



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 Прикладная информатика

О Т Ч Е Т

по домашнему заданию № 1

Название: Параллельное соединение автоматов

Дисциплина: Прикладная теория цифровых автоматов

Студент ИУ6-44Б  
(Группа)

Д. Ю. Колдин  
(Подпись, дата) (И.О. Фамилия)


Преподаватель

Ю.И. Бауман  
(Подпись, дата) (И.О. Фамилия)

Москва, 2022

## Содержание

ВВЕДЕНИЕ.....	3
1. СОСТАВЛЕНИЕ АВТОМАТОВ.....	4
1.1 Спецификация автомата.....	4
1.2 Первый автомат «Электрощиток».....	4
1.3 Второй автомат «Водопроводный кран».....	5
2. РЕАЛИЗАЦИЯ ЦИФРОВОГО АВТОМАТА ОБСЛУЖИВАНИЯ КВАРТИРЫ КОММУНАЛЬНЫМИ УСЛУГАМИ.....	8
3. ТЕСТИРОВАНИЕ АВТОМАТА .....	13
ЗАКЛЮЧЕНИЕ .....	16
СПИСОК ЛИТЕРАТУРЫ.....	17



## ВВЕДЕНИЕ

В данной работе было реализовано параллельное соединение цифровых автоматов.

Существуют два способа реализации автомата: аппаратный и программный. Существует два возможных способа реализации конечных автоматов: программный и аппаратный. Программная реализация производится на любом языке программирования высокого уровня. Аппаратная реализация требует построения устройств памяти для запоминания текущего состояния автомата, а также проводится с помощью логических функциональных элементов.

В данной работе был использован программный способ реализации цифрового автомата. Это было сделано, так как этот способ подразумевает вариативность реализации, возможность тестирования и отладки во время разработки программы. В отличие от аппаратной реализации цифровых автоматов, к программам можно добавлять новые функции по мере изменения целей, под которые она разрабатывается.

Задание варианта №1: Последовательное соединение автоматов.

Цель работы - закрепить навыки реализации конечных цифровых автоматов.

Для выполнения поставленной цели необходимо выполнить следующие задачи.

Задачи:

- Изучить задание своего варианта;
- Описать автомат, соответствующий условию задачи;
- Изучить способы реализации цифровых автоматов;
- Выбрать один из способов реализации автоматов;
- Реализовать описанный цифровой автомат;
- Оформить отчёт.



# 1. СОСТАВЛЕНИЕ АВТОМАТОВ

## 1.1 Спецификация автомата

Создадим автомат работы обслуживания квартиры коммунальными услугами. В данной работе рассматривались услуги по электроснабжению и водопроводу. Данный автомат состоит из двух, работающих независимо и подключенных параллельно, автоматов: электрощиток квартиры и водопроводный кран.

Составим спецификацию для данных автоматов:

## 1.2 Первый автомат «Электрощиток»

Состояния автомата:

- $q_0$  – электричество выключено;
- $q_1$  – электричество включено;
- $q_2$  – аварийное состояние.

Входные сигналы:

- $a$  – переключение тумблера;
- $b$  – скачок напряжения;
- $c$  – оставить текущее состояние.

Выходные сигналы:

- $0$  – отсутствие электричества в квартире;
- $1$  – наличие электричества в квартире;

Ниже представлена таблица, которая описывает конечный автомат, составленный по условию задачи:

Таблица 1 – таблица переходов автомата «Электрощиток»

Состояние	$\delta$			$\lambda$		
	$a$	$b$	$c$	$a$	$b$	$c$
$q_0$	$q_1$	-	$q_0$	1	-	0
$q_1$	$q_0$	$q_2$	$q_1$	0	0	1
$q_2$	$q_2$	$q_2$	$q_2$	0	0	0

На рисунке 1 представлен описанный автомат в виде графа.

