Вы скажете: "Тоска НЕ зеленая." Или: "Неверно, что тоска зеленая." (Ваше высказывание обозначим В). Нетрудно заметить, что значения истинности высказываний А и В находятся в определенной связи: если А истинно, то В ложно, и наоборот. Операция, с помощью которой из высказывания А получается высказывание В, называется логическим отрицанием и само высказывание В называется отрицанием высказывания А и обозначается  $\neg A$ .

Таким образом, отрицанием  $\neg A$  некоторого высказывания А называется такое высказывание, которое истинно, когда А ложно, и ложно, когда А истинно.

Отрицание высказывания А обозначим  $\neg A$ . Определение отрицания может быть записано с помощью так называемой таблицы истинности:

A  $\neg A$

И Л

Л И

В ней указано, какие значения истинности (Истина, Ложь) принимает отрицание  $\neg A$  в зависимости от значений истинности исходного высказывания А.

Логическое умножение (конъюнкция) от латинского conjunctio - союз, связь.

Если два высказывания соединены союзом "И", то полученное сложное высказывание обычно считается истинным тогда и только тогда, когда истинны оба составляющие его высказывания. Если хотя бы одно из составляющих высказываний ложно, то и полученное из них с помощью союза "И" сложное высказывание также считается ложным. Например, возьмем два высказывания: "У кота есть хвост" (А), "У

зайца есть хвост" (B). Сложное высказывание "У кота есть хвост и у зайца есть хвост" истинно, т.к. истинны оба высказывания A и B. Но если взять другие высказывания: "У кота длинный хвост" (C), "У зайца длинный хвост" (D), то сложное высказывание "У кота длинный хвост и у зайца длинный хвост" будет ложным, т.к. ложно высказывание (D). Таким образом, исходя из обычного смысла союза "И", приходим к определению соответствующей логической операции - конъюнкции.

Таким образом, конъюнкцией двух высказываний A и B называется такое высказывание, которое истинно тогда и только тогда, когда истинны оба высказывания A и B.

Конъюнкцию высказываний A и B мы обозначим:  $A \& B$ . Знак  $\&$  - амперсент - читается как английское "and" (помните Procter & Gamble или Wash & Go?). Часто встречается обозначение  $A \text{ л } B$ . Иногда, для краткости, пишут просто  $AB$ .

Определение конъюнкции может быть записано в виде таблицы истинности, в которой для каждого из четырех возможных наборов значений исходных высказываний A и B задается соответствующее значение конъюнкции  $A \& B$ :

$A \text{ л } B \text{ л } A \& B$

и и и

и л л

л и л

л л л

Определение конъюнкции двух высказываний естественным образом распространяется на любое конечное число составляющих: конъюнкция  $A_1 \& A_2 \& A_3 \& \dots \& A_n$  истинна тогда и только тогда, когда истинны все высказывания  $A_1, A_2, A_3, \dots, A_n$  (а, следовательно, ложна, когда ложно хотя бы одно из этих высказываний).

Логическое сложение (дизъюнкция) от латинского *disjunctio* - разобщение, различие. Если два высказывания соединены союзом "ИЛИ", то полученное сложное высказывание обычно считается истинным, когда истинно хотя бы одно из составляющих высказываний. Например, возьмем два высказывания: "Мел черный." (A), "Доска черная." (B). Высказывание "Мел черный или доска черная" будет истинным, т.к. одно из исходных высказываний (B) истинно.

Таким образом, дизъюнкцией двух высказываний называется такое новое высказывание, которое истинно тогда и только тогда, когда истинно хотя бы одно из этих высказываний.

Дизъюнкцию высказываний A и B мы обозначим символом  $A \vee B$  и будем читать: A или B.

Определение дизъюнкции двух высказываний естественным образом распространяется на любое конечное число составляющих: дизъюнкция  $A_1 \vee A_2 \vee A_3 \vee \dots \vee A_n$  истинна тогда и только тогда, когда истинно хотя бы одно из высказываний  $A_1, A_2, A_3, \dots, A_n$  (а следовательно, ложна, когда ложны все эти высказывания).

Логическое следование (импликация) от латинского *implicatio* - тесно связываю. В наших рассуждениях, особенно в математических доказательствах, мы часто пользуемся сложными высказываниями, образованными с помощью слов "если...", "то...". Здесь высказывание, расположенное после слова "если", называется

основанием или посылкой, а высказывание, расположенное после слова "то", называется следствием или заключением.

