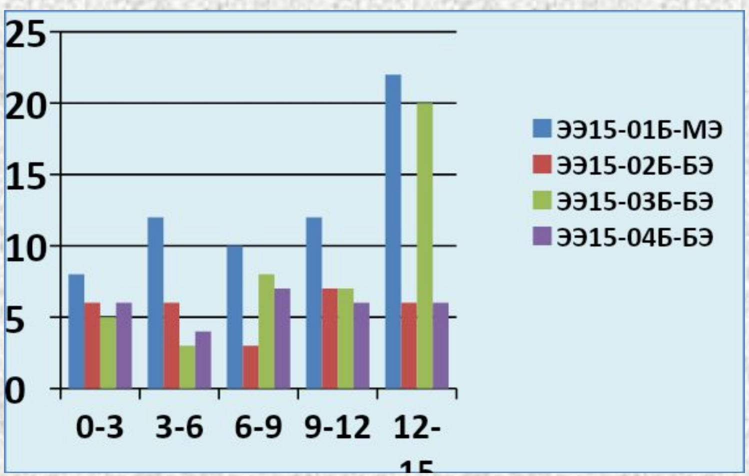
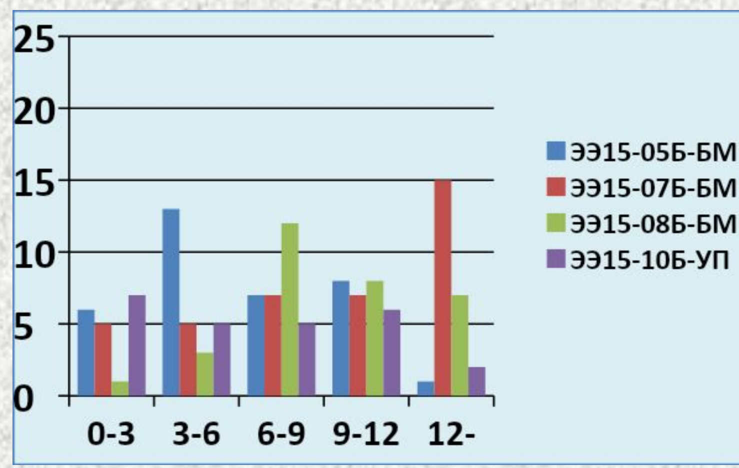
Распределение результатов контрольных работ с учетом всех попыток

Контрольное задание 1:

06.10.2015 - 10.11.2015 Помехоустойчивое кодирование. Решение логических

Интервалы оценки , баллы	Количество результатов испытаний (по группам и по всему списку студентов)							
	ЭЭ15-01Б-МЭ	ЭЭ15-02Б-БЭ	ЭЭ15-03Б-БЭ	ЭЭ15-04Б-БЭ	ЭЭ15-05Б-БМ	ЭЭ15-07Б-БМ	ЭЭ15-08Б-БМ	ЭЭ15-10Б-УП	Всего, по общему списку
0-3	8	6	5	6	6	5	1	7	48
3-6	12	6	3	4	13	5	3	5	51
6-9	10	3	8	7	7	7	12	5	62
9-12	12	7	7	6	8	7	8	6	61
12-15	22	6	20	6	1	15	7	2	79



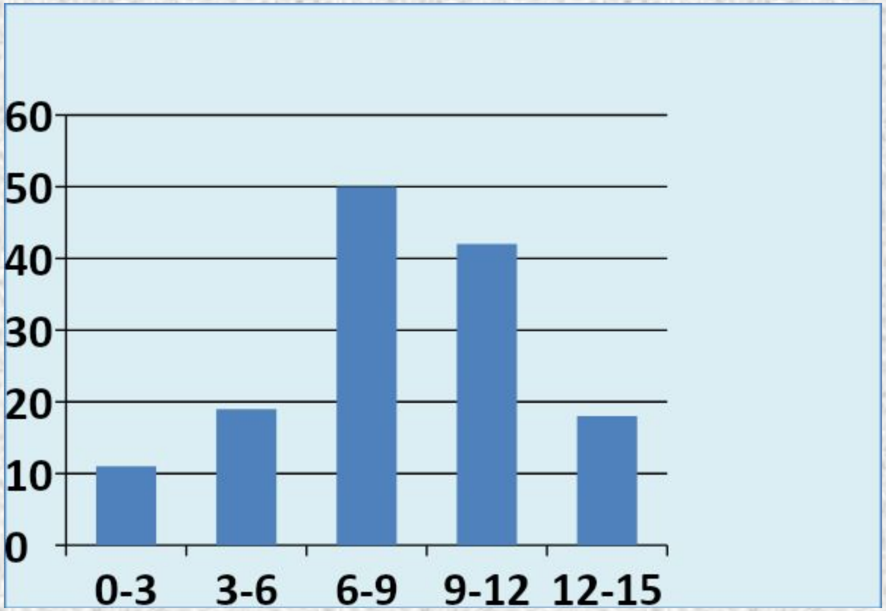


Распределение зачетных результатов контрольных работ

Контрольное задание 1:

06.10.2015 - 10.11.2015 Помехоустойчивое кодирование. Решение логических задач

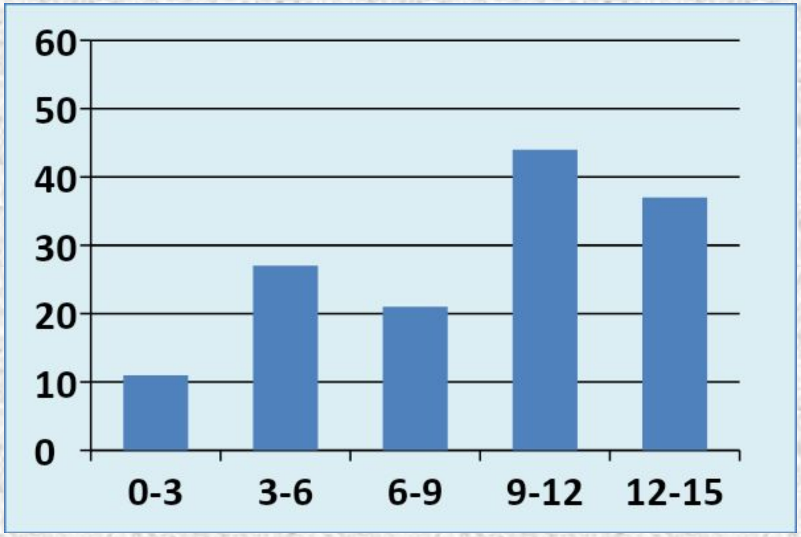
Названия строк	Количество по полю «Среднее по полю Оценка/15,00»
0-3	11
3-6	19
6-9	50
9-12	42
12-15	18
Общий итог	140



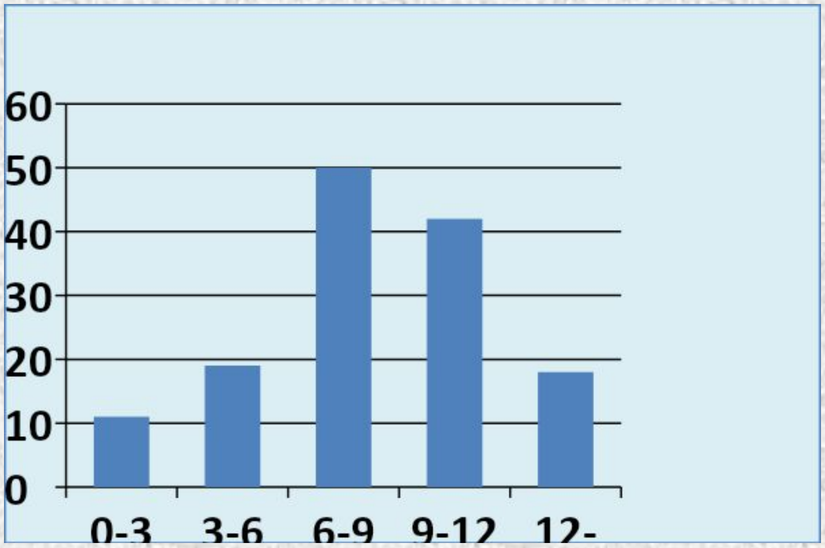
Распределение зачетных результатов контрольных работ

Контрольное задание 2:
11.10.2015 - 16.10.2015 Основы математической логики. Таблицы истинности

Названия строк	Количество по полю «Среднее по полю Оценка/15,00»
0-3	11
3-6	27
6-9	21
9-12	44
12-15	37
Общий итог	140

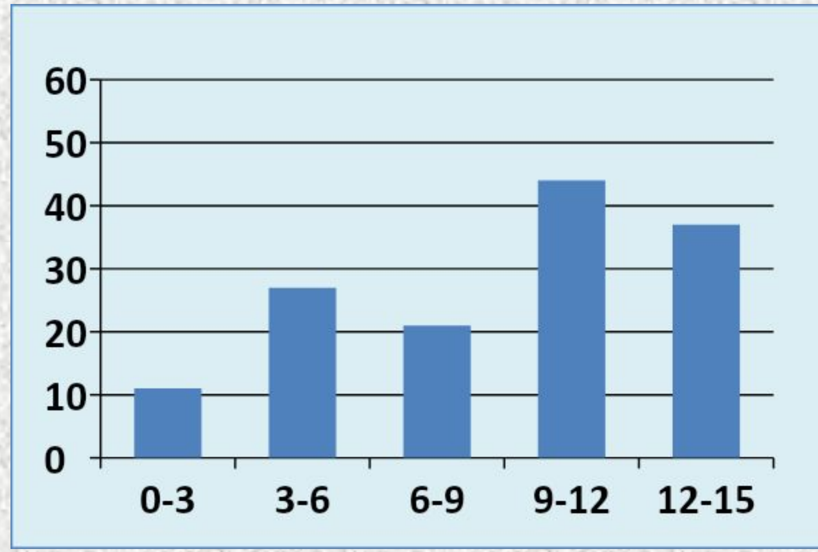


Распределение зачетных результатов контрольных работ



06.10.2015 - 10.11.2015

Помехоустойчивое кодирование. Решение логических задач



11.10.2015 - 16.10.2015

Основы математической логики. Таблицы истинности

Использование двоичной системы представления данных

Принцип программного управления

Принцип однородности памяти

Принцип хранимой программы

Принцип адресности

Основные принципы построения архитектуры ЭВМ

Логические основы построения и работы ЭВМ

Таблицы истинности

Логические функции

Аксиомы алгебры логики

Элементарные логические операции

Логические элементы компьютера, реализующие элементарные логические функции (И,ИЛИ, НЕ, ИЛИ-НЕ, И-НЕ).
Электронные схемы (сумматор, триггер).

Базовые логические элементы ЭВМ

Основы алгебры логики

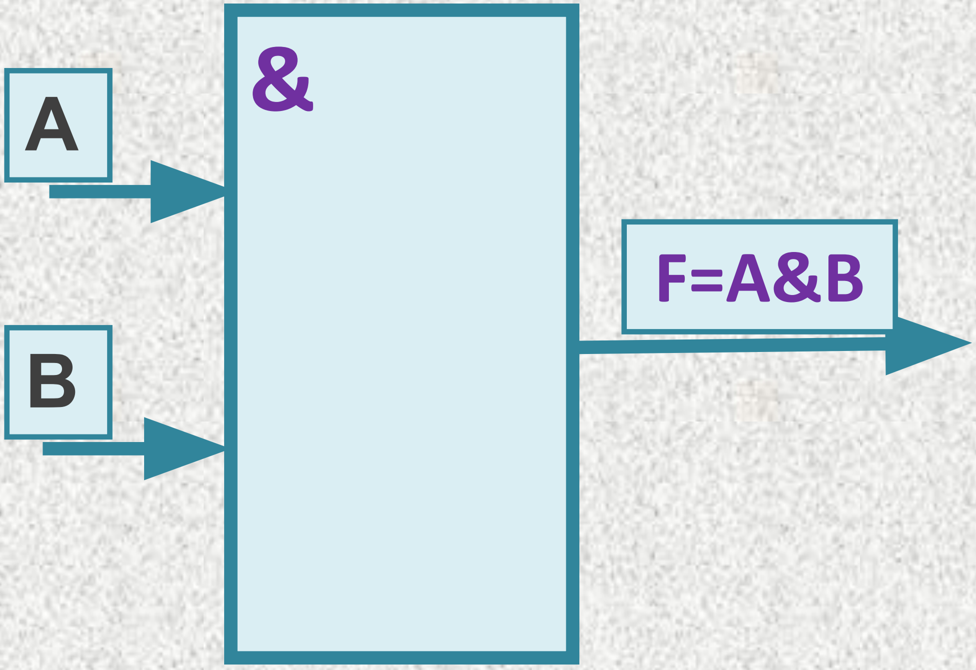
Логические элементы компьютера

Клод Шеннон
(1916 г.)

