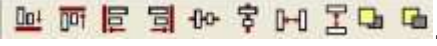




Цель работы:

1. Закрепление навыков работы с интерфейсом Infinity HMI.
2. Обязательное применение изученных инструментов графики и расположения для эффективной работы при создании объектов.
3. Применение для этих целей ранее не упоминавшихся инструментов, назначение которых интуитивно понятно: .
4. Работа с локальными переменными.

Упражнение 1: Создание графической мнемосхемы учебного стенда «Булевы функции».

1. Запустите **InfinityHMI** и создайте в личной директории новый файл **Фамилия3.xml**.
2. В меню **Сервис** выберите пункт **Параметры** и вкладку Редактор. Настройте данное окно как показано на Рис. 1. Данным действием вы включите сетку для удобства выполнения работы. Цвет сетки поставьте, например, черный.

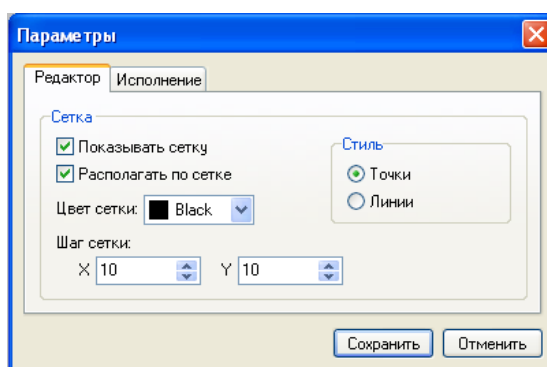


Рис. 1

3. В главном меню **Вид** отметьте **Цветовую палитру** (Рис. 2). Цвет палитры выберите темно – серый. После этого все объекты типа **Прямоугольник** и **Эллипс/Круг** будут иметь темно – серую заливку и палитру, если она мешает можно убрать.

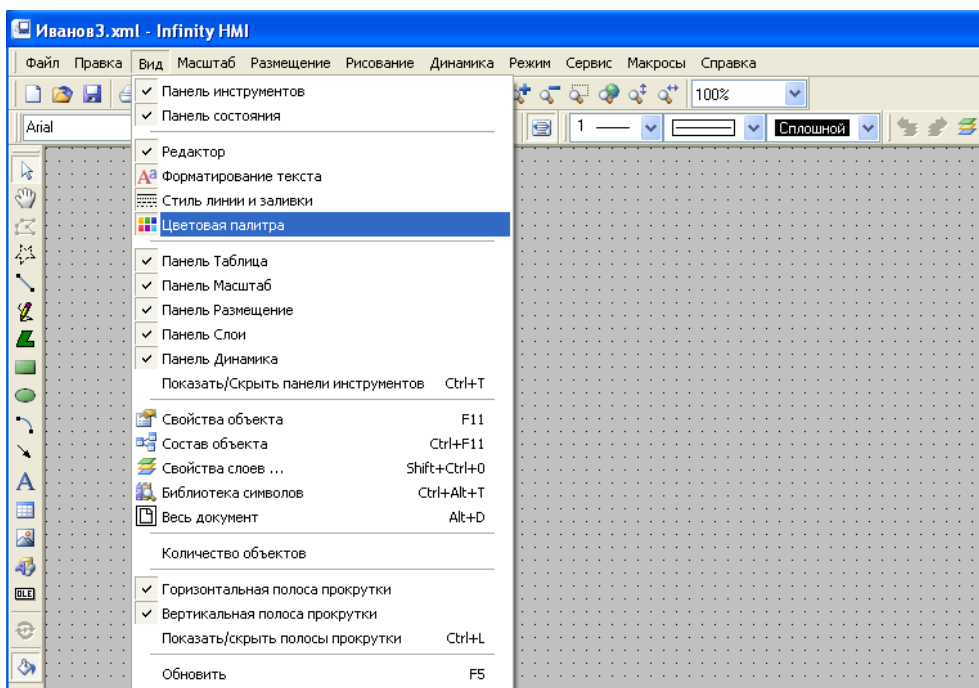


Рис. 2



4. Нарисуйте объект Логический элемент И. Размеры приблизительно, визуалью около 20мм × 40мм (Рис. 3).

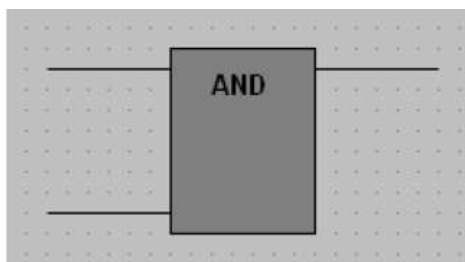


Рис. 3

5. Создайте объект **Кнопка** и настройте **Свойства объекта**, как на Рис. 4

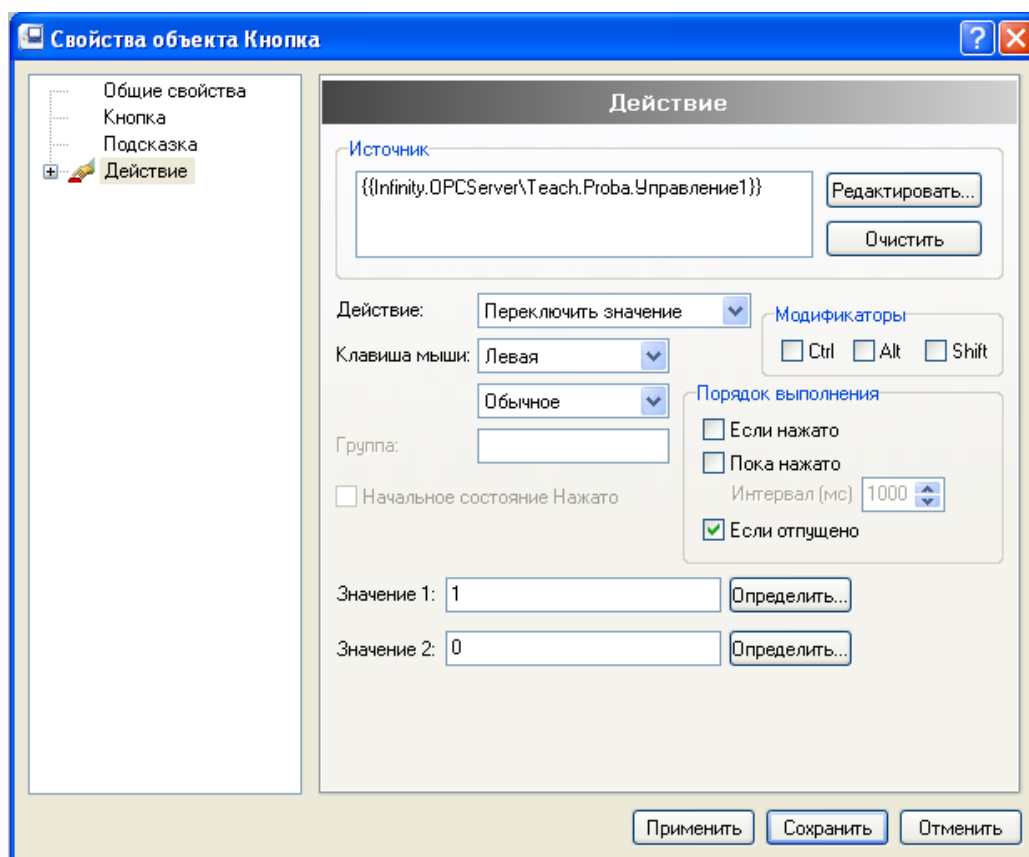


Рис. 4

6. Повторите действия п.5 для второго сигнала **Управление2**. Внешний вид экранной формы показан на Рис. 5. Для эффективного размещения объектов применяйте соответствующие инструменты:

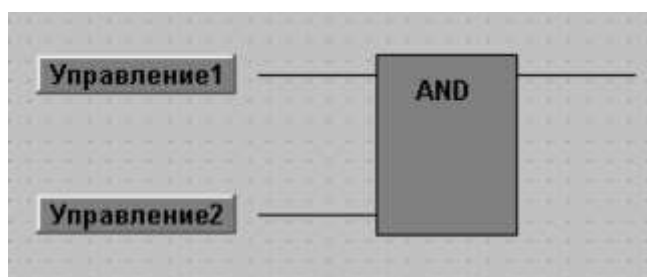
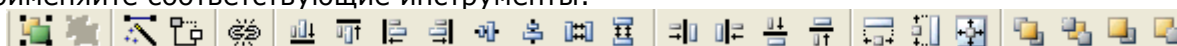


Рис. 5



7. На Рис. 5 есть семь объектов, которые в избежание смещения относительно друг друга можно (и нужно!) сгруппировать. Доступ к любому объекту в сгруппированном символе осуществляется вызовом на выделенном символе из контекстного меню (правая кнопка мыши) опции **Свойства объекта**. После этого появится окно **Свойства объекта**. В свойствах объекта в закладке **Дочерние объекты** (Рис. 6) можно выбрать любой объект, который находится в составе данного объекта, и изменить его свойства. После того, как необходимые изменения были произведены нужно нажать на кнопку **Назад** (Рис. 7).

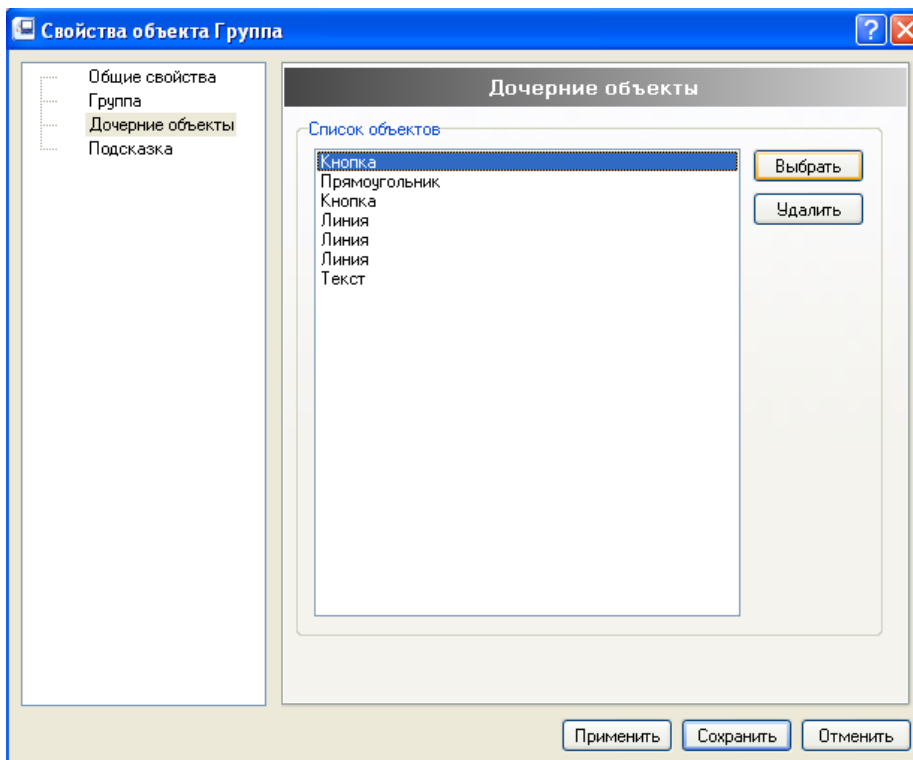


Рис. 6

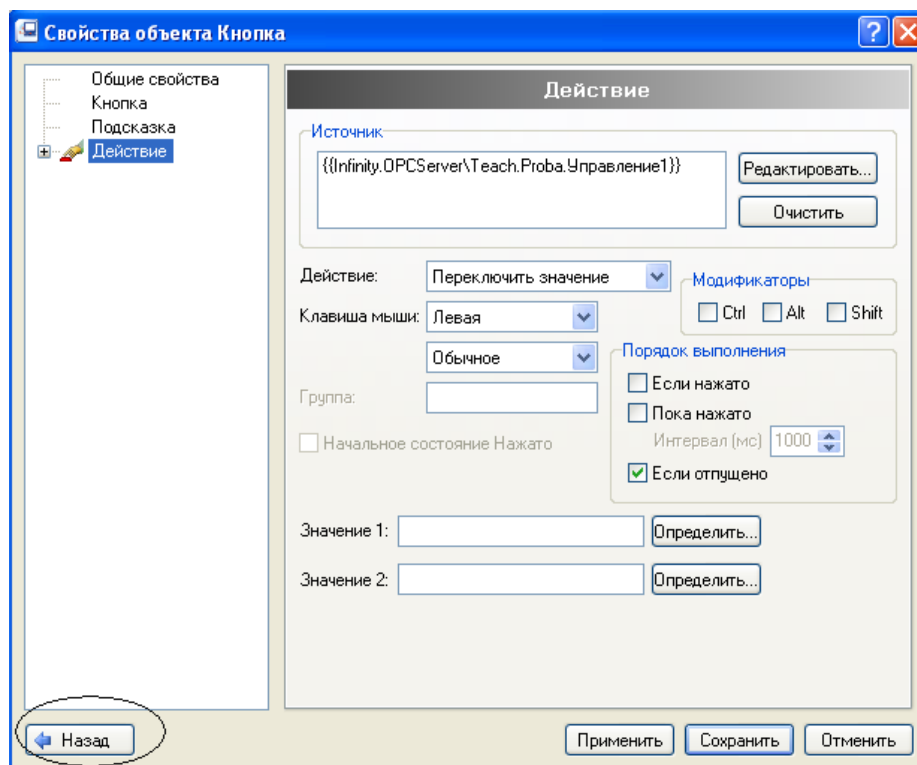


Рис. 7



8. Представьте, что битовый сигнал **Управление1** будет приходиться на верхний по схеме вход логического элемента, а **Управление2** на нижний. Битовый сигнал **Состояние1** – это выходной сигнал (выход элемента). Примените к этим линиям динамику так, чтобы при единичном значении сигналов соответствующие линии становились зелеными, а при нулевых значениях красными. В режиме **Режим-Исполнение** все линии должны быть черными. Сейчас и далее группируйте объекты в символы, а доступ к объектам в символах осуществляйте опцией **Свойства объекта**.

