

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Транспортные и технологические машины»

# СИСТЕМЫ УПРАВЛЕНИЯ МАШИНАМИ

*Методические рекомендации к лабораторным работам  
для студентов специальности*

*1-36 11 01 «Инновационная техника для строительного комплекса  
(по направлениям)» очной формы обучения*



Могилев 2023

УДК 004.42:629  
ББК 32.973.2:39  
С40

Рекомендовано к изданию  
учебно-методическим отделом  
Белорусско-Российского университета

Одобрено кафедрой «Транспортные и технологические машины»  
«27» сентября 2023 г., протокол № 2

Составитель канд. техн. наук, доц. В. В. Береснев

Рецензент канд. техн. наук, доц. А. П. Прудников

Методические рекомендации к лабораторным работам предназначены для студентов специальности 1-36 11 01 «Инновационная техника для строительного комплекса (по направлениям)» очной формы обучения.

Учебное издание

## СИСТЕМЫ УПРАВЛЕНИЯ МАШИНАМИ

Ответственный за выпуск	И. В. Лесковец
Корректор	И. В. Голубцова
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.  
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 26 экз. Заказ №

Издатель и полиграфическое исполнение:  
Межгосударственное образовательное учреждение высшего образования  
«Белорусско-Российский университет».  
Свидетельство о государственной регистрации издателя,  
изготовителя, распространителя печатных изданий  
№ 1/156 от 07.03.2019.  
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский  
университет, 2023

## Содержание

Введение.....	4
1 Лабораторная работа № 1. Общие сведения о программном пакете FluidSIM.....	5
2 Лабораторная работа № 2. Методика проектирования и моделирования гидравлических схем.....	10
3 Лабораторная работа № 3. Методика проектирования и моделирования электрогидравлических схем.....	13
4 Лабораторная работа № 4. Логические элементы.....	18
5 Лабораторная работа № 5. Составление логических схем на основе таблиц истинности.....	24
6 Лабораторная работа № 6. Минимизация булевых выражений с использованием карт Карно.....	27
7 Лабораторная работа № 7. Язык программирования Grafset.....	31
Список литературы.....	41

## Введение

Системы электрогидроавтоматики являются одним из основных классов систем промышленной автоматизации. Интеграция электрических и гидравлических устройств автоматики играет важную роль в решении многих задач, связанных с разработкой и реализацией современного оборудования строительных и дорожных машин.

В этих условиях инженеру, занимающемуся проектированием, наладкой и эксплуатацией систем электрогидроавтоматики, важно знать их структурно-функциональную организацию, технические параметры аппаратных средств, методы автоматизированного проектирования и моделирования указанных систем с использованием инструментального программного обеспечения.

Целью лабораторных занятий является изучения основных приемов работы в программном пакете FluidSIM и получение практических навыков в проектировании и моделировании систем электрогидроавтоматики.

Задание выдается преподавателем. Отчет представляется в виде файла FluidSIM с выполненным заданием.

# 1 Лабораторная работа № 1. Общие сведения о программном пакете FluidSIM

**Цель работы:** приобрести навыки работы с программным пакетом FluidSIM.

## 1.1 Назначение программного пакета FluidSIM

Программный пакет FluidSIM предназначен для моделирования гидравлических и электрогидравлических систем на этапе принятия схемотехнического решения. Моделируемая система представляется схемой в условных графических обозначениях (символах). Принятые для обозначения символы гидравлических и электрических элементов в FluidSIM такие же, как и условные графические обозначения, применяемые при составлении гидравлических и электрических схем. Они соответствуют символам стандартов DIN, ISO, SAE и ГОСТу. По существу, эта программа позволяет автоматизировать процесс создания электрогидравлической системы и проверять ее работоспособность благодаря реалистичному моделированию. При этом в процессе разработки гидравлической и электрической схем системы FluidSIM проверяет, является ли определенное соединение компонентов допустимым, и в случае нахождения ошибки показывает ее с комментариями.

Также FluidSIM предназначен для накопления, передачи и визуализации знаний в области гидравлики, т. к. компоненты сопровождаются текстовыми описаниями, фотографиями и т. п. FluidSIM может обмениваться информацией с внешним контроллером или многими приложениями Windows.

## 1.2 Запуск FluidSIM и создание нового проекта

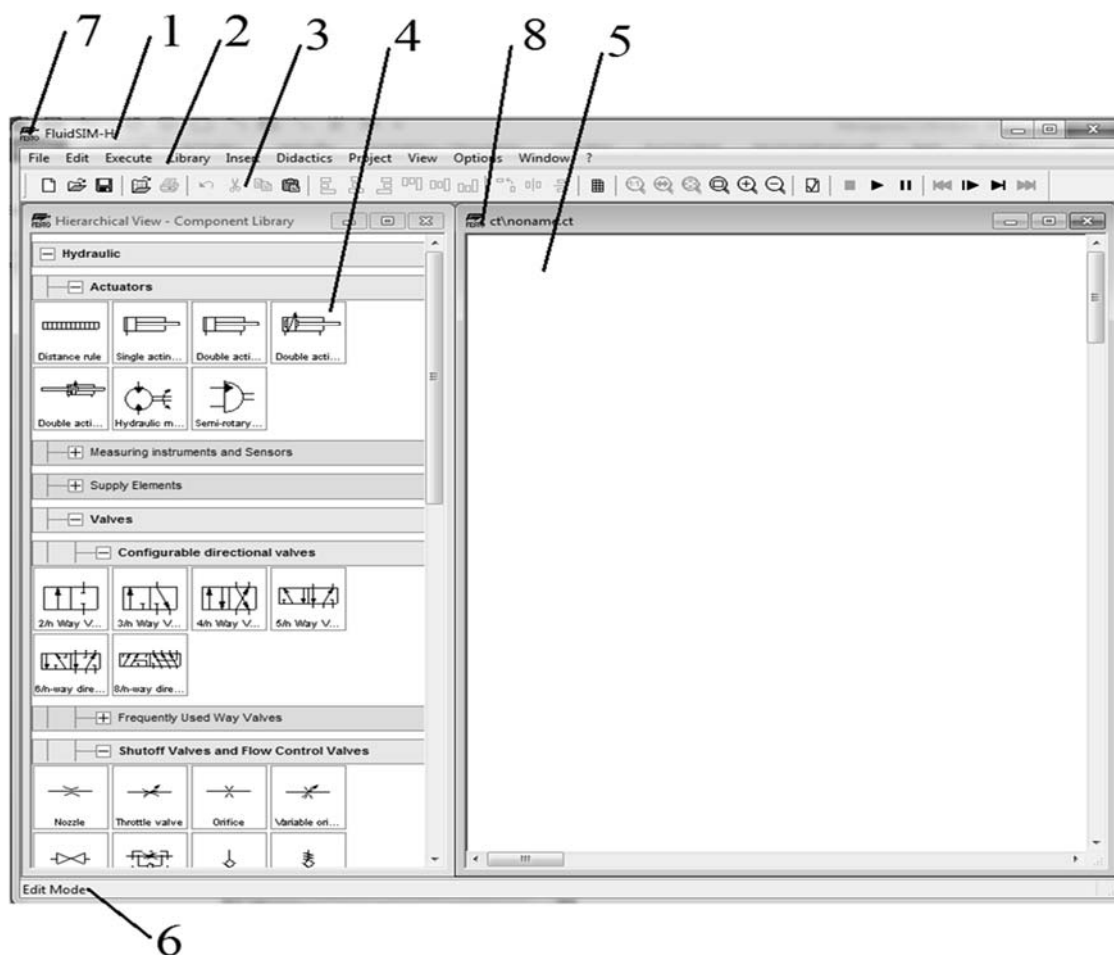
Для запуска программного пакета FluidSIM дважды щелкните левой кнопкой мыши на пиктограмме Festo FluidSIM, изображенной на рисунке 1.1.



Рисунок 1.1 – Пиктограмма Festo FluidSIM

После этого необходимо создать новое рабочее поле. Для этого в меню File нужно выбрать пункт New либо нажать на панели инструментов кнопку New. В результате этих действий в основном окне программы появится рабочее поле.

После этого «перетаскиванием» добиваемся удобного расположения окон. В результате получим окно, изображенное на рисунке 1.2.



1 – строка заголовка программы; 2 – главное меню; 3 – панели инструментов; 4 – библиотека гидравлических и электрических элементов и узлов; 5 – рабочее поле; 6 – строка состояния; 7 – системное меню; 8 – строка заголовка рабочего поля

Рисунок 1.2 – Основное окно FluidSIM с библиотекой и рабочим полем

### ***1.3 Основные элементы интерфейса FluidSIM***

После запуска программного пакета FluidSIM сначала отобразится заставка, а затем основное окно (см. рисунок 1.2).

Окно содержит:

1) строку заголовка программы. Строка заголовка программы содержит имя данной программы;

2) главное меню. Главное меню содержит все функции, которые необходимы для моделирования и построения диаграмм цепей. В главном меню находятся как стандартные функции Windows, так и собственные функции: Execute (выполнение функций), Library (библиотека), Didactics (обучение) и Project (проект);

3) панели инструментов. Содержит кнопки (пиктограммы) для наиболее часто используемых функций. Для отображения названия функции необходимо навести указатель мыши на соответствующую пиктограмму. Чтобы отобразить/скрыть панели инструментов, в главном меню выберите пункт View (Вид) – Toolbars (Панели инструментов). Положение панелей инструментов

можно изменять путем их «перетаскивания». Назначение кнопок на панели инструментов приведено в таблице 1.1;

4) библиотека гидравлических и электрических элементов и узлов. Содержит гидравлические и электрические элементы и узлы, которые используются для создания диаграмм. Положение библиотеки можно изменять путём её «перетаскивания». Чтобы отобразить/скрыть библиотеку, в главном меню выберите пункт Library (Библиотека) и далее – необходимую библиотеку;

5) рабочее поле. Рабочее поле представляет собой место, в котором будет создаваться и моделироваться диаграмма цепи;

6) строка состояния. Отображает информацию о текущих вычислениях и событиях, происходящих во время работы FluidSIM;

7) системное меню;

8) строка заголовка рабочего поля. Содержит имя и место расположения рабочего поля.

Таблица 1.1 – Кнопки на панели инструментов

Графическое изображение	Название	Назначение
	Создать	Создаёт новое рабочее поле
	Просмотр	Показывает сохраненные диаграммы цепи
	Открыть	Открывает сохраненную диаграмму цепи
	Сохранить	Сохраняет диаграмму цепи
	Печать	Печатает
	Отменить	Отменяет последний шаг редактирования
	Вырезать	Вырезает выделенные компоненты и переносит их в буфер обмена
	Копировать	Копирует выделенные элементы в буфер обмена
	Вставить	Переносит компонент из буфера обмена в область рисования
	Горизонтальное выравнивание слева	Выравнивает группу компонентов по горизонтали: по левому краю
	Горизонтальное выравнивание по центру	Выравнивает группу компонентов по горизонтали: по центру
	Горизонтальное выравнивание справа	Выравнивает группу компонентов по горизонтали: по правому краю

## Окончание таблицы 1.1

Графическое изображение	Название	Назначение
	Вертикальное выравнивание сверху	Выравнивает группу компонентов по вертикали: по верхнему краю
	Вертикальное выравнивание по центру	Выравнивает группу компонентов по вертикали: по левому центру
	Вертикальное выравнивание снизу	Выравнивает группу компонентов по вертикали: по нижнему краю
	Показывать сетку	Активирует фоновую сетку
	Стандартный размер	Показывает диаграмму цепи без увеличения или уменьшения
	Предыдущий вид	Переключает между последним видом и текущим увеличением
	Подогнать по окну	Масштабирует таким образом, что вся диаграмма может быть показана в окне
	Масштабировать по «резиновому обручу»	Превращает курсор мыши в «обруч» и увеличивает область выделенную «резиновым обручем»
	Увеличить	Увеличивает изображение диаграммы цепи
	Уменьшить	Уменьшает изображение диаграммы цепи
	Поверхностно проверить	Проверяет текущую диаграмму цепи на наличие ошибок рисования
	Стоп	Переключает текущую диаграмму в Режим редактирования
	Старт	Запускает моделирование
	Пауза	Останавливает моделирование текущей диаграммы без выхода из Режим моделирования
	Сброс	Переводит процесс моделирования в начальное состояние и моделирование сразу же начинается вновь
	Одиночный шаг	Останавливает моделирование через незначительное время
	Моделировать до изменения состояния	Запускает моделирование, которое продолжается до изменения состояния, после этого моделирование переходит на паузу
	Следующая тема	Вызывает переход к следующей теме презентации



### ***1.3 Библиотеки основных элементов пневматики и устройств электроавтоматики***

В библиотеке пневматических элементов и узлов находятся символы компонентов, которые используются при моделировании гидравлических систем. В библиотеку пневматических элементов входят следующие группы элементов:

- элементы питания рабочей жидкостью;
- распределители с механическим управлением;
- распределители с электромагнитным управлением;
- распределители с пневматическим управлением;
- клапаны и дроссели;
- датчики давления;
- исполнительные устройства;
- измерительные устройства.