Клод Шеннон впервые доказал применимость булевой алгебры в теории контактных и релейно-контактных схем и использовал ее в своих работах (1938 г.)

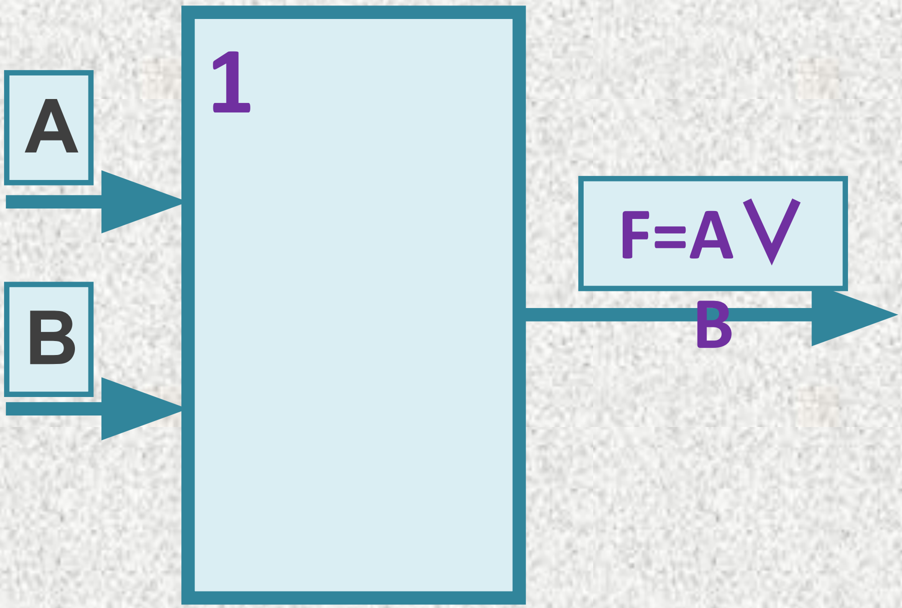
Логические операции «И», «ИЛИ», «НЕ» лежат в основе работы преобразователей информации любого компьютера.

Конъюнктор - логический элемент «**И**», преобразует входные сигналы и выдает результат логического умножения.



A	B	F=A&B
0	0	0
0	1	0
1	0	0
1	1	1

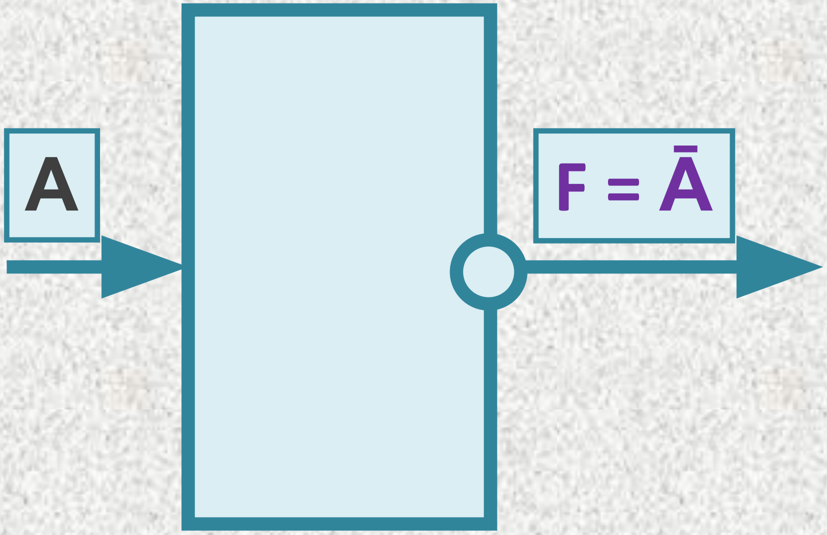
Дизъюнктор - логический элемент «**ИЛИ**», преобразует входные сигналы и выдает результат логического сложения.



A	B	F=A∨B
0	0	0
0	1	1
1	0	1
1	1	1

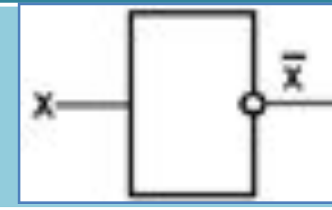
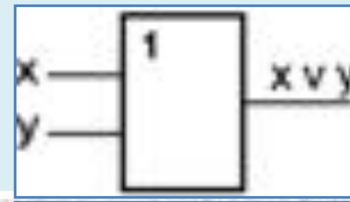
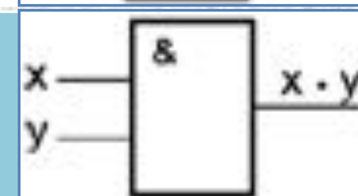
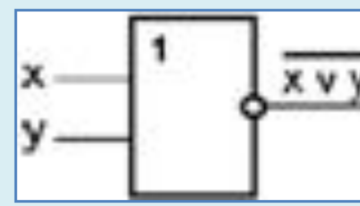
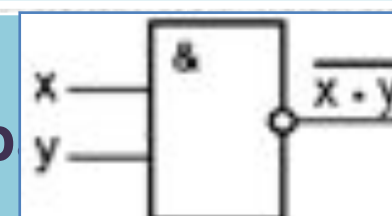
Инвертор - **логический элемент «НЕ».**

Преобразует входной сигнал и выдает результат логического отрицания.

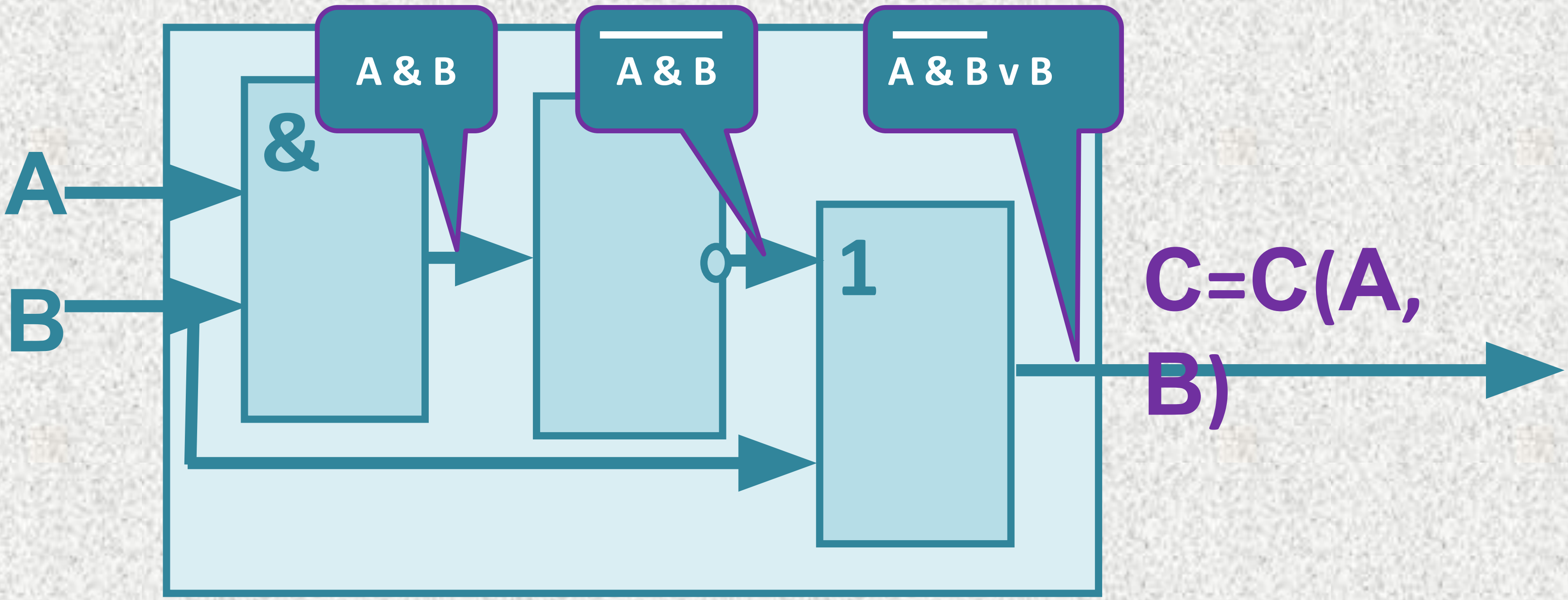


A	F = Ā
0	1
1	0

Логические элементы компьютера

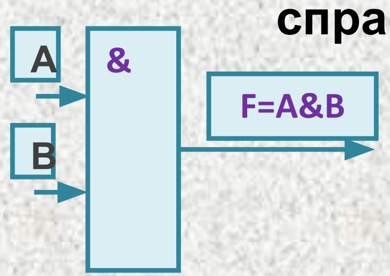
Функция	Логический элемент
Инверсия: F(x)=¬x	Схема НЕ (инвертор): 
Дизъюнкция: F(x,y)=x V y	Схема ИЛИ (дизъюнктор): 
Конъюнкция: F(x,y)=x & y	Схема И (конъюнктор): 
Инверсия дизъюнкции (стрелка Пирса): F(x,y)=x ↓ y=¬(x V y)	Схема ИЛИ—НЕ (элемент Пирса): 
Инверсия конъюнкции: (штрих Шеффера) F(x,y)=x | y = ¬(x & y)	Схема И—НЕ (элемент Шеффер 

Логическая схема устройства:

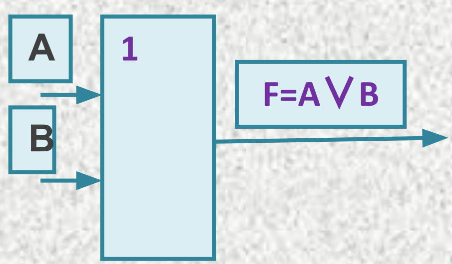


Формула функции: C=C(A, B)= $\overline{A \& B} \vee B$

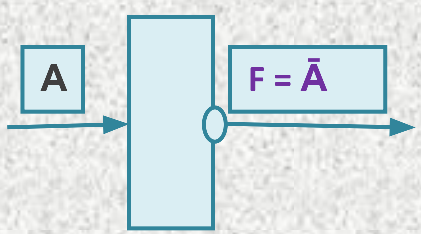
Логические схемы вентилей (для справки):



Конъюнктор



Дизъюнктор



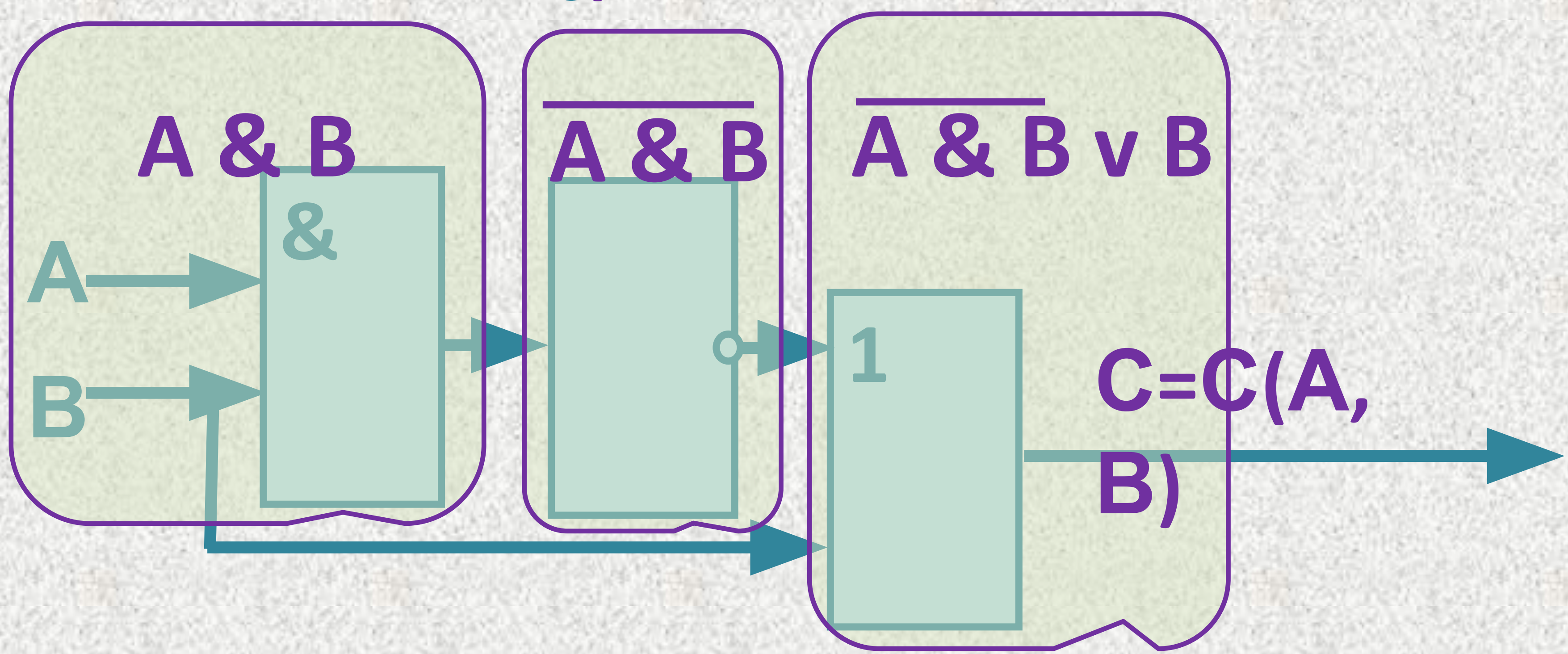
Инвертор

Зная логическую схему устройства, можно составить его формулу функции.

