

## Лекция. Информационно-логические основы построения и функционирования ПК

### План:

1. Основные логические элементы и функции в системе ПК.
2. Анализ и синтез логических высказываний.
3. Операции над логическими высказываниями.

### Вопрос 1.

Математическая логика, основы которой были заложены Г.Лейбницем еще в 17 веке, сформировалась как научная дисциплина только в середине 19 века благодаря работам математика **Джорджа Буля**, который создал алгебру логики (булева алгебра).

Электронные микросхемы компьютера представляют собой автономные системы большого количества логических элементов, обрабатывающих электрические сигналы типа "0" и "1" (соответственно: "ложь" и "истина" – булевы логические константы). Благодаря этому происходит кодирование, передача, обработка и хранение любой информации в ПЭВМ. **В основе микросхем лежит Булева Алгебра.**

**Логическое высказывание** – это повествовательное предложение, относительно которого можно однозначно сказать, истинно оно или ложно.

#### Обозначение высказываний:

**A** – Сейчас идет дождь.

**B** – Форточка открыта. **простые высказывания (элементарные)**

Любое высказывание может быть ложно (0) или истинно (1).

**Составные высказывания** строятся из простых с помощью логических связок (операций) "**и**", "**или**", "**не**", "**если ... то**", "**тогда и только тогда**" и др.

**Логический элемент компьютера** — это часть электронной логической схемы, которая реализует элементарную логическую функцию.

**Каждый логический элемент** имеет свое условное обозначение (несколько входов и, как правило, один выход), которое выражает его логическую функцию, но не указывает на то, какая именно электронная схема в нем реализована. Это упрощает запись и понимание сложных логических схем (табл. 1).

#### Основные логические элементы:

- Инвертор
- Конъюнктор
- Дизъюнктор
- Элемент Шеффера
- Элемент Пирса

## Основные логические функции:

- Инверсия – логическое отрицание, функция **НЕ**
- Конъюнкция – логическое умножение, функция **И**
- Дизъюнкция – логическое сложение, функция **ИЛИ**
- Импликация – логическое следование  $\rightarrow$
- Эквивалентность – логическое равенство  $\leftrightarrow$
- Функция Шеффера – отрицание логического умножения **И-НЕ**
- Функция Пирса – отрицание логического сложения **ИЛИ-НЕ**

**Таблица истинности** это табличное представление логической схемы (операции), в которой перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.

## Вопрос 2. Анализ и синтез логических высказываний.

**Логическая операция** – это способ построения сложного высказывания из данных высказываний, при котором значение истинности сложного высказывания полностью определяется значениями истинности исходных высказываний.

**1. Логическое отрицание** образуется из высказывания с помощью добавления частицы «**НЕ**» к сказуемому или использования оборота речи «неверно, что ...».

**Например:** *Я не знаю китайского языка.*

*Неверно, что я знаю китайский язык.*

**Обозначение инверсии:** **НЕ А;  $\bar{A}$ ; A; NOT A**

При этом инверсия (логическое отрицание) показывается в логической формуле в виде горизонтальной линии над аргументом; при графическом изображении – в виде кружка на выходе из логического элемента.

# Таблица истинности для инверсии

A	$\bar{A}$	Смысл высказывания A для указанных значений	Значение высказывания: <i>Я не знаю китайского языка</i>
0	1	<i>Я не знаю китайского языка</i>	Истина
1	0	<i>Я знаю китайский язык</i>	Ложь

Из таблицы истинности следует, что инверсия высказывания истинна, когда высказывание ложно.

**2. Логическое умножение (конъюнкция)** образуется соединением двух высказываний в одно с помощью союза «**И**».

**Обозначение конъюнкции:** **A И B;  $A \wedge B$ ; A&B; A AND B.**

**Например:** На автостоянке обычно стоят две машины: «Мерседес» и «Жигули», но может находиться и какая-то одна из них или не быть ни одной. Обозначим высказывания:

$A$  = На автостоянке стоит «Мерседес».

