1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Применение алгебры логики для описания логических элементов и систем

Системы логического уравнения (СЛУ) бывают комбинационными и последовательностными. Последние называются также событийно – управляемыми автоматами.

В комбинационных СЛУ выходные сигналы формируются только при определённых комбинациях входных логических сигналов, принимающих значения 0 или 1. В последовательностных СЛУ выходные сигналы зависят не только от комбинации входных, но и последовательности их поступления во времени, что обеспечивается наличием элементов памяти. В настоящее время последовательстные СЛУ в зависимости от сложности решаемых задач выполняются на базе программируемых логических контроллеров – ПЛК.

В курсовой работе студентам предстоит выполнить анализ и синтез комбинационной СЛУ на контактных и бесконтактных элементах (для четырёх входных и двух выходных сигналов.) При проектировании этой же СЛУ в среде CoDeSys вводим дополнительные условия (Приложение 3), которые приводят её в класс событийно-управляемой логики. Эту задачу выполнить на контактных элементах было бы значительно труднее.

Словесное изложение работы даже простых СЛУ выглядит громоздко и затрудняет проведение анализа. Математическим аппаратом для описания СЛУ служит двузначная (бинарная) алгебра логики, все переменные в которой могут принимать только два значения (**0** или **1**).

Основным понятием, используемым при анализе и синтезе систем логического управления, является логическая функция.

Принципы построения логической функции по алгоритму работы системы автоматики иллюстрируется ниже на примере.

Логическая функция выражает зависимость выходных переменных от входных и также принимает, в зависимости от значения последних и связывающих их логические действия, два состояния: **0** или **1**. Можно встретить и такую запись этих состояний : Ложь или Истина; FALSE или TRUE).

Так как каждая переменная может иметь только два значения, то возможное количество различных комбинаций (наборов) $N_{комб}$ для n переменных будет равно:

$$N_{комб} = 2^n$$

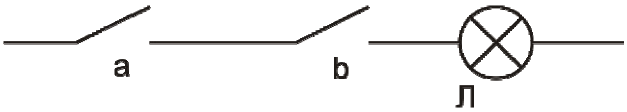
Все действия над переменными в бинарной алгебре выполняются с помощью следующих основных операций, которые наглядно иллюстрируются соответствующими релейно-контактными схемами.

В настоящее время «релейная автоматика» звучит весьма архаично (вроде как ламповый компьютер). Но, зная навыки построения контактных СЛУ и используя современный язык LD в среде CoDeSys, можно легко эти схемы перевести в программу для ПЛК. Это язык и был в свое время разработан для инженеров, знающих релейные автоматы, но не владеющими навыками программирования на языках высокого уровня.

Логическое умножение

Логическое умножение (конъюнкция, функция "**И**") $Л=a\cdot b$**;** равнозначно последовательному соединению контактов. Все возможные комбинации входных сигналов и соответствующие им значения функции сведены в таблицу состояний. Очевидно, что только в одном случае результатом логического умножения станет единица, т.е. лампа **Л** "сработает", если замкнуть контакты и *a,* и *b*. Из этой словесной формулы союз "**И**" перешел в обозначение функции (как синоним логическому умножению) и в название бесконтактного элемента.

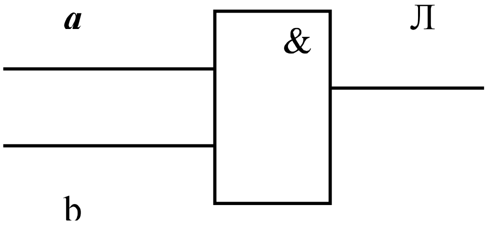
а) Релейно-контактный вариант реализации:



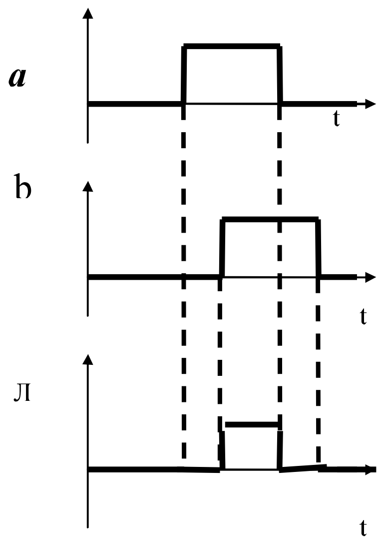
б). Таблица состояний:

a	b	Л
0	0	0
0	1	0
1	0	0
1	1	1

в). Обозначение бесконтактного элемента:



г). Временные диаграммы:



На выходе такого элемента появится сигнал (потенциал), если на его входах *a* и *b* одновременно действуют единичные сигналы.