Анализируя формулу функции, можно создать логическую схему устройства.

Формула функции: $C=C(A, B)= \overline{A \,\&\, B} \vee B$

Логическая схема устройства:



Канонические формы булевых функций

Элементарной конъюнкцией называется конъюнкция нескольких переменных и/или их инверсий, причем среди переменных могут быть одинаковые.

Примеры:

¬X&X

X&¬Z

¬ X&Y& ¬Z

Дизъюнктивной нормальной формой (ДНФ) функции *F* называется равносильная ей формула, представляющая собой дизъюнкцию элементарных конъюнкций (логическую сумму логических произведений).

ДНФ не содержит скобок и общих для нескольких аргументов отрицаний.

Пример: X &X &¬Y V ¬X&Z

Совершенной дизъюнктивной нормальной формой (СДНФ) функции $F(x_1, x_2, \ldots, x_n)$ называется ДНФ, равная 1 на тех же наборах, что и функция F, и обладающая четырьмя свойствами совершенства.

Четыре «свойства совершенства» ДНФ формулы функции:

1. Каждое логическое слагаемое формулы содержит все аргументы функции.
2. Все логические слагаемые формулы различны.
3. Ни одно логическое слагаемое формулы не содержит одновременно аргумент функции и его инверсию.
4. Ни одно логическое слагаемое формулы не содержит один аргумент более одного раза.

Теорема

Пусть $F(x_1, x_2, \ldots, x_n)$ **– булева функция, не равная тождественно нулю, тогда существует СДНФ, выражающая функцию** $F(x_1, x_2, \ldots, x_n)$**.**

СДНФ функции $F(x_1, x_2, \ldots, x_n)$ **можно получить**

- с помощью равносильных преобразований,

- с помощью таблицы истинности.

Получение СДНФ функции с помощью равносильных преобразований

Для формулы функции получить ДНФ. Затем помощью равносильных преобразований добиться выполнения свойств совершенства для нее.

Основные приемы:

1) Пусть *B* есть слагаемое в ДНФ функции, не содержащее *x1*, тогда:

B* ≡ *B&(x1 V ¬x1)* ≡ *B & x1 V B & ¬x1

2) Если в ДНФ встретится два одинаковых слагаемых

B V B, то оставить одно: *B* ≡ *B V B*

3) Если в некоторое слагаемое *B* переменная x1 входит дважды, то лишнюю надо отбросить: *x1 ≡ x1 & x1*

4) Если слагаемое *B* в ДНФ содержит переменную *x1* и ее отрицание *¬x1*, то *B* можно

Переход от таблицы истинности функции к СДНФ

Правило перехода от таблицы истинности к формуле функции в СДНФ:

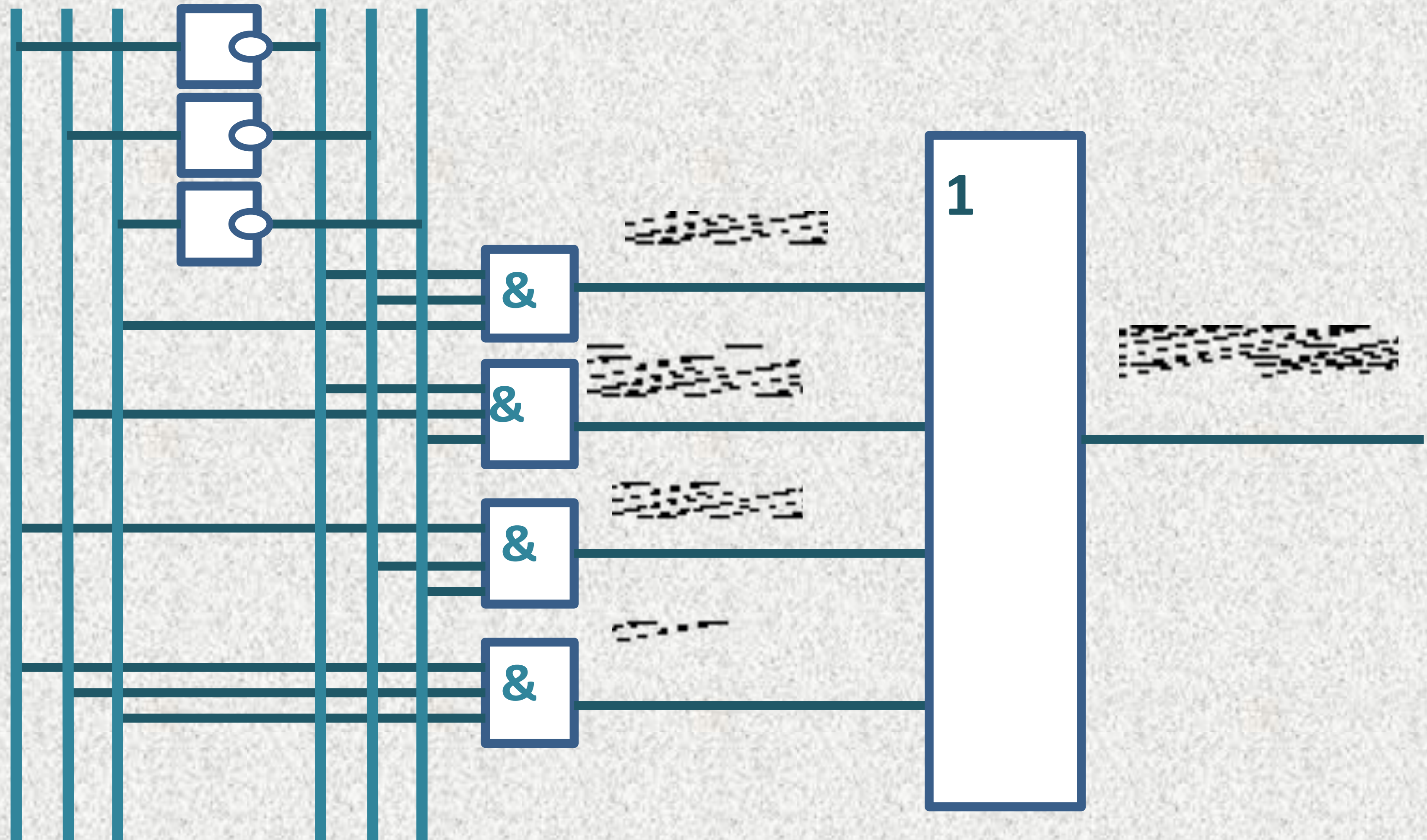
- записать дизъюнкцию полных конъюнкций по числу единичных значений функции,
- подписать под ними наборы значений аргументов, на которых функция равна единице,
- и проставить операцию инверсии для аргументов, которым соответствуют нули в двоичных номерах этих наборов.

Пример 1. Задана таблица истинности булевой функции F(X, Y, Z). Составить СДНФ и логическую схему функции F(X, Y, Z).

Порядковый номер набора аргументов функции	X	Y	Z	F(X,Y,Z)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

Порядковый номер набора аргументов функции	X	Y	Z	F(X,Y,Z)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

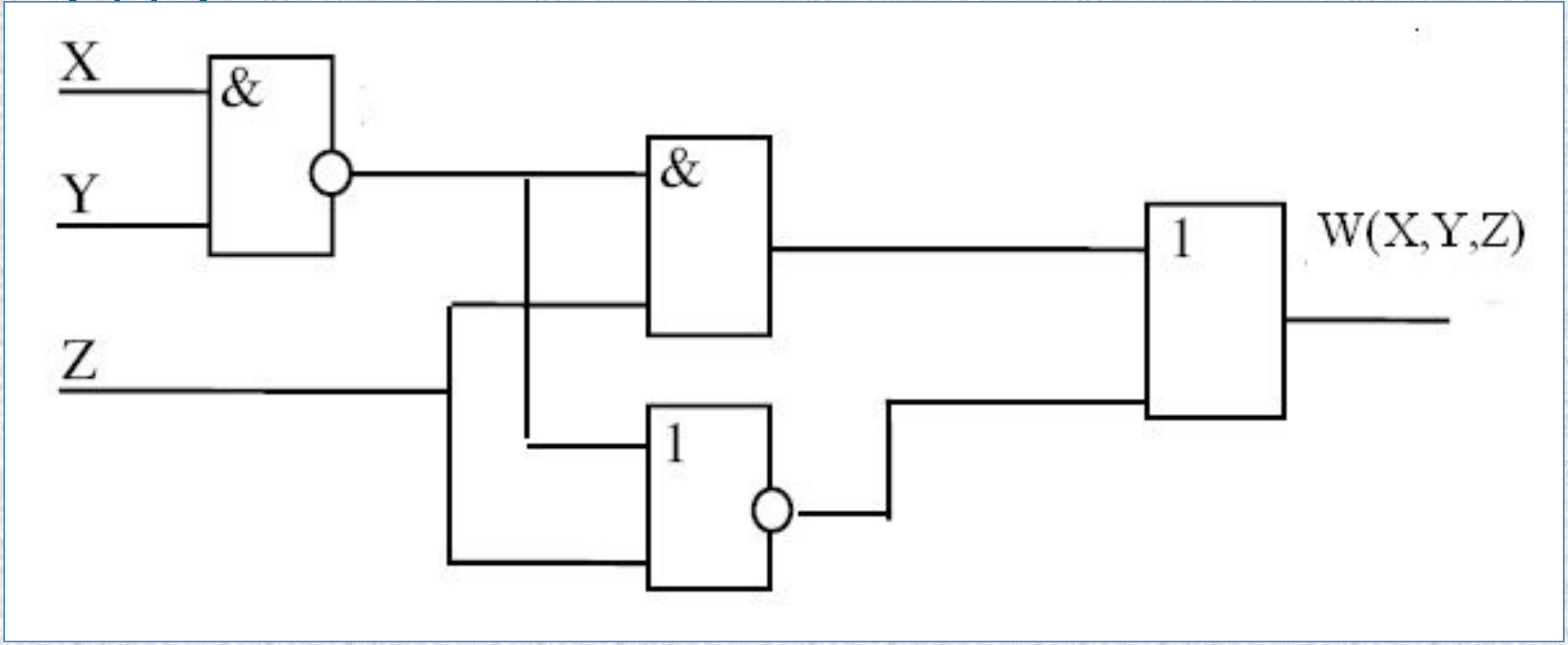
Используя вентили: инвертор, конъюнктор и дизъюнктор, построить логическую схему для функции:



Переход от логической схемы к формуле функции

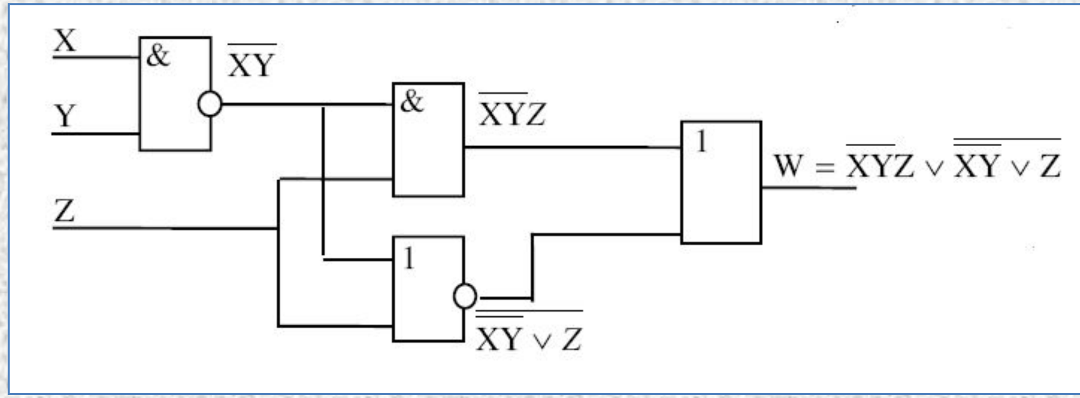
Пример 2. Задана логическая схема функции W(X,Y,Z)