$B$  = На автостоянке стоят «Жигули».

$(A \text{ конъюнкция } B)$  = На автостоянке стоят «Мерседес» и «Жигули».

**Таблица истинности для конъюнкции**

$A$	$B$	$A \& B$	Смысл высказываний $A$ и $B$ для указанных значений		Значение высказывания На автостоянке стоят «Мерседес» и «Жигули»
0	0	0	«Мерседес» не стоит	«Жигули» не стоят	Ложь
0	1	0	«Мерседес» не стоит	«Жигули» стоят	Ложь
1	0	0	«Мерседес» стоит	«Жигули» не стоят	Ложь
1	1	1	«Мерседес» стоит	«Жигули» стоят	Истина

Из таблицы истинности следует, что конъюнкция двух высказываний истинна тогда и только тогда, когда оба высказывания истинны, и ложна, когда хотя бы одно высказывание ложно.

### 3. Логическое сложение (дизъюнкция)

**Логическое сложение** образуется соединением двух высказываний в одно с помощью союза «ИЛИ».

**Например:**

На автостоянке обычно стоят две машины: «Мерседес» и «Жигули», но может находиться и какая-то одна из них или не быть ни одной.

Обозначим высказывания:

$A$  = На автостоянке стоит «Мерседес».

$B$  = На автостоянке стоят «Жигули».

$(A \text{ дизъюнкция } B)$  = На автостоянке стоят «Мерседес» или «Жигули».

**Обозначение дизъюнкции:**

$A \text{ ИЛИ } B$ ;  $A \vee B$ ;  $A | B$ ;  $A \text{ OR } B$ ;  $A + B$ .

**Таблица истинности для дизъюнкции**

$A$	$B$	$A \vee B$	Смысл высказываний $A$ и $B$ для указанных значений		Значение высказывания На автостоянке стоят «Мерседес» и «Жигули»
0	0	0	«Мерседес» не стоит	«Жигули» не стоят	Ложь
0	1	1	«Мерседес» не стоит	«Жигули» стоят	Истина
1	0	1	«Мерседес» стоит	«Жигули» не стоят	Истина
1	1	1	«Мерседес» стоит	«Жигули» стоят	Истина

Из таблицы истинности следует, что дизъюнкция двух высказываний ложна тогда и только тогда, когда оба высказывания ложны, и истинна, когда хотя бы одно высказывание истинно.



#### 4. Логическое следование (импликация $\rightarrow$ )

**Логическое следование** образуется соединением двух высказываний в одно с помощью оборота речи «если ..., то ...».

**Например:**  $A$ =Если клятва дана, то она должна выполняться.  
 $B$ =Если число делится на 9, то оно делится на 3.

В логике допустимо рассматривать и бессмысленные с житейской точки зрения высказывания:  $C$  = Если коровы летают, то  $2+2=5$ .

Пусть даны высказывания:  $A$ =На улице дождь.  $B$ =Асфальт мокрый.  
( $A$  импликация  $B$ )= Если на улице дождь, то асфальт мокрый.

**Обозначение конъюнкции:**  $A \rightarrow B$ ;  $A \Rightarrow B$ ;  
если  $A$ , то  $B$ ;  $A$  влечет  $B$ ;  $B$  следует из  $A$ .

Таблица истинности для импликации

$A$	$B$	$A \Rightarrow B$	Смысл высказываний $A$ и $B$ для указанных значений		Значение высказывания <i>Если на улице дождь, то асфальт мокрый</i>
0	0	1	Дождя нет	Асфальт сухой	Истина
0	1	1	Дождя нет	Асфальт мокрый	Истина
1	0	0	Дождь идет	Асфальт сухой	Ложь
1	1	1	Дождь идет	Асфальт мокрый	Истина

Из таблицы истинности следует, что импликация двух высказываний ложна тогда и только тогда, когда из истинного высказывания следует ложное.