9. Примените к объекту **Прямоугольник** динамику **Динамическое действие**, как показано на Рис. 8. В поле **Значение (пока нажато)** введите выражение логического умножения для сигналов **Управление1** и **Управление2**. Для ввода выражений служит **Изменение источника данных**, в котором есть кнопки ввода функций и, главное вызов тегов. Настроив **Свойства объекта**, помните, что вы определяете условие, по которому выражение поля **Значение** будет записываться в **Источник данных**. **Свойства объекта** на Рис. 8 настроены таким образом, что выражение будет записываться в **Источник данных** один раз в 50 мс. Сразу после перевода мнемосхемы в режим **Режим-Разработка**.

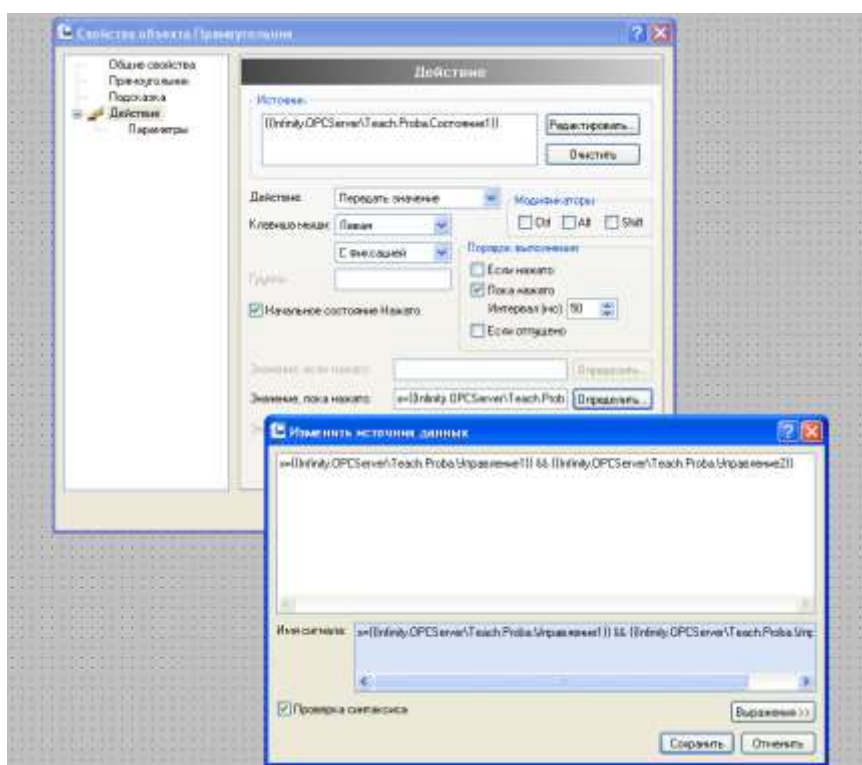


Рис. 8

10. Закройте **Свойства объекта** и проверьте работу мнемосхемы. Любой ноль на входе даст ноль на выходе.

11.



Еще раз напоминаем: перед выходом из режима Проект – Старт приводите сигналы в исходное (нулевое) состояние.

12. Создайте объекты логических элементов: **И, ИЛИ, И-НЕ, ИЛИ-НЕ**. Рис. 9

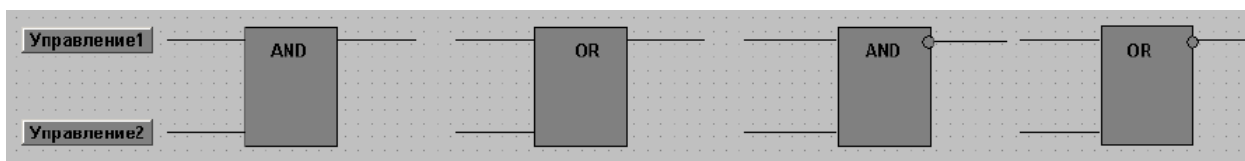


Рис. 9



13. Входные сигналы для всех прежние: **Управление1, Управление2**, а выходные – **Состояние2, Состояние3, Состояние4**, соответственно. К выходным сигналам примените динамику **Динамический Цвет** по аналогии с выходом созданного объекта **Логический элемент И**. Логические выражения, естественно, должны быть для каждого логического элемента свои. **Необходимо заметить, что логические отрицание в Infinity NMI обозначается как восклицательный знак ! и имеет высший приоритет, там, где это требуется, применяйте простые () скобки.** Очевидно, что это замечание будет актуально во время ввода выражений для функций И-НЕ, ИЛИ-НЕ, отсутствие скобок перед инверсией выражения для этих функций $x=!(\&\&)$. **Также следует учитывать, что при написании формул каждую операцию необходимо заключать в простые скобки.**

14. Расположите объекты в рабочей области как показано на Рис. 10 и сгруппируйте их в один символ.

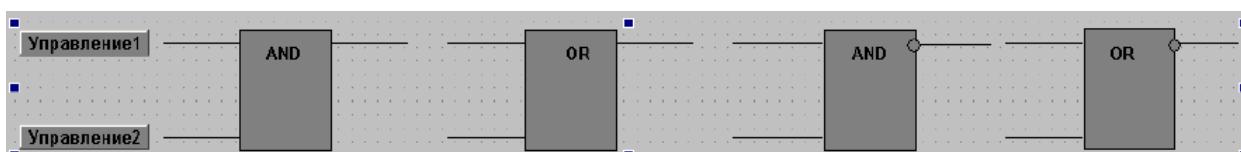


Рис. 10

15. Нарисуйте «основание» стенда, цвет заливки: **синий**, объем: **выпуклый**.

16. С помощью инструмента **Текст А** создайте название стенда. Вид экранной формы показан на Рис. 11.



Рис. 11

17. Проверьте экранную форму в режиме **Режим-Исполнение**.

18. С помощью необходимых инструментов создайте объект Таблица, где будут отображаться значение всех сигналов. Общий вид может быть таким, как на Рис. 12.

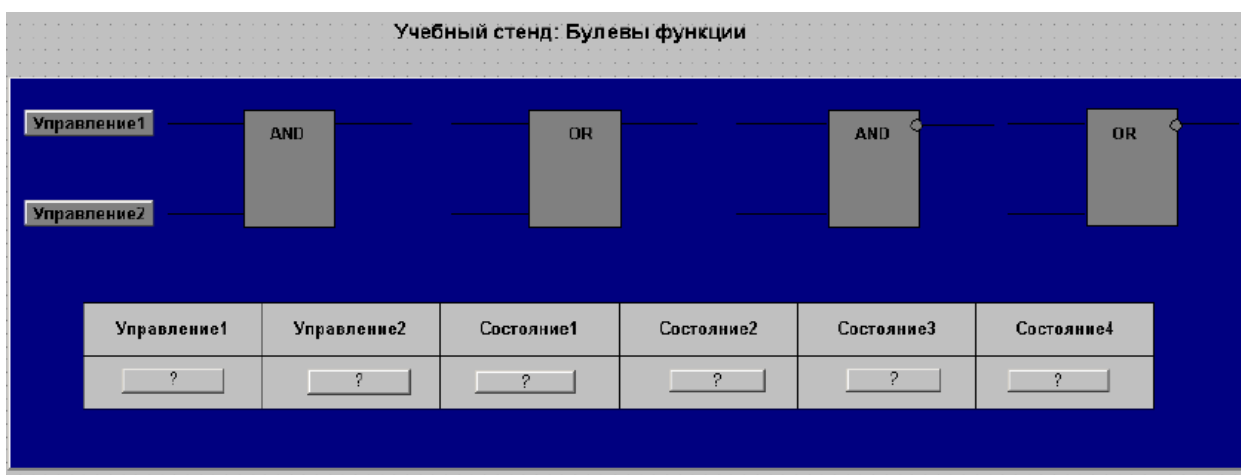


Рис. 12

19. Сохраните экранную форму.