Построить таблицу истинности и СДНФ функции W(X,Y,Z)



Логическая схема функции W(X,Y,Z)

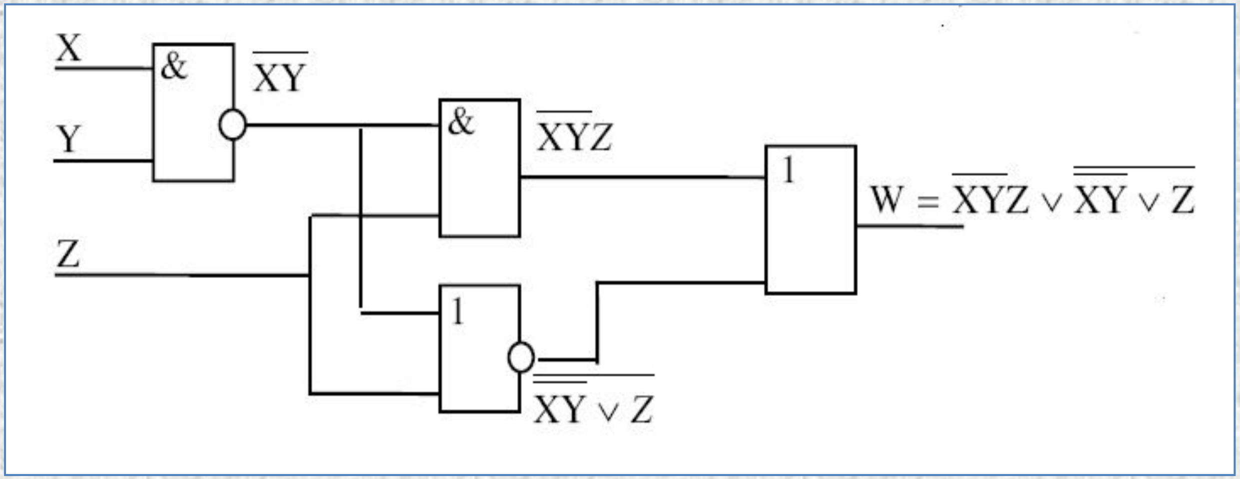
1. Для перехода от формулы к дизъюнктивной нормальной форме функции выписать выходные формулы всех логических элементов схемы.



2. Построить таблицу истинности найденной функции, определить номера наборов переменных, на которых W(X,Y,Z) принимает зн

x	y	z	не(x и y)	не(x и y) и z	не(x и y) или z	НЕ (не(x и y) или z)	W(X,Y,Z)
0	**0**	**0**	1	0	1	0	**0**
0	**0**	**1**	1	1	1	0	**1**
0	**1**	**0**	1	0	1	0	**0**
0	**1**	**1**	1	1	1	0	**1**
1	**0**	**0**	1	0	1	0	**0**
1	**0**	**1**	1	1	1	0	**1**
1	**1**	**0**	0	0	0	1	**1**
1	**1**	**1**	0	0	1	0	**0**

Наборы: 001, 011, 101, и 110, порядковые номера наборов: 1, 3, 5,



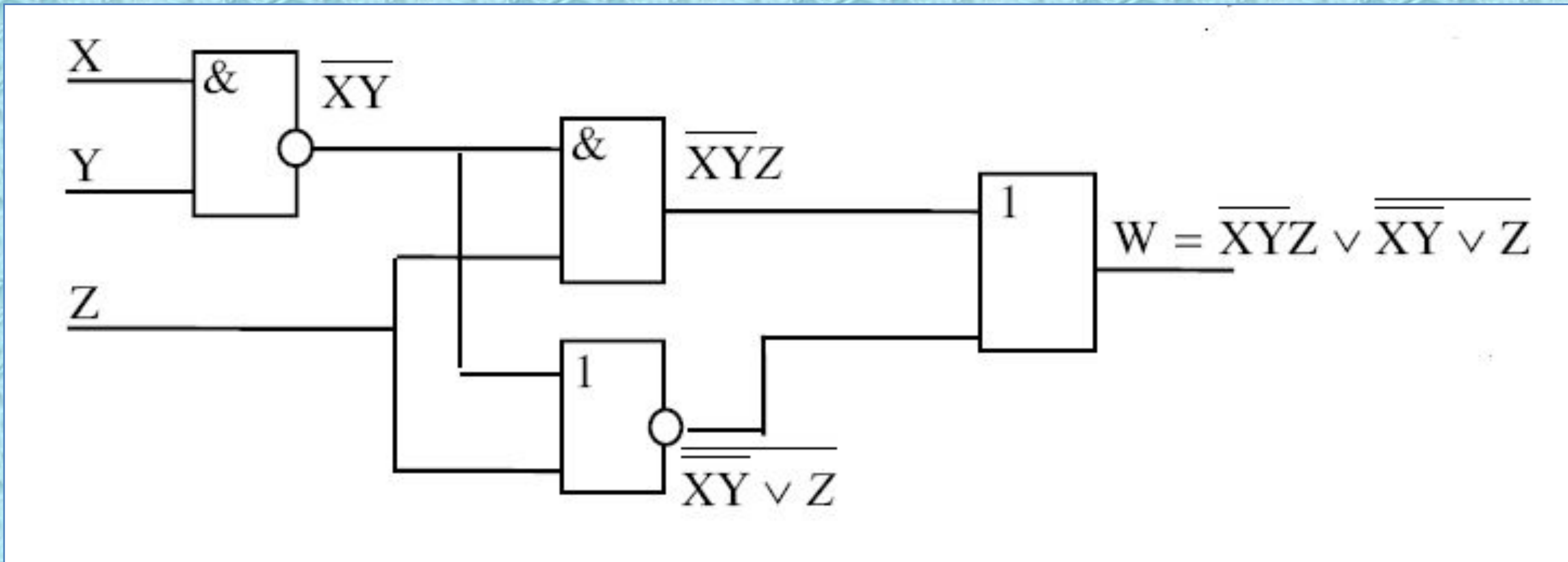
$$W(X, Y, Z) = K(1) + K(3) + K(5) + K(6)$$

$$W(X,Y,Z) = \begin{Bmatrix} XYZ & XYZ & XYZ & XYZ \\ 0\ 0\ 1 & 0\ 1\ 1 & 1\ 0\ 1 & 1\ 1\ 0 \end{Bmatrix}$$

$$W(X,Y,Z) = \overline{X} \cdot \overline{Y} \cdot Z + \overline{X} \cdot Y \cdot Z + X \cdot \overline{Y} \cdot Z + X \cdot \overline{Y} \cdot \overline{Z}$$

x	y	z	W(X,Y,Z)
0	0	0	**0**
0	0	1	1
0	1	0	**0**
0	1	1	1
1	0	0	**0**
1	0	1	1
1	1	0	1
1	1	1	**0**

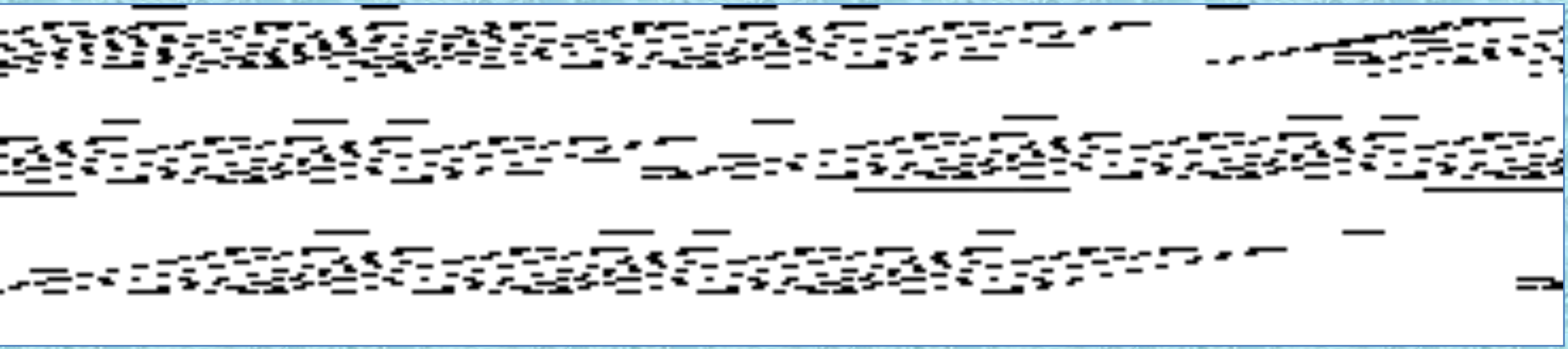
1. Для перехода от формулы к дизъюнктивной нормальной форме функции выписать выходные формулы всех логических элементов схемы



2. При наличии общих для нескольких переменных инверсий, используя правило де Моргана, «опускать» их на переменные. При наличии двойных отрицаний – убирать их:

Раскрыть все скобки и добавить недостающие переменные умножением членов полученной ДНФ на скобки, равные единице:

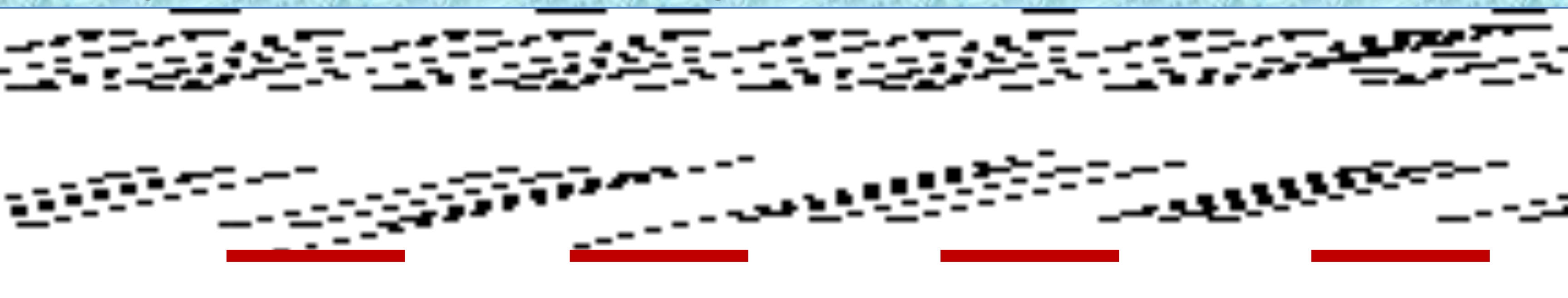
Раскрыть скобки и оставить в формуле только по одной из повторяющихся конституент:



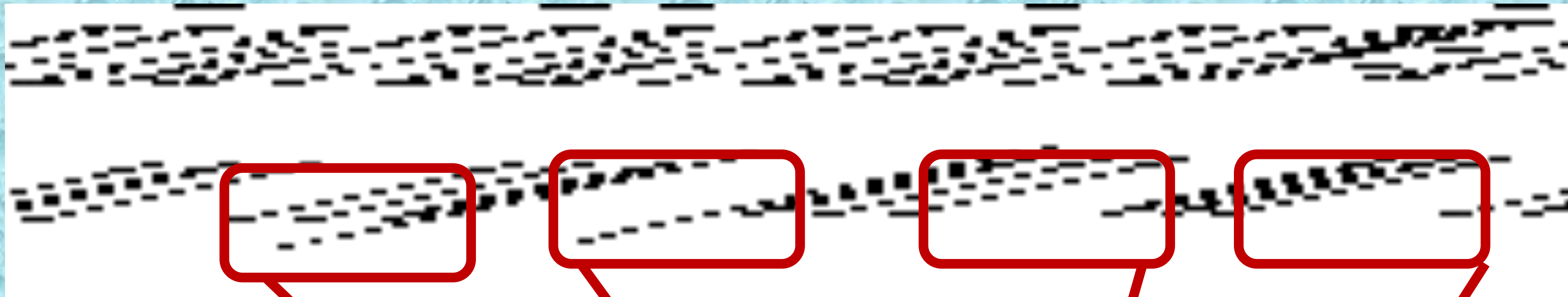
СДНФ функции W(X, Y, Z)

3. Построить таблицу истинности найденной функции.

