

АРХИТЕКТУРА ЭВМ И ВС ТЕМА 1.4., 3.10,11,12

Преподаватель:
Шершова Л.Н.

Тема 1.4. Логические основы ЭВМ, элементы и узлы

Занятие 10. Базовые логические операции и схемы. Таблицы истинности. Схемные логические элементы и узлы ЭВМ и их классификация: вентили, триггеры, регистры, полусумматоры, сумматоры, шифраторы, дешифраторы. Таблицы истинности RS-, JK- и T-триггера.

Занятие 11. Работа логических узлов ЭВМ. Анализ схемы регистра, как основного логического узла ЭВМ. Режимы работы регистра.

Занятие 12. Работа и особенности логических элементов ЭВМ. Построение элементарных логических схем: триггеры, сумматоры, шифраторы и дешифраторы.

Логические основы устройства компьютера

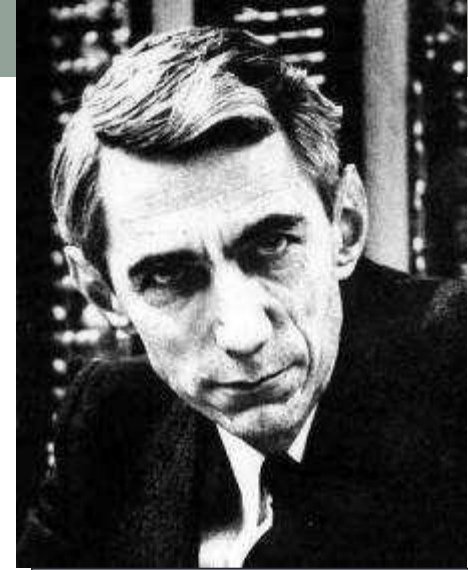
Двоичная система оказалась удобной в качестве языка логики. Это поняли спустя 100 лет после работ Буля.



(1839 - 1914)

С 1886 г. американский логик **Чарльз Сандерс Пирс** (в честь его названа логическая операция – «стрелка Пирса») работает над модификацией и расширением булевой алгебры. Пирс первый осознал, что бинарная логика имеет сходство с работой электрических переключательных схем. Электрический переключатель либо пропускает ток (истина), либо не пропускает (ложь). Пирс даже придумал простую электрическую логическую схему, но так и не собрал ее.

Американец **Клод Шеннон** — основоположник теории информации, разработчик теоретических основ вычислительной техники, математик и специалист по электронике *раскрыл связи между двоичным кодированием, алгеброй логики и электрическими схемами (релейными), т.е. наполнил логические выражения физическим смыслом, создал алгебру релейных схем, на которой основана теория бесконтактных логических элементов.*



1916 – 2001гг.

Принципы работы вычислительных машин в своей основе просты.

Работа ЭВМ состоит в операциях над числами и символами, закодированными двумя цифрами – 0 и 1, и пересылке этой информации по линиям связи. А работа всех устройств ЭВМ заключается в операциях над этими последовательностями из 0 и 1

Средством обработки двоичных сигналов в ЭВМ являются логические элементы.

Для реализации **любых логических операций** над двоичными сигналами достаточно элементов **трех типов** – элементов, реализующих три основные логические операции: **И, ИЛИ, НЕ.**

Логические элементы - это электронные схемы с одним или несколькими входами и одним выходом, через которые проходят электрические сигналы, представляющие цифры 0 и 1.

Логические схемы - схемы, состоящие из логических элементов. Они используются при построении вычислительных устройств для обеспечения выполнения арифметических, логических и других операций над машинными словами.

В основе метода построения и анализа **логических схем** лежит аппарат **алгебры логики.**

Вентили

В основе построения компьютеров, а точнее аппаратного обеспечения, лежат так называемые **вентили**. Они представляют собой достаточно простые элементы, которые можно комбинировать между собой, создавая тем самым различные схемы. Одни схемы подходят для осуществления арифметических операций, а на основе других строят различную память ЭВМ.

Простейший вентиль представляет собой транзисторный инвертор, который преобразует низкое напряжение в высокое или наоборот (высокое в низкое). Это можно представить как преобразование логического нуля в логическую единицу или наоборот. Т.е. получаем **вентиль НЕ**.

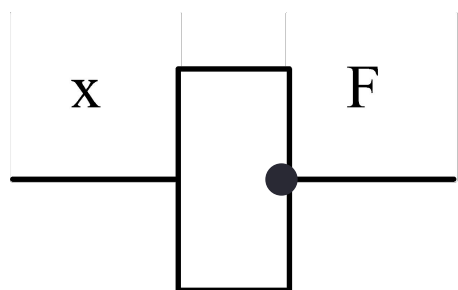
Соединив пару транзисторов различным способом, получают **вентили ИЛИ-НЕ** и **И-НЕ**. Эти **вентили** принимают уже не один, а два и более входных сигнала. Выходной сигнал всегда один и зависит (выдает высокое или низкое напряжение) от входных сигналов. В случае **вентиля ИЛИ-НЕ** получить высокое напряжение (логическую единицу) можно только при условии низкого напряжения на всех входах. В случае **вентиля И-НЕ** все наоборот: логическая единица получается, если все входные сигналы будут нулевыми. Как видно, это обратно таким привычным логическим операциям как **И** и **ИЛИ**. Однако обычно используются **вентили И-НЕ** и **ИЛИ-НЕ**, т.к. их реализация проще: **И-НЕ** и **ИЛИ-НЕ** реализуются двумя транзисторами, тогда как логические **И** и **ИЛИ** тремя.

Выходной сигнал вентиля можно выражать как функцию от входных.

Транзистору требуется очень мало времени для переключения из одного состояния в другое (время переключения оценивается в наносекундах). И в этом одно из существенных преимуществ схем, построенных на их основе.

Основные логические элементы (вентили):

1. Элемент НЕ (инвертор) Функция: Таблица истинности:

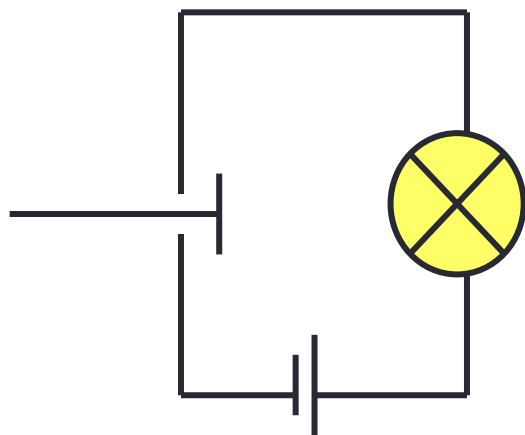


$$F = \text{не } X$$

$$F = \bar{x}$$

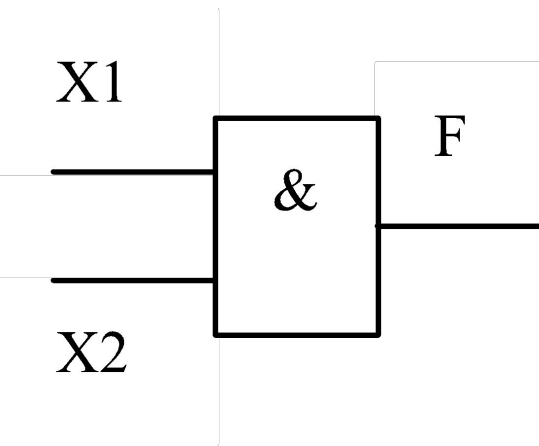
X	F
0	1
1	0

У инвертора один вход и один выход. Сигнал на выходе **F** появится тогда, когда на входе его нет, и наоборот.