Рассмотрим пример: из арифметики. Вам должно быть известно, что утверждение "если каждое слагаемое делится на 3, то и сумма делится на 3" истинно, т.е. из высказывания "каждое слагаемое делится на 3" следует высказывание "сумма делится на 3". Посмотрим, какие наборы значений истинности посылки и заключения возможны, когда истинно все утверждение. Возьмем, например, в качестве слагаемых числа 6 и 9. В этом случае истинны и посылка, и заключение, и все утверждение. Если же взять числа 4 и 5, то посылка будет ложной, а заключение истинным. Для чисел 4 и 7 и посылка и заключение ложны. (Если Вы сомневаетесь в истинности высказывания для последнего случая попробуйте произнести его в сослагательном наклонении: если бы числа 4 и 7 делились бы на 3, то и их сумма делилась бы на 3). Очевидно, что только один случай невозможен: мы не найдем таких двух слагаемых, чтобы каждое из них делилось на 3, а их сумма не делилась на 3, т.е. чтобы посылка была истинной, а заключение ложным. Из истины не может следовать ложь, иначе логика теряет смысл. Высказывание "Если А, то В" с логической точки зрения имеет тот же смысл, что и высказывание "неверно, что А истинно и В ложно". Это означает, что функцию импликации можно заменить комбинацией двух функций (отрицания и конъюнкции). Обычно, когда мы хотим установить ложность высказывания "Если А, то В", мы стараемся показать, что возможен случай, когда А истинно, а В ложно (доказательство "от противного"). Обозначим импликацию символом  $\Rightarrow$  и запись "А  $\Rightarrow$  В" будем читать: "Из А следует В".

Таким образом, импликацией А  $\Rightarrow$  В называется высказывание, которое ложно тогда и только тогда, когда А истинно и В ложно.

Логическое тождество (эквиваленция).

Интуитивно можно догадаться, что высказывания эквивалентны (равносильными), когда их значения истинности одинаковы. Например, эквивалентны высказывания: "железо тяжелое" и "пух легкий", так же как и высказывания: "железо легкое" и "пух тяжелый". Обозначим эквиваленцию символом  $\Leftrightarrow$  и запись "А  $\Leftrightarrow$  В" будем читать "А эквивалентно В", или "А равносильно В", или "А, если и только если В".

Таким образом, эквиваленцией двух высказываний А и В называется такое высказывание, которое истинно тогда и только тогда, когда оба эти высказывания А и В истинны или оба ложны.

Отметим, что высказывание типа "А, если и только если В" можно заменить высказыванием "Если А, то В и, если В, то А" (обдумайте это на досуге и обратите внимание на символ  $\Leftrightarrow$ ). Следовательно, функцию эквиваленции можно заменить комбинацией функций импликации и конъюнкции. Запишем таблицу истинности для эквиваленции:

Приведем примеры записи сложных высказываний с помощью обозначения логических связей:

"Быть или не быть - вот в чем вопрос." (В. Шекспир)  $A \vee \neg A \Leftrightarrow B$

"Если хочешь быть красивым, поступи в гусары." (К. Прутков)  $A \Rightarrow B$

### 3. Построение таблиц истинности для логических функций

Логическая функция - это функция, в которой переменные принимают только два значения: логическая единица или логический ноль. Истинность или ложность сложных суждений представляет собой функцию истинности или ложности простых. Эту функцию называют булевой функцией суждений  $f(a, b)$ .

Любая логическая функция может быть задана с помощью таблицы истинности, в левой части которой записывается набор аргументов, а в правой части - соответствующие значения логической функции.

При построении таблицы истинности необходимо учитывать порядок выполнения логических операций. 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.

Предлагается следующий алгоритм построения таблицы истинности.

1. Определить количество наборов входных переменных - всевозможных сочетаний значений переменных, входящих в выражения, по формуле:  $Q=2^n$ , где  $n$  - количество входных переменных. Оно определяет количество строк таблицы.
2. Внести в таблицу все наборы входных переменных.
3. Определить количество логических операций и последовательность их выполнения.
4. Заполнить столбцы результатами выполнения логических операций в обозначенной последовательности.

Чтобы не повторить или не пропустить ни одного возможного сочетания значений входных переменных, следует пользоваться одним из предлагаемых ниже способов заполнения таблицы.