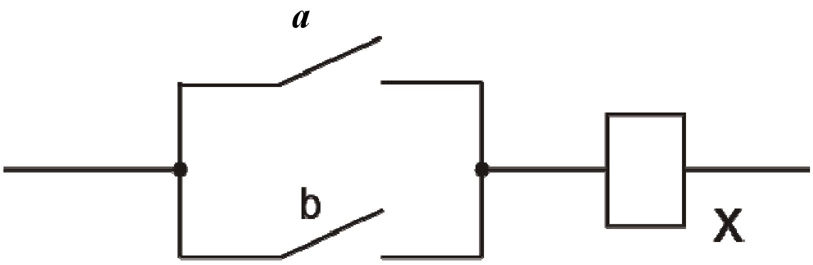
Применяются и другие обозначения операции логического умножения:

$$a \cdot b = a \wedge b = a \& b$$

Логическое сложение

Результат логического сложения (дизъюнкции, операции "**ИЛИ**") $\boldsymbol{X = a + b}$**;** легко установить из анализа схемы с параллельным соединением контактов.

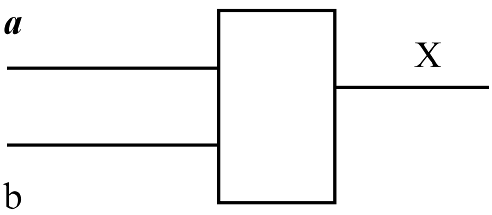
а) Релейно-контактный вариант реализации:



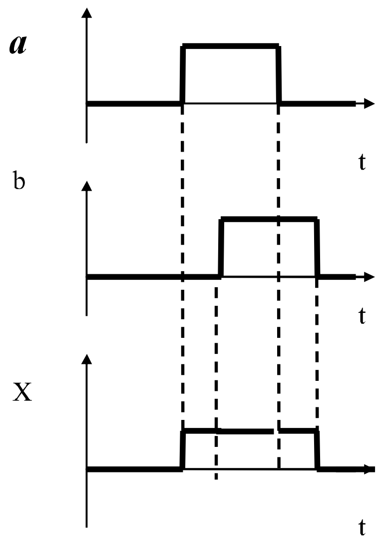
б) Таблица состояний:

a	b	X
0	0	0
0	1	1
1	0	1
1	1	1

в) Обозначение бесконтактного элемента:



г). Временные диаграммы:



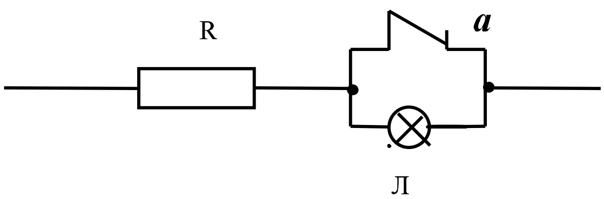
Очевидно, что катушка реле **X** получит питание, если замкнуть контакты или *a*, или *b*, или оба вместе.

Вместо знака "+" иногда употребляют «∨»: a+b=a∨b

<u>Логическое отрицание.</u>

Логическое отрицание (инверсия, операция "**НЕ**"), $Л = \overline{a}$ означающая, что значение логической функции **Л** противоположно или неравносильно значению переменной *a*. В нашем примере лампа горит (**Л=1**), если контакт *a* разомкнут (***a=0***), и лампа гаснет (**Л=0**), если контакт замкнуть (***a=1***).

