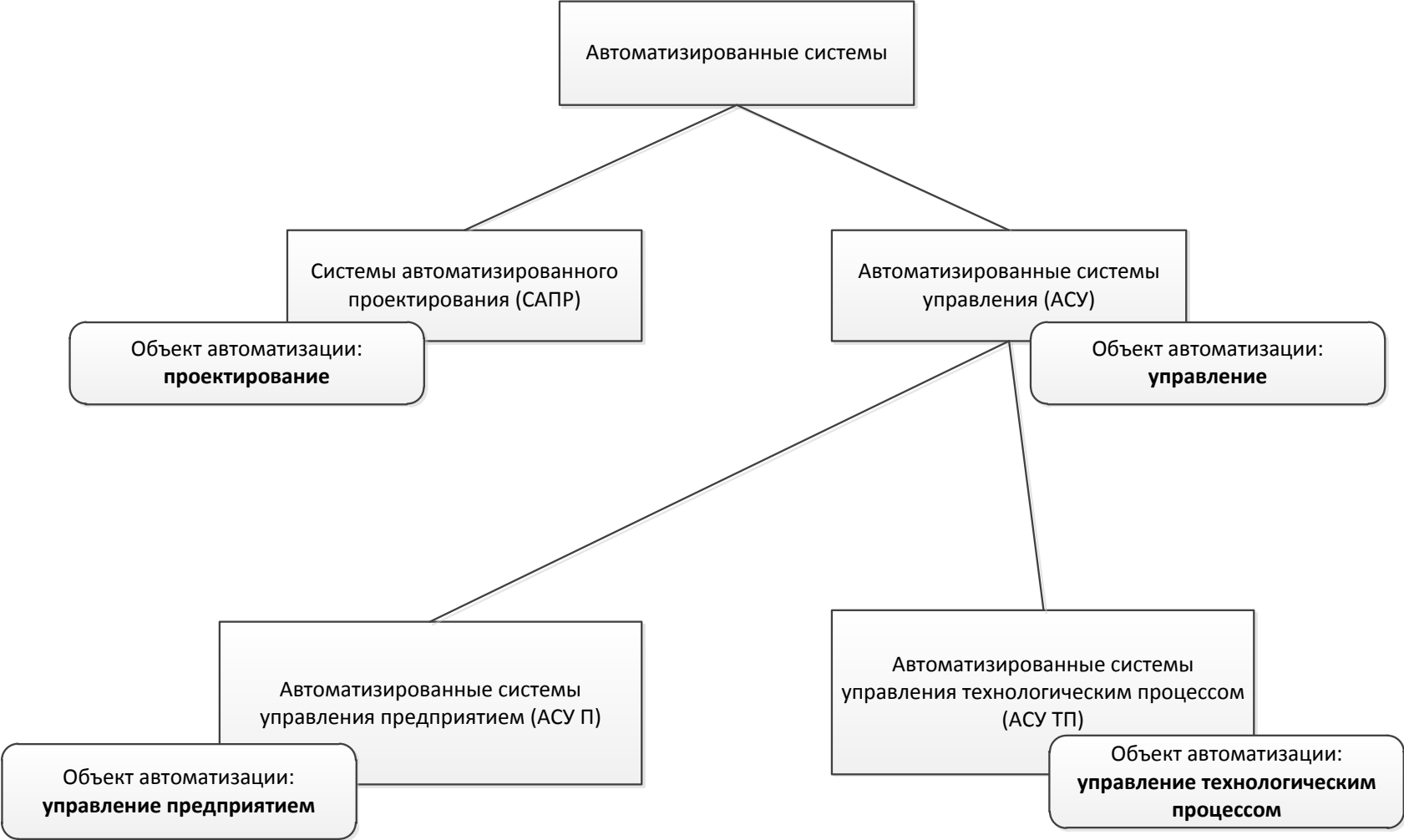


СРЕДСТВА АВТОМАТИЗАЦИИ УПРАВЛЕНИЯ

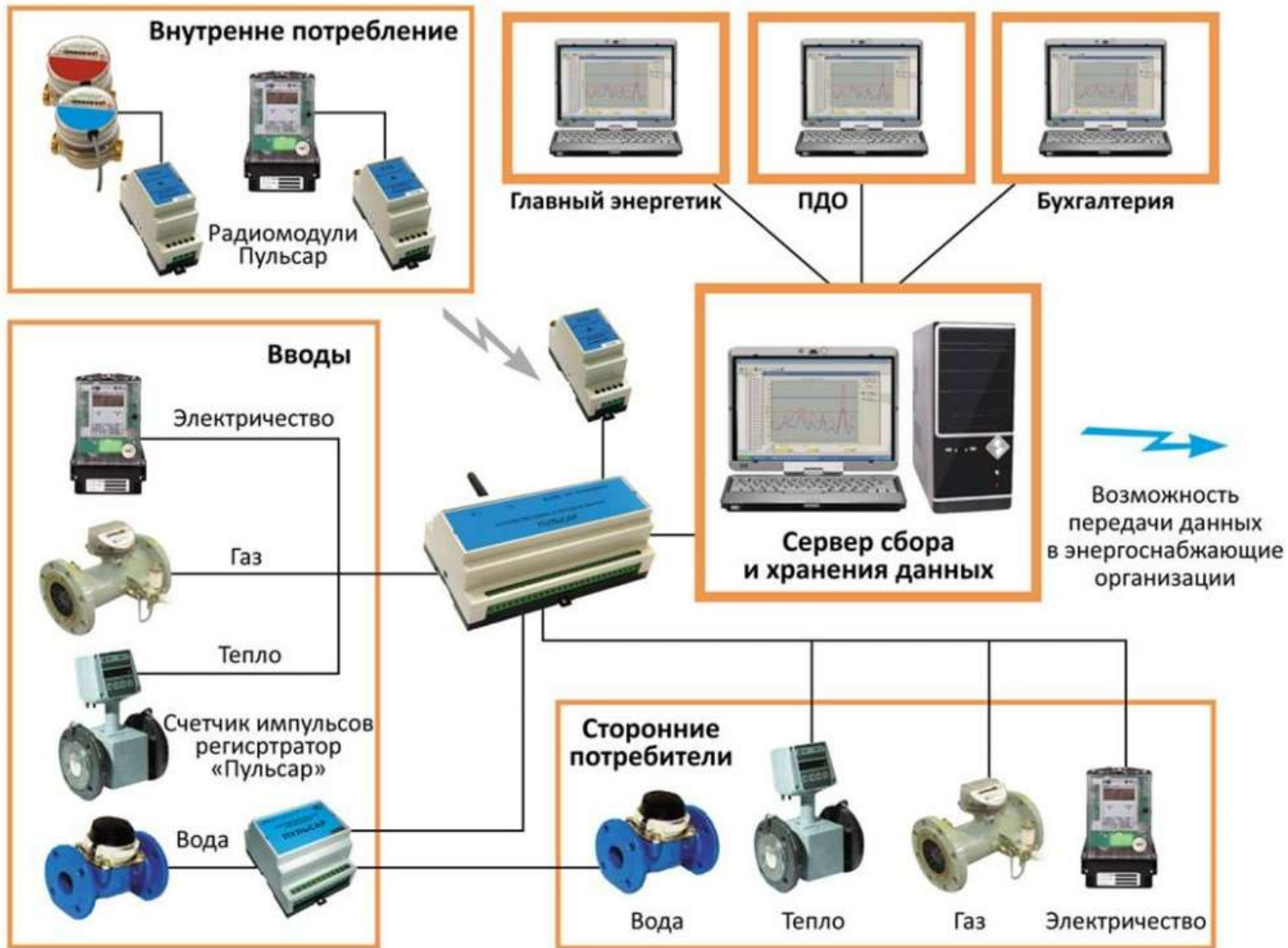
Введение в средства автоматизации и управления

Типы автоматизированных систем

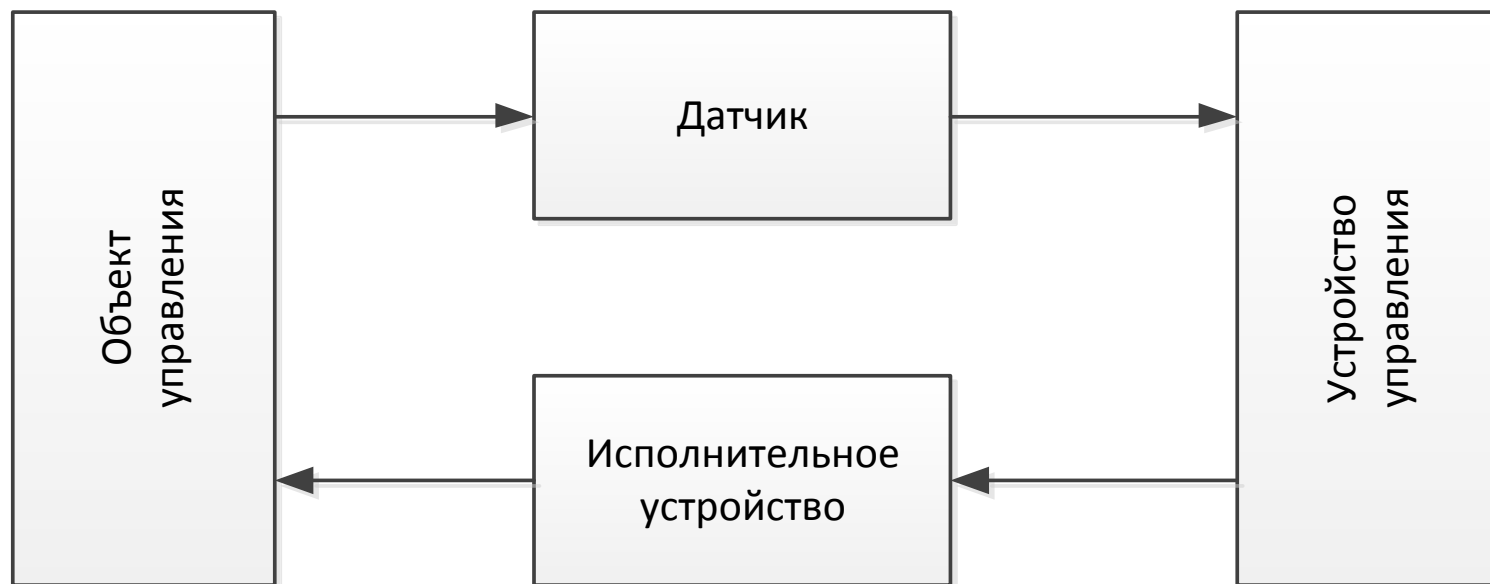




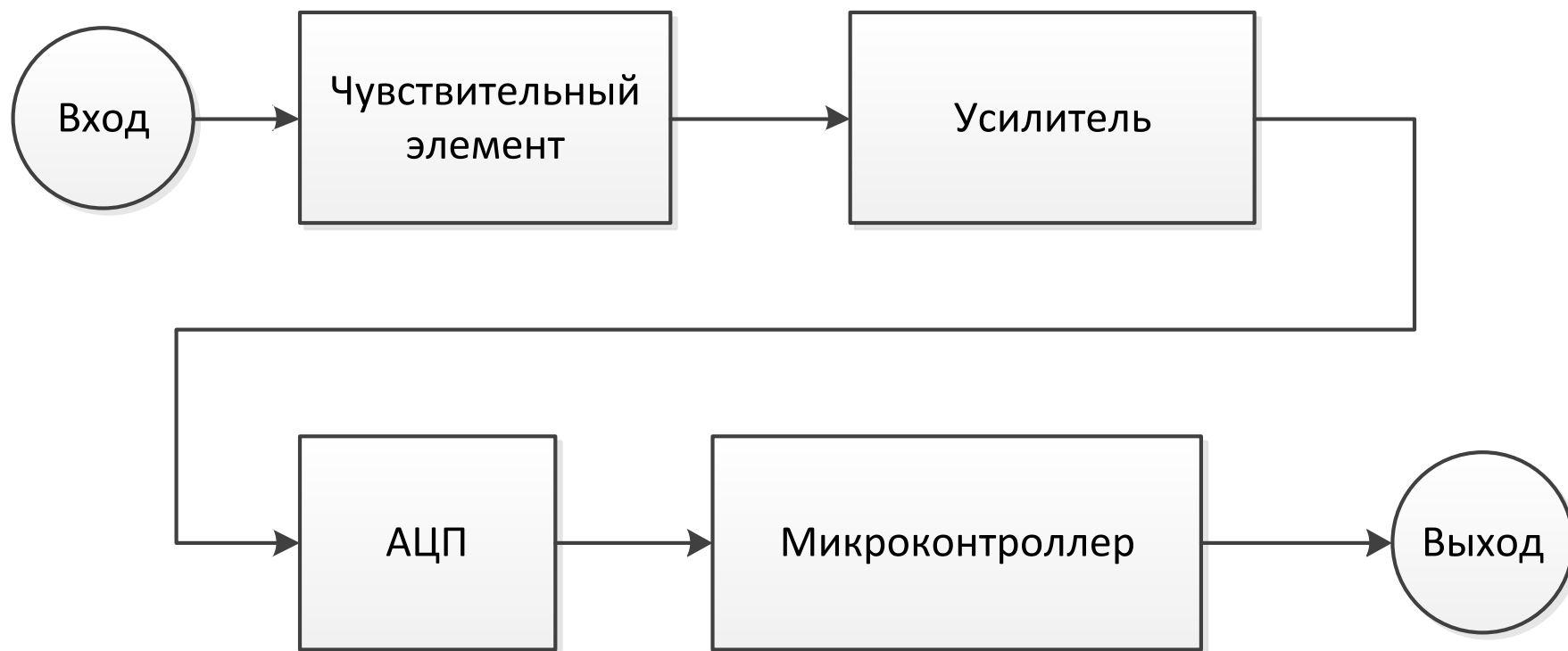
Промышленное предприятие



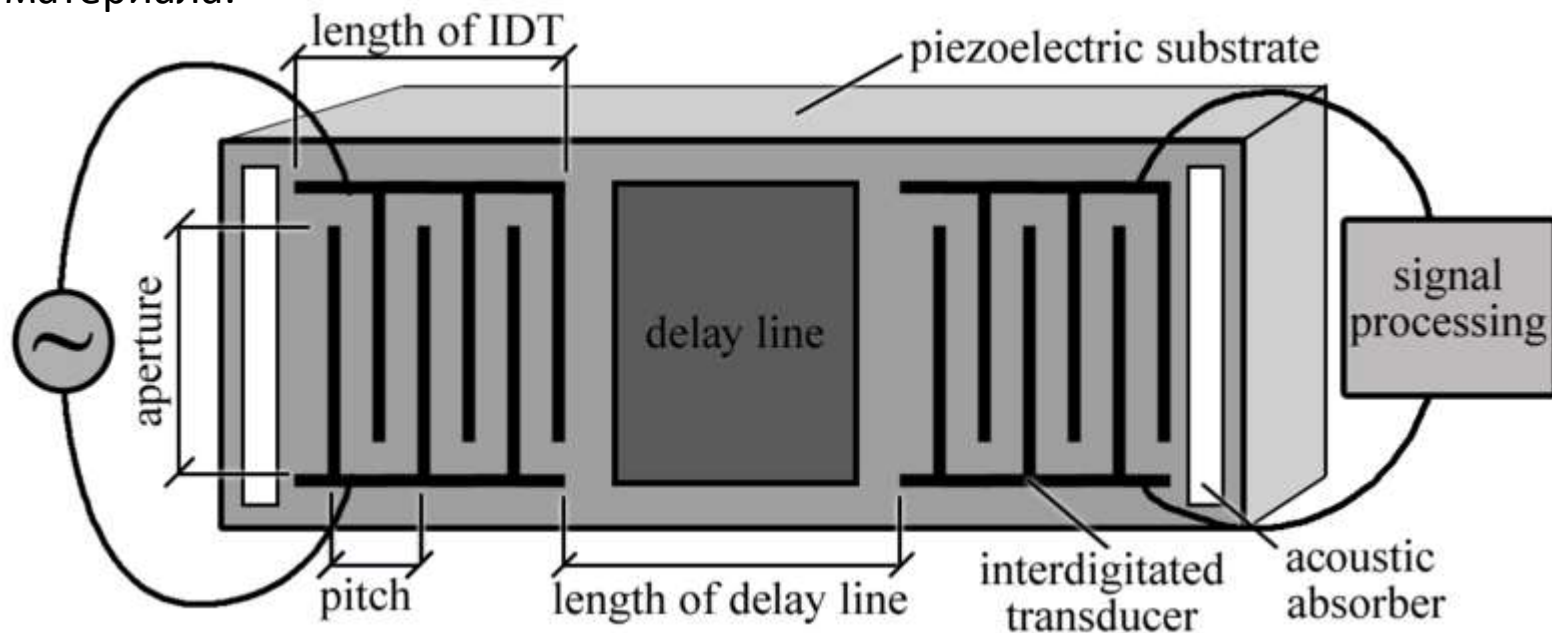
Система управления



Структура интеллектуального датчика



Датчики на поверхностных акустических волнах относятся к классу изделий микросистемотехники (MEMS). Они основаны на модуляции поверхностных акустических волн при взаимодействии с материалом помещённым вблизи поверхности датчика. Датчик преобразовывает входной электрический сигнал в механические волны, которые взаимодействуют с материалом. Затем устройство преобразовывает эту волну обратно в электрический сигнал. Изменения в амплитуде, фазе, частоте между входным и выходным электрическими сигналами можно использовать для определения свойств материала.





Серия T7335A



Серия T7335B



Серия T7335C

НАЗНАЧЕНИЕ

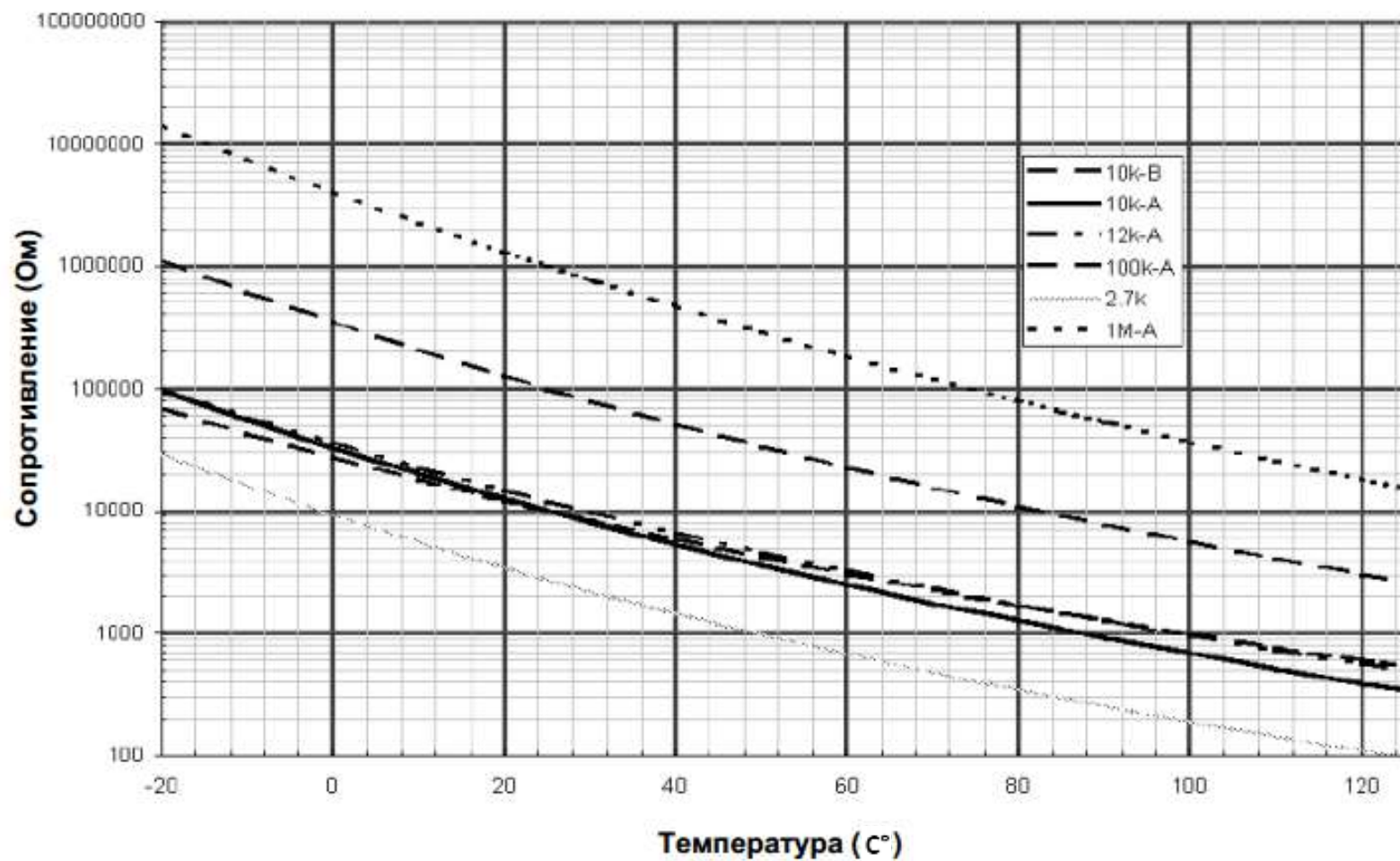
Термисторные датчики температуры T7335 предназначены для использования с электронными регуляторами водонагревателей.

Термисторные датчики температуры T7335 вырабатывают электрический сигнал, подаваемый в регулятор водонагревателя.

Термисторные датчики температуры T7335 представляют собой датчики прямого погружного типа.

Термисторные датчики температуры T7335 предназначены для использования в трубчатых гильзах с целью измерения температуры в накопительных баках и т.п.

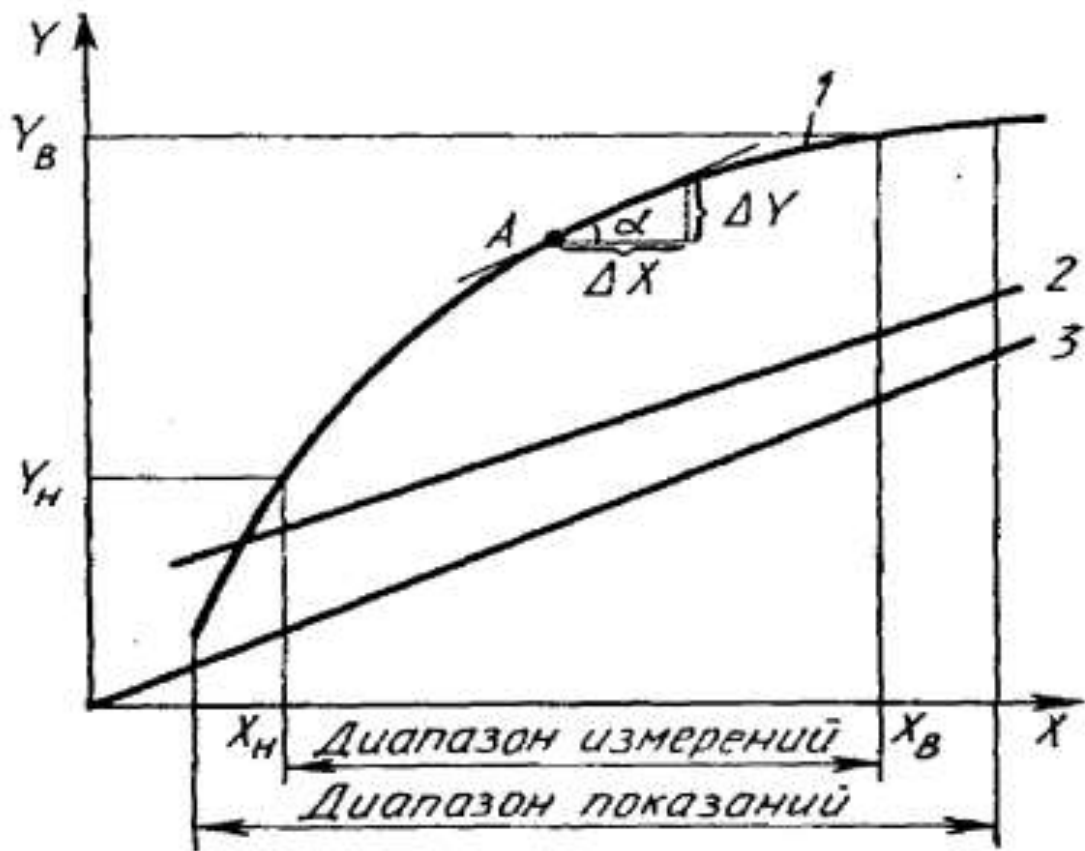
Датчики T7335D быстро монтируются и не требуют опорожнения системы при замене.

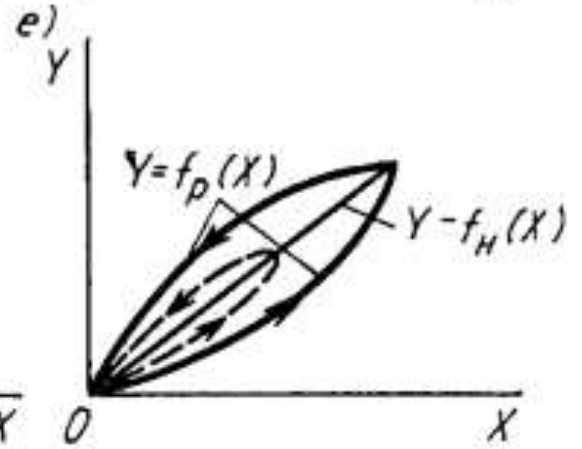
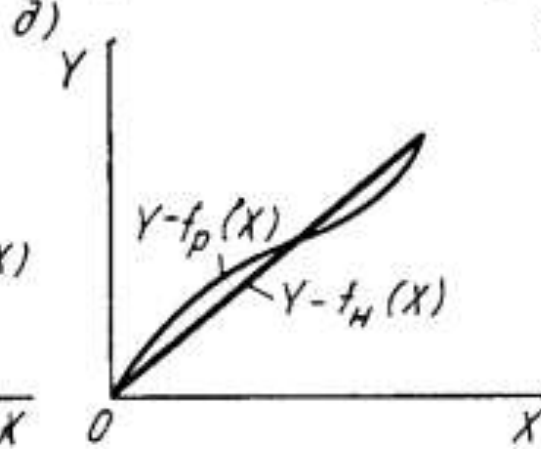
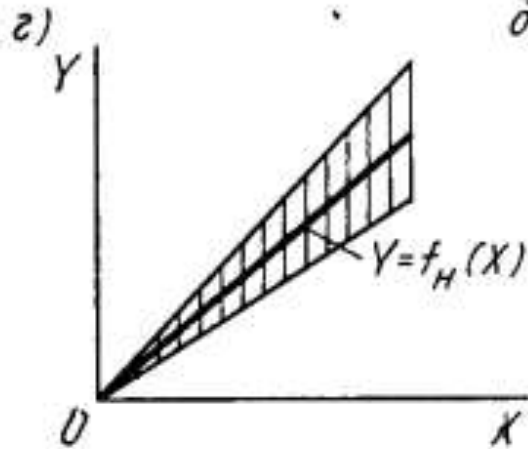
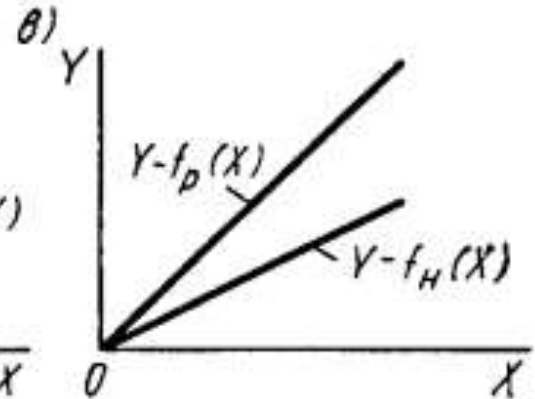
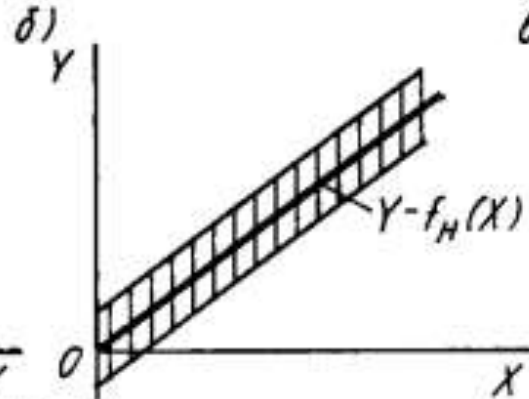
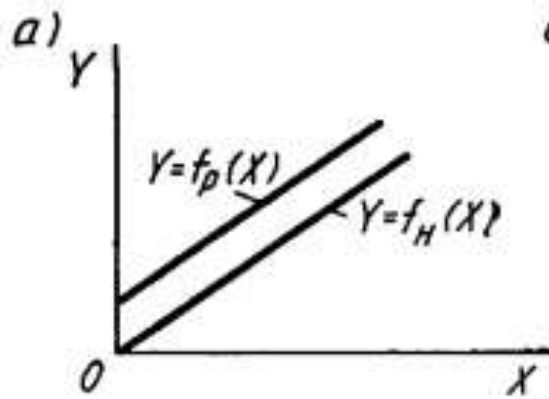


Номинальные характеристики сопротивления

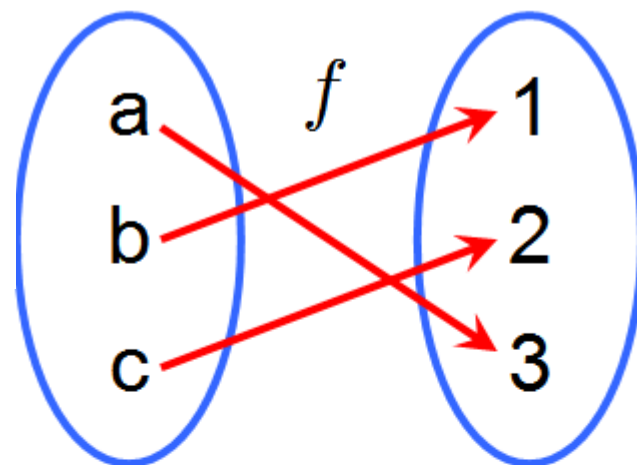
Таблица 1. Температурные характеристики термисторов 2.7к, 10к-А, 10к-В и 12к-А

Коэффициент Стейнхарда-Харта	Кривые							
	2,7 кОм		10к-А		10к-В		12к-А	
C	1,94567E-18		8,77547E-08		1,89916E-07		1,28318E-07	
B	2,51842E-04		2,34108E-04		2,52293E-04		2,40423E-04	
A	1,36421E-03		1,12924E-03		8,84204E-04		9,89458E-04	
f	1		1,01707E+00		1		1	
T [C]	R	$\alpha[1^{\circ}C]$	R	$\alpha[1^{\circ}C]$	R	$\alpha[1^{\circ}C]$	R	$\alpha[1^{\circ}C]$
-20	29,30 кОм	-6,2 %	98,66 кОм	-5,8 %	67,38 кОм	-4,8 %	98,97 кОм	-5,4 %
-10	16,14 кОм	-5,7 %	56,25 кОм	-5,4 %	42,14 кОм	-4,6 %	58,88 кОм	-5,0 %
0	9,292 кОм	-5,3 %	33,21 кОм	-5,1 %	27,06 кОм	-4,3 %	36,13 кОм	-4,7 %
10	5,561 кОм	-5,0 %	20,24 кОм	-4,8 %	17,82 кОм	-4,1 %	22,80 кОм	-4,5 %
20	3,446 кОм	-4,6 %	12,71 кОм	-4,5 %	12,00 кОм	-3,8 %	14,77 кОм	-4,2 %
25	2,746 кОм	-4,5 %	10,17 кОм	-4,4 %	9,92 кОм	-3,7 %	12,00 кОм	-4,1 %
30	2,205 кОм	-4,3 %	8,194 кОм	-4,3 %	8,251 кОм	-3,6 %	9,804 кОм	-4,0 %
40	1,451 кОм	-4,0 %	5,416 кОм	-4,0 %	5,786 кОм	-3,5 %	6,652 кОм	-3,8 %
50	980,1	-3,8 %	3,663 кОм	-3,8 %	4,132 кОм	-3,3 %	4,607 кОм	-3,6 %
60	677,8	-3,6 %	2,530 кОм	-3,6 %	3,000 кОм	-3,1 %	3,252 кОм	-3,4 %
70	478,9	-3,4 %	1,782 кОм	-3,4 %	2,213 кОм	-3,0 %	2,337 кОм	-3,2 %
80	345,1	-3,2 %	1,278 кОм	-3,2 %	1,656 кОм	-2,8 %	1,707 кОм	-3,1 %
85	295,0	-3,1 %	1,089 кОм	-3,2 %	1,441 кОм	-2,8 %	1,467 кОм	-3,0 %
90	253,2	-3,0 %	931,6 Ом	-3,1 %	1256,9 Ом	-2,7 %	1,266 кОм	-2,9 %
100	188,9	-2,9 %	690,0 Ом	-2,9 %	965,9 Ом	-2,6 %	952,3 Ом	-2,8 %
110	143,1	-2,7 %	518,5 Ом	-2,8 %	751,2 Ом	-2,5 %	726,0 Ом	-2,6 %
120	109,9	-2,6 %	395,0 Ом	-2,7 %	590,8 Ом	-2,3 %	560,4 Ом	-2,5 %
125	96,83	-2,5 %	346,4 Ом	-2,6 %	526,0 Ом	-2,3 %	494,6 Ом	-2,5 %



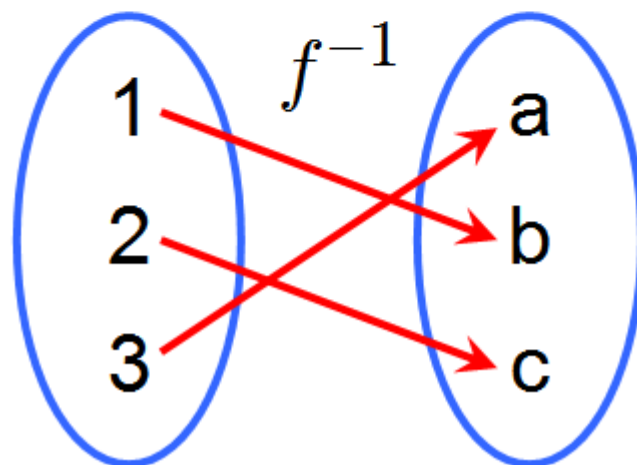


Обратная функция — функция, обращающая зависимость, выражаемую данной функцией. Например, если функция от x даёт y , то обратная ей функция от y даёт x . Обратная функция функции f обычно обозначается f^{-1} , иногда также используется обозначение f^{inv} .



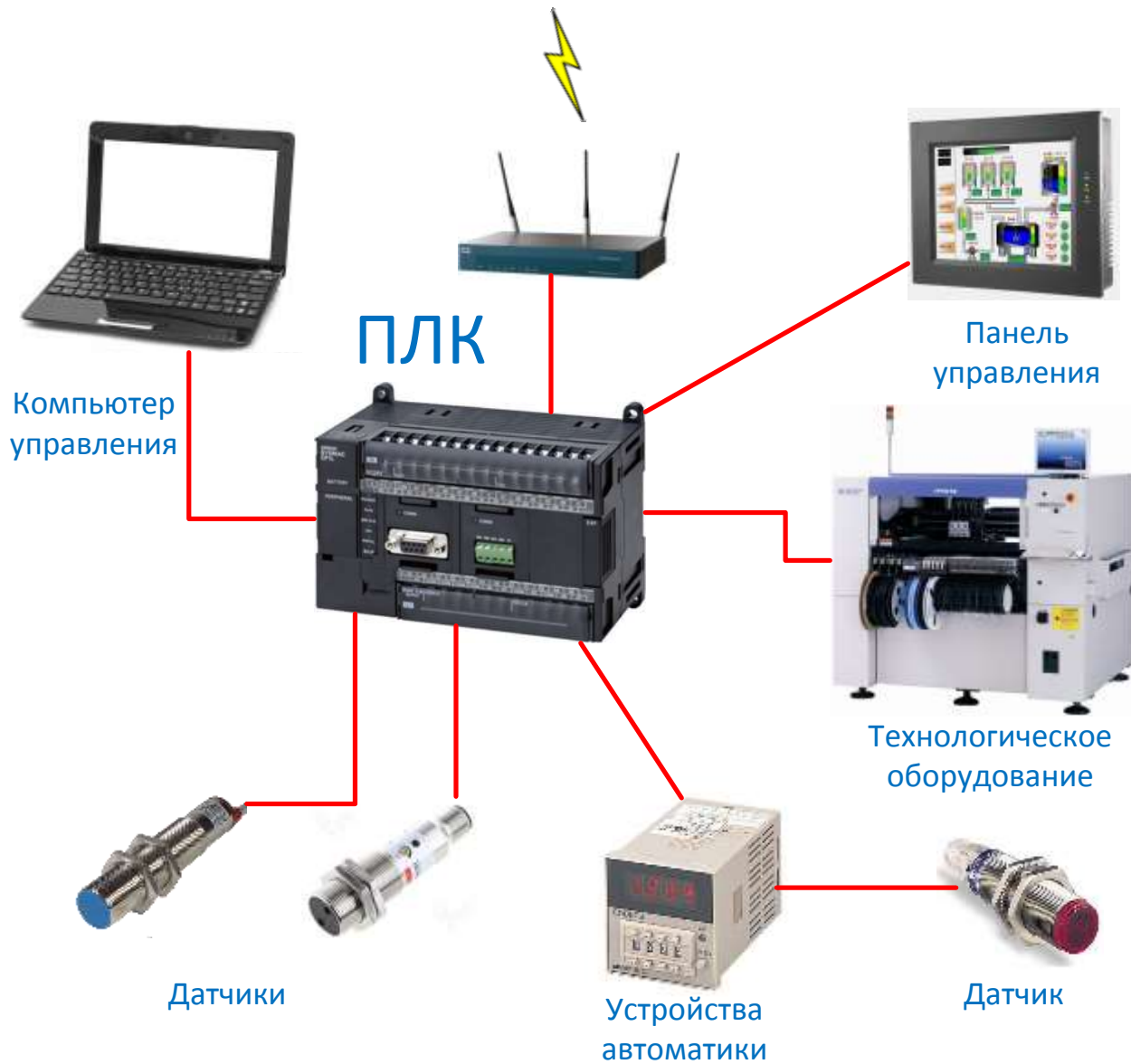
Функция $g : Y \rightarrow X$ является обратной к функции $f : X \rightarrow Y$, если выполнены следующие тождества:

- $f(g(y)) = y$ для всех $y \in Y$;
- $g(f(x)) = x$ для всех $x \in X$.