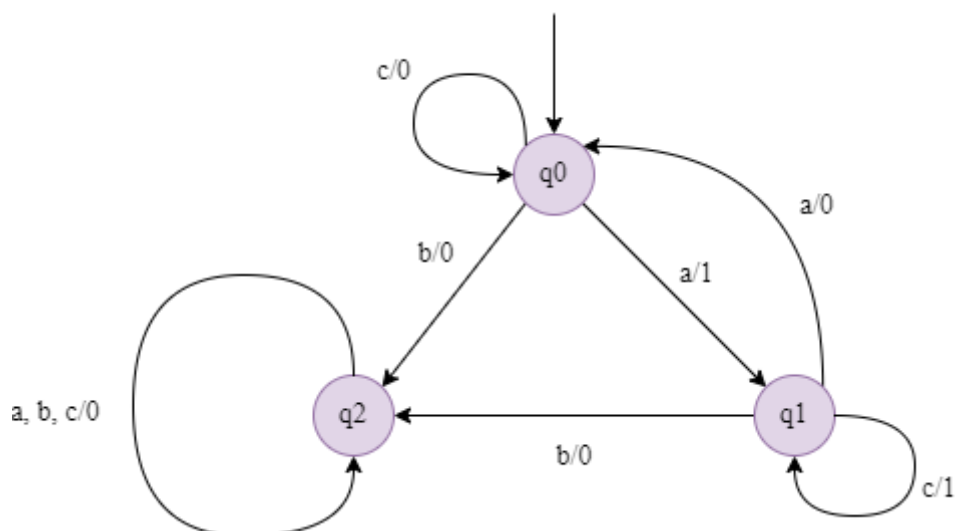


Рисунок 1 – Граф переходов цифрового автомата

### 1.3 Второй автомат «Водопроводный кран»

*Состояния автомата:*

- s0 – кран закрыт;
- s1 – кран открыт.

*Входные сигналы:*

- a – поворот крана;
- b – оставить текущее состояние.

*Выходные сигналы:*

- 0 – вода не течёт;
- 1 – вода течет.

Таблица 2 – Таблица переходов и выходов автомата «Водопроводный кран»

Состояние	$\delta$		$\lambda$	
	a	b	a	b
s0	s1	s0	1	0
s1	s0	s1	0	1

На рисунке 2 представлен описанный автомат в виде графа.

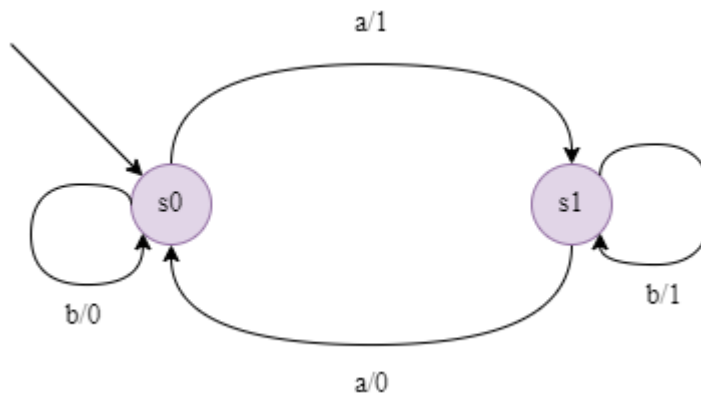


Рисунок 2 – Граф переходов цифрового автомата

Параллельное соединение автомата также является автоматом, входным алфавитом которого является прямое произведение входного алфавита полуавтоматов, которые включены в него, аналогично определяются выходные алфавиты и возможные состояния автомата. Ниже представлена схема соединения описанных автоматов (рисунок 3). Используемые обозначения:  $Q$ ,  $A_1$ ,  $Y_1$  – множество состояний, входной и выходной алфавиты полуавтомата «Электрощиток» соответственно.  $S$ ,  $A_2$ ,  $Y_2$  – множество состояний, входной и выходной алфавиты полуавтомата «Водопроводный кран» соответственно.