Способ 1. Каждый набор значений исходных переменных есть код числа в двоичной системе счисления, причем количество разрядов числа равно количеству входных переменных. Первый набор - число 0. Прибавляя к текущему числу каждый раз по 1, получаем очередной набор. Последний набор - максимальное значение двоичного числа для данной длины кода.

Например, для функции от трех переменных последовательность наборов состоит из чисел:

000  
001  
010  
011

100  
101  
110  
111

Способ 2. Для функции от трех переменных последовательность данных можно получить следующим путем:

- а) разделить колонку значений первой переменной пополам и заполнить верхнюю половину нулями, нижнюю половину единицами;
- б) в следующей колонке для второй переменной половинку снова разделить пополам и заполнить группами нулей и единиц; аналогично заполнить вторую половинку;
- в) так делать до тех пор, пока группы нулей и единиц не будут состоять из одного символа.

Способ 3. Воспользоваться известной таблицей истинности для двух аргументов. Добавляя третий аргумент, сначала записать первые 4 строки таблицы, сочетая их со значением третьего аргумента, равным 0, а затем еще раз записать эти же 4 строки, но теперь уже со значением третьего аргумента, равным 1. В результате в таблице для трех аргументов окажется 8 строк:

000  
010  
100  
110  
001  
011  
101  
111

Например, построим таблицу истинности для логической функции:

Количество входных переменных в заданном выражении равно трем (А,В,С). Значит, количество входных наборов  $Q=2^3=8$ .

Столбцы таблицы истинности соответствуют значениям исходных выражений А,В,С, промежуточных результатов и  $(B \vee C)$ , а также искомого окончательного значения сложного арифметического выражения :

А	В	С	$B \vee C$
0	0	0	0
0	0	1	1
0	1	1	1
0	1	0	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

7.4. Логические функции и их преобразования. Законы логики

Для операций конъюнкции, дизъюнкции и инверсии определены законы булевой алгебры, позволяющие производить тождественные (равносильные) преобразования логических выражений.

Законы логики

1.  $\neg\neg A \Leftrightarrow A$  закон двойного отрицания;
2.  $A \& B \Leftrightarrow B \& A$  коммутативность конъюнкции;
3.  $A \vee B \Leftrightarrow B \vee A$  коммутативность дизъюнкции;
4.  $A \& (B \& C) \Leftrightarrow (A \& B) \& C$  ассоциативность конъюнкции;
5.  $A \vee (B \vee C) \Leftrightarrow (A \vee B) \vee C$  ассоциативность дизъюнкции;
6.  $A \& (B \vee C) \Leftrightarrow (A \& B) \vee (A \& C)$  дистрибутивность конъюнкции относительно дизъюнкции;
7.  $A \vee (B \& C) \Leftrightarrow (A \vee B) \& (A \vee C)$  дистрибутивность дизъюнкции относительно конъюнкции;
8.  $A \& A \Leftrightarrow A$
9.  $A \vee A \Leftrightarrow A$
10.  $A \vee \neg A \Leftrightarrow И$  закон исключенного третьего;
11.  $A \& \neg A \Leftrightarrow Л$  закон непротиворечия;
12.  $A \& И \Leftrightarrow A$
13.  $A \vee И \Leftrightarrow И$
14.  $A \& Л \Leftrightarrow Л$
15.  $A \vee Л \Leftrightarrow A$
16.  $\neg(A \& B) \Leftrightarrow \neg A \vee \neg B$  законы де Моргана;
17.  $\neg(A \vee B) \Leftrightarrow \neg A \& \neg B$
18.  $A \Rightarrow B \Leftrightarrow \neg A \vee B$  замена импликации.

Основываясь на законах, можно выполнять упрощение сложных логических выражений. Такой процесс замены сложной логической функции более простой, но равносильной ей, называется минимизацией функции.

Пример 1. Упростить выражения так, чтобы в полученных формулах не содержалось отрицания сложных высказываний.

Решение

Пример 2. Минимизировать функцию

Решение

При упрощении выражения использовались формулы поглощения и склеивания.

Пример 3. Найти отрицание следующего высказывания: "Если урок будет интересным, то никто из учеников (Миша, Вика, Света) не будет смотреть в окно".

Решение

Обозначим высказывания:

У - "Урок интересный";

М - "Миша смотрит в окно";

В - "Вика смотрит в окно";

С - "Света смотрит в окно".