Сначала определить номера наборов, на которых полученные конституенты равны 1:



Это наборы: 011, 001, 101, и 110, порядковые номера наборов: 3, 1, 5, 6.
Значения функции на этих наборах равны единице, на остальных – нулю.
Занести значения *W(X,Y,Z)* в таблицу истинности.



Порядковый номер набора аргументов функции	X	Y	Z	W(X,Y,Z)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

4. Таблица истинности функции W(X, Y, Z) построена

Операцию конъюнкции называют

двойственной операции дизъюнкции,

а операцию дизъюнкции -

двойственной операции конъюнкции.

Пусть функция *F* содержит только операции конъюнкции, дизъюнкции и отрицания.

Формулы *F* и *F** называются *двойственными*, если формула *F** получается из формулы *F* путем замены в ней каждой операции на двойственную.

Например, для формулы *F ≡ (x V y) & z* двойственной формулой будет формула *F*≡ (x & y) V z.*

Теорема. Если формулы A и B равносильны, то равносильны и им двойственные формулы, то есть A*≡B*.

Элементарной дизъюнкцией называется дизъюнкция нескольких переменных и/или их инверсий.
Примеры:

¬X V X
X V ¬ Z
¬X V Y V¬Z

Конъюнктивной нормальной формой (КНФ) функции A(X, Y, Z) называется равносильная ей формула, представляющая собой конъюнкцию элементарных дизъюнкций (логическое произведение логических сумм).
Пример: *(X V X V ¬ Y) & (¬X V Z)*

Совершенной конъюнктивной нормальной формой (СКНФ) функции $F(x_1, x_2, \dots, x_n)$ называется равносильная ей КНФ функции $F(x_1, x_2, \dots, x_n)$, обладающая следующими свойствами:

1) каждая элементарная дизъюнкция, входящая в КНФ функции $F(x_1, x_2, \dots, x_n)$, содержит все аргументы функции $F(x_1, x_2, \dots, x_n)$;
2) все элементарные дизъюнкции, входящие в КНФ, различны;
3) каждая элементарная дизъюнкция, входящая в КНФ функции $F(x_1, x_2, \dots, x_n)$, содержит аргумент только один раз;
4) ни одна элементарная дизъюнкция, входящая в КНФ функции $F(x_1, x_2, \dots, x_n)$, не содержит одновременно аргумент и его инверсию.

СКНФ функции $F(x_1, x_2, \ldots, x_n)$ можно получить:

- с помощью таблицы истинности,

- преобразовать СДНФ, используя закон двойственности,

- с помощью равносильных преобразований.

Построение СКНФ функции по таблице истинности:

1. В таблице истинности отметить наборы аргументов, на которых значение функции равно нулю.
2. Составить дизъюнкцию полных конъюнкций, число дизъюнктов равно числу нулевых значений функции.
3. Сопоставить дизъюнкты и отмеченные в п. 1 наборы аргументов: если значение аргумента в наборе равно нулю, то в конъюнкцию включается аргумент, иначе - его инверсия.

Правило получения СКНФ функции F с помощью равносильных преобразований

Для функции F получить любую КНФ. Затем помощью равносильных преобразований добиться выполнения свойств совершенной конъюнктивной нормальной формы.

1. Если элементарная дизъюнкция B, входящая в КНФ, не содержит аргумент x1, тогда заменить:

   B ≡ B V (x1 & ¬x1) ≡ (B V x1) & (B V ¬x1).

1. Если КНФ содержит две одинаковые элементарные дизъюнкции, то одну можно исключить, так как B & B ≡ B.
2. Если в некоторую элементарную дизъюнкцию B аргумент x1 входит дважды, то лишний нужно исключить, так как: x1 ≡ x1 V x1.
3. Если в элементарную дизъюнкцию входит аргумент x1 и его отрицание ¬x1, то их можно исключить, т.к.

   x1 V ¬ x1≡1,

Электронные схемы

Пример

Построить логическую схему функции, суммирующей два одноразрядных двоичных числа *A* и *B* и вырабатывающей их сумму *S* и перенос *P*.

Решение

<u>Этап 1</u>. Построить таблицы истинности функций S и P:

S=(A+B) (mod 2),

если (A=1 и B=1), то S=0, а

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

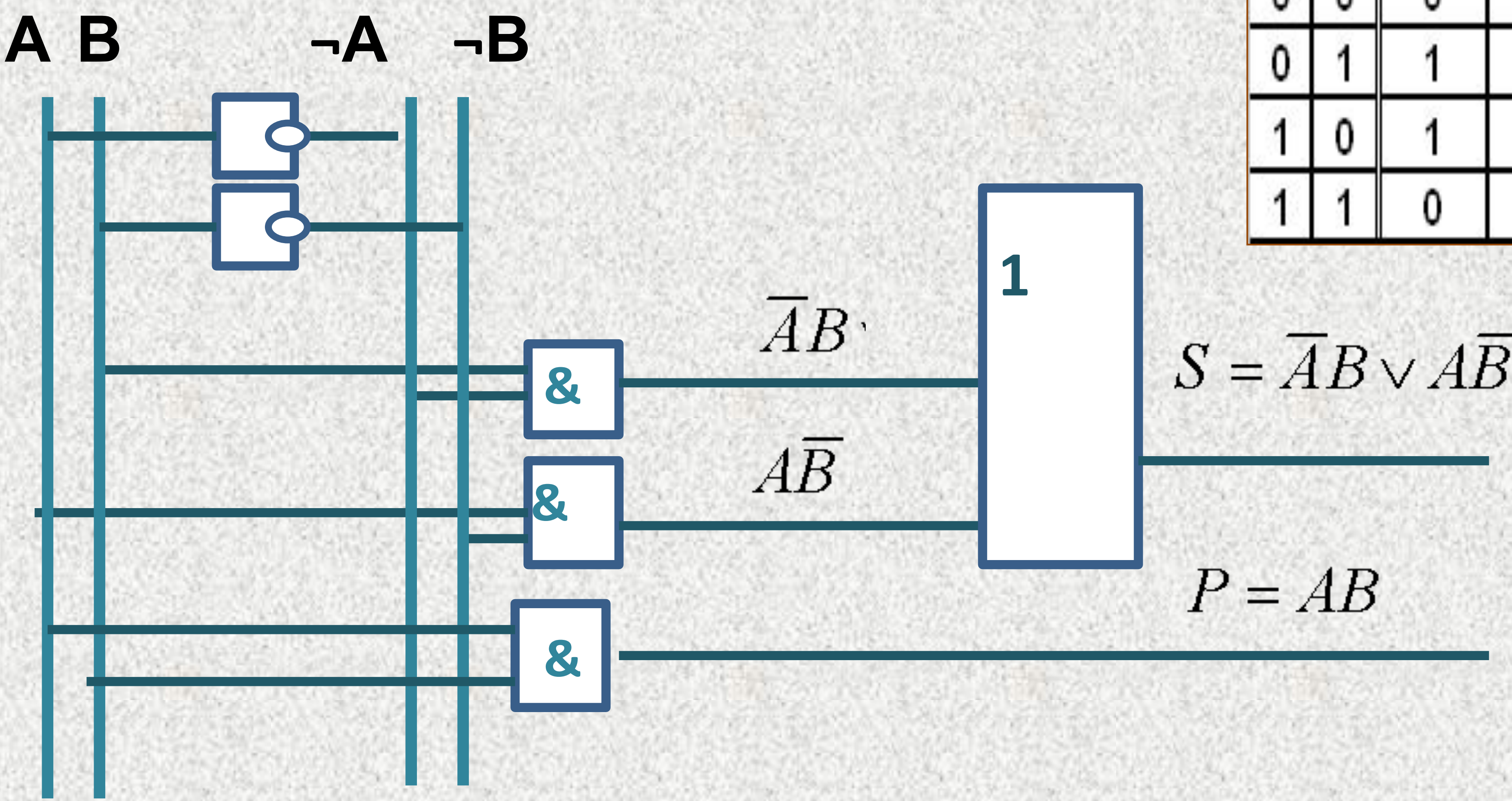
<u>Этап 2</u>. СДНФ функций:

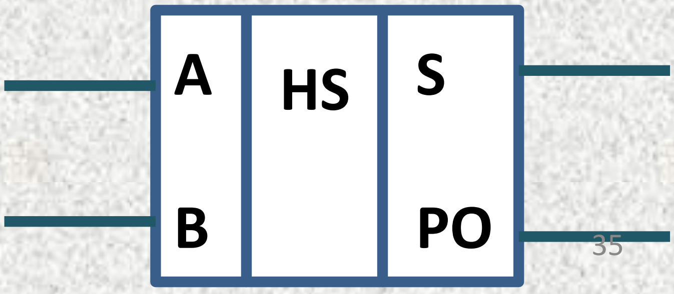
суммы: $S = \overline{A}B \vee A\overline{B}$

переноса: $P = AB$

Этап 3. Логическая схема функций суммы S и переноса P

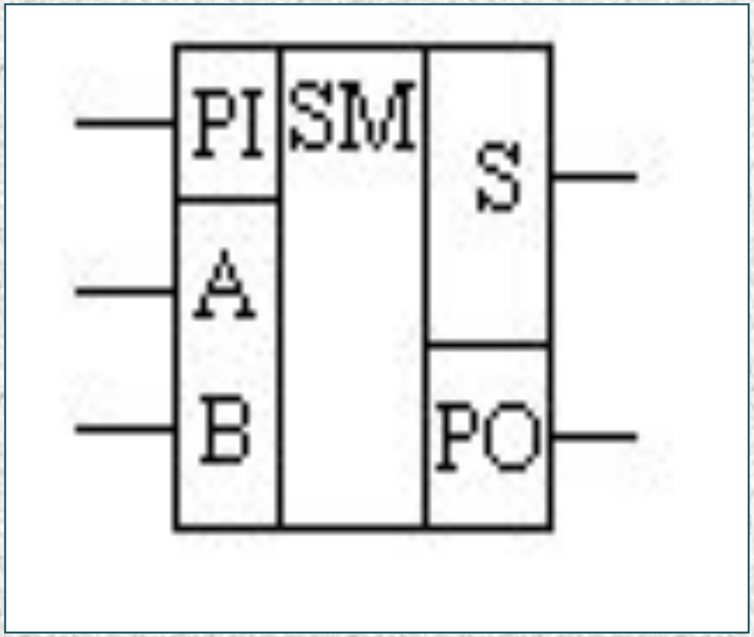
A	B	S	P
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1





Логическая схема полусумматора

Условно-графическое изображение полного двоичного одноразрядного сумматора на схемах



http://digteh.ru/digital/sum.php

Для того чтобы получить многоразрядный сумматор, достаточно соединить входы и выходы переносов соответствующих двоичных разрядов.