Программируемые логические контроллеры

Локальная вычислительная сеть предприятия



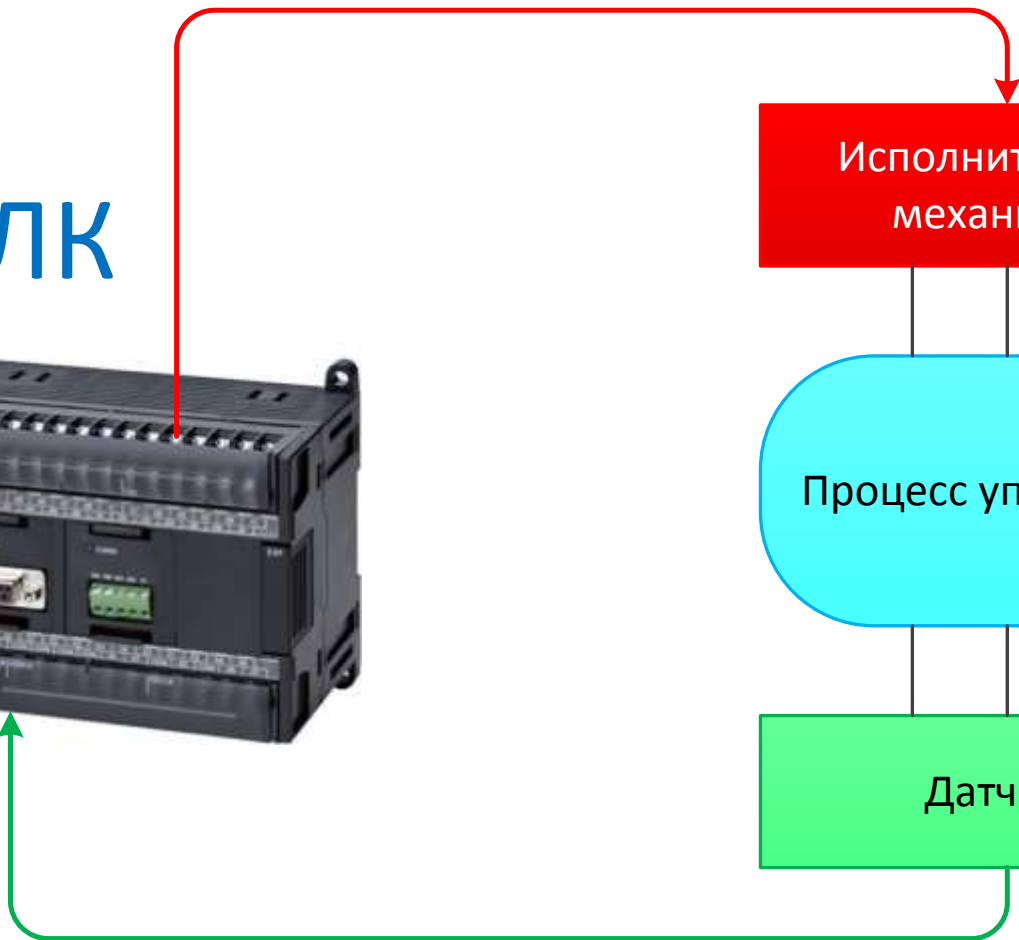
ПЛК

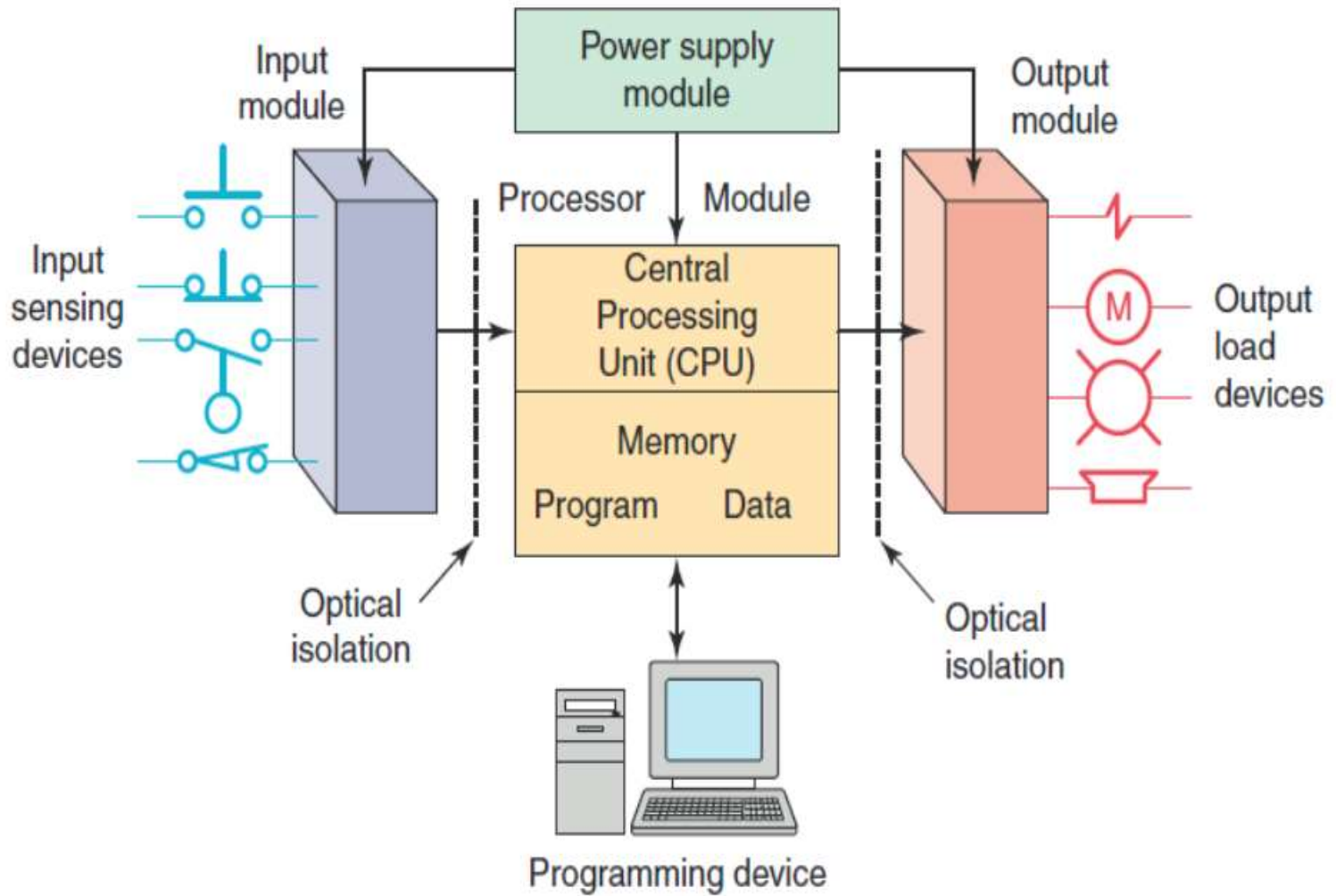


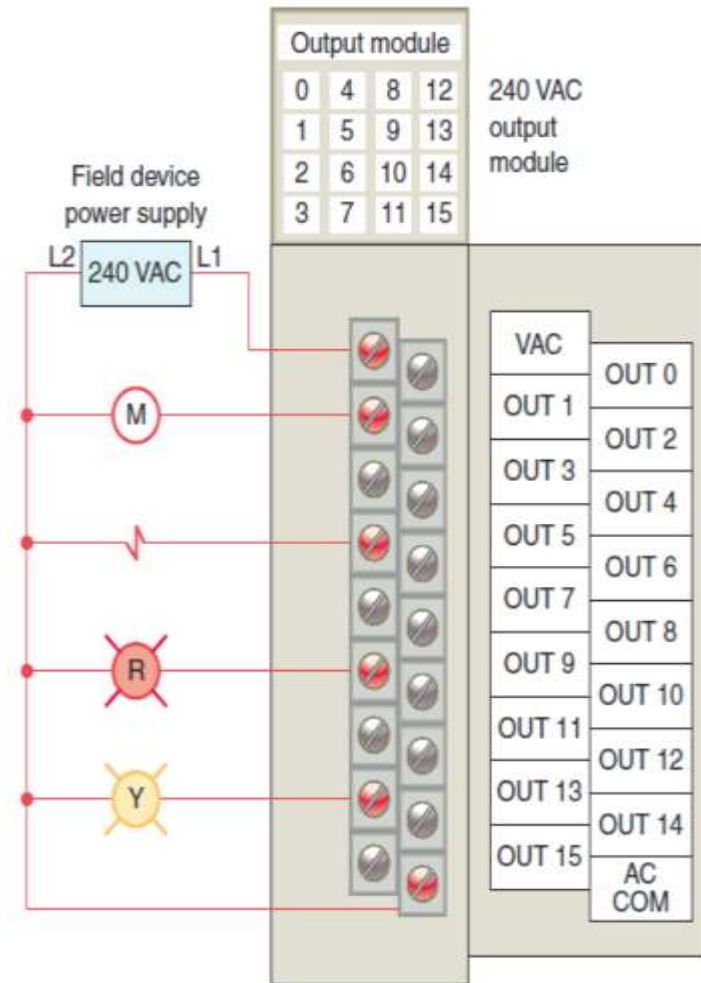
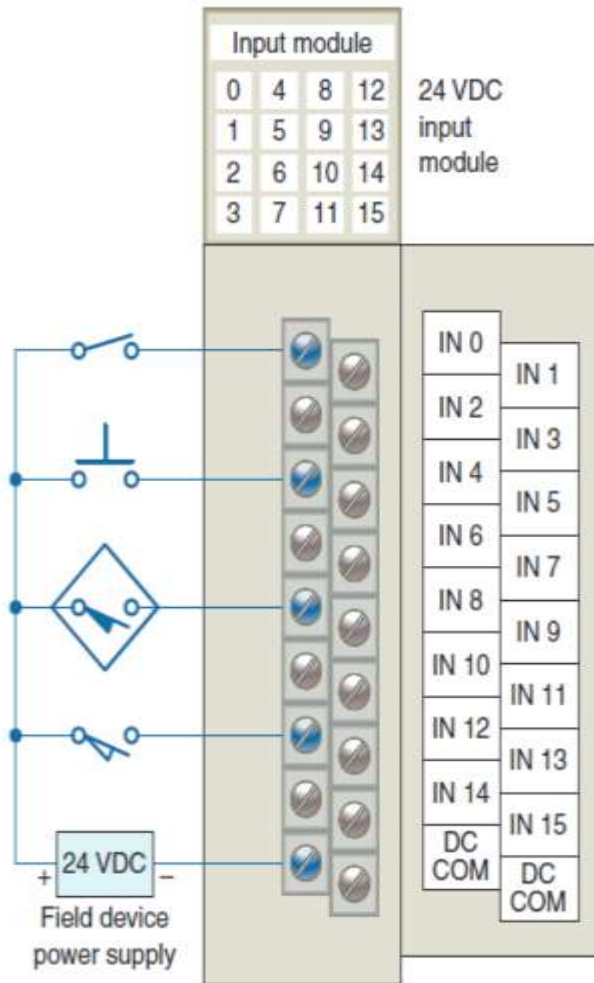
Исполнительные механизмы

Процесс управления

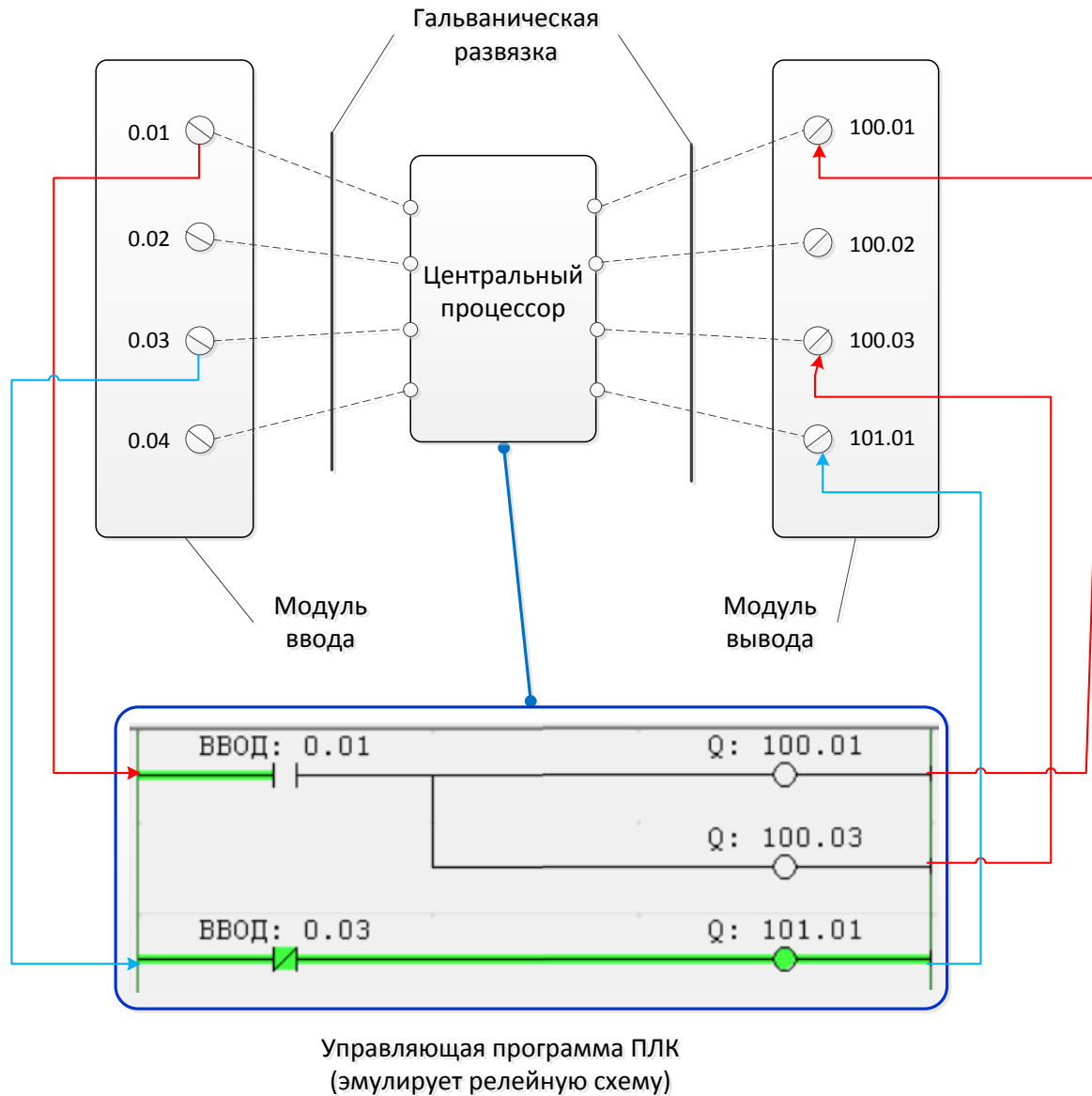
Датчики

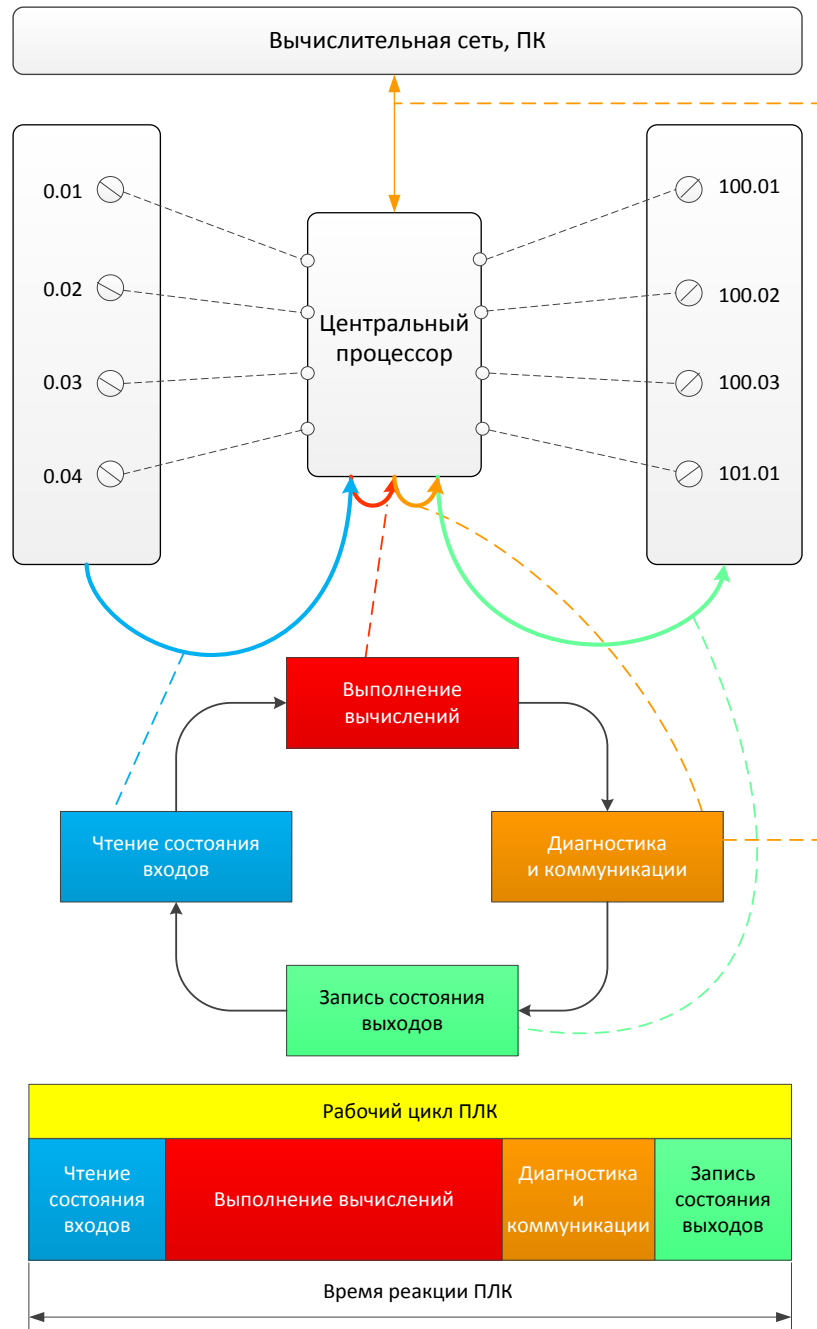






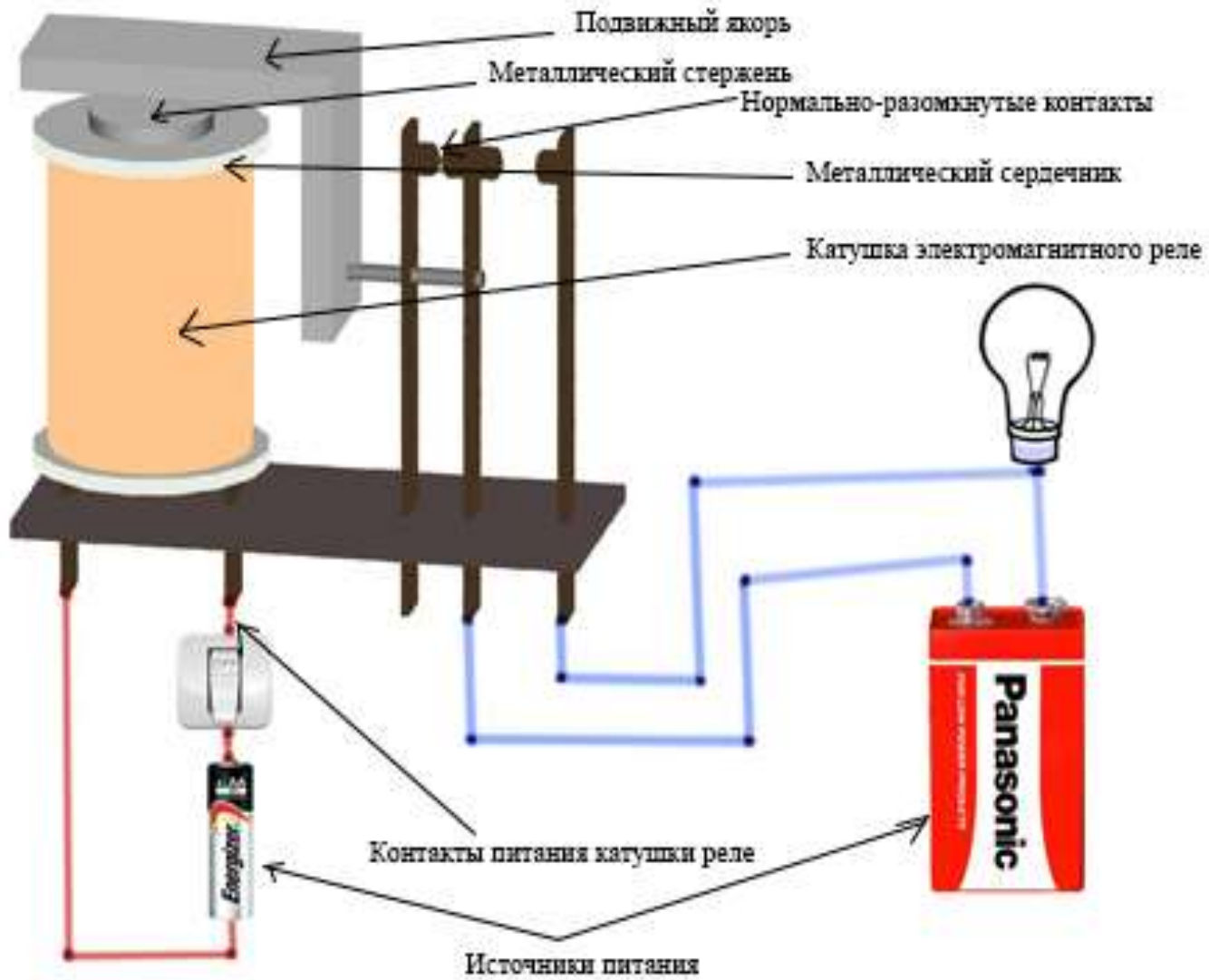
Входы и выходы ПЛК гальванически развязаны с центральным процессором, а, следовательно, и друг с другом

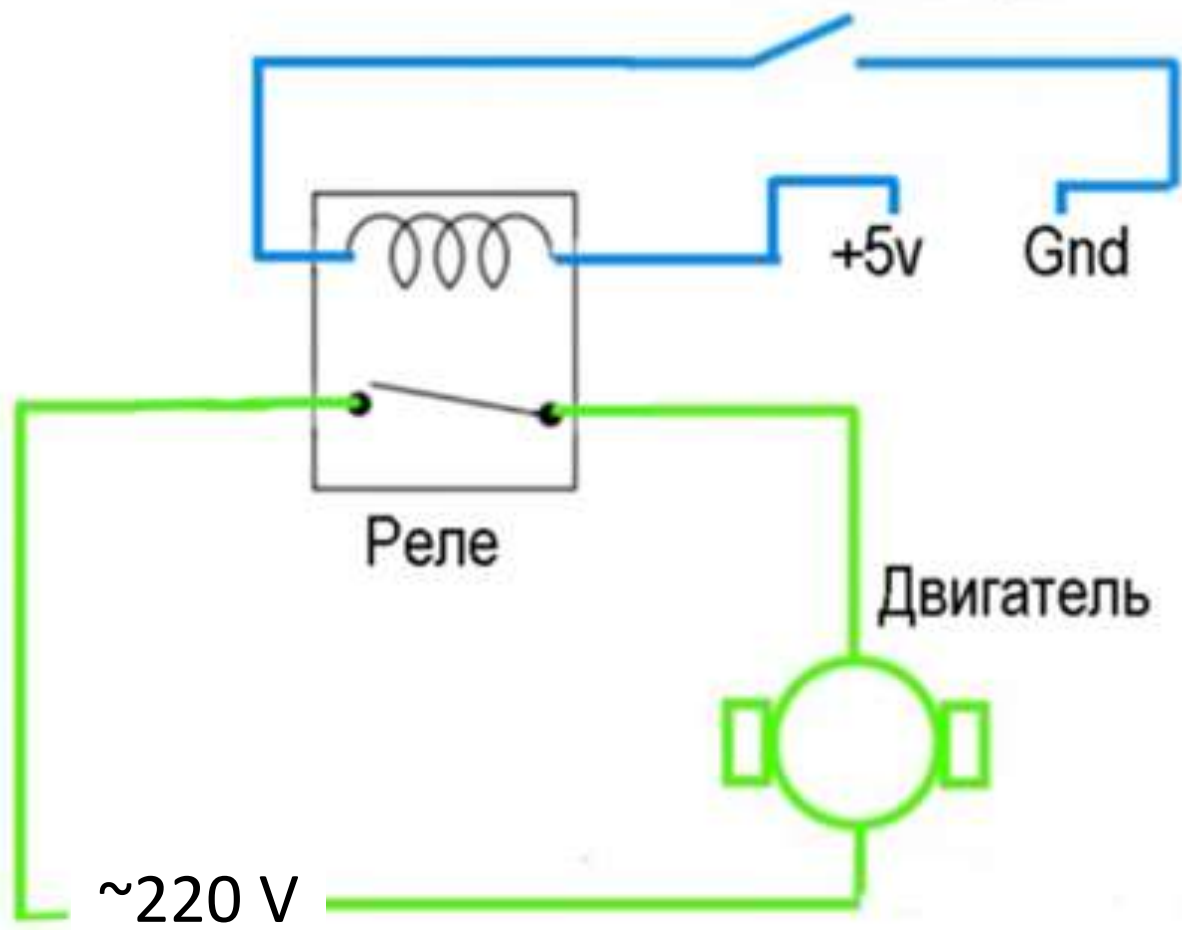




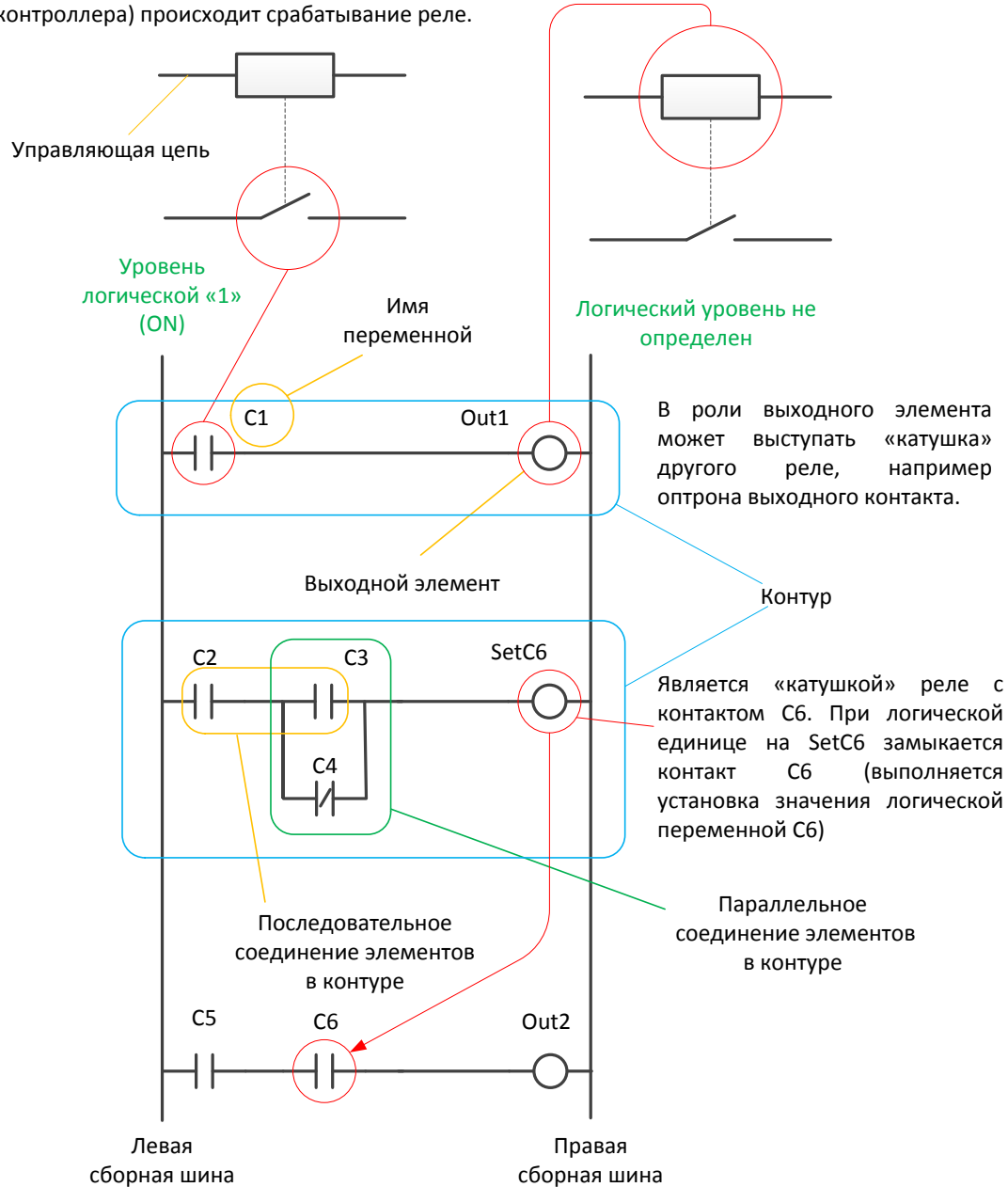
Языки программирования ПЛК IEC 61131-3

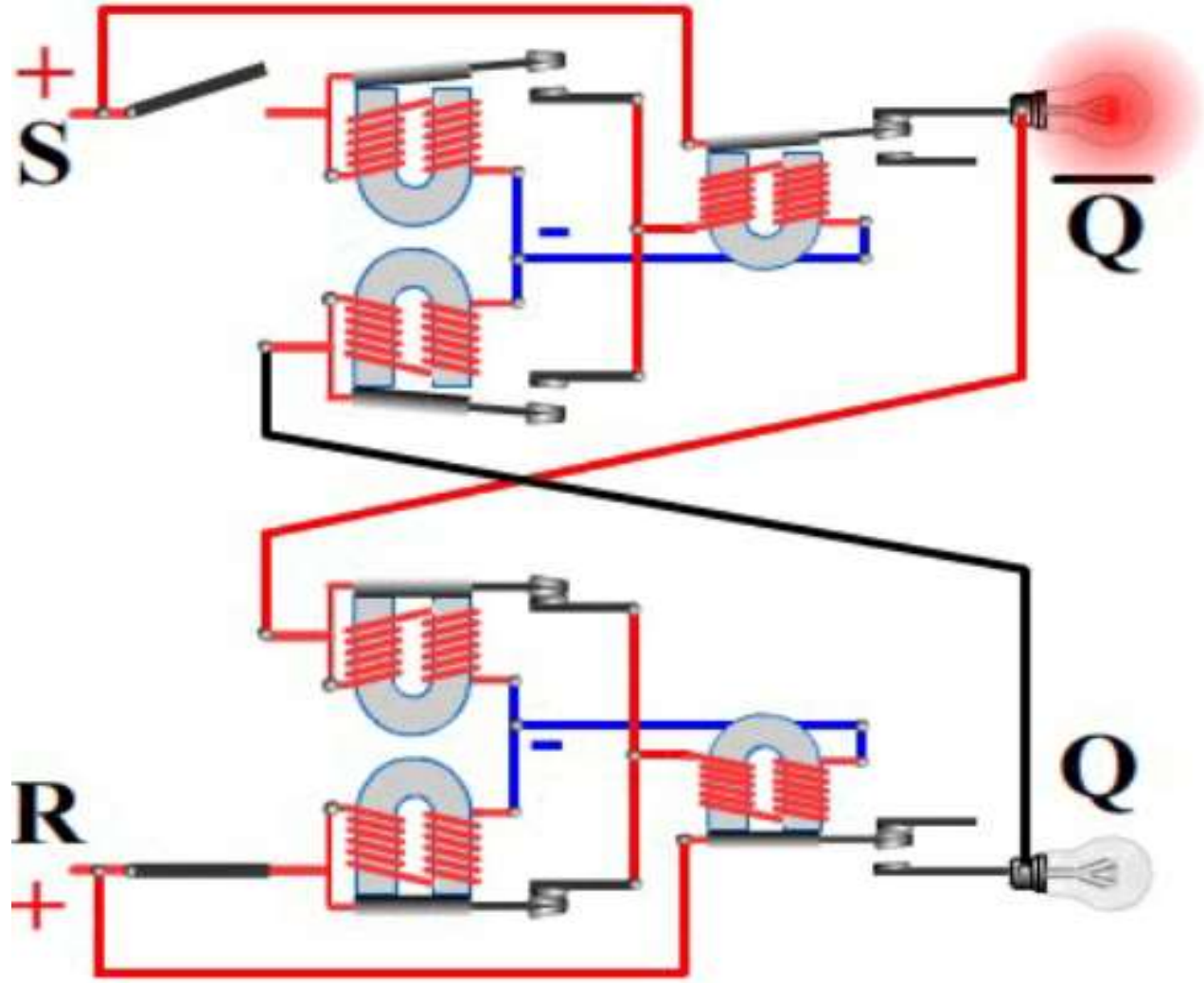
Название языка		Описание	
Графические языки			
LD	Ladder Diagram	Релейно-Контактные Схемы (РКС)	Представляет собой программную реализацию электрических схем на базе электромагнитных реле.
FBD	Function Block Diagram	Функциональные блоковые диаграммы	Программа создается путём соединения множества функциональных блоков. Функциональный блок (ФБ) выражает некую подпрограмму. Каждый ФБ имеет входы (слева) и выходы (справа).
SFC	Sequential Function Chart	Последовательные функциональные диаграммы	Описывает последовательность состояний и условий переходов. Создан на базе математического аппарата сетей Петри.
Текстовые языки			
ST	Structured Text	Структурированный текст	Паскалеподобный язык программирования
IL	Instruction List	Список инструкций	Аппаратно-независимый низкоуровневый ассемблероподобный язык.





В роли управляющей цепи может выступать цепь входного контакта контроллера, при активации которой (подачи напряжения на вход контроллера) происходит срабатывание реле.





LD	ЕСКД	Название	Описание
		Нормально разомкнутый контакт	Разомкнут при значении «ложь» (логическая единица, «выключен», OFF), назначенной ему переменной. Замыкается при значении «истина».
		Нормально замкнутый контакт	Замкнут, если переменная имеет значение <i>ложь</i> , и разомкнут, если переменная имеет значение <i>истина</i> .
		Итог логической цепочки (контура), «катушка», обычная неинверсная «катушка»	В случае активации контура (ток течет от левой сборной шины к правой), реализует установку значения целевой переменной (активацию выходного контакта ПЛК, включение / выключение контакта (контактов) другого реле, активацию входа функционального блока (счетчика, таймера и т.д.).
		Инверсный итог логической цепочки (контура), инверсная «катушка»	В случае отсутствия активации контура реализует установку значения целевой переменной.