При упрощении выражения использовались формула замены операций и закон де Моргана.

Пример 4. Определить участника преступления, исходя из двух посылок: логический компьютер таблица

1) "Если Иванов не участвовал или Петров участвовал, то Сидоров участвовал";

2) "Если Иванов не участвовал, то Сидоров не участвовал".

Решение

Составим выражения:

I - "Иванов участвовал в преступлении";

P - "Петров участвовал в преступлении";

S - "Сидоров участвовал в преступлении".

Запишем посылки в виде формул:

Тогда

Проверим результат, используя таблицу истинности:

Ответ: Иванов участвовал в преступлении.

Построение логической функции по ее таблице истинности

Мы научились составлять таблицу истинности для логической функции. Попробуем решить обратную задачу.

Рассмотрим строки, где значение истинности функции Z истинно ( $Z=1$ ). Функцию для этой таблицы истинности можно составить следующим образом:  $Z(X,Y) = (\neg X \& \neg Y) \vee (X \& \neg Y)$ .

Каждой строке, где функция истинна (равна 1), соответствует скобка, представляющая собой конъюнкцию аргументов, причем если значение аргумента 0, то мы берем его с отрицанием. Все скобки соединены между собой операцией дизъюнкции. Полученную формулу можно упростить, применив законы логики:

$$\begin{aligned} Z(X,Y) &\Leftrightarrow ((\neg X \& \neg Y) \vee X) \& ((\neg X \& Y) \vee \neg Y) \Leftrightarrow (X \vee (\neg X \& \neg Y)) \& (\neg Y \vee (\neg X \& \neg Y)) \Leftrightarrow \\ &((X \vee \neg X) \& (X \vee \neg Y)) \& ((Y \vee \neg X) \& (\neg Y \vee \neg Y)) \Leftrightarrow (1 \& (X \vee \neg Y)) \& ((\neg Y \vee \neg X) \& \neg Y) \Leftrightarrow (X \vee \neg Y) \& ((\neg Y \vee \neg X) \& \neg Y). \end{aligned}$$

Проверьте полученную формулу: составьте таблицу истинности для функции  $Z(X,Y)$ .

Запишите правила конструирования логической функции по ее таблице истинности:

1. Выделить в таблице истинности те строки, в которых значение функции равно 1.
2. Выписать искомую формулу в виде дизъюнкции нескольких логических элементов. Число этих элементов равно числу выделенных строк.
3. Каждый логический элемент в этой дизъюнкции записать в виде конъюнкции аргументов функции.
4. Если значение какого-либо аргумента функции в соответствующей строке таблице равно 0, то этот аргумент мы берем с отрицанием.
5. Построение логических схем

Знания из области математической логики можно использовать для конструирования 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. Устройства, фиксирующие два устойчивых состояния, называются бистабильными (например, выключатель, реле). Если вы помните, первые вычислительные машины были релейными. Позднее были созданы новые устройства управления электричеством - электронные схемы, состоящие из набора полупроводниковых элементов. Такие электронные схемы, которые преобразовывают сигналы только двух фиксированных напряжений электрического тока (бистабильные), стали называть логическими элементами. На элементарном уровне конъюнкцию можно представить себе в виде последовательно соединенных выключателей, а дизъюнкцию - в виде параллельно соединенных выключателей:

Логические элементы имеют один или несколько входов и один выход, через которые проходят электрические сигналы, обозначаемые условно 0, если "отсутствует" электрический сигнал, и 1, если "имеется" электрический сигнал. Простейшим логическим элементом является инвертор, выполняющий функцию отрицания. Если на вход поступает сигнал, соответствующий 1, то на выходе будет 0. И наоборот. У этого элемента один вход и один выход. На функциональных схемах он обозначается:

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

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

Специальных логических элементов для импликации и эквивалентности нет, т.к.  $A \Rightarrow B$  можно заменить на  $\neg A \vee B$ ;  $A \Leftrightarrow B$  можно заменить на  $(A \& B) \vee (\neg A \& \neg B)$ . Другие логические элементы построены из этих трех простейших и выполняют более сложные логические преобразования информации. Сигнал, выработанный одним логическим элементом, можно подавать на вход другого элемента, это дает возможность образовывать цепочки из отдельных логических элементов. Например: логический компьютер таблица

Эта схема соответствует сложной логической функции  $F(A,B) = \neg(A \vee B)$ . Попробуйте проследить изменения электрического сигнала в этой схеме. Например,

какое значение электрического сигнала (0 или 1) будет на выходе, если на входе:  $A=1$  и  $B=0$ .