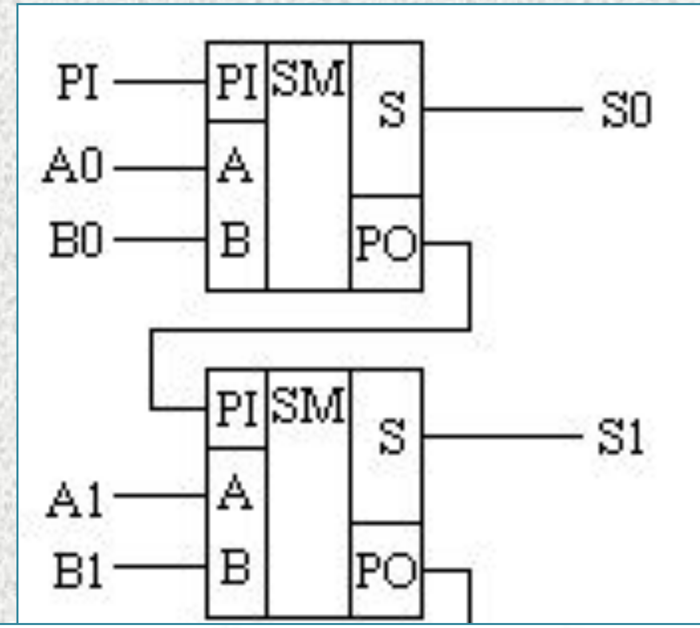
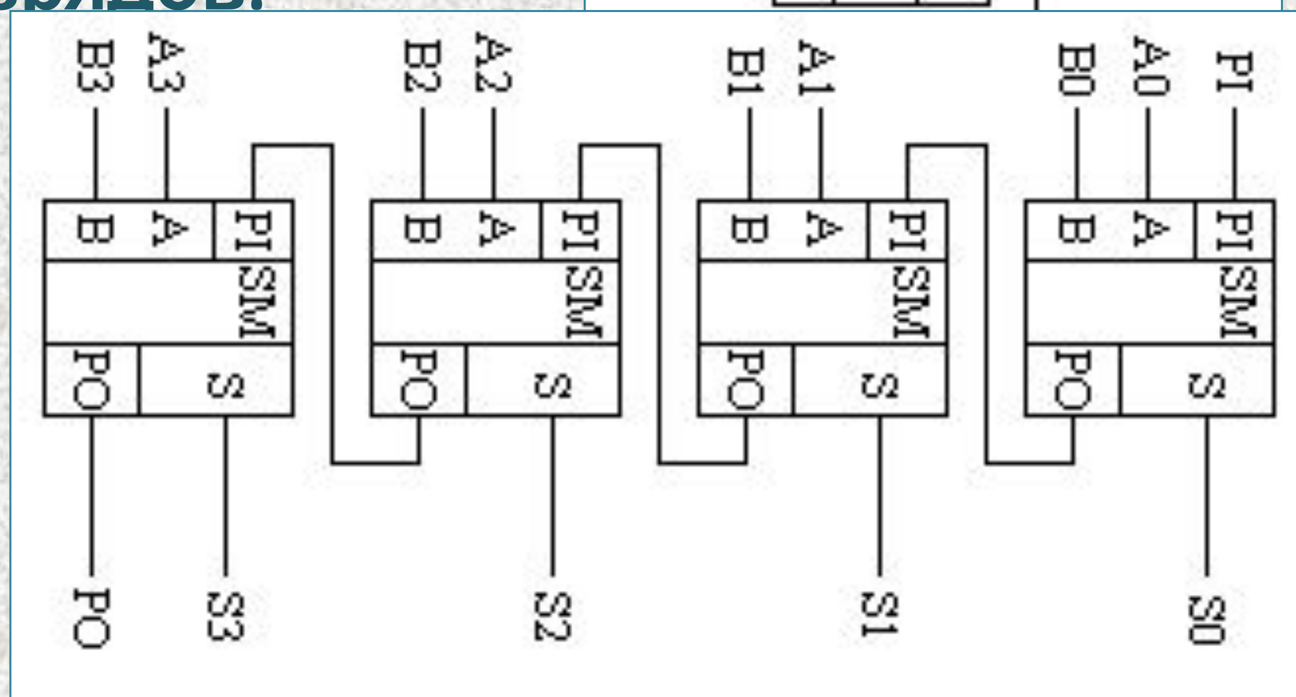
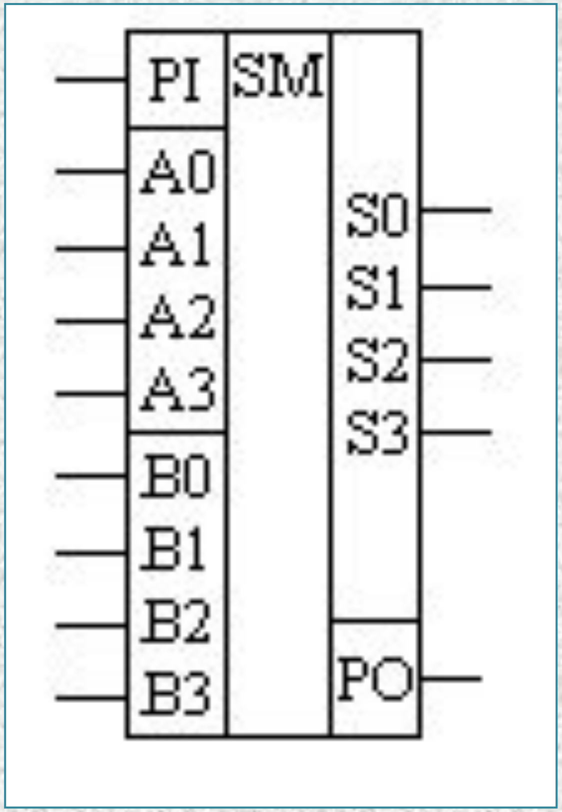


Схема соединения одноразрядных сумматоров для реализации четырехразрядного сумматора:



http://digteh.ru/digital/sum.php

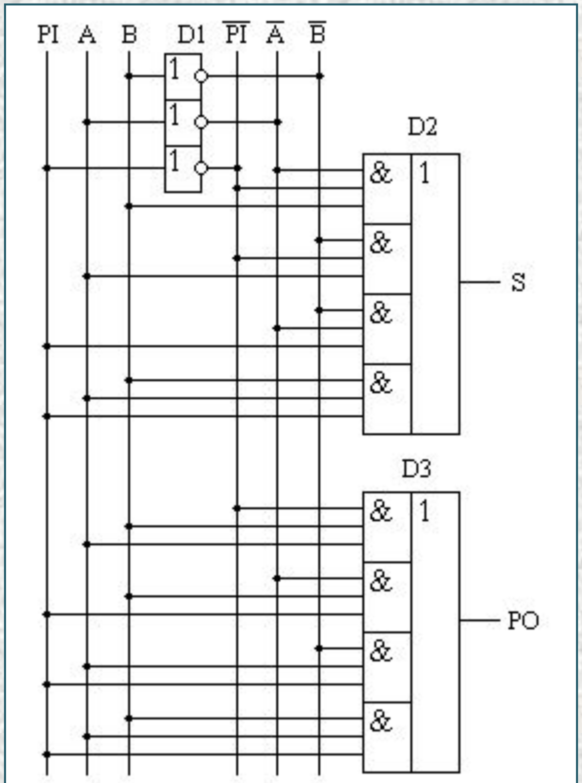
Условно-графическое изображение полного двоичного четырехразрядного сумматора :



http://digteh.ru/digital/sum.php

PI	A	B	S	PO
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

Таблица истинности полного двоичного одноразрядного сумматора

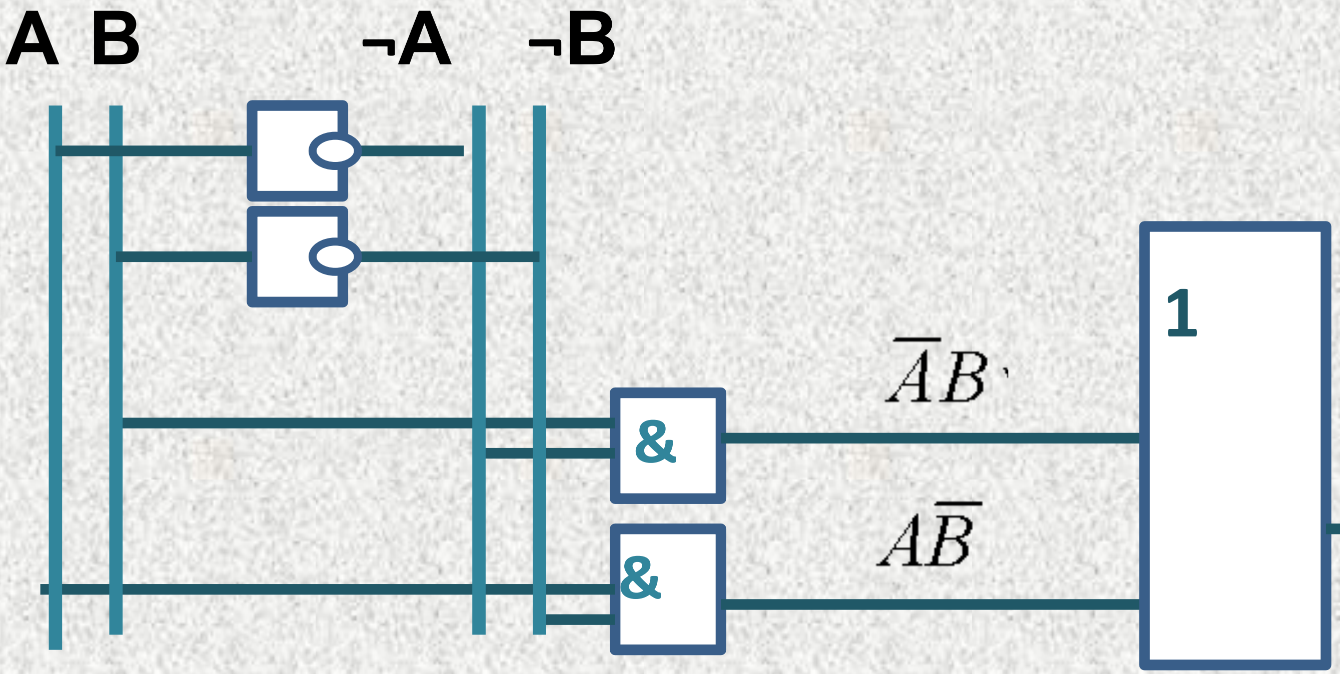


Логическая схема, реализующая таблицу истинности полного двоичного одноразрядного сумматора.