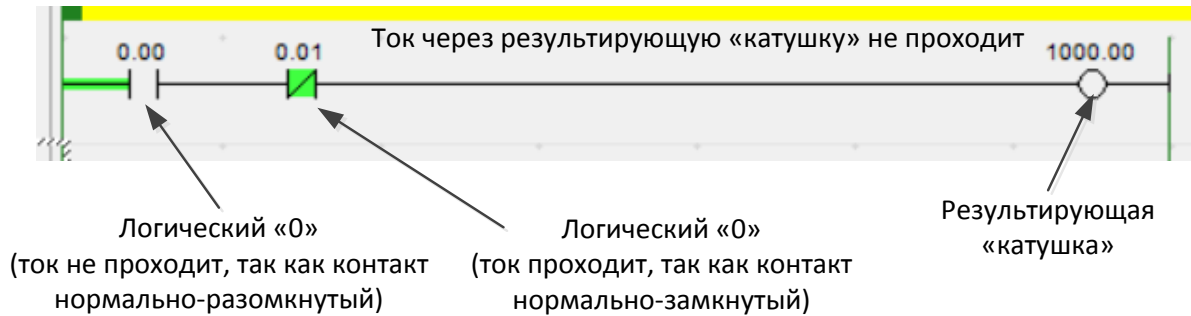
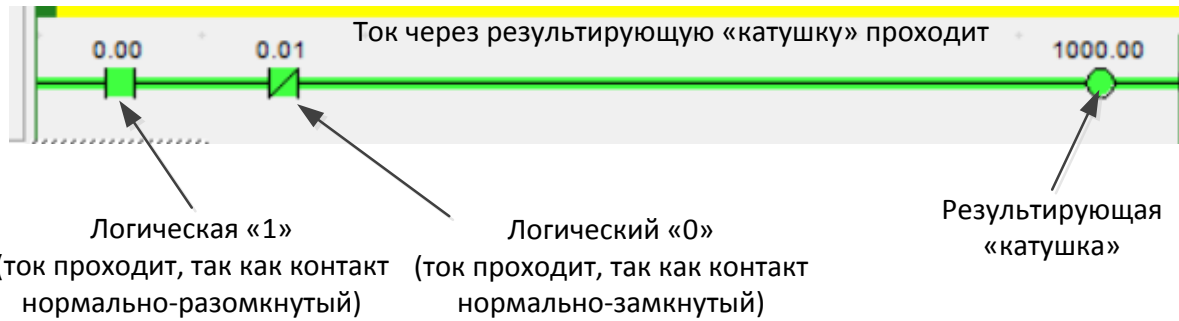
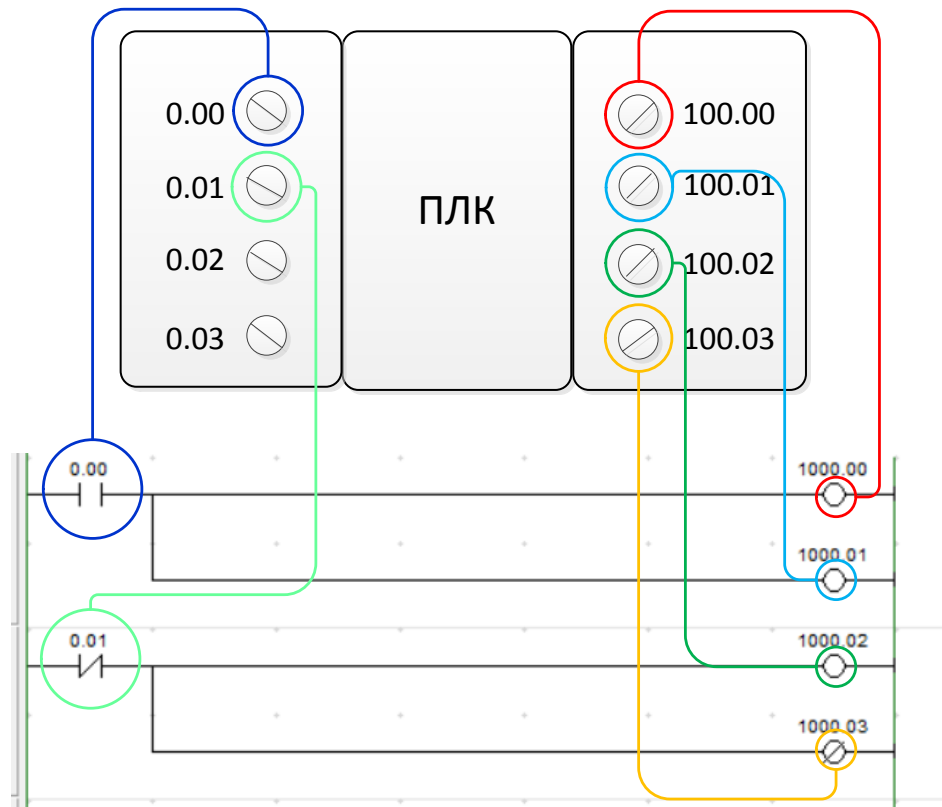
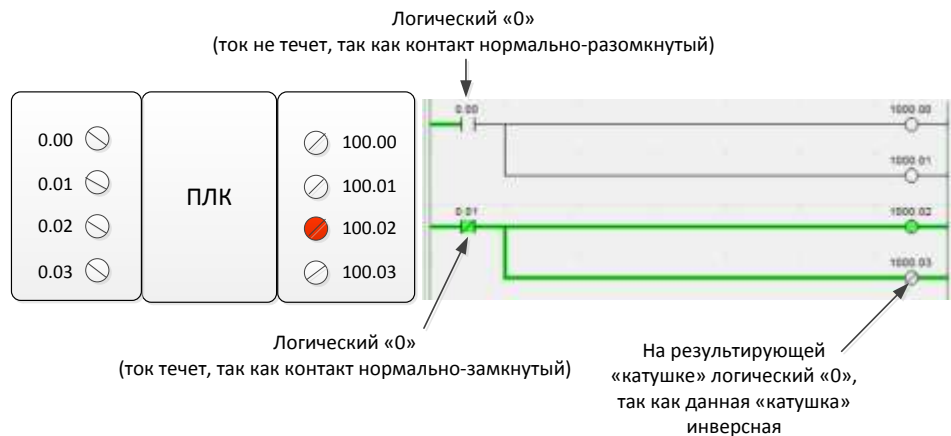


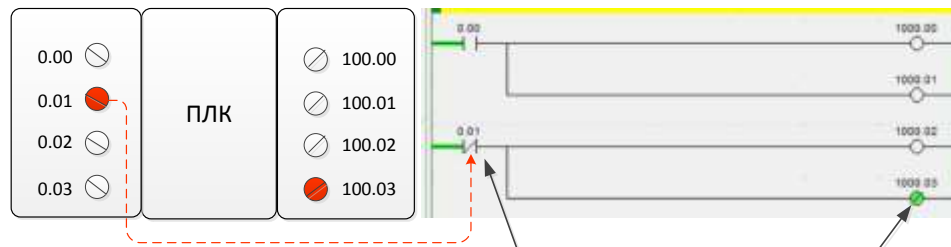
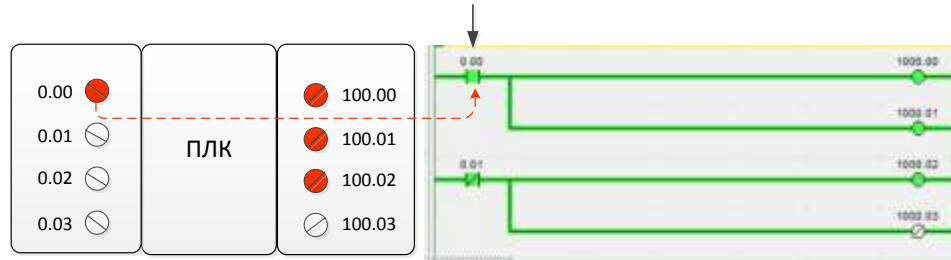
Таблица истинности, описывающая состояния входов и выходов ПЛК

Входы		Выходы			
0.00	0.01	1000.00	1000.01	1000.02	1000.03
0	0	0	0	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	1	1	1	0	1



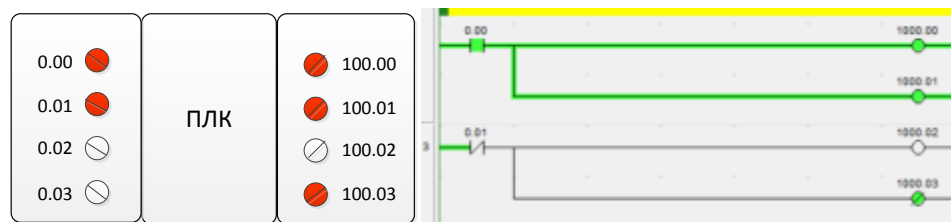


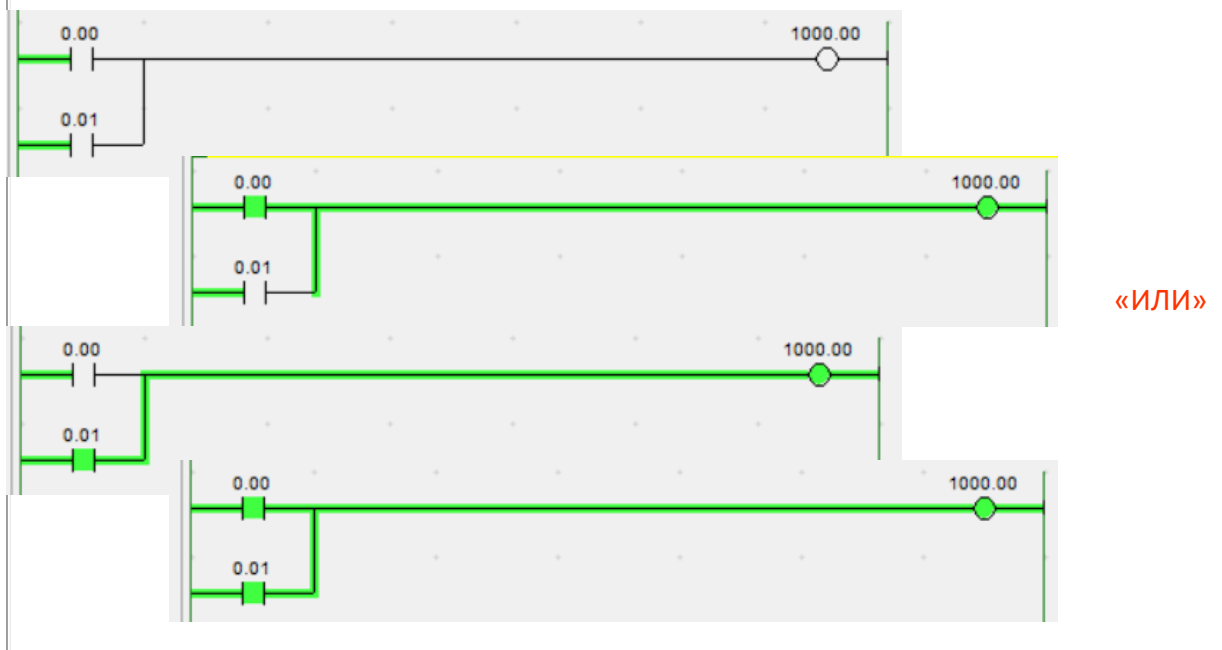
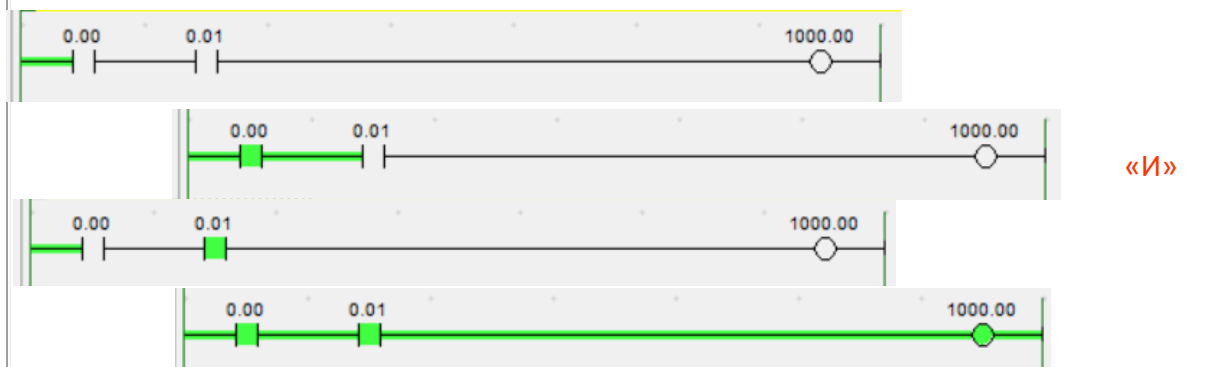
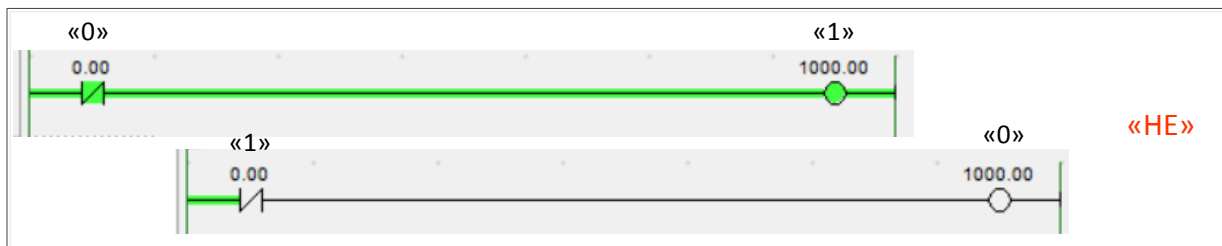
При подачи напряжения на вход 0.00, устанавливается логическая «1» (ток течет, так как контакт нормально-разомкнутый)



При подачи напряжения на вход 0.01, устанавливается логическая «1» (ток не течет, так как контакт нормально-замкнутый)

На результирующей «катушке» логическая «1», так как данная «катушка» инверсная





Организация памяти ПЛК CP1L

Область памяти		Диапазон	Размер
Область CIO	Входы	CIO 0 - CIO 99	100 слов (1600 бит)
	Выходы	CIO 100 - CIO 199	100 слов (1600 бит)
	Область связей 1:1 (1:1 Link)	CIO 3000 - CIO 3063	64 слова (1024 бит)
	Область последовательных связей ПЛК (Serial PLC link)	CIO 3100 - CIO 3189	90 слов (1440 бит)
	Рабочая область	CIO 3800 - CIO 6143	900 слов (14400 бит)
Рабочая область W		W000 - W511	512 слов
Область хранения H		H000 - H511	512 слов
Вспомогательная область A		A000 - A959	960 слов
Область TR		TR0 - TR15	16 бит
Память данных DM		D00000 - D32767	32768 слов
Флаги выполнения таймеров		T0000 - T4095	4096 битов
Флаги выполнения счетчиков		C0000 - C4095	4096 битов
Текущие значения таймеров		T0000 - T4095	4096 слов
Текущие значения счетчиков		C0000 - C4095	4096 слов
Область флагов задач ТК		TK0 - TK31	32 бита
Индексные регистры IR		IR0 - IR15	16 регистров
Регистры данных DR		DR0 - DR15	16 регистров

Схема адресации слов и битов



Номер бита (00 ... 15)

Адрес слова

Бит 03 слова 0001 в области СІО

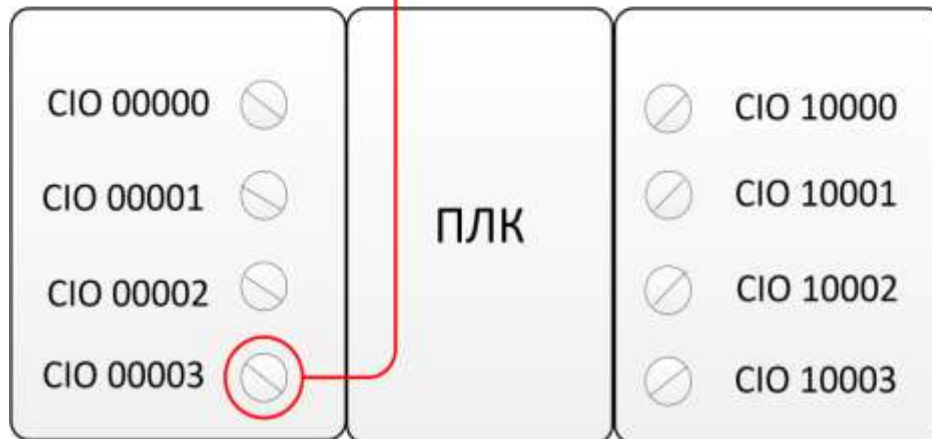
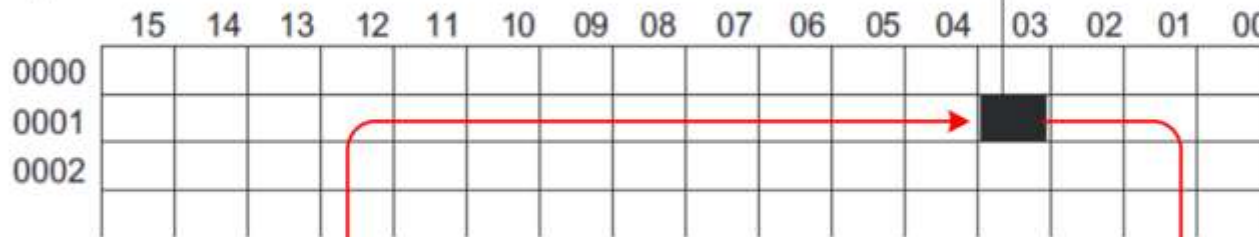
0001 03

Номер бита (03)

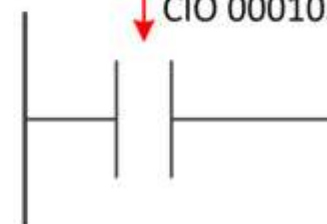
Адрес слова: 0001

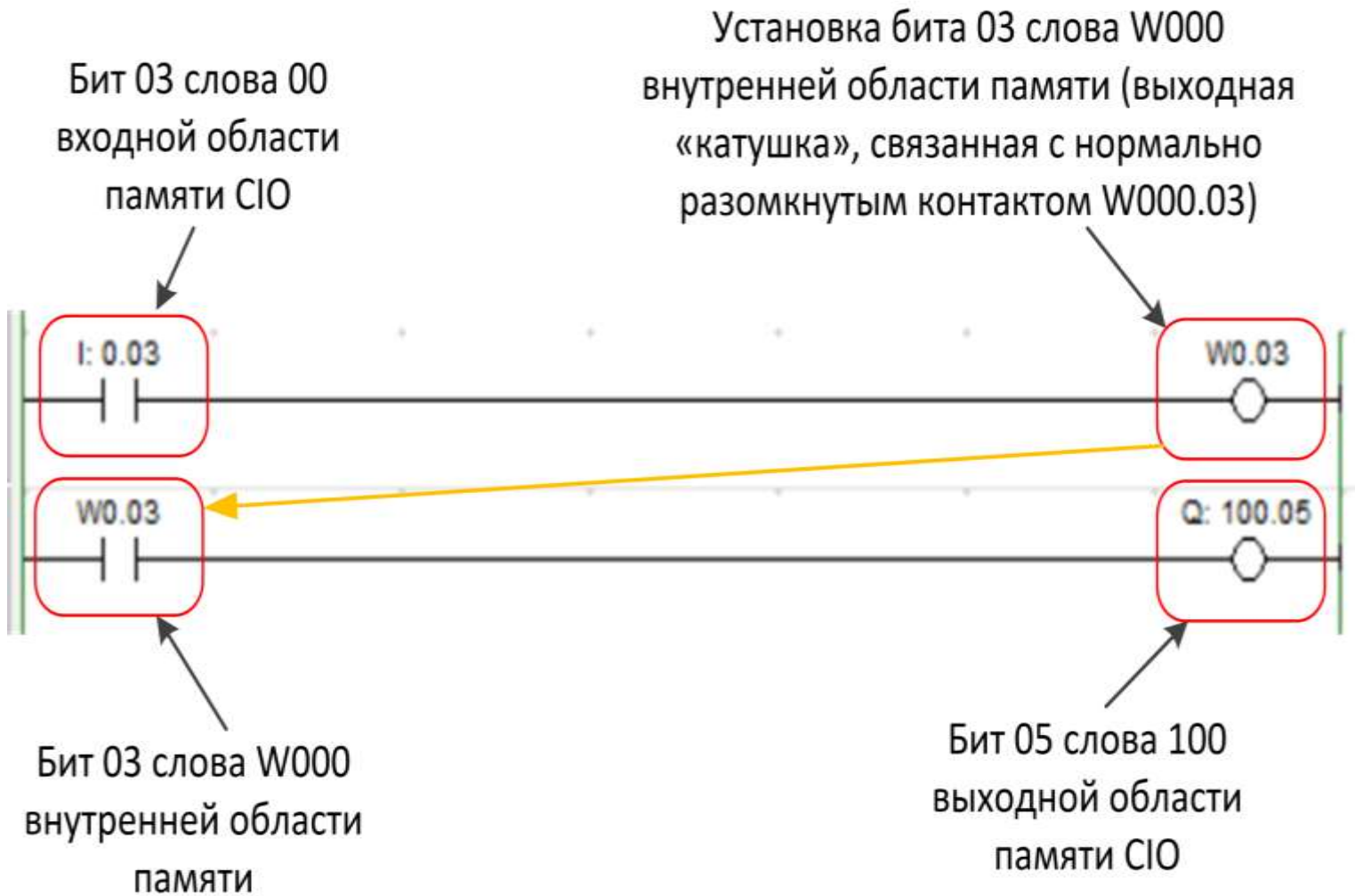
Бит: СІО 000103

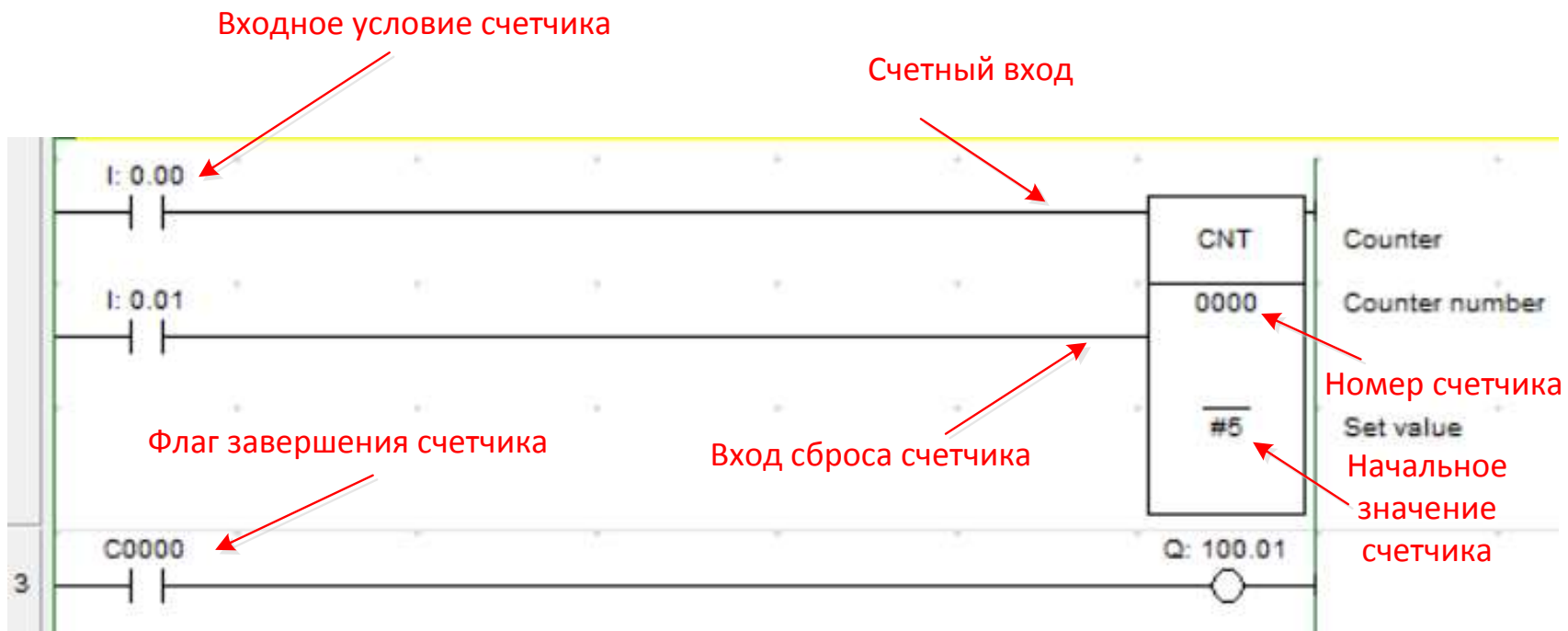
Слово



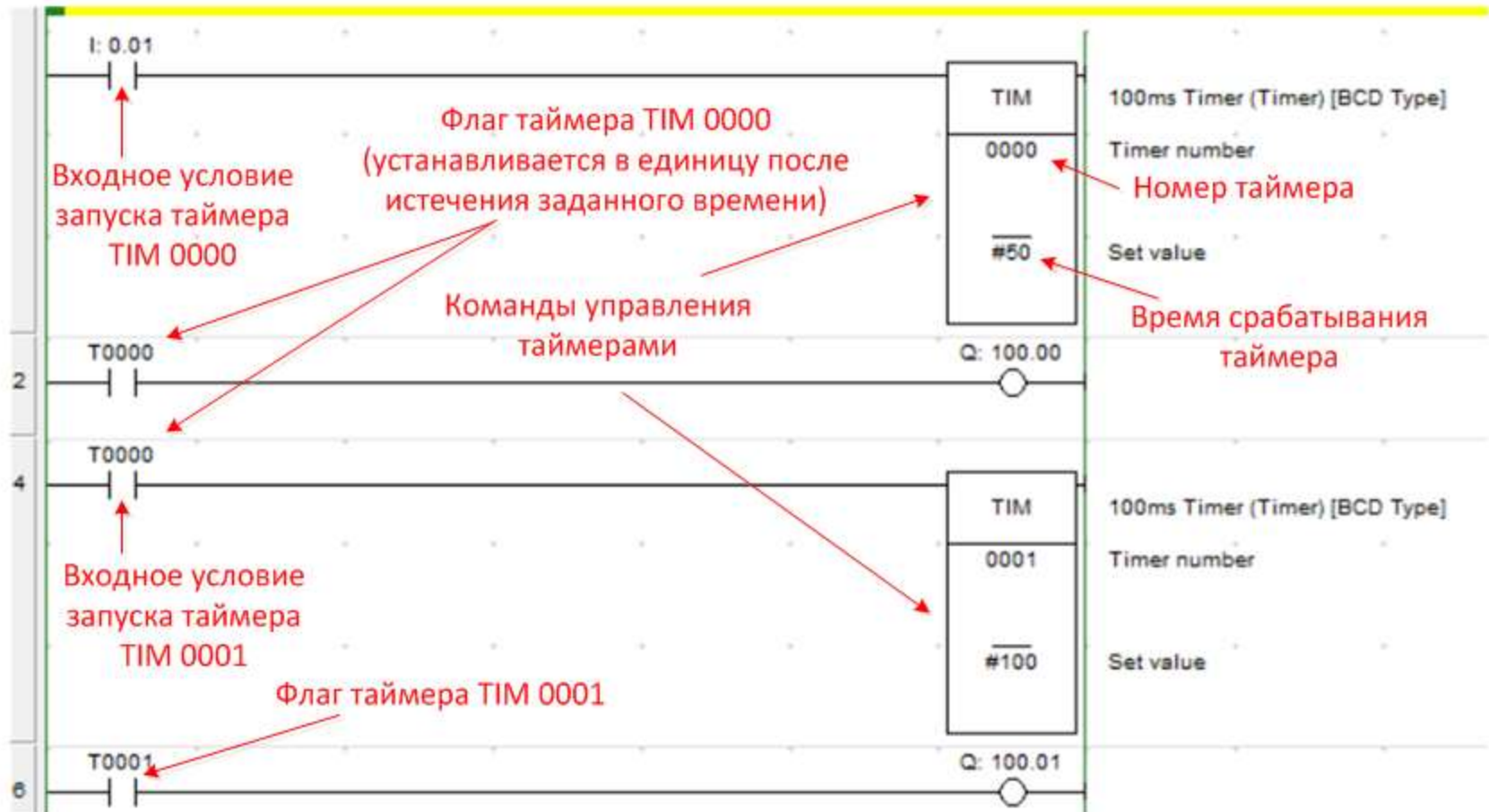
СІО 000103



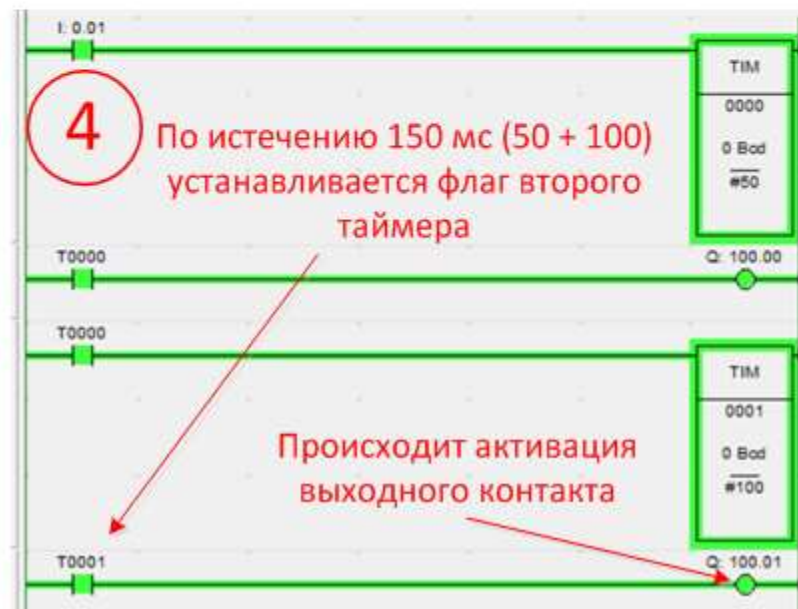
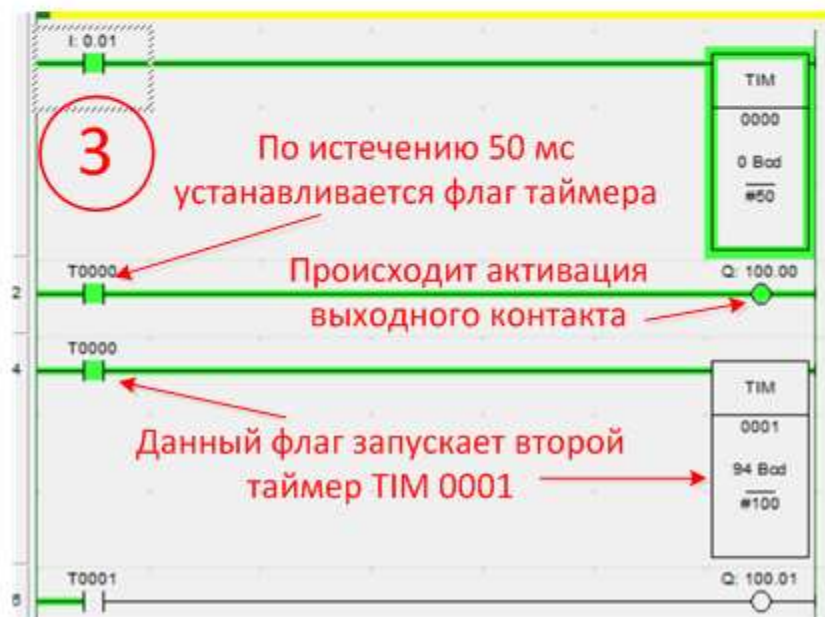
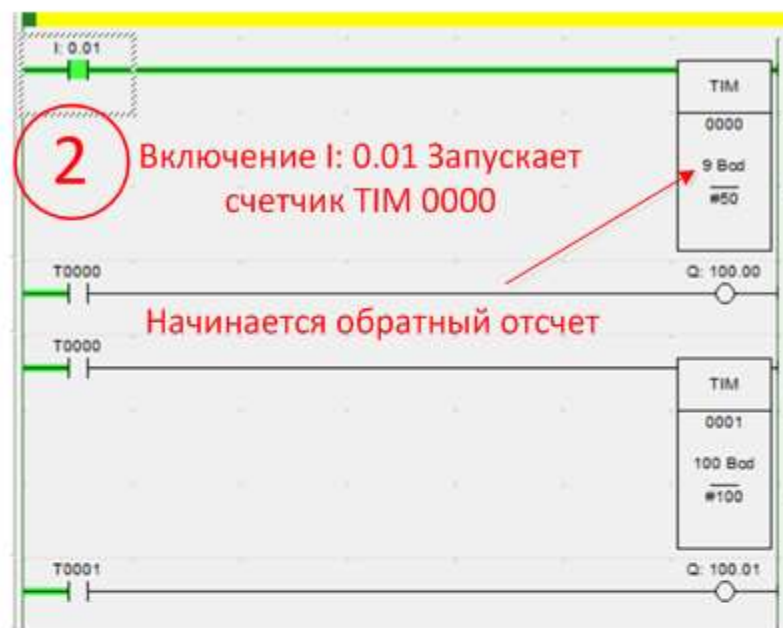
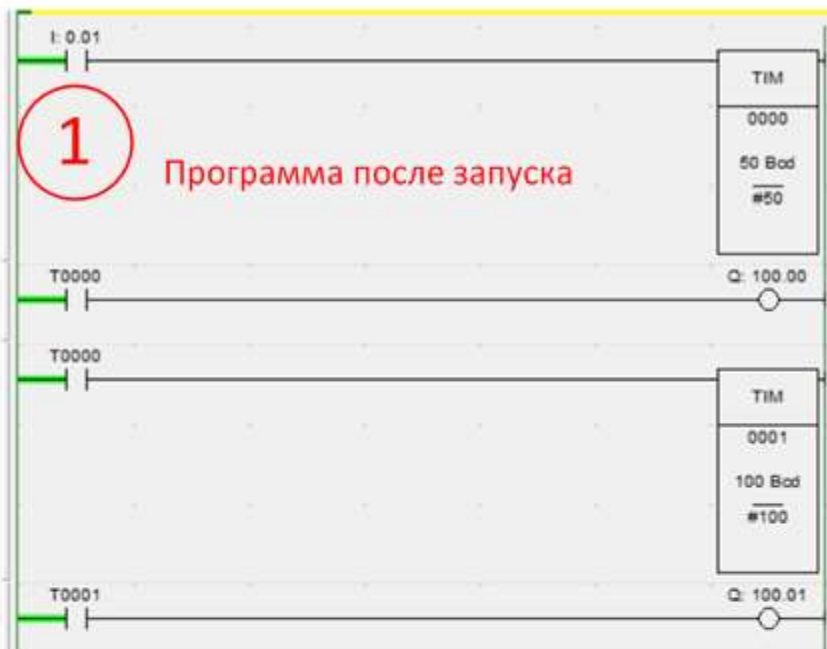


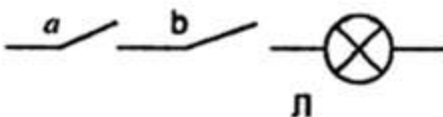
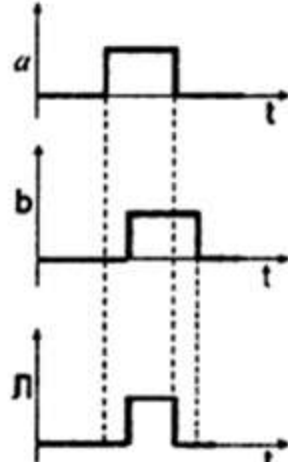
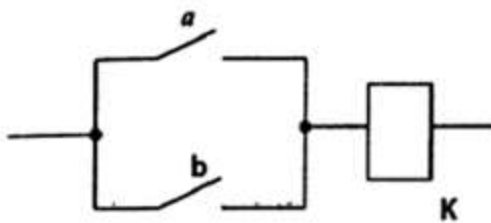
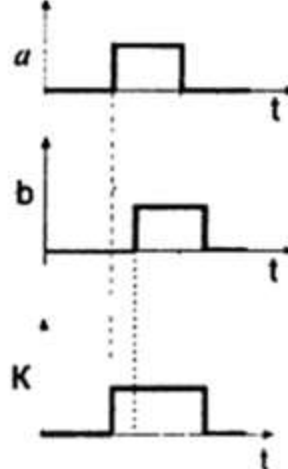


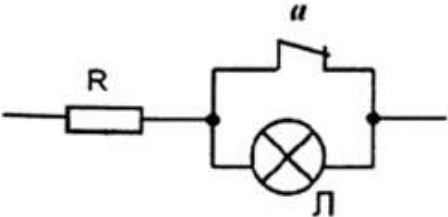
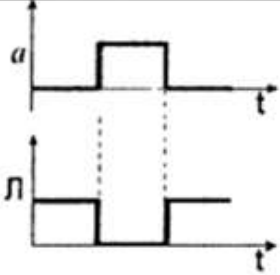
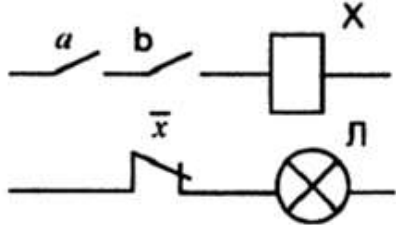
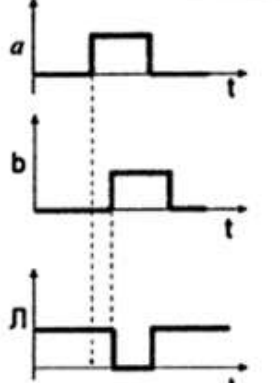
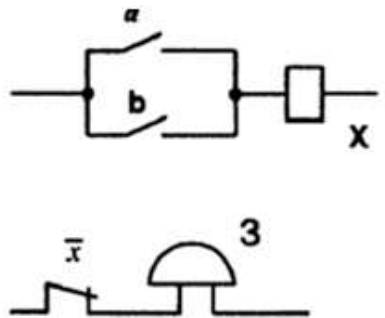
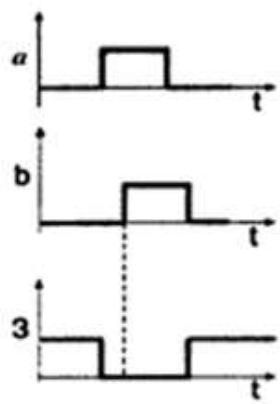
ПРОГРАММА НА LD, ИСПОЛЬЗУЮЩАЯ ТАЙМЕРЫ

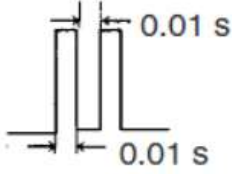
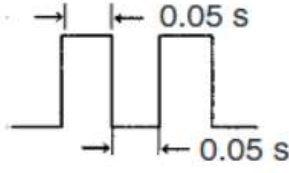
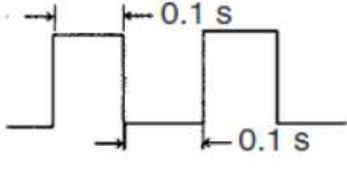
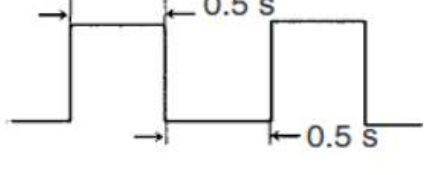
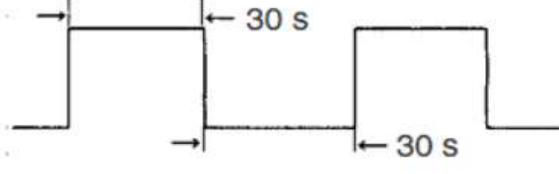


ВЫПОЛНЕНИЕ ПРОГРАММЫ НА LD, ИСПОЛЬЗУЮЩЕЙ ТАЙМЕРЫ



Операция	Релейно-контактная схема	Таблица состояний	Временные диаграммы															
Логическое умножение		<table border="1" data-bbox="1101 414 1284 678"> <thead> <tr> <th>a</th> <th>b</th> <th>Л</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	a	b	Л	0	0	0	0	1	0	1	0	0	1	1	1	
a	b	Л																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
Логическое сложение		<table border="1" data-bbox="1101 906 1284 1170"> <thead> <tr> <th>a</th> <th>b</th> <th>К</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	a	b	К	0	0	0	0	1	1	1	0	1	1	1	1	
a	b	К																
0	0	0																
0	1	1																
1	0	1																
1	1	1																

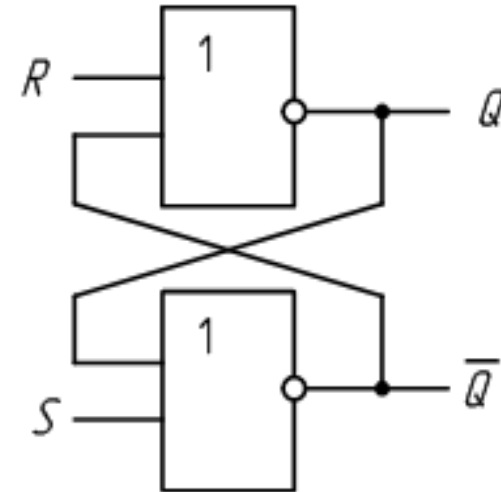
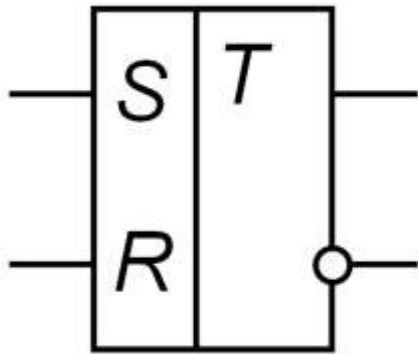
<p>Логическое отрицание</p>		<table border="1" data-bbox="1130 171 1246 321"> <thead> <tr> <th>a</th> <th>Л</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	a	Л	0	1	1	0										
a	Л																	
0	1																	
1	0																	
<p>Функция «И-НЕ»</p>		<table border="1" data-bbox="1101 478 1265 721"> <thead> <tr> <th>a</th> <th>b</th> <th>Л</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	a	b	Л	0	0	1	0	1	1	1	0	1	1	1	0	
a	b	Л																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
<p>Функция «ИЛИ-НЕ»</p>		<table border="1" data-bbox="1101 892 1265 1135"> <thead> <tr> <th>a</th> <th>b</th> <th>З</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	a	b	З	0	0	0	0	1	1	1	0	1	1	1	1	
a	b	З																
0	0	0																
0	1	1																
1	0	1																
1	1	1																

Название	Символ	Описание работы	
Тактовые импульсы 0,02 с	P_0_02_s		ВКЛ в теч. 0,01 с ВЫКЛ в теч. 0,01 с
Тактовые импульсы 0,1 с	P_0_1s		ВКЛ в теч. 0,05 с ВЫКЛ в теч. 0,05 с
Тактовые импульсы 0,2 с	P_0_2s		ВКЛ в теч. 0,1 с ВЫКЛ в теч. 0,1 с
Тактовые импульсы 1 с	P_1s		ВКЛ в теч. 0,5 с ВЫКЛ в теч. 0,5 с
Тактовые импульсы 1 мин	P_1min		ВКЛ в теч. 30 с ВЫКЛ в теч. 30 с

Триггер – это электронное логическое устройство с положительной обратной связью, имеющее два устойчивых состояния – единичное и нулевое, которые обозначаются соответственно 1 и 0. Такое устройство может сохранять своё состояние теоретически бесконечно долго (при наличии питания). Триггер относится к цифровым устройствам с памятью или к автоматам. Переключение триггера происходит по входному сигналу извне.