В библиотеке электрических устройств находятся символы элементов, которые используются при моделировании электрических систем. В библиотеку электрических элементов входят следующие группы элементов:

- элементы подачи питания;
- сигнальные устройства;
- контакты;
- контакты с задержкой времени;
- контакты концевых выключателей;
- переключатели с ручным управлением;
- датчики давления;
- датчики положения;
- реле;
- логические элементы.

### ***1.4 Порядок выполнения***

Порядок выполнения следующий:

- запустить FluidSIM;
- создать новый документ;
- установить гидравлические и электрические элементы;
- изучить их параметры настройки;
- выйти из FluidSIM.

### ***Контрольные вопросы***

- 1 Как запустить моделирование?
- 2 Как пошагово запустить моделирование?
- 3 Как вызвать меню свойств элементов?

## 2 Лабораторная работа № 2. Методика проектирования и моделирования гидравлических схем

**Цель работы:** приобрести навыки проектирования и моделирования гидравлических схем.

### 2.1 Запуск FluidSIM и создание нового проекта

Для запуска программного пакета FluidSIM дважды щелкните левой кнопкой мыши на пиктограмме Festo FluidSIM, изображенной на рисунке 2.1.



Рисунок 2.1 – Пиктограмма Festo FluidSIM

После этого необходимо создать новое рабочее поле. Для этого в меню File нужно выбрать пункт New либо нажать на панели инструментов кнопку New. В результате этих действий в основном окне программы появится рабочее поле.

После этого «перетаскиванием» добиваемся удобного расположения окон. В результате получим окно, изображенное на рисунке 1.2.

### 2.2 Проектирование гидравлической схемы

Из библиотеки элементов на рабочее поле выберем следующие элементы.

- 1 Гидроцилиндр двухстороннего действия.
- 2 Распределитель 4/2 с электромагнитным управлением и с возвратной пружиной.
- 3 Предохранительный клапан.
- 4 Насос.
- 5 Бак.

Так как из библиотеки был выбран стандартный 4/2-распределитель, то необходимо изменить параметры настройки данного распределителя. Для этого нужно нажать два раза левой кнопкой мыши (ЛКМ) на условном обозначении распределителя и в окне Configure Way Valve (рисунок 2.2) задать для него использование электромагнитного привода и возвратной пружины. Для этого в поле left actuator (левый привод) выберем катушку электромагнита, а в поле right actuator (правый привод) необходимо установить галочку у словосочетания Spring-returned («Возвратная пружина»).

После этого закрываем окно нажатием кнопки «ОК».

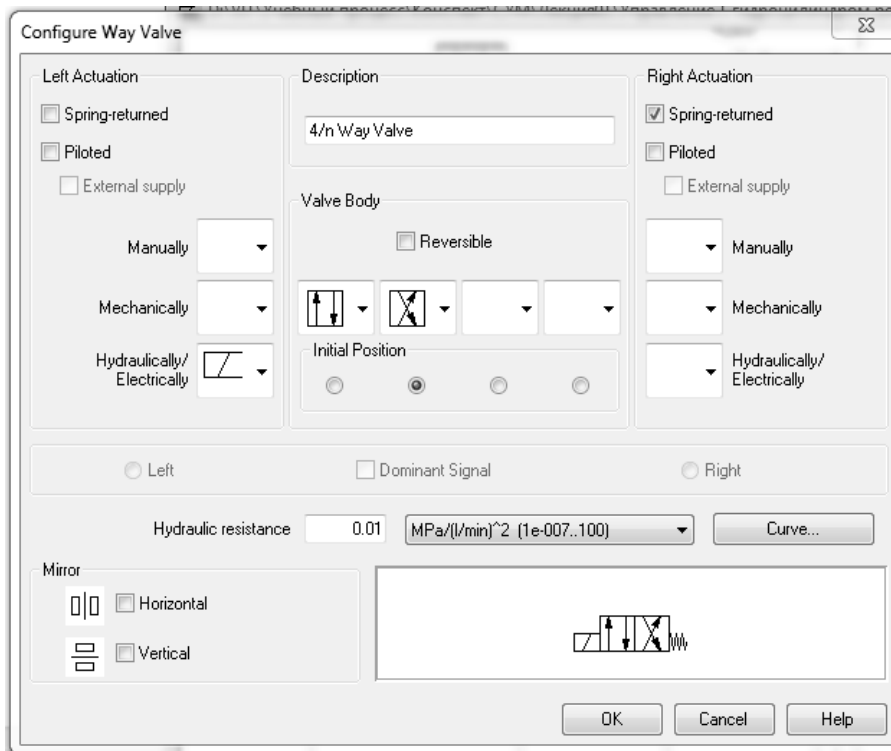


Рисунок 2.2 – Задание параметров настройки гидрораспределителя

В результате получим гидравлическую схему без соединений, изображенную на рисунке 2.3.

Теперь необходимо соединить компоненты. Для этого наводим курсор на место соединения, курсор превратился в указатель в виде прицела. Затем нажимаем ЛКМ и перемещаем курсор, при этом курсор изображается в виде прицела со стрелочками. Наводим курсор на место соединения (при этом стрелочки курсора повернулись во внутрь) и отпускаем ЛКМ. В итоге два места соединения соединились линией. Для создания Т-образного соединения нужно привести курсор в виде прицела со стрелками во внутрь на соединительную линию и отпустить ЛКМ. После того как все компоненты соединены, получим пневматическую схему, показанную на рисунке 2.4.

Для проверки работоспособности гидросхемы нажимаем кнопку Старт. Нажимая ЛКМ на левую секцию распределителя и отпуская ее, смотрим, как работает гидросхема.

### 2.3 Порядок выполнения

Порядок выполнения следующий:

- запустить FluidSIM;
- создать новый документ;
- установить гидравлические элементы;
- соединить их трубопроводами;
- проверить работоспособность гидросхемы;
- выйти из FluidSIM.

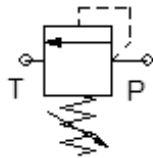
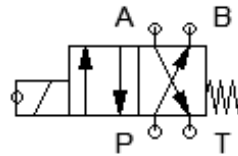
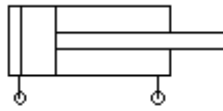


Рисунок 2.3 – Гидравлическая схема без соединений

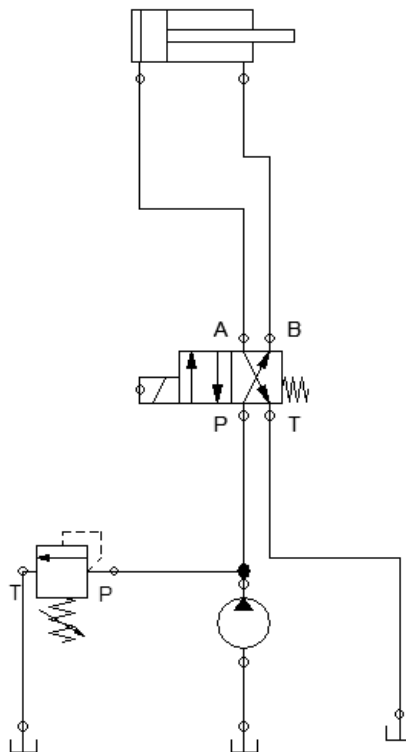


Рисунок 2.4 – Гидравлическая схема

## Контрольные вопросы

- 1 Назначение параметров настройки гидравлических элементов.
- 2 Порядок изменения параметров настройки гидравлических элементов.
- 3 Влияние параметров настройки гидравлических элементов на работу гидросистемы.

## 3 Лабораторная работа № 3. Методика проектирования и моделирования электрогидравлических схем

**Цель работы:** приобрести навыки проектирования и моделирования электрогидравлических схем.

### 3.1 Проектирование простой электрогидравлической схемы

В данном примере рассмотрены основные приемы работы в программной среде FluidSIM при проектировании пневматических и электрических схем.

**Задача.** Спроектировать электрогидравлическую схему, обеспечивающую разное возвратно поступательное движение гидроцилиндра в автоматическом режиме.

**3.1.1 Проектирование гидравлической схемы.** Проектирование гидравлической схемы (см. предыдущую тему).

**3.1.2 Проектирование электрической схемы.** Для того чтобы автоматизировать движение гидроцилиндра с использованием распределителя с одним соленоидом и пружиной, необходимо установить на гидроцилиндр датчики положения (рисунок 3.1).

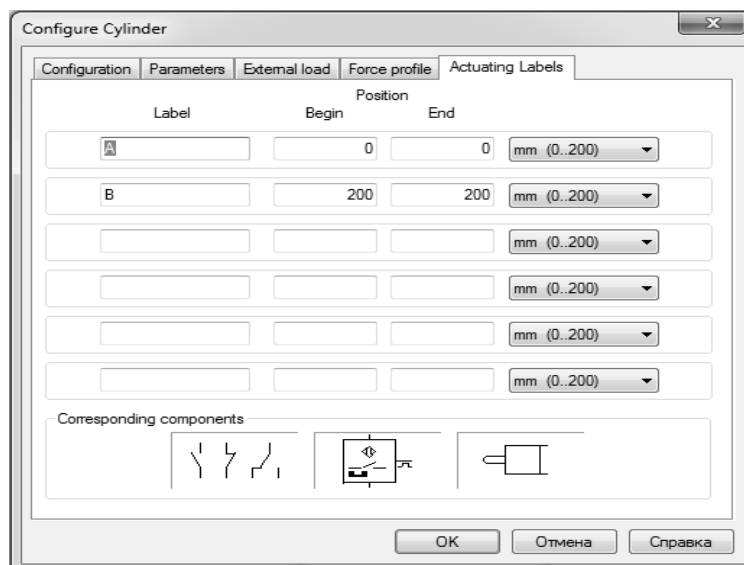


Рисунок 3.1 – Установка датчиков положения на гидроцилиндр

Значения задаем таким образом, чтобы фиксировать положения гидроцилиндра в крайних положениях.

На гидроцилиндре появится следующее изображение (рисунок 3.2).

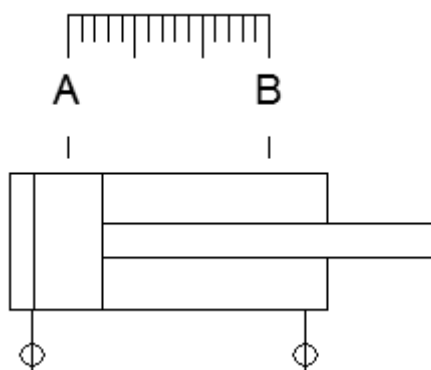


Рисунок 3.2 – Обозначение датчиков положения на схеме

Датчики срабатывают, когда поршень гидроцилиндра находится напротив зоны действия датчика, и отключаются, как только поршень отходит от данной зоны. При автоматизации движения гидроцилиндра данный фактор необходимо постоянно держать во внимании.

Для составления схемы автоматизации вначале необходимо составить подробное описание работы системы (таблица 3.1).

Таблица 3.1 – Описание работы системы

Описание необходимых операций (алгоритм)	Входной сигнал		Выходной сигнал
	А	В	
Когда гидроцилиндр Ц находится в положении А, нормально разомкнутые контакты датчика А замкнуты и подают питание на соленоид 1S	1	0	1S = 1
Питание подается на реле 1R, которое своими контактами шунтирует контакты датчика А, т. е. когда контакты датчика А разомкнутся, ток будет протекать через контакты реле 1R на саму катушку реле и соленоид 1S. Распределитель передвинется вправо			1R = 1
Когда гидроцилиндр Ц дойдет до положения В, нормально замкнутые контакты датчика В разомкнутся и снимут питание с соленоида 1S. Распределитель передвинется влево. Гидроцилиндр будет двигаться до положения А	0	1	1S = 0

Теперь спроектируем электрическую схему, которая будет управлять распределителем. Для этого из библиотеки на рабочее поле выберем следующие элементы.

- 1 Две шины питания 0 и 24 В.
- 2 Нормально разомкнутая кнопка с фиксацией.
- 3 Электромагнит, управляющий распределителем.

- 4 Реле.
- 5 Два нормально разомкнутых контакта.
- 6 Один нормально замкнутый контакт.

Затем соединяем выбранные элементы. Соединения элементов электрической схемы выполняются аналогично соединениям в пневматической схеме. В результате получим электрическую схему, показанную на рисунке 3.3.