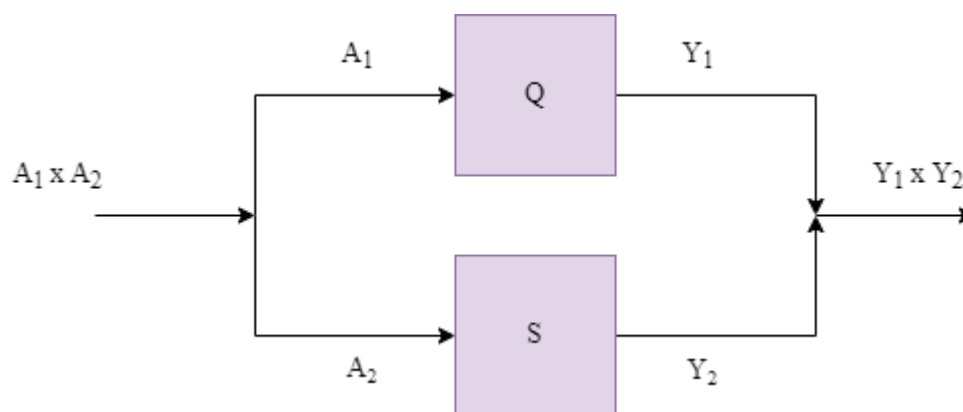


Рисунок 3 – Схема соединения реализованных полуавтоматов

Граф переходов автомата, получающегося в результате параллельного соединения двух полуавтоматов, представлен на рисунке 4:

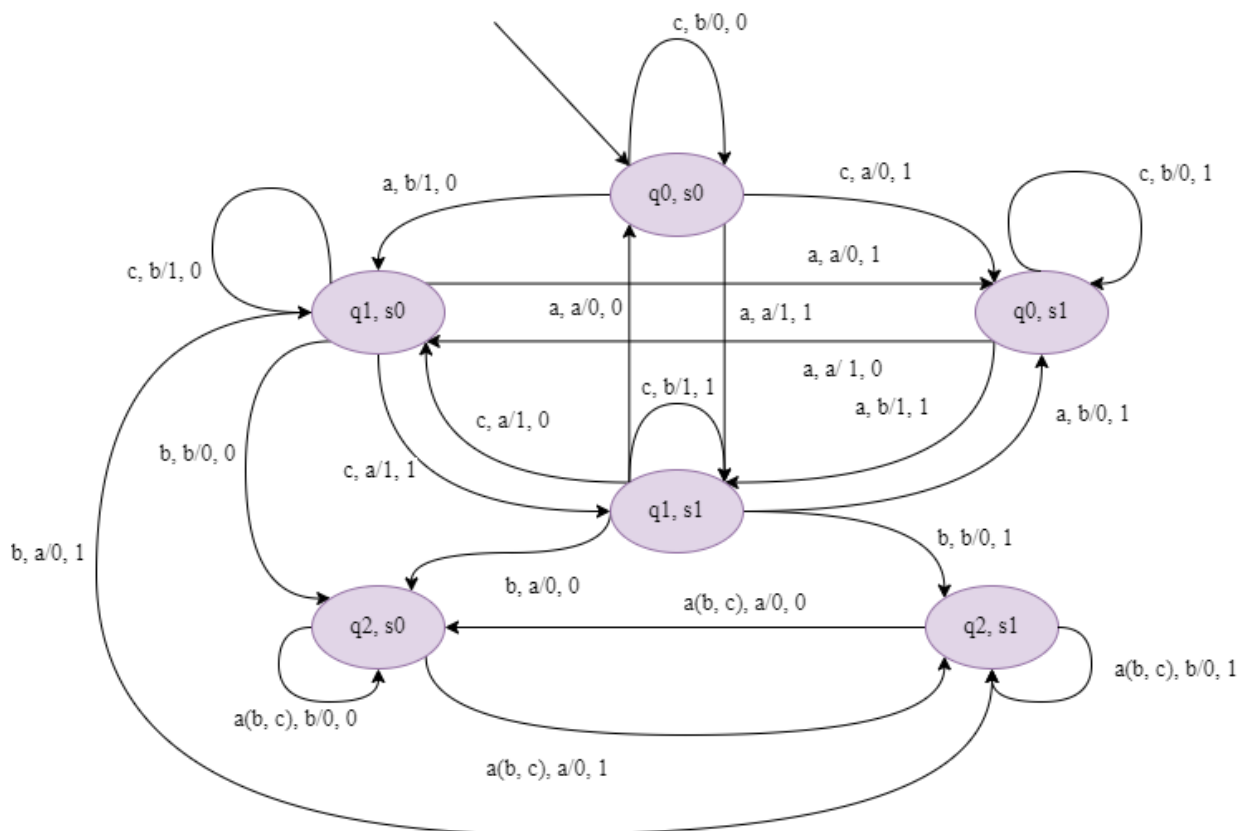


Рисунок 4 – Граф переходов результирующего автомата

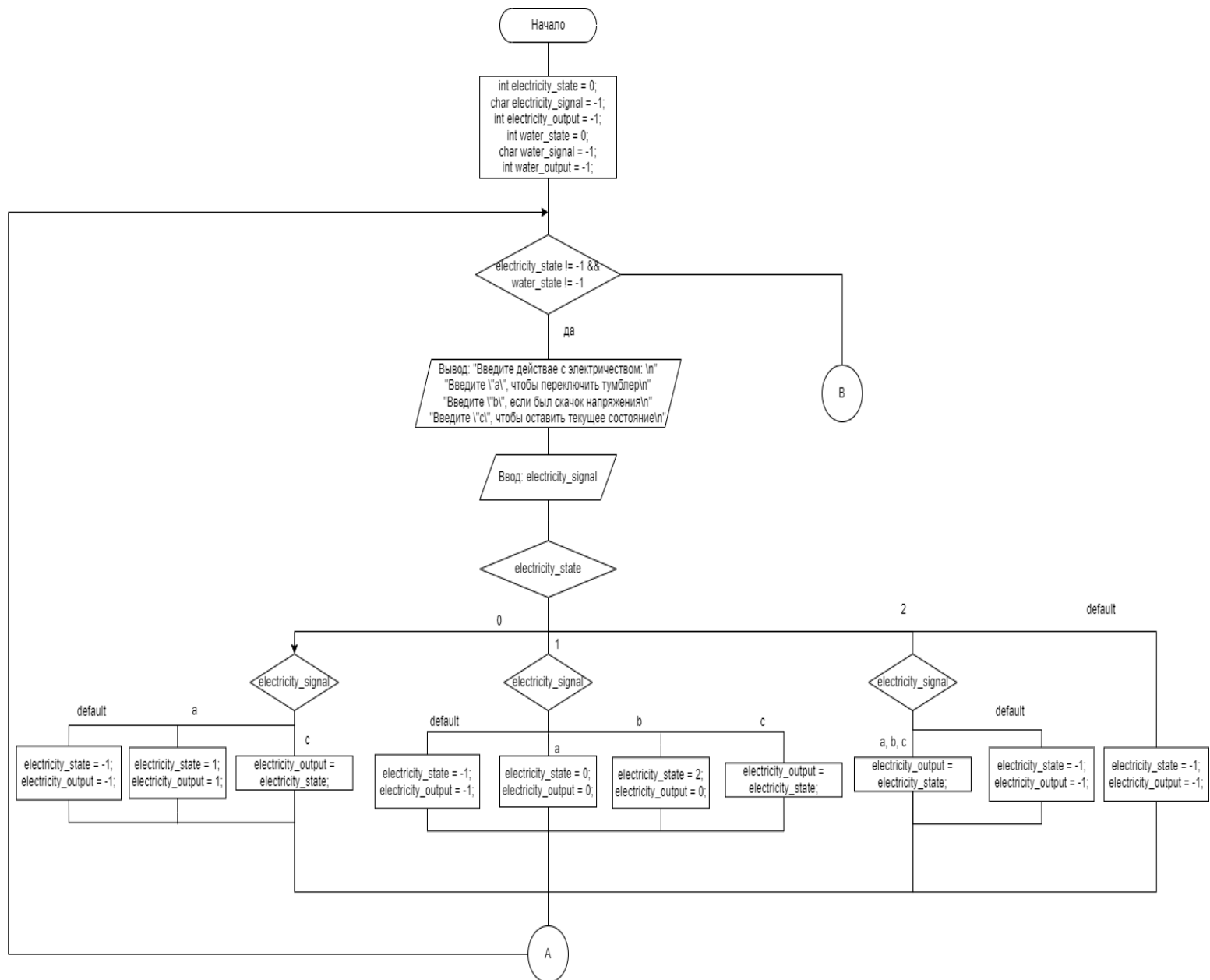
Возможные переходы и выходные сигналы автомата представлены в таблице 3:

Таблица 3 – Таблица переходов и выходов автомата

Состояния	$\delta$						$\lambda$					
	a, a	a, b	b, a	b, b	c, a	c, b	a, a	a, b	b, a	b, b	c, a	c, b
q0, s0	q1, s1	q1, s0	-	-	q0, s1	q0, s0	1, 1	1, 0	-	-	0, 1	0, 0
q1, s0	q0, s1	q0, s0	q2, s1	q2, s0	q1, s1	q1, s0	0, 1	0, 0	0, 1	0, 0	1, 1	1, 0
q2, s0	q2, s1	q2, s0	q2, s1	q2, s0	q2, s1	q2, s0	0, 1	0, 0	0, 1	0, 0	0, 1	0, 0
q0, s1	q1, s0	q1, s1	-	-	q0, s0	q0, s1	1, 0	1, 1	-	-	0, 0	0, 1
q1, s1	q0, s0	q0, s1	q2, s0	q2, s1	q1, s0	q1, s1	0, 0	0, 1	0, 0	0, 1	1, 0	1, 1
q2, s1	q2, s0	q2, s1	q2, s0	q2, s1	q2, s0	q2, s1	0, 0	0, 1	0, 0	0, 1	0, 0	0, 1

## 2. РЕАЛИЗАЦИЯ ЦИФРОВОГО АВТОМАТА ОБСЛУЖИВАНИЯ КВАРТИРЫ КОММУНАЛЬНЫМИ УСЛУГАМИ

Для реализации описанных цифровых автоматов ~~была~~ разработана схема алгоритма (рисунок 5) и написана программа на языке C++(листинг 1).