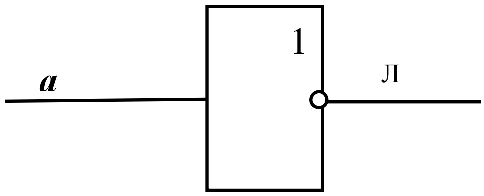
а) Релейно-контактный вариант реализации:



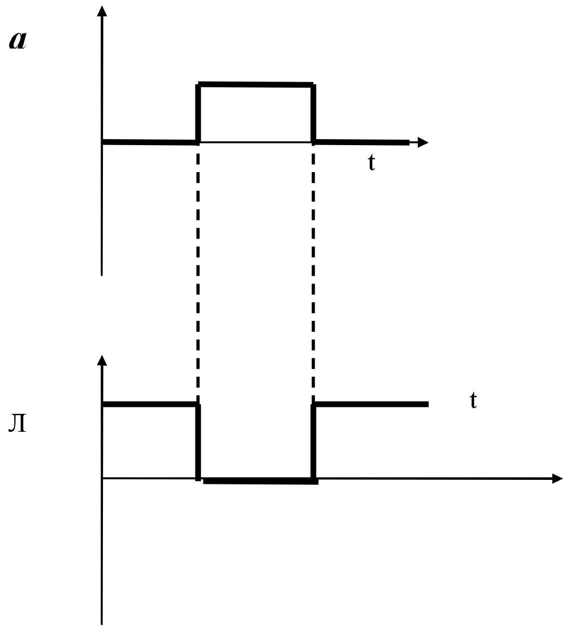
б) Таблица состояний:

a	Л
0	1
1	0

в) Обозначение бесконтактного элемента:



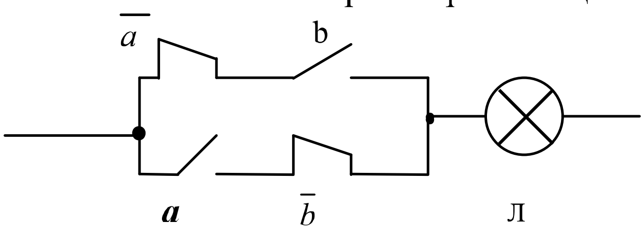
г) Временные диаграммы:



<u>Логическое сложение по модулю 2,</u> функция **”ИСКЛЮЧАЮЩЕЕ ИЛИ”**

$$Л = a \cdot \overline{b} + \overline{a} \cdot b$$

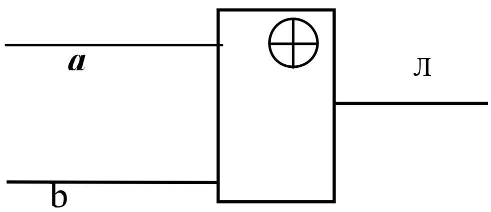
а) Релейно-контактный вариант реализации:



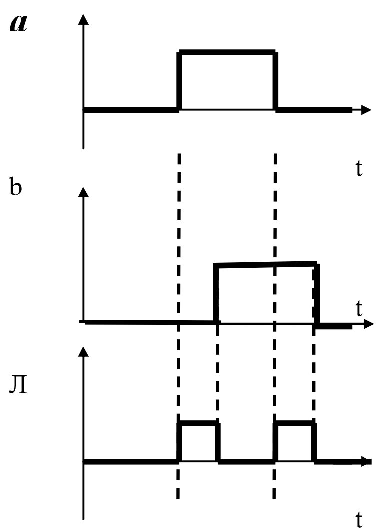
б) Таблица состояний:

a	b	Л
0	0	0
0	1	1
1	0	1
1	1	0

в) Обозначение бесконтактного элемента:



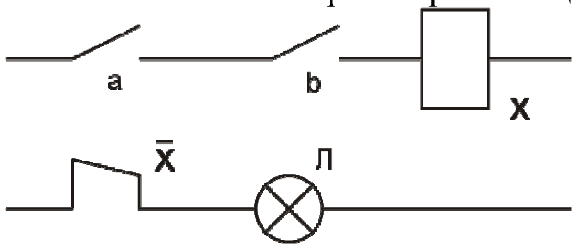
г) Временные диаграммы:



Инверсия конъюнкции, функция **“И - НЕ”:**

$$Л = \overline{a \cdot b}$$

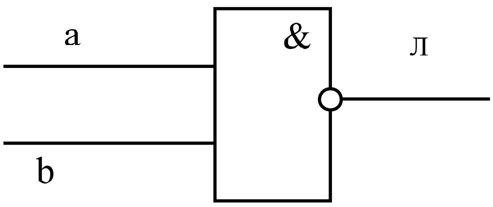
а) Релейно-контактный вариант реализации:



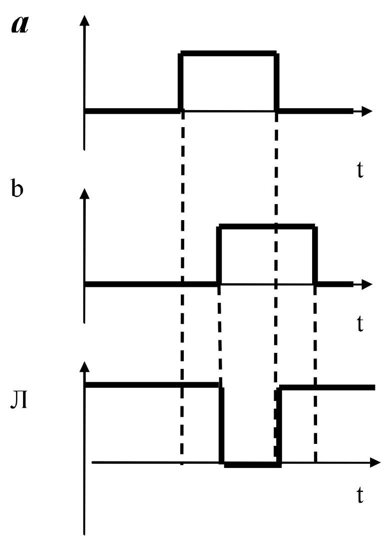
б) Таблица состояний:

a	b	Л
0	0	1
0	1	1
1	0	1
1	1	0

в) Обозначение бесконтактного элемента:



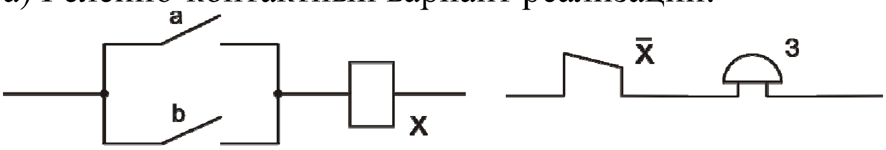
г) Временные диаграммы:



Инверсия дизъюнкции, функция **“ИЛИ -НЕ”:**

$$З = \overline{a + b}$$

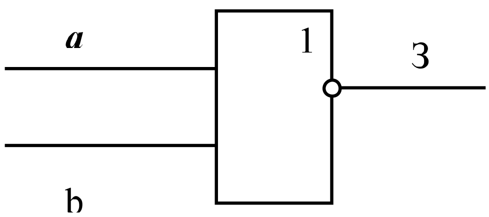
а) Релейно-контактный вариант реализации:



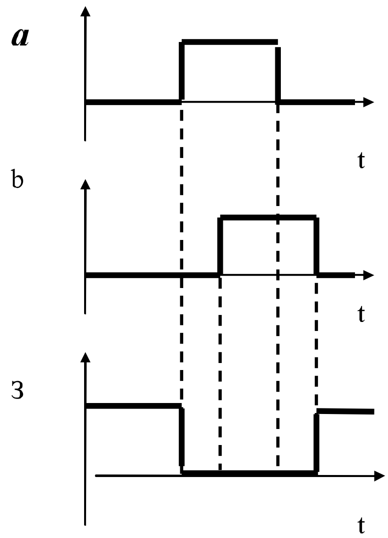
б) Таблица состояний:

a	b	З
0	0	1
0	1	0
1	0	0
1	1	0

в) Обозначение бесконтактного элемента:



г) Временные диаграммы:



Указанные логические операции справедливы и для большего числа переменных. Возрастет при этом лишь количество возможных комбинаций, т.е. число строк в таблице состояний.

Знак "=", который в обычной алгебре является знаком равенства, в данном применении выражает равносильность связываемых логических операций, так как сами функции лишены количественной меры и могут принимать лишь два качественных состояния: **0** или **1**.

На схемах, во избежание ошибок, входы бесконтактных логических элементов рисуют слева или сверху, а выходы - справа или снизу большей грани прямоугольника , условно изображающего элемент .

Простейший элемент "**НЕ**" (инвертор), является четырехполюсником. Более сложные элементы, имеющие несколько входов и выходов, - многополюсники. Но для упрощения схем общие (заземленные) клеммы входов и выходов, как и вспомогательные цепи (питания, смещения) не показывают, оставляя только используемые потенциальные вводы и выводы.

Все основные логические операции могут быть представлены через основные (элементарные) действия: **И, ИЛИ, НЕ** (рис.1).