Упражнение 2: Создание графической мнемосхемы «RS – триггер с прямыми входами на элементах ИЛИ-НЕ».



Создание мнемосхем цифровых элементов в лабораторной работе не преследует цели изучения логики их работы. Задача состоит в динамизации этих объектов, которая делает доступной для визуального восприятия поведение задействованных в мнемосхеме сигналов. Кроме того, поведение сигналов зависит от применения к ним выражений. Можно просто сложить два сигнала и получить функцию ИЛИ, а можно построить, например, мнемосхему триггера – элемента, широко применяемого в системах автоматики. Простой RS – триггер или триггер – защелку можно построить на основе двух элементов ИЛИ-НЕ с взаимными перекрестными связями (см. Рис. 13). Функция, описывающая работу такого триггера,

выглядит так: $R + \bar{Q}$, а $\bar{Q} = \overline{S + Q}$.

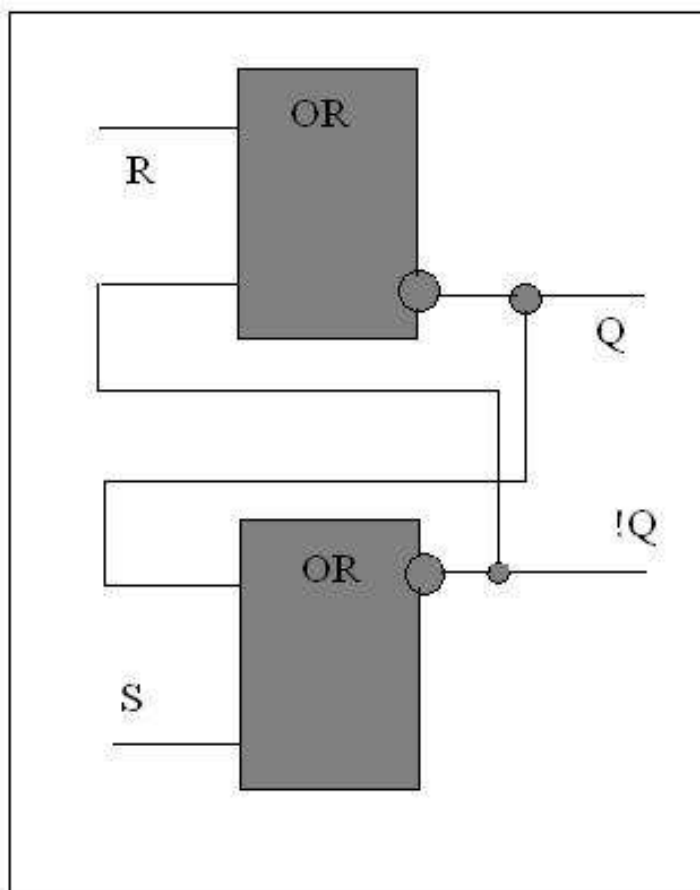


Рис. 13



В верхнем элементе происходит логическое сложение сигнала R с сигналом Q и инверсия этой суммы, поскольку элемент ИЛИ-НЕ. Работа нижнего элемента идентична, сигналы S и Q, соответственно. Активным входным уровнем является логическая единица.



1. Создайте новый файл **Фамилия4.xml**. Включите сетку и нарисуйте схему, подобную, показанной на Рис. 13. Для этого создайте прямоугольник, цвет заливки – темно – серый, надпись внутри – OR(ИЛИ). Элемент **Круг** (заливка темно - серая) обозначает функцию логического отрицания, расположите его, как показано на Рис. 13 (верхний элемент).

2. Сгруппируйте три элемента в символ.

3. Сделайте копию символа и разместите оба символа друг над другом, с помощью панели инструментов **Расположение** добейтесь их выравнивания (например, по левому краю).

4. Выделите нижний символ и в вызванном контекстном меню, выберите опцию **Свойства объекта**.

5. Переместите элемент **Круг** вверх (пусть выходы логических элементов будут симметричны – внешний вид мнемосхемы имеет значение!) и выйдите из режима **Свойства объекта**. Нарисуйте перекрестные связи. Для эффективного выполнения этой операции воспользуйтесь описанной ниже последовательностью.

6. Выберите инструмент **Линия** и совместите указатель мыши с левой стороной нижнего символа строго напротив элемента **Круг**.

7. Нажмите левую клавишу мыши и ведите линию на 15-20 мм. влево, следя за тем, чтобы не было переломов линии. После того, как линия нужной длины нарисована, отпустите и снова нажмите левую клавишу мыши. Ведите линию вверх, повторяйте эти действия пока ломанная не замкнется на элементе **Круг** верхнего символа. Выделите все отрезки ломанной и сгруппируйте в символ.

8. Прodelайте тоже самое для второй перекрестной связи.

9. Добавьте элемент **Линия** для входов и выходов.

10. Нарисуйте точки пересечения (серая заливка), как показано на Рис. 13.

11. Выделите символ первой ломанной, точку ее пересечения с выходом и сам выход, затем сгруппируйте их в один символ.

12. Также прodelайте для второй ломанной.

13. Выберите объект **Кнопка**, название – **Сброс**, настройка динамики **Действие**, как на Рис. 14. В качестве сигнала сброса будет использован сигнал **Управление1**.