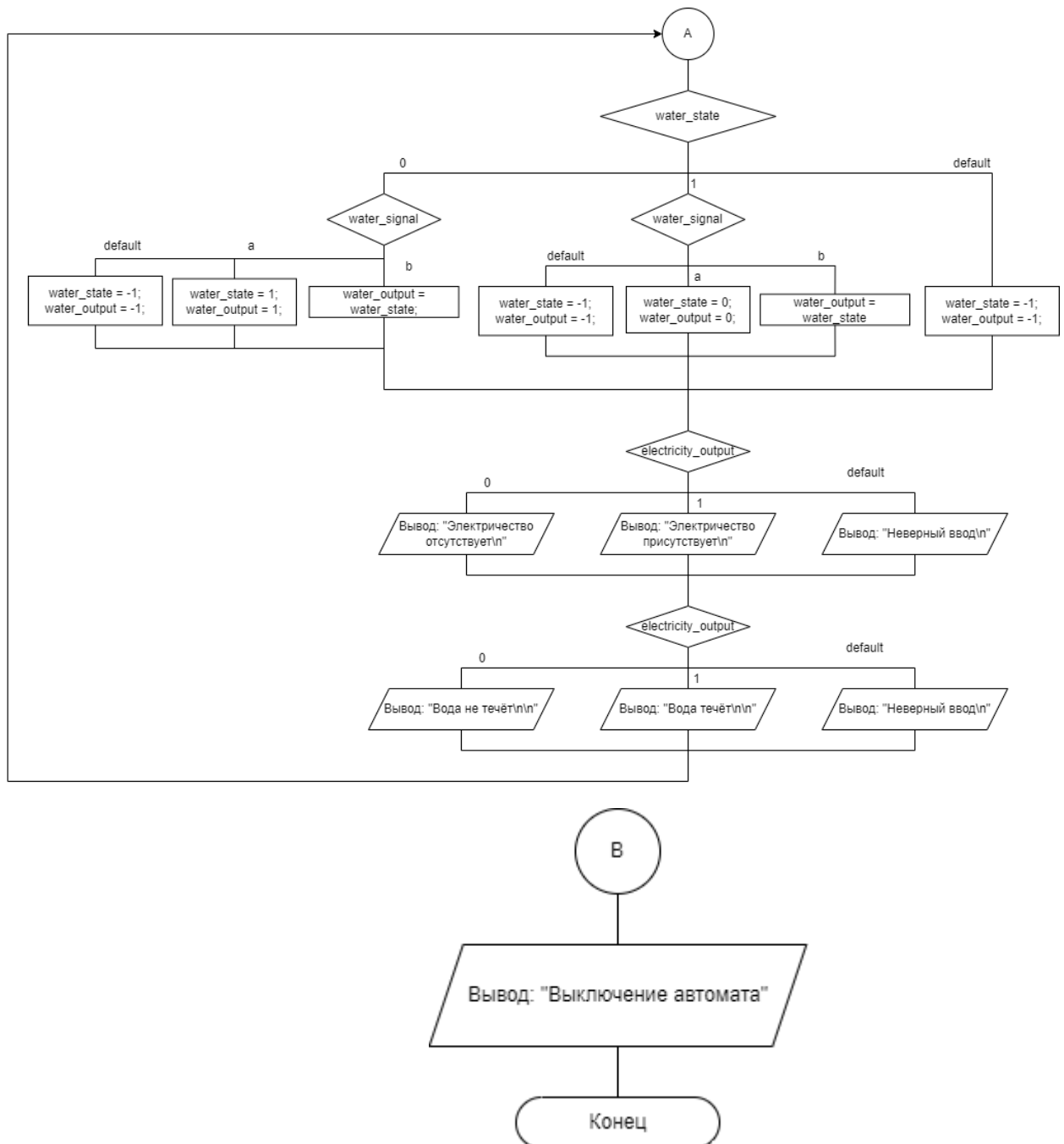


Рисунок 5 – Схема алгоритма

Листинг 1 – Написанная программная реализация:

```

#include <iostream>
using namespace std;

int main() {
    setlocale(LC_ALL, "ru");
    int electricity_state = 0;
    char electricity_signal = -1;
    int electricity_output = -1;
    int water_state = 0;
    char water_signal = -1;
    int water_output = -1;

    while (electricity_state != -1 && water_state != -1) {
        cout << "Введите действие с электричеством: \n";
    }
  
```

```

cout << "Введите \"a\", чтобы переключить тумблер\n";
cout << "Введите \"b\", если был скачок напряжения\n";
cout << "Введите \"c\", чтобы оставить текущее состояние\n";
cin >> electricity_signal;

switch (electricity_state)
{
case 0:
{
    switch (electricity_signal)
    {
    case 'a':
    {
        electricity_state = 1;
        electricity_output = 1;
        break;
    }
    case 'c':
    {
        electricity_output = electricity_state;
        break;
    }
    default: {
        electricity_state = -1;
        electricity_output = -1;
        break;
    }
    }
    break;
}
case 1:
{
    switch (electricity_signal)
    {
    case 'a':
    {
        electricity_state = 0;
        electricity_output = 0;
        break;
    }
    case 'b':
    {
        electricity_state = 2;
        electricity_output = 0;
        break;
    }
    case 'c':
    {
        electricity_output = electricity_state;
        break;
    }
    default: {
        electricity_state = -1;
        electricity_output = -1;
        break;
    }
    }
    break;
}
case 2:
{
    switch (electricity_signal)
    {
    case 'a':

```

```

        case 'b':
        case 'c':
        {
            electricity_output = 0;
            break;
        }
        default: {
            electricity_state = -1;
            electricity_output = -1;
            break;
        }
    }
    break;
}
default: {
    electricity_state = -1;
    electricity_output = -1;
    break;
}
}

// Water processing.....

cout << "Введите действие с водопроводным краном: \n";
cout << "Введите \"a\", чтобы повернуть кран\n";
cout << "Введите \"b\", чтобы оставить текущее состояние\n";
cin >> water_signal;

switch (water_state)
{
case 0:
{
    switch (water_signal)
    {
        case 'a':
        {
            water_state = 1;
            water_output = 1;
            break;
        }
        case 'b':
        {
            water_output = water_state;
            break;
        }
        default: {
            water_state = -1;
            water_output = -1;
            break;
        }
    }
    break;
}
case 1:
{
    switch (water_signal)
    {
        case 'a':
        {
            water_state = 0;
            water_output = 0;
            break;
        }
        case 'b':

