Некоторые понятия математической логики

Алгебра

- *Алгеброй* называется непустое множество элементов, над которыми определены некоторые операции (часто определяемые по сходству с операциями сложения и умножения чисел), удовлетворяющие определенным аксиомам – *аксиомам алгебры* .
- Операция называется *n-местной*, если она связывает *n* операндов (участников этой операции).
- Совокупность операций алгебры называется ее *сигнатурой*, а совокупность элементов алгебры – *носителем* алгебры.

Основные понятия логики

- *Утверждение* – основное (неопределяемое) понятие математической логики, т.е. некоторая единица, неделимая с точки зрения отражения смысла информации.
- *Высказывание* – некоторое повествовательное утверждение, про которое можно однозначно сказать, истинно оно или ложно. Эти два значения всевозможных *высказываний* обозначаются *"истина"* и *"ложь"*, "true" и "false" или "1" и "0".
- Переменная, значениями которой могут быть лишь значения "1" или "0", называется *логической переменной* или *булевой переменной*.

Пример. Рассмотрим словосочетания:

1. Москва – столица США.
2. Житель города Москва.
3. 5 – 7 + 8.
4. 5 – 9 + 28 = 4.
5. В Антарктиде живут тигры.

Высказывание должно быть однозначно истинным или однозначно ложным, поэтому *высказываниями* являются только утверждения 1), 4), 5).

Высказывания

- Рассматривая *высказывания*, мы не обращаем внимания на их внутреннюю структуру и можем разлагать их на структурные части, равно как и объединять их.
- *Пример.* Построим из ниже заданных простых *высказываний* составные, сложные *высказывания*, принимающие значение *"истина"*, *"ложь"*:
- "Зима – холодное время года".
- "Пальто – теплая одежда".
- "Камин – источник тепла".
- Истинным будет, например, сложное *высказывание*: "Зима – холодное время года и зимой носят пальто", а ложным будет, например, *высказывание*: "Некоторые ходят в пальто, поэтому на улице зима".

Предикат

- *Предикат – высказывательная форма* с логическими переменными (множество значений этих переменных определено), имеющая смысл при любых допустимых значениях этих переменных. Количество переменных в записи *предиката* называется его *местностью*.
- Простые *высказывания* или *предикаты* не зависят от других *высказываний* или *предикатов* ("не разбиваемы на более простые"), а сложные – зависит хотя бы от двух простых.
- *Пример*. Выражение "x = y" – *предикат*, "x > 5" – *предикат*, а "7 > 5" – *высказывание*.
- Утверждение "Хорошо" не является *высказыванием*, утверждение "Оценка студента А по информатике – хорошая" – простое *высказывание*, утверждение "Вчера была ясная погода, я был вчера на рыбалке" – сложное *высказывание*, состоящее из двух простых.

Логическая функция

- *Логической (булевой) функцией f(x)* называется некоторая функциональная зависимость, в которой аргумент x – логическая переменная с заданным множеством изменений аргумента, а значения функции $f(x)$ берутся из двухэлементного множества $R(f) = \{0,1\}$.

Алгебра предикатов

- Множество логических переменных $x, y \in X$ с определенными над ним *операциями*:

  $\overline{x}$ – *отрицания, инверсии*, частица *НЕ*;

  $x \vee y$ – *дизъюнкции, логического сложения*, союз *ИЛИ*;

  $x \wedge y$ – *конъюнкции, логического умножения*, союз *И*;

  называется *алгеброй предикатов* (или *высказываний*), если эти операции удовлетворяют следующим *аксиомам*:

Аксиомы

- Аксиома снятия двойного отрицания:

$$\overline{\overline{x}} = x$$

- Аксиомы переместительности *операндов* относительно *операций* дизъюнкции и конъюнкции (*коммутативность*):

$$x \wedge y = y \wedge x, x \vee y = y \vee x$$

- Аксиомы переместительности *операций* дизъюнкции и конъюнкции относительно *операндов* (*ассоциативность*):

$$(x \wedge y) \wedge z = x \wedge (y \wedge z), (x \vee y) \vee z = x \vee (y \vee z)$$

Аксиомы

- Аксиомы одинаковых операндов (*идемпотентность*):

$$x \wedge x = x, x \vee x = x$$

- Аксиомы поглощения (множителем – множителя-суммы или слагаемым – слагаемого-произведения):

$$x \wedge (x \vee y) = x, x \vee (x \wedge y) = x$$

- Аксиомы распределения операции дизъюнкции относительно конъюнкции и наоборот (*дистрибутивность*):