Такие цепи из логических элементов называются логическими устройствами.

Логические устройства же, соединяясь, в свою очередь образуют функциональные схемы (их еще называют структурными или логическими схемами). По заданной функциональной схеме можно определить логическую формулу, по которой эта схема работает, и наоборот.

Пример 1. Логическая схема для функции будет выглядеть следующим образом:

Правила составления электронных логических схем по заданным таблицам истинности остаются такими же, как для контактных схем.

Пример 2. Составить логическую схему для тайного голосования трех персон А, В, С, условия которого определяются следующей таблицей истинности:

A 0 0 0 0 1 1 1 1

B 0 0 1 1 0 0 1 1

C 0 1 0 1 0 1 0 1

F 0 0 0 1 0 1 1 1

Решение

По таблице построим СДНФ логической функции и упростим ее:

Правильность полученной формулы можно проверить, составив для нее таблицу истинности:

Значение полученной функции совпадает с исходным, что можно заметить, сравнивая таблицы.

Логическая схема полученной функции имеет вид:

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

Логический элемент И-НЕ состоит из конъюнктора и инвертора:

Выходная функция выражается формулой .

Логический элемент ИЛИ-НЕ состоит из дизъюнктора и инвертора:

Выходная функция выражается формулой .

6. 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.

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.

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

Пусть нам необходимо сложить двоичные числа  $X$  и  $Y$ . Через  $P$  и  $Z$  обозначим первую и вторую цифру суммы:  $X + Y = PZ$ . Вспомните таблицу сложения двоичных чисел.

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

2. Сконструируем функции  $P(X,Y)$  и  $Z(X,Y)$  по этой таблице:

$$P(X,Y) = X \oplus Y; Z(X,Y) = (\neg X \& Y) \vee (X \& \neg Y).$$

Преобразуем вторую формулу, пользуясь законами логики.

$$\begin{aligned} (\neg X \& Y) \vee (X \& \neg Y) &\Leftrightarrow ((\neg X \& Y) \vee X) \& ((\neg X \& Y) \vee \neg Y) \Leftrightarrow (X \vee (\neg X \& Y)) \& (\neg Y \vee (Y \& \neg X)) \\ &\Leftrightarrow ((X \vee \neg X) \& (X \vee Y)) \& ((\neg Y \vee Y) \& (\neg Y \vee \neg X)) \Leftrightarrow (1 \& (X \vee Y)) \& ((\neg Y \vee \neg X)) \Leftrightarrow (X \vee Y) \& \neg(X \& Y). \end{aligned}$$

3. Теперь можно построить функциональную схему одноразрядного полусумматора:

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

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. 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.

Общая схема сумматора:

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:

Триггер имеет два симметричных входа S и R, которые используются для установки в единичное состояние и сброса, - в нулевое. Еще у него есть два симметричных выхода Q и  $\bar{Q}$ , причем выходной сигнал  $\bar{Q}$  является логическим отрицанием сигнала Q.

За единичное состояние триггера условно принимают такое, при котором  $Q=1$ .

На каждый из входов S и R могут подаваться входные сигналы в виде кратковременных импульсов. Наличие импульса на входе считается единицей, а его отсутствие - нулем.

Ниже показана схема реализации триггера с помощью элементов ИЛИ-НЕ и соответствующая таблица истинности.

Два одинаковых двухвходовых логических элемента ИЛИ-НЕ соединены симметричным образом. Сигнал, поданный на один из входов каждого элемента, снимается с выхода другого. Наличие такого соединения и дает триггеру возможность сохранять свое состояние после прекращения действия сигналов (никакой другой логический элемент не в состоянии поддерживать сигнал на выходе после прекращения действия входного напряжения).

Проанализируем возможные комбинации значений входов R и S триггера, используя его схему и таблицу истинности схемы.