Триггеры предназначены для запоминания двоичной информации. Использование триггеров позволяет реализовать устройства оперативной памяти (т.е. памяти, информация в которой хранится только на время вычислений). Кроме того, триггеры могут использоваться для построения некоторых цифровых устройств с памятью, таких как счётчики, преобразователи последовательного кода в параллельный, цифровые линии задержки.

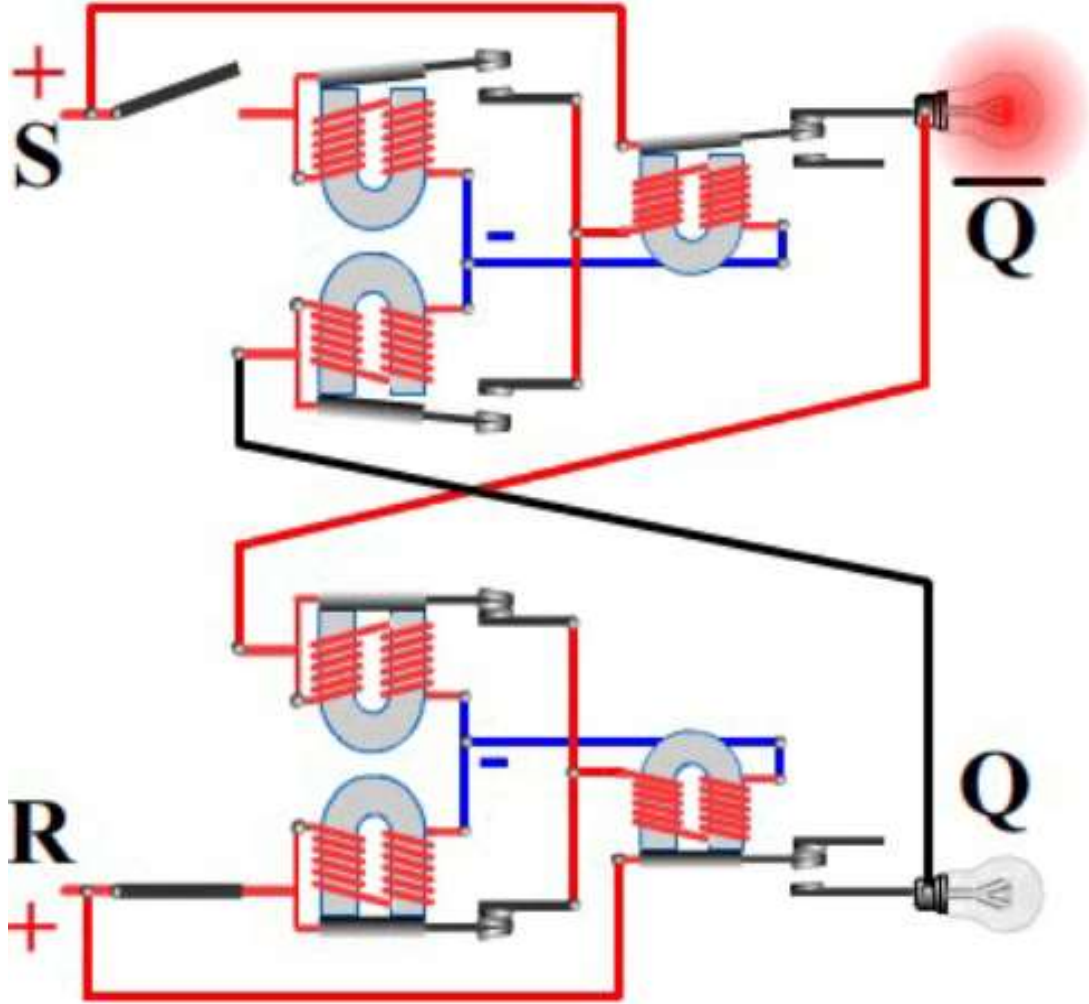
RS-триггер – асинхронный триггер, который сохраняет своё предыдущее состояние при нулевых значениях на входах, и меняет своё выходное состояние при подаче на один из его входов единицы. RS-триггер является основным триггером, на основе которого строятся все остальные триггеры.



Входы и выходы RS-триггера	
Входы	Выходы
R – установка 0 (от слова reset); S – установка 1 (от слова set);	Q – прямой; Q – обратный(инверсный).

Таблица переходов RS-триггера в базисе ИЛИ-НЕ

R	S	Q(t)	Q(t+1)	Комментарии
0	0	0	0	Режим хранения информации R=S=0
0	0	1	1	
0	1	0	1	Режим установки единицы S=1
0	1	1	1	
1	0	0	0	Режим установки нуля R=1
1	0	1	0	
1	1	0	-	R=S=1 запрещённая комбинация
1	1	1	-	



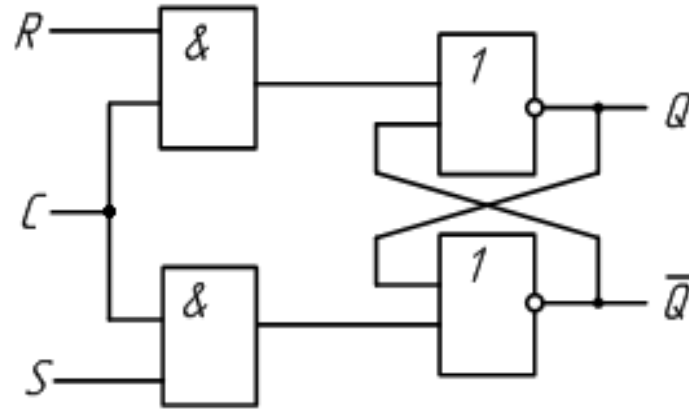
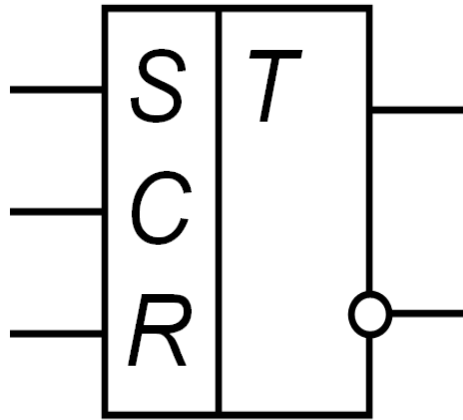
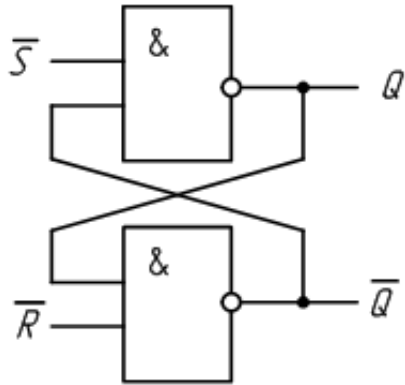


Таблица переходов синхронного RS-триггера

R	S	C	Q(t)	Q(t+1)	Пояснение
0	0	1	0	0	Режим хранения информации R=S=0
0	0	1	1	1	
0	1	1	0	1	Режим установки единицы S=1
0	1	1	1	1	
1	0	1	0	0	Режим установки нуля R=1
1	0	1	1	0	
1	1	1	0	-	R=S=1 запрещённая комбинация
1	1	1	1	-	

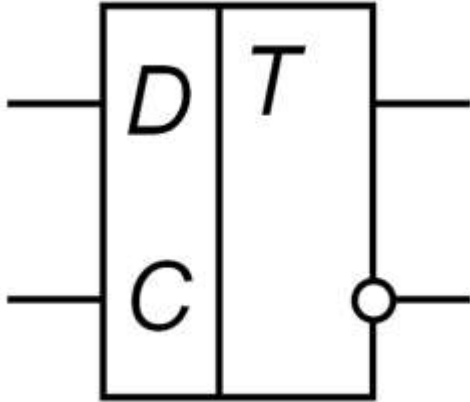


RS-триггер с инверсными входами

Таблица переходов RS-триггера в базисе 2И-НЕ

R	S	Q(t)	Q(t+1)	Комментарии
0	0	0	-	R=S=0 запрещённая комбинация
0	0	1	-	
0	1	0	0	Режим установки нуля R=0
0	1	1	0	
1	0	0	1	Режим установки единицы S=0
1	0	1	1	
1	1	0	0	Режим хранения информации R=S=1
1	1	1	1	

D-триггер

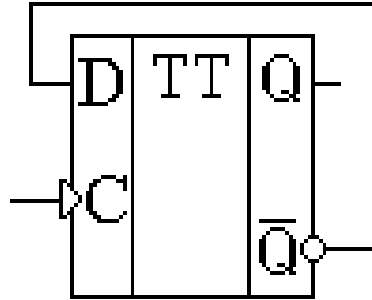


D-триггер (от англ. delay – задержка, либо от data – данные) запоминает состояние входа и выдаёт его на выход. В основном используется для реализации защёлки. D-триггер имеет 1 информационный вход (D-вход). Бывают только синхронные D-триггер. Состояние информационного входа передаётся на выход под действием синхроимпульса (вход C).

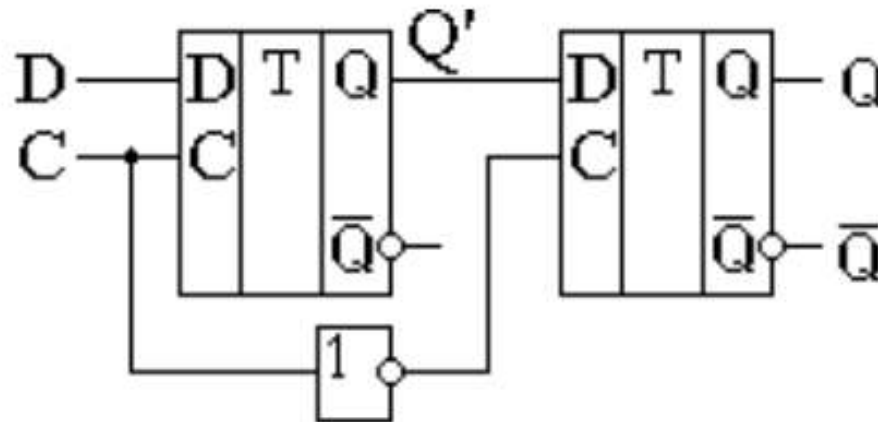
Таблица переходов D – триггера

C	D	Q(t)	Q(t+1)	Режимы
0	-	0	0	Режим хранения информации
0	-	1	1	
1	0	-	0	Режим записи информации
1	1	-	1	

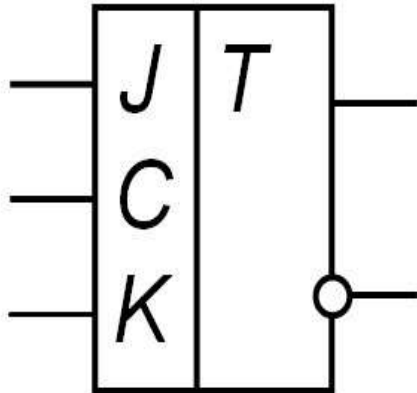
D-триггер, работающий по фронту



Триггер, запоминающий входную информацию по фронту, может быть построен из двух триггеров, работающих по потенциалу. Сигнал синхронизации будем подавать на эти триггеры в противофазе.



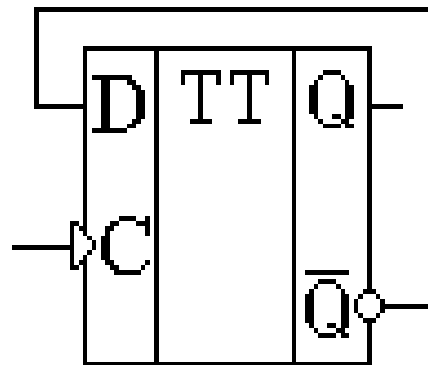
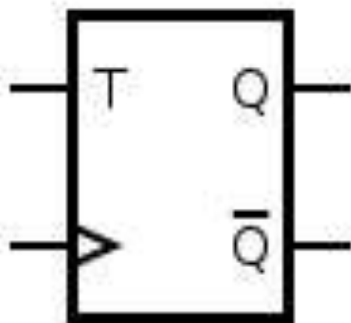
JK-триггер



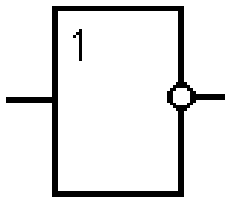
K	J	C	Q(t)	Q(t+1)
0	0	1	0	0
0	0	1	1	1
0	1	1	0	1
0	1	1	1	1
1	0	1	0	0
1	0	1	1	0
1	1	1	0	1
1	1	1	1	0

T-триггер

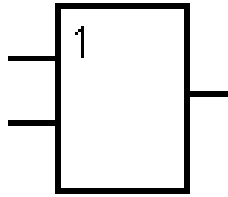
T-триггер (от англ. Toggle – переключатель) часто называют счётным триггером, так как он является простейшим счётчиком до 2. Асинхронный T-триггер не имеет входа разрешения счёта – T и переключается по каждому тактовому импульсу на входе C. Синхронный триггер часто применяется для понижения частоты в 2 раза.



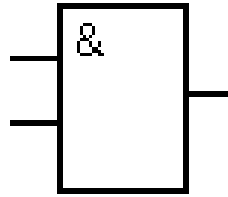
НЕ



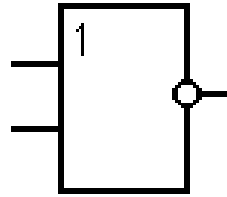
ИЛИ



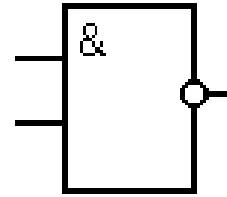
И



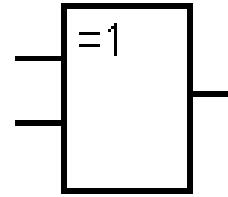
ИЛИ-НЕ



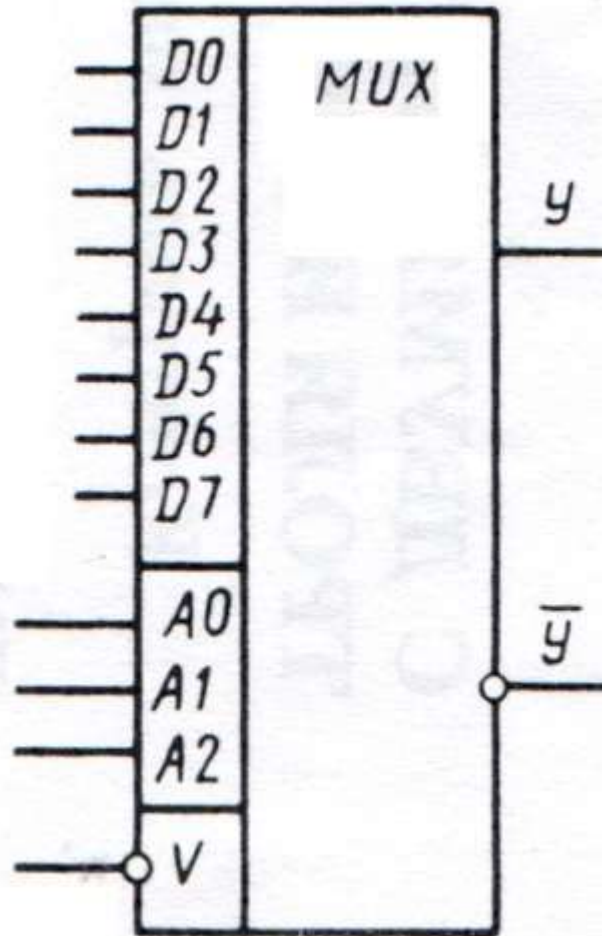
И-НЕ



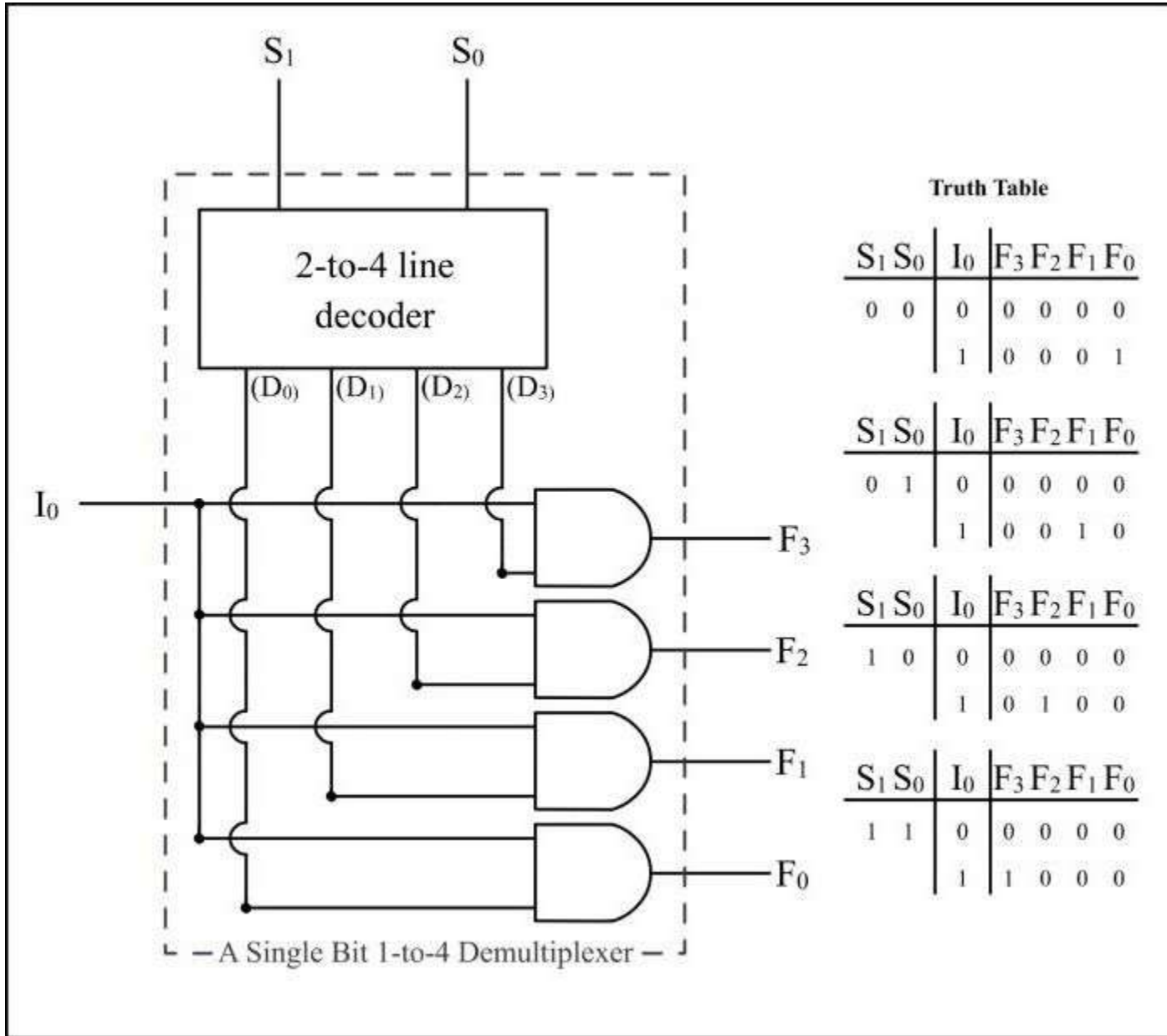
ИСКЛ.ИЛИ



Мультиплексор



Мультиплексор (схема)



Truth Table

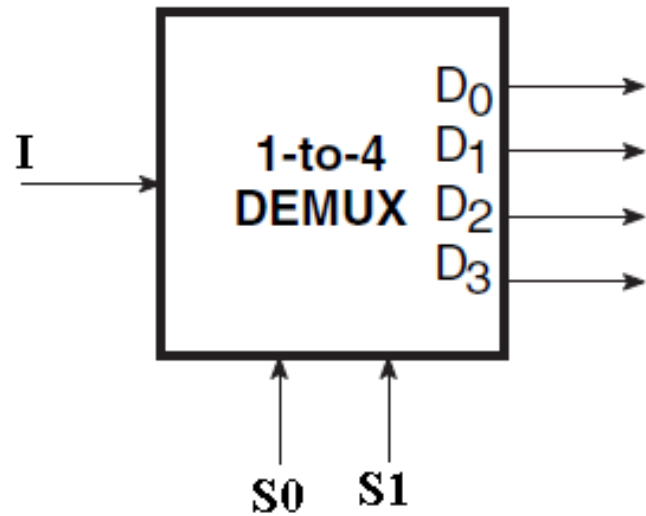
S_1	S_0	I_0	F_3	F_2	F_1	F_0
0	0	0	0	0	0	0
		1	0	0	0	1

S_1	S_0	I_0	F_3	F_2	F_1	F_0
0	1	0	0	0	0	0
		1	0	0	1	0

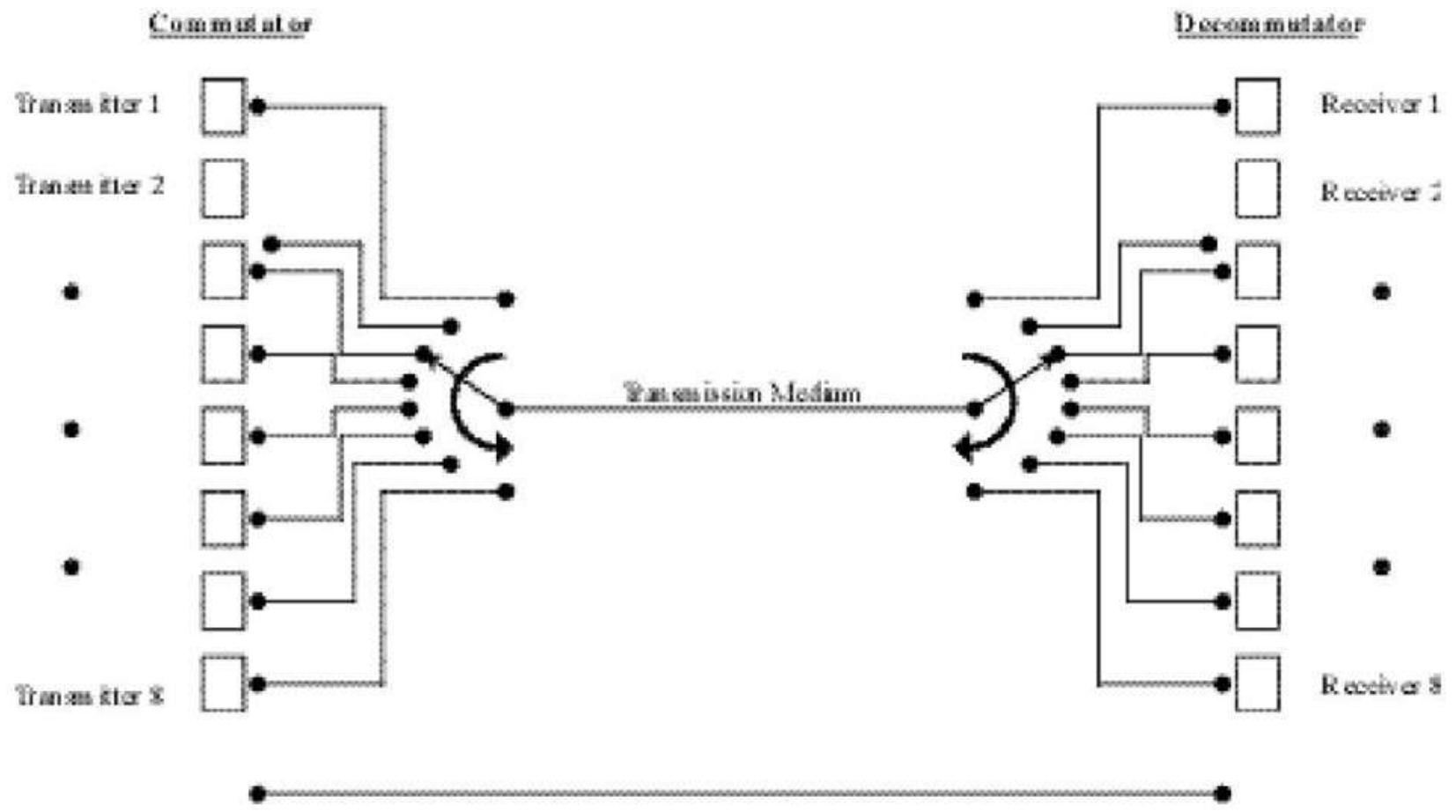
S_1	S_0	I_0	F_3	F_2	F_1	F_0
1	0	0	0	0	0	0
		1	0	1	0	0

S_1	S_0	I_0	F_3	F_2	F_1	F_0
1	1	0	0	0	0	0
		1	1	0	0	0

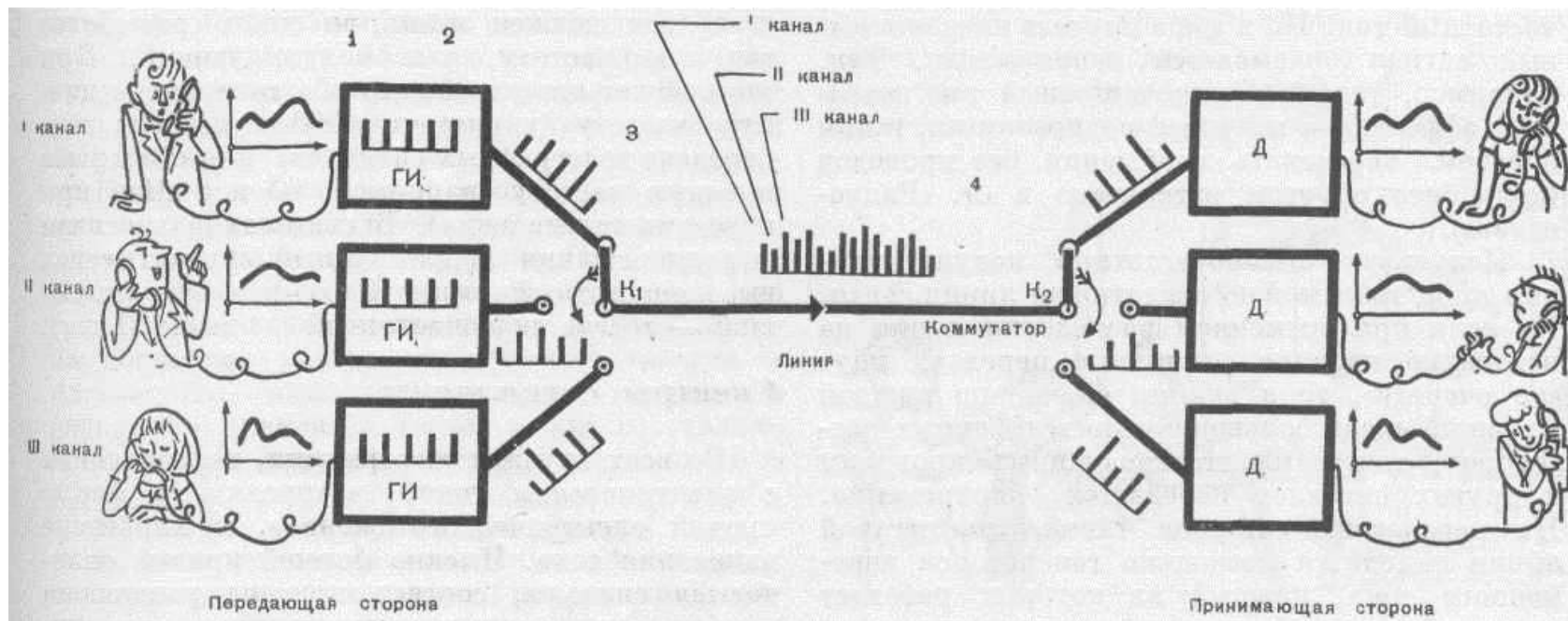
Демультимплексор



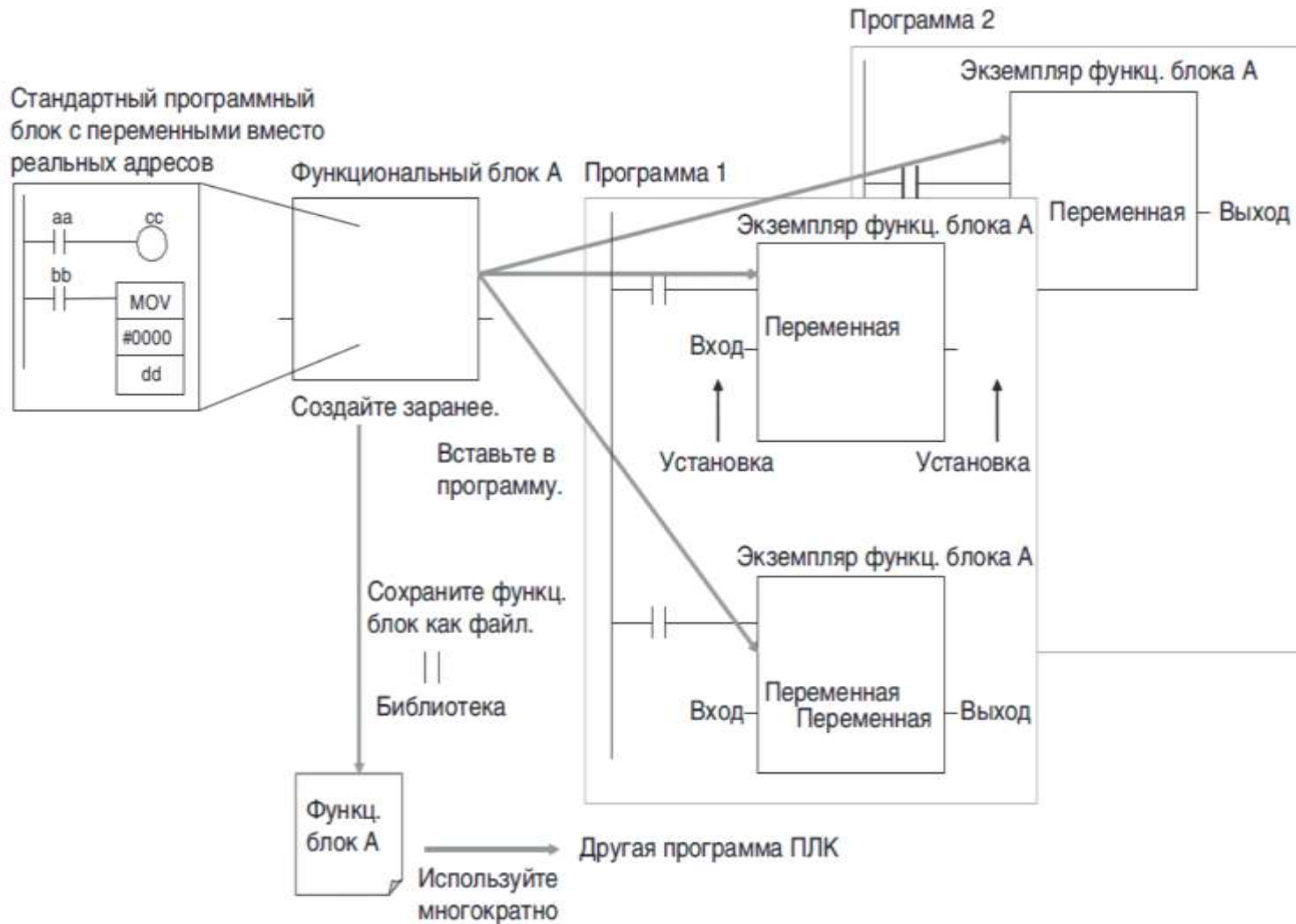
I	Select		O/P			
	S0	S1	D0	D1	D2	D3
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

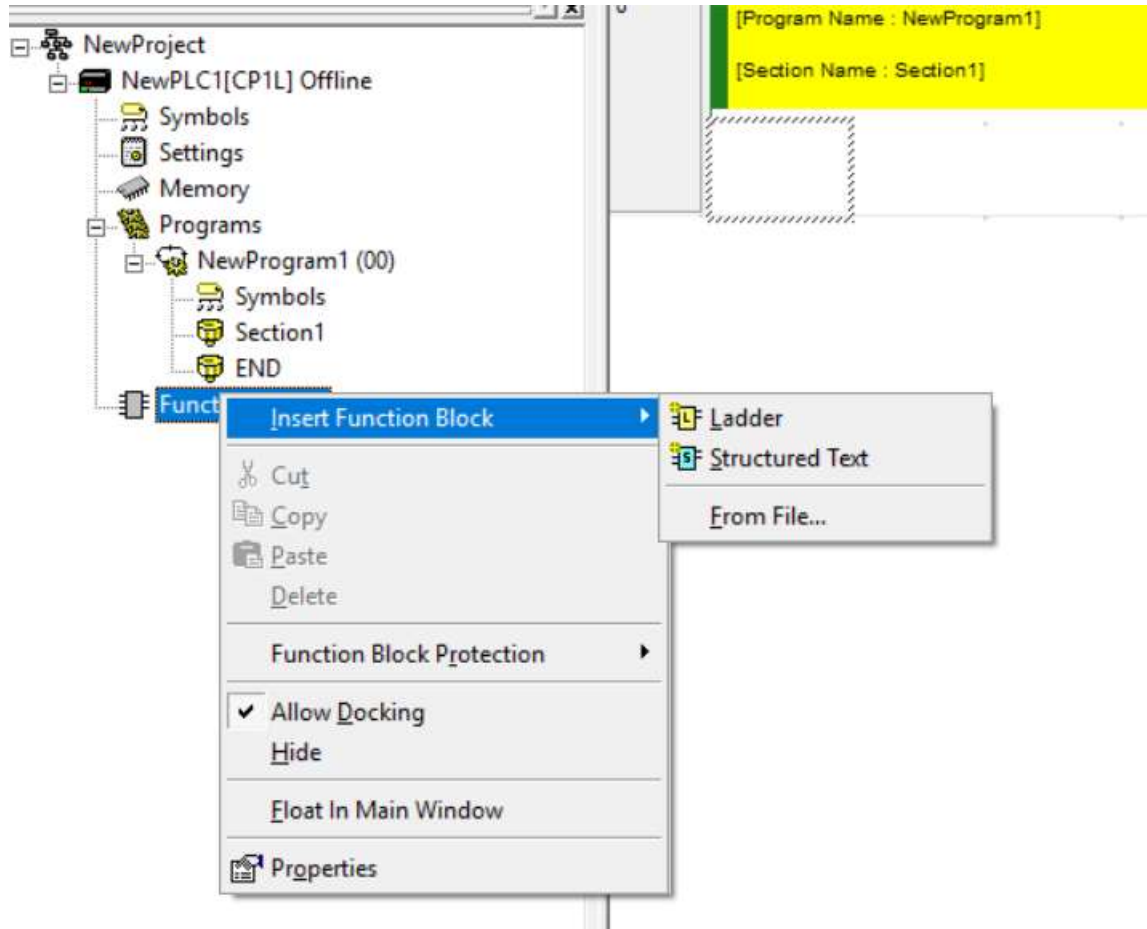


Временное разделение каналов



Концепция функционального блока





The project tree on the left shows the following structure:

- NewProject
 - NewPLC1[CP1L] Offline
 - Symbols
 - Settings
 - Memory
 - Programs
 - NewProgram1 (00)
 - Symbols
 - Section1
 - END
 - Function Blocks
 - FunctionBlock1

[Program Name : NewProgram1]
[Section Name : Section1]

Function Block Properties

General | Protection | Comments | Memory

Name:

Display the inside of FB

Author:

Revision:

Function Block Properties



General | Protection | Comments | Memory

Name:

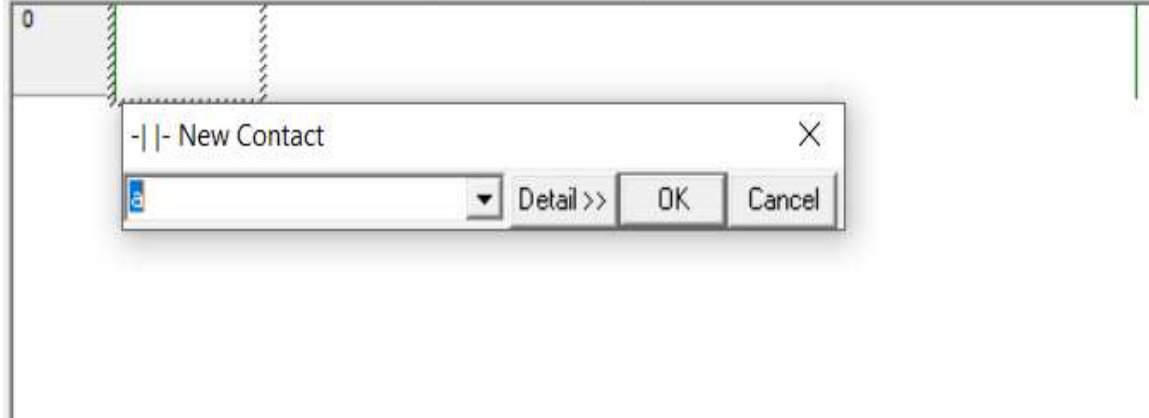
Display the inside of FB

Author:

Revision:

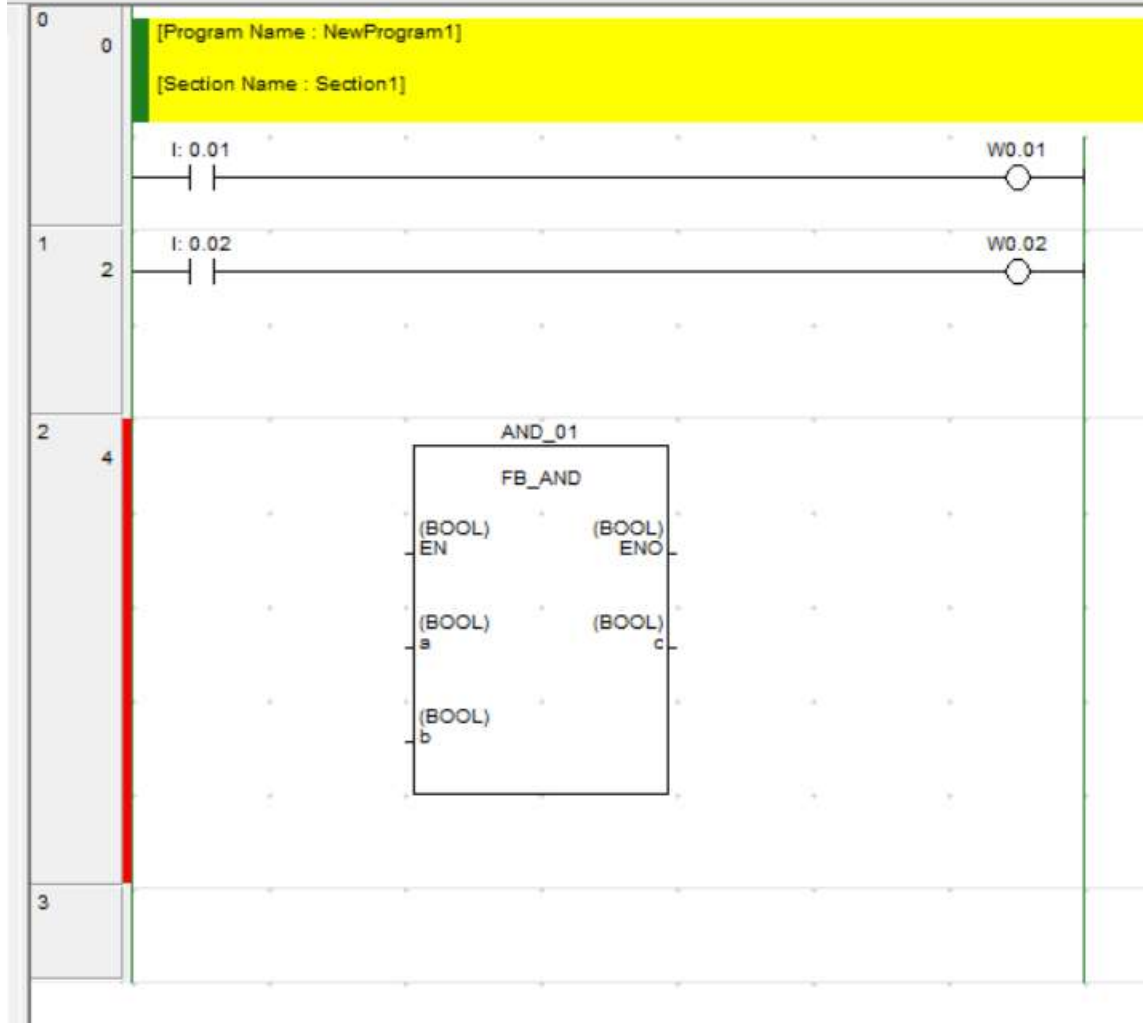
Name	Data Type	AT	Initial Va
ENO	BOOL		FALSE
c	BOOL		FALSE

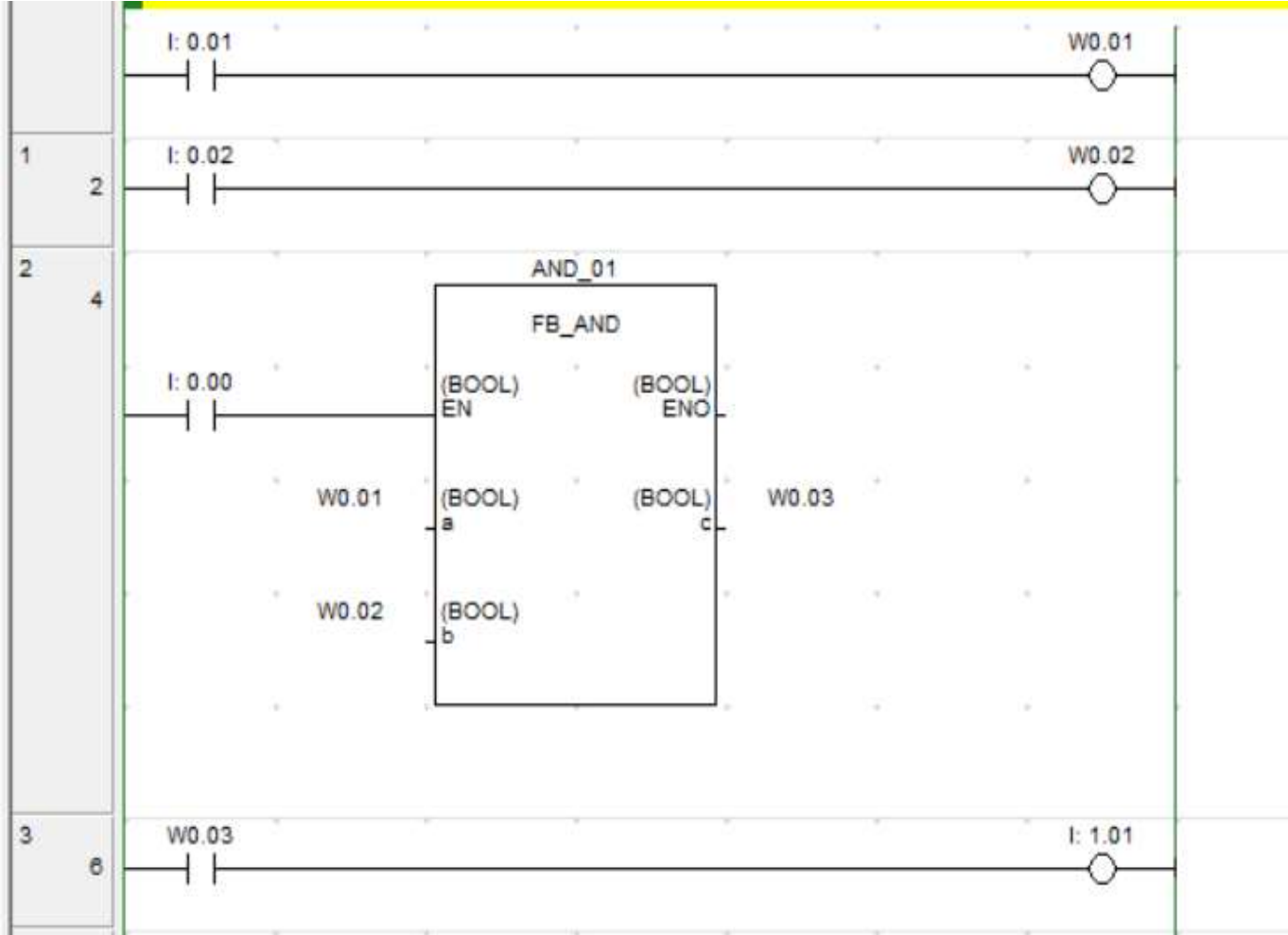
Internals Inputs **Outputs** In Out Externals

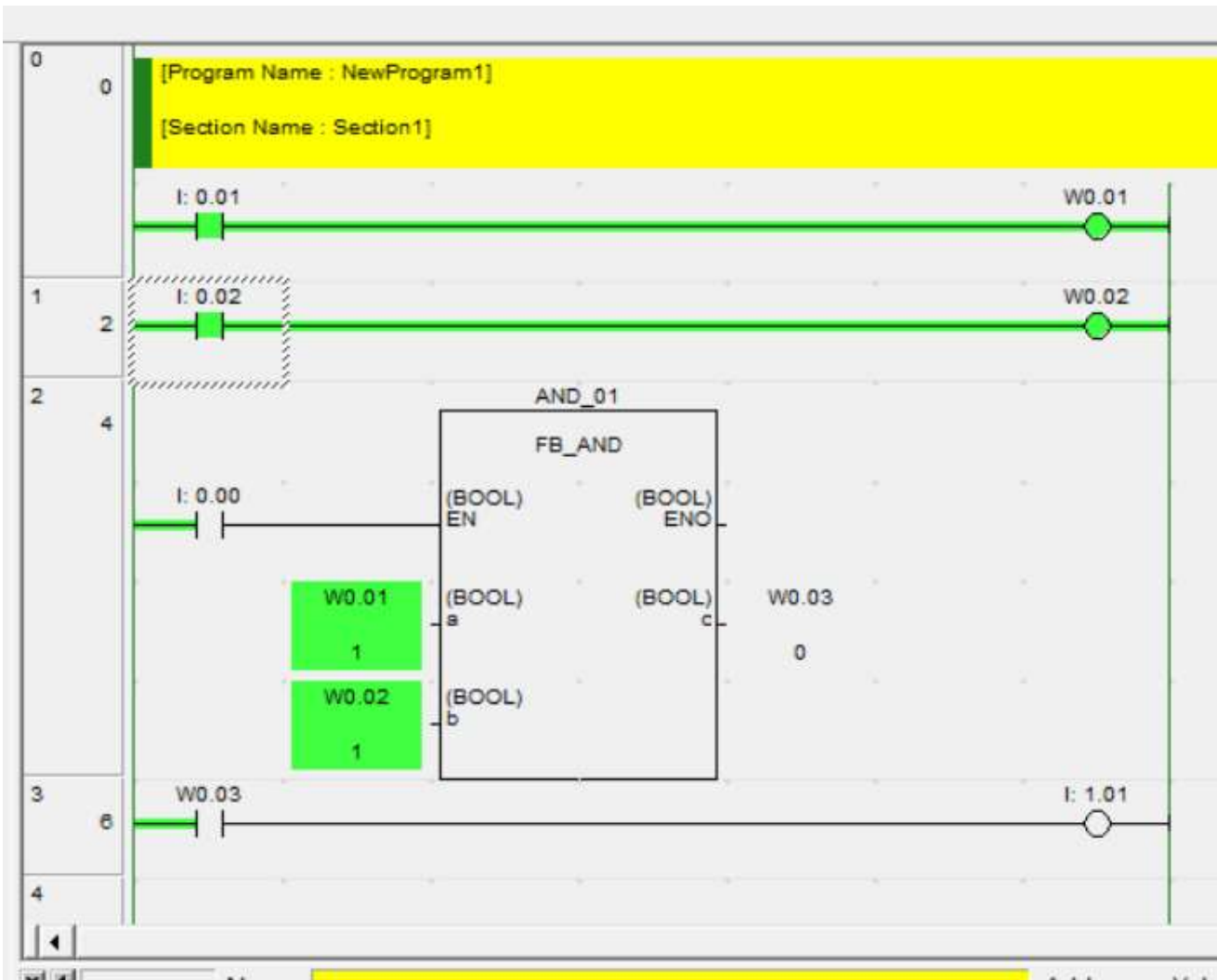


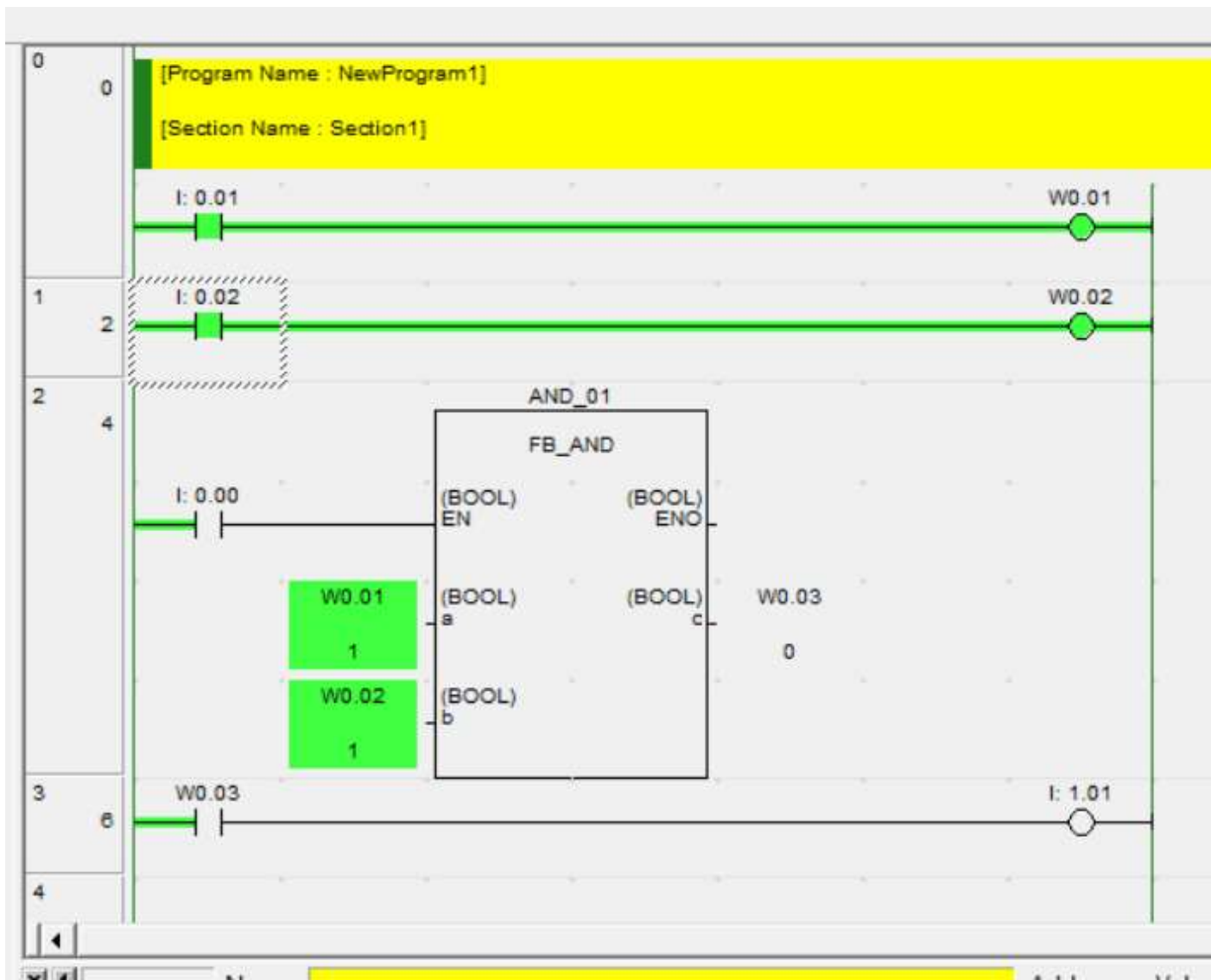
-|- New Contact ✕

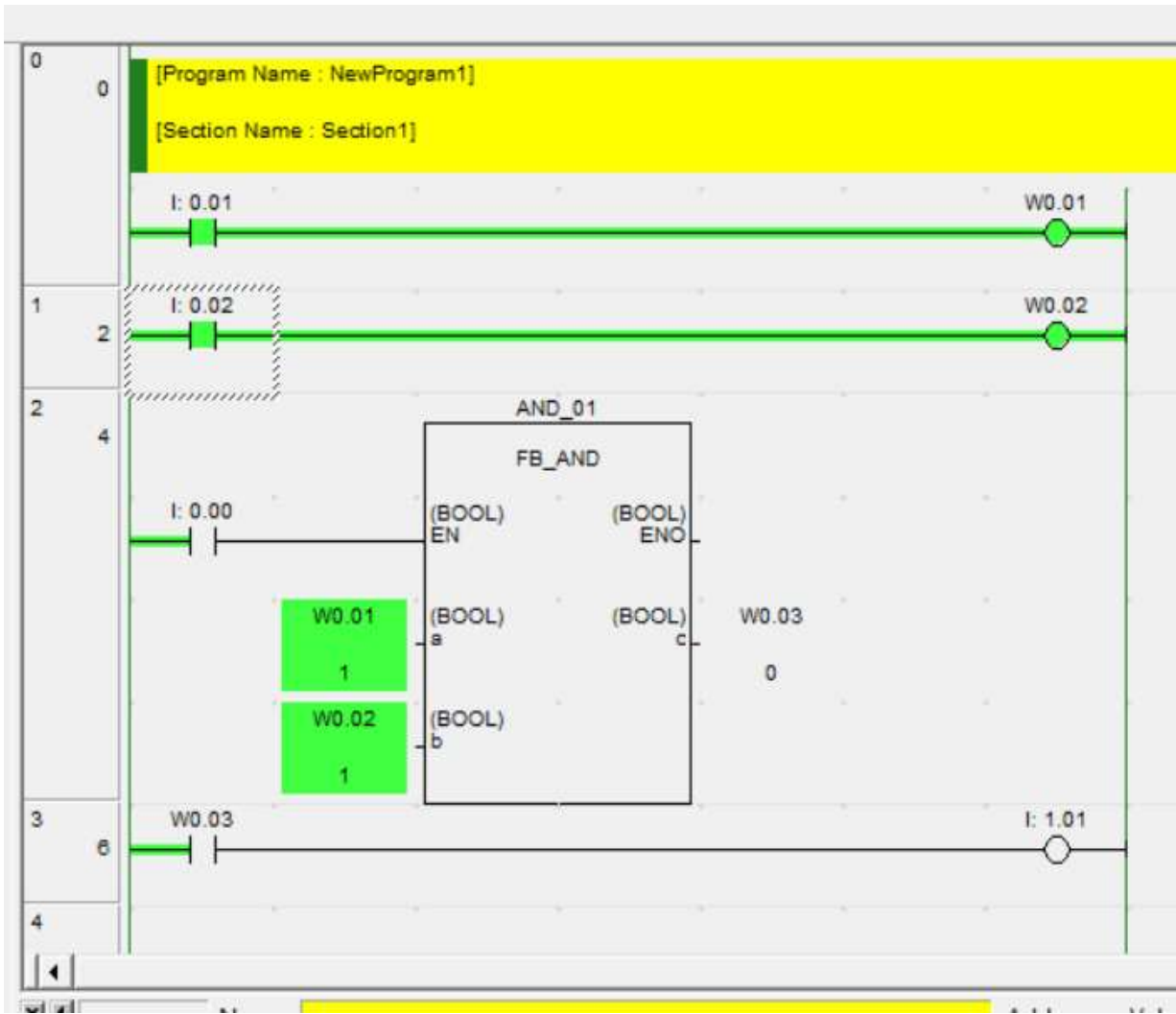
▼ Detail >> OK Cancel

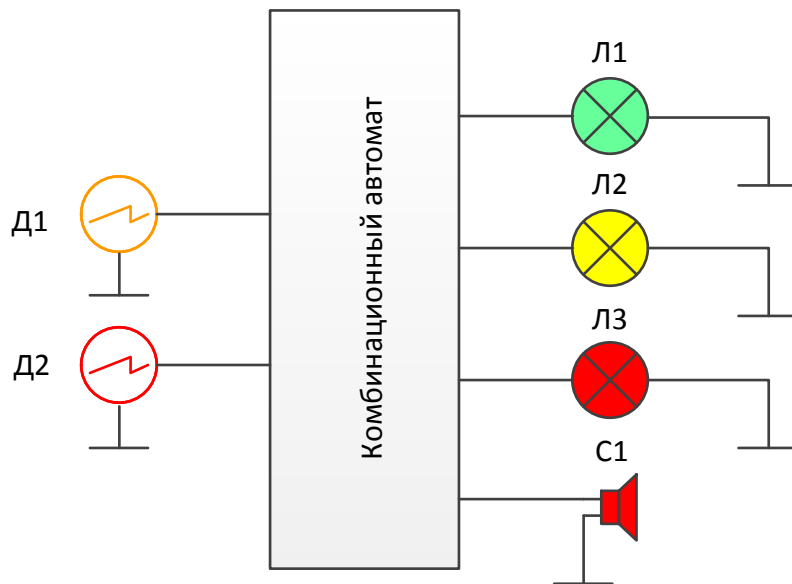












Описание алгоритма работы комбинационного автомата		
№	Выходное устройство	Условие срабатывания
1	Л1	Если ни один датчик не активирован (D1 и D2 имеют состояние логического нуля: $D1 = 0, D2 = 0$).
2	Л2	Если активирован датчик D1, а D2 не активирован ($D1 = 1, D2 = 0$).
3	Л3	Если активирован датчик D2. Состояние D1 не имеет значения ($D2 = 1, D1 = *$).
4	С1	Если активированы датчики D1 и D2 ($D1 = 1, D2 = 1$).

Таблица истинности (состояний) комбинационного автомата

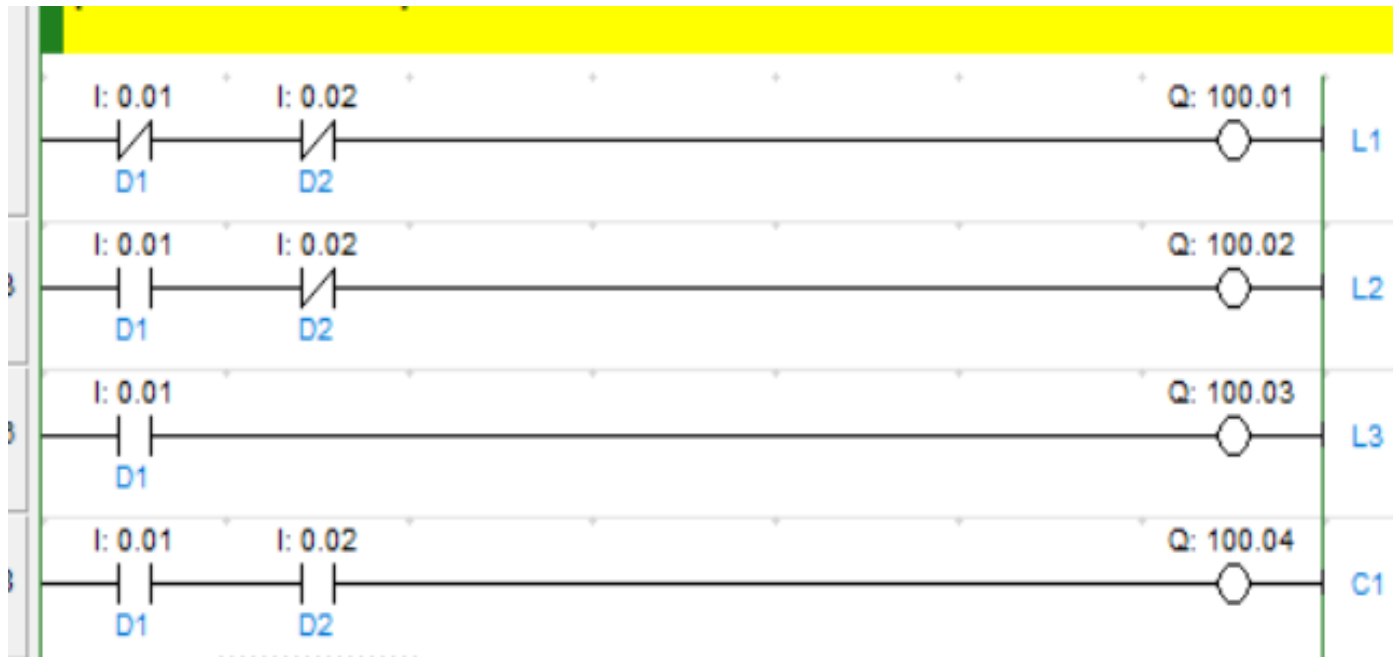
№	Вход		Выход			
	Д1	Д2	Л1	Л2	Л3	С1
1	0	0	1	0	0	0
2	0	1	0	0	1	0
3	1	0	0	1	0	0
4	1	1	0	0	1	1

$$Л1 = \overline{Д1} \wedge \overline{Д2}$$

$$Л2 = Д1 \wedge \overline{Д2}$$

$$Л3 = (\overline{Д1} \wedge Д2) \vee (Д1 \wedge Д2) = Д1 \text{ (закон исключения)}$$

$$С1 = Д1 \wedge Д2$$



Описание алгоритма работы комбинационного автомата		
№	Выходное устройство	Условие срабатывания
1	Л1	Если ни один датчик не активирован (Д1 и Д2 имеют состояние логического нуля: $D1 = 0, D2 = 0$).
2	Л2	Если активирован датчик Д1, а Д2 не активирован ($D1 = 1, D2 = 0$).
3	Л3	Если активирован датчик Д2. Состояние Д1 не имеет значения ($D2 = 1, D1 = *$).
4	С1	Если активированы датчики Д1 и Д2 ($D1 = 1, D2 = 1$).

Двойного отрицания:

$$\overline{\overline{A}} = A$$

Переместительный (коммутативный):

$$A \vee B = B \vee A$$

$$A \wedge B = B \wedge A$$

Сочетательный (ассоциативный):

$$(A \vee B) \vee C = A \vee (B \vee C)$$

$$(A \wedge B) \wedge C = A \wedge (B \wedge C)$$

Распределительный (дистрибутивный)

$$(A \vee B) \wedge C = (A \wedge C) \vee (B \wedge C)$$

$$(A \wedge B) \vee C = (A \vee C) \wedge (B \vee C)$$

Закон де Моргана:

$$\overline{A \vee B} = \overline{A} \wedge \overline{B}$$

$$\overline{A \wedge B} = \overline{A} \vee \overline{B}$$

Закон идемпотентности:

$$A \vee A = A$$

$$A \wedge A = A$$

Законы исключения констант:

$$A \vee 1 = 1$$

$$A \vee 0 = A$$

$$A \wedge 1 = A$$

$$A \wedge 0 = 0$$

Закон противоречия:

$$A \wedge \bar{A} = 0$$

Закон исключения третьего:

$$A \vee \bar{A} = 1$$

Закон поглощения:

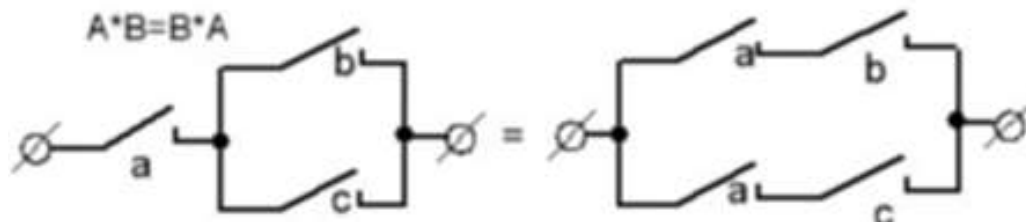
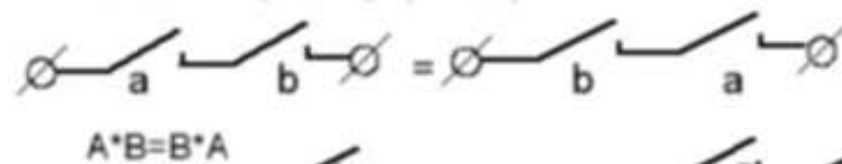
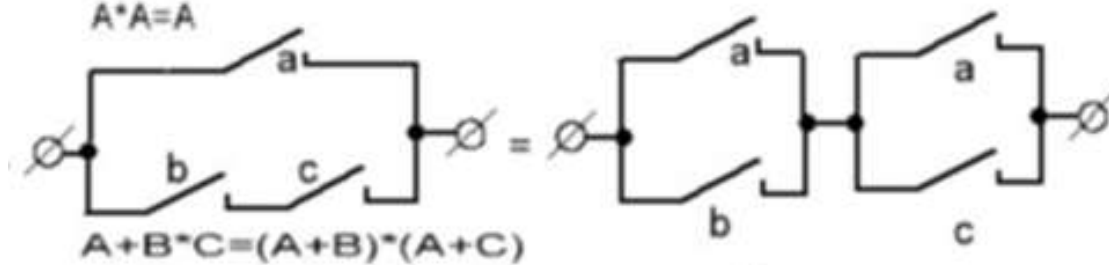
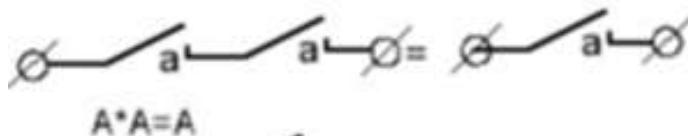
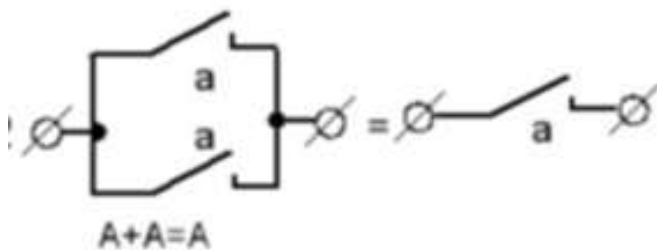
$$A \vee (A \wedge B) = A$$

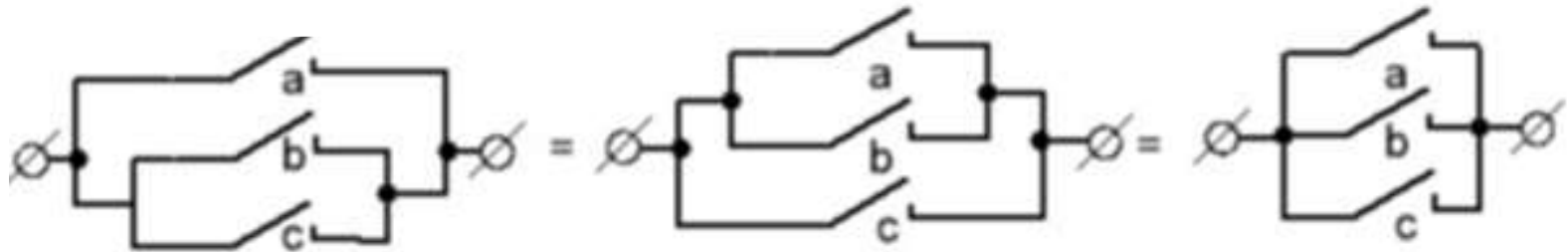
$$A \wedge (A \vee B) = A$$

Закон исключения (склеивания):

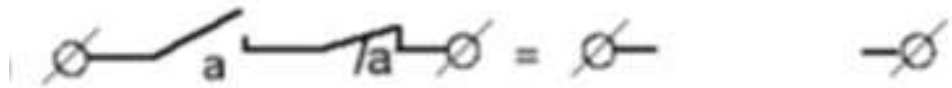
$$(A \wedge B) \vee (\bar{A} \wedge B) = B$$

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

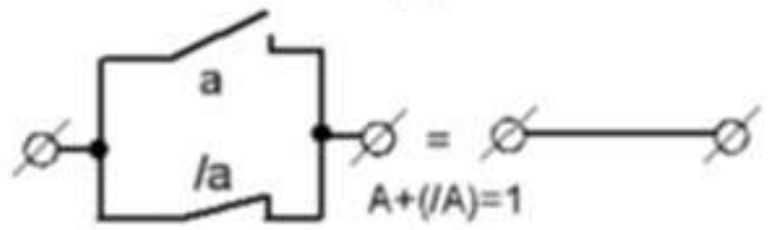




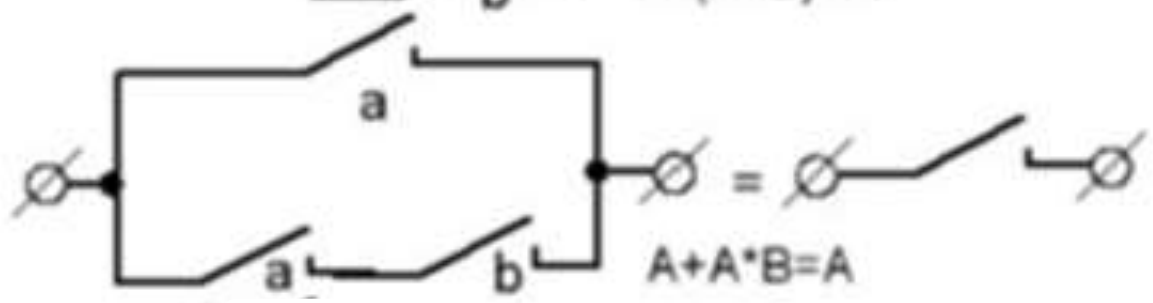
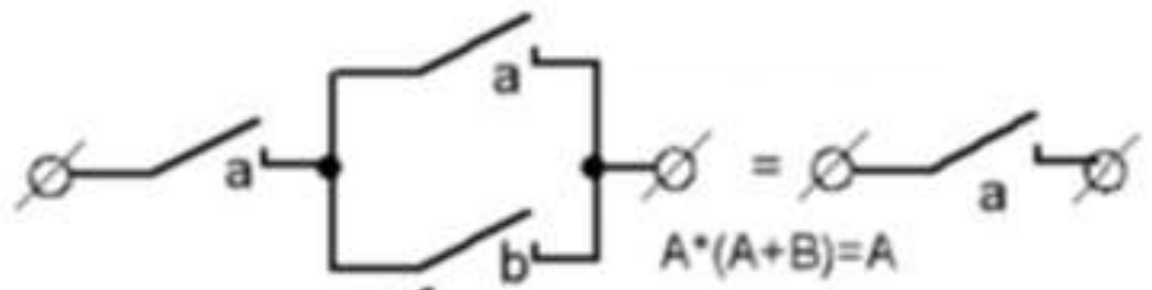
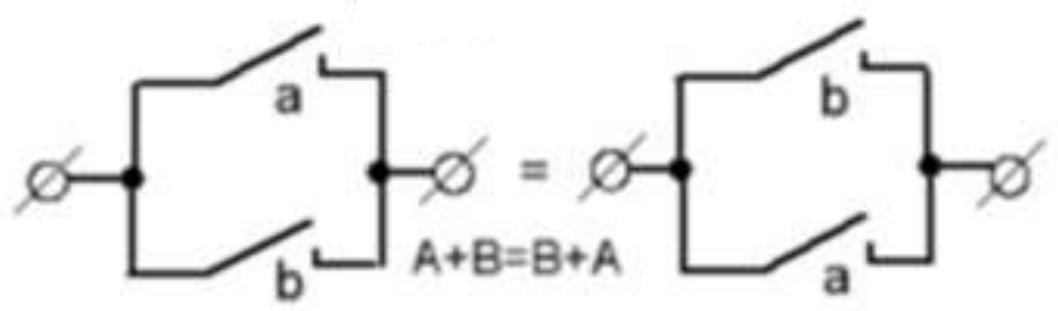
$$A + (B + C) = (A + B) + C = A + B + C$$



$$A * (A + B) = A$$



$$A + (A + B) = A + B$$



Дана схема системы автоматического управления, которая следит за уровнем воды, степенью ее загрязнения и ее температурой. Для этого в бассейн установлены три индикатора, которые отвечают за значения входов системы логического управления (СЛУ):

А – степень загрязнения воды;

В – уровень воды в бассейне;

С – температура воды.

Как только один из индикаторов включается, то на основе комбинационной схемы, в зависимости от логики системы, сигналы подаются на выходы СЛУ:

Н – магнитный пускатель, который запускает насос.

СЛ – сигнальная лампа, которая указывает недопустимую степень загрязнения воды;

С – сливное устройство.

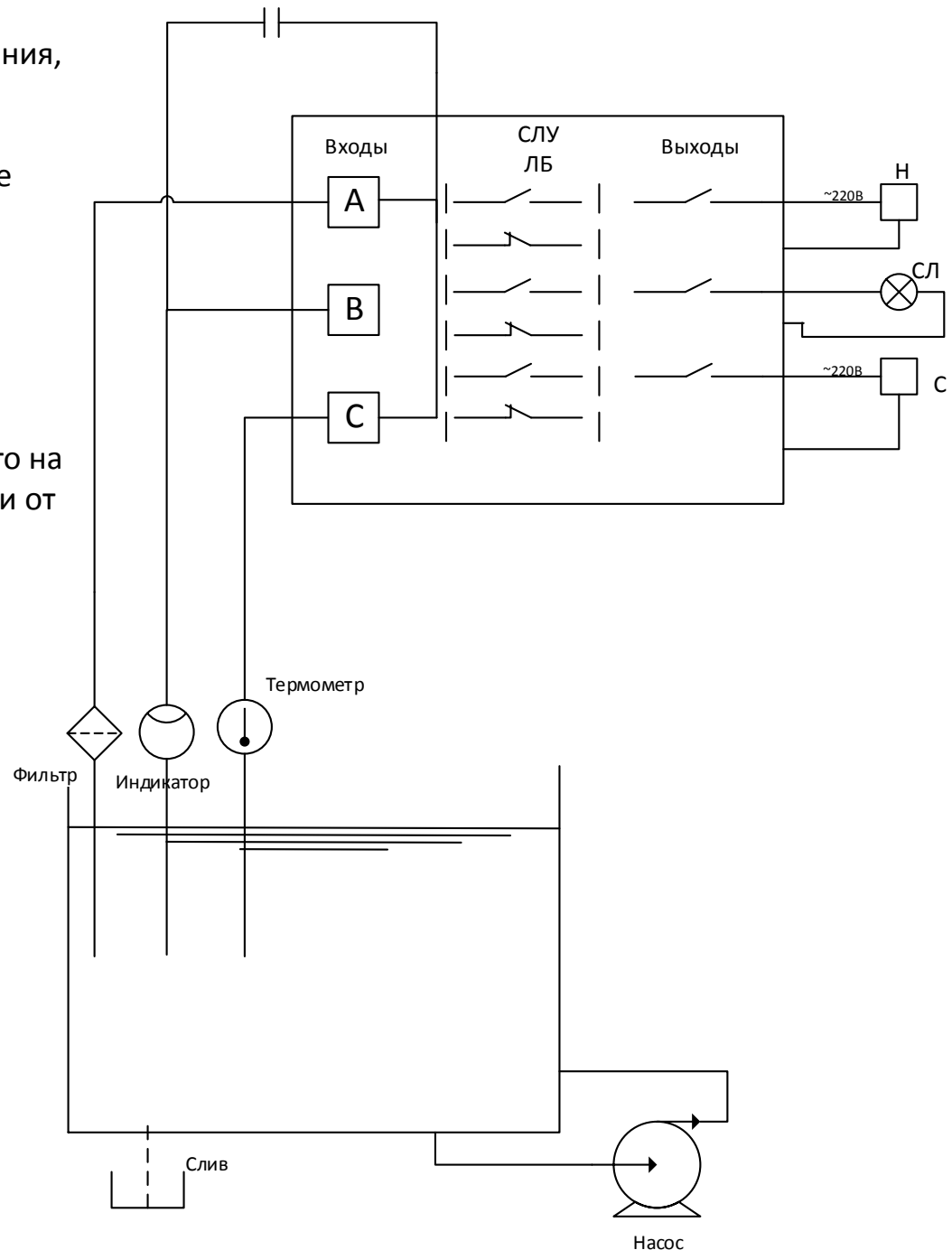
СЛУ – система логического управления;

ЛБ – логический блок;

Н – магнитный пускатель;

СЛ – сигнальная лампа;

С – сливное устройство.