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z), x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

Аксиомы

- Аксиомы де Моргана (перенесения бинарной операции на операнды):

$$\overline{x \wedge y} = \overline{x} \vee \overline{y}, \overline{x \vee y} = \overline{x} \wedge \overline{y}$$

- Аксиомы нейтральности (взаимноинверсных множителей или слагаемых):

$$x \wedge (y \vee \overline{y}) = x, x \vee (y \wedge \overline{y}) = x$$

- Аксиома существования единицы (истина, true, 1) и нуля (ложь, false, 0), причем,

$$\overline{0} = 1, \overline{1} = 0, \overline{x} \vee x = 1, \overline{x} \wedge x = 0$$

Точность

- Из этих *аксиом* следует ряд полезных соотношений, например,

$$x \wedge 1 = x,$$
$$x \vee 0 = x,$$
$$x \vee 1 = 1,$$
$$x \wedge 0 = 0,$$

Базовые операции алгебры предикатов

- Три базовые операции *алгебры предикатов* определяются таблицей их значений, так как в *алгебре предикатов* из-за дискретности значений *логических функций* часто используется табличная форма задания функции. Булеву функцию n переменных можно полностью определить таблицей из 2^n строк.
- Эти операции определяются соответствующей *таблицей истинности* вида:

x	y	$\overline{x}$	$x \wedge y$	$x \vee y$
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

Другие обозначения

Отрицание (логическое НЕ):

$\overline{x}$ $\neg x$ $!x$

Дизъюнкция (логическое ИЛИ):

$x \vee y$ $x \parallel y$, $x \mid y$, x OR y.

Конъюнкция (логическое И):

$x \wedge y$ x && y, x & y, x AND y.

Пример

- Заполним таблицу истинности трехместной *логической функции* $w = x \wedge y \wedge z$. Эта таблица имеет вид

x	y	z	$y \wedge z$	w
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
1	0	0	0	0
0	1	1	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Небазовые операции

Кроме указанных трех *базовых* операций можно с их помощью ввести еще следующие важные операции алгебры предикатов (можно их назвать *небазовыми* операциями):

- *импликации*: $(x \to y) \equiv (\overline{x} \vee y)$

→ служит для задания так называемых условных

⇒ высказываний «если…, то …», «когда …, тогда », «поэтому»

- *эквивалентности*: $(x \leftrightarrow y) \equiv (x \wedge y \vee \overline{x} \wedge \overline{y})$

↔ служит для задания высказываний «тогда и только

~ тогда, когда … », « если и только если»

- *исключающего ИЛИ*: xor

x	y	x → y	x ↔ y	x xor y
0	0	1	1	0
0	1	1	0	1
1	0	0	0	1
1	1	1	1	0

- При выполнении логических операций в компьютере они сводятся к поразрядному сравнению битовых комбинаций. Эти операции достаточно быстро (аппаратно) выполняемы, так как сводятся к выяснению совпадения или несовпадения битов.
- В логических формулах определено старшинство операций, например: скобки, отрицание, конъюнкция, дизъюнкция (остальные, не базовые операции пока не учитываем).
- Всегда истинные формулы называют ***тавтологиями.***
- Логические функции *эквивалентны*, если совпадают их таблицы истинности, то есть совпадают области определения и значения, а также сами значения функции при одних и тех же наборах переменных из числа всех допустимых значений.

- Задача доказательства равенства двух логических выражений (функций) состоит в установлении эквивалентности этих функций на некотором множестве значений всех переменных, входящих в данную функцию.
- *Пример.* Докажем равенство логических выражений:

$$\overline{x \vee y} \vee \overline{x} \wedge y \wedge \overline{x} = \overline{x}$$

- Используя *аксиомы алгебры предикатов*, получаем равенства

$$\overline{x \vee y} \vee \overline{x} \wedge y \wedge \overline{x} = \overline{x} \wedge \overline{y} \vee \overline{x} \wedge y \wedge \overline{x} = \overline{x} \wedge \overline{y} \vee \overline{x} = \overline{x}$$

- Левая часть равенства приведена к правой части, то есть данное равенство доказано полностью.

