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:

$$N_{комб} = 2^n$$

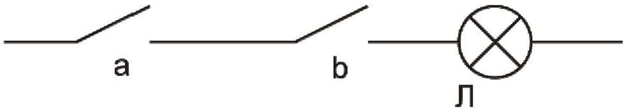
Все действия над переменными в бинарной алгебре выполняются с помощью следующих основных операций, которые наглядно иллюстрируются соответствующими релейно-контактными схемами.

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.

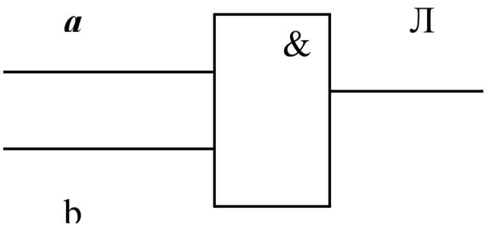
а) Релейно-контактный вариант реализации:



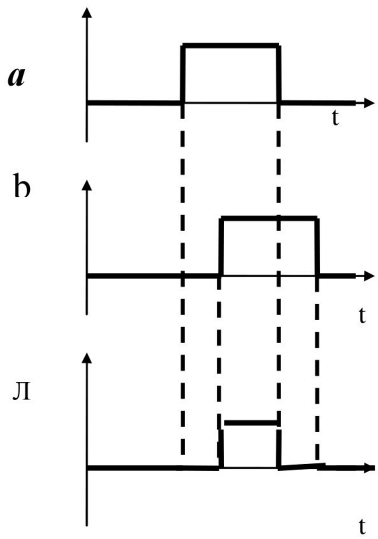
б). Таблица состояний:

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.

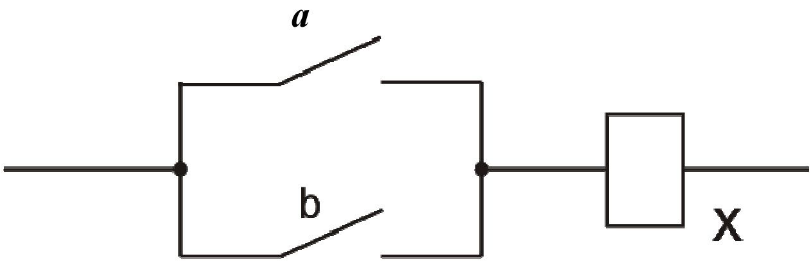
Применяются и другие обозначения операции логического умножения:

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

Логическое сложение

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

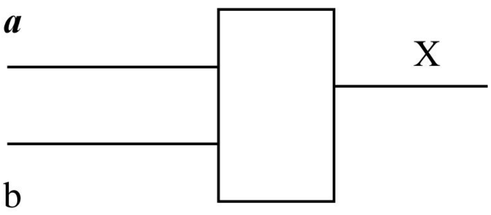
а) Релейно-контактный вариант реализации:



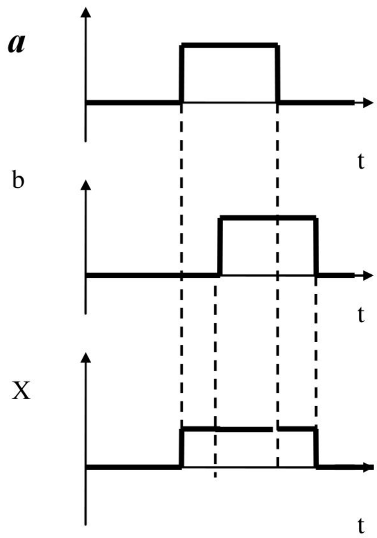
б) Таблица состояний:

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

в) Обозначение бесконтактного элемента:



г). Временные диаграммы:



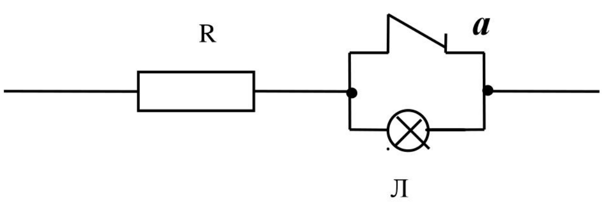
Очевидно, что катушка реле **X** получит питание, если замкнуть контакты или *a,* или *b*, или оба вместе.

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***).

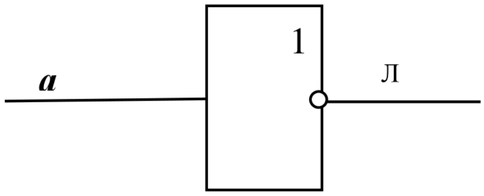
а) Релейно-контактный вариант реализации:



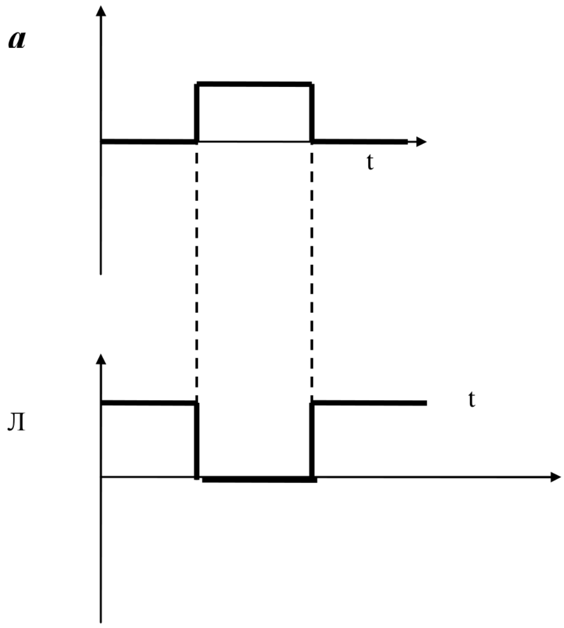
б) Таблица состояний:

a	Л
0	1
1	0

в) Обозначение бесконтактного элемента:



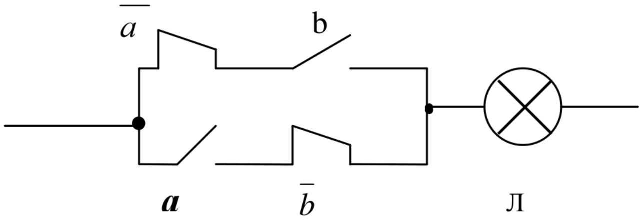
г) Временные диаграммы:



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

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

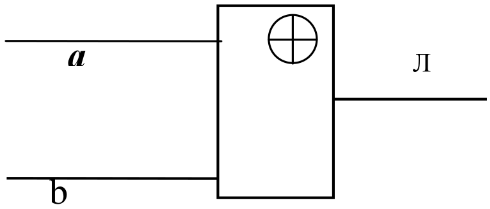
а) Релейно-контактный вариант реализации:



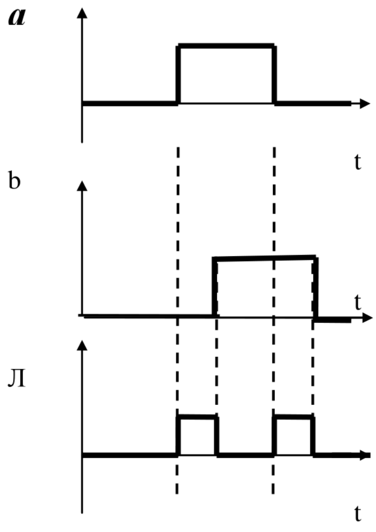
б) Таблица состояний:

a	b	Л
0	0	0
0	1	1
1	0	1
1	1	0

в) Обозначение бесконтактного элемента:



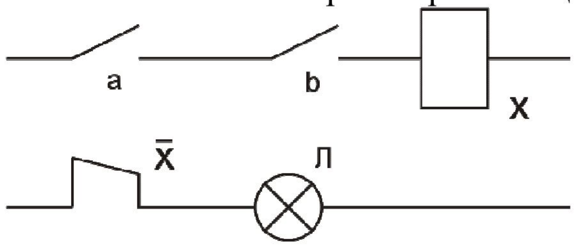
г) Временные диаграммы:



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

$$Л = \overline{a \cdot b}$$

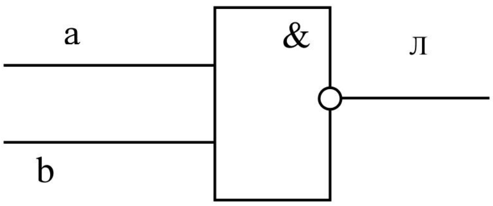
а) Релейно-контактный вариант реализации:



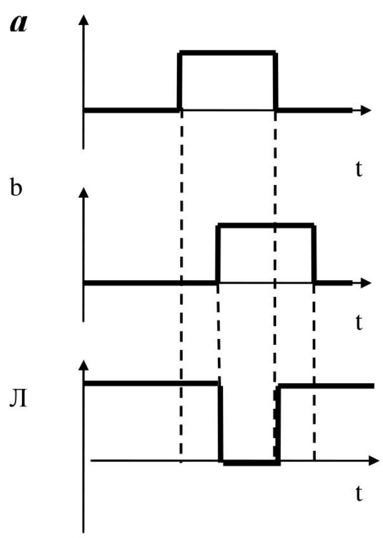
б) Таблица состояний:

a	b	Л
0	0	1
0	1	1
1	0	1
1	1	0

в) Обозначение бесконтактного элемента:



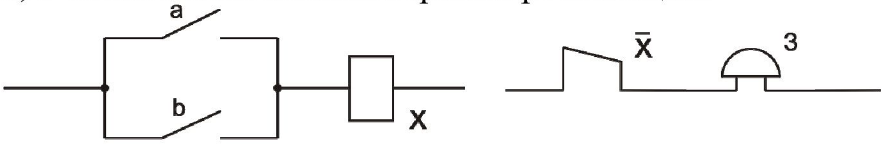
г) Временные диаграммы:



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

$$З = \overline{a + b}$$

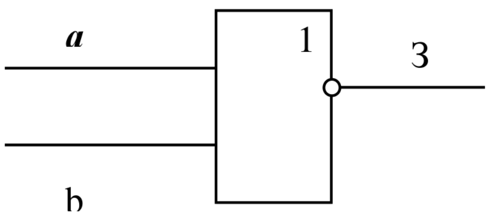
а) Релейно-контактный вариант реализации:



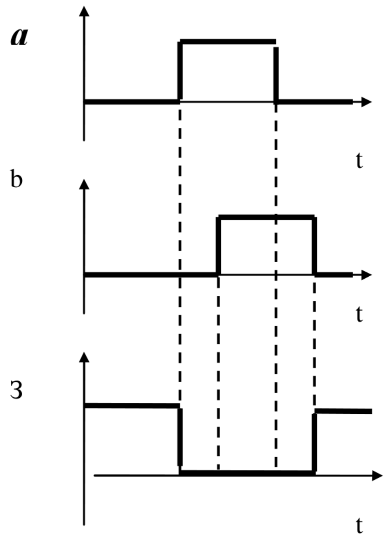
б) Таблица состояний:

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:

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

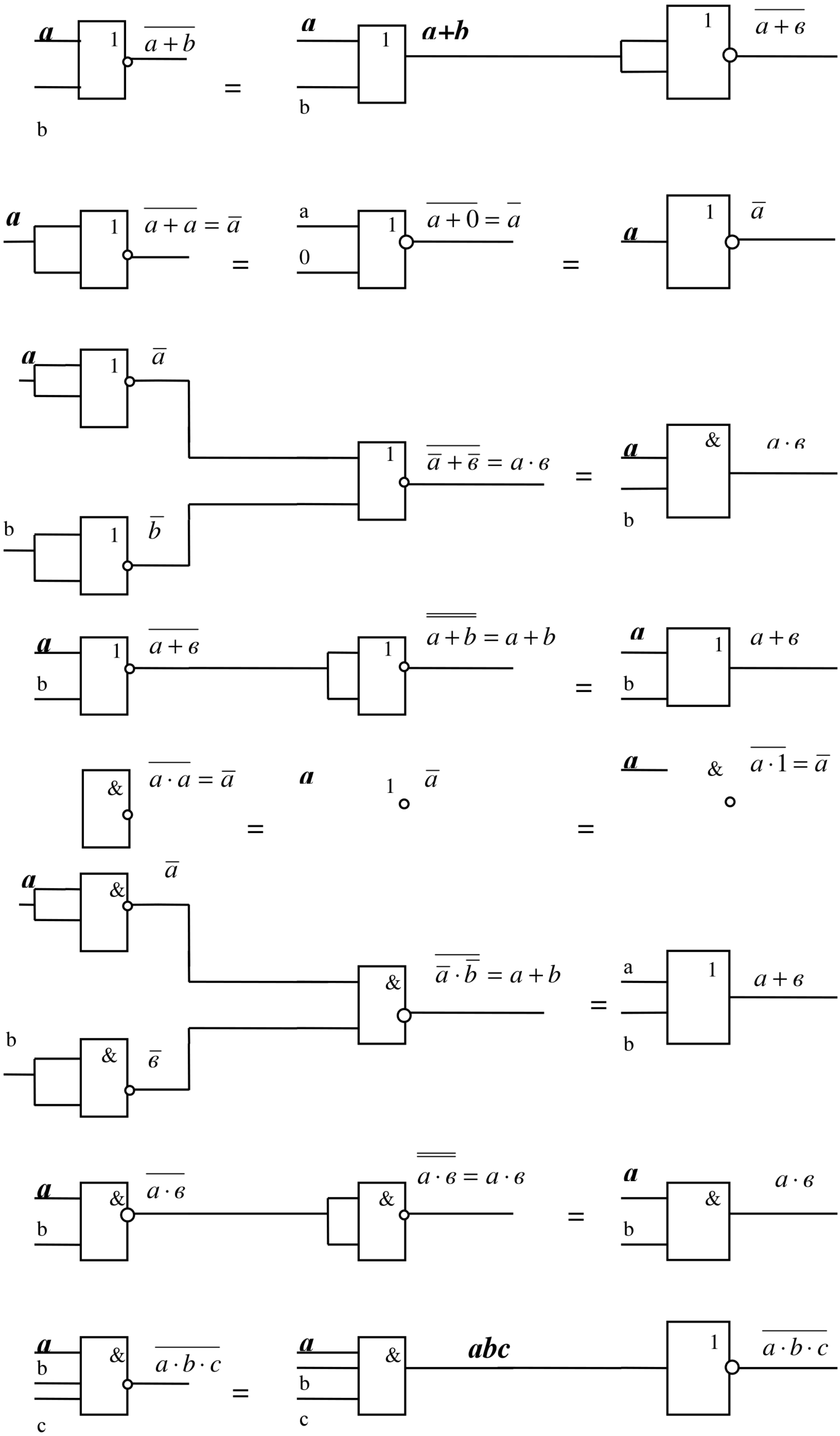


Рис. 1. Примеры элементарных логических операций

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;*

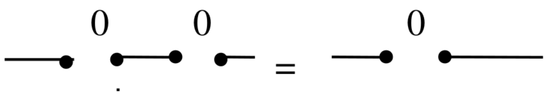
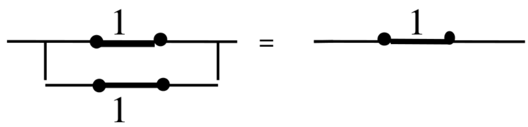
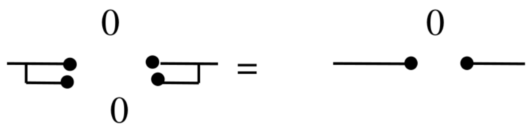
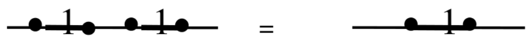
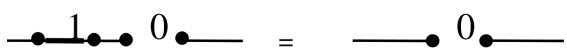
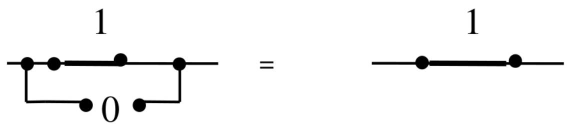
б) относительно логического сложения: *a b+c=(a+c)(b+c).*