Лампочка горит, если выключатель не включен

2. Элемент И Конъюнктор (логическое умножение)



Функция:

$$F = x1 \text{ и } x2$$

$$F = x1 \wedge x2$$

$$F = x1 \cdot x2$$

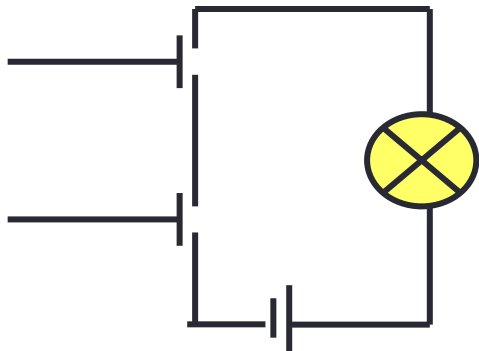
$$F = x1 \& x2$$

Таблица ИСТИННОСТИ:

x1	x2	F
0	0	0
0	1	0
1	0	0
1	1	1

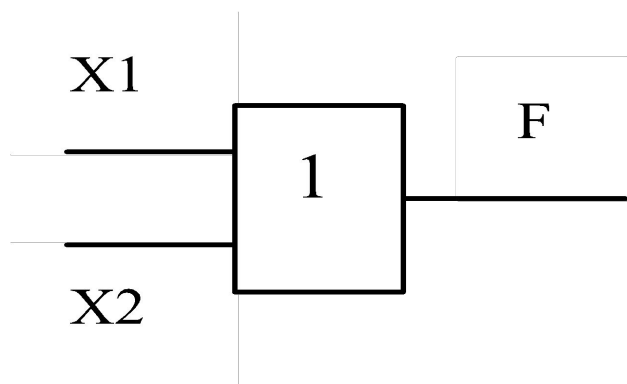
Элемент **И** имеет не менее двух входов и один выход. X1, X2 - входные сигналы, F – выходной сигнал.

Логика элемента И заключается в том, что на его выходе F будет сформирован сигнал 1 тогда и только тогда, когда на каждом из его входов будет сигнал 1.



Лампочка горит тогда и только тогда, когда включены оба выключателя

3.Элемент ИЛИ Дизъюнктор (логическое сложение)



Функция:

$$F = x1 \text{ или } x2$$

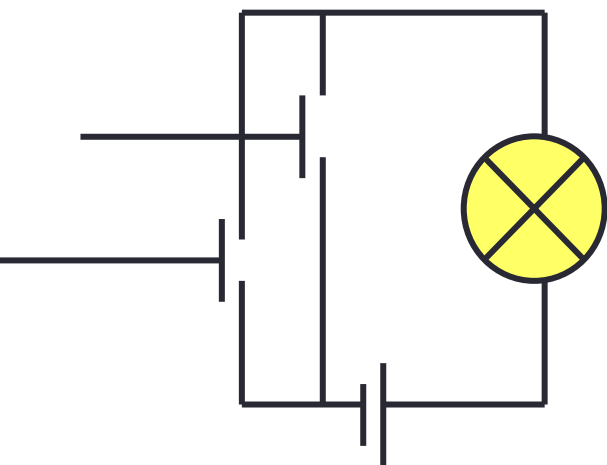
$$F = x1 \vee x2$$

$$F = x1 + x2$$

Таблица истинности:

x1	x2	F
0	0	0
0	1	1
1	0	1
1	1	1

Имеет не менее двух входов и один выход. Сигнал 0 на выходе F элемента ИЛИ появится только в том случае, если сигнал 1 не поступил ни на один из входов.



Лампочка горит, если включен хотя бы один выключатель

Итак, компьютер выполняет арифметические и логические операции при помощи т.н. *базовых логических элементов*, которые также еще называют вентиля

- Вентиль «И» – конъюнктор.
Реализует конъюнкцию.
- Вентиль «ИЛИ» – дизъюнктор.
Реализует дизъюнкцию.
- Вентиль «НЕ» – инвертор.
Реализует инверсию.
- Любая логическая операция может быть представлена через конъюнкцию, дизъюнкцию и инверсию.
- Любой сколь угодно сложный элемент компьютера может быть сконструирован из элементарных вентиляей.

Сигналы-аргументы и сигналы-функции

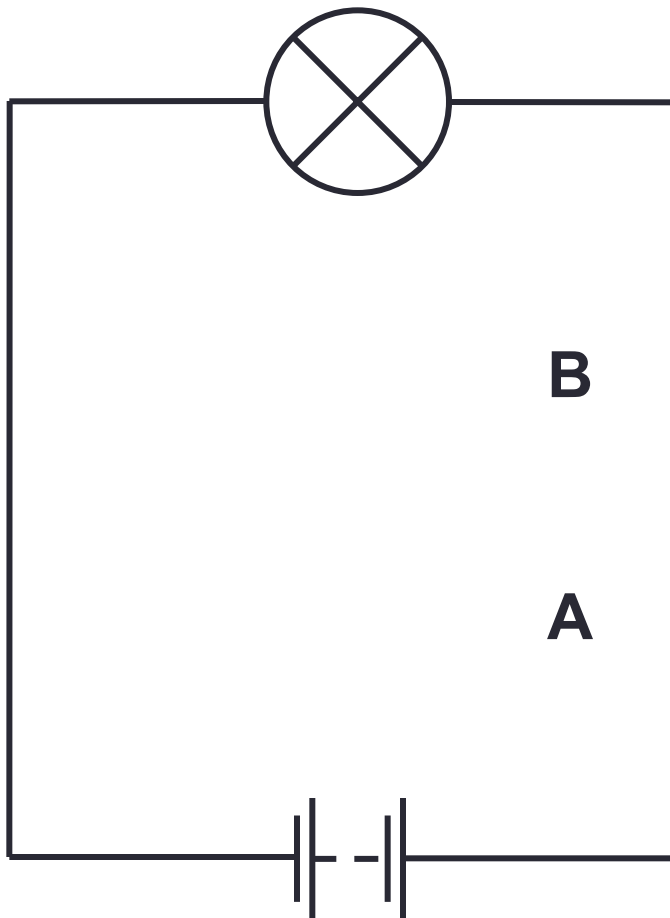
Вентили оперируют с электрическими импульсами:

- Импульс имеется –
логический смысл сигнала «1»
- Импульса нет –
логический смысл сигнала «0»

На входы вентиля подаются импульсы – значения аргументов, на выходе вентиля появляется сигнал – значение функции