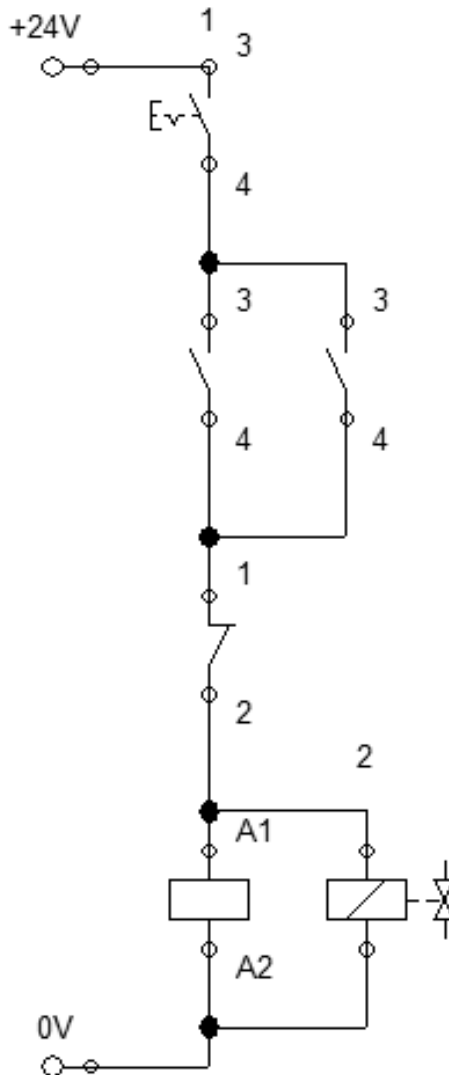


Рисунок 3.3 – Электрическая схема

*3.1.3 Проектирование электрогидравлической схемы.* Для проектирования электропневматической схемы необходимо установить связь между электрической и гидравлической схемой. Т. е. необходимо связать электромагнит, управляющий распределителем, датчики положения гидроцилиндра, реле с контактами электрической схемы. Это делается с помощью меток.

Для задания меток нажимаем 2 раза ЛКМ на электромагнит, и в открывшемся окне Valve solenoid (Электромагнит распределителя) в поле Label (Метка) вводим 1S (рисунок 3.4).

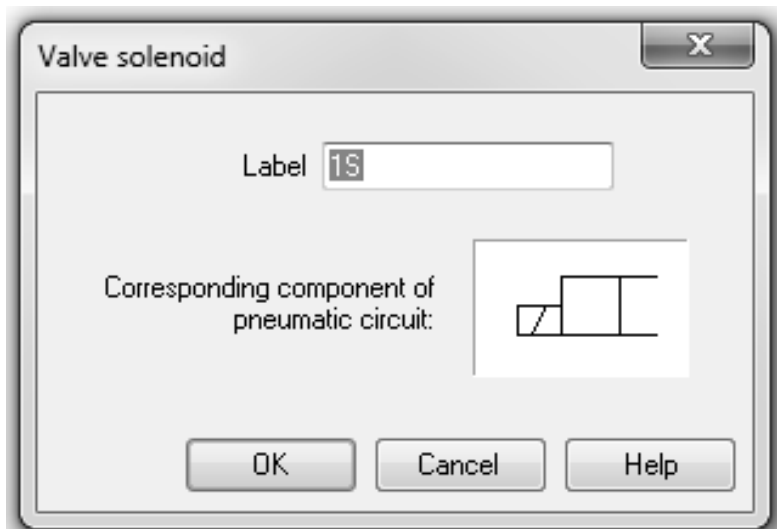


Рисунок 3.4 – Окно Valve solenoid

После этого нажимаем два раза ЛКМ на точку соединения электромагнита распределителя и в окне Valve solenoid connection вводим метку 1S (рисунок 3.5).

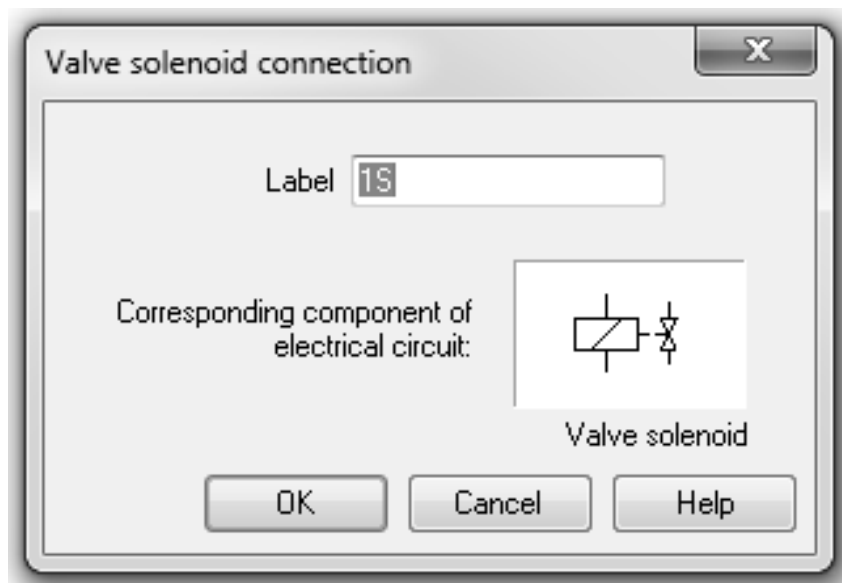


Рисунок 3.5 – Окно Valve solenoid connection

Такие же операции производим для датчиков положение гидроцилиндра и реле. В итоге составили гидроэлектрическую систему, изображенную на рисунке 3.6.

Для моделирования работы собранной пневмоэлектрической схемы необходимо перейти в режим моделирования. Для этого необходимо на панели инструментов нажать кнопку «Старт».

Промоделируем работу гидроэлектрической схемы. Для этого нажмем левую кнопку мыши на нормально разомкнутой кнопке. При этом электрические линии станут красными, это означает, что по ним протекает ток. Гидравлические линии станут коричневыми, это означает, что в них имеется давление.



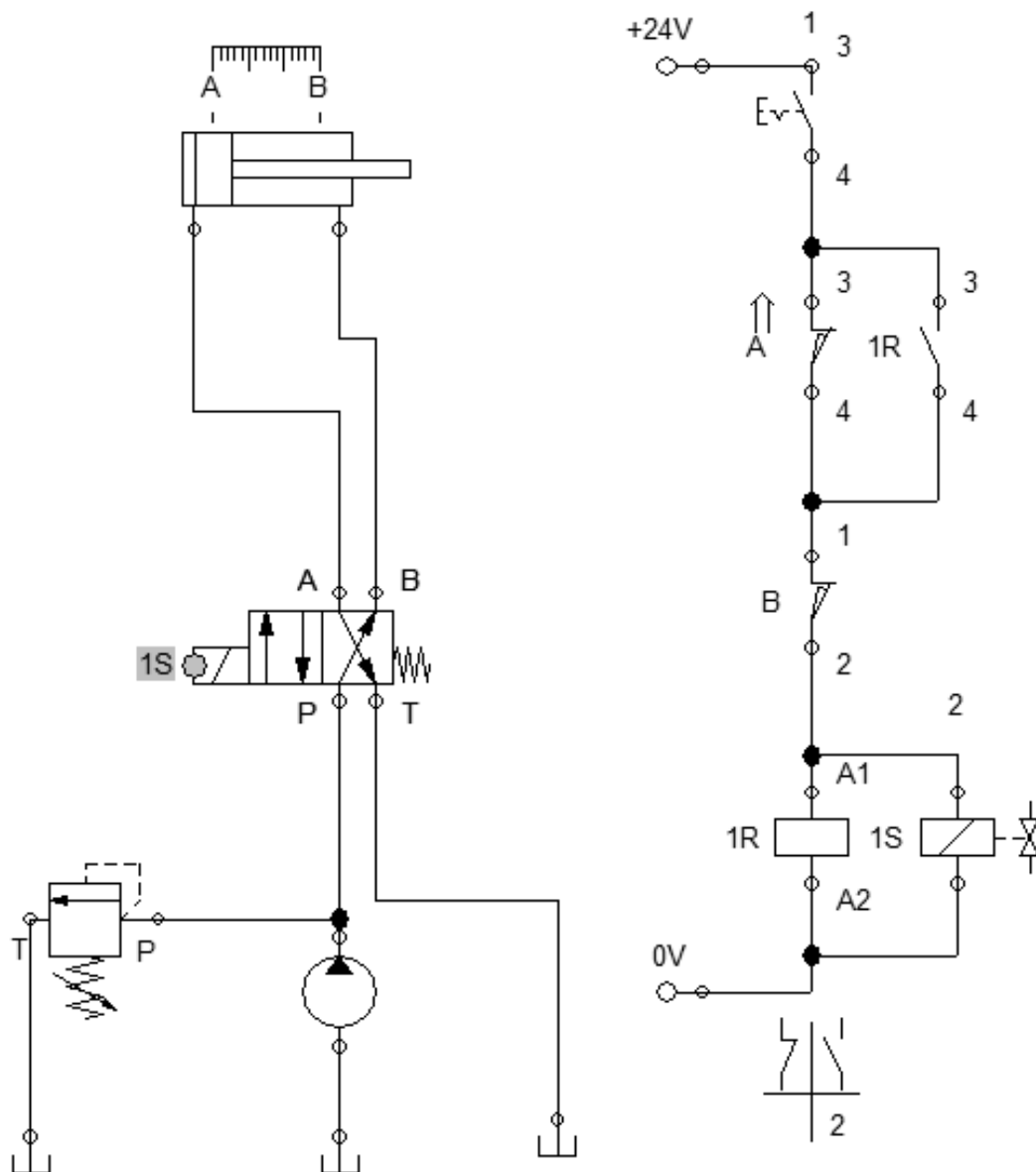


Рисунок 3.6 – Гидроэлектрическая система

### 3.2 Порядок выполнения

Порядок выполнения следующий:

- запустить FluidSIM;
- создать новый документ;
- установить гидравлические элементы;
- соединить их трубопроводами;
- проверить работоспособность гидросхемы;
- составить описание работы системы;
- спроектировать электросхему;
- установить связь между электрической и гидравлической схемой;
- проверить работоспособность системы;
- выйти из FluidSIM.

### ***Контрольные вопросы***

- 1 Каким образом устанавливается связь между электрической и гидравлической схемой?
- 2 Как установить датчик на гидроцилиндр?
- 3 Как датчики гидроцилиндра и контакты реле учитываются на электросхеме?

## **4 Лабораторная работа № 4. Логические элементы**

**Цель работы:** изучить работу логических элементов.

### ***4.1 Основные сведения***

Любое цифровое устройство – персональный компьютер, или современная система автоматики, состоит из цифровых интегральных микросхем (ИМС), которые выполняют определённые сложные функции. Но для выполнения одной сложной функции необходимо выполнить несколько простейших функций. Например, сложение двух двоичных чисел размером в один байт происходит внутри цифровой микросхемы, называемой «процессор» и выполняется в несколько этапов большим количеством логических элементов, находящихся внутри процессора. Двоичные числа сначала запоминаются в буферной памяти процессора, потом переписываются в специальные «главные» регистры процессора, после выполняется их сложение, запоминание результата в другом регистре, и лишь после результат сложения выводится через буферную память из процессора на другие устройства компьютера.

Процессор состоит из функциональных узлов: интерфейсов ввода-вывода, ячеек памяти – буферных регистров и «аккумуляторов», сумматоров, регистров сдвига и т. д. Эти функциональные узлы состоят из простейших логических элементов, которые, в свою очередь, состоят из полупроводниковых транзисторов, диодов и резисторов. При конструировании простых триггерных и других электронных импульсных схем сложные процессоры не применить, а использовать транзисторные каскады – «прошлый век». Тут и приходят на помощь логические элементы.

Логические элементы – это простейшие «кубики», составные части цифровой микросхемы, выполняющие определённые логические функции. При этом цифровая микросхема может содержать в себе от одного до нескольких единиц, десятков и до нескольких сотен тысяч логических элементов в зависимости от степени интеграции. Для того чтобы разобраться, что такое логические элементы, будем рассматривать самые простейшие из них. А потом, наращивая знания, разберёмся и с более сложными цифровыми элементами.

Начнём с того, что единица цифровой информации – «один бит». Он может принимать два логических состояния – логический ноль «0», когда напря-

жение равно нулю (низкий уровень), и состояние логической единицы «1», когда напряжение равно напряжению питания микросхемы (высокий уровень).

Для того чтобы понять принципы работы простейших логических элементов, используется «таблица истинности».