```

```

        {
            water_output = water_state;
            break;
        }
        default: {
            water_state = -1;
            water_output = -1;
            break;
        }
    }
    break;
}
default: {
    electricity_state = -1;
    electricity_output = -1;
    break;
}
}

// OUTPUT PROCESSING.....
switch (electricity_output)
{
case 0:
{
    cout << "Электричество отсутствует\n";
    break;
}
case 1: {
    cout << "Электричество присутствует\n";
    break;
}
default: {
    cout << "Неверный ввод\n";
    break;
}
}

switch (water_output)
{
case 0:
{
    cout << "Вода не течёт\n\n";
    break;
}
case 1: {
    cout << "Вода течёт\n\n";
    break;
}
default: {
    cout << "Неверный ввод\n\n";
    break;
}
}
}
return 0;
}

```

### 3/ ТЕСТИРОВАНИЕ АВТОМАТА

Для анализа правильности функционирования созданной реализации конечного автомата необходимо проверить верность переходов его состояний и выходных сигналов. Т.к. реализация автомата программная, то тестировать работу автомата будем, основываясь на значениях переменных `electricity_state` и `water_state`, определяющих текущее состояние автомата, а также результата вывода сообщений на экран, определяющих выходящий сигнал.

*Проверка сигнала (a, a) из состояния (q0, s0).* Программа выведет сообщение о включении электричества и воды, состояния изменятся на q1, s1. Данный тест представлен на рисунке 6.

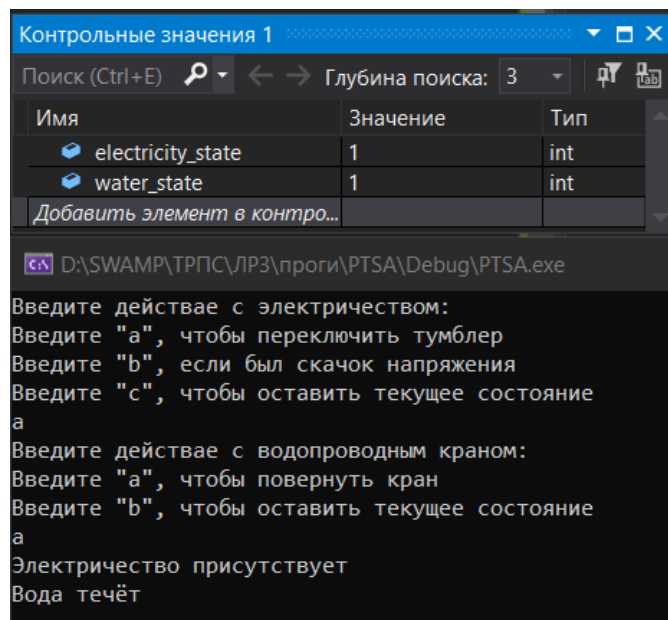


Рисунок 6 – Тест №1

*Проверка сигнала (a, a) из состояния (q1, s1).* Программа выведет сообщение о выключении электричества и воды, состояния изменятся на q0, s0. Данный тест представлен на рисунке 7.

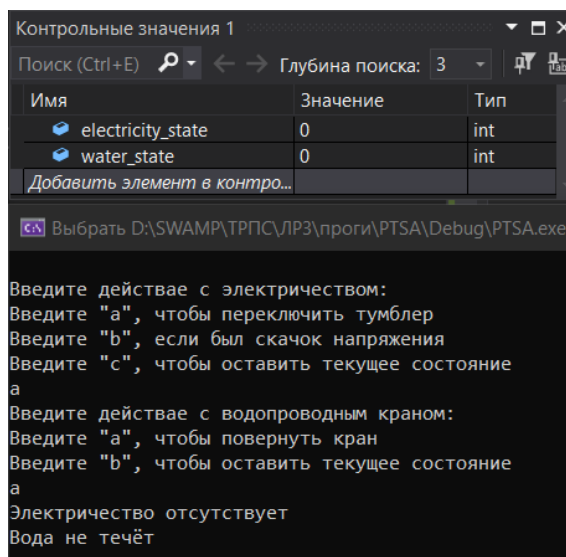


Рисунок 7 – Тест №2

*Проверка сигнала (с, b) из состояния (q0, s0).* Программа выведет сообщение о выключении электричества и воды, состояния не изменятся. Данный тест представлен на рисунке 8.