Условия для работы Н:

Н срабатывает, если срабатывает С, но не срабатывают А и В.

Н срабатывает, если срабатывают С и А, но не срабатывает В.

Н срабатывает, если срабатывают С и В, но не срабатывает А.

Условия для работы СЛ:

СЛ срабатывает, если срабатывает А, но не срабатывают В и С.

СЛ срабатывает, если срабатывает В, но не срабатывают А и С.

СЛ срабатывает, если срабатывают А и В, но не срабатывает С.

Условия для работы С:

С срабатывает, если срабатывает А, В или С.

A	B	C	H	СЛ	c
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	0	1	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1
1	1	1	0	0	1

$$H_1 = \bar{a} \cdot \bar{b} \cdot c$$

$$H_2 = \bar{a} \cdot b \cdot c$$

$$H_3 = a \cdot \bar{b} \cdot c$$

$$СЛ_1 = a \cdot \bar{b} \cdot \bar{c}$$

$$СЛ_2 = \bar{a} \cdot b \cdot \bar{c}$$

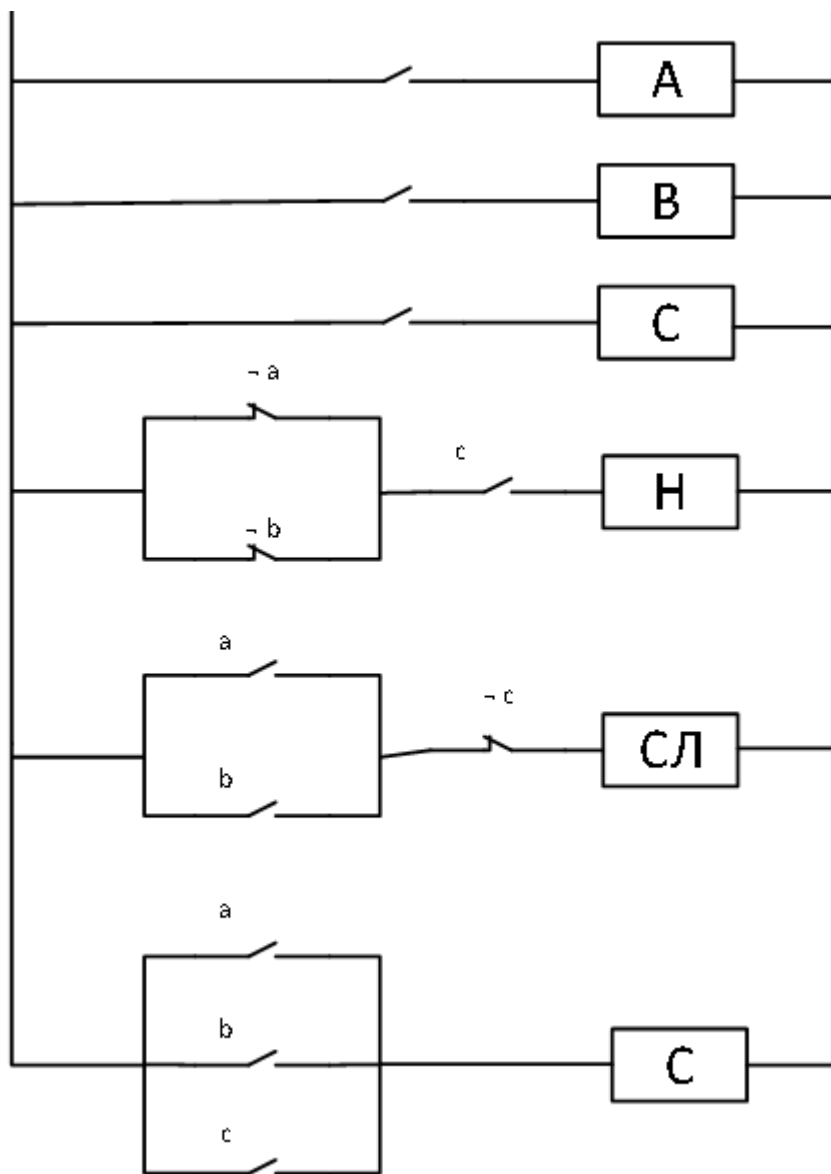
$$СЛ_3 = a \cdot \bar{b} \cdot \bar{c}$$

$$C = a + b + c$$

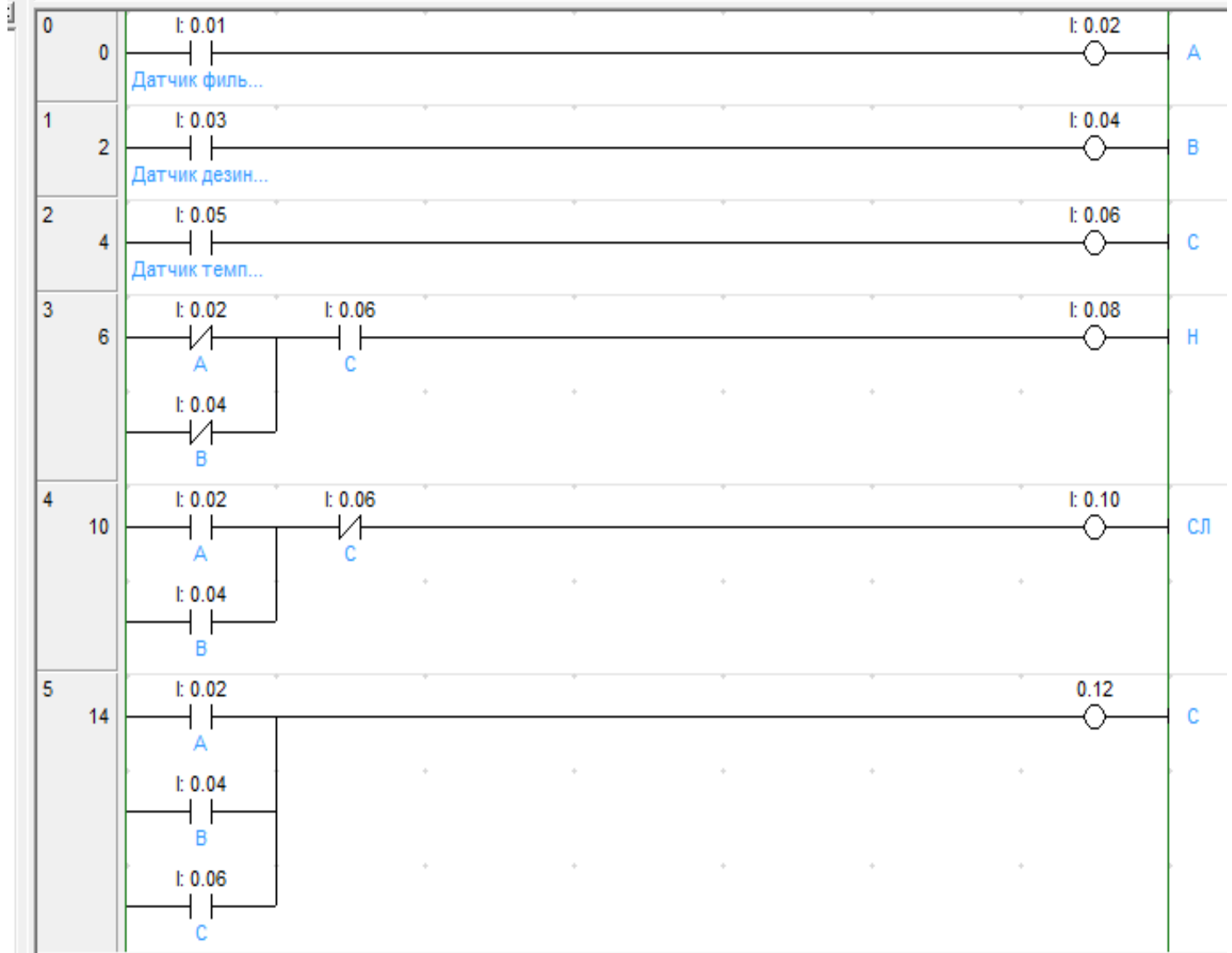
$$H = H_1 + H_2 + H_3 = \bar{a} \cdot \bar{b} \cdot c + a \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot c = \bar{b} \cdot c + \bar{a} \cdot b \cdot c = c \cdot (\bar{b} + \bar{a} \cdot b) = (\bar{a} + \bar{b}) \cdot c$$

$$СЛ = СЛ_1 + СЛ_2 + СЛ_3 = a \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} = a \cdot \bar{b} \cdot \bar{c} + b \cdot \bar{c} = \bar{c} \cdot (a \cdot \bar{b} + b) = (a + b) \cdot \bar{c}$$

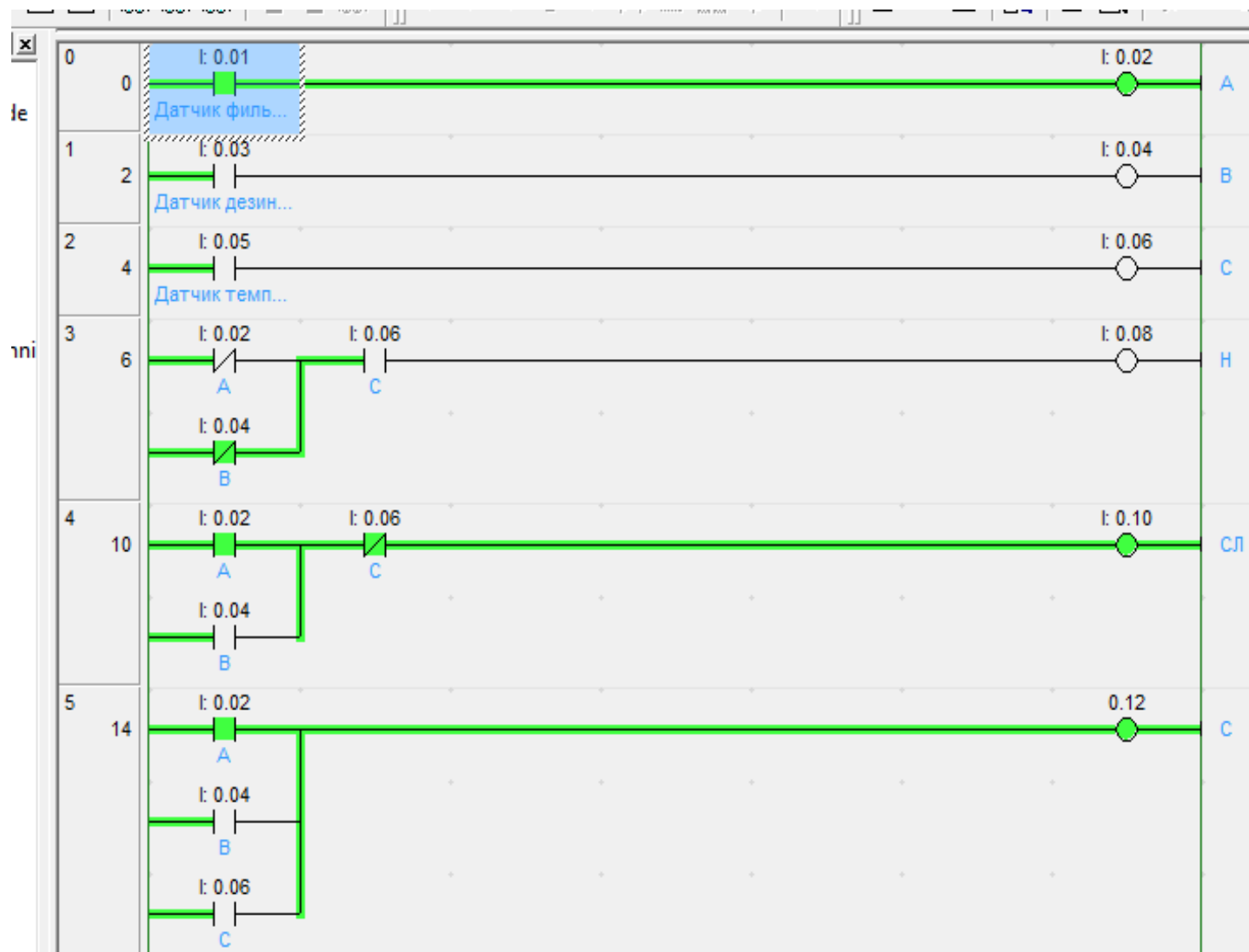
$$C = a + b + c$$



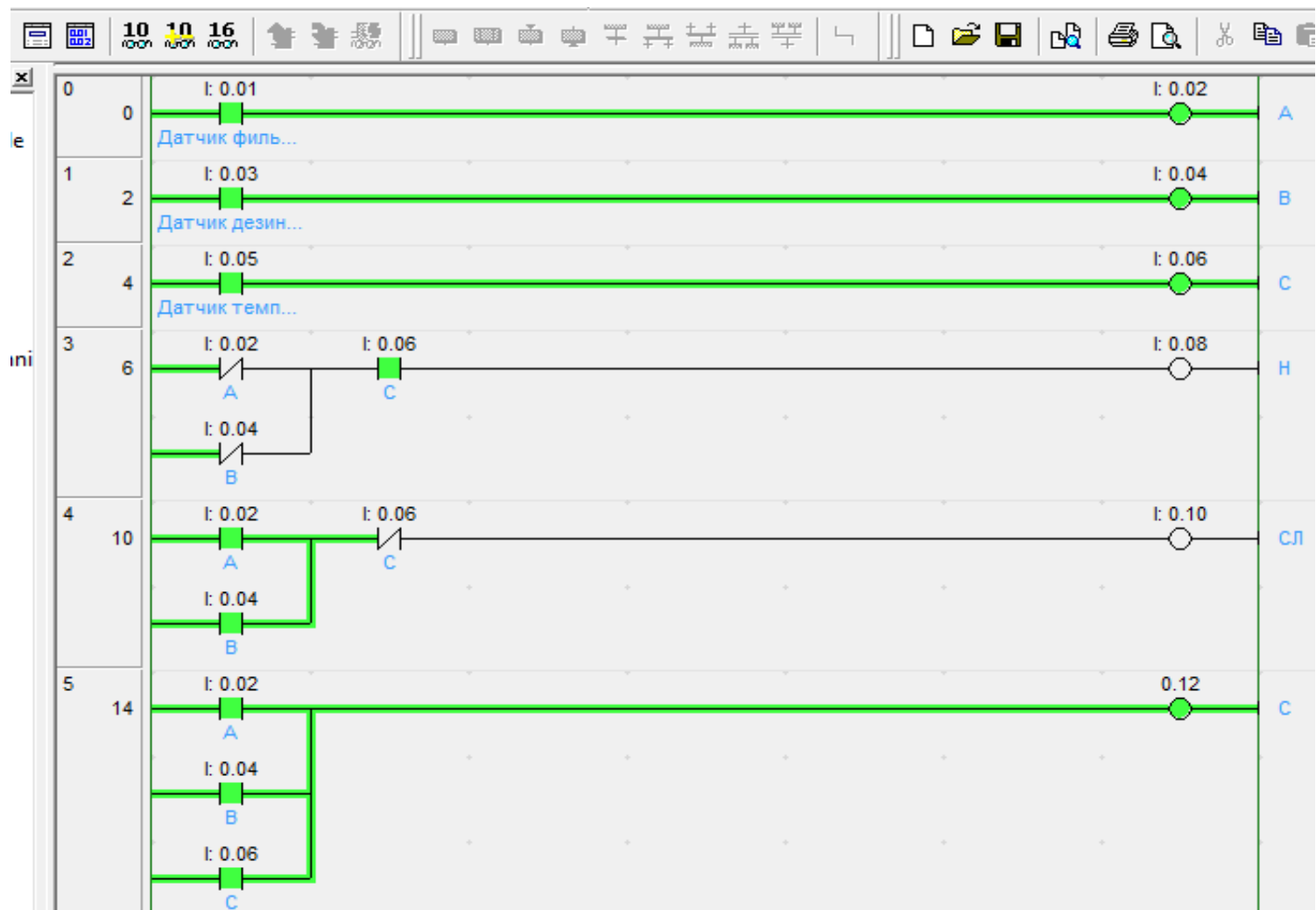
Датчик фильтра
Датчик индикатора
Датчик температуры
Цепь включения насоса
Цепь включения сигнальной лампы
Цепь включения слива



Работа системы при включенном датчике загрязнения воды



Работа системы при трех включенных датчиках



Элементы программы на ST

Данные

Выражения

Переменные

Массивы

Структуры

Присваивания

Управляющие конструкции

Выбора

Циклы

IF

CASE

FOR

WHILE

REPEAT

Оператор присваивания

:=

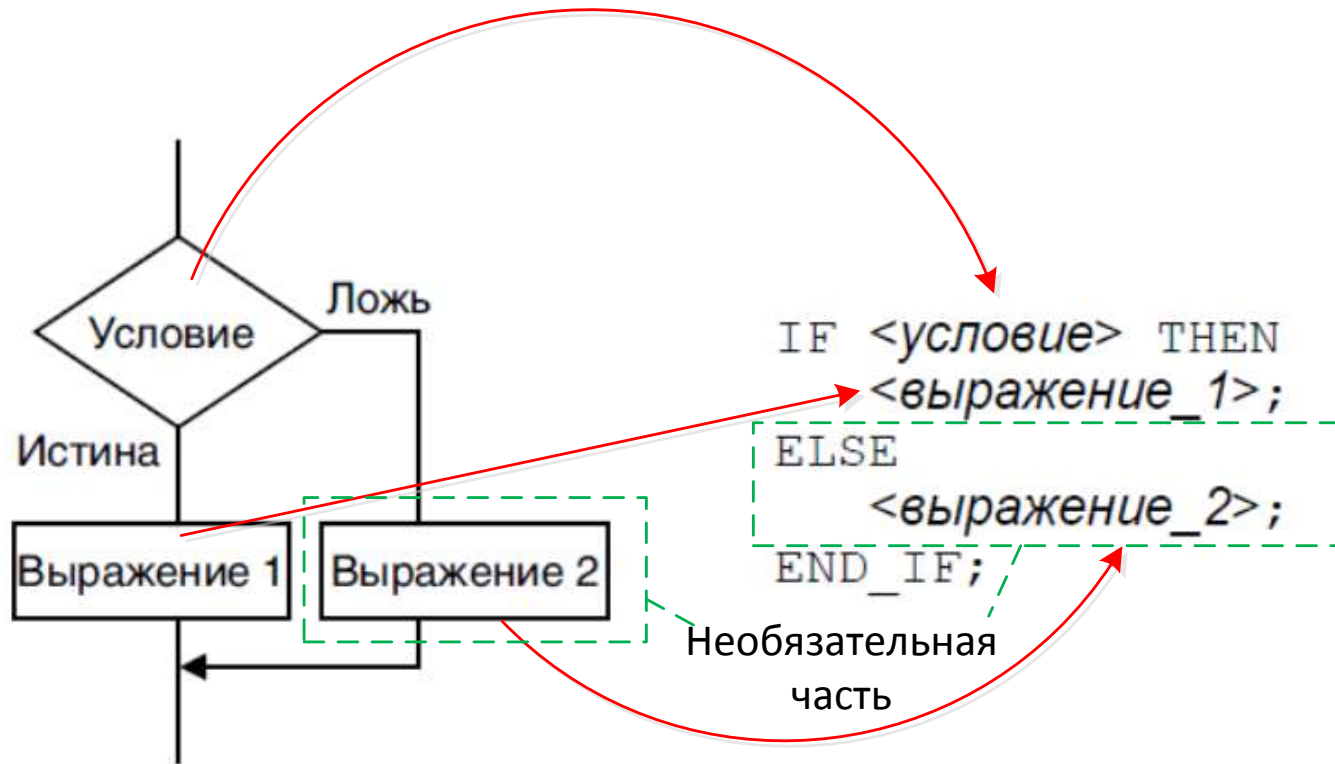
A := X + 1;

A := B;

A := 10;

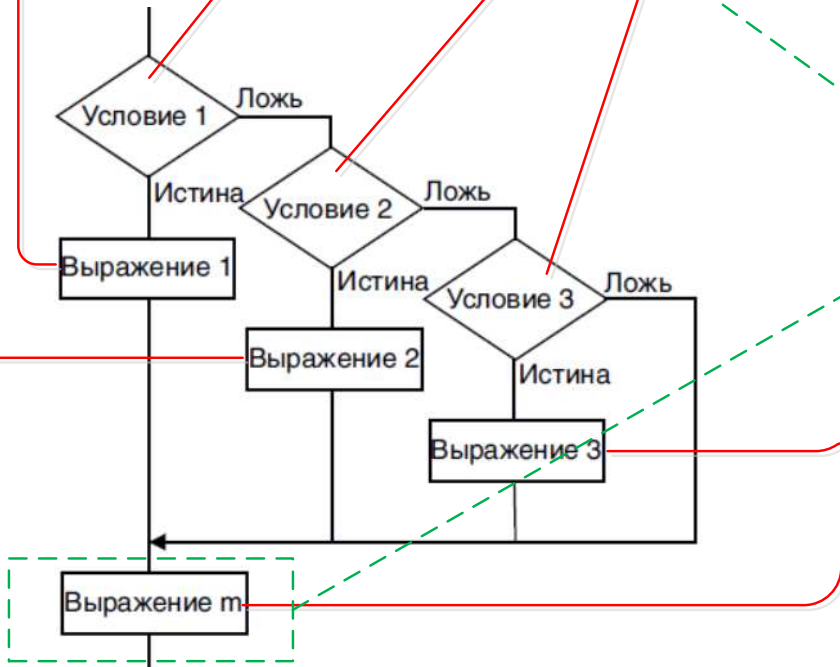
A := A + 1;

Примеры ввода в различных форматах		
Формат	Способ ввода	Пример (для десятичного значения 12)
Десятичный	Просто число (без префикса)	12
16-ричный	Число с префиксом 16#	16#C
8-ричный	Число с префиксом 8#	8#14
Двоичный	Число с префиксом 2#	2#1100
Строковый	Строковая константа заключается в одинарные кавычки. В данном случае «12» – это строка, состоящая из двух символов «1» и «2»	'12'



```
IF A>0 THEN  
    X:=25;  
ELSE  
    X:=10;  
END_IF;
```

```
IF <условие_1> THEN <выражение_1>;  
  ELSIF <условие_2> THEN <выражение_2>;  
  ELSIF <условие_3> THEN <выражение_3>;  
  ELSE <выражение_m>;  
END_IF;
```

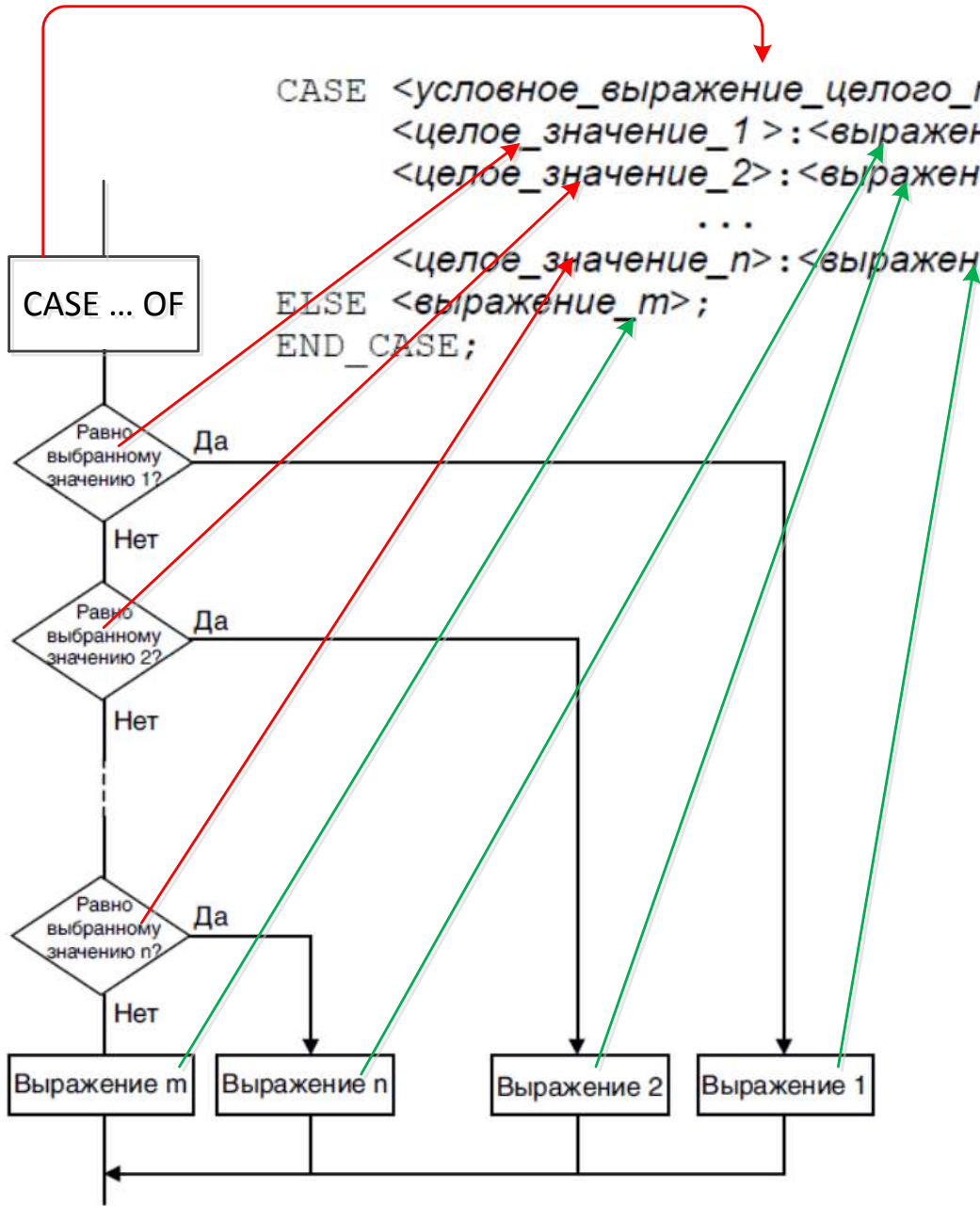


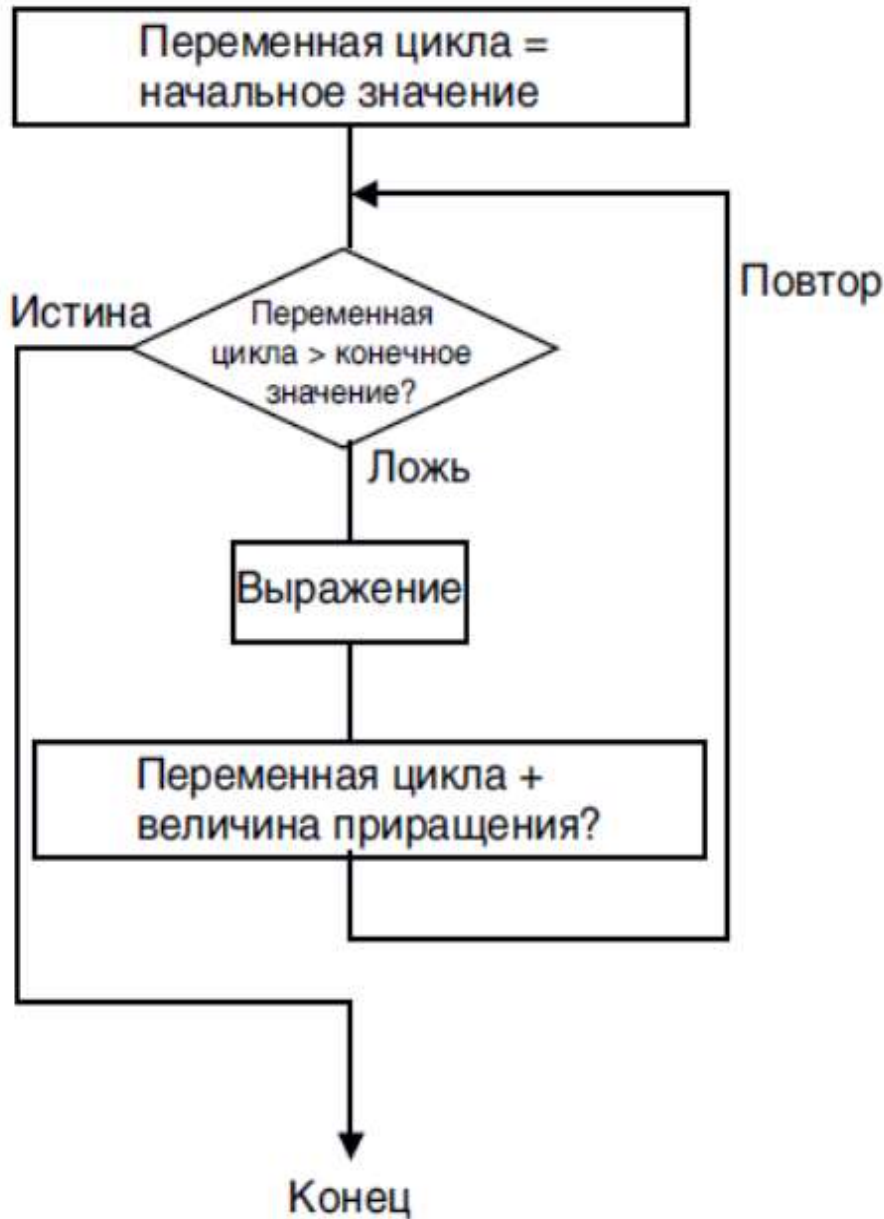
Необязательная
часть

```
IF A>0 THEN X:=10;  
  ELSIF B=1 THEN X:=1;  
  ELSIF B=2 THEN X:=2;  
  ELSE X:=0;  
END_IF;
```

```
CASE <условное_выражение_целого_типа> OF
  <целое_значение_1> : <выражение_1>;
  <целое_значение_2> : <выражение_2>;
  ...
  <целое_значение_n> : <выражение_n>;
ELSE <выражение_m>;
END_CASE;
```

```
CASE A OF
  1:X:=1;
  2,5:X:=2;
  6..10:X:=3;
  11,12,15..20:X:=4;
  ELSE Y:=0;
END_CASE;
```

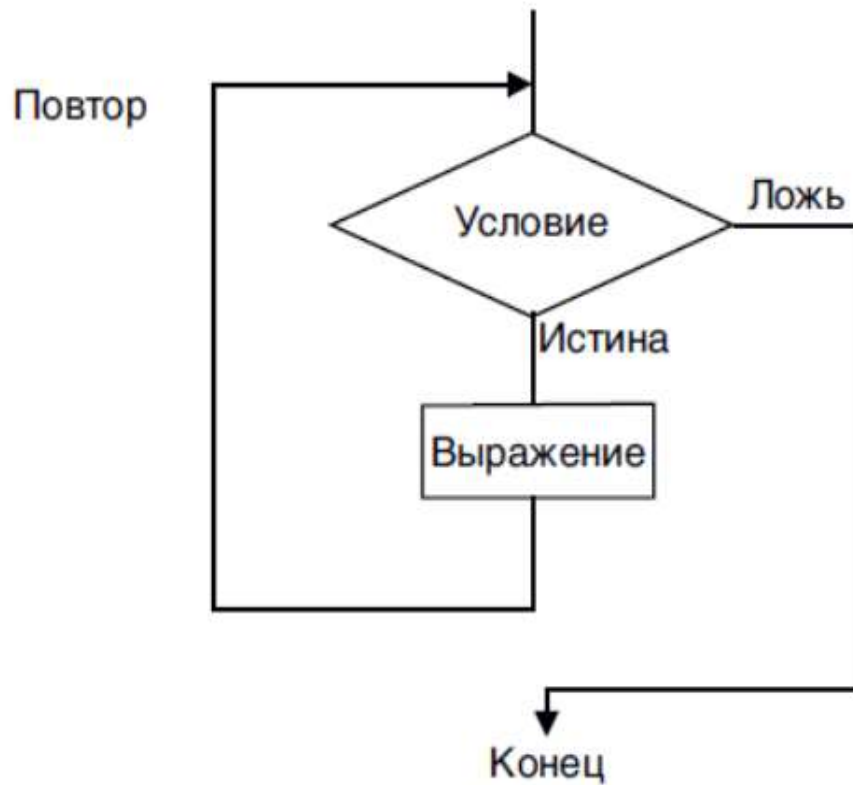




Цикл со счетчиком FOR

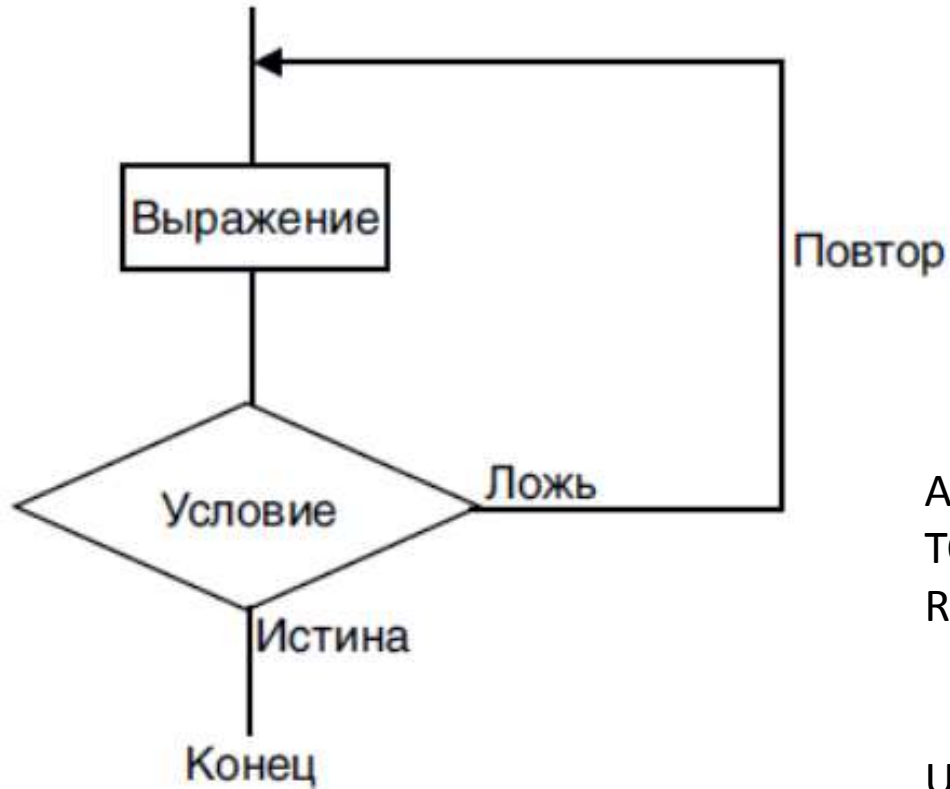
```
MAX:=0;  
MIN:=1000;  
FOR n:=1 TO 50 BY 1 DO  
    IF DATA[n]>MAX THEN  
        MAX:=DATA[n];  
    END_IF;  
    IF DATA[n]<MIN THEN  
        MIN:=DATA[n];  
    END_IF;  
END_FOR;
```

Цикл с предусловием WHILE



```
n:=1;  
WHILE X<3000 DO  
  X:=X*2;  
  DATA[n]:=X;  
  n:=n+1;  
END_WHIE;
```

Цикл с постусловием REPEAT



```
A:=1;  
TOTAL:=0;  
REPEAT  
    TOTAL:=TOTAL+A;  
    A:=A+1;  
UNTIL A>10  
END_REPEAT;
```