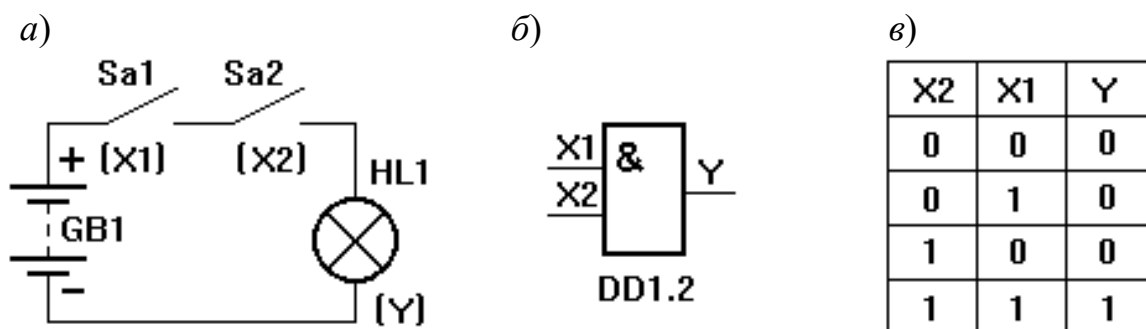
Логические элементы могут работать в режимах положительной и отрицательной логики. Для электронных логических элементов в режиме положительной логики логической единице соответствует высокий уровень напряжения, а логическому нулю – низкий уровень напряжения. В режиме отрицательной логики логической единице соответствует низкий уровень напряжения, а логическому нулю – высокий.

Для контактно-релейных схем в режиме положительной логики логической единице соответствует замкнутый контакт ключа или реле, а логическому нулю – разомкнутый. Светящийся индикатор (лампочка, светодиод) соответствует логической единице, а несветящийся – логическому нулю.

Логические элементы, реализующие для режима положительной логики операцию И, для режима отрицательной логики выполняют операцию ИЛИ, и наоборот. Так, например, микросхема, реализующая для положительной логики функции элемента 2И-НЕ, будет выполнять для отрицательной логики функции элемента 2ИЛИ-НЕ.

Как правило, паспортное обозначение логического элемента соответствует функции, реализуемой «положительной логикой». Логические элементы И, ИЛИ, НЕ имеют один выход, число входов логических элементов И, ИЛИ может быть любым, начиная с двух. Логические элементы И и ИЛИ, выпускаемые в составе микросхем, обычно имеют 2, 3, 4, 8 входов. В названии элемента первая цифра указывает число входов.

Прежде всего рассмотрим реализацию логических элементов с помощью контактно-релейных схем. Рассмотрим логический элемент 2И. Он выполняет операцию логического умножения. На рисунке 4.1, а приведена контактно-релейная схема логического элемента 2И для режима положительной логики.



а – электрическая схема; б – условное обозначение; в – таблица истинности

Рисунок 4.1 – Логический элемент И

Обозначение логического элемента 2И на принципиальных схемах показано на рисунке 4.1, б. Знак & (амперсанта) в левом верхнем углу прямоугольника

указывает, что это логический элемент И. Первые две буквы обозначения DD1.2 указывают на то, что это цифровая микросхема, цифра слева от точки указывает номер микросхемы на принципиальной схеме, а цифра справа от точки – номер логического элемента в составе данной микросхемы.

Функционирование логического элемента обычно задают таблицей истинности (рисунок 4.1, в). Контактно-релейная схема логического элемента 2И (режим положительной логики) позволяет легко составить таблицу истинности этого элемента. Так как микросхема имеет для подачи входных сигналов два входа, то возможны  $2 \times 2 = 4$  различных комбинации входных сигналов. Необходимо проанализировать состояние лампочки при различных положениях тумблеров Sa1, Sa2, т. е. рассмотреть четыре различные комбинации состояний тумблеров (см. рисунок 4.1, в).

Введение понятия активного логического уровня существенно облегчает анализ функционирования сложных цифровых устройств. Активным логическим уровнем на входе элемента (логический ноль, логическая единица) называется такой уровень, который однозначно задает состояние на выходе элемента независимо от логических уровней на остальных входах элемента. Активный логический уровень на одном из входов элемента определяет уровень на его выходе. Уровни, обратные активным, называются пассивными логическими уровнями.

Активным логическим уровнем для элементов И является логический ноль. Пусть, например, имеем логический элемент 8И. Необходимо проанализировать  $2^8 = 256$  различных состояний для составления таблицы истинности этого элемента. Воспользуемся понятием активного логического уровня. Если хотя бы на одном из входов этого элемента будет активный логический уровень, то состояние на выходе элемента определено однозначно и нет необходимости анализировать состояния на остальных входах элемента.

Таким образом, таблицу истинности логического элемента 8И можно свести к двум строчкам: на выходе этого элемента будет логическая единица, если на всех входах будут сигналы логической единицы, и на выходе будет логический ноль, если хотя бы на одном из входов элемента будет сигнал логического нуля.

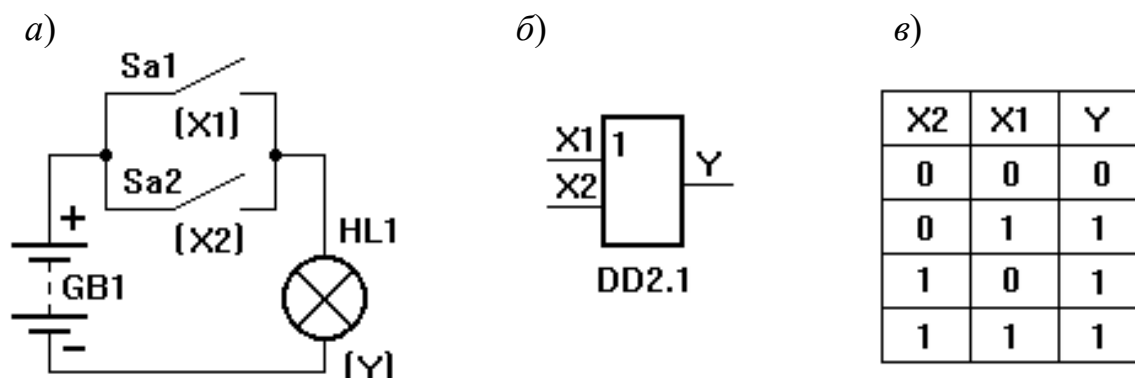
Логическая операция, выполняемая элементом И,

$$Y = X1 \wedge X2 \text{ или } Y = X1 \cdot X2.$$

Логический элемент 2ИЛИ выполняет логическую операцию логического сложения. Контактно-релейная схема элемента приведена на рисунке 4.2, а, а его условное обозначение – на рисунке 4.2, б. Знание контактно-релейной схемы элемента позволяет составить таблицу истинности (рисунок 4.2, в). Лампочка будет гореть, если замкнуты контакты хотя бы одного тумблера, т. е. активным логическим уровнем для элементов ИЛИ является уровень логической единицы.

Логический элемент НЕ выполняет операцию отрицания, и для этого элемента проще составить сразу таблицу истинности, а не вычерчивать сначала

контактно-релейную схему, а затем по ней составлять таблицу истинности. Для логических элементов И и ИЛИ проще сначала вычертить контактно-релейную схему, а уже потом составлять таблицу истинности.



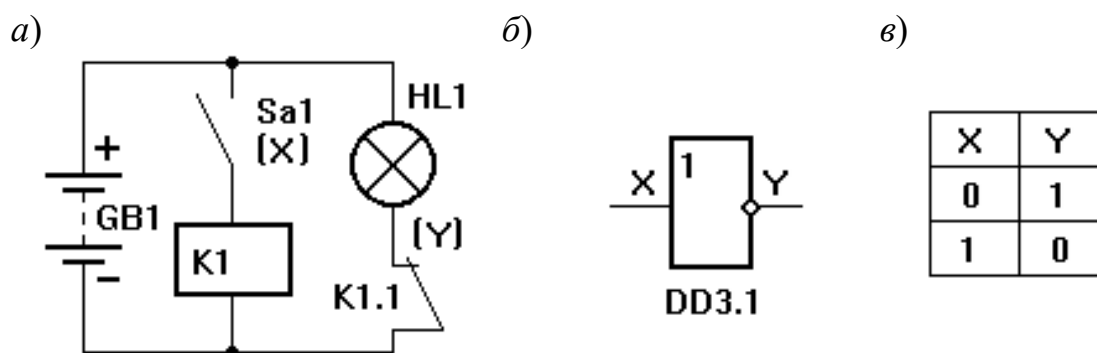
*a* – электрическая схема; *б* – условное обозначение; *в* – таблица истинности

Рисунок 4.2 – Логический элемент ИЛИ

Логическая операция, выполняемая элементом ИЛИ,

$$Y = X1 \vee X2 \text{ или } Y = X1 + X2.$$

Напомним алгоритм работы электромагнитного реле с нормально замкнутыми контактами: при отсутствии электрического тока через обмотку реле контакты реле замкнуты, а при протекании достаточного тока через обмотку реле контакты реле разомкнуты. Контактная релейная схема элемента НЕ приведена на рисунке 4.3, *a*, а его условное обозначение – на рисунке 4.3, *б*.



*a* – электрическая схема; *б* – условное обозначение; *в* – таблица истинности

Рисунок 4.3 – Логический элемент НЕ

Проанализируем работу контактно-релейной схемы логического элемента НЕ (см. рисунок 4.3, *a*). Если контакты ключа Sa1 разомкнуты, то через обмотку К электромагнитного реле ток протекать не будет. Контакты К1.1 (цифра слева от точки указывает номер реле на принципиальной схеме, а цифра справа – номер контактной группы данного реле) будут замкнуты (электромагнит-

ное реле с нормально замкнутыми контактами). Электрическая лампочка HL1 в этом случае будет гореть, что для режима положительной логики будет означать логическую единицу. При замкнутых контактах ключа Sa1 (на входе элемента логическая единица) через обмотку реле протекает ток, достаточный для размыкания контактов K1.1, поэтому лампочка перестает гореть (логический нуль). В результате анализа получили, что сигнал на выходе элемента противоположен сигналу на входе, т. е. если на входе элемента сигнал логической единицы, то на выходе элемента сигнал логического нуля, и наоборот (рисунок 4.3, в).

Логическая операция, выполняемая элементом ИЛИ,

$$Y = \bar{X}.$$

Триггер – это электронное устройство, которое предназначается для записи и хранения информации. Обычно он имеет два выхода: прямой и инверсный; и некоторое количество входов в зависимости от выполняемой задачи. Под действием входных сигналов изменяется состояние выходов. Напряжение на выходах изменяется резко – скачкообразно. Для изготовления триггеров обычно используются биполярные, униполярные транзисторы (полупроводниковые приборы).

Информация может записываться в триггеры свободно (непрерывно), т. е. при подаче сигналов на вход состояние выхода меняется в реальном времени. Такие триггеры называются асинхронными. А может информация записываться, только когда активен синхронизирующий сигнал. При отсутствии положительного уровня напряжения на нем информация на выходах измениться не может – синхронные (тактируемые) триггеры.

RS-триггер именуется так из-за названия его входов:

R – reset (сбросить);

S – set (установить).

Он оснащен двумя входами, как говорилось, и двумя выходами:

Q – прямой выход;

$\bar{Q}$  – инверсный.

Асинхронный RS-триггер можно реализовать на логических элементах двумя схемами:

2ИЛИ-НЕ;

2И-НЕ.

Схема реализована на двух логических ИЛИ-НЕ. По рисунку 4.4 рассмотрим принцип работы приведенного RS-триггера. В нулевой момент времени, когда ни на один вход (R и S) не подана логическая единица, прямой выход  $Q=0$ , соответственно, инверсный  $\bar{Q}=1$ . Если на вход S подать напряжение, уровень которого будет соответствовать единице, то выход Q скачкообразно изменит свое значение на 1, а  $\bar{Q}$  на 0. Это произойдет запись информации. Если убрать единицу с Set, тогда выходы не изменят свое состояние, останутся такими, какими были, – проявление свойства памяти. При подаче положительного сигнала на вход сброса, т. е.  $R=1$ , инверсный выход резко станет равен 1,

а прямой  $Q = 0$ . В работе RS-триггера есть недостаток: существует запрещенная комбинация. Нельзя одновременно подавать единичные сигналы на оба входа, нормальная работа триггера в этом случае невозможна.

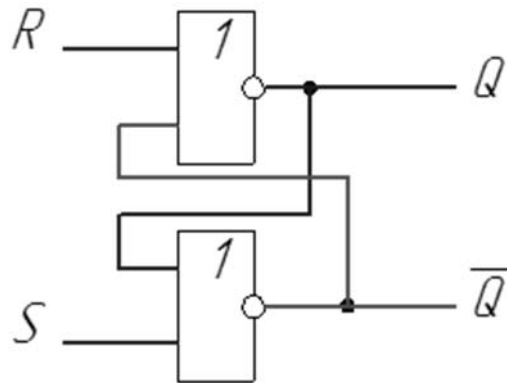


Рисунок 4.4 – Схема асинхронного RS-триггера на логических 2ИЛИ-НЕ элементах

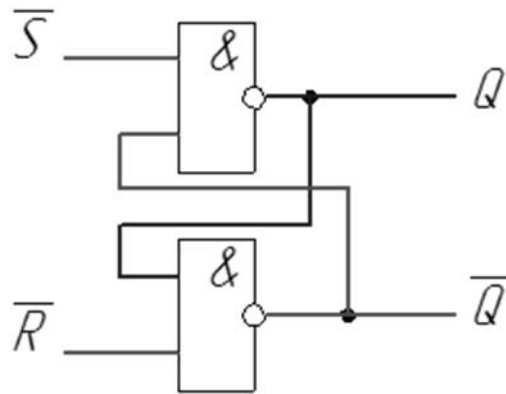


Рисунок 4.5 – Схема асинхронного RS-триггера на логических 2И-НЕ элементах

Вторая схема собрана с помощью двух логических элементов И-НЕ. Разница между ними заключается в том, что управление в прошлой схеме осуществлялось положительным сигналом (единицей), а в текущей активный уровень – ноль. Работают обе схемы идентично, поэтому описание принципа действия здесь не требуется.