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

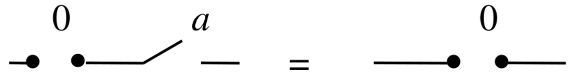
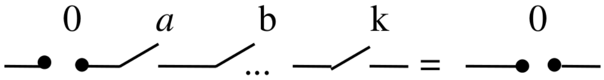
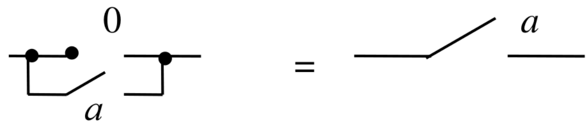
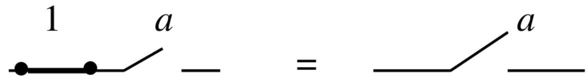
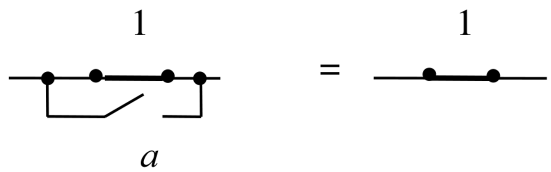
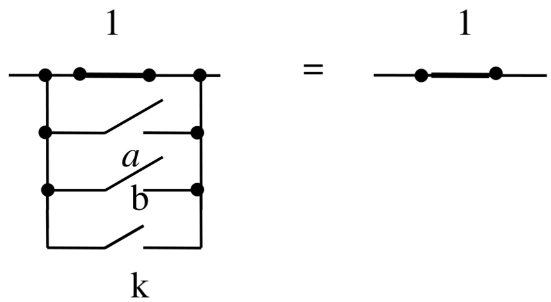
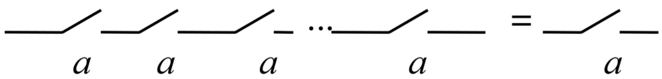
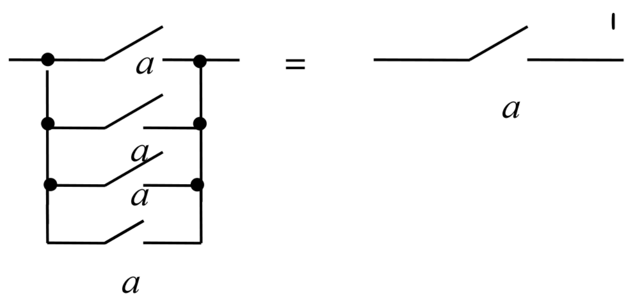
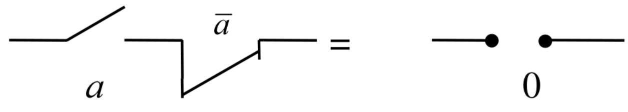
Но есть и специфические аксиомы, законы и теоремы, которые легко воспринимаются при рассмотрении соответствующих релейно-контактных схем (табл. 1).

Таблица 1.

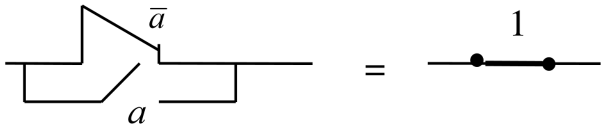
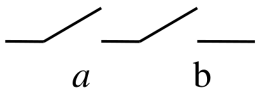
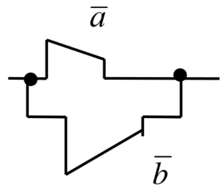
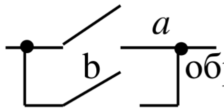
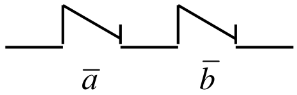
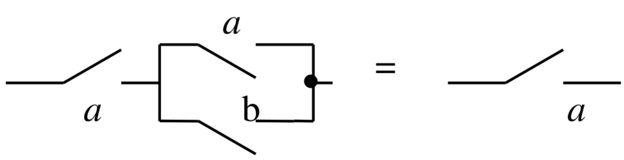
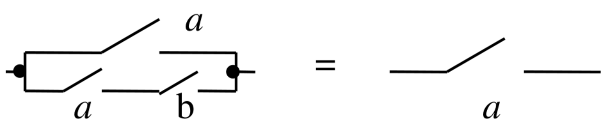
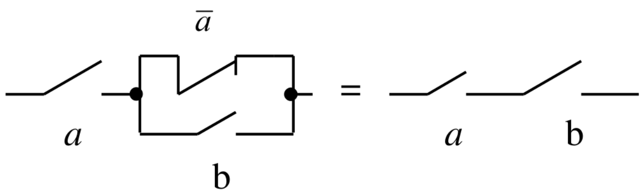
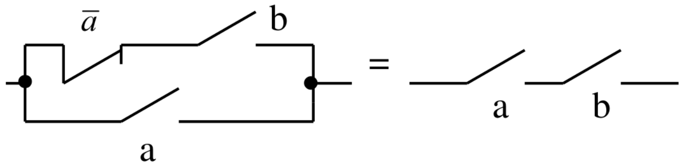
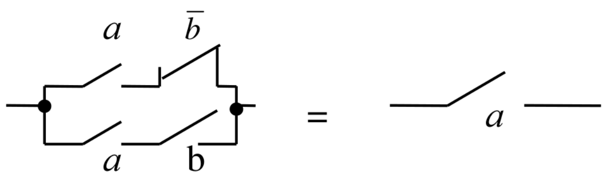
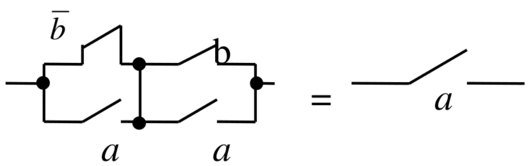
Основные аксиомы и законы алгебры логики