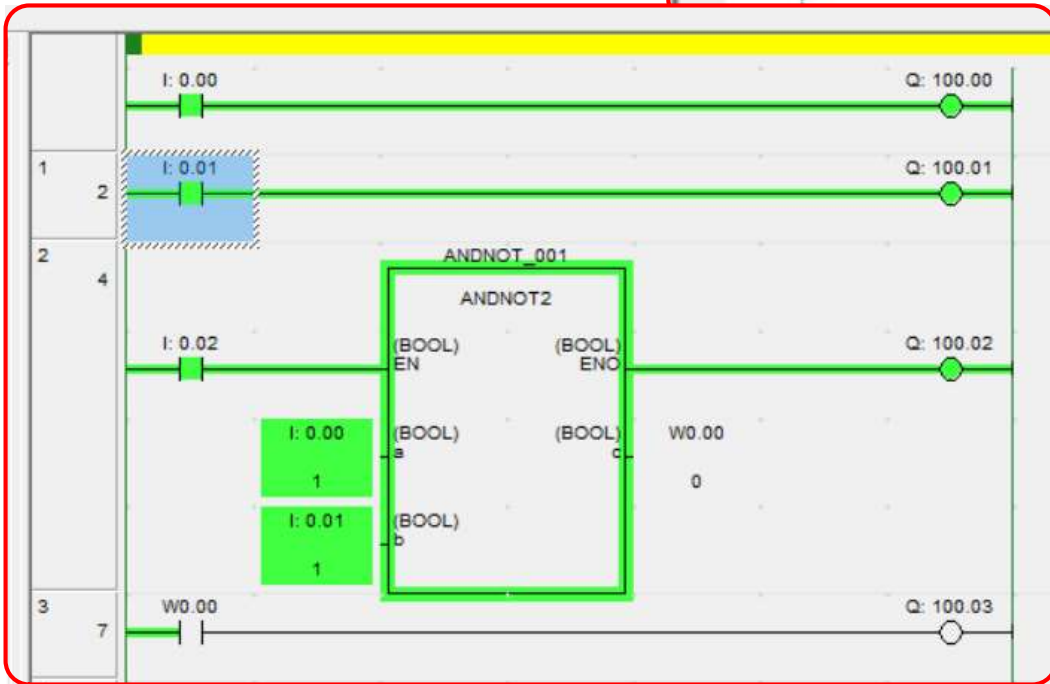
Определение переменных

Name	Data Type
EN	BOOL
a	BOOL
b	BOOL

Internals	Inputs	Outputs	In Out

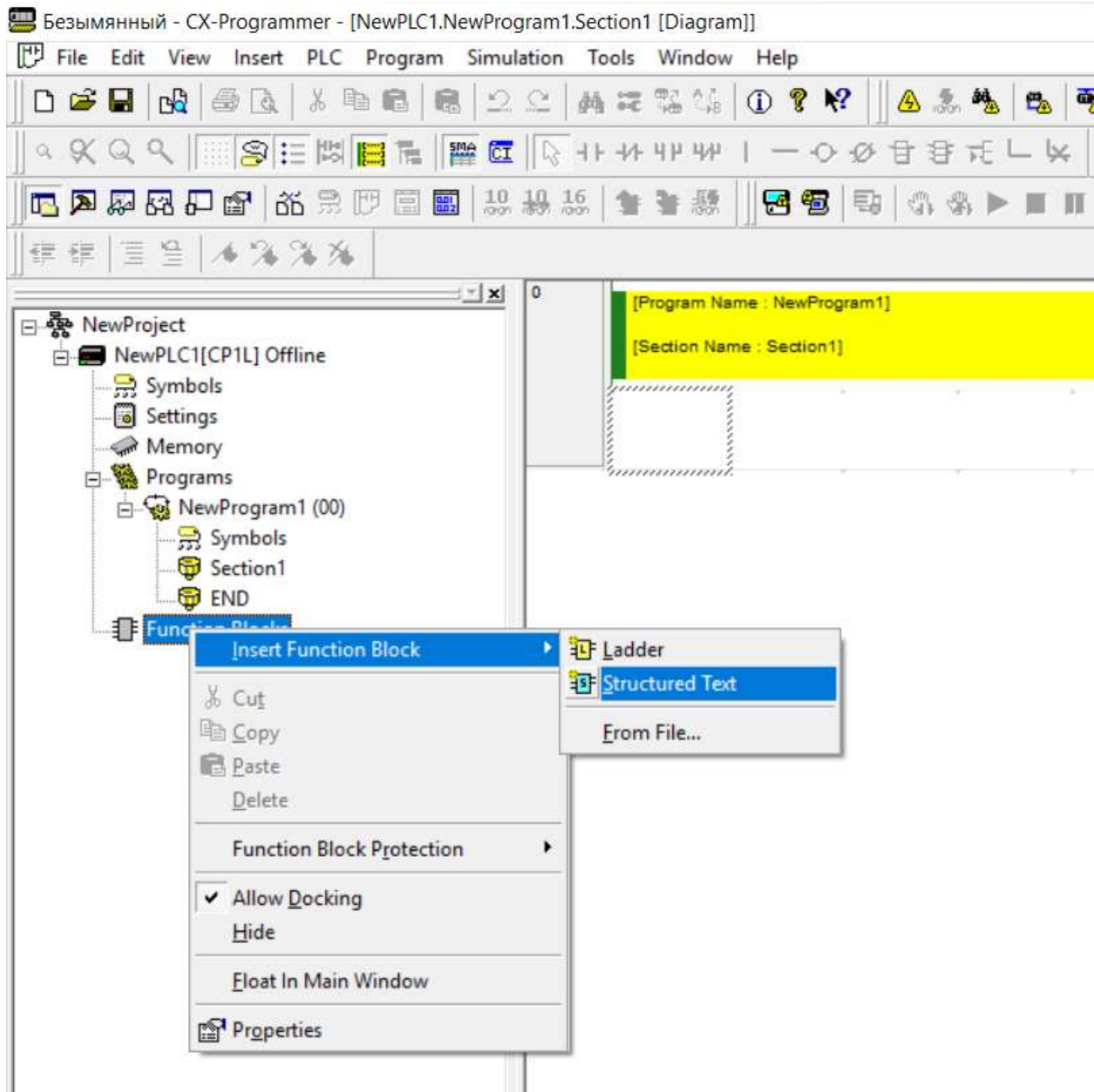
```
1 c := NOT (a AND b);
```

Алгоритм



Определение функционального блока

Экземпляр функционального блока



Function Block Properties



General

Protection

Comments

Memory

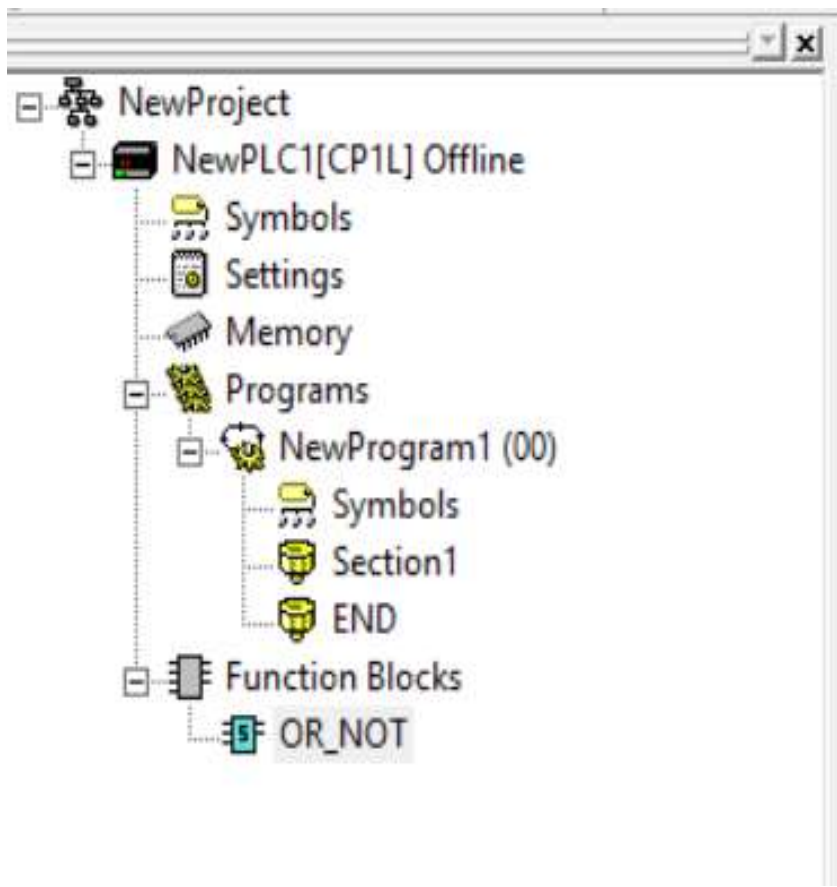
Name:

OR NOT

Display the inside of FB

Author:

Revision:



Безымянный - CX-Programmer - [NewPLC1.OR_NOT [FB Structured Text]]



The screenshot shows the project tree on the left and a variable declaration table on the right. The project tree is expanded to show the 'Function Blocks' folder, which contains the 'OR_NOT' block. The variable declaration table has two columns: 'Name' and 'Data Type'. The first row contains 'EN' and 'BOOL'. A context menu is open over the table, with 'Insert Variable...' selected. The menu also includes options like 'Edit...', 'Insert Variable', 'Cut', 'Copy', 'Paste', 'Find...', 'Replace...', 'Delete', and 'Rename'. Below the table, there are tabs for 'Internals' and 'In Out', and a vertical line with the number '1' is visible.

Name	Data Type
EN	BOOL

New Variable



Name:

Data Type:

Usage:

Initial Value: Retain

OK

Cancel

Advanced...

Comment:

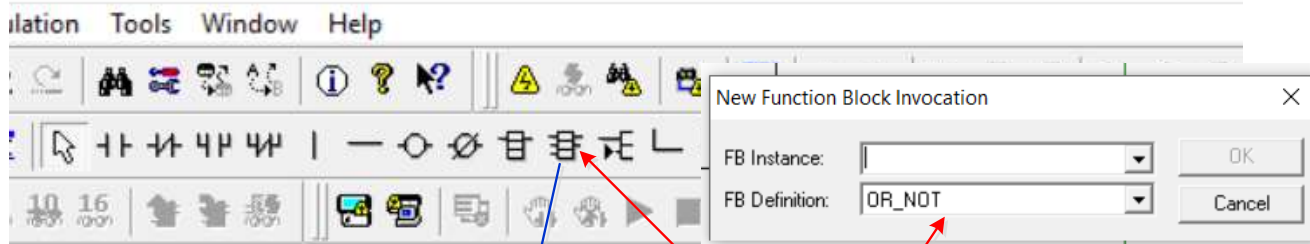
Name	Data Type	AT	Initial Value	Ret
EN	BOOL		FALSE	
A	BOOL		FALSE	
B	BOOL		FALSE	
<div style="display: flex; justify-content: space-around; border: 1px solid #ccc; padding: 2px;"> Internals Inputs Outputs In Out Externals </div>				
1				

Name	Data Type	AT	Initial Value	Ret
ENO	BOOL		FALSE	
C	BOOL		FALSE	
<div style="display: flex; justify-content: space-around; border: 1px solid #ccc; padding: 2px;"> Internals Inputs Outputs In Out Externals </div>				
1				

Name	Data Type
ENO	BOOL
C	BOOL

Internals Inputs Outputs In

```
1 C := NOT(A OR B);
```



Реализацию функционального блока можно создать с помощью данной кнопки

Назначение ячеек памяти, из которых будет копироваться информация во входные переменные (входные параметры) A и B функционального блока

Результат правильной работы функционального блока (обязательно должен быть «нагружен»)

Назначение выходной ячейки, куда будет помещен результат выполнения операции (значение выходной переменной C)

Кнопка активации функционального блока (аналог выбора кристалла («чип селек») в интегральных схемах. Обязательно должен быть подключен.

Создание на ST RS-триггера в виде функционального блока

Таблица переходов RS-триггера				
R	S	Q(t)	Q(t+1)	Комментарии
0	0	0	0	Режим хранения информации R=S=0
0	0	1	1	
0	1	0	1	
0	1	1	1	Режим установки единицы S=1
1	0	0	0	Режим установки нуля R=1
1	0	1	0	
1	1	0	-	R=S=1 запрещённая комбинация
1	1	1	-	

$$Q(t + 1) = (\bar{R} \wedge \bar{S} \wedge Q(t)) \vee (\bar{R} \wedge S \wedge \overline{Q(t)}) \vee (R \wedge \bar{S} \wedge \overline{Q(t)})$$

$$\bar{Q} = \text{Not } Q$$

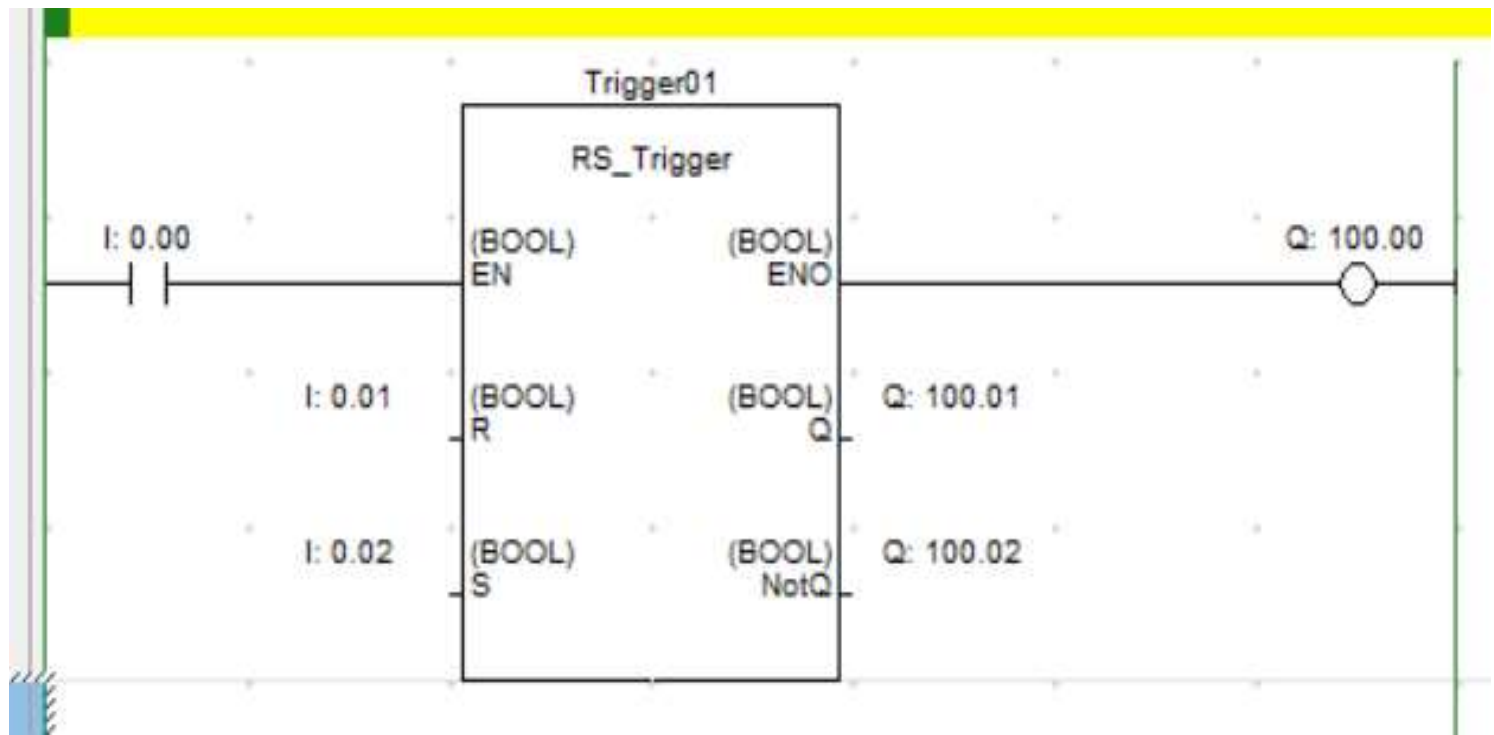
Name	Data Type	AT	Initial Value
EN	BOOL		FALSE
R	BOOL		FALSE
S	BOOL		FALSE

Name	Data Type	AT	Initial Value
ENO	BOOL		FALSE
Q	BOOL		FALSE
NotQ	BOOL		FALSE

```

1 IF (NOT((R = TRUE) AND (S = TRUE))) THEN
2     Q := ((NOT R) AND (NOT S) AND Q) OR
3         ((NOT R) AND S AND (NOT Q)) OR
4         ((NOT R) AND S AND Q);
5 END_IF;
6 NotQ := NOT Q;

```



Создадим функциональный блок на языке ST управления светофором, работающий в следующем режиме. Время красного сигнала – 8 секунд, желтого (после красного) – 3 секунды, зеленого – 5 секунд, желтого (после зеленого) – 3 секунды. Функциональный блок тактируется положительным фронтом внешних импульсов, следующих с периодом 1 секунда.

Name	Data Type	AT	Initial Value	R
temp	BOOL		FALSE	
s	INT		0	

Internals Inputs Outputs In Out Externals

Name	Data Type	AT	Initial Value
EN	BOOL		FALSE
ClockPulse	BOOL		FALSE

Internals Inputs Outputs In Out Externals

Name	Data Type	AT	Initial Value
ENO	BOOL		FALSE
Red	BOOL		FALSE
Yellow	BOOL		FALSE
Green	BOOL		FALSE

Name	Data Type	AT
EN	BOOL	
ClockPulse	BOOL	

Internals	Inputs	Outputs	In Out	Externals
-----------	--------	---------	--------	-----------

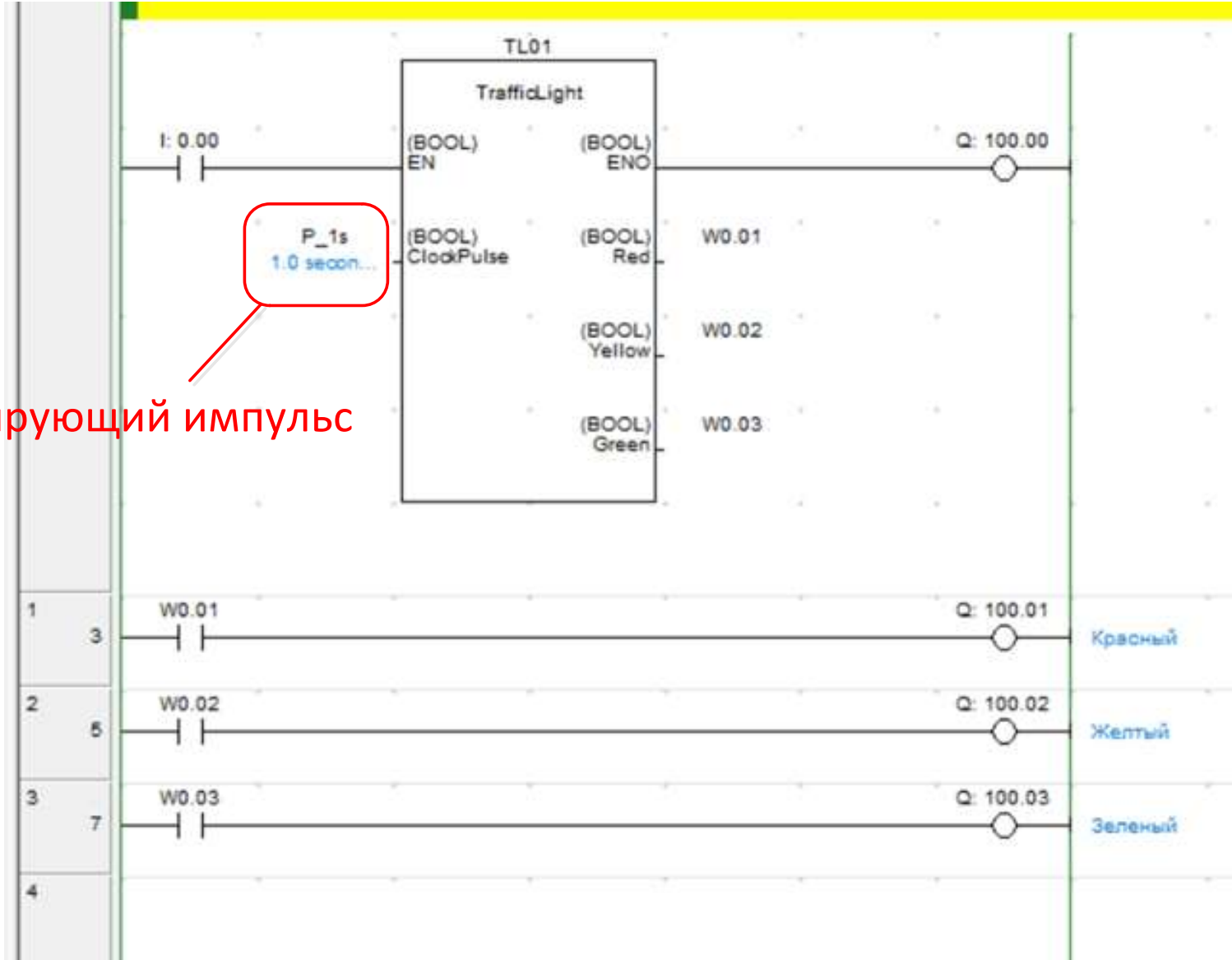
```
1 IF(ClockPulse <> temp) THEN
2   IF(ClockPulse = TRUE) THEN
3     s := s + 1;
4   END_IF;
5   temp := ClockPulse;
6   IF(s > 18) THEN
7     s := 0;
8   END_IF;
9 END_IF;
10 CASE s OF
11   0..7 : Red := TRUE; Green := FALSE; Yellow := FALSE;
12   8..10 : Yellow := TRUE; Green := FALSE; Red := FALSE;
13   11..15 : Green := TRUE; Red := FALSE; Yellow := FALSE;
14   16..18 : Yellow := TRUE; Green := FALSE; Red := FALSE;
15 END_CASE;
```

Выделение
положительного фронта

Блок тактирования
(срабатывает по положительному
и отрицательному фронту)

Блок «обнуления». Перезапускает
циклический алгоритм работы светофора

Переменная, содержащее значение
предыдущего фронта



Тактирующий импульс

Принципы построения и производства технических средств автоматизации

Обеспечивает все виды технической совместимости

Конструктивная совместимость	Функциональная совместимость	Коммутационная совместимость	Информационная совместимость
Обеспечивается базовыми несущими конструкциями (БНК)	Обеспечивается стандартными электронными модулями (СЭМ)	Обеспечивается интерфейсными модулями	Обеспечивается программными модулями

В радиоэлектронной аппаратуре различного назначения

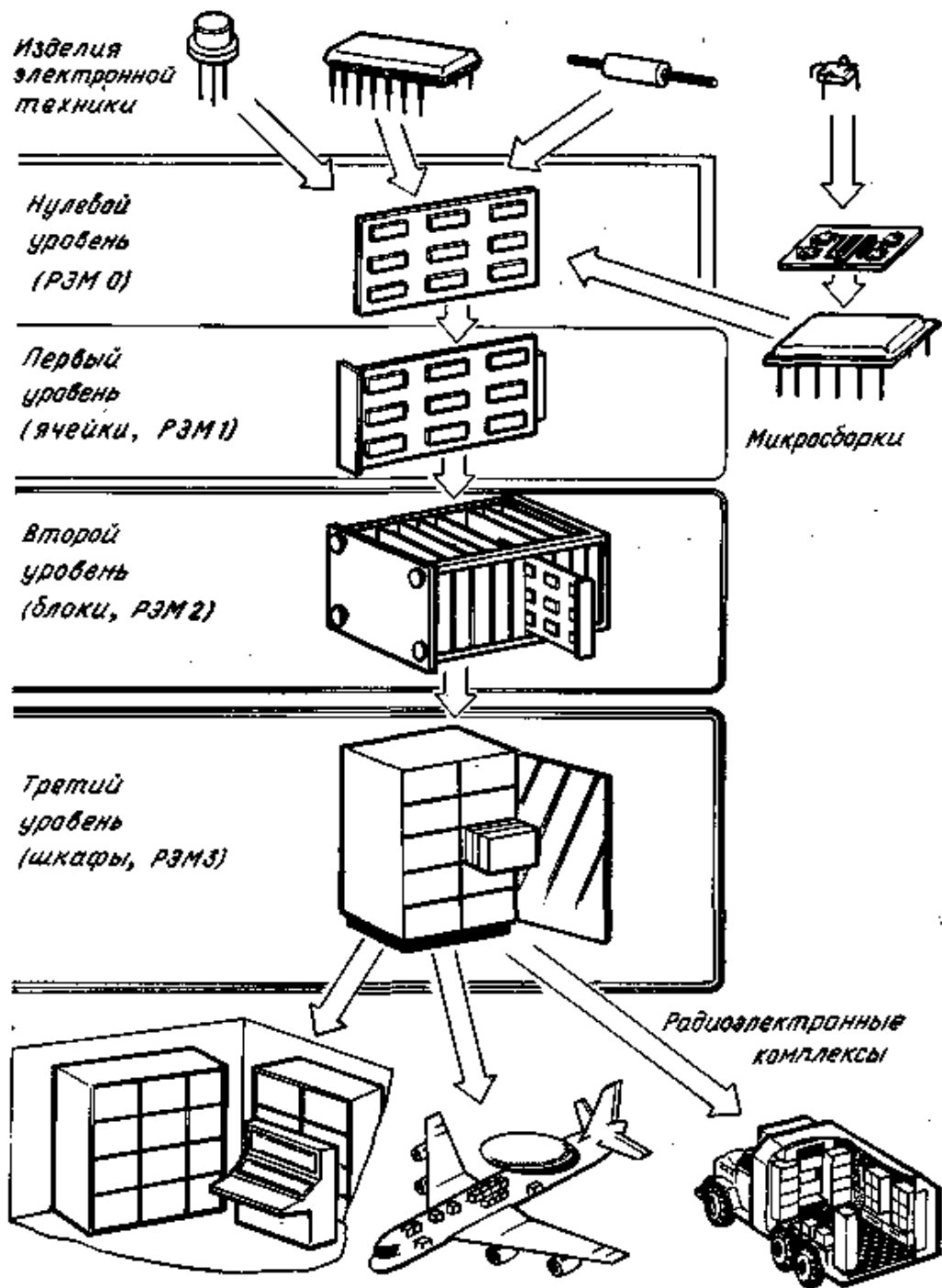
Аппаратура связи	РЛС и навигационные системы	Информационно-управляющие системы и АСУ	Контрольно-измерительная аппаратура	Системы единого времени	Вычислительная техника
------------------	-----------------------------	---	-------------------------------------	-------------------------	------------------------

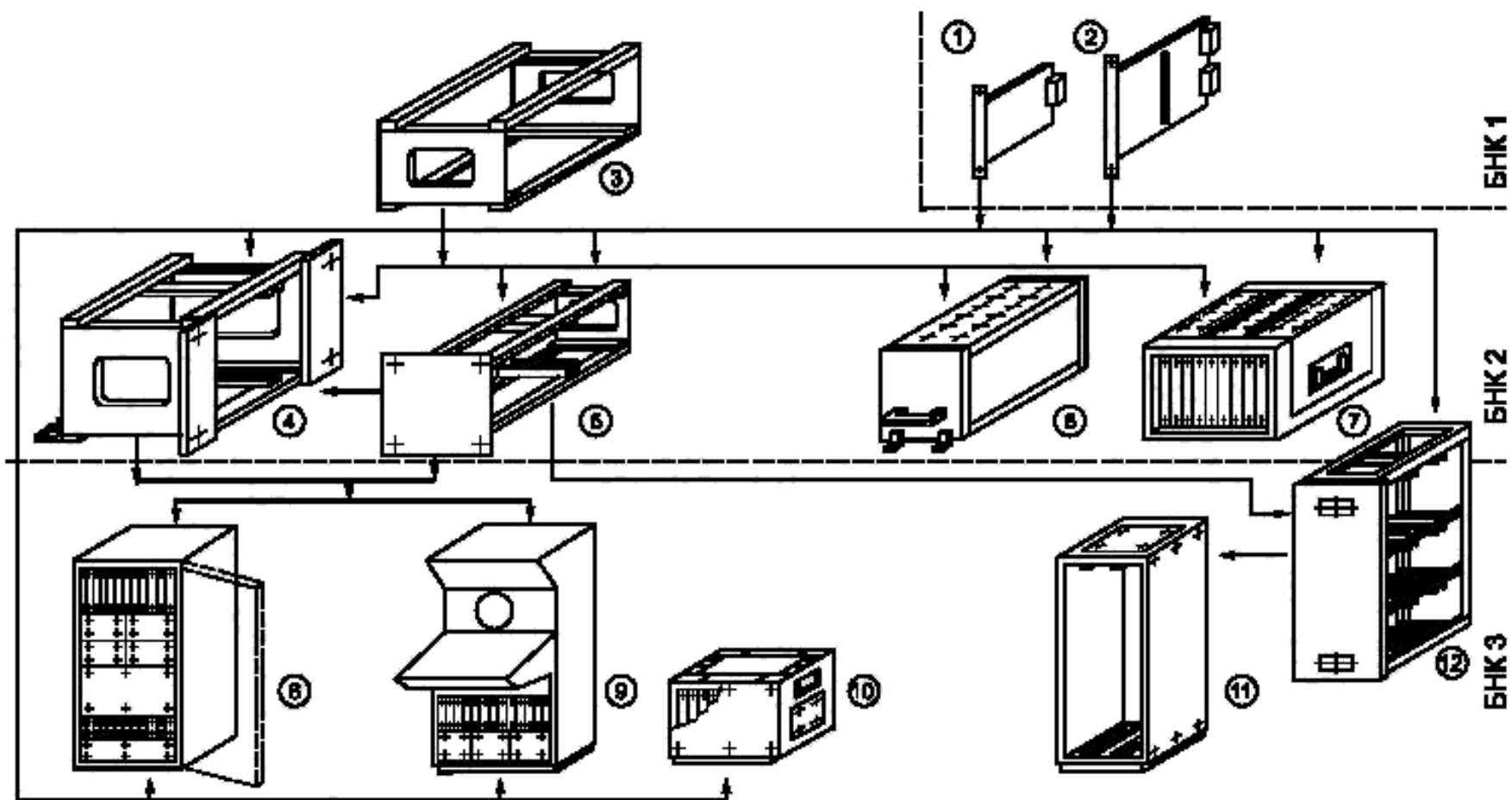
Устанавливает разукрупнение РЭС по функциональной сложности



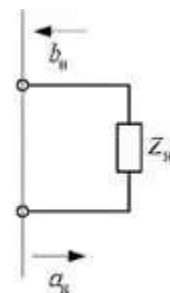
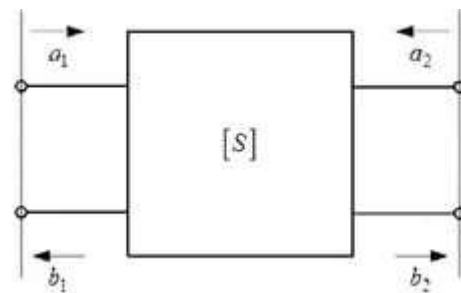
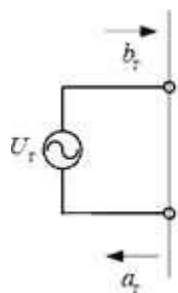
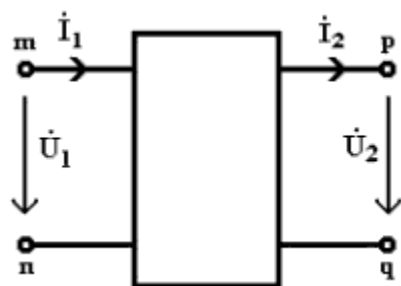
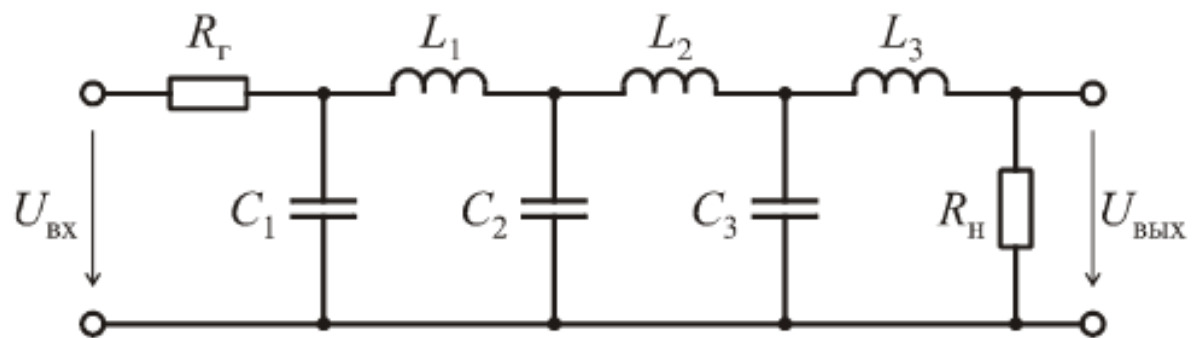
Устанавливает разукрупнение РЭС по конструктивной сложности







1, 2 — ячейки; 3 — корпус блока; 4 — блочный каркас; 5 — корпус вставного блока; 6 — корпус блока (авиационного); 7 — корпус контейнера (авиационного); 8 — корпус шкафа; 9 — корпус пульта; 10 — корпус моноблока; 11 — корпус шкафа для выдвижной стойки; 12 — выдвижная стойка



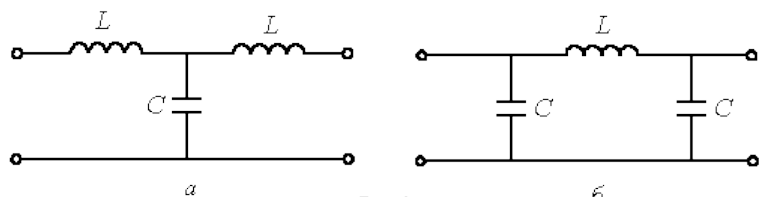
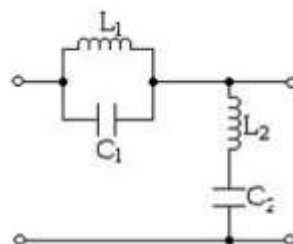
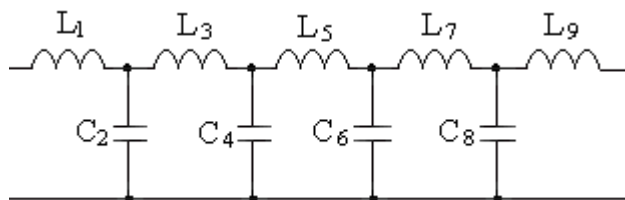
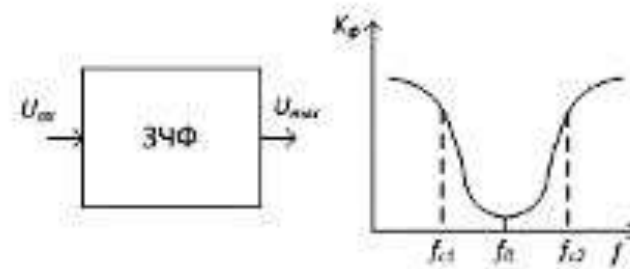
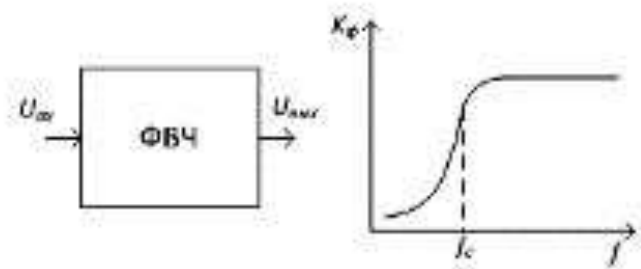
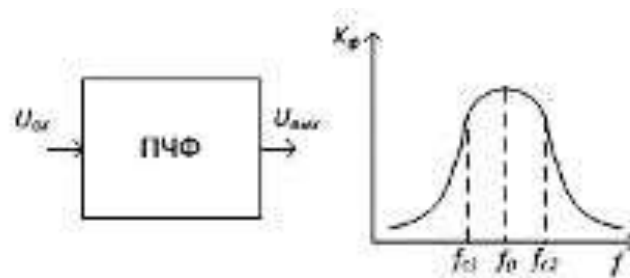
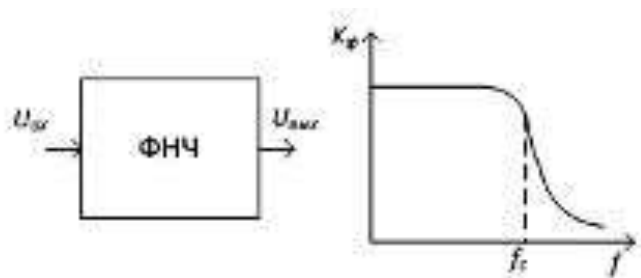
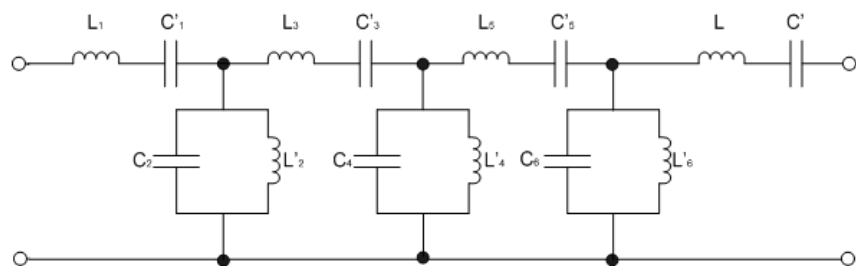
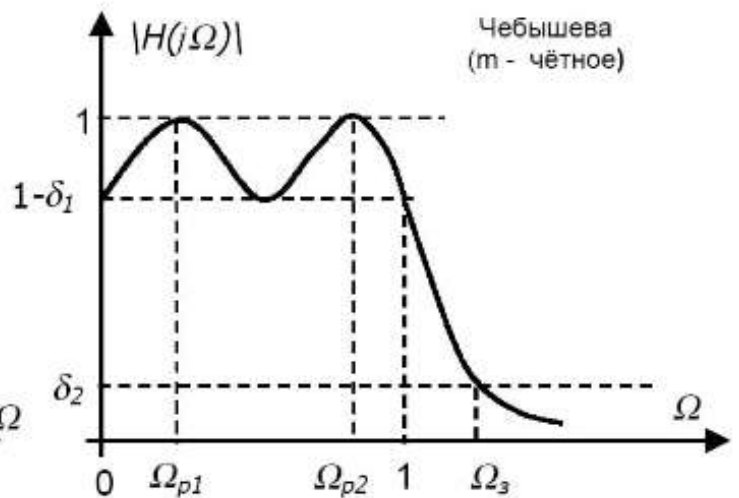
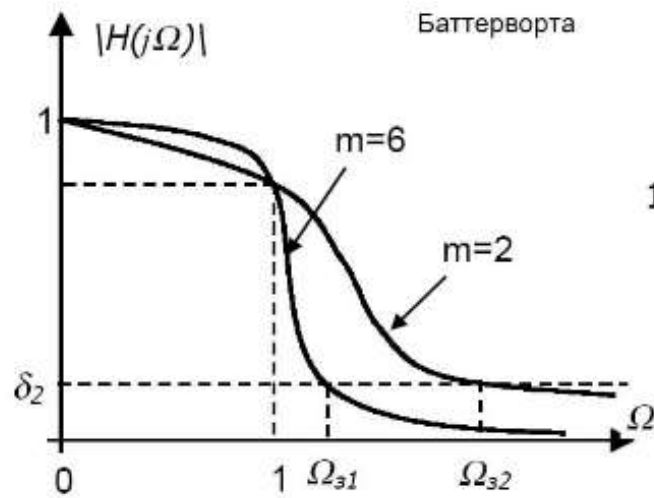
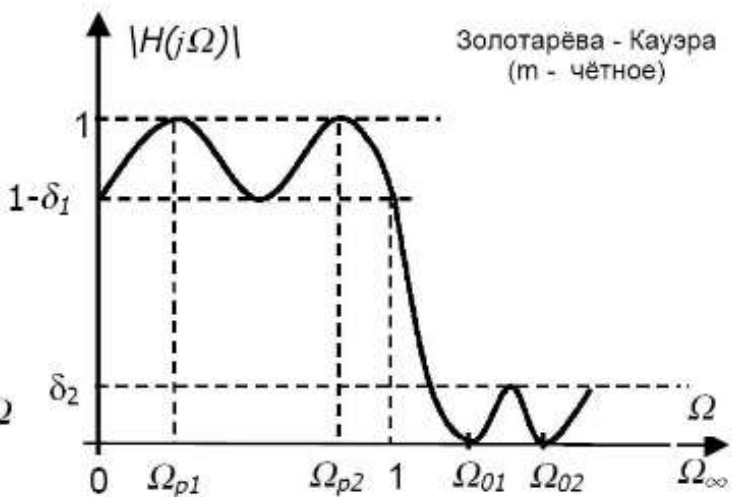
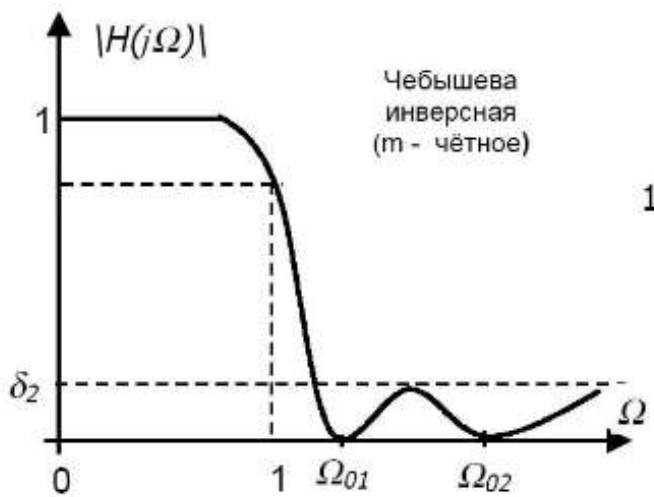
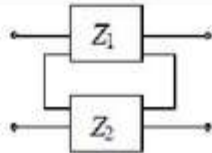
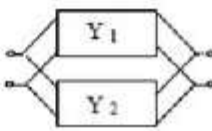
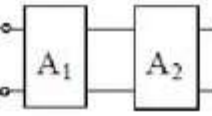
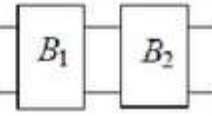
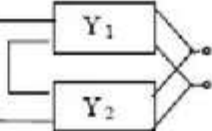
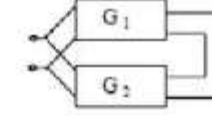