Логическая схема типа «И» (конъюнктор)

$$0 \wedge 0 = 0$$

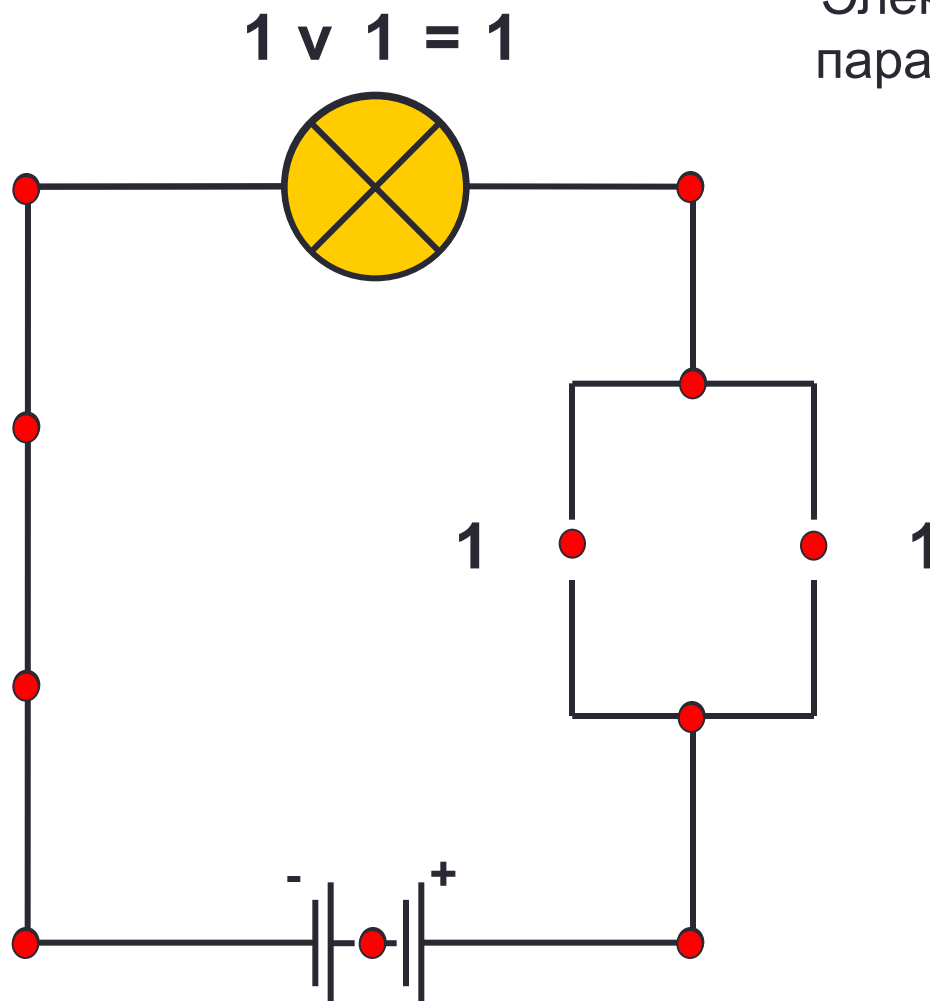


Электрическая цепь из двух
последовательно подключенных
выключателей

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

Логическая схема типа «ИЛИ» (дизъюнктор)

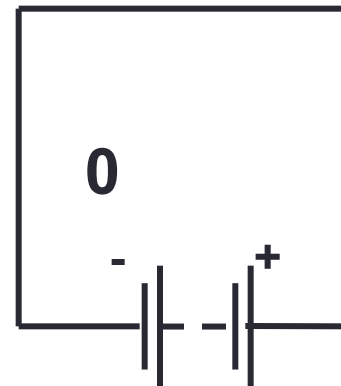
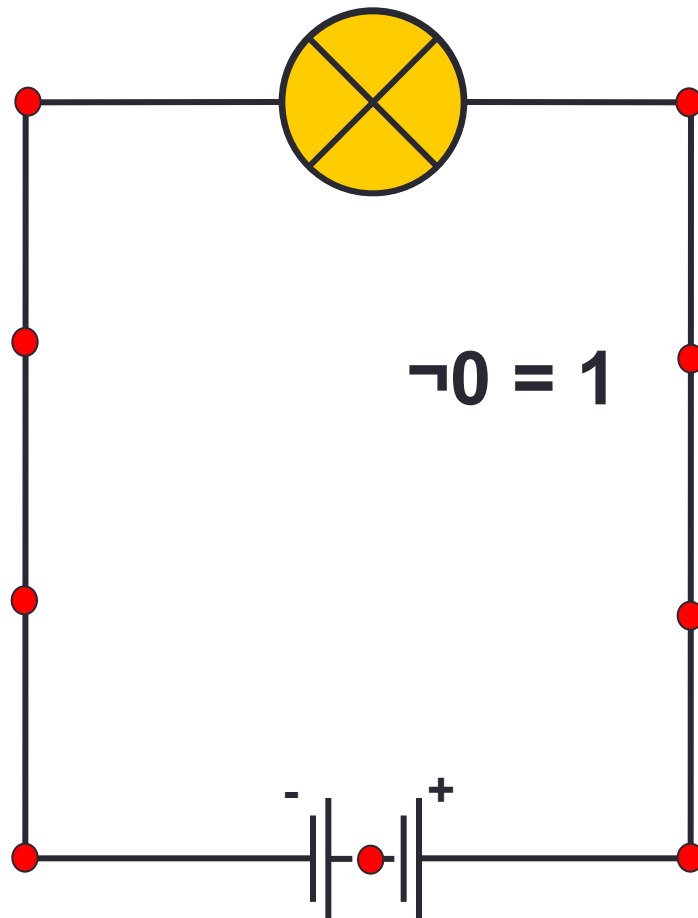
Электрическая цепь из двух
параллельно подключенных
выключателей



A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

Логическая схема типа «НЕ» (инвертор)

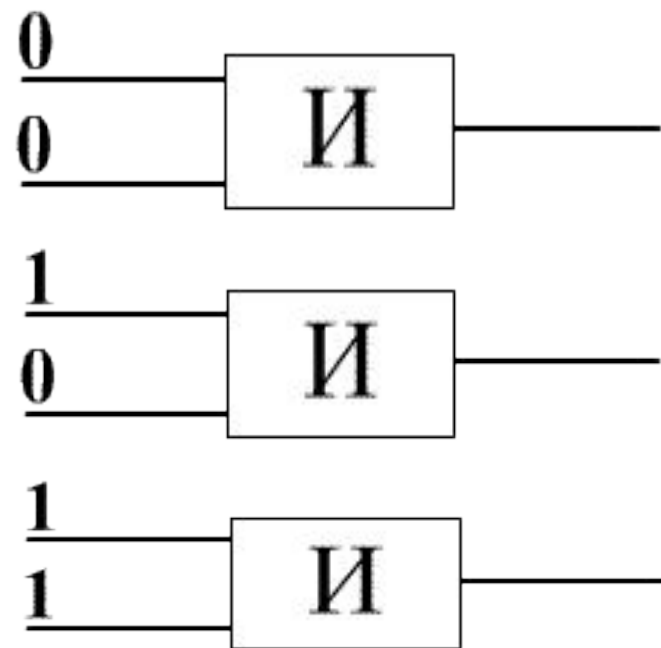
Электрическая цепь с одним
автоматическим выключателем



A	$\neg A$
0	1
1	0

Конъюнктор

- На входы конъюнктора подаются сигналы **0** или **1**
- На выходе конъюнктора появляются сигналы **0** или **1** в соответствии с таблицей ИСТИННОСТИ



Дизъюнктор

- На входы дизъюнктора подаются сигналы **0** или **1**
- На выходе дизъюнктора появляются сигналы **0** или **1** в соответствии с таблицей ИСТИННОСТИ



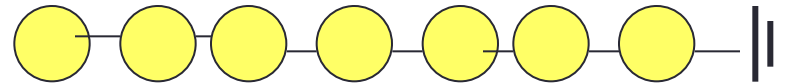
Инвертор

- На входы инвертора подаются сигналы **0** или **1**
- На выходе инвертора появляются сигналы **1** или **0** в соответствии с таблицей ИСТИННОСТИ



Примеры:

1. В старых елочных гирляндах лампочки включались последовательно. Гирлянда работала тогда и только тогда, когда все лампочки были исправны. На какую логическую операцию это похоже?