#### 5. Логическое равенство (эквивалентность $\leftrightarrow$ )

**Логическое равенство** образуется соединением двух высказываний в одно с помощью оборота речи «...тогда и только тогда, когда ...».

**Например:** Угол называется прямым тогда, когда он равен 90 градусам. Обозначим высказывания:  $A$ =Число делится на 3 без остатка.  
 $B$ =Сумма цифр числа делится нацело на 3.

( $A$  эквивалентно  $B$ ) = Число кратно 3 тогда и только тогда, когда сумма его цифр делится нацело на 3.

**Обозначение эквивалентности:**  $A \equiv B$ ;  $A \leftrightarrow B$ ;  $A \sim B$ .

Таблица истинности для эквивалентности

$A$	$B$	$A \leftrightarrow B$	Смысл высказываний $A$ и $B$ для указанных значений		Значение высказывания <i>Число кратно трем тогда и только тогда, когда сумма его цифр делится нацело на 3</i>
0	0	1	Число не кратно трем	Сумма цифр не кратно трем	Истина
0	1	0	Число не кратно трем	Сумма цифр кратно трем	Ложь
1	0	0	Число кратно трем	Сумма цифр не кратно трем	Ложь
1	1	1	Число кратно трем	Сумма цифр кратно трем	Истина

Из таблицы истинности следует, что эквивалентность двух высказываний ложна тогда и только тогда, когда оба высказывания истинны или оба ложны.

**ОПЕРАЦИЯ ШЕФФЕРА: функция отрицания логического умножения**

**Таблица истинности схемы И—НЕ**

x	y	$\overline{x*y}$
0	0	1
0	1	1
1	0	1
1	1	0

**ОПЕРАЦИЯ ПИРСА: функция отрицания логического сложения**

**Таблица истинности схемы ИЛИ—НЕ**

x	y	$\overline{x+y}$
0	0	1
0	1	0
1	0	0
1	1	0

**Вопрос 3. Операции над логическими высказываниями.**

При вычислении логического (булева) выражения учитывается следующий приоритет выполняемых логических операций:

- 1) операции в скобках;
- 2) инверсия;
- 3) конъюнкция;
- 4) дизъюнкция.

**Пример1.** Определить логическое значение булева высказывания L, если X=1, C=0:

$$\begin{aligned} L &= \overline{C} \vee X \wedge C \vee \overline{X} \wedge \overline{C} = \overline{0} \vee 1 \wedge 0 \vee \overline{1} \wedge \overline{0} = 1 \vee 1 \wedge 0 \vee \overline{0} = \\ &= 1 \vee 1 \wedge 0 \vee 1 = 1 \vee 0 \vee 1 = 1 \vee 1 = 1 \end{aligned}$$

С помощью операций И, ИЛИ и НЕ можно реализовать любую логическую операцию.

**Например:**

Прибор имеет три датчика и может работать, если два из них исправны. Записать в виде формулы ситуацию «авария».

A – "Датчик № 1 неисправен".

B – "Датчик № 2 неисправен".

C – "Датчик № 3 неисправен".

Аварийный сигнал:

X – "Неисправны два датчика".

X – "Неисправны датчики № 1 и № 2" или  
 "Неисправны датчики № 1 и № 3" или  
 "Неисправны датчики № 2 и № 3".