№ 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
		$a+\overline{a}=1$	
	6.Законы инверсии	$\overline{a\cdot b}=\overline{a}+\overline{b}$	 обратно 
		$\overline{a+b}=\overline{a}\cdot\overline{b}$	 обратно 
	7.Законы поглощения	a (a+b) = a	
		a + ab = a	
		$a(\overline{a}+b)=a\cdot b$	
		$a+\overline{a}\cdot b=a+b$	
	8.Закон двойного отрицания	$\overline{\overline{a}}=a;\overline{\overline{0}}=0;\overline{\overline{1}}=1.$	
	9.Законы склеивания	$a\cdot b+a\cdot\overline{b}=a$	
		$(a+\overline{b})\cdot(a+b)=a$	

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:

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

Действительно, последовательно соединенные замыкающий (a) и размыкающий ($\bar{a}$) 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:

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

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	$\overline{a \cdot b}$	$\bar{a}$	$\bar{b}$	$\bar{a} + \bar{b}$
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 :

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

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

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

2. ПРИМЕР ПРОЕКТИРОВАНИЯ СЛУ.

Система логического управления содержит три приёмных элемента A,B,C и два исполнительных элемента X,Y.

Алгоритм работы системы следующий:

Для элемента X (в нашем примере маломощный электродвигатель постоянного тока):

1. Элемент X срабатывает, если срабатывают A,B, но не срабатывает C.
2. Элемент X срабатывает, если срабатывают A,C, но не срабатывает B.
3. Элемент X срабатывает, если срабатывает A, но не срабатывают B и C.

Для элемента Y (в нашем примере сигнальная лампа):

1.Y срабатывает, если срабатывает A, но не срабатывают B и C;
2.Y срабатывает, если срабатывает B, но не срабатывают A и C;

2.1 РЕШЕНИЕ. Составляем основной элемент синтеза - таблицу состояния.

В таблице состояния рассматриваем все возможные комбинации состояний приемных элементов. Так как приемных элементов три - то возможное число комбинаций состояния равно восьми, т.е. имеем восемь строк в таблице состояний. Состояние исполнительных элементов записываем в соответствии с алгоритмом: если элемент срабатывает - ставится 1, в противном случае - 0.

Таблица 1.

	A	B	C	X	Y
1	0	0	0	0	0
2	0	0	1	0	0
3	0	1	0	0	1
4	0	1	1	0	0
5	1	0	0	1	1
6	1	0	1	1	0
7	1	1	0	1	0
8	1	1	1	0	0

Перейдем к составлению логической функции. Для этого составим частные условия срабатывания для элемента X:

$X_1 = a \cdot \overline{b} \cdot \overline{c}$;

$X_2 = a \cdot \overline{b} \cdot c$;

$X_3 = a \cdot b \cdot \overline{c}$.

Общие условия срабатывания запишем как дизъюнкцию частных условий срабатывания. Это означает, что элемент X сработает, если будет выполнено или одно частное условие срабатывания, или все частные условия, или их комбинация.

$$X = X_1 + X_2 + X_3 = a \cdot \bar{b} \cdot \bar{c} + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c} = a \cdot \bar{b}\left(\underbrace{c + \bar{c}}_{1}\right) + a \cdot b \cdot \bar{c} = a \cdot \bar{b} + a \cdot b \cdot \bar{c} = a\left(\bar{b} + b \cdot \bar{c}\right) = a\left(\bar{b} + \bar{c}\right).$$

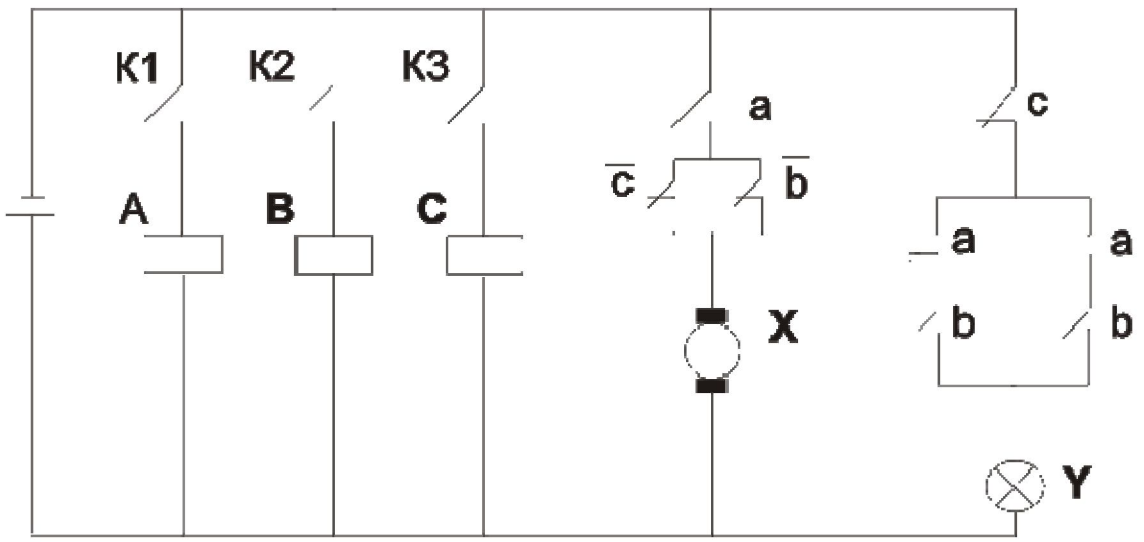
Аналогично составляем логическую функцию для элемента Y.