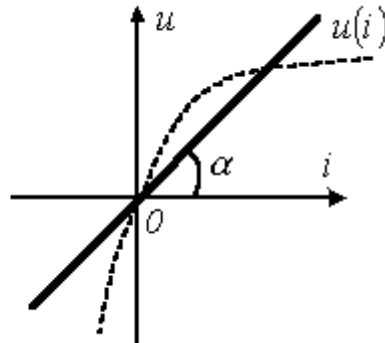
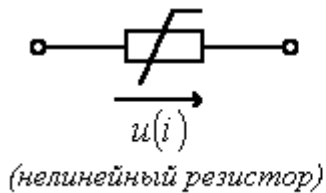
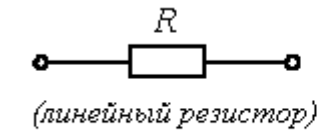
Рис.1



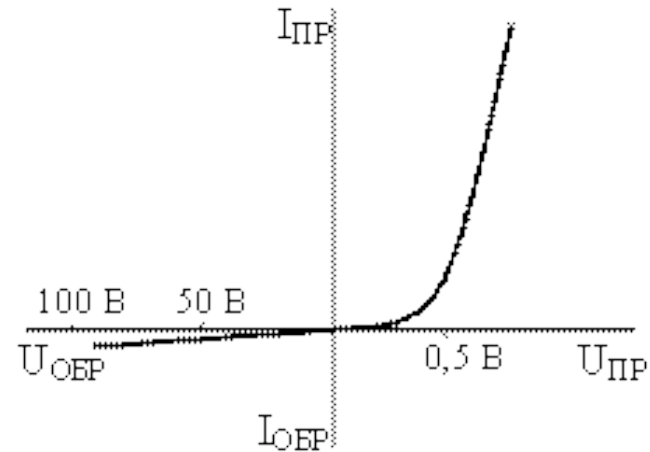
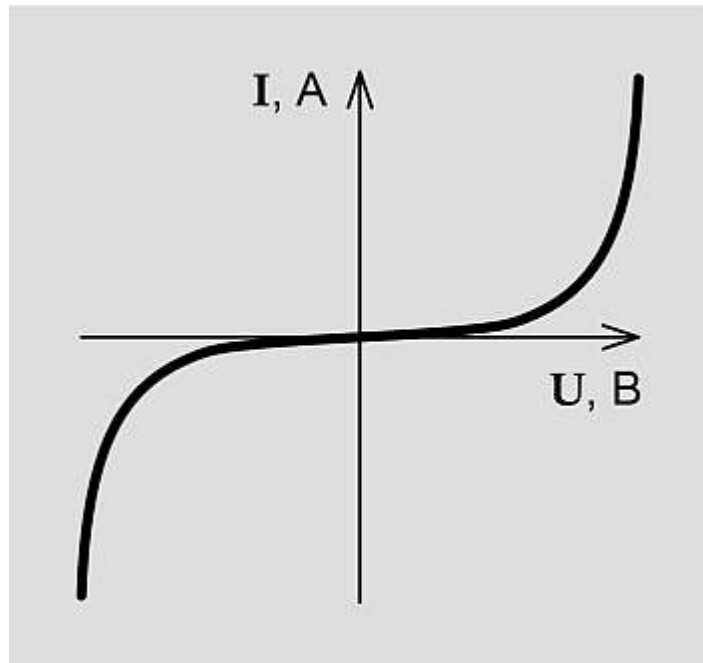


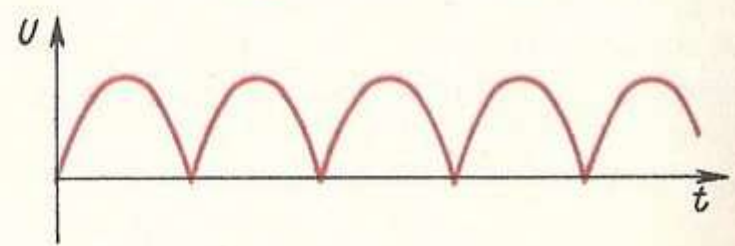
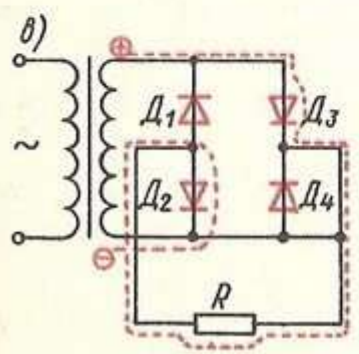
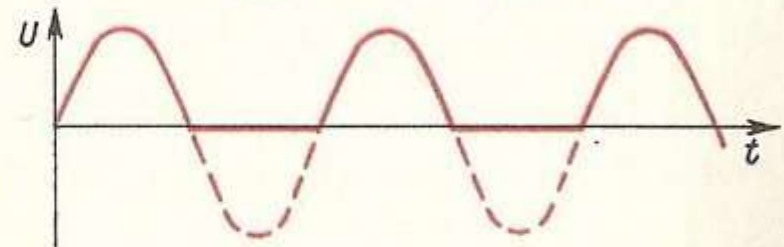
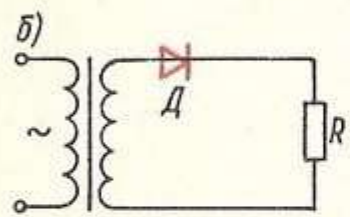
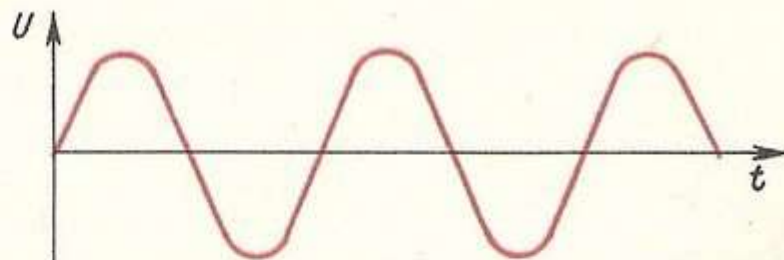
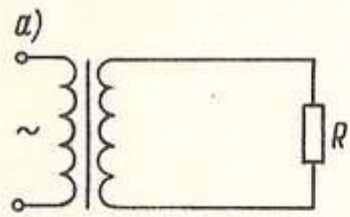
Соединения четырехполюсников

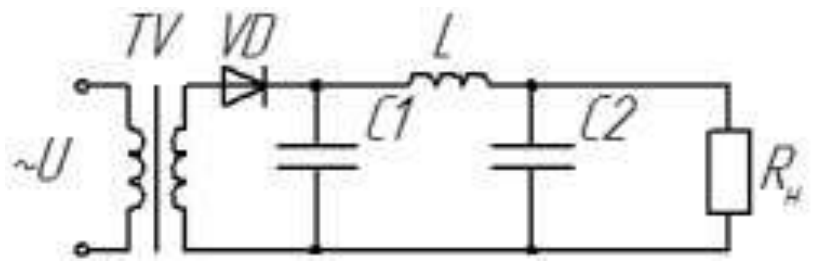
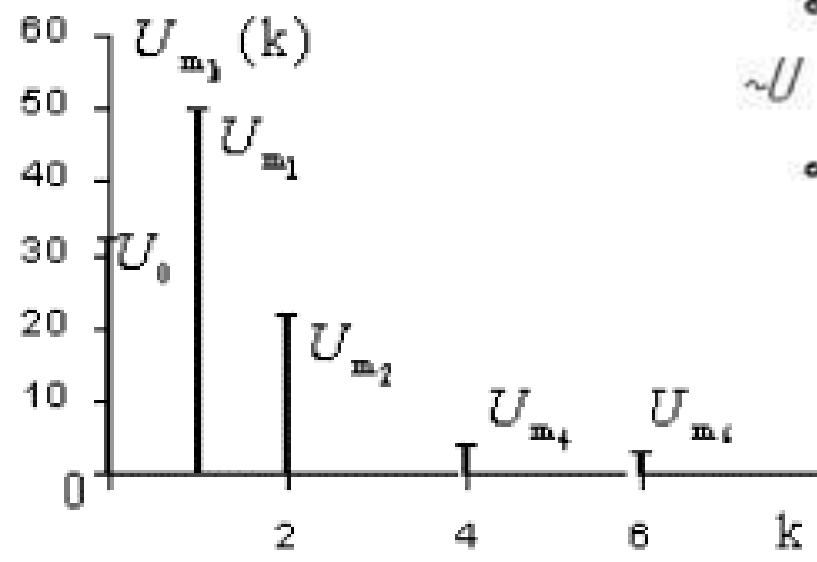
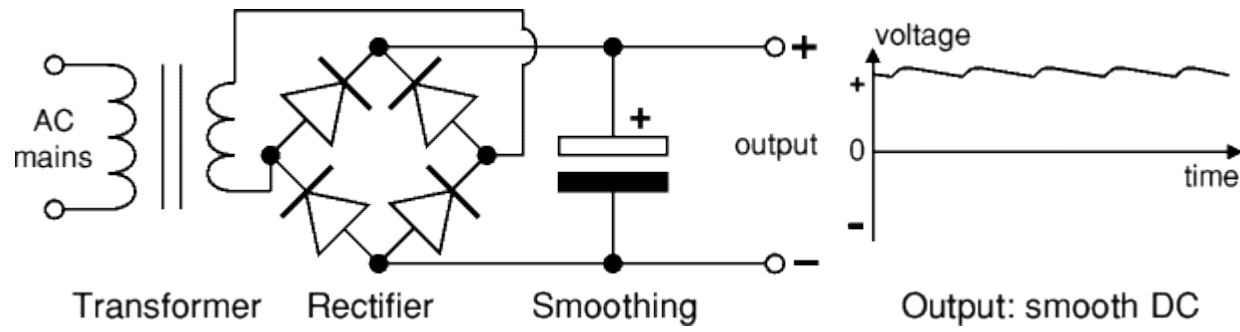
Вид параметра	Система уравнений четырехполюсника	Схема соединения четырехполюсника	Результирующее соотношение
1	2	3	4
Z-параметры	$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = [Z] \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$		$Z = Z_1 + Z_2$
Y-параметры	$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = [Y] \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$		$Y = Y_1 + Y_2$
A-параметры	$\begin{bmatrix} U_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} U_2 \\ I_2 \end{bmatrix} = [A] \begin{bmatrix} U_2 \\ I_2 \end{bmatrix}$		$A = A_1 \times A_2$
A ⁻¹ -параметры	$\begin{bmatrix} U_2 \\ I_2 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} U_1 \\ I_1 \end{bmatrix} = [B] \begin{bmatrix} U_1 \\ I_1 \end{bmatrix}$		$B = B_2 \times B_1$
1	2	3	4
H-параметры	$\begin{bmatrix} U_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ U_2 \end{bmatrix} = [H] \begin{bmatrix} I_1 \\ U_2 \end{bmatrix}$		$H = H_1 + H_2$
G-параметры	$\begin{bmatrix} I_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} U_1 \\ I_2 \end{bmatrix} = [G] \begin{bmatrix} U_1 \\ I_2 \end{bmatrix}$		$G = G_1 + G_2$

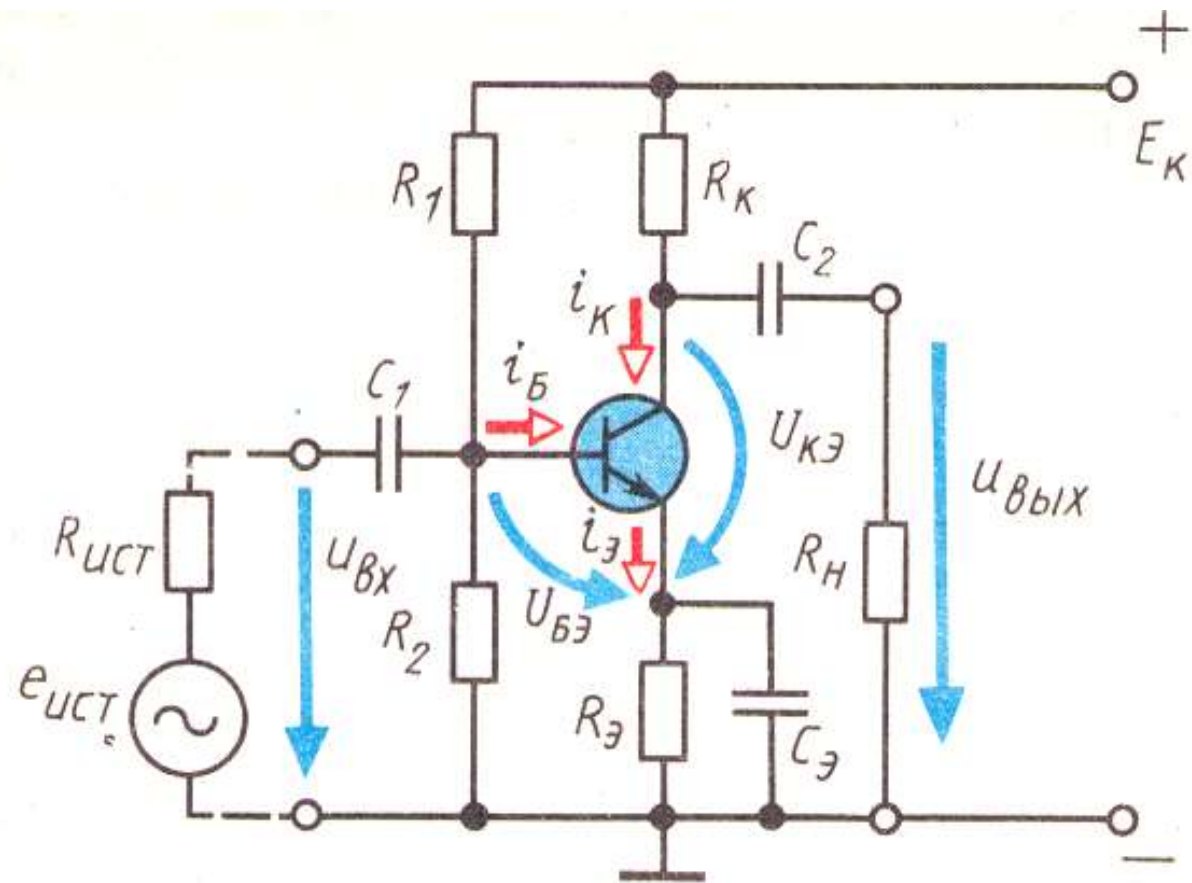


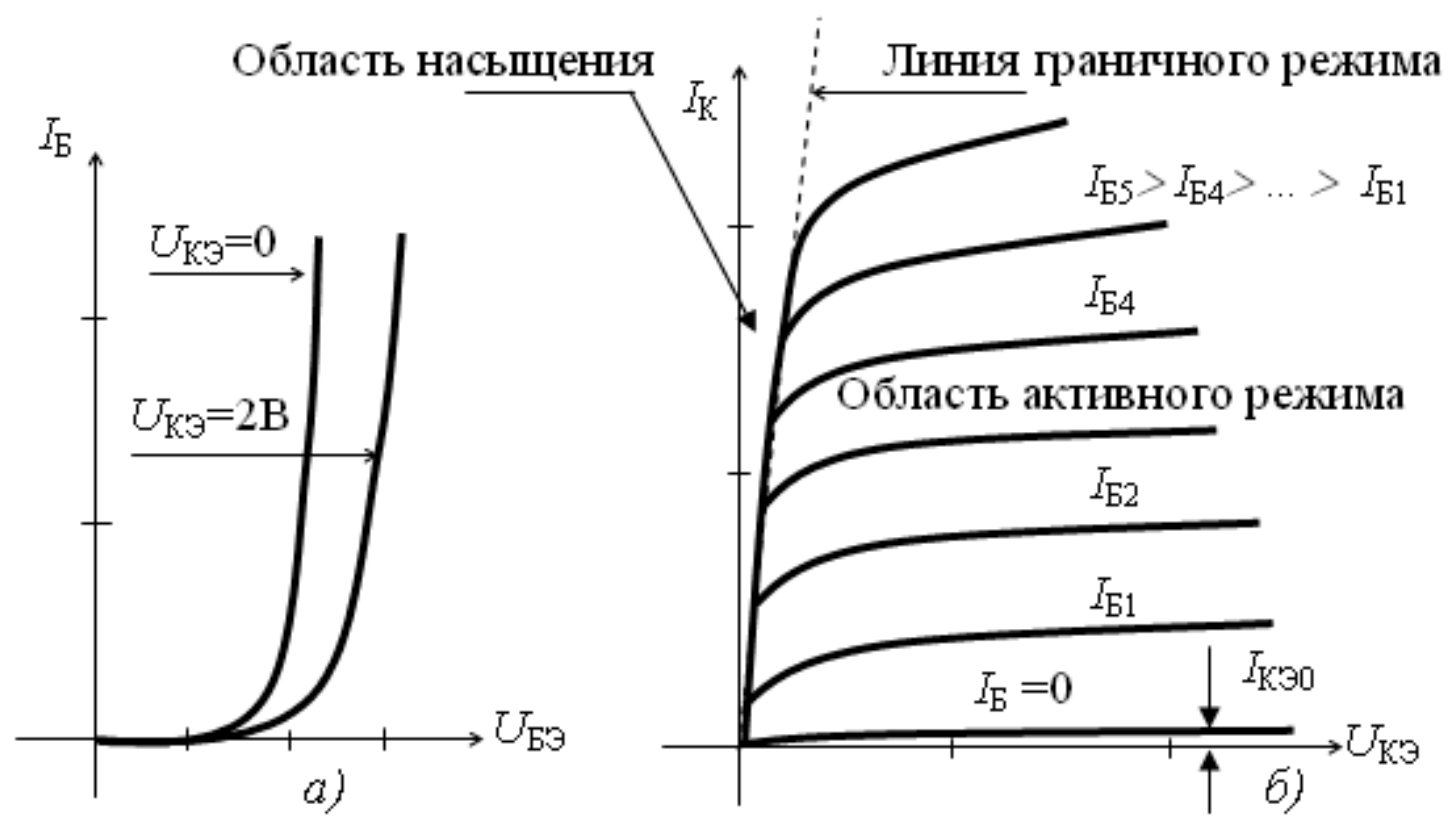
Для линейных электрических цепей справедлив **принцип наложения**, согласно которому ток (напряжение) любой ветви равен сумме частичных токов (напряжений), создаваемых в этой ветви каждым из источников в отдельности. Этот принцип лежит в основе **метода наложения**. Метод наложения применим только для расчета линейных цепей.

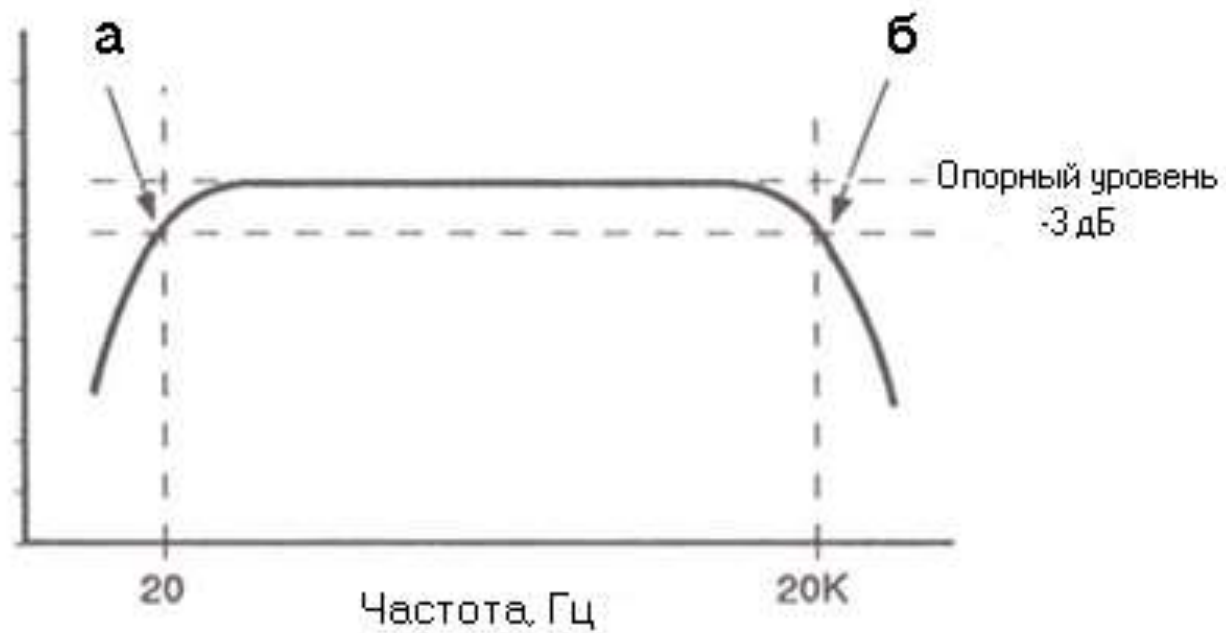


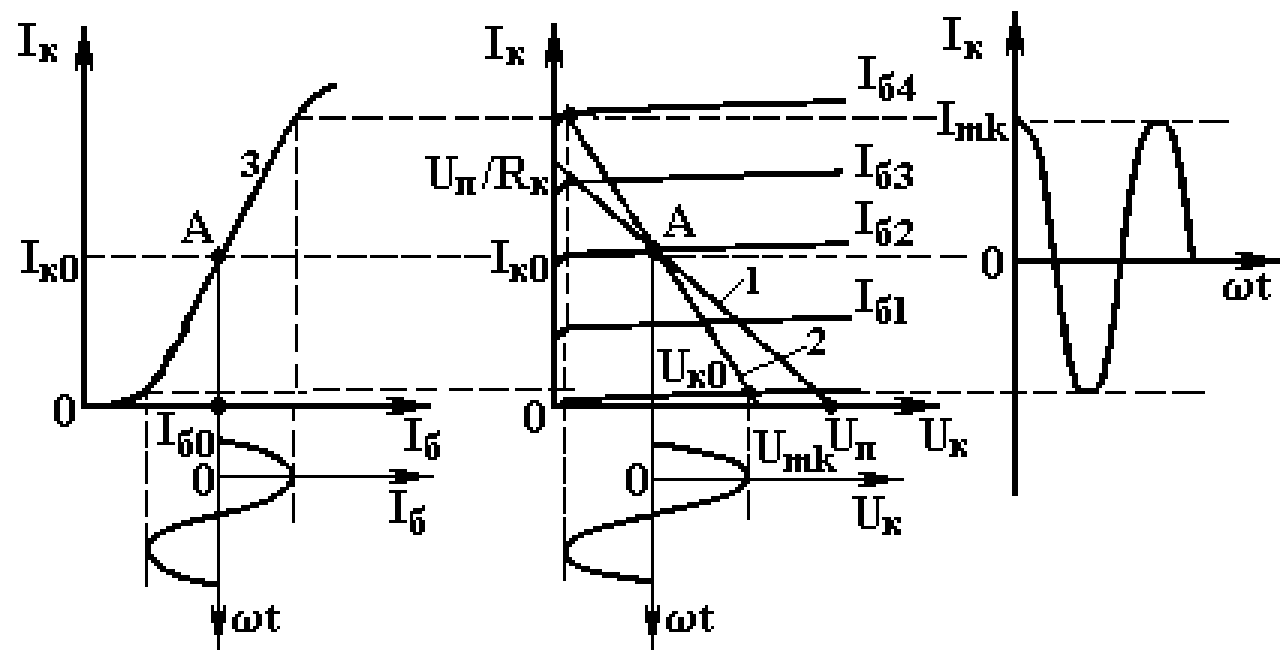


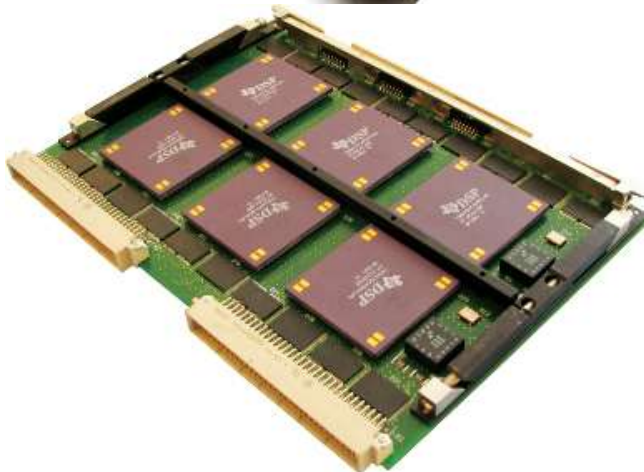






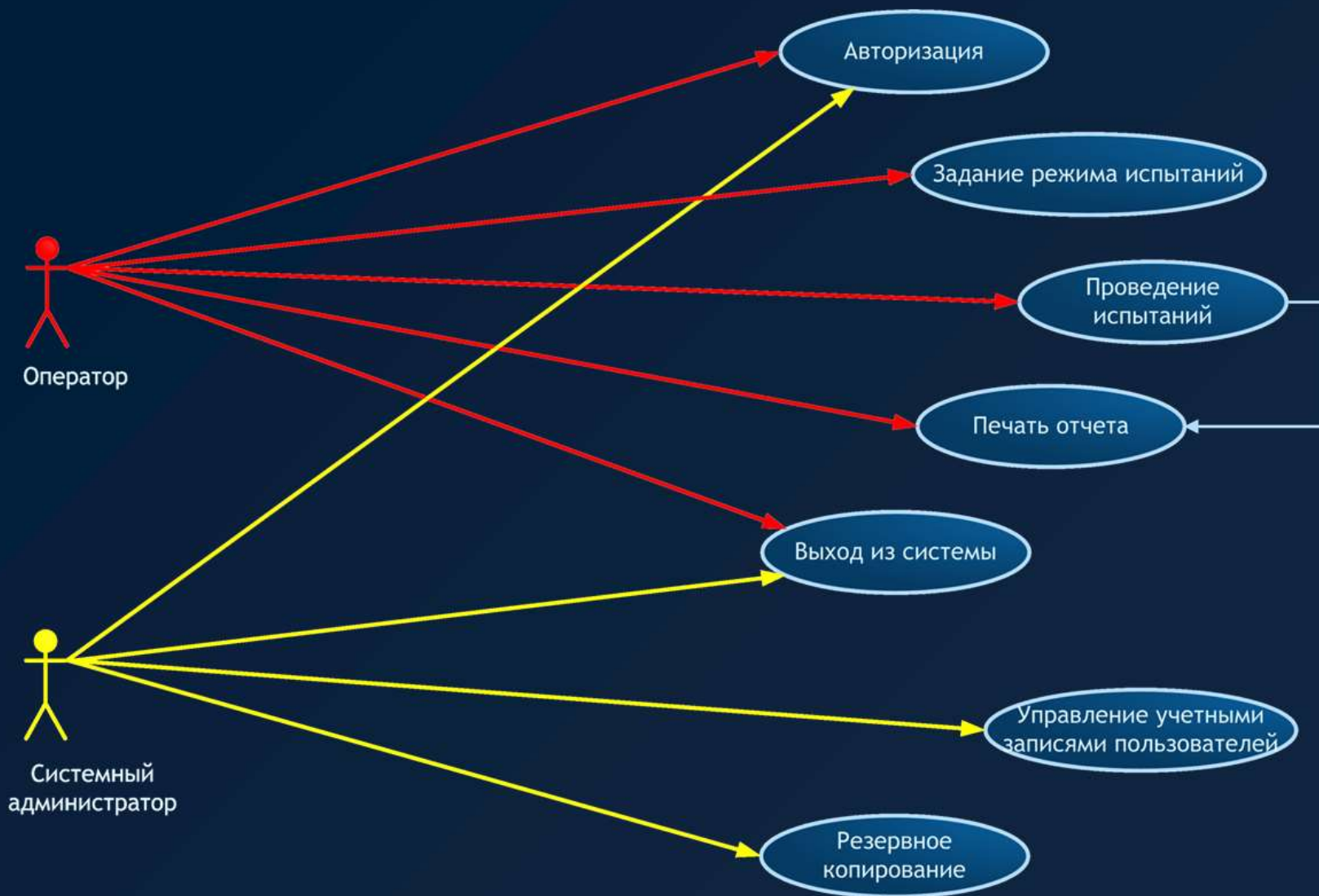


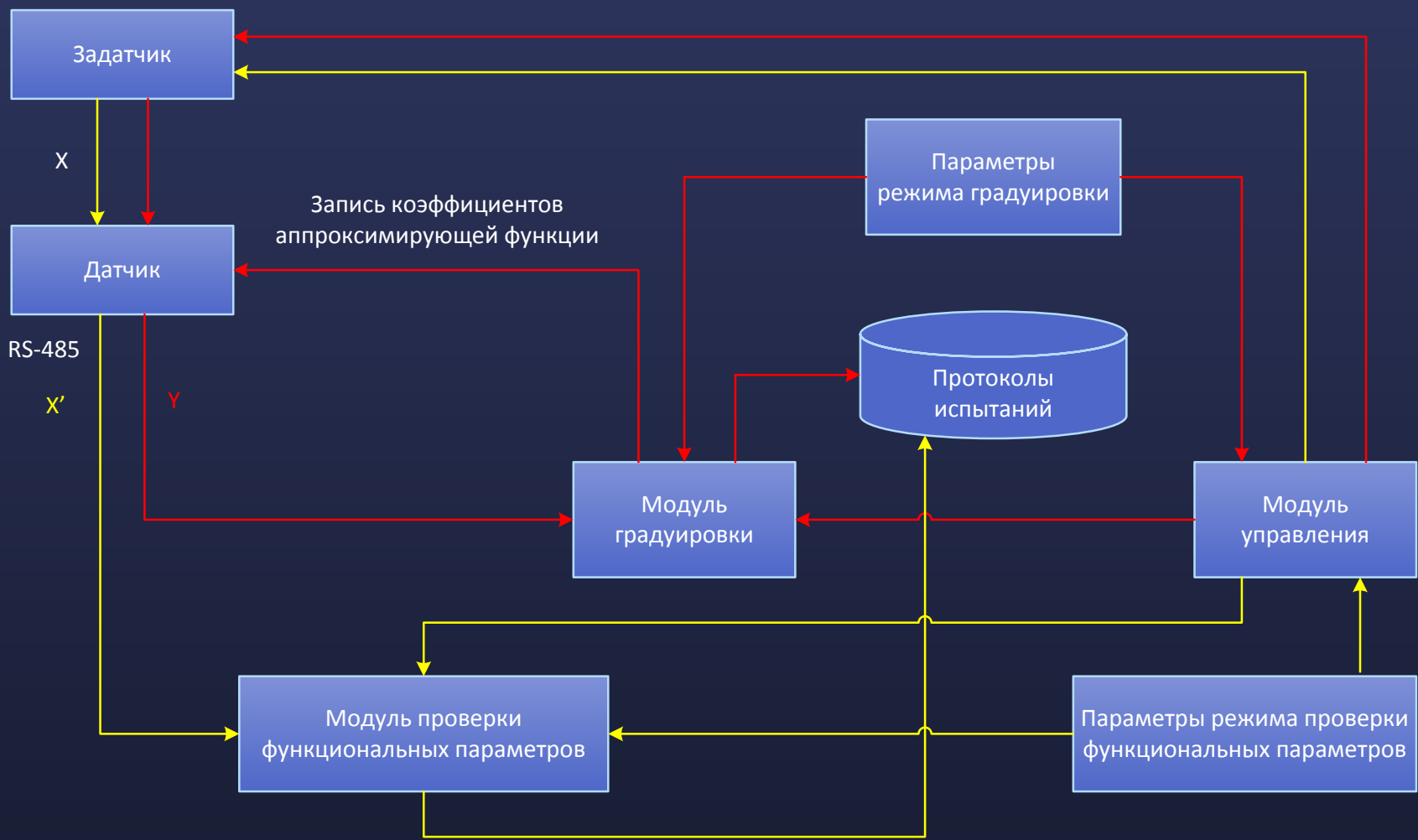






Варианты использования ПО АИО

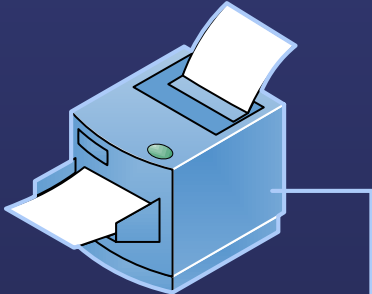




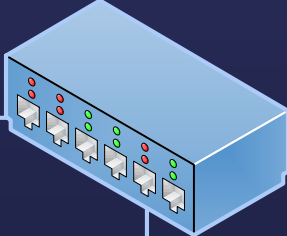
Управляющий компьютер

Задатчик

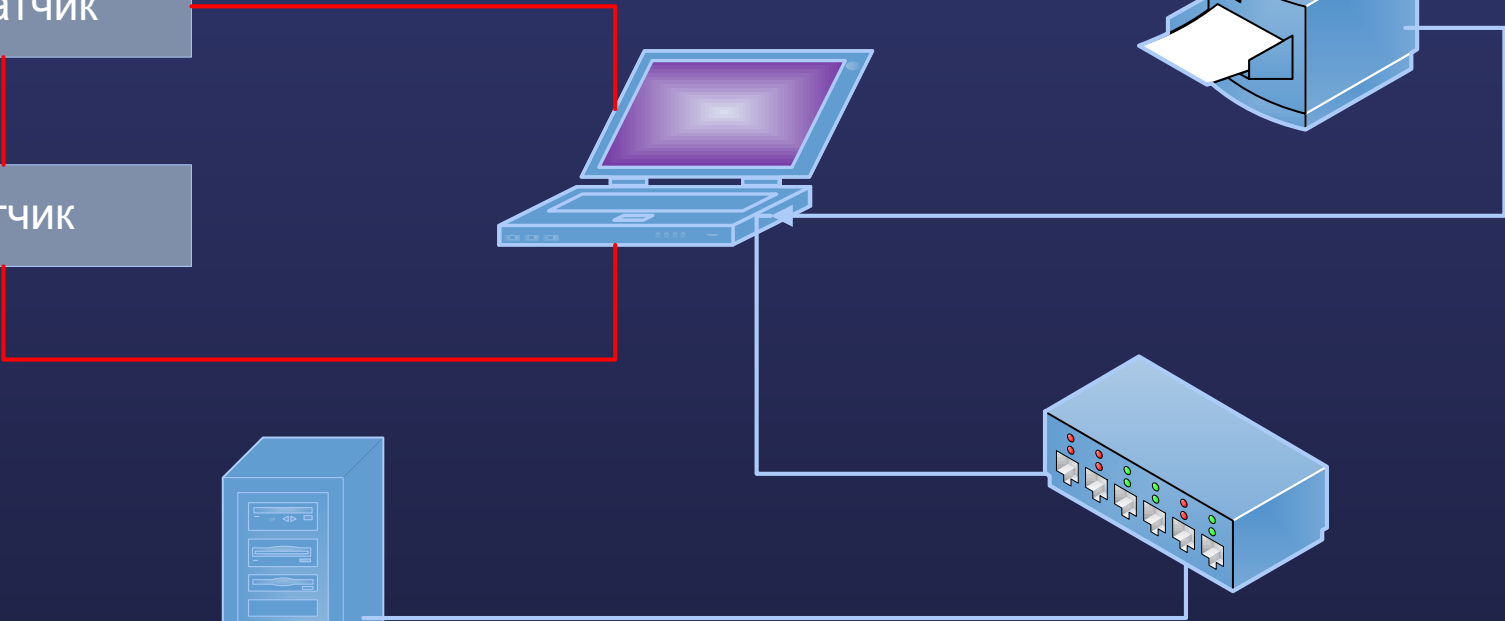
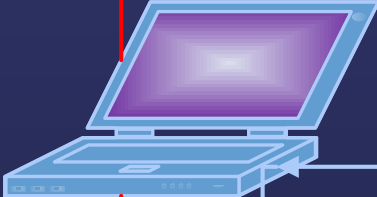
Датчик



Сервер



Маршрутизатор



Показатель	До внедрения	После внедрения	Эффективность
Число операторов	2	1	100%
Наличие ошибок	Возможны	Отсутствуют	Повышение точности калибровки
Время, необходимое для калибровки одной системы СКДС-СИШ (в состав системы входит 10 шайб)	3 смены	1,5 смены	100%

Возможность проведения испытаний с несколькими одноканальными или многоканальными датчиками