**Логическая формула**

$$X = A \cdot B + A \cdot C + B \cdot C$$

**Составление таблиц истинности**

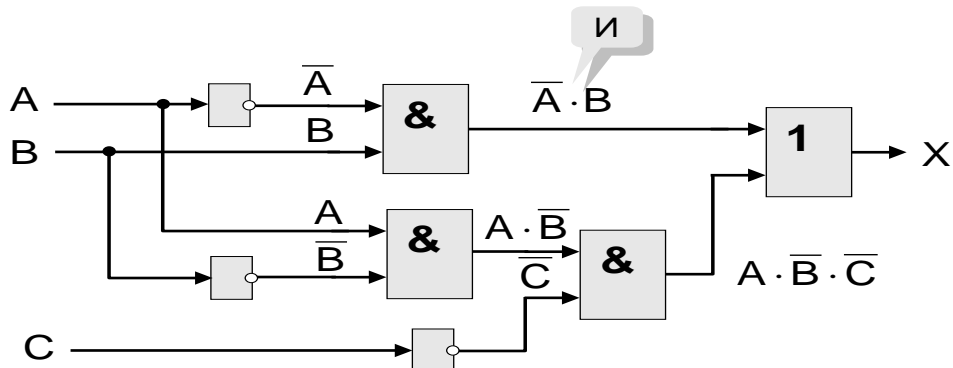
$$X = A \wedge B \vee A \wedge C \vee B \wedge C$$

	A	B	C	A·B	A·C	B·C	X
0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
2	0	1	0	0	0	0	0
3	0	1	1	0	0	1	1
4	1	0	0	0	0	0	0
5	1	0	1	0	1	0	1
6	1	1	0	1	0	0	1
7	1	1	1	1	1	1	1

## Составление схем

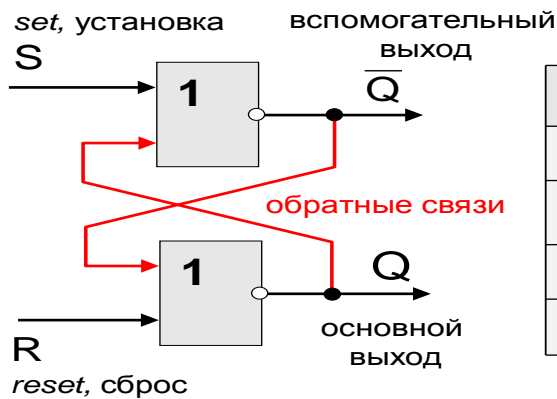
последняя операция - ИЛИ

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



## Триггер (англ. *trigger* – защёлка)

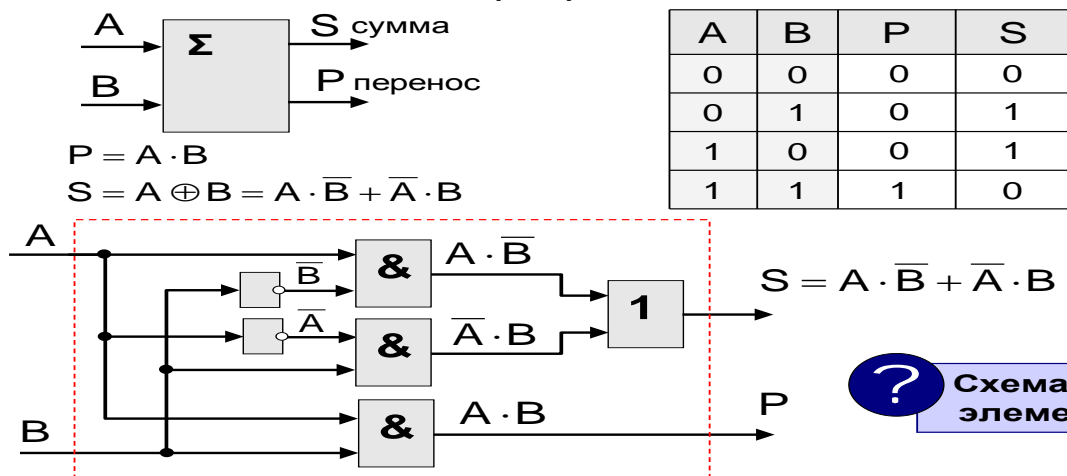
**Триггер** – это логическая схема, способная хранить 1 бит информации (1 или 0). Строится на 2-х элементах **ИЛИ-НЕ** или на 2-х элементах **И-НЕ**.



S	R	Q	$\bar{Q}$	режим
0	0	Q	$\bar{Q}$	хранение
0	1	0	1	сброс
1	0	1	0	установка 1
1	1	0	0	запрещен

## Полусумматор

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



## Сумматор

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