$Y_1 = \bar{a} \cdot b \cdot \bar{c}$;

$Y_2 = a \cdot \bar{b} \cdot \bar{c}$;

$Y = Y_1 + Y_2 = \bar{a} \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} = \bar{c}\left(\bar{a} \cdot b + a \cdot \bar{b}\right).$

Релейно-контактные варианты проектируемой системы логического управления имеют следующий вид:



Напоминаем, что если приемный элемент, например А, не сработал, то состояние его замыкающих контактов будет равносильно 0, а размыкающих – 1, т.е. a=0, $\bar{a}=1$.

При срабатывания А произойдет инверсия, т.е. a=1 и $\bar{a}=0$.

2.2. Проверим условия срабатывания и несрабатывания X.

Для первой строки таблицы состояний (A=0, B=0, C=0):

$$X = a\left(\bar{b} + \bar{c}\right) = 0(1+1) = 0$$

Для второй строки (A=0, B=0, C=1):

$$X = a\left(\bar{b} + \bar{c}\right) = 0(1+0) = 0$$

Для третьей строки (A=0, B=1, C=0):

$$X = a\left(\bar{b} + \bar{c}\right) = 0(0+1) = 0$$

Для четвертой строки (A=0, B=1, C=1):

$$X = a\left(\bar{b} + \bar{c}\right) = 0(0+0) = 0$$

Для пятой строки (A=1, B=0, C=0):

$$X = a\left(\bar{b} + \bar{c}\right) = 1(1+1) = 1$$

Для шестой строки (A=1, B=0, C=1):

$$X = a\left(\bar{b} + \bar{c}\right) = 1(1+0) = 1$$

Для седьмой строки (A=1, B=1, C=0):

$$X = a\left(\bar{b} + \bar{c}\right) = 1(0+1) = 1$$

Для восьмой строки (A=1, B=1, C=1):

$$X = a\left(\bar{b} + \bar{c}\right) = 1(0+0) = 0$$

Аналогично можно проверить эти условия для Y.

Бесконтактный вариант СЛУ выполним в базисе элементов И-НЕ. Для этого следует логические функции для X и Y записать в более удобном для выбранного базиса с использованием очевидных преобразований:

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

$$Y = \bar c\left(a \cdot \bar b + \bar a \cdot b\right) = \bar c \cdot a \cdot \bar b + \bar c \cdot \bar a \cdot b = \overline{\overline{\bar c \cdot a \cdot \bar b} \cdot \overline{\bar c \cdot \bar a \cdot b}}..$$

Наметим шины a,b и c , на которые будут поступать потенциалы от приемных элементов.

Легко представить бесконтактный вариант проектируемой СЛУ (рис.2).