Логическое умножение: $F=A \& B \& C \& D$

2. В современных гирляндах лампочки подключены параллельно. На какую логическую операцию это похоже?

Логическое сложение: $F=A \vee B \vee C \vee D$

3. Выключатель. Если свет не горел, то его включают, если горел – выключают. **Инверсия**

В роли “элементарной частицы” в ЭВМ всегда выступает разновидность выключателя. И если правильно соединить очень много выключателей и поставить очень много людей, которые будут ими щелкать в нужный момент, то получится вычислительная машина.

С помощью логических элементов **НЕ, И, ИЛИ**

можно реализовать (собрать как из конструктора) типовые функциональные узлы (блоки) ЭВМ:

- ❖ *триггеры*
- ❖ *регистры*
- ❖ *полусумматоры*
- ❖ *сумматоры*
- ❖ *шифраторы*
- ❖ *дешифраторы*

Чтобы понять, как работает интересующее нас устройство, необходимо понять логику его работы, т.е. найти соответствие между входными и выходными сигналами, **для этого:**

- ✓ **составить таблицу истинности**
- ✓ **по таблице записать логическую функцию**
- ✓ **построить логическую схему**

ПОСТРОЕНИЕ ЛОГИЧЕСКИХ ФУНКЦИЙ И СХЕМ

ПО ЗАДАННОЙ ТАБЛИЦЕ :

- I. Выписывается **таблица истинности функции**.
- II. По данной таблице определяется **логическая функция** (формула) с помощью следующего метода, называемого **дизъюнктивная совершенная нормальная форма (ДСНФ)**:
 1. В заданной таблице выбираются наборы переменных, при которых значения функции равно 1.
 2. Для каждого такого набора записываются конъюнкции (\wedge) всех входных переменных, имеющие значение 1. При этом те переменные, которые имеют значение 0, записываются с отрицанием.
 3. Все полученные конъюнкции объединяются знаками дизъюнкции (\vee). Это и будет искомая логическая функция, которую можно будет упростить (минимизировать) по законам Булевой алгебры.
- III. По упрощенной логической функции строится **логическая схема**

Пример. По заданной таблице истинности записать логическую функцию, упростить ее и построить логическую схему.

x	y	F
0	0	1
0	1	1
1	0	1
1	1	0

$$\begin{aligned} &\bar{x} \wedge \bar{y} \\ &\bar{x} \wedge y \\ &x \wedge \bar{y} \end{aligned}$$

1. Запишем **конъюнкцию** для каждой строки, где значение функции = **1**. Переменные, значения которых равны **0**, запишем с **отрицанием**.

2. Объединив полученные конъюнкции дизъюнкцией, получим следующую логическую функцию.

$$F = (\bar{X} \wedge \bar{Y}) \vee (\bar{X} \wedge Y) \vee (X \wedge \bar{Y})$$

3. Упростим: $F = (\bar{X} \wedge \bar{Y}) \vee (\bar{X} \wedge Y) \vee (X \wedge \bar{Y}) = \bar{X} \vee (X \wedge \bar{Y})$

4. По полученной функции построим логическую схему:

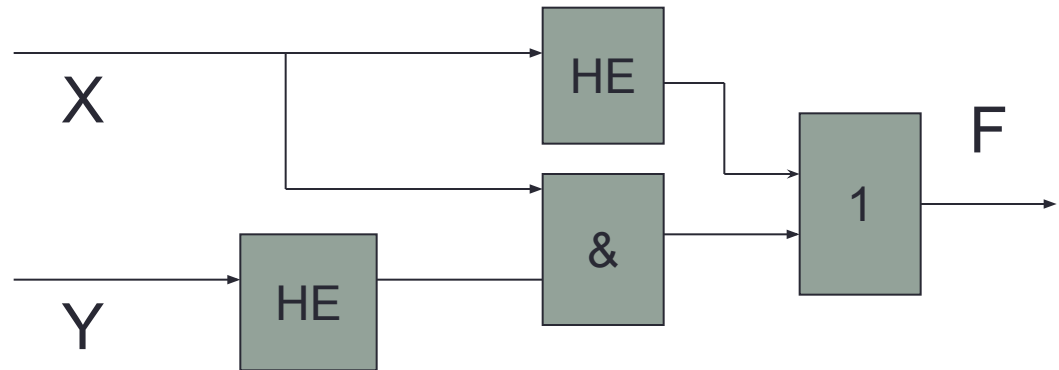
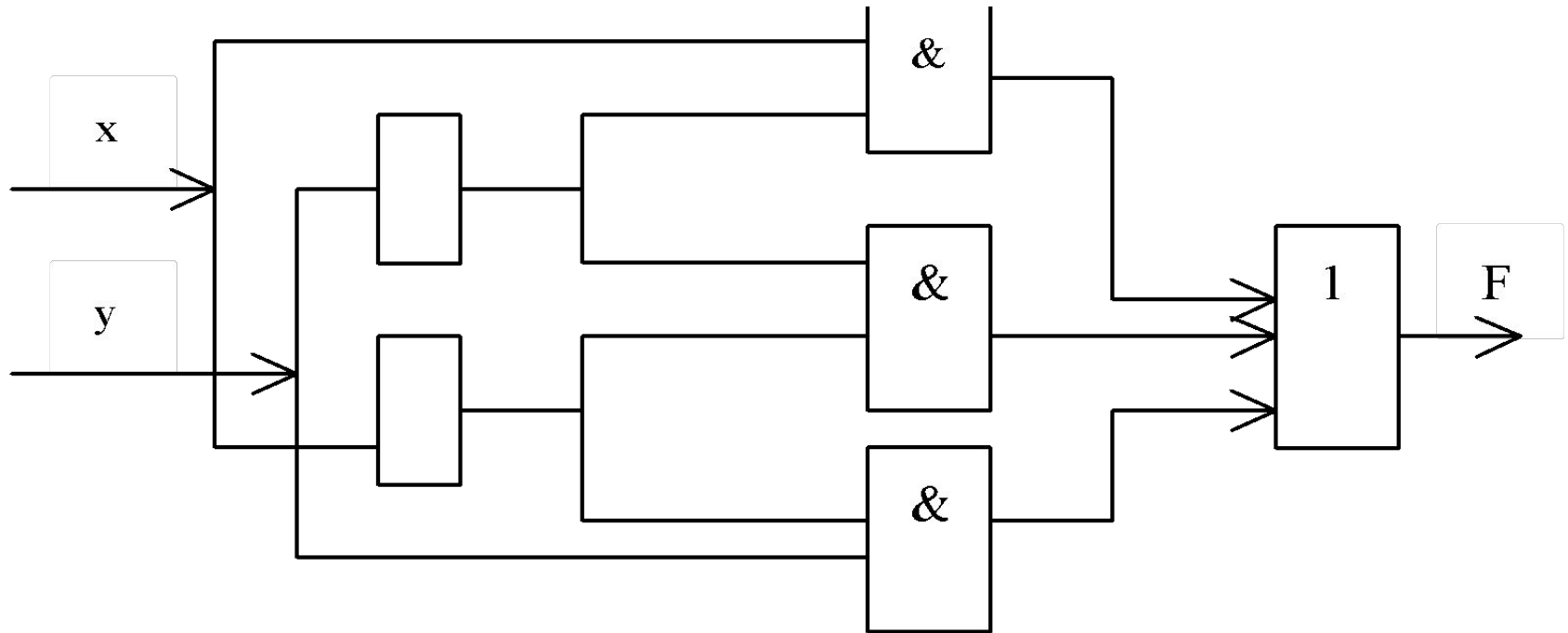