Информационно-логическая (инфологическая) задача

- Логические функции позволяют эффективно решать так называемые *инфологические (информационно-логические) задачи*, доказывать утверждения.
- *Информационно-логическая (инфологическая) задача* – это задача, в которой необходимо установить некоторые информационные или логические связи и сделать необходимые причинно-следственные *логические выводы*.
- Эти задачи возникают в различных областях и часто являются плохо формализованными и структурированными. Их нужно хорошо формализовать и структурировать. Насколько хорошо будет возможно это сделать – настолько хорошо и полно будет решена рассматриваемая проблема или задача.

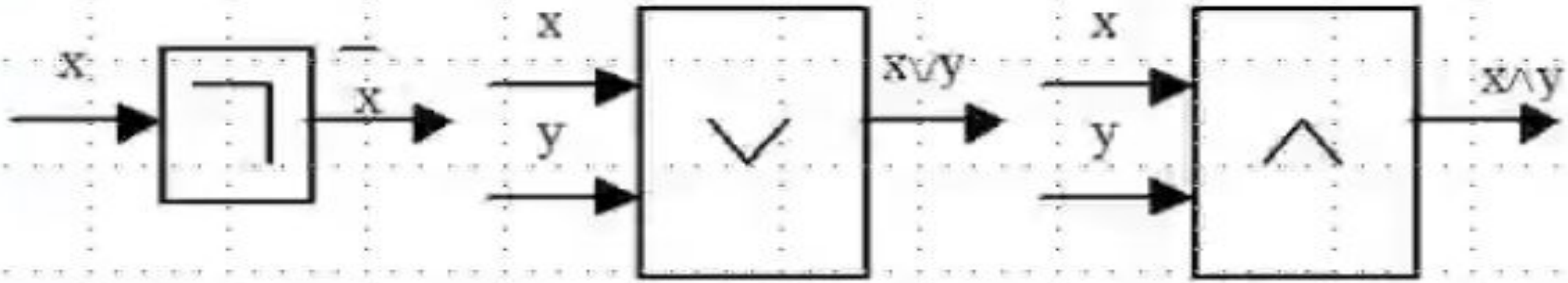
Вентили, логические схемы

- Цифровая схема – это схема, в которой есть только два логических значения.
- Обычно сигнал от 0 до 1В представляет одно значение (например, 0), а сигнал от 2 до 5В – другое значение (например, 1)
- Любой компьютер, точнее, любой его электронный логический блок состоит из десятков и сотен тысяч так называемых *вентилей* (логических устройств, базовых *логических схем*), которые работают по принципу крана (отсюда и название), открывая или закрывая путь сигналам.
- *Логические схемы* предназначены для реализации различных функций алгебры логики (т.е. для вычисления различных функции от двузначных сигналов) и реализуются с помощью трех базовых логических элементов (инвертор, конъюнктор, дизъюнктор).

- Логические функции *отрицания, дизъюнкции* и *конъюнкции* реализуют, соответственно, логические схемы, называемые *инвертором, дизъюнктором* и *конъюнктором.*
- Основные схемы компьютера строится из логических элементов подобно тому, как сложная логическая функция получается путем комбинации более простых функций.
- *Триггер* – электронная схема, имеющий два устойчивых состояния, является типичным запоминающим элементом, способным хранить 1 бит информации.
- *Регистр* – совокупность триггеров, предназначенных для хранения числа в двоичном коде.
- *Сумматор* – устройство, обеспечивающее суммирование двоичных чисел с учетом переноса из предыдущего разряда.

Пример

- Схематически *инвертор*, *дизъюнктор* и *конъюнктор* на логических схемах различных устройств можно изображать условно следующим образом .



Условные обозначения вентилей

Пример:

- В двоичной системе таблицу суммирования цифры x и цифры y и получения цифры z с учетом переноса p в некотором разряде чисел x и y можно изобразить таблицей вида