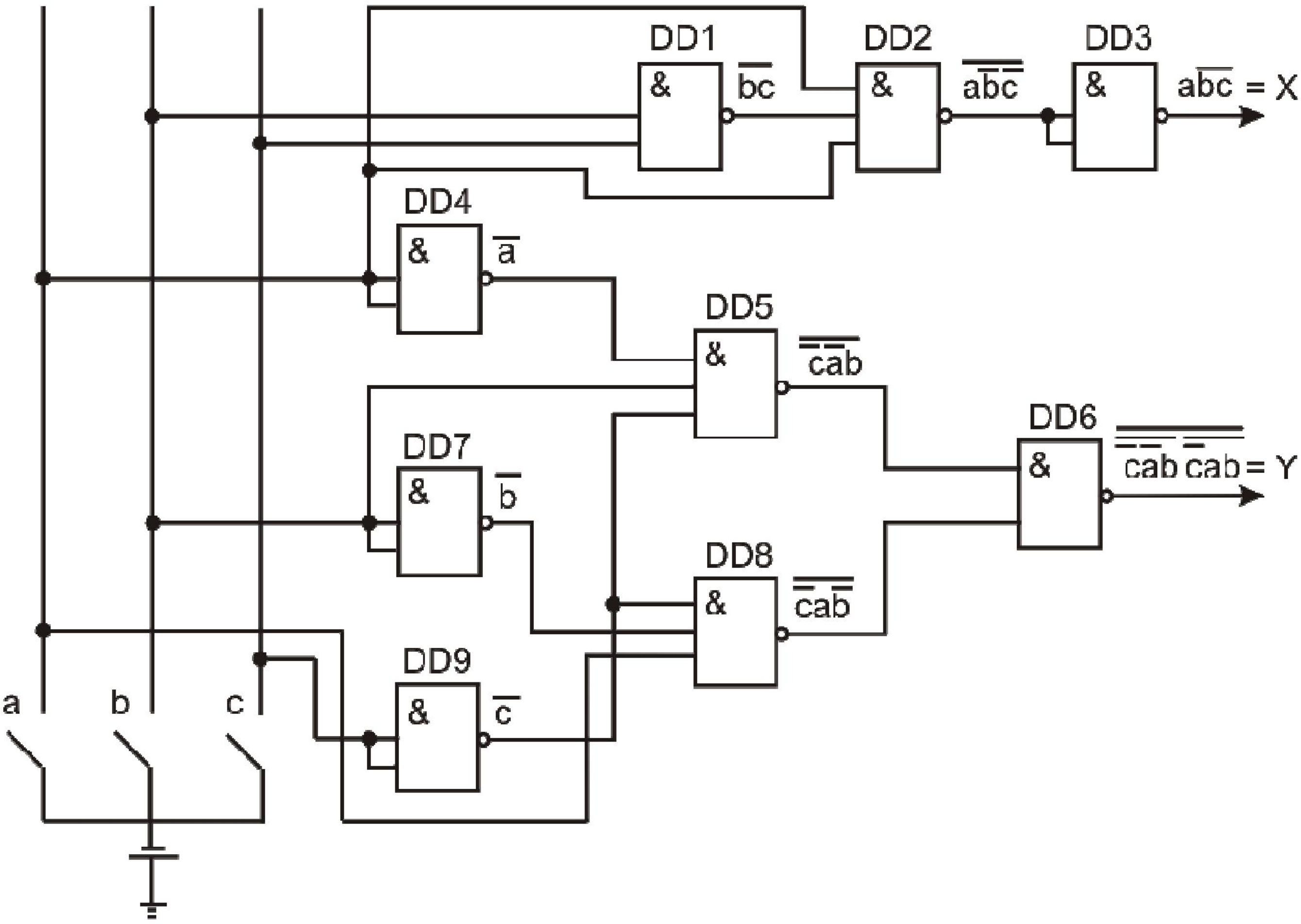


Рис. 2. Бесконтактная СЛУ в базисе И-НЕ.

Следовательно, для реализации этой схемы потребуется 9 элементов И-НЕ. Если бы потребовалось проектируемую СЛУ выполнить в базисе элементов ИЛИ-НЕ, то следовало бы выполнить следующие очевидные преобразования:

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

$$Y = \bar c \cdot \left(\bar a \cdot b + a \cdot \bar b\right) = \bar c \cdot \bar a \cdot b + \bar c \cdot a \cdot \bar b = \overline{c + a + \bar b} + \overline{c + \bar a + b} = \overline{\overline{\overline{c + a + \bar b} + \overline{c + \bar a + b}}}.$$

Для реализации функции X потребовалось бы 5 элементов, а для функции Y – 6 элементов ИЛИ-НЕ, имеющих по три входа.

Приложение 1

Две последние цифры номера зачетной книжки	Условия срабатывания исполнительных элементов	
	X	Y
00; 35; 70.	A, B, и C; A, C и D; B и C.	A и C; A и D; A и B; C; A,B и C.
01; 36; 71.	A, C и D; B, C и D; C и D.	A и D; A, B и C; A, C и D.
02; 37; 72.	B, C и D; A и D; A и C.	A,B и C; A и C; B,C и D.
03; 38; 73.	B и C; A, C и D; B, C и D.	B,C и D; A и D; C и D.
04; 39; 74.	A, B и C; A, C и D; A, B и D	B и D; A, B и D; B, C и D.
05; 40; 75.	C и D; B, C и D; A, C и D.	A и C; C и B,Д; B,C и D.
06; 41; 76.	A, C и D; A, B и D; B; C;	A и D; A и C; C, B и D; A,C и D
07; 42; 77.	A,B и C; B, C и D; C иDД;	D; A, B, C и D; A; A,B и C.
08; 43; 78.	A, B и C; B,C и D, A.	A; B; B и D; C и D; A,B и C
09; 44; 79.	A и B; A и C; B и C; C иD.	C и D; D; A и B; A и D.
10; 45; 80.	C и D; B,C и D.	A и C; A и D; B и C; B,C и D.

11; 46; 81.	C и D; B, C и D; A, B и D; A	A и B; C и D; A; B; C.
12; 47; 82.	C и D; B, C и D; B и C; A.	A и C; C и B; A.
13; 48; 83.	A, C и D; A, B и D; B; C.	B, C и D; A и D; C и D; A,C и D.
14; 49; 84.	A, B и C; A и C; B, C и D.	C и D; B, C и D; A, C и D.
15; 50; 85.	A, C иD; B, C и D; C и D; D.	A и D; A и C; A, B и C; A, C и D.
16; 51; 86.	A, C и D; A и C; B; D.	A и C; A и D; B и C.
17; 52; 87.	C и D; B, C и D; C и D; B.	B, C и D; A и D; A и C.
18; 53; 88.	A, B и D; A, B и C; C и D.	A и C; C и D; A; B.
19; 54; 89.	A, B и C; C и D; B и D.	A и C; B и D; A,B,C и D.
20; 55; 90.	A, B и C; A и C; C и D; A.	A и D; B, C, и D; A, B иD; A, B и C.
21; 56; 91.	A и C; B и D; C и D; D.	A, B и D; C иD; B и C.
22; 57; 92.	A, B и D; A; B; C; D.	C и D; B и C; A,B и D.