## 4.2 Порядок выполнения

Порядок выполнения следующий:

- запустить FluidSIM;
- создать новый документ;
- установить логические элементы;
- подключить к источнику питания;
- составить таблицу истинности;
- создать новый документ;
- собрать триггер;

- проверить работоспособность схемы;
- выйти из FluidSIM.

### ***Контрольные вопросы***

- 1 Таблицы истинности логических элементов.
- 2 Назначение триггеров.
- 3 Отличия работы триггеров, составленных на логических элементах И-НЕ и ИЛИ-НЕ.

## **5 Лабораторная работа № 5. Составление логических схем на основе таблиц истинности**

**Цель работы:** получить навыки составления логических схем на основе таблиц истинности.

### ***5.1 Основные сведения***

В обыденной речи для образования сложного предложения из простых мы всегда используем связки – особые части речи, соединяющие отдельные предложения. Наиболее часто при этом употребляются связки и, или, нет, если ... то, только если и тогда и только тогда. В логике также существует ряд таких, привычных нам связок, выполняющих те же функции. Естественно, существуют отличия связок обыденной речи от связок в логике (например, в логике смысл высказываний должен быть определен однозначно). Но цель – определение истинности одинакова в обоих случаях. Далее будут рассмотрены все виды логических связок и алгоритмы их построения.

Таблица истинности – таблица, с помощью которой устанавливается истинностное значение сложного высказывания при данных значениях входящих в него простых высказываний. В классической математической логике предполагается, что каждое простое (не содержащее логических связок) высказывание является либо истинным, либо ложным, но не тем и другим одновременно. Нам не известно, истинно или ложно данное простое высказывание, чтобы установить это, потребовалось бы обратиться к фактам действительности, но логика этого не делает. Однако мы знаем, что у высказывания имеется лишь две возможности – быть истинным либо быть ложным. Когда с помощью логических связок мы соединяем простые высказывания в сложное, встает вопрос: при каких условиях сложное высказывание считается истинным, а при каких – ложным? Для ответа на этот вопрос и служат таблицы истинности. Каждая логическая связка имеет свою таблицу, которая показывает, при каких наборах значений простых высказываний сложное высказывание с этой связкой будет истинным, а при каких – ложным.

Для того чтобы составить таблицу истинности автоматизации работы объ-



екта, необходимо четкое понимание его функционирования, знание последовательности выполнения операций, а также исключение поломки механической части в случае отказа электронных систем.

Прежде чем приступить к разработке таблицы истинности, необходимо составить словесное описание объекта и разделить его на простые действия в зависимости от срабатывания датчиков.

Составление логической функции работы объекта и реализацию его на логических элементах рассмотрим на примере произвольной таблицы истинности (таблица 5.1).

Таблица 5.1 – Таблица истинности работы объекта

A	B	C	Y
1	1	1	0
0	1	1	1
1	0	1	0
0	0	1	1
1	1	0	0
0	1	0	1
1	0	0	0
0	0	0	0

В таблице 5.1 буквами А, В и С обозначены входные сигналы от дискретных датчиков. При этом приняты обозначения: 1 – датчик сработал, 0 – нет, Y – выходной сигнал. При этом приняты обозначения: 1 – сигнал подается на исполнительный механизм, 0 – нет.

Для составления булевой функции для данного объекта необходимо учитывать только строки с  $Y = 1$ . При этом сигналы в строке перемножаются, а строки складываются. Сигнал со значением 0 обозначается как инверсный.

Булева функция будет иметь следующий вид:

$$\bar{A} \cdot B \cdot C + \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} = Y .$$

Построение следует начинать справа на лево. В начале обозначаем логический элемент ИЛИ с тремя входами, т. к. имеются три элемента, которые суммируются. Затем обозначаем три элемента И с тремя входами, т. к. перемножаются по три сигнала. К каждому элементу И подводим соответствующий сигнал А, В, С. Если сигнал инверсный (т. е. с черточкой наверху), то перед входом ставим логический элемент НЕ.

При составлении схемы во избежание путаницы необходимо во всех элементах И соблюдать одну и ту же последовательность сигналов А, В, С.

Схема для полученной булевой функции представлена на рисунке 5.1.

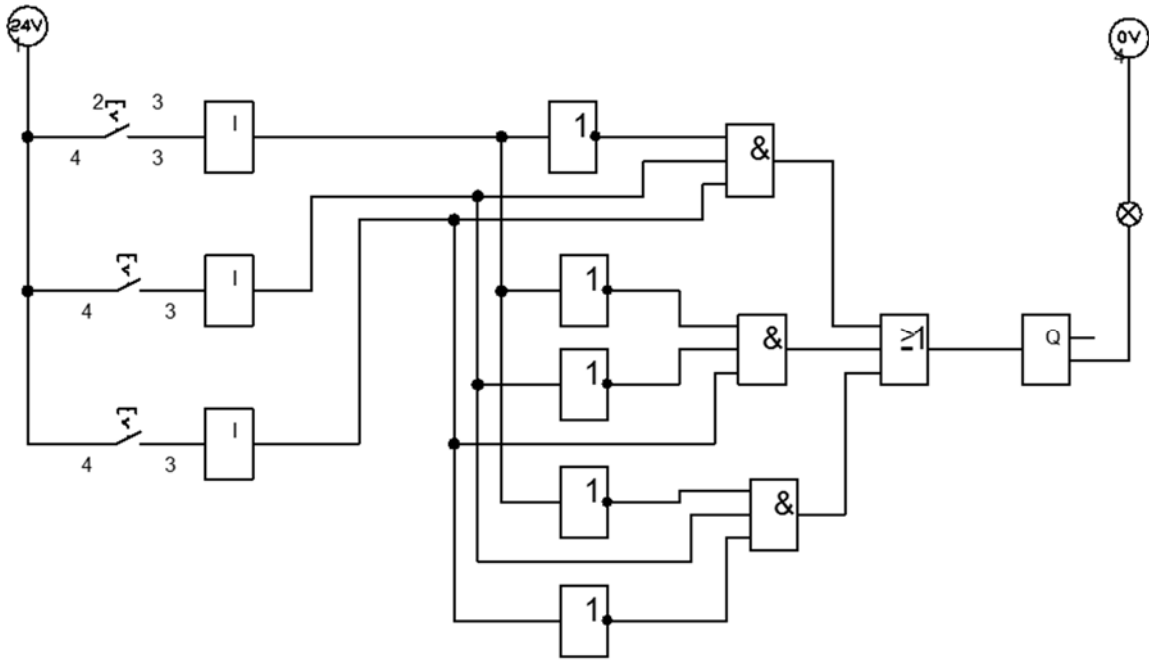


Рисунок 5.1 – Логическая схема

Анализируя данную схему, видим, что в схеме имеются элементы, присутствующие в программном пакете FluidSIM:

 – входное устройство согласования;

 – выходное устройство согласования.

## 5.2 Порядок выполнения

Порядок выполнения следующий:

- по заданной таблице истинности составить булеву функцию;
- запустить FluidSIM;
- создать новый документ;
- установить логические элементы;
- соединить элементы в соответствии с булевой функцией;
- проверить работоспособность схемы;
- выйти из FluidSIM.

## Контрольные вопросы

- 1 Правила составления булевой функции по таблице истинности.
- 2 Правила составления таблицы истинности.
- 3 Правила составления логической схемы в соответствии с булевой функцией.

## 6 Лабораторная работа № 6. Минимизация булевых выражений с использованием карт Карно

**Цель работы:** получить навыки минимизации булевых выражений с использованием карт Карно.

### 6.1 Основные сведения

Карта Карно – графический способ минимизации переключательных (булевых) функций, обеспечивающий относительную простоту работы с большими выражениями и устранение потенциальных гонок. Представляет собой операции попарного неполного склеивания и элементарного поглощения. Карты Карно рассматриваются как перестроенная соответствующим образом таблица истинности функции.

Карты Карно были изобретены в 1952 г. Эдвардом В. Вейчем, усовершенствованы в 1953 г. Морисом Карно, физиком из Bell Labs, и были призваны помочь упростить цифровые электронные схемы. В карту Карно булевы переменные передаются из таблицы истинности и упорядочиваются с помощью кода Грея, в котором каждое следующее число отличается от предыдущего только одним разрядом.

При составлении карты Карно необходимо следить за тем, чтобы инвертировался только один сигнал. На рисунке 6.1 показаны карты Карно для двух и четырех сигналов.

a)	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td></td> <td><math>B</math></td> <td><math>\bar{B}</math></td> </tr> <tr> <td><math>A</math></td> <td></td> <td></td> </tr> <tr> <td><math>\bar{A}</math></td> <td></td> <td></td> </tr> </table>		$B$	$\bar{B}$	$A$			$\bar{A}$		
	$B$	$\bar{B}$								
$A$										
$\bar{A}$										

б)	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td></td> <td><math>CD</math></td> <td><math>\bar{C}D</math></td> <td><math>C\bar{D}</math></td> <td><math>\bar{C}\bar{D}</math></td> </tr> <tr> <td><math>AB</math></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td><math>\bar{A}B</math></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td><math>A\bar{B}</math></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td><math>\bar{A}\bar{B}</math></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>		$CD$	$\bar{C}D$	$C\bar{D}$	$\bar{C}\bar{D}$	$AB$					$\bar{A}B$					$A\bar{B}$					$\bar{A}\bar{B}$				
	$CD$	$\bar{C}D$	$C\bar{D}$	$\bar{C}\bar{D}$																						
$AB$																										
$\bar{A}B$																										
$A\bar{B}$																										
$\bar{A}\bar{B}$																										

*a* – карта Карно для двух сигналов; *б* – карта Карно для четырех сигналов

Рисунок 6.1 – Карта Карно для двух (*a*) и для четырех (*б*) сигналов

Карта Карно может быть составлена для любого количества переменных, однако удобно работать при количестве переменных не более пяти. По сути, карта Карно – это таблица истинности, составленная в 2-мерном виде. Благодаря использованию кода Грея в ней верхняя строка является соседней с нижней, а правый столбец – соседний с левым, т. е. вся карта Карно сворачивается в фигуру тор (бублик). На пересечении строки и столбца проставляется соответствующее значение из таблицы истинности. После того как карта заполнена, можно приступать к минимизации. При этом рассматриваем только те клетки, которые содержат единицы. Сама минимизация производится по следующим

правилам.

1 Объединяем смежные клетки, содержащие единицы, в область так, чтобы одна область содержала  $2^n$  ( $n$  целое число =  $0 \dots \infty$ ) клеток (помним про то, что крайние строки и столбцы являются соседними между собой), в области не должно находиться клеток, содержащих нули.

2 Область должна располагаться симметрично оси(ей) (оси располагаются через каждые четыре клетки).

3 Несмежные области, расположенные симметрично оси(ей), могут объединяться в одну.

4 Область должна быть как можно больше, а количество областей как можно меньше.

5 Области могут пересекаться.

6 Возможно несколько вариантов покрытия.

Далее берём первую область и смотрим, какие переменные не меняются в пределах этой области, выписываем конъюнкцию этих переменных; если неменяющаяся переменная нулевая, проставляем над ней инверсию. Берём следующую область, выполняем то же самое, что и для первой, и т. д. для всех областей. Конъюнкции областей объединяем дизъюнкцией. Минимизацию булевых выражений покажем на примерах.