При записи и чтении сложных логических функций предполагается, что знак инверсии связывает сильнее, чем другие знаки, а знак умножения связывает сильнее знака логического сложения. Этот принцип позволяет сокращать количество различных скобок. Например, логическую функцию:

$$X = \overline{\left(\left((a \cdot b) + \overline{(c+d)}\right) \cdot c + a\right)} \cdot b$$

следует записать в более простой форме:

$$X = \overline{\left(a \cdot b + \overline{c+d}\right) \cdot c + a} \cdot b$$

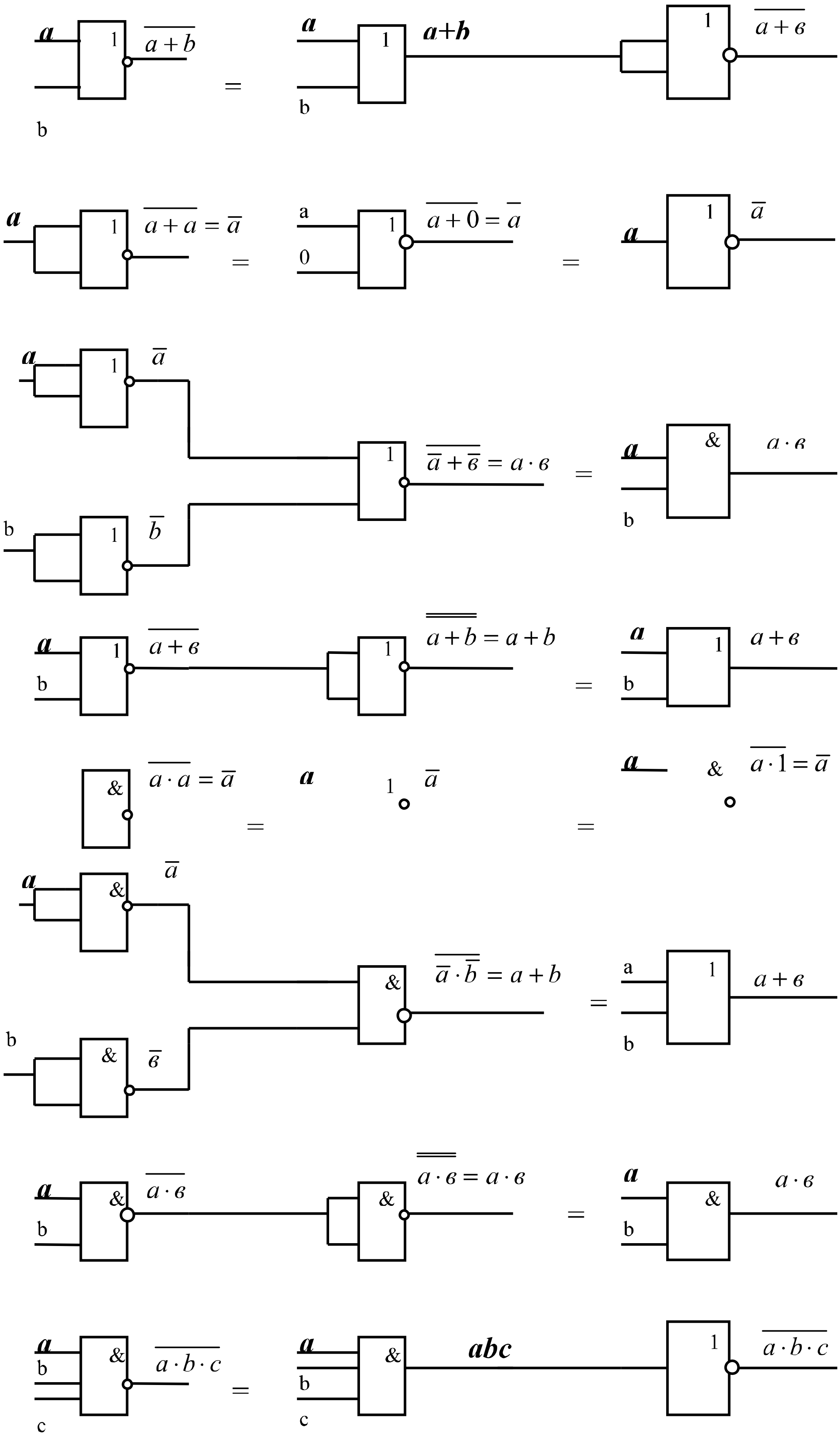


Рис. 1. Примеры элементарных логических операций

Как и в обычной алгебре здесь действуют законы:

Переместительный (коммутативный)

а) относительно логического умножения: *a b=b a;*

б) относительно логического сложения: *a+b=b+a;*

Сочетательный (ассоциативный)

а) относительно логического умножения: *(a b)c=(a c)b*;

б) относительно логического сложения: *(a+b)+c=a+(b+c);*

Распределительный (дистрибутивный)

а) относительно логического умножения: *(a+b)c=a c+b c;*

б) относительно логического сложения: *a b+c=(a+c)(b+c).*

Следует обратить внимание на отсутствие формальной аналогии между распределительным законом относительно логического сложения для бинарной алгебры и таким же законом для обычной алгебры.