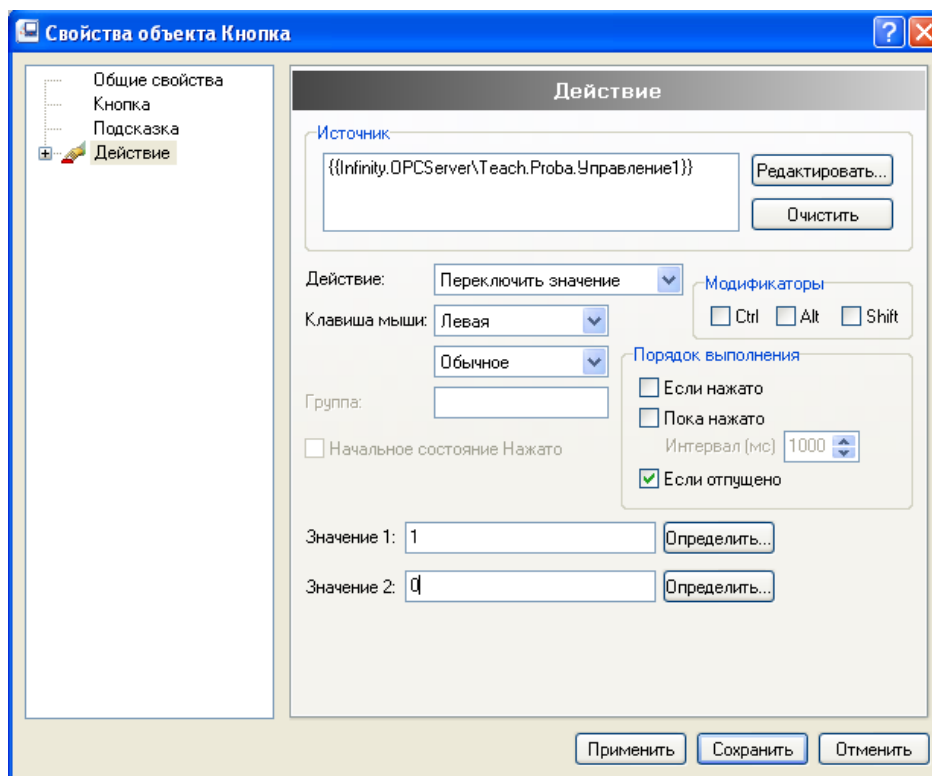


Рис. 14



14. Расположите кнопку **Сброс** возле соответствующего входа верхнего элемента ИЛИ-НЕ.
15. Примените к линии, обозначающей этот вход, динамику **Цвет** так, чтобы при значении сигнала **Управление1** – единица, линия становилась зеленой, иначе красной.
16. Создайте второй объект **Кнопка**, название **Запись**, сигнал **Управление2**. Настройка вкладки динамики аналогична. Расположите кнопку **Запись** возле соответствующего входа нижнего элемента.
17. Примените к линии, обозначающей этот вход, динамику **Цвет** так, чтобы при значении сигнала **Управление2** – единица, линия становилась зеленой, иначе – красной.
18. Выделите сгруппированный символ, куда входит выход **Q**.
19. Примените к этому символу динамику **Цвет**: выходным сигналом **Q** будет сигнал **Состояние1**. Настройка динамики показана на Рис. 15. Цветовая политика та же: красный цвет – нулевое значение сигнала, зеленый – единичное. В режим **Режим-Исполнение** ломаная линия, выход **Q** и контур точки их соединения будут расцветчиваться в соответствии с выбранной цветовой политикой. В режиме **Разработка** все элементы должны быть нейтрального цвета (мы выбрали серый цвет для объектов и черный для линий).

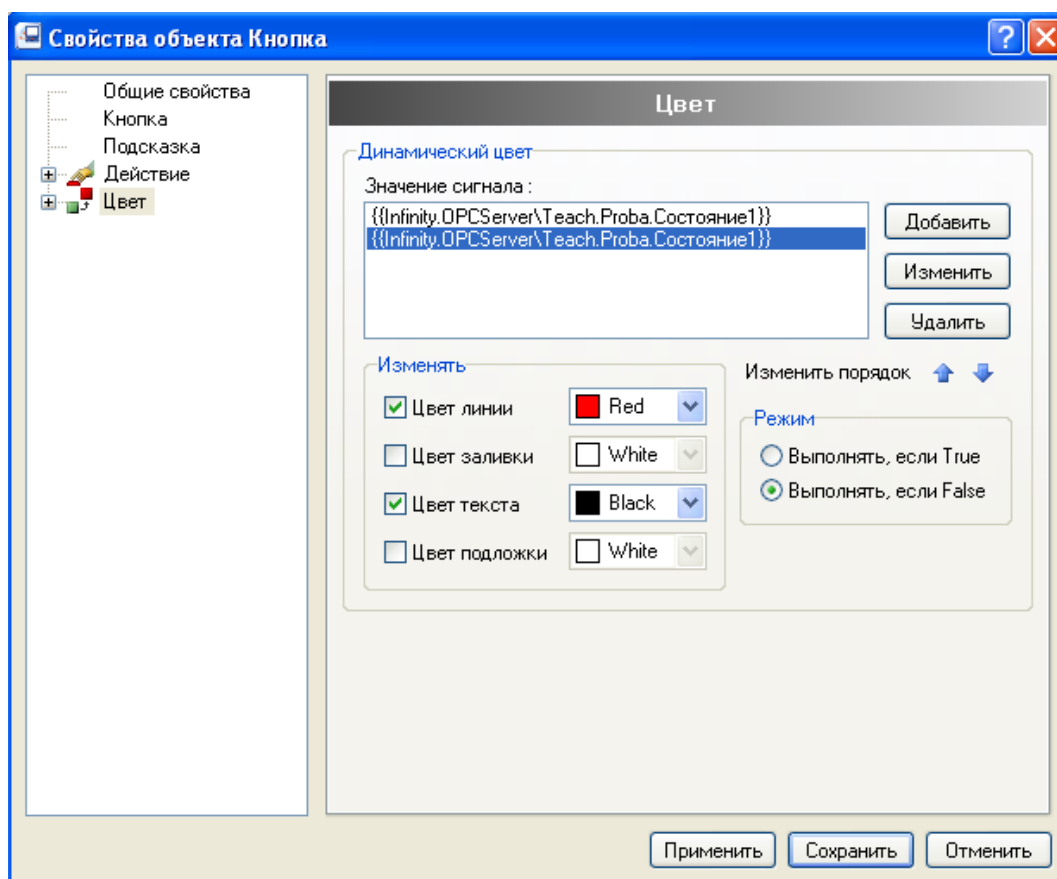


Рис. 15

20. Примените динамику **Цвет** к другому сгруппированному символу, куда входит инверсный выход **!Q**. В качестве сигнала используйте сигнал **Состояние2**.
21. Сохраните файл и проверьте работу кнопок и динамики цвет в режиме **Режим-Исполнение**. Если кнопки переключают сигналы **Управление1**, **Управление2** и корректно отображаются цвета линий соответствующих входов триггера, то можно приступать к реализации собственно триггера, работа которого определяется приведенной в п.1. формулой.



Применительно к сигналам, которые используются в мнемосхеме, логика работы триггера на элементах ИЛИ-НЕ будет выглядеть так: **Состояние1=!(Управление1||Состояние2)** **Состояние2=!(Управление2||Состояние1)**. Сигнал Управление1 (Сброс) складывается со значением сигнала Состояние2 (инверсный выход триггера), далее эта сумма инвертируется и записывается в сигнал Состояние2 (инверсный выход триггера). Поскольку путь к сигналу задается тегом, то логика примет вид:



22. Примените к верхнему по схеме символу динамику **Динамическое действие**. Настройте **Свойства объекта**, как показано на Рис. 16