**Пример 1** – Минимизация булева выражения путем объединения несмежных областей.

$$\overline{A}BCD + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}\overline{D} = Y.$$

$\square$	$CD \square$	$\overline{C}\overline{D} \square$	$\overline{C}D \square$	$C\overline{D} \square$
$AB \square$	$\square$	$\square$	$\square$	$\square$
$\overline{A}B \square$	1	$\square$	1	$\square$
$\overline{A}\overline{B} \square$	1	$\square$	1	$\square$
$A\overline{B} \square$	$\square$	$\square$	$\square$	$\square$

$$\overline{A}CD + \overline{A}\overline{C}\overline{D} = Y.$$

**Пример 2** – Минимизация булева выражения путем объединения областей в квадрат.

$$\overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}\overline{C}\overline{D} = Y.$$

$\square$	$CD \square$	$\overline{C}\overline{D} \square$	$\overline{C}D \square$	$C\overline{D} \square$
$AB \square$	$\square$	$\square$	$\square$	$\square$
$\overline{A}B \square$	$\square$	1	1	$\square$
$\overline{A}\overline{B} \square$	$\square$	1	1	$\square$
$A\overline{B} \square$	$\square$	$\square$	$\square$	$\square$

$$\overline{A}\overline{C} = Y.$$

**Пример 3** – Минимизация булева выражения путем сворачивания карты в вертикальный цилиндр.

$$ABCD + ABC\bar{D} = Y.$$

	$CD$	$\bar{C}D$	$C\bar{D}$	$\bar{C}\bar{D}$
$AB$	1			1
$\bar{A}B$				
$\bar{A}\bar{B}$				
$A\bar{B}$				

$$ABC = Y.$$

**Пример 4** – Минимизация булева выражения путем сворачивания карты в горизонтальный цилиндр.

$$A\bar{B}\bar{C}D + A\bar{B}C\bar{D} = Y.$$

	$CD$	$\bar{C}D$	$C\bar{D}$	$\bar{C}\bar{D}$
$AB$		1		
$\bar{A}B$				
$\bar{A}\bar{B}$				
$A\bar{B}$		1		

$$A\bar{C}D = Y.$$

**Пример 5** – Минимизация булева выражения путем сворачивания карты в шар.

$$ABCD + ABC\bar{D} + \bar{A}BCD + \bar{A}BC\bar{D} = Y.$$

	$CD$	$\bar{C}D$	$C\bar{D}$	$\bar{C}\bar{D}$
$AB$	1			1
$\bar{A}B$				
$\bar{A}\bar{B}$				
$A\bar{B}$	1			1

$$AC = Y.$$

**Пример 6** – Минимизация булева выражения путем объединения линейной области.

$$\overline{A}BCD + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D = Y.$$

	$CD$	$\overline{C}D$	$\overline{C}\overline{D}$	$C\overline{D}$
$AB$				
$\overline{A}B$				
$\overline{A}\overline{B}$	1	1	1	1
$A\overline{B}$				

$$\overline{A}\overline{B} = Y.$$

$$\overline{A}BCD + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D} = Y.$$

	$CD$	$\overline{C}D$	$\overline{C}\overline{D}$	$C\overline{D}$
$AB$				
$\overline{A}B$				
$\overline{A}\overline{B}$	1	1	1	
$A\overline{B}$				

$$\overline{A}\overline{B}D + \overline{A}\overline{B}\overline{C} = Y.$$

**Пример 7** – Приведенная функция не минимизируется.

$$ABCD + \overline{A}B\overline{C}D + \overline{A}\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D} = Y.$$

	$CD$	$\overline{C}D$	$\overline{C}\overline{D}$	$C\overline{D}$
$AB$	1			
$\overline{A}B$		1		
$\overline{A}\overline{B}$			1	
$A\overline{B}$				1

$$ABCD + \overline{A}B\overline{C}D + \overline{A}\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D} = Y.$$

## 6.2 Порядок выполнения

Порядок выполнения следующий:

- запустить FluidSIM;
- создать новый документ;
- установить логические элементы;
- соединить элементы в соответствии с булевой функцией;
- составить карту Карно;
- упростить функцию;
- составить логическую схему в соответствии с упрощенной булевой

функцией;

- проверить работоспособность схем;
- выйти из FluidSIM.

### ***Контрольные вопросы***

- 1 Правила составления карт Карно.
- 2 Правила минимизации булевых функций с использованием карт Карно.
- 3 Правила объединения элементов в картах Карно.

## **7 Лабораторная работа № 7. Язык программирования Grafset**

**Цель работы:** получить навыки работы с языком программирования Grafset.

### ***7.1 Общие сведения***

Автоматические регуляторы, или программируемые логические контроллеры (ПЛК), являются широко распространенными средствами автоматизации в составе локальных и распределенных систем контроля и управления.

Термином ПЛК обозначают устройства, осуществляющие преобразование, обработку, хранение информации и выработку команд управления или управляющих регулирующих воздействий, реализованные на базе микропроцессорной техники и являющиеся, по сути, специализированными управляющими вычислительными комплексами для работы в локальных и распределенных системах управления в реальном масштабе времени.

Требования, предъявляемые к ПЛК, разнообразны, т. к. ПЛК используются для всевозможных видов деятельности (рабочими, технологами, инженерами). ПЛК может представлять собой либо целый, неделимый продукт, поставляемый одним производителем, либо несколько продуктов – составных частей, поставляемых одним или разными производителями. По функциональному назначению в ПЛК можно выделить следующие основные части: процессор, осуществляющий прием, обработку и выдачу информации; устройство сопряжения процессора с объектом (УСО), устройство сопряжения процессора с человеком-оператором и программное обеспечение (ПО).

Языки программирования ПЛК во многом определяются конкретными приложениями, а также требованиями современных сложных распределенных систем управления.

Историческое развитие ПЛК привело к возникновению разных решений, связанных с конкретным производителем и в той или иной степени удовлетворяющих общим стандартам (DIN 19239, 19237, 40719 Part 6 и VDI 2880). Однако элементы языков, несмотря на похожесть, у разных производителей отличались, существовали машинно-зависимые особенности языков, которые приходилось учитывать.

Для решения этой проблемы Международная электротехническая комиссия (МЭК) – всемирная организация, с 1906 г. разрабатывающая международные стандарты в области электротехники, электроники, телекоммуникаций и средств информационных технологий – основала в 1975 г. рабочий комитет 65А для разработки стандарта для ПЛК, который в 1993 г. принял на ПЛК международный стандарт IEC 1131 (в настоящее время действует IEC 61131-3), третья часть которого имеет дело с языками для ПЛК. В него вошли пять языков:

- 1) LD (Ladder Diagram) – вариант класса языков релейно-контактных схем;
- 2) FBD (Functional Block Diagram) – аналогичен функциональной схеме электронного устройства с использованием логических элементов;
- 3) GRAFCET (SFC) (Sequential Function Chart) – язык диаграммного типа, аналогичный блок-схемам алгоритмов;
- 4) ST (Structured Text) – текстовый высокоуровневый язык общего назначения, по синтаксису напоминает Паскаль;
- 5) IL (Instruction List) – текстовый язык низкого уровня, по синтаксису напоминает Ассемблер.

Язык GRAFCET (SFC) предложен в 1979 г. (его прототипом является язык Grafset компании Telemecanique, стандарт IEC 848) как графический язык, предназначенный для описания функций последовательных операций и позволяющий ясное и однозначное определение желаемого поведения системы управления. Отличается строгим математическим определением (каждое операционное состояние может быть декодировано с достаточно малым анализом). Удобен для описания как последовательных процессов, так и пакетных или параллельных процессов, легко комбинируется с другими языками (язык спецификаций). Обеспечивает улучшение понимания оборудования при формировании модели приложения. Обладает развитыми механизмами синхронизации. Использует простые динамические правила.

Диагностика языка GRAFCET (SFC) позволяет обнаружить ошибки проекта до того, как приложение тестируется в режиме Online. Интерактивно анимируемый язык GRAFCET (SFC) также служит для облегчения диагностики оборудования и ошибок системы управления и после ввода в действие. Использование языка GRAFCET (SFC) как диагностического инструмента может уменьшать среднее время устранения таких неполадок. GRAFCET (SFC) широко принят как стандарт в Европе.

Язык GRAFCET (SFC) использует следующие основные структурные элементы для создания программы: шаг (Step), переход (Transition), действие (Action). Меню элементов представлено на рисунке 7.1.

Шаг имеет несколько типов. Для выбора типа шага необходимо установить шаг на лист и дважды щелкнуть по нему левой клавишей мыши. Откроется окно (рисунок 7.2).

Название шага может содержать следующие символы: «0-9», «a-z», «A-Z» и подчеркивание «\_».

Можно выбрать один из следующих семи типов ступеней: простой шаг (simple step), начальный шаг (initial step), макро-шаг (macro-step), ввод макроса



(macro input), вывод макроса (macro output), шаг включения (enclosing step) и начальный шаг включения (initial enclosing step).

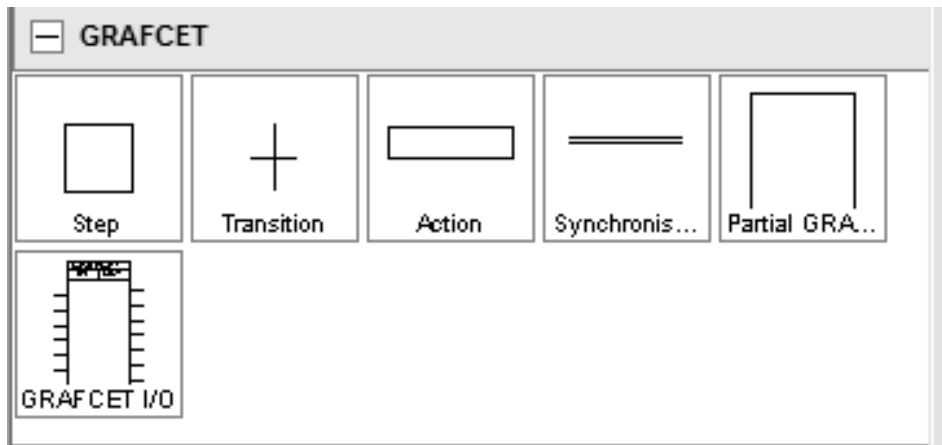


Рисунок 7.1 – Меню элементов языка GRAFCET (SFC)

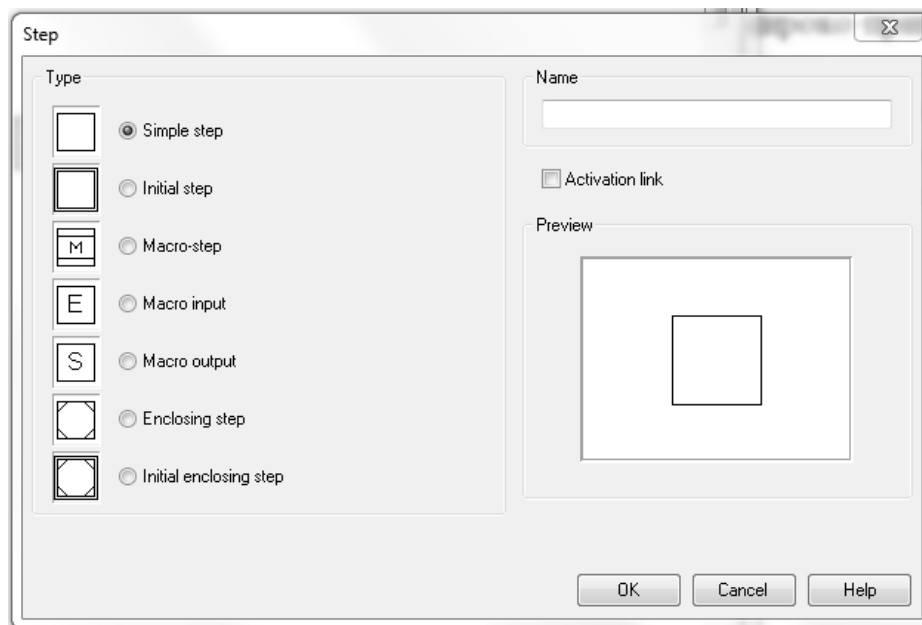


Рисунок 7.2 – Выбор типа шага

Кроме того, можно указать шаг активации (activation link).

Для задания параметров перехода необходимо установить шаг на лист и дважды щелкнуть по нему левой клавишей мыши. Откроется окно (рисунок 7.3).

Можно указать название перехода, которое показано слева от перехода в скобках.

Ввод условия перехода поддерживается кнопками для специальных символов (AND, OR, NOT, падающий фронт, передний фронт, задержка). Через кнопку «Переменная (Variable)» можно выбрать существующую переменную GRAFCET из списка. В качестве альтернативы можно показать описательный текст. Для этого нужно выбрать опцию «Отображать описание вместо формулы (Display description instead of formula)».

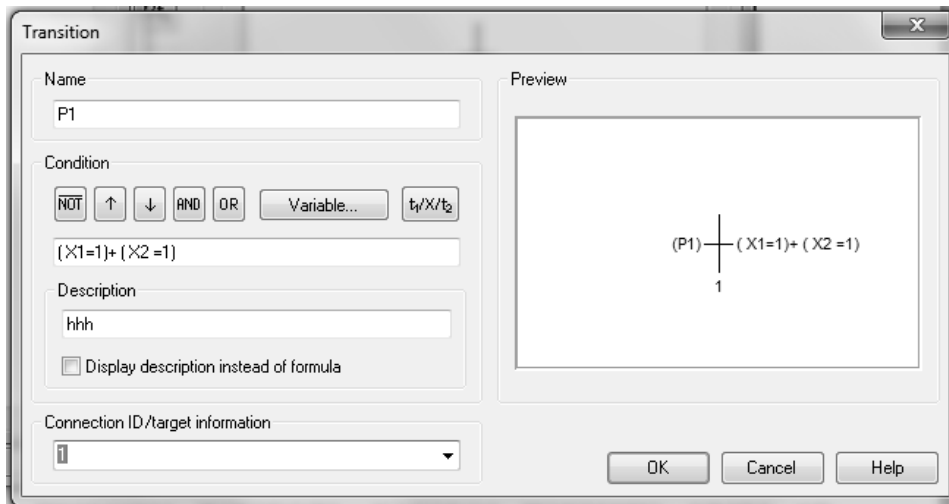


Рисунок 7.3 – Параметры перехода

В поле «Идентификация соединения / целевая информация (Connection ID/target information)» можно ввести шаг, который ссылается на вывод перехода, не рисуя соединительную линию. Можно выбрать существующий шаг из списка.