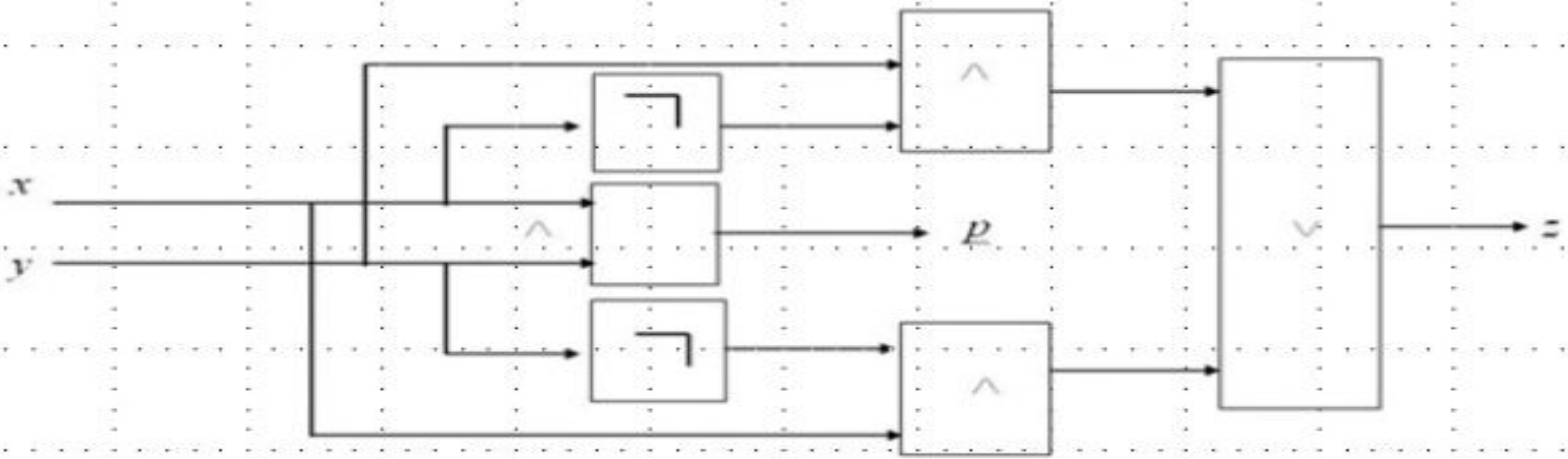
x	y	z	p
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Эту таблицу можно интерпретировать как совместно изображаемую таблицу логических функций (предикатов) вида

$$z = \overline{x} \wedge y \vee x \wedge \overline{y}$$

$$P = x \wedge y$$

- Логический элемент, соответствующий этим функциям, называется одноразрядным *сумматором* и имеет следующую схему:



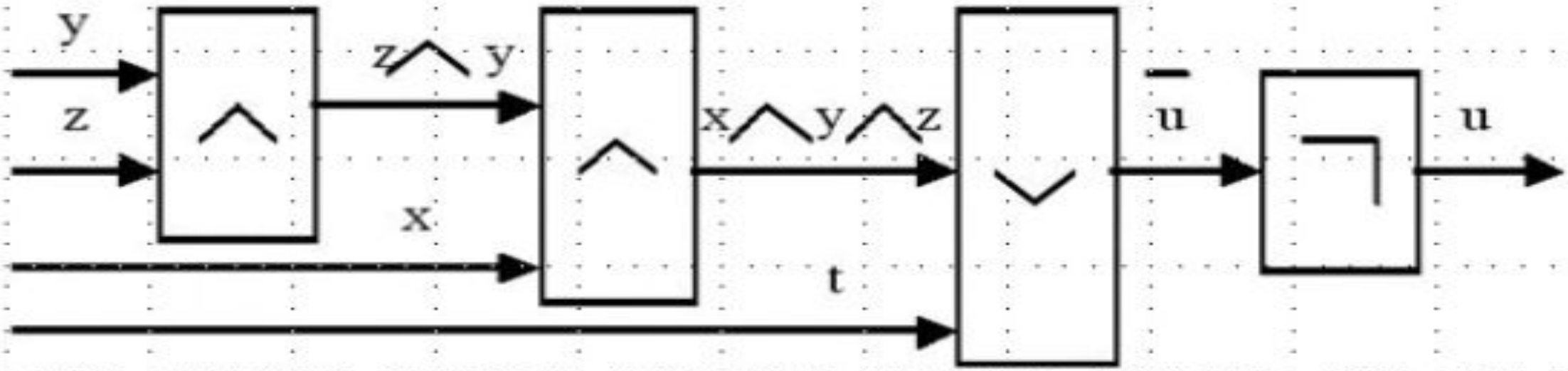
- Важной задачей информатики является минимизация числа *вентилей* для реализации той или иной схемы (устройства), что необходимо для более рационального, эффективного воплощения этих схем, для большей производительности и меньшей стоимости ЭВМ.
- Эту задачу решают с помощью методов теоретической информатики (методов булевой алгебры).