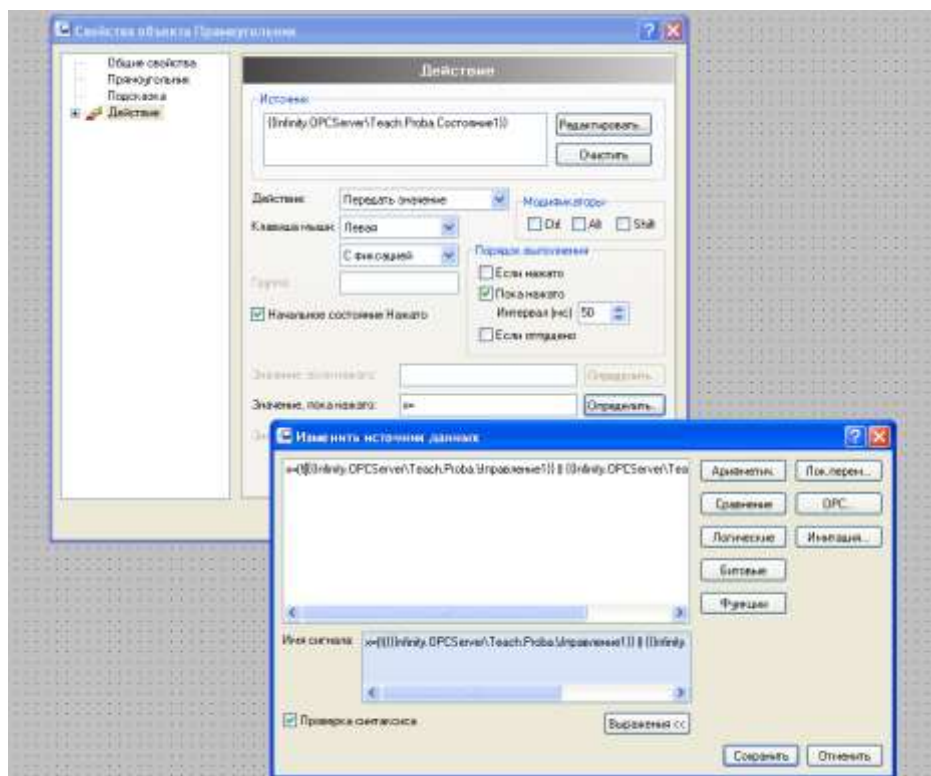


Рис. 16

23. Примените такую же динамику, но с соответствующими сигналами, к нижнему по схеме логическому элементу.

24. Проверьте работу мнемосхемы в режиме **Режим-Исполнение**. Как и реальный триггер, мнемосхема может некорректно работать при одновременной подаче на входы записи и сброса активных уровней, это будет наблюдаться в первый момент времени после перехода в режим **Режим-Исполнение**. Такой же эффект будет наблюдаться при слишком кратковременном нажатии на кнопку при переводе триггера в противоположное состояние. Нажмите любую из кнопок, и триггер займет предписанное логикой состояние. Если триггер защелкивает выходы, после снятия управляющих сигналов, значит, динамика настроена, верно. Вид экранной формы триггера на элементах ИЛИ-НЕ показан на Рис. 17.

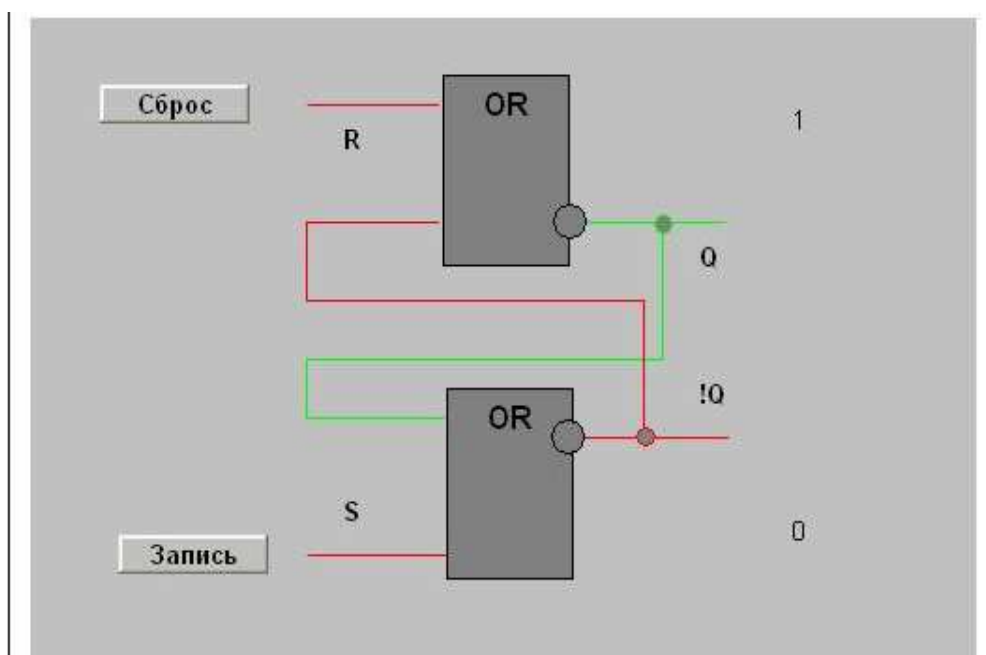


Рис. 16

Упражнение 3: Создание графической мнемосхемы «Функциональный блок Т».



Функциональный блок Т представляет собой тот же триггер – защелку, но с логикой, основанной на функции логического И-НЕ. В предыдущем упражнении было необходимо детализировать объект, вплоть до динамизации перекрестных связей, чтобы работа триггера была наглядна. Теперь возникает необходимость в создании одного функционального блока, как это принято, например, на электрических принципиальных схемах. Все это, конечно, не в ущерб возложенных на мнемосхему функций. Триггер выглядит так, как показан на Рис. 18. Логика работы $Q = \bar{S} \wedge \bar{Q} \wedge \bar{R}$. Из формулы видно, что кратковременная подача на вход R единицы сбросит единственный выход Q в ноль, после чего кратковременной подачей единицы на вход S можно снова перевести выход Q в активное состояние.

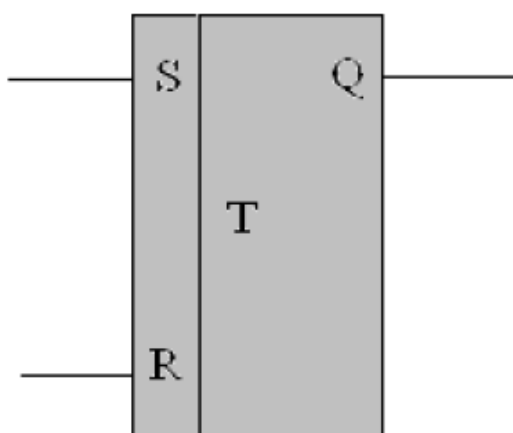


Рис. 18

1. Создайте файл с именем **Фамилия5.xml**.
2. Нарисуйте мнемосхему триггера с темно – серым цветом заливки. Внешний вид показан на Рис. 19. Сигналы: **Управление2** – Запись (S), **Управление1** – Сброс (R), **Состояние1** – Выход (Q).