Для выбора типа действия необходимо установить действие на лист и дважды щелкнуть по нему левой клавишей мыши. Откроется окно (рисунок 7.4).

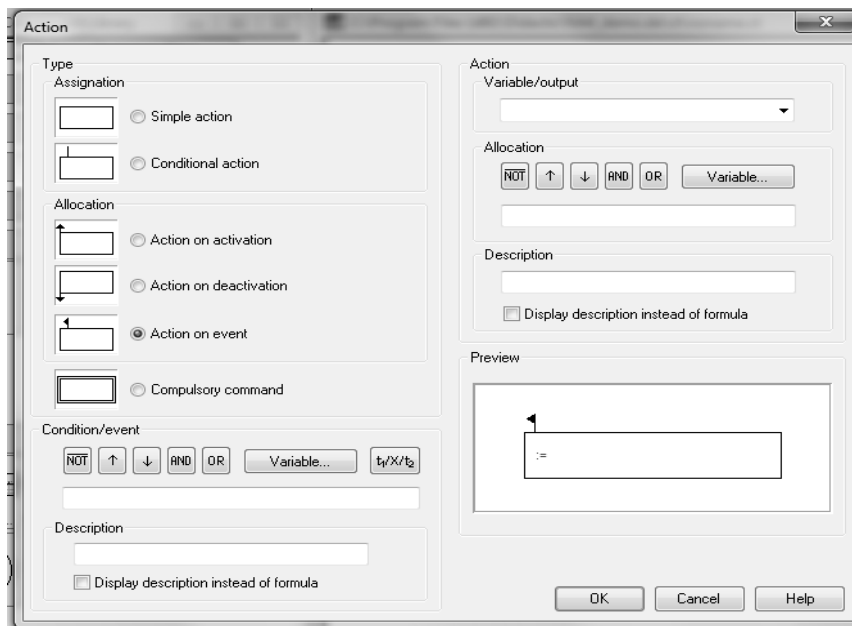


Рисунок 7.4 – Выбор типа действия

Существует три типа действий: назначения, распределения и обязательные команды. Вид окна «Действие» меняется в зависимости от выбранного типа действия.

Для назначений и распределений можно ввести переменную или выход, значение которых изменяется действием. Имя переменной может содержать следующие символы: «0-9», «a-z», «A-Z» и подчеркивание «\_».

Для «условного действия (conditional action)» или «действия по событию (action on event)» можно ввести условие, которое должно быть выполнено до того, как действие будет выполнено. Ввод условия поддерживается кнопками для специальных символов (AND, OR, NOT, падающий фронт, передний фронт, задержка). Через переменную можно выбрать существующую переменную GRAFCET из списка. В качестве альтернативы можно показать описательный текст. Для этого нужно выбрать опцию «Отображать описание вместо формулы (Display description instead of formula)».

Для распределения «действие при активации (action on activation)», «действие при деактивации (action on deactivation)» и «действие на событие (action on event)» можно ввести любой термин, значение которого должно быть присвоено переменной действия. Ввод термина поддерживается кнопками для специальных символов (AND, OR, NOT, падающий фронт, передний фронт). Через переменную можно выбрать существующую переменную GRAFCET из списка. В качестве альтернативы можно показать описательный текст. Для этого нужно выбрать опцию «Отображать описание вместо формулы (Display description instead of formula)».

Для «принудительной команды (compulsory command)» можно напрямую ввести имя подпрограммы GRAFCET или выбрать существующий подпрограмму GRAFCET из списка. Также можно ввести соответствующие шаги напрямую или выбрать их из списка существующих шагов. Разделить имена шагов запятыми. Можно выбрать специальные команды «\*» и «INIT» с помощью соответствующих кнопок.

Компонент ввода / вывода GRAFCET (рисунок 7.5) используется для связывания переменных GRAFCET с электрической частью схемы. Можно ввести восемь входных переменных GRAFCET и восемь выходных переменных GRAFCET в компонент ввода / вывода GRAFCET. Переменные действия являются выходными. Входы могут быть распределениями и условиями действий и переходов.

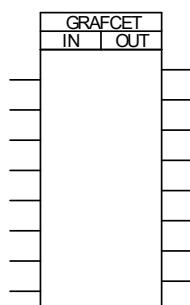


Рисунок 7.5 – Компонент ввода/вывода (GRAFCET I/O)

Если потенциал создается как вход компонента GRAFCET I / O, соответствующая переменная устанавливается в «1». Если выходная переменная имеет значение, отличное от «0», потенциал на 24 В создается на соответствующем выходе компонента GRAFCET I / O.

При двойном щелчке по данному компоненту открывается окно переменных (рисунок 7.6).

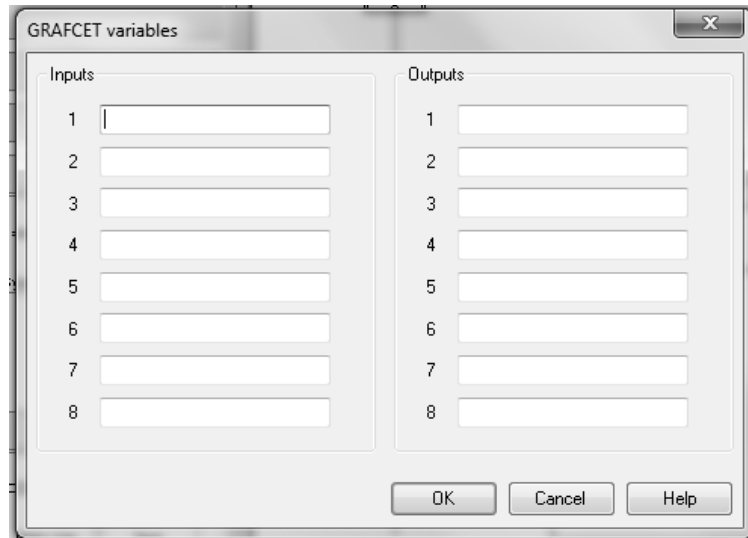


Рисунок 7.6 – Окно переменных

## 7.2 Графическое представление структур последовательности

Существует три основных типа структур последовательности, которые генерируются путем объединения шагов и переходов.

**Линейная последовательность.** В линейной последовательности за каждым шагом следует один последующий переход, который активируется с помощью одного шага в последовательности последовательно, исключая первый шаг. И каждый шаг соединяется только с одним последующим переходом, за исключением конечного шага. Линейная последовательность представлена на рисунке 7.7.

**Альтернативное разветвление.** Для альтернативного ветвления имеется несколько переходов. Шаг будет активирован, когда будет выполнено первое условие перехода последовательности. В связи с тем, что в случае альтернативного ветвления необходимо выбрать одну последовательность, различные условия перехода должны быть взаимоисключающими. Альтернативное разветвление представлено на рисунке 7.8.

**Параллельное разветвление.** При параллельном ветвлении некоторые последовательности будут активированы одновременно при выполнении одного условия перехода. Но каждая ветка работает независимо. Последовательности объединяются обратно в основную первичную последовательность. Шаг под второй двойной линией не будет выполняться до тех пор, пока все параллельные последовательности не будут полностью обработаны и не будут выполнены условия перехода. Параллельное разветвление представлено на рисунке 7.9.

В качестве примера использования GRAFCET рассмотрим систему, в которой гидроцилиндр совершает возвратно-поступательное движение (рисунок 7.10).

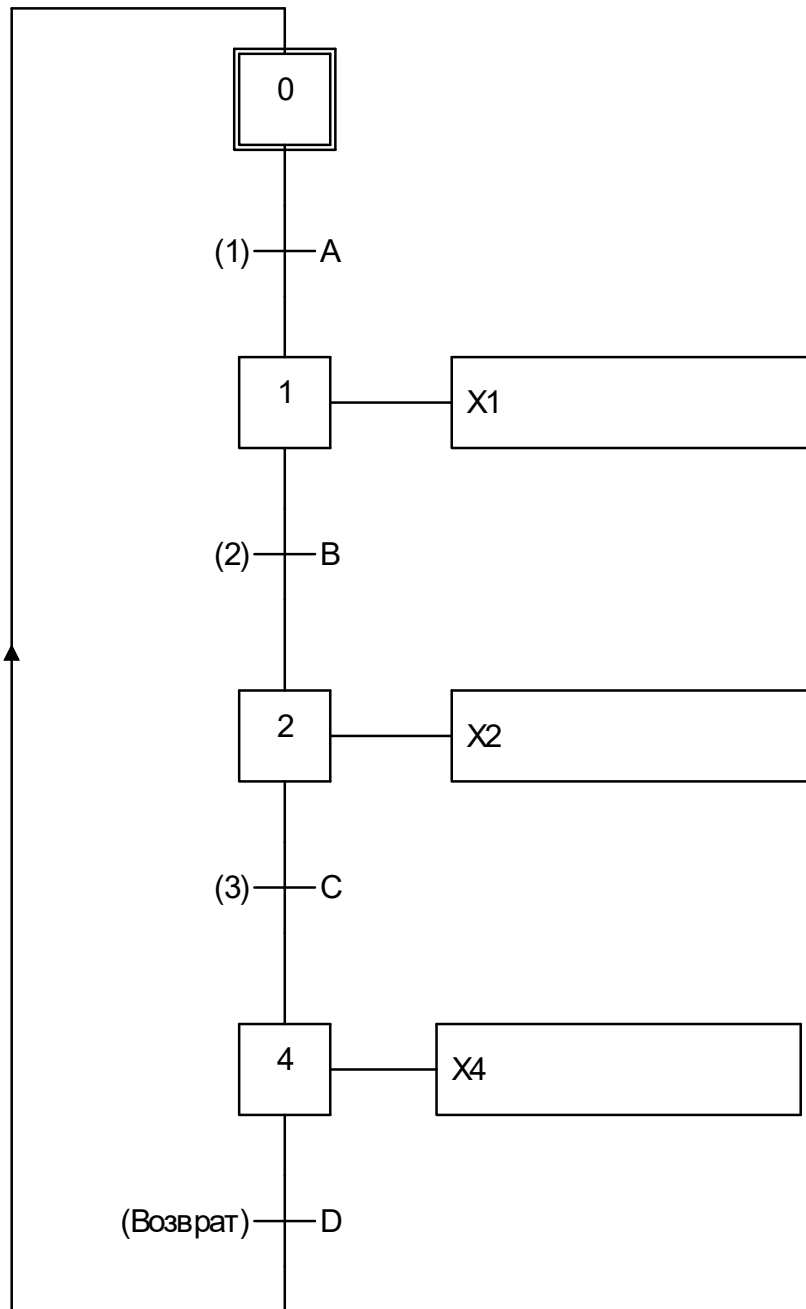


Рисунок 7.7 – Линейная последовательность

При включении кнопки KN гидроцилиндр находится в крайнем левом положении. Контакты датчика 1B1 замкнуты. Переход 10 на диаграмме GRAFCET производит проверку. Если на компонент ввода/вывода подается сигнал на вход  $i_1$ , то выполняется шаг 1. Сигнал с O1 подается на соленоид 1M1 распределителя. Распределитель перемещается вправо, жидкость поступает в поршневую полость гидроцилиндра и он движется вправо до срабатывания датчика 1B2. Контакты датчика 1B2 замыкаются.

Переход 11 на диаграмме GRAFCET производит проверку. Если на компонент ввода/вывода подается сигнал на вход  $i_2$ , то выполняется шаг 2. Сигнал с O2 подается на соленоид 1M2 распределителя. Распределитель перемещается

влево, жидкость поступает в штоковую полость гидроцилиндра и он движется влево до срабатывания датчика 1В1. Далее процесс повторяется.

Процесс продолжается до тех пор пока кнопка КN включена.