Схема по не упрощенной логической функции

$$F = (\bar{X} \wedge \bar{Y}) \vee (\bar{X} \wedge Y) \vee (X \wedge \bar{Y})$$



3. Составить схему, работа которой задана таблицей истинности:

A	B	C	F(A,B,C)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

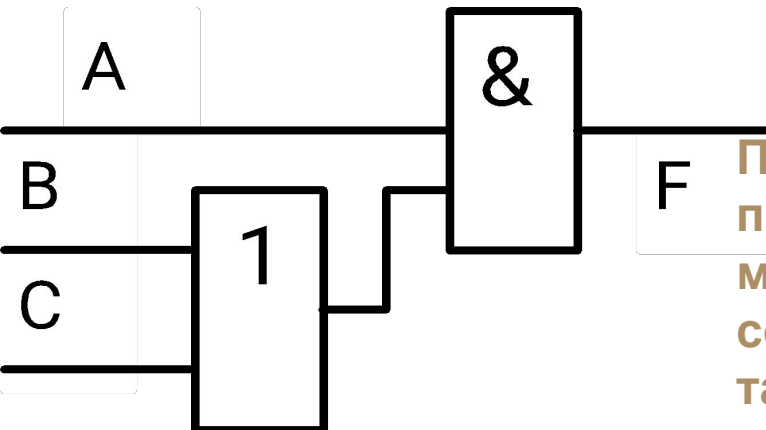
а) Составим логическую формулу схемы:

$$F = (A \wedge \bar{B} \wedge C) \vee (A \wedge B \wedge \bar{C}) \vee (A \wedge B \wedge C)$$

б) Упростим полученную формулу:

$$\begin{aligned} F &= (A \wedge \bar{B} \wedge C) \vee (A \wedge B \wedge \bar{C}) \vee (A \wedge B \wedge C) = \\ &= (A \wedge \bar{B} \wedge C) \vee (A \wedge B) \vee (C \wedge \bar{C}) = (A \wedge \bar{B} \wedge C) \vee (A \wedge B) = \\ &= A \wedge (B \vee (\bar{B} \wedge C)) = A \wedge (B \vee C) \end{aligned}$$

в) по упрощенной (минимизированной) функции составим логическую схему:



Правильность полученной формулы можно проверить сопоставлением таблиц истинности по последним столбцам.

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

ПОСТРОЕНИЕ ТАБЛИЦЫ ИСТИННОСТИ И ЛОГИЧЕСКОЙ ФУНКЦИИ

ПО ЗАДАННОЙ ЛОГИЧЕСКОЙ СХЕМЕ

Задание. Запишите логическую функцию, описывающую состояние схемы, составьте таблицу истинности:

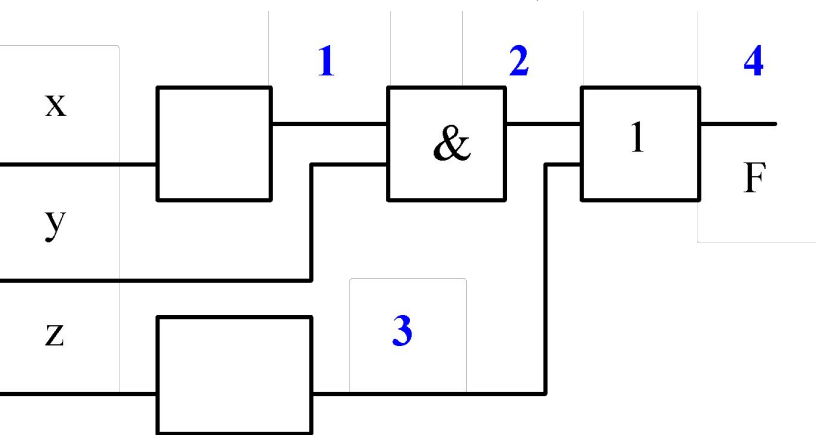


Таблица истинности:

x	y	z	\bar{x}	$\bar{x} \wedge y$	\bar{z}	$(\bar{x} \wedge y) \vee \bar{z}$
0	0	0	1	0	1	1
0	0	1	1	0	0	0
0	1	0	1	1	1	1
0	1	1	1	1	0	1
1	0	0	0	0	1	1
1	0	1	0	0	0	0
1	1	0	0	0	1	1
1	1	1	0	0	0	0

Для записи функции необходимо записать значения на выходе каждого элемента схемы:

1. \bar{x}
2. $\bar{x} \wedge y$
3. \bar{z}
4. $(\bar{x} \wedge y) \vee \bar{z}$

Следовательно получится функция: $F = (\bar{x} \wedge y) \vee \bar{z}$

Триггер

Память (устройство, предназначенное для хранения данных и команд) является важной частью компьютера. Можно сказать, что она его и определяет: если вычислительное устройство не имеет памяти, то оно уже не компьютер.

Элементарной единицей компьютерной памяти является бит. **Поэтому требуется устройство, способное находиться в двух состояниях, т.е. хранить единицу или ноль.** Также это устройство должно уметь быстро переключаться из одного состояния в другое под внешним воздействием, что дает возможность изменять информацию. Ну и наконец, устройство должно позволять определять его состояние, т.е. предоставлять во вне информацию о своем состоянии.

Устройством, способным запоминать, хранить и позволяющим считывать информацию, является триггер. Он был изобретен в начале XX века Бонч-Бруевичем.

Разнообразие триггеров весьма велико. Наиболее простой из них так называемый RS-триггер, который собирается из двух вентилях. Обычно используют вентили ИЛИ-НЕ или И-НЕ.