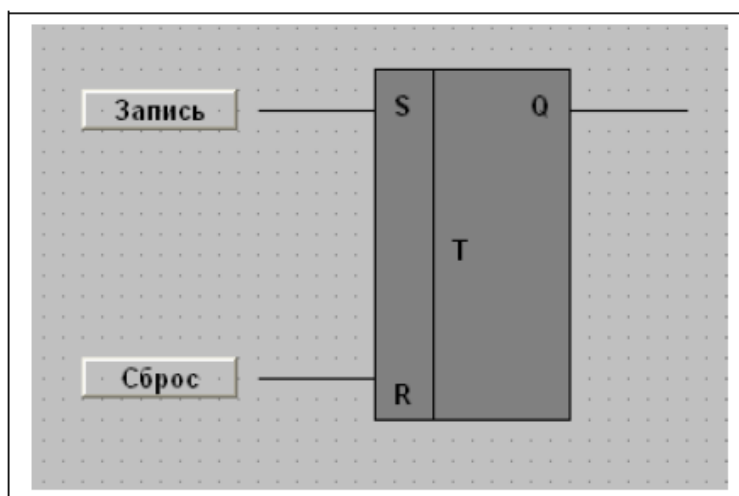


Рис. 19

3. Скопируйте кнопки «Сброс» и «Запись» из предыдущего файла и разместите их возле входов триггера соответственно.
4. Примените к входам и выходу триггера динамику **Цвет** по аналогии с

предыдущей мнемосхемой

5. Для вычисления выражения примените к объекту (к любому, но по смыслу разумно это сделать на символе изображения триггера) динамику **Динамическое действие** и настройте ее в соответствии с упоминавшейся формулой, как показано на Рис. 20.

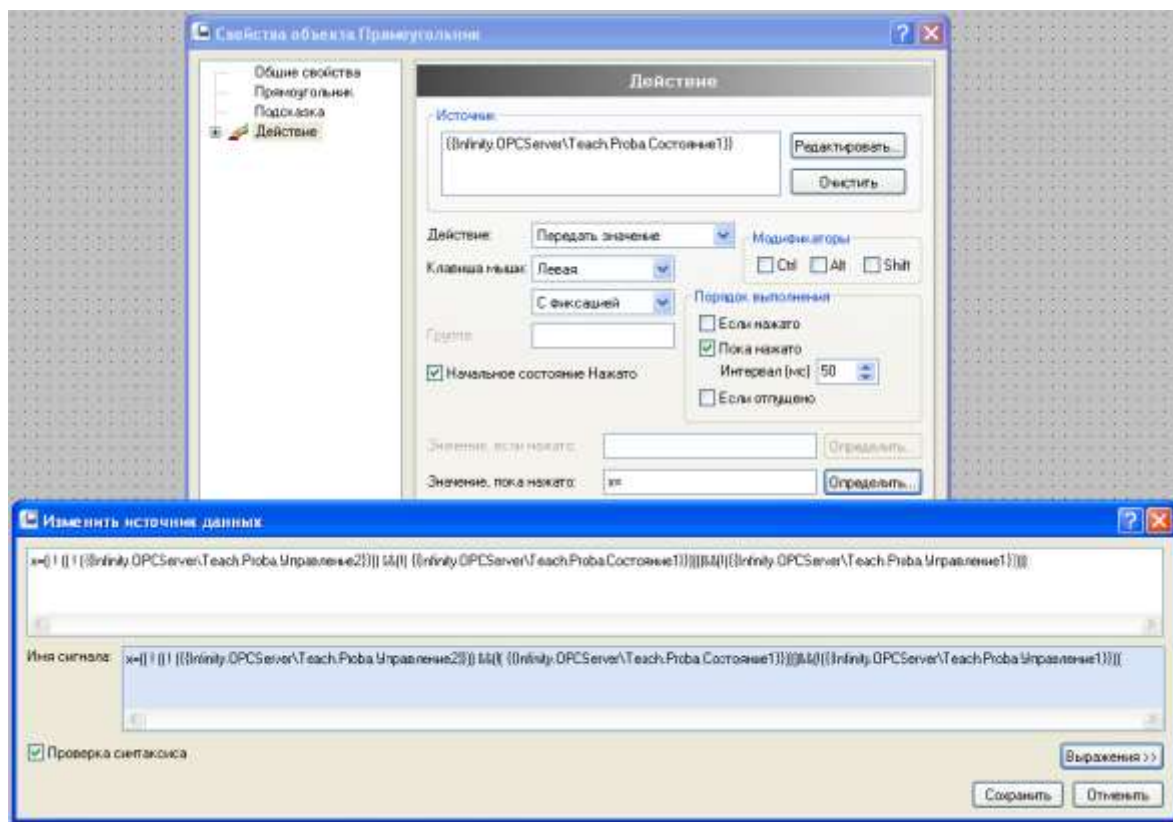


Рис. 20

6. Проверьте мнемосхему в режиме **Режим-Исполнение**, если она работает корректно, сохраните мнемосхему.

Упражнение 4: Создание графической мнемосхемы «Блок автоматики».



Представьте технологический процесс, в котором участвуют два исполнительных механизма. Это задвижка, через которую в емкость поступает вода, и насос, который осуществляет отгрузку. Насос и задвижка по условию задания не должны работать одновременно, то есть переключение между режимами набора и отгрузки только через кнопку «Сброс режима». Эту логику и будет обеспечивать блок автоматики. Функциональная схема блока автоматики показана на Рис. 21. В схему входят два триггера: T1 и T2 с логикой работы $Q = \bar{S} \wedge \bar{Q} \wedge \bar{R}$, два элемента И: AND1 и AND2, два инвертора: NOT1 и NOT2 и 3 кнопки без фиксации: SW1, SW2, SW3. При замыкании кнопок на соответствующие входы приходит логическая единица – активный уровень.

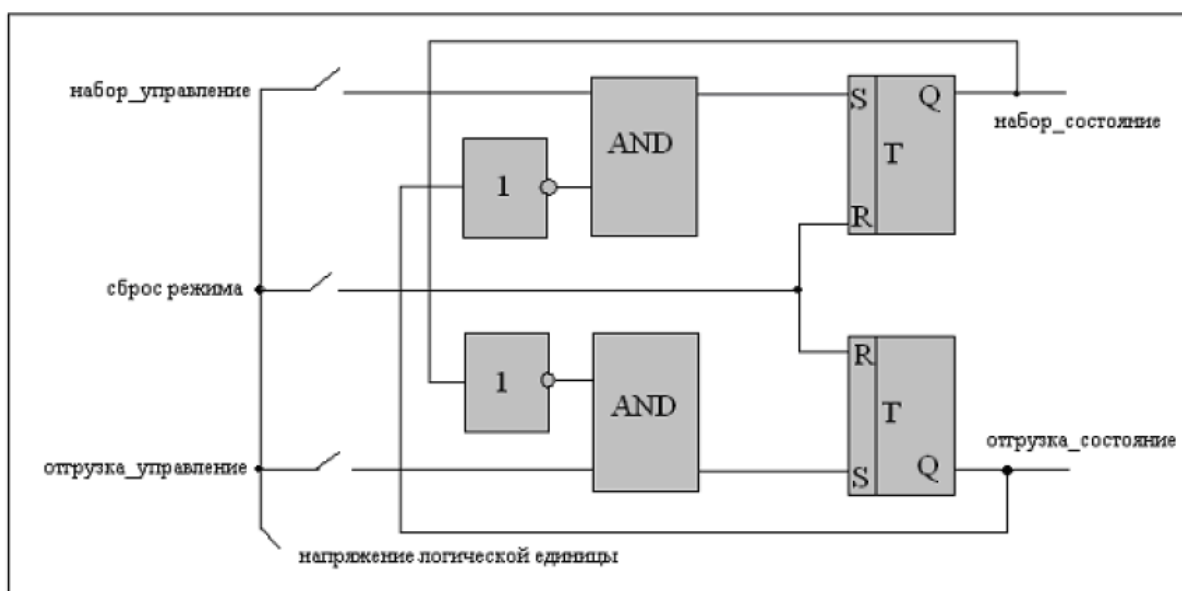


Рис. 21

1. Создайте новый файл **Фамилия6.xml** и включите сетку (на большинстве рисунков она не показана).