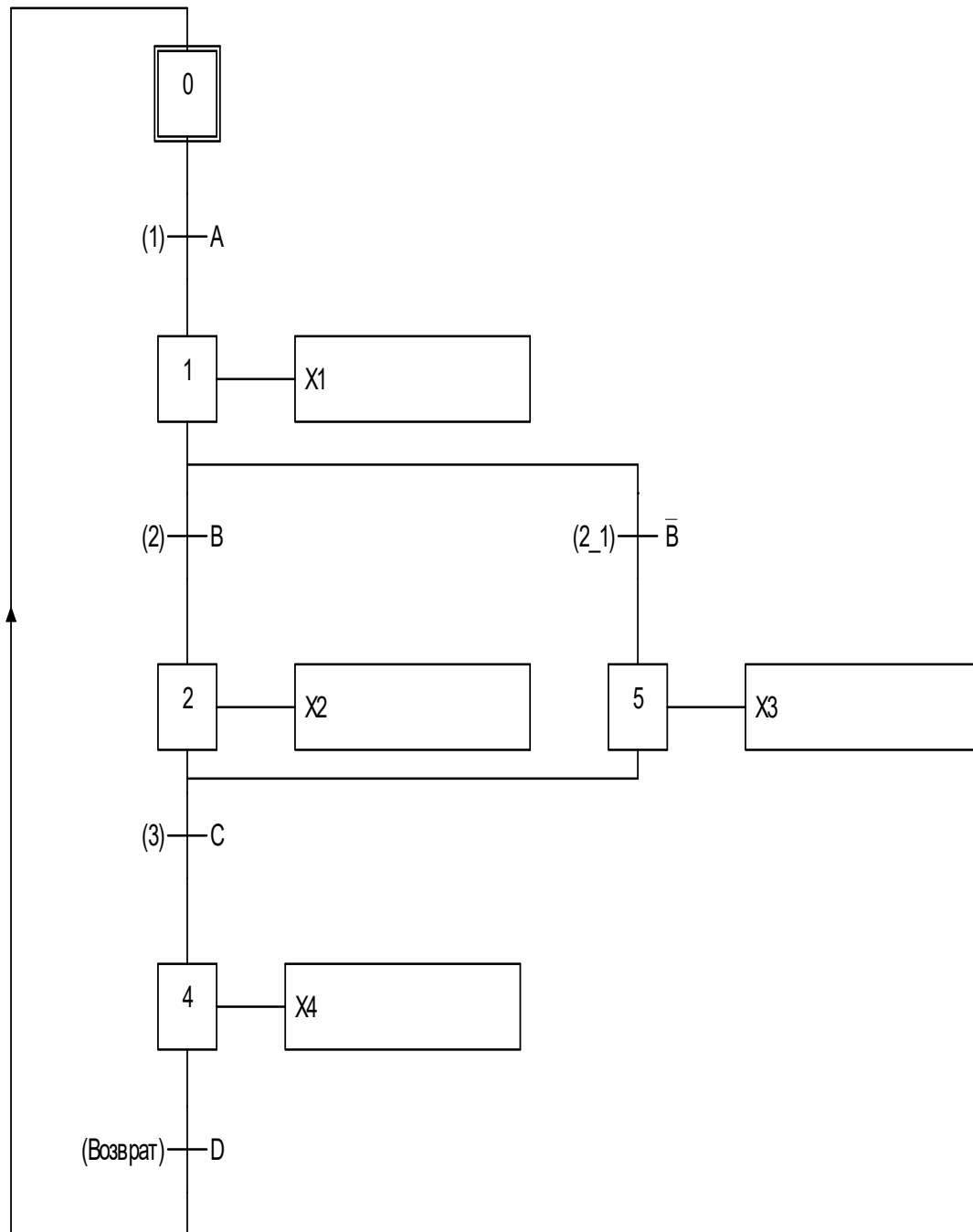


Рисунок 7.8 – Альтернативное разветвление

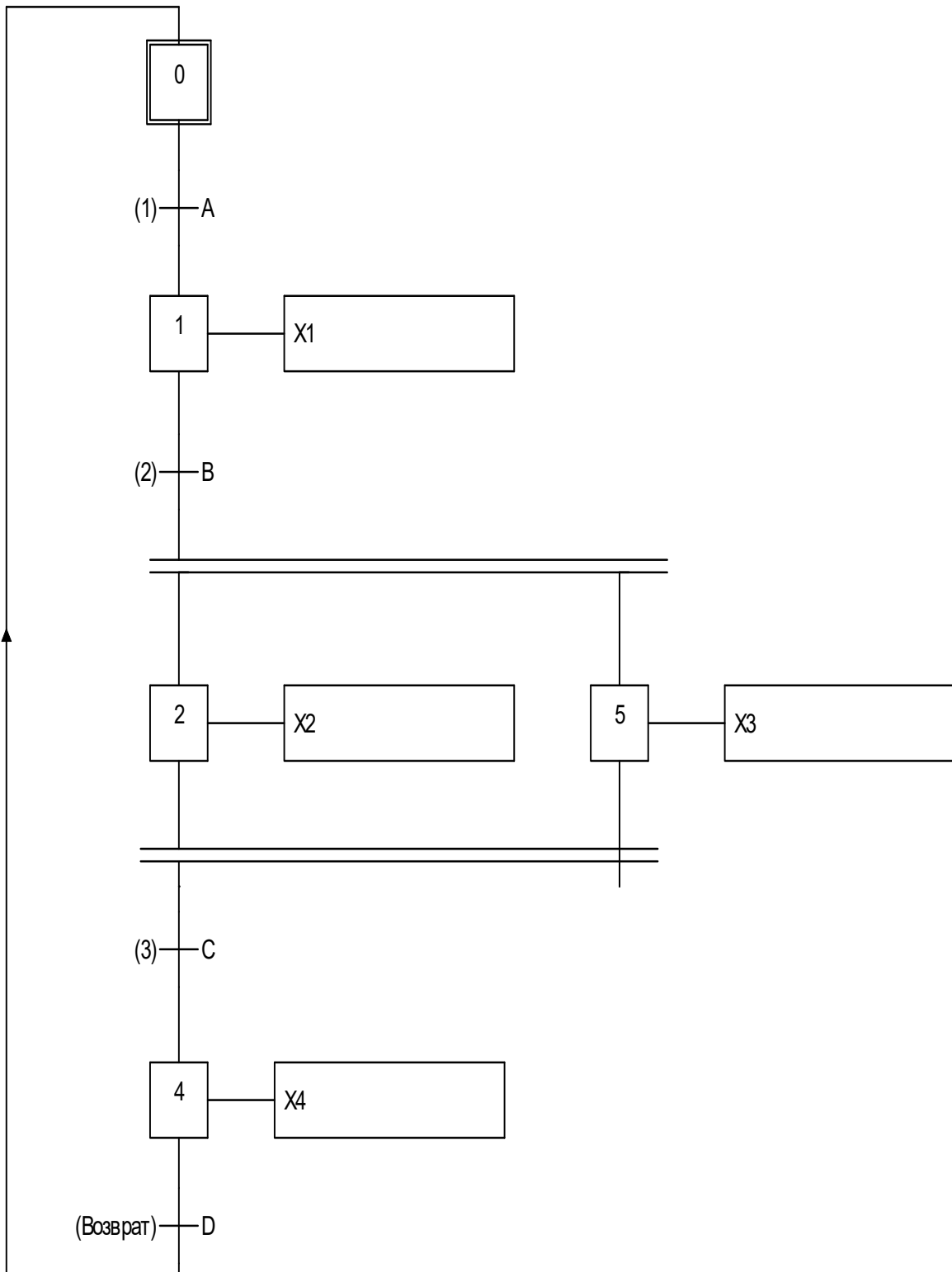


Рисунок 7.9 – Параллельное разветвление

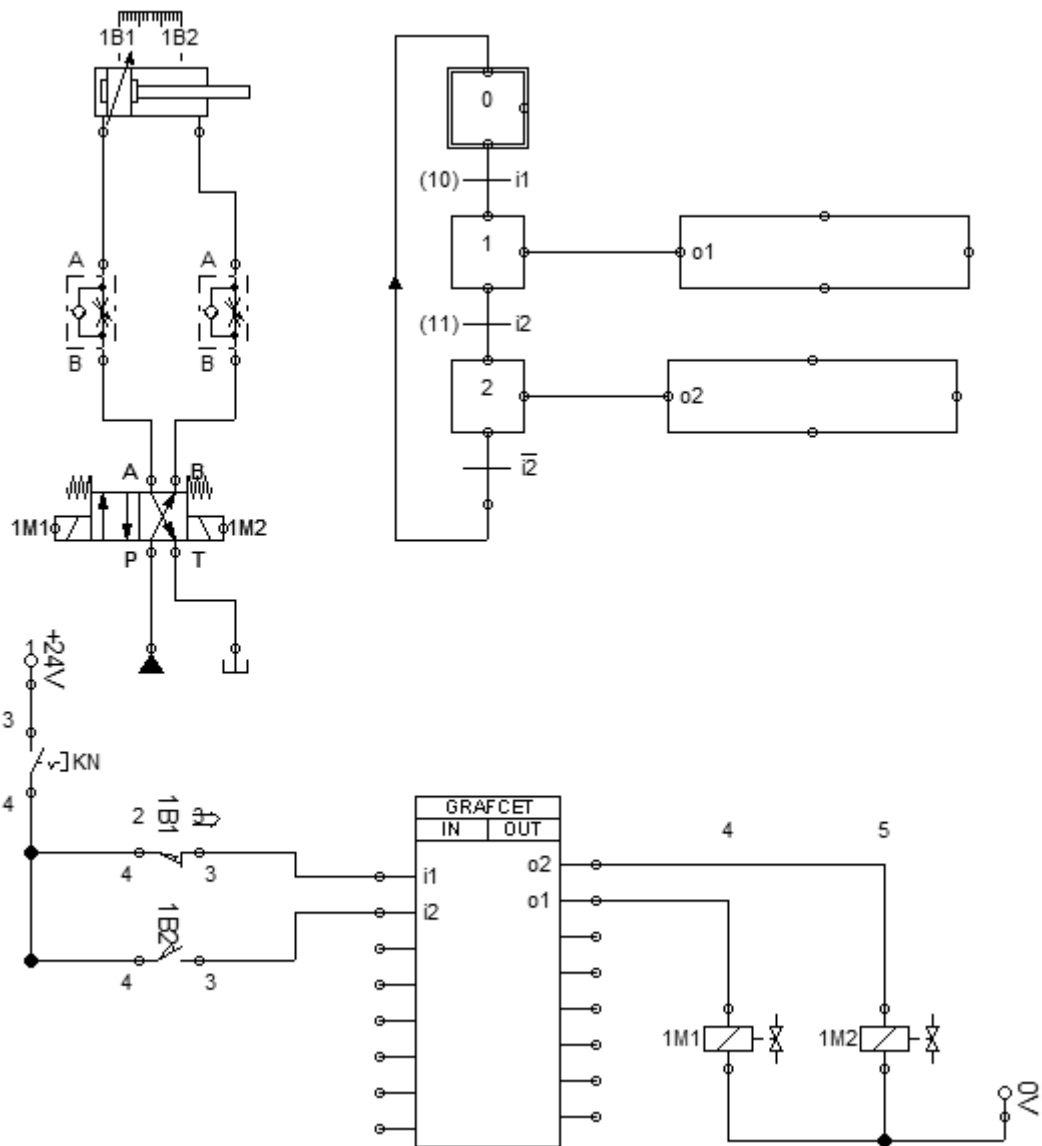


Рисунок 7.10 – Реализация возвратно-поступательного движения гидроцилиндра с использованием языка GRAFCET

### 7.3 Порядок выполнения

Порядок выполнения следующий:

- запустить FluidSIM;
- создать новый документ;
- составить гидравлическую схему с заданными параметрами;
- проверить работоспособность гидросхемы;
- составить описание работы системы;
- составить программу на языке Grafset;
- установить связь между программой на языке Grafset и гидравлической схемой;
- проверить работоспособность системы;
- выйти из FluidSIM.



### ***Контрольные вопросы***

- 1 Назначение языка Grafset.
- 2 Основные элементы языка Grafset.
- 3 Правила установления связи между программой на языке Grafset и гидравлической схемой.

### **Список литературы**

- 1 **Комиссаров, Ю. А.** Общая электротехника и электроника : учебник / Ю. А. Комиссаров, Г. И. Бабокин; под ред. П. Д. Саркисова. – 2-е изд., испр. и доп. – Москва : ИНФРА-М, 2017. – 479 с.
- 2 **Сазанов, И. И.** Гидравлика : учебник / И. И. Сазанов, А. Г. Схиртладзе, В. И. Иванов. – Москва : КУРС; ИНФРА-М, 2017. – 320 с.
- 3 **Певзнер, Л. Д.** Практикум по математическим основам теории систем : учебное пособие / Л. Д. Певзнер. – Санкт-Петербург; Москва ; Краснодар : Лань, 2013. – 400 с. : ил.
- 4 **Вавилов, А. В.** Проектирование гидроприводов строительных и дорожных машин: учебно-методическое пособие для студентов специальности 1-36 11 01 «Подъемно-транспортные, строительные, дорожные машины и оборудование» / А. В. Вавилов. – Минск : БНТУ, 2012. – 74 с.