1. Пусть поданы сигналы  $S=1, R=0$ . Независимо от состояния другого входа на выходе верхнего элемента появится 0. Этот нулевой сигнал подается на вход нижнего элемента, где уже есть  $R=0$ . Выход нижнего элемента станет равным 1. Эта единица возвращается на вход первого элемента. Теперь состояние другого входа (S) этого

элемента роли не играет: если даже убрать входной сигнал  $S$ , состояние триггера останется без изменения. Поскольку  $Q=1$ , триггер перешел в единичное состояние, устойчивое, пока не придут новые внешние сигналы.

2. При  $S=0$  и  $R=1$  вследствие симметричности схемы все происходит аналогично, но теперь на выходе  $Q$  будет 0. То есть при подаче сигнала на вход  $R$  триггер сбрасывается в устойчивое нулевое состояние.

3. При окончании действия обоих сигналов ( $R=0$  и  $S=0$ ) триггер сохраняет на выходе  $Q$  тот сигнал, который был установлен входным импульсом  $S$  или  $R$ .

4. подача импульсов одновременно на входы  $R$  и  $S$  может привести к неоднозначному результату, поэтому эта комбинация входных сигналов ( $R=1$  и  $S=1$ ) запрещена.

Один триггер хранит один бит информации. Для хранения одного байта информации необходимо 8 триггеров. Современные микросхемы памяти содержат миллионы триггеров.

По технологии изготовления память делится на статическую и динамическую. На триггерах основана статическая память, а динамическая устроена по принципу конденсатора: заряженный конденсатор соответствует единице, а незаряженный - нулю.

Динамическая память проще по устройству, имеет больший объем и дешевле. В силу этих преимуществ в настоящее время основной объем оперативного запоминающего устройства компьютера является динамическим.

Однако статическая память имеет более высокое быстродействие. Кэш-память имеет статическую природу, что позволяет согласовать высокое быстродействие процессора и низкую скорость работы динамической памяти.

Конденсаторы динамической памяти постепенно разряжаются через внешние цепи, и потому требуют периодической подзарядки, чтобы не потерять информацию. Этот процесс называется регенерацией памяти, его наличие усложняет подключение микросхем динамической памяти. Микросхема статической памяти сильнее нагревается при работе, так как использует активные элементы - транзисторы.

Некоторое количество триггеров, объединенных вместе общей системой управления, называется регистром. Регистры содержатся в различных вычислительных узлах компьютера - процессоре, периферийных устройствах и т.д. Регистр - это устройство, предназначенное для хранения многозарядного двоичного числового кода, которым можно представлять и адрес, и команду, и данные.

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.

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 Т-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.

Техническая сторона логики компьютера основана на технологии транзистора, что позволяет получать одну из двух возможных единиц информации (0 и 1), оперируя с передачей или отсутствием передачи тока. На следующем уровне вступает система носителей (переносчиков) информации - это нули и единицы (0 и 1), которые отражают в себе реальную информацию путем применения, как систем счисления, так и системы команд микропроцессора.

Логика микропроцессора (МП). МП имеет (может иметь) встроенную логику, основанную на формальной логике человека. Таким образом, имеется возможность обрабатывать информацию. Так, например, получая группу информационных потоков (два потока), МП может, используя формальную логику, выдать один искомый поток.

Логика операционной системы. Хотя технически возможна работа на компьютере без использования операционной системы (ОС), так как в ПЗУ находятся (могут находиться) для этого специальные программы, целесообразность использования ОС никем не подвергается сомнению. ОС представляет собой программу (группу программ), которая обеспечивает полный системный интерфейс компьютера. И, наконец, логика прикладных программ. В данном случае все зависит от фантазии и профессионализма программиста или пользователя.

Проиллюстрируем уровни логики ЭВМ. Получив на два регистра по единице и команду реализовать функцию AND, МП, применив заложенную в нем формальную логику, выдает на результирующий регистр единицу. Эта единица некоторым образом интерпретируется операционной системой, например, как истинность выполненной операции, а затем передается (может передаваться) прикладной программе, которая в свою очередь так же интерпретирует полученную информацию и производит в соответствии с этим некоторые действия.

Вопросы для самоконтроля

1. Основные логические операции: конъюнкция, дизъюнкция (оба вида), отрицание, импликация, эквивалентность. Примеры логических выражений.
2. Таблица истинности. Примеры.  $A \text{ and not } A$ ;  $A \text{ or not } A$
3. Основные законы математической логики: перестановочное, сочетательное и распределительное
4. Законы де Моргана (закон отрицания).
5. (Совершенная) дизъюнктивная нормальная форма. Примеры.