Пример

Построим схему для логической функции

$$u = \overline{x \wedge y \wedge z \vee t}$$

Схема, построенная для этой логической функции



Изображения 5 основных вентилей

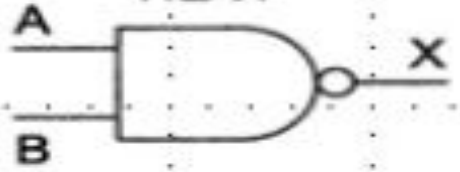
Для реализации любой логической операции над двоичными сигналами достаточно элементов трех типов: **И, ИЛИ, НЕ**. Однако, существуют микросхемы, реализующие более сложные логические функции: **НЕ-И** (операция Шеффера) и **НЕ-ИЛИ** (стрелка Пирса).

НЕ

A	X
0	1
1	0

а

НЕ-И



A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

б

НЕ-ИЛИ

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

в

И

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

г

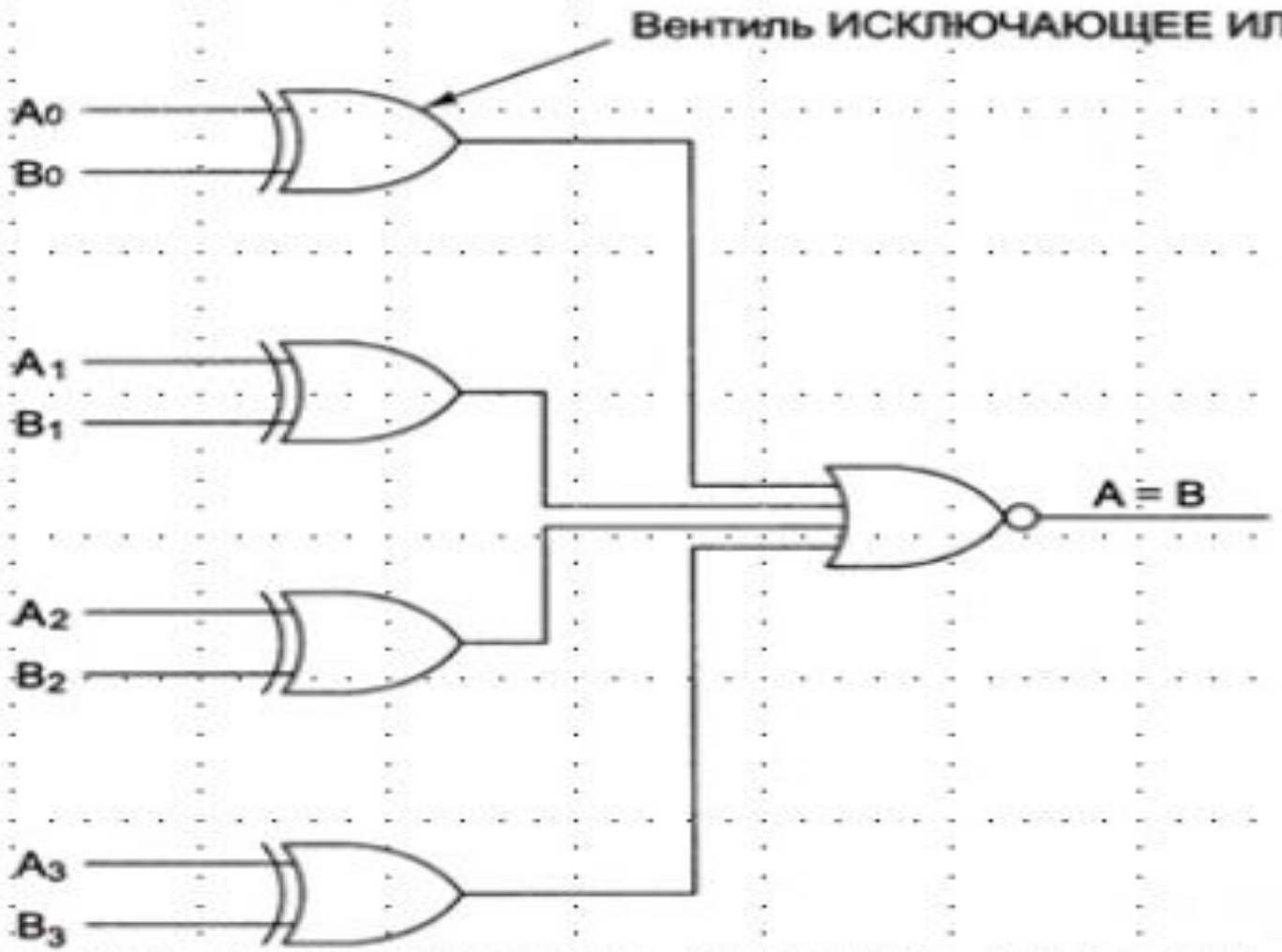
ИЛИ

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

д

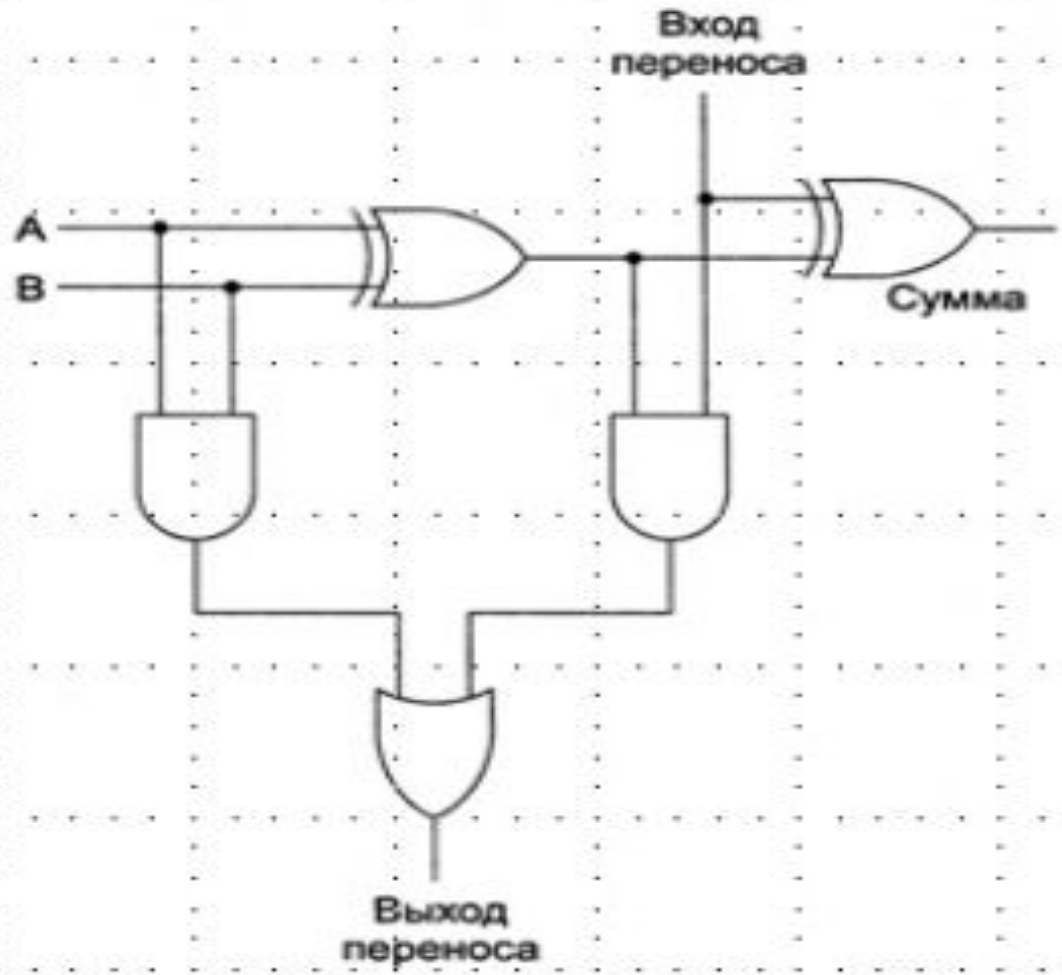
Пример арифметической схемы: компараторы

- Компаратор сравнивает два слова, поступающие на вход.
- Используются вентиля – ИСКЛЮЧАЮЩЕЕ ИЛИ, который дает 0, если сигналы на входе равны, и 1 в противном случае



Пример арифметической схемы: полный сумматор

A	B	Вход переноса	Сумма	Выход переноса
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



- Вычисляется бит суммы и бит переноса в зависимости от трех входов.