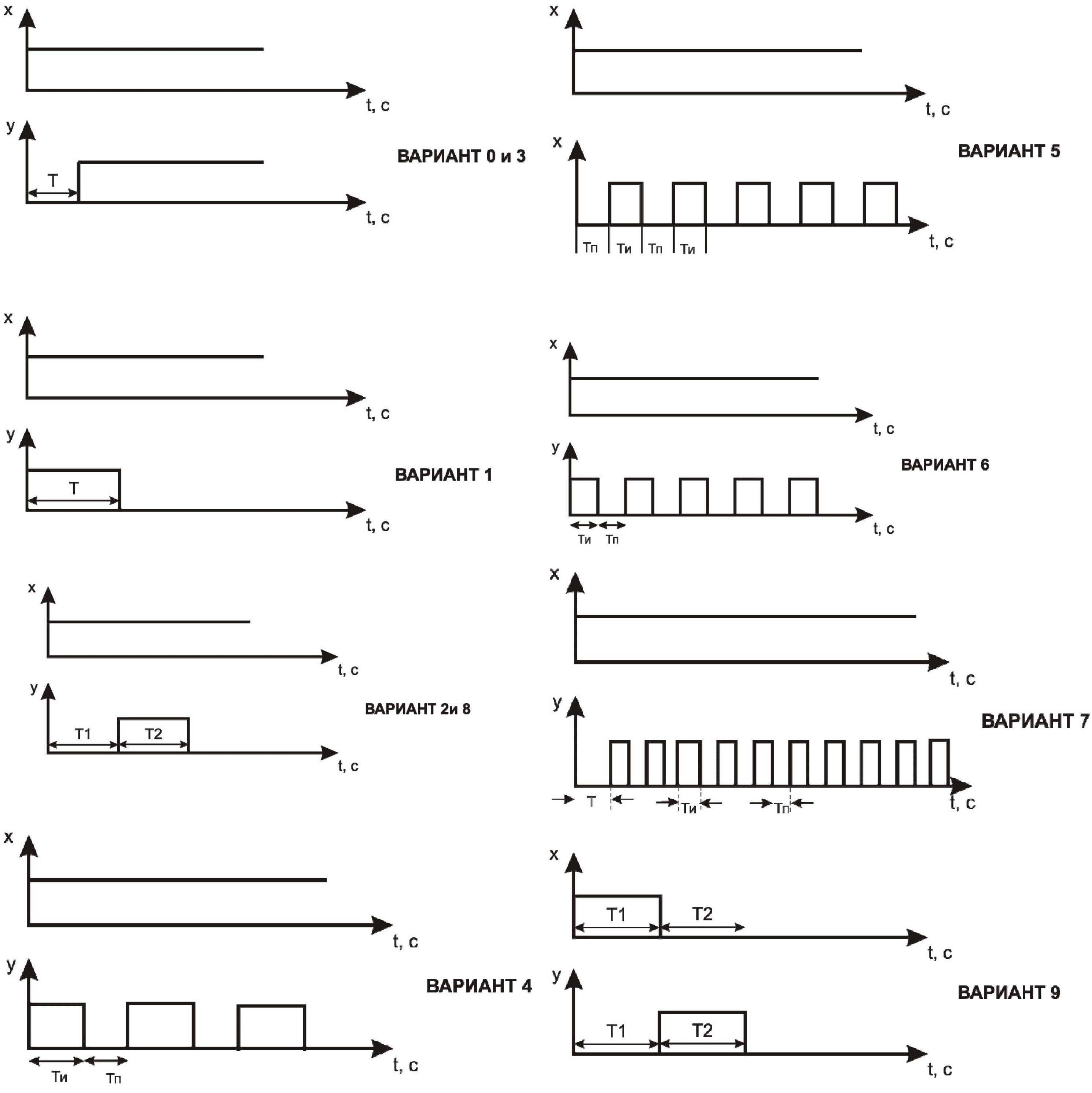
23; 58; 93.	A, B и C; B, C и D; C и D; C	A, B и C; C; B; B и D.
24; 59; 94.	A, B и D; A, C и D; D; B.	A и D; B и C; A, B и D; A.
25; 60; 95.	B, C и D; B и D; A и D; A.	A, C и D; A и B; B,C и D; B.
26; 61; 96.	A, B и C; A, C и D; B, C и D; C и D.	A и B; A и C; A и B; A и D, C и D.
27; 62; 97.	A, B и D; B, C и D; A и D; B и D.	A, B, C и D; B,C и D; C и D; D.
28; 63; 98.	A, B и C; B,C и D; C и D; C.	A и B; B и C; C и D; A; B.
29; 64; 99.	A, B и D; B,C и D; C и D; A.	A; A и B; B,C и D.
30; 65.	A,C и D; A и D; C и D; C.	A и B; A и C; A, B и D; A и D.
31; 66.	A, B, C и D; A и C; C и D; B	B и C; C и D; A; C.
32; 67.	A, C и D; A и B; B и D; A.	A и B; B и C; C и D; C; D.
33; 68.	A, B, C и D; A и B; B и C; C.	A, B и C; B, C и D; C и D; D; B и C
34; 69.	A, B и D; B, C и D; C и D; A и B.	A и B; B и C; A и D; A, C и D.

ПРИЛОЖЕНИЕ 2
К пункту 1.3. задания

Если последняя цифра зачетной книжки чётная-то систему выполнить на блоках И-НЕ, если нечётная, то на блоках ИЛИ-НЕ.

ПРИЛОЖЕНИЕ 3

Временные диаграммы работы исполнительных механизмов X и Y



Литература:

1. Петров И.В. Программируемые контроллеры. Стандартные языки и приемы прикладного программирования. М.: Солон-Пресс. – 2004. – 253с.
2. Парр Э.Программируемые контроллеры. М.: Бином. – 2007. – 516с.
3. Карпов Ю.Г. Теория автоматов. Учебник для вузов. М.: ПИТЕР. – 2002. – 206с.
4. Калабеков В.А. Цифровые устройства и микропроцессорные системы./ В.А.Калабеков - М.: Горячая линия – Телеком, 2000.-С. 336с.
5. Минаев И.Г., Самойленко В.В. Программируемые логические контроллеры. Практическое руководство для начинающего инженера. Ставрополь, - «АГРУС», 2009.-100с.
6. Минаев И.Г., Шарапов В.М., Самойленко В.В., Ушкур Д.Г.. Программируемые логические контроллеры в автоматизированных системах управления. Ставрополь, - «АГРУС»,2010 – 128с.