Триггер (*trigger* - защелка, спусковой крючок) —

запоминающее устройство (хранит 1 бит информации)

Триггер имеет

два входа:

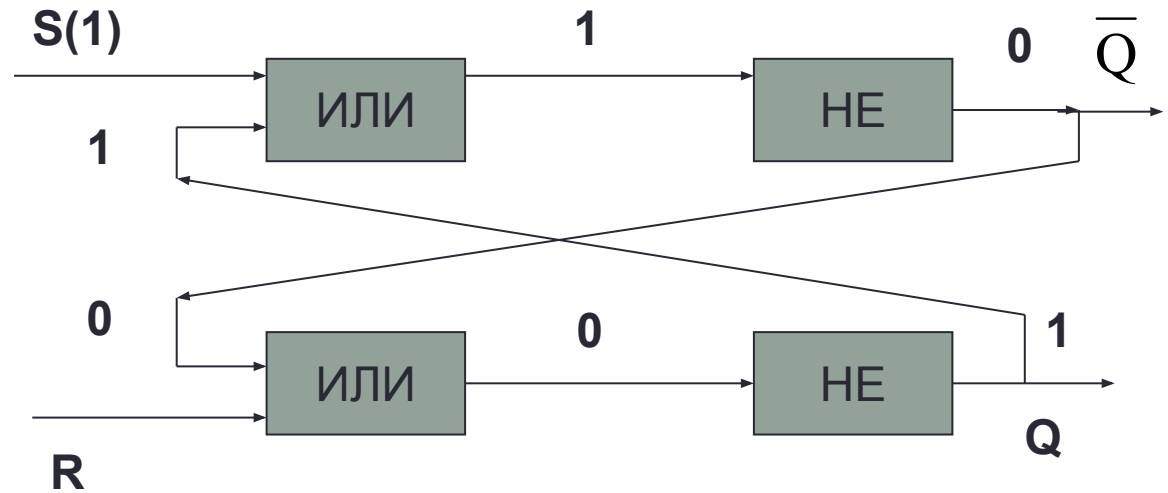
S (set – установка) и

R (reset – сброс) и

два выхода

Q (прямой) и

НЕQ (инверсный)



Входы		Состояние Q
S	R	
0	0	Q
1	0	1
0	1	0
1	1	Недопустимо

В обычном состоянии на входы триггера подан сигнал «0» и триггер хранит «0». Для записи «1» на вход S (set – установочный) подается сигнал «1». При последовательном рассмотрении прохождения сигнала по схеме видно, что триггер переходит в это состояние и будет устойчиво находиться в нем и после того, как сигнал на входе S исчезнет. Триггер запомнил «1», т.е. с выхода триггера Q можно считывать «1».

Чтобы сбросить информацию и подготовиться к приему новой, на вход R (сброс) подается сигнал «1», после чего триггер возвратится к исходному «нулевому» состоянию.

Регистр

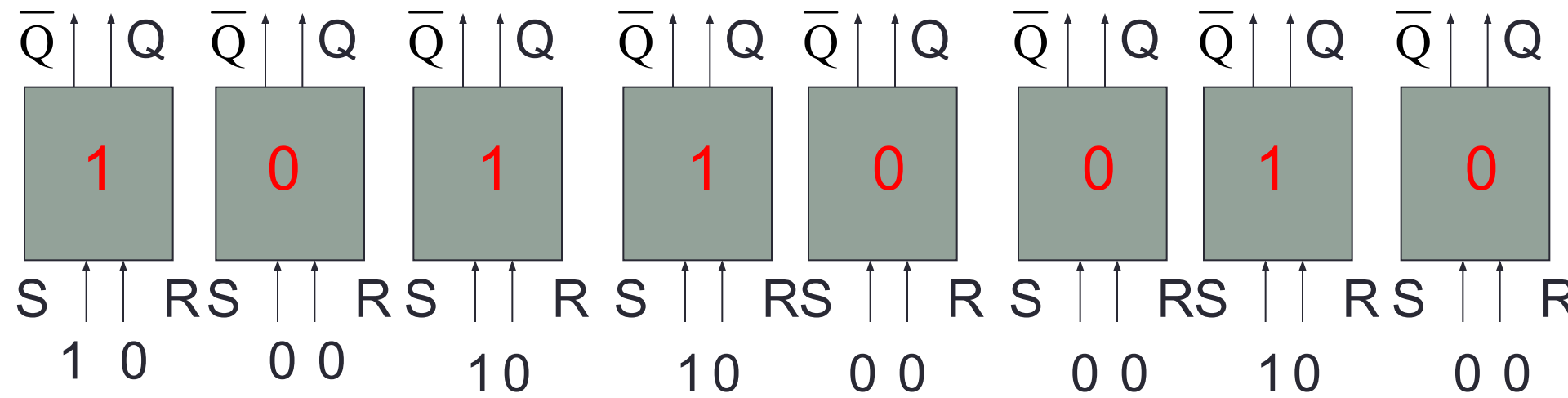
Регистр — последовательное или параллельное логическое устройство, используемое для хранения n -разрядных двоичных чисел и выполнения преобразований над ними.

Регистр представляет собой **упорядоченную последовательность триггеров**, обычно D , число которых соответствует числу разрядов в слове. С каждым регистром обычно связано комбинационное цифровое устройство, с помощью которого обеспечивается выполнение некоторых операций над словами.

Фактически любое цифровое устройство можно представить в виде совокупности регистров, соединённых друг с другом при помощи комбинационных цифровых устройств.

Основой построения регистров являются D -триггеры, RS -триггеры.

Регистр – устройство, состоящее из последовательности триггеров. Предназначен для хранения многоразрядного двоичного числового кода, которым может быть представлять и адрес, и команду, и данные



Число триггеров в регистре называется разрядностью компьютера, которая может быть равна 8, 16, **32**, **64**...

Задачи.

1. Сколько триггеров необходимо для хранения информации объемом:

1 байт - **8**

1 Кбайт - **8192**

1 Мбайт - **8388608**

Сумматор – устройство для сложения двоичных чисел. Сумматор – основа микропроцессора, т.к все операции в микропроцессоре сводятся к сложению.

Схемные логические элементы ЭВМ

Сумматоры и полусумматоры

Арифметико-логическое устройство процессора (АЛУ) обязательно содержит в своем составе такие элементы как сумматоры. **Эти схемы позволяют складывать двоичные числа.** Как происходит сложение? Допустим, требуется сложить двоичные числа 1001 и 0011. Сначала складываем младшие разряды (последние цифры): $1+1=10$. Т.е. в младшем разряде будет 0, а единица – это перенос в старший разряд. Далее: $0 + 1 + 1$ (от переноса) = 10, т.е. в данном разряде снова запишется 0, а единица уйдет в старший разряд. На третьем шаге: $0 + 0 + 1$ (от переноса) = 1. В итоге сумма равна 1100.