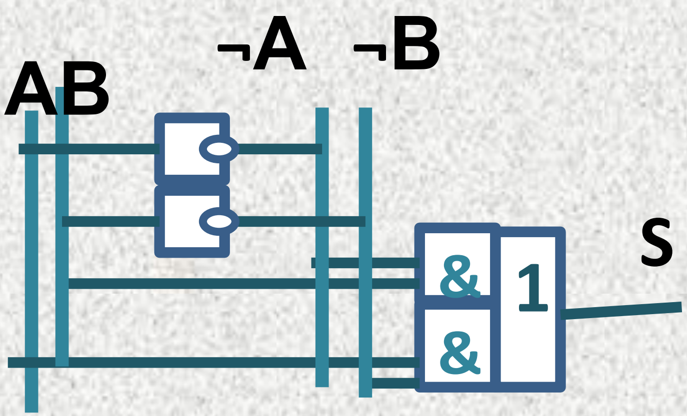
http://digteh.ru/digital/sum.php

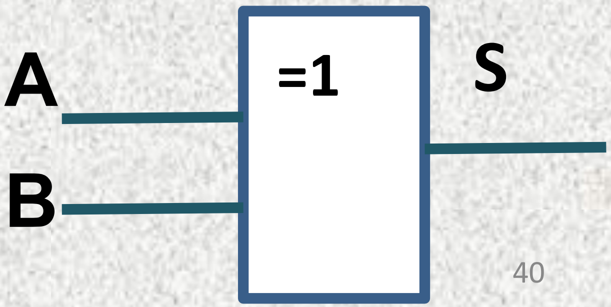
Сумма по модулю 2

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

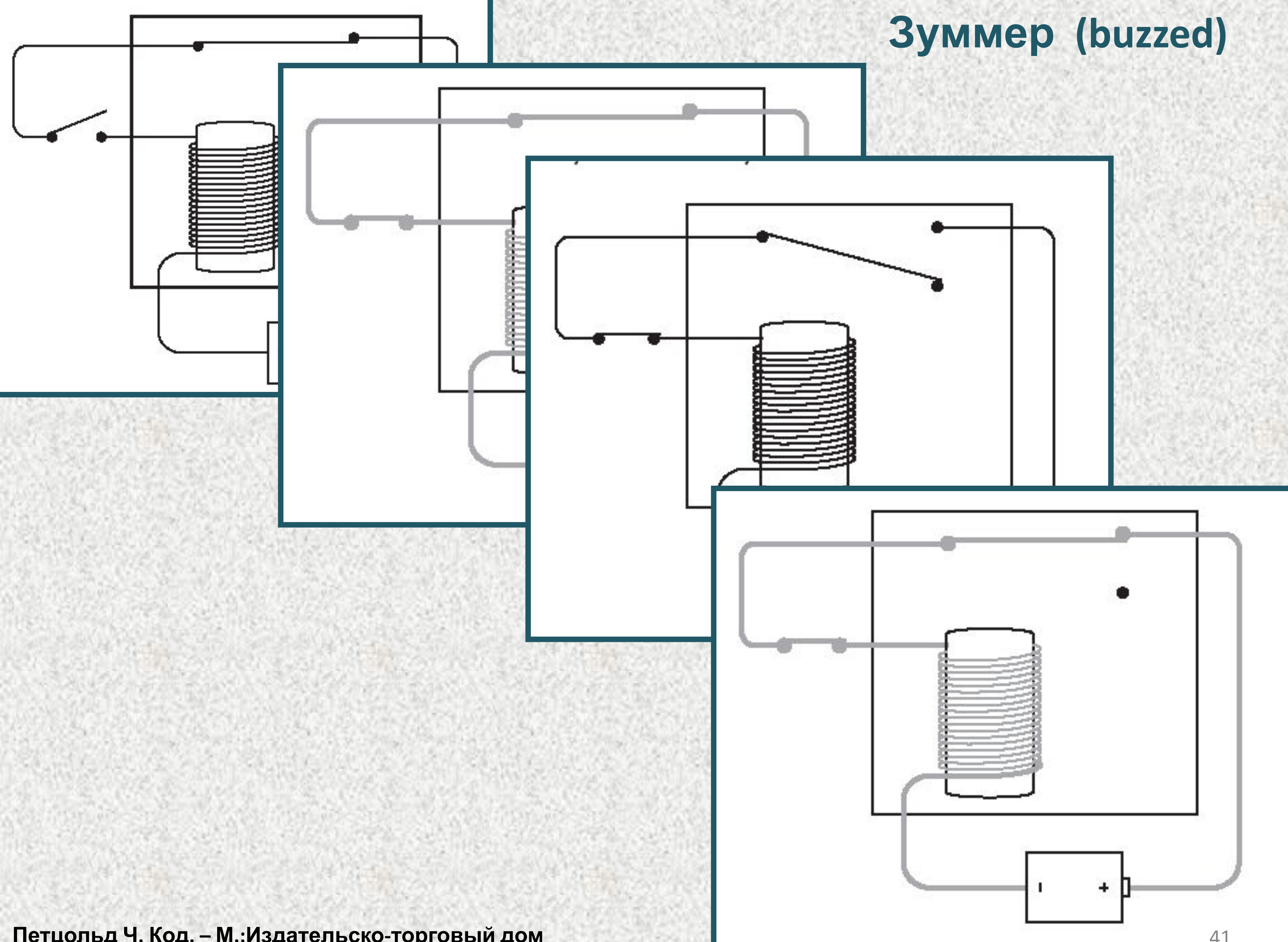


$S = \bar{A}B \vee A\bar{B}$



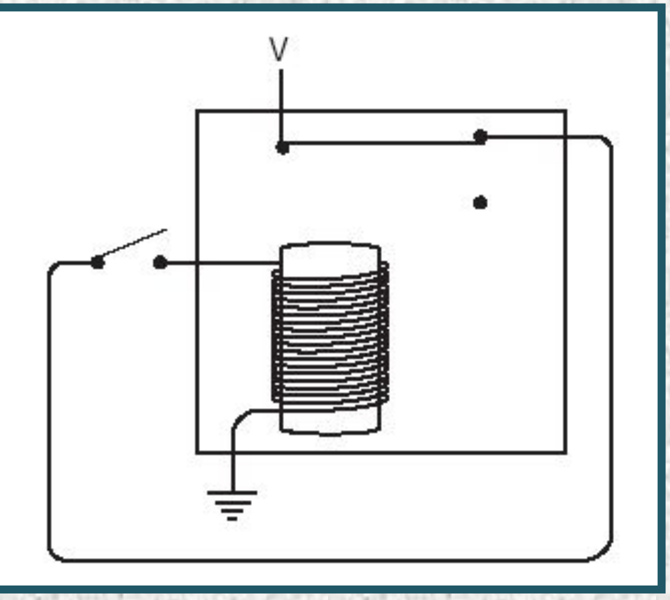


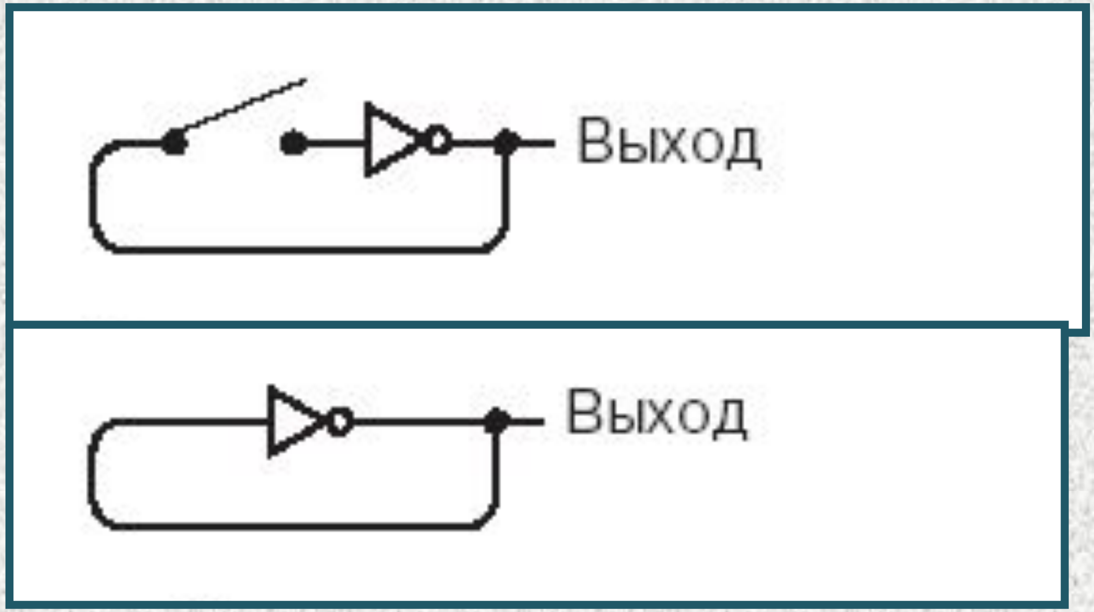
Зуммер (buzzed)

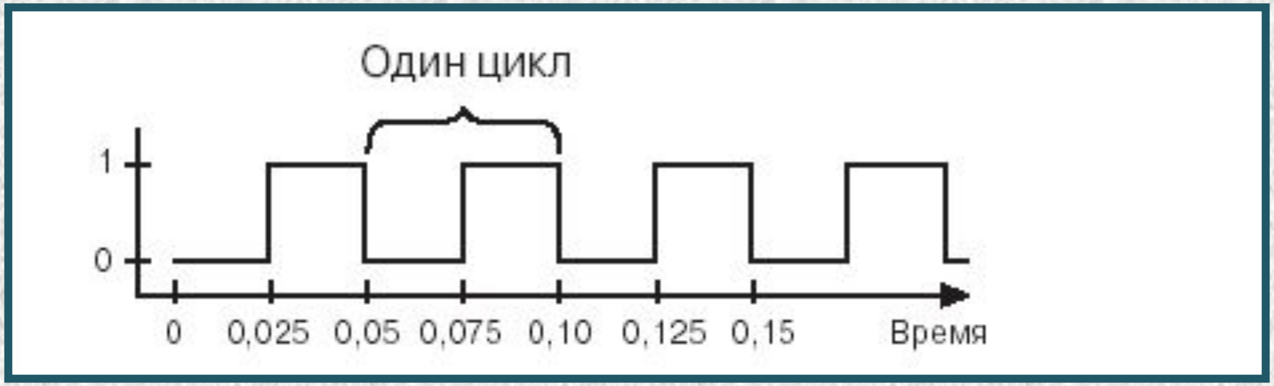


Петцольд Ч. Код. – М.:Издательско-торговый дом «Русская Редакция», 2001. – 512 с.

Вибратор (oscillator) – реле для организации синхронной работы компонентов ЭВМ

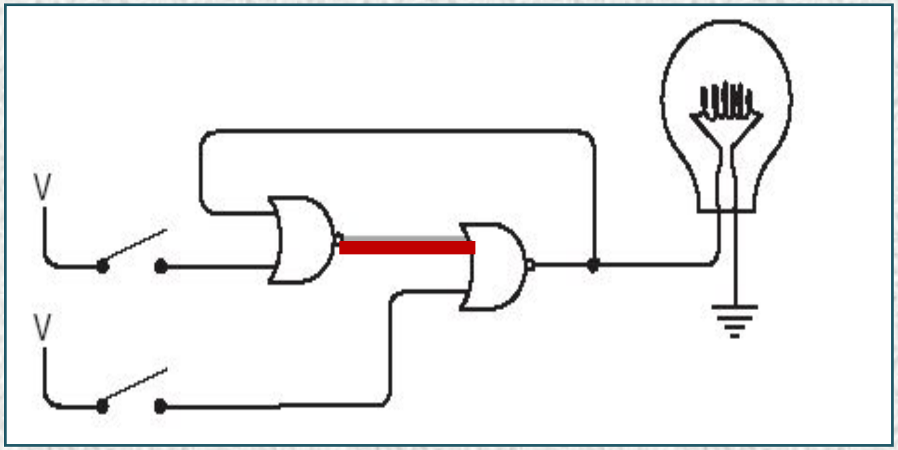


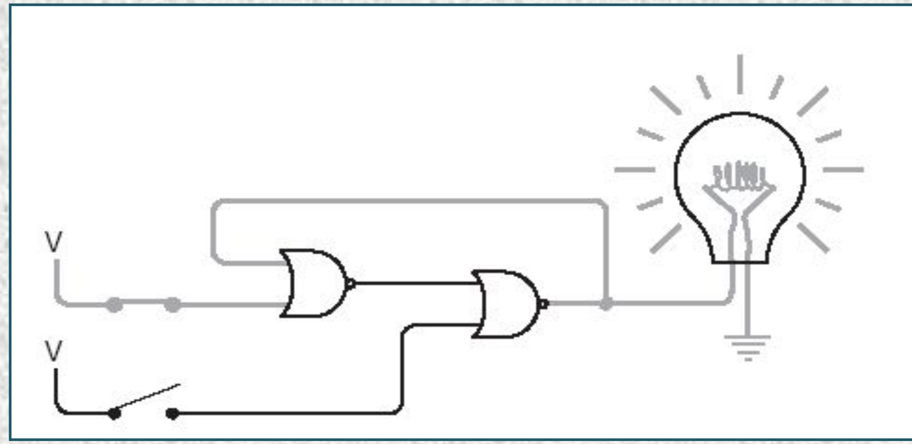


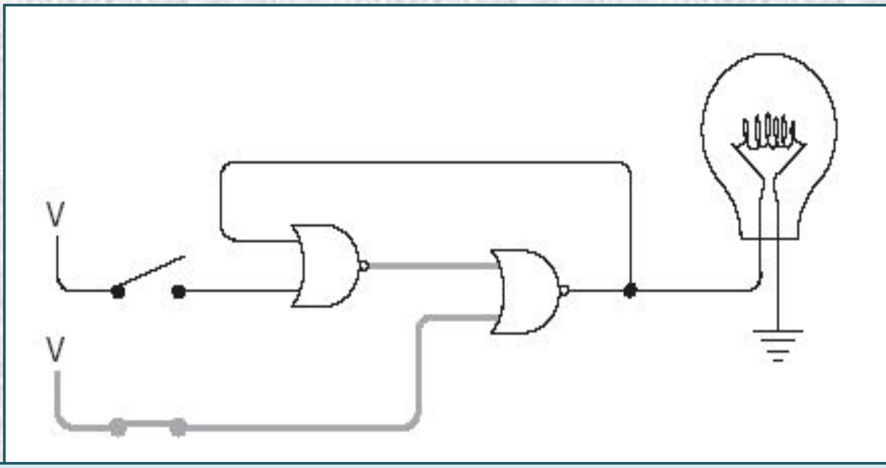


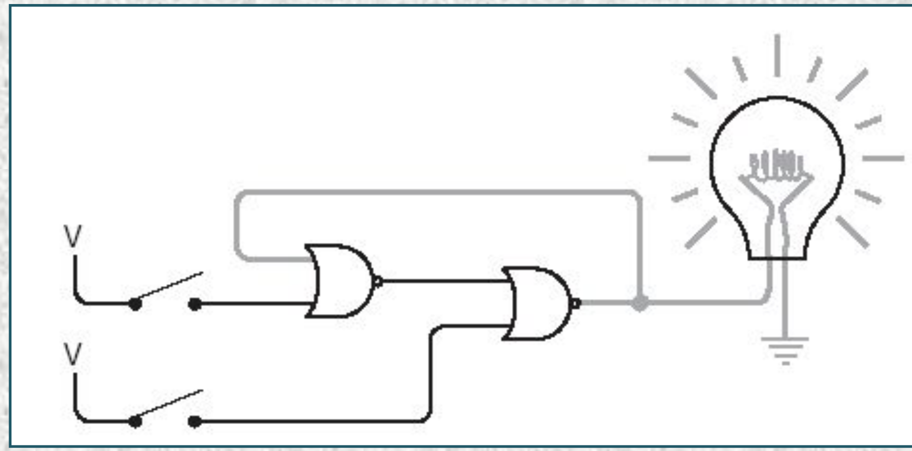
Частота колебаний в секунду = 1/0,05 сек = 20 Гц

Петцольд Ч. Код. – М.:Издательско-торговый дом «Русская Редакция», 2001. – 512 с.