Но есть и специфические аксиомы, законы и теоремы, которые легко воспринимаются при рассмотрении соответствующих релейно-контактных схем (табл. 1).

Таблица 1.

Основные аксиомы и законы алгебры логики

№ п/п	Наименование	Формулы	Схемы
1	2	3	4

1:˙ Аксиомы

$0 \cdot 0 = 0$

$1+1= 1$

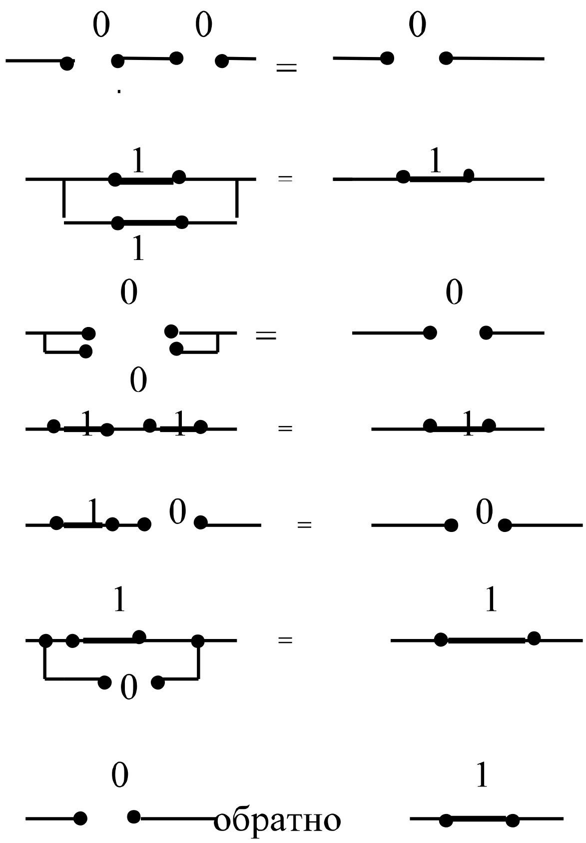
$0+0=0$

$1 \cdot 1 = 1$

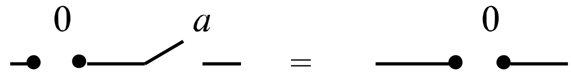
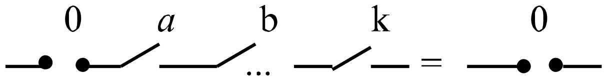
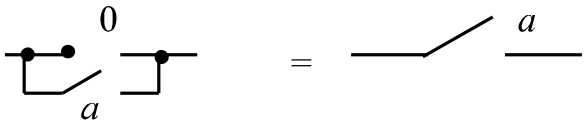
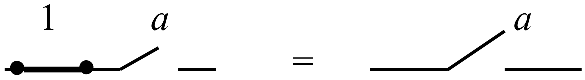
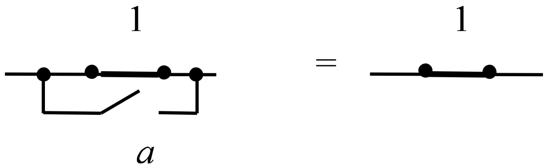
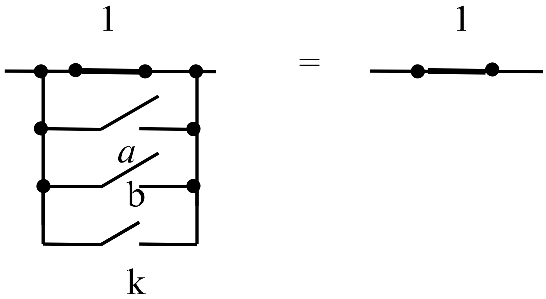
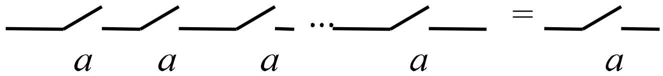
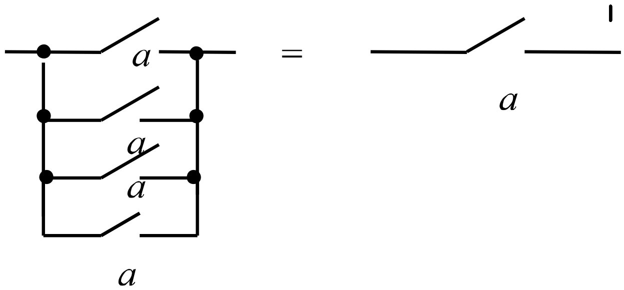
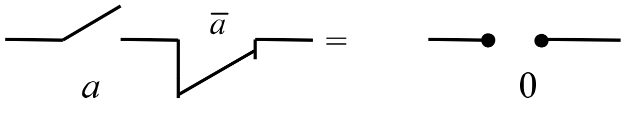
$1 \cdot 0 = 0$