Полусумматор

Теперь не будем обращать внимание на перенос из предыдущего разряда и рассмотрим только, как формируется сумма текущего разряда. Если были даны две единицы или два нуля, то сумма текущего разряда равна 0. Если одно из двух слагаемых равно единице, то сумма равна единице. Получить такие результаты можно при использовании **вентиля ИСКЛЮЧАЮЩЕГО ИЛИ.**

Перенос единицы в следующий разряд происходит, если два слагаемых равны единице. И это реализуемо **вентилем И.**

Тогда сложение в пределах одного разряда (без учета возможной пришедшей единицы из младшего разряда) можно реализовать изображенной ниже схемой, которая называется **полусумматором.** У **полусумматора два входа** (для слагаемых) и **два выхода** (для суммы и переноса). На схеме изображен полусумматор, состоящий из **вентилей ИСКЛЮЧАЮЩЕЕ ИЛИ и И.**

Полусумматор – реализует суммирование одноразрядных двоичных чисел **без учета переноса из младшего разряда.**

Слагаемые		Перенос	Сумма
A	B	P	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

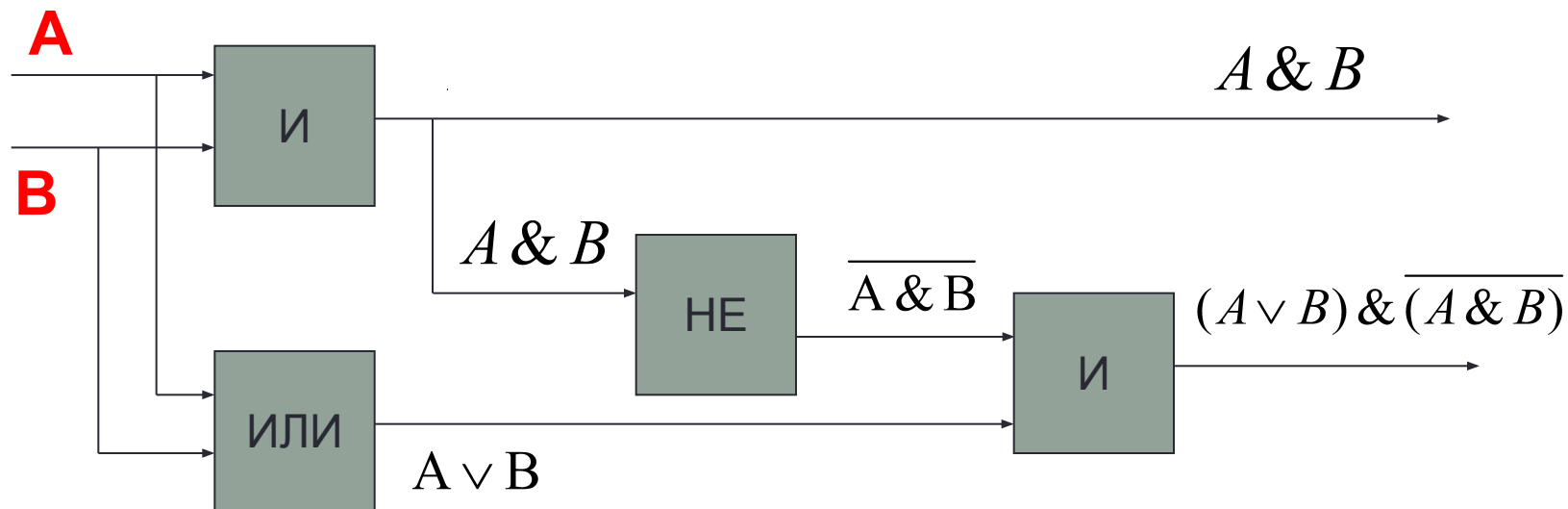
$P = A \& B$
 Значения S будут соответствовать сумме, если результат логического сложения умножить на инверсный перенос.
 Тогда

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

A	B	$A \vee B$	$A \& B$	$\overline{A \& B}$	$(A \vee B) \& \overline{(A \& B)}$
0	0	0	0	1	0
0	1	1	0	1	1
1	0	1	0	1	1
1	1	1	1	0	0

Схема полусумматора двоичных чисел:

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



Полусумматор. Арифметическое сложение двоичных чисел

В каждом разряде образуется сумма цифр в соответствующих разрядах слагаемых, при этом возможен перенос единицы в старший разряд

Без переноса

$$\begin{array}{r} + 0000 \ 0001 \\ 0000 \ 0010 \\ \hline 0000 \ 0011 \end{array}$$

С переносом

$$\begin{array}{r} + 0000 \ 0011 \\ 0000 \ 0010 \\ \hline 0000 \ 0101 \end{array}$$

Обозначим слагаемые через A и B ,
перенос – через P , а сумму – через S

Таблица сложения одноразрядных двоичных чисел:

Слагаемые		Перенос	Сумма
A	B	P	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Очевидно, что $P = A \wedge B$

Получаем формулу для вычисления S

Если сравнить $A \vee B$ с S :

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

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

то очевидно, что они практически идентичны.

Чтобы равенство оказалось полным нужно выражение $A \vee B$ умножить на $\neg P$

Получаем формулу для вычисления S

$$S = (A \vee B) \wedge \neg P \Rightarrow (A \vee B) \wedge \neg(A \wedge B)$$

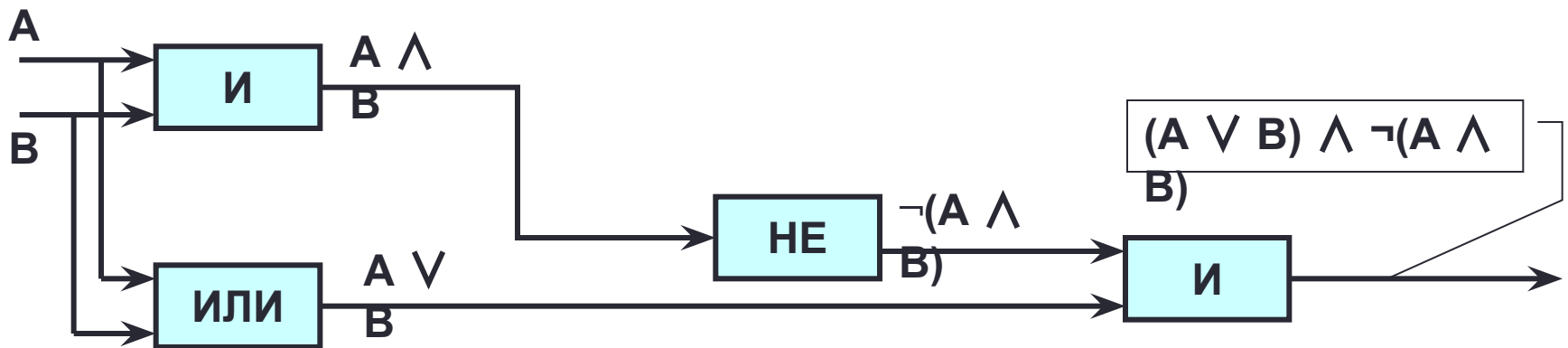
A	B	$A \vee B$	$A \wedge B$	$\neg(A \wedge B)$	$(A \vee B) \wedge \neg(A \wedge B)$
0	0	0	0	1	0
0	1	1	0	1	1
1	0	1	0	1	1
1	1	1	1	0	0

Теперь, имея элементарные логические выражения, можно построить логическую схему устройства для сложения одноразрядных двоичных чисел (полусумматора)