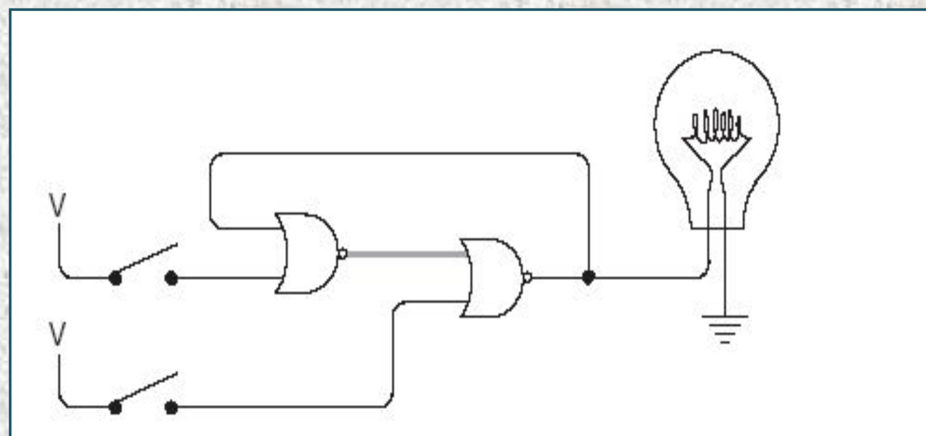




Триггер (f … ***стойчивых***
состояни …
переключ …
Триггер с … **ию.**
Состояни … **ирует,**
какой пер … **мкнут**
последни …

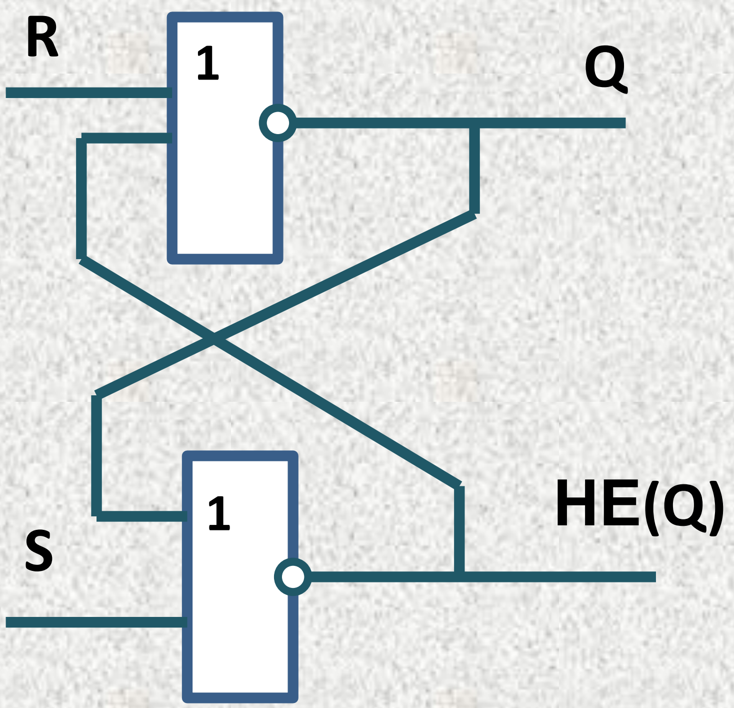
		ИЛИ-НЕ
0	0	1
0	1	0
1	0	0
1	1	0

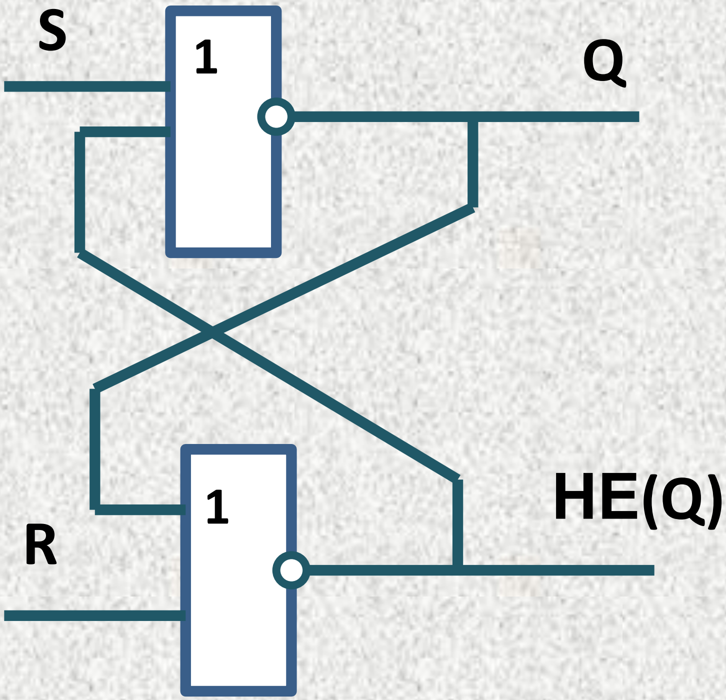
Если лам … **дним был**
замкнут верхний переключатель, а если не



Для хранения информации в ОП и регистрах ЦП применяется устройство ТРИГГЕР. Ячейка памяти состоит из 8, 16 или 32 триггеров, что и определяет разрядность ЦП.

Триггер строится из двух элементов «ИЛИ» и двух элементов «НЕ».





S	R	Q	HE(Q)
0	0	не меняются	
0	1	0	1
1	0	1	0
1	1	*	*

RS - триггер

Выходы: Q (лампочка), НЕ(Q) – противоположный ему.
Входы R (reset) и S (set).

Если S=1 (соответствует замкнутому верхнему переключателю),
то выход Q=1, НЕ(Q)= 0.

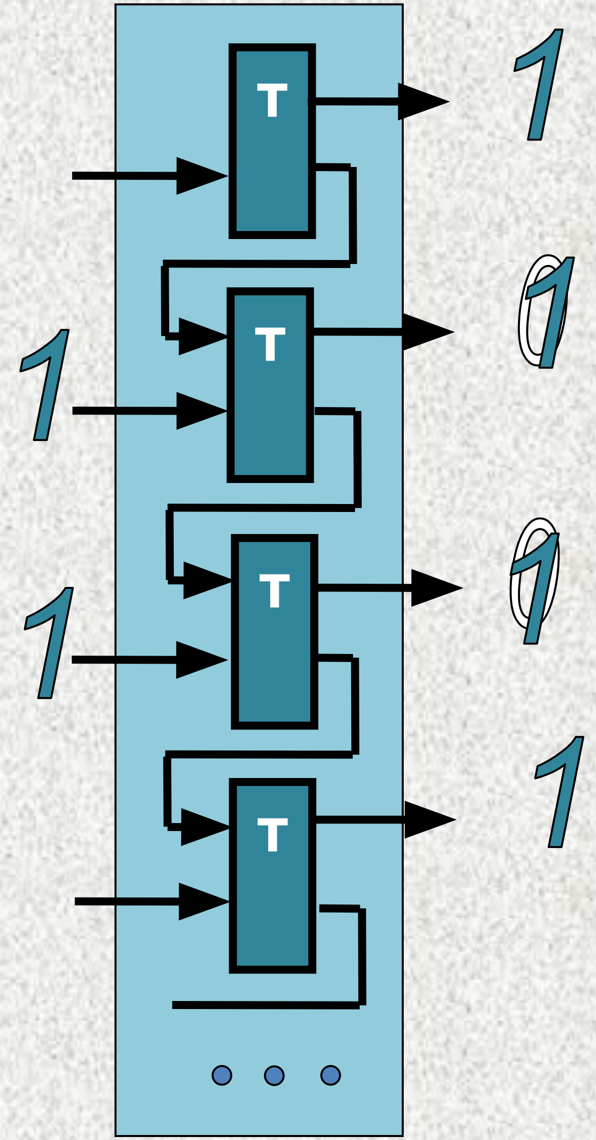
Если R=1 (соответствует замкнутому нижнему переключателю),
то выход Q=0, НЕ(Q)= 0.

Если R=0 и S=0,
то значение выхода Q зависит от

Петцольд Ч. Код. – М.:Издательско-торговый дом «Русская Редакция», 2001. – 512 с.

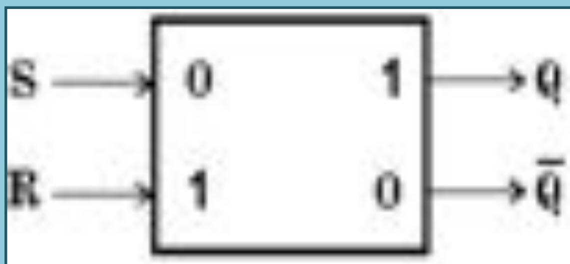
Узлы и память ЭВМ состоят из отдельных логических элементов

- Несколько триггеров объединяются в группы - регистры, которые используются в качестве запоминающих устройств (ЗУ). Регистр из N триггеров хранит N-разрядные двоичные слова.
- ОЗУ ЭВМ конструируется в виде набора регистров. Один регистр образует одну ячейку памяти, которая имеет свой номер



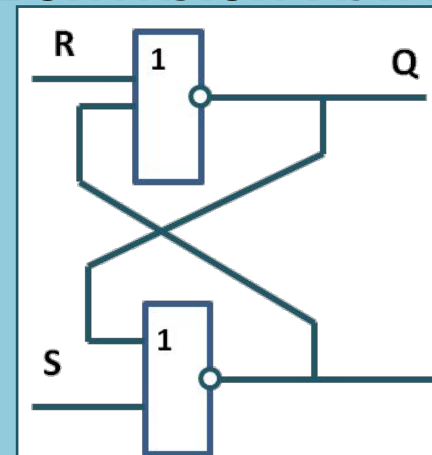
<table>
<tr><th>Электронная схема</th><th>Реализация с помощью вентилей</th></tr>
<tr><td>

RS - триггер:



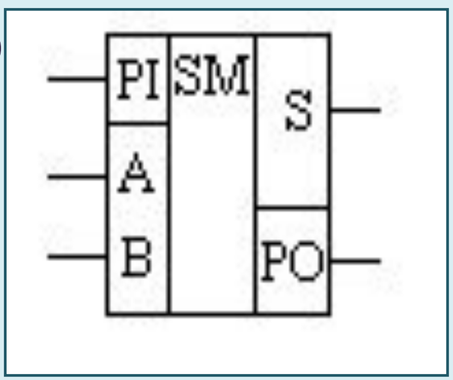
</td><td>

Реализация триггера с помощью вентилей ИЛИ—НЕ:

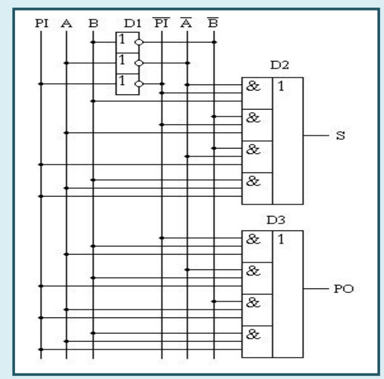


</td></tr>
<tr><td>

Сумматор



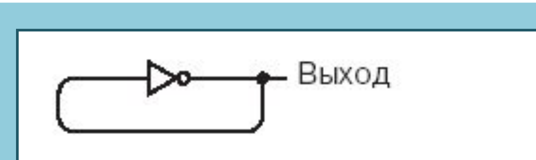
</td><td>



</td></tr>
<tr><td>

Вибратор:

</td><td>



</td></tr>
</table>

Дата, тема лекции	Самостоятельная работа в LMS MOODLE, сроки выполнения
19.10.2015	22.10.2015 - 27.10.2015
Логические основы построения и работы ЭВМ Базовые логические элементы ЭВМ: - логические элементы компьютера, реализующие элементарные логические функции (И, ИЛИ, НЕ, ИЛИ-НЕ, И-НЕ), -электронные схемы (сумматор, триггер).	Выполнить контрольные задания в тестовой форме: *22.10.2015 - 27.10.2015 Базовые логические элементы ЭВМ* Пароль для доступа к тесту – *хартли.* В каждом варианте – 3 задания, время решения -30 минут, максимальная оценка – 15 баллов. Допускается не более трех попыток выполнения. Итоговая оценка – среднее арифметическое набранных баллов по всем попыткам.