$1+0=1$

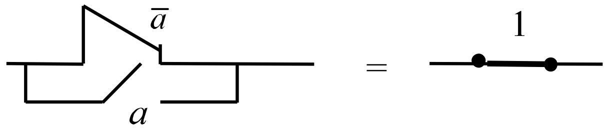
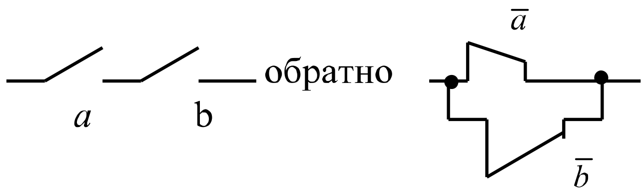
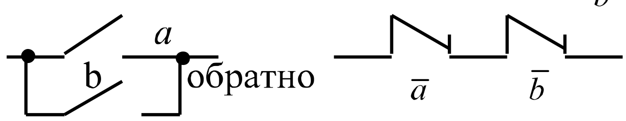
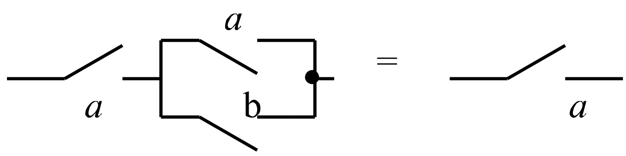
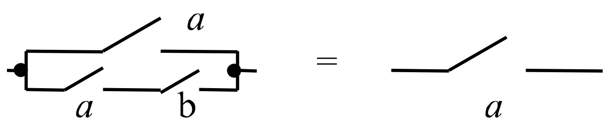
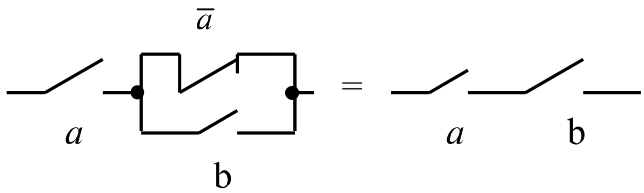
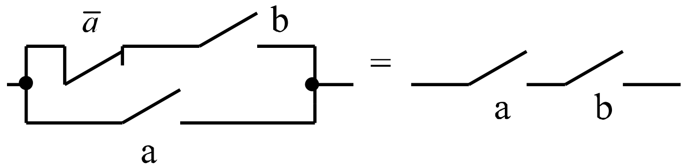
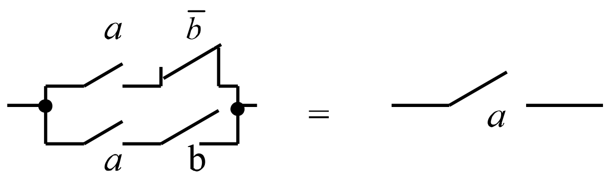
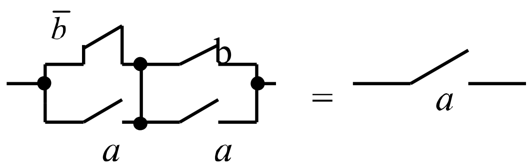
$\overline{0} = 1$



Продолжение таблицы 1.