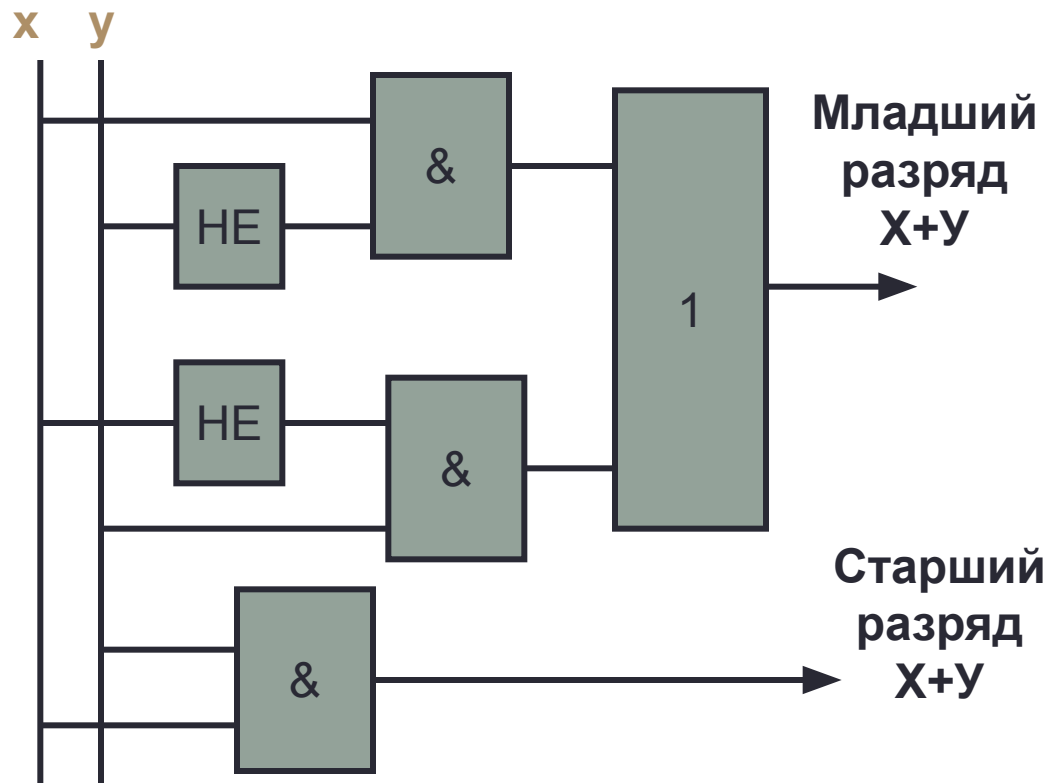
Логическая схема двоичного полусумматора

Полусумматор называется так, потому, что здесь не учитывается перенос единицы из младшего разряда

$$(A \vee B) \wedge \neg(A \wedge B)$$



Сумматор для двух одноразрядных чисел



x	y	Старший разряд	Младший разряд
0	0		
1	0		
0	1		
1	1		

Подавая на входы x и y сигналы 0 и 1, на выходах получим два сигнала, которые поразрядно кодируют сумму двух однозначных чисел.

А т.к. действия над числами, записанными в позиционной системе счисления, выполняются поразрядно, то ясно, что аналогичным образом можно построить электронные схемы для сложения многозначных чисел, представленных в двоичной системе счисления

Полный одноразрядный сумматор

Должен иметь три входа (A, B и P_0) и два выхода (S и P)

Слагаемые		Переносы		Сумма
A	B	P_0	P	S
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1

Формула полного одноразрядного сумматора

- **P** принимает значение **1** когда хотя бы две из трех переменных равны **1**:

$$P = (A \wedge B) \vee (A \wedge P_0) \vee (B \wedge P_0)$$

- Сумма равна произведению логического сложения (**A**, **B** и **P₀**) на инвертированный перенос $\neg P$:

$$S = (A \vee B \vee P_0) \wedge \neg P$$

- Это выражение справедливо во всех случаях, кроме одного, когда **A**, **B** и **P₀** равны **1**:

Формула полного одноразрядного сумматора

Правильное значение суммы – 1. Для ее получения необходимо полученное выражение сложить с произведением этих же переменных:

$$S = (A \vee B \vee P_0) \wedge \neg P \vee (A \wedge B \wedge P_0)$$

Многоразрядный сумматор

Построен на основе полных одноразрядных сумматоров (по одному на каждый разряд), причем таким образом, чтобы выход (перенос) младшего сумматора был подключен ко входу старшего сумматора

Контрольные вопросы:

1. Логические элементы – это...
2. Логические схемы – это...
3. Вентиль – это...
4. Виды вентиляей, их название, обозначение, логическая схема и таблица истинности.
5. Для реализации любых логических операций над двоичными сигналами каких логических элементов достаточно?
6. Можно ли сконструировать любой сколь угодно сложный элемент компьютера из элементарных вентиляей?
7. Какой сигнал подается на вход вентиля и что появляется на выходе?
8. Перечислите типовые функциональные узлы (блоки) ЭВМ
9. Чтобы понять логику работы узла ЭВМ надо найти соответствие между входным и выходным сигналом, а для этого рекомендуется ... 1)...., 2)...., 3).....
0. Приведите пример построения логической схемы по заданной таблице истинности.

Контрольные вопросы:

1. Что такое триггер? Приведите схему RS-триггера.
2. Что за устройство регистр? Проведите анализ схемы регистра, как основного логического узла ЭВМ (воспользуйтесь дополнительными источниками информации).
3. Что позволяют делать сумматоры и полусумматоры? В чем их отличие?
4. Приведите таблицу истинности и логическую схему полусумматора двоичных чисел.
5. Приведите формулу полного одноразрядного сумматора.
6. Все математические действия в компьютере сводятся к ...?
7. Сумматоры составляют основу....?
8. Составьте таблицы истинности JK- и T-триггера (воспользуйтесь дополнительными источниками информации).
9. Для чего предназначены шифраторы и дешифраторы? Приведите логические схемы этих устройств (воспользуйтесь дополнительными источниками информации).

Рекомендуемая литература

Рекомендуемая основная литература

1. Баула В.Г., Томилин А.Н., Волканов Д.Ю. Архитектура ЭВМ и операционные среды. – М.: Академия, 2012. – 336 с.
2. Жмакин А.П. Архитектура ЭВМ (+ CD-ROM). – СПб.: БХВ-Петербург, 2010. – 352 с.
3. Кузин А.В., Пескова С.А. Архитектура ЭВМ и вычислительных систем. – М.: Инфра-М, 2010. – 282 с.
4. Максимов Н.В., Попов И.И., Партыка Т.Л. Архитектура ЭВМ и вычислительных систем. – М.: Форум, 2012. – 512 с.
5. Новожилов О.П. Архитектура ЭВМ и систем. – М.:Юрайт, 2012. – 528 с.
6. Орлов С.А., Цилькер Б.Я. Организация ЭВМ и систем. – СПб.: Питер, 2011. – 688с.

Рекомендуемая дополнительная литература

1. Бигелоу Стивен Устройство и ремонт персонального компьютера. Аппаратная платформа и основные компоненты. – М.: Бином-Пресс, 2010. – 976 с.
2. Бройдо В.Л., Ильина О.П. Вычислительные системы, сети и телекоммуникации. – СПб.: Питер, 2009. – 766 с.
3. Горнец Н.Н., Роцин А.Г. ЭВМ и периферийные устройства. Компьютеры и вычислительные системы. – М.: Академия, 2012. – 240 с.
4. Горнец Н.Н., Роцин А.Г., Соломенцев В.В. Организация ЭВМ и систем. – М.: Академия, 2008. – 320 с.
5. Старков В.В. Архитектура персонального компьютера: Организация, устройство, работа. – М.: Телеком Сохранность, 2009. – 536 с.
6. Таненбаум Э. Архитектура компьютера. – СПб.: Питер, 2009. -844 с.
7. Ташков П. Компьютер: энциклопедия. – СПб.: Питер, 2009. -670 с.
8. Ресурсы Интернет.