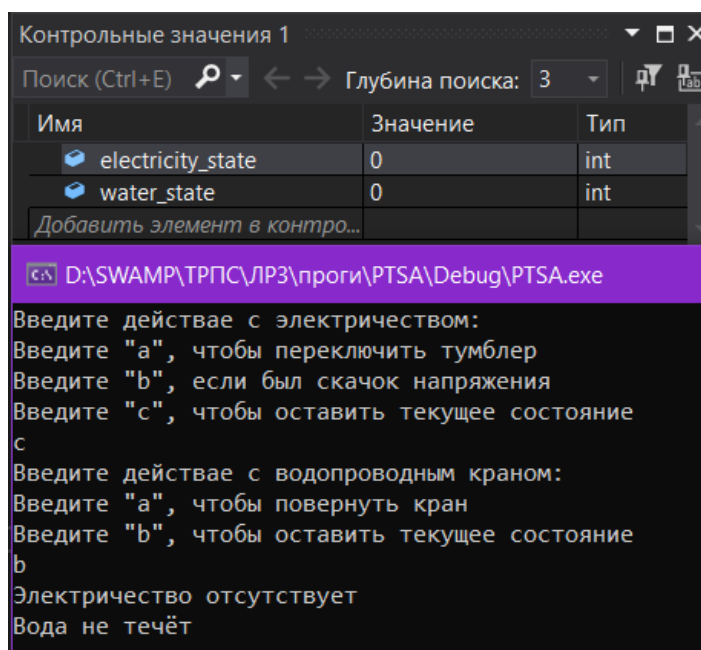


Рисунок 8 – Тест №3

*Проверка сигнала (b, b) из состояния (q1, s1).* Программа выведет сообщение о выключении электричества и включении воды, состояния изменятся на q2, s1. Данный тест представлен на рисунке 9.

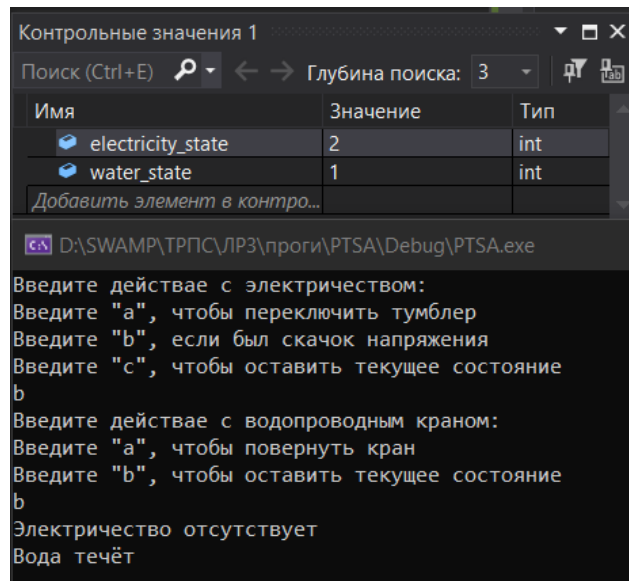


Рисунок 9 – Тест №4

## **ЗАКЛЮЧЕНИЕ**

- 1) При выполнении домашнего задания были изучены способы соединения автоматов.
- 2) Были получены практические навыки в представлении автоматов различными способами.
- 3) Закреплены навыки программной реализации конечных автоматов на примере реализации автомата, включающего в себя два полуавтомата, соединенных параллельно.
- 4) Программно реализован конечный автомат в среде разработки Visual Studio 2019 на языке C++ стандарта 17.
- 5) Закреплены навыки оформления отчета по проделанной работе с учетом требований ГОСТ 7.32-2017.



## СПИСОК ЛИТЕРАТУРЫ

- 1) Теория автоматов : метод. указания к выполнению курсовых работ / сост. : А. А. Городов, А. Л. Мыльников, А. М. Попов / Сиб. гос. аэрокосмич. ун-т. – Красноярск, 2014. – 52 с
- 2) Теория автоматов / Ю. Г. Карпов – СПб.: Питер, 2003. – с.: ил.
- 3) 1. Лекционные занятия МГТУ им. Н.Э. Баумана для 2-ого курса факультета ИУ6 с ведущим преподавателем Губарем А.М.
- 4) 2. Семинарские занятия МГТУ им. Н.Э. Баумана для 2-ого курса факультета ИУ6 с ведущим преподавателем Бауманом Ю.И.