1	2	3	4
		$\overline{1} = 0$	
	2. Законы нулевого множества	$0\,a = 0$	
		$0\,a\,\mathrm{b}...\mathrm{k} = 0$	
		$0 + a = a$	
	3.Законы универсального множества	$1\,a = a$	
		$1 + a = 1$	
		$1+a+\mathrm{b}+...+\mathrm{k}=1$	
	4. Законы повторения	$a\,a\,a....a = a$	
		$a+a+a+...+a=a$	
	5.Законы дополнительности	$a \cdot \overline{a} = 0$	

Продолжение таблицы 1.

1	2	3	4
		$a+\bar{a}=1$	
	6.Законы инверсии	$\overline{a\cdot b}=\bar{a}+\bar{b}$	
		$\overline{a+b}=\bar{a}\cdot\bar{b}$	
	7.Законы поглощения	a (a+b) = a	
		a + ab = a	
		$a(\bar{a}+b)=a\cdot b$	
		$a+\bar{a}\cdot b=a+b$	
	8.Закон двойного отрицания	$\bar{\bar{a}}=a;\bar{\bar{0}}=0;\bar{\bar{1}}=1.$	
	9.Законы склеивания	$a\cdot b+a\cdot\bar{b}=a$	
		$(a+\bar{b})\cdot(a+b)=a$	

Следует напомнить, что контакты, обозначенные одинаковыми буквами, принадлежат одному реле, то есть они в идеализированном виде срабатывают одновременно. Поэтому не вызывает сомнения запись, например, закона дополнительности:

$$a \cdot \overline{a} = 0 \qquad a + \overline{a} = 1$$

Действительно, последовательно соединенные замыкающий (a) и размыкающий ($\overline{a}$) контакты одного и того же реле (**A**) всегда будут создавать разрыв цепи (**0**).

Параллельная цепь этих же контактов равносильна постоянной перемычке (шунту) с проводимостью **1**.

Очень полезными для анализа и синтеза СЛУ являются законы инверсии:

$$\overline{a\,b} = \overline{a} + \overline{b}; \quad \overline{a + b} = \overline{a}\,\overline{b}$$

Эти законы легко доказать методом перебора всех возможных комбинаций переменных a и b . Если окажется, что для каждой комбинации переменных логические функции совпадут, то они равносильны. Например, рассмотрим первый закон инверсии, для чего составим таблицу состояний, в которой число различных комбинаций входных сигналов равно четырем (первый и второй столбцы), а в остальных столбцах таблицы приведены результаты элементарных логических операций. Таблица состояний представлена в таблице 2.

Таблица 2.

Таблица состояний

1	2	3	4	5	6	7
a	b	ab	$\overline{a \cdot b}$	$\overline{a}$	$\overline{b}$	$\overline{a} + \overline{b}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Столбцы третий, пятый и шестой - вспомогательные, содержащие результаты промежуточных вычислений. Как видно из таблицы значения ab и $a + b$ полностью совпадали для каждой комбинации переменных a и b.

Законы инверсии справедливы для любого числа переменных, причем представленных как в нормальной, так и инверсной форме :

$$\overline{\overline{a} \cdot \overline{b} \cdot \overline{c}} = \overline{\overline{a}} + \overline{\overline{b}} + \overline{\overline{c}} = a + b + c \;; \qquad \overline{a + 0} = \overline{a} \cdot \overline{0} = \overline{a} \cdot 1 = \overline{a}$$

$$\overline{\overline{a} + \overline{b} + \overline{c}} = \overline{\overline{a}} \cdot \overline{\overline{b}} \cdot \overline{\overline{c}} = a \cdot b \cdot c; \qquad \overline{a \cdot 1} = \overline{a} + \overline{1} = \overline{a} + 0 = \overline{a}$$

$$\overline{a + b \cdot c + \overline{d}} = \overline{a} \cdot \overline{b \cdot c} \cdot \overline{\overline{d}} = \overline{a}\left(\overline{b} + \overline{c}\right)d \;; \qquad \overline{\overline{a \cdot b} \cdot c} = \overline{\overline{ab}} + \overline{c} = ab + \overline{c} \,.$$