2. Создайте объекты, показанные на Рис. 22. Требования: цвет заливки объектов – **темно – серый**, размещение текста названия функциональных блоков – **по середине** и **по центру объектов**. Объекты располагать относительно друг друга с помощью необходимых инструментов панели **Расположение**.

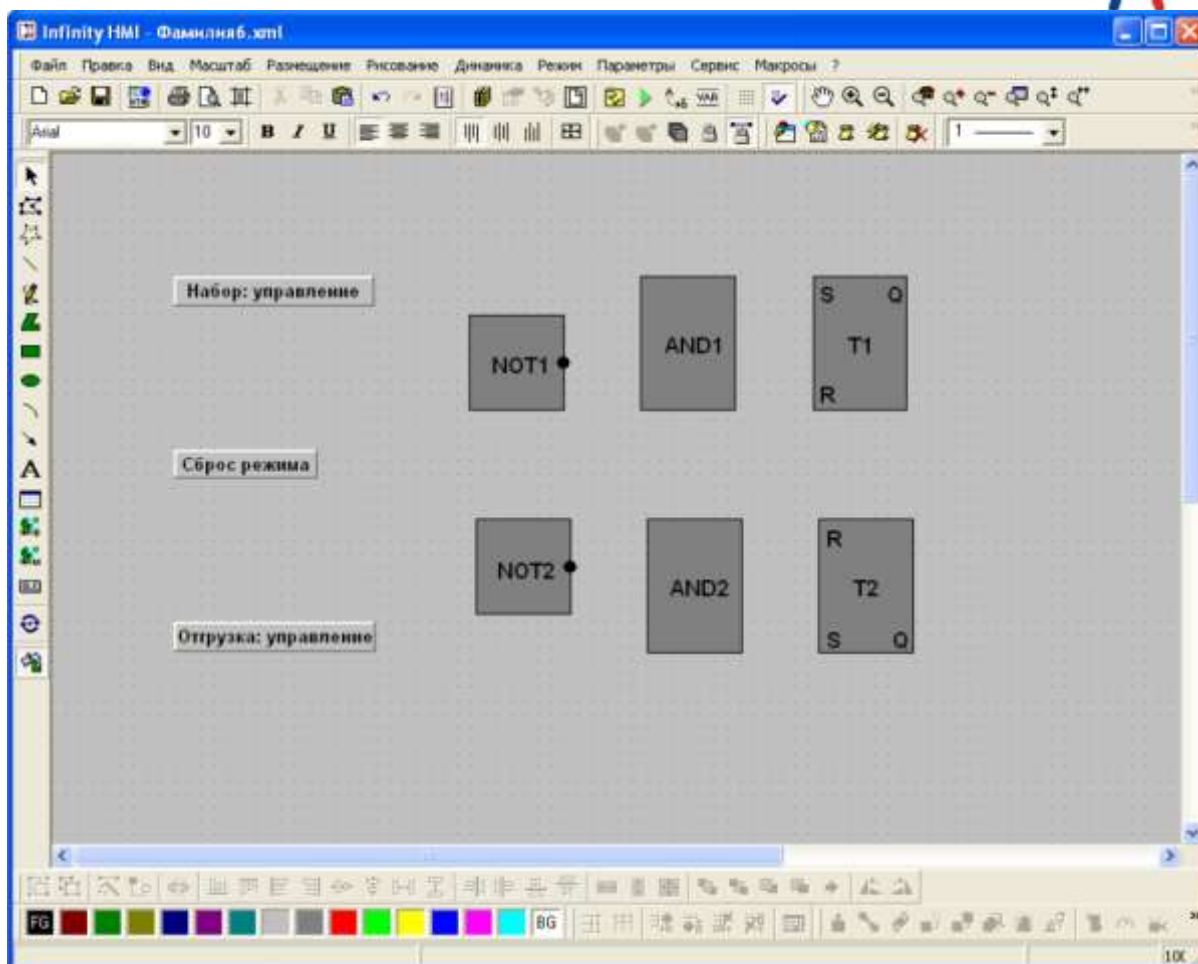


Рис. 22

3. Настройте динамику **Динамическое действие** кнопок по аналогии с мнемосхемой в предыдущем файле. Рис. 23

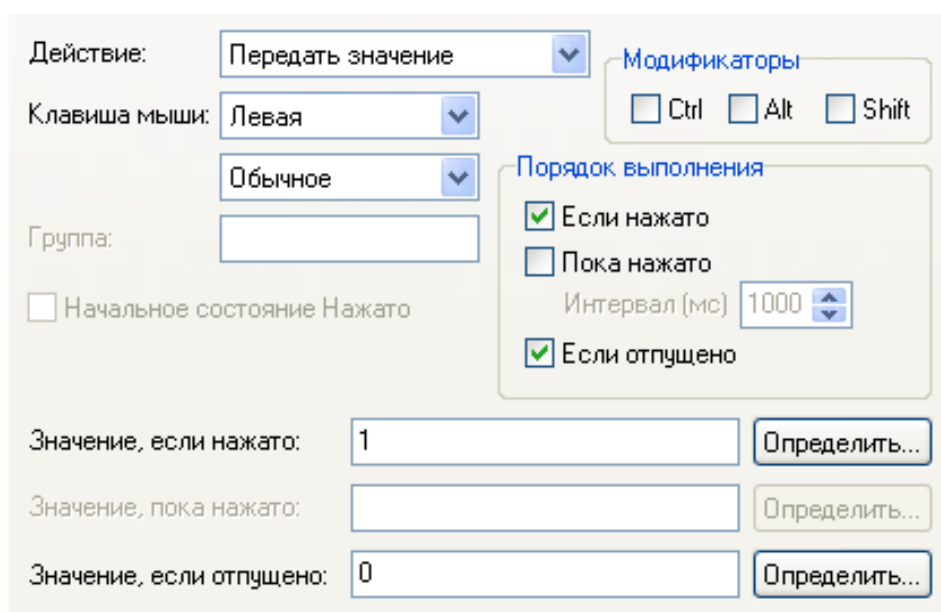



Рис. 23

Тег для кнопки «**Набор: управление**»: **Infinity.OPCServer\Work.Valve.In**
 Тег для кнопки «**Сброс режима**»: **Infinity.OPCServer\Work.Reset**
 Тег для кнопки «**Отгрузка: управление**»: **Infinity.OPCServer\Work.Pump.In**



Выберите панель инструментов **Стиль линий и заливка**

 , которая располагается в верхней части главного окна под панелью главного меню, и поставьте толщину линии **2**. Теперь все вновь нарисованные линии будут иметь толщину **2**. Нарисуйте связи, как показано на Рис. 24.

4. Сгруппируйте кнопки и функционально им соответствующие линии в символы. На Рис. 24 эти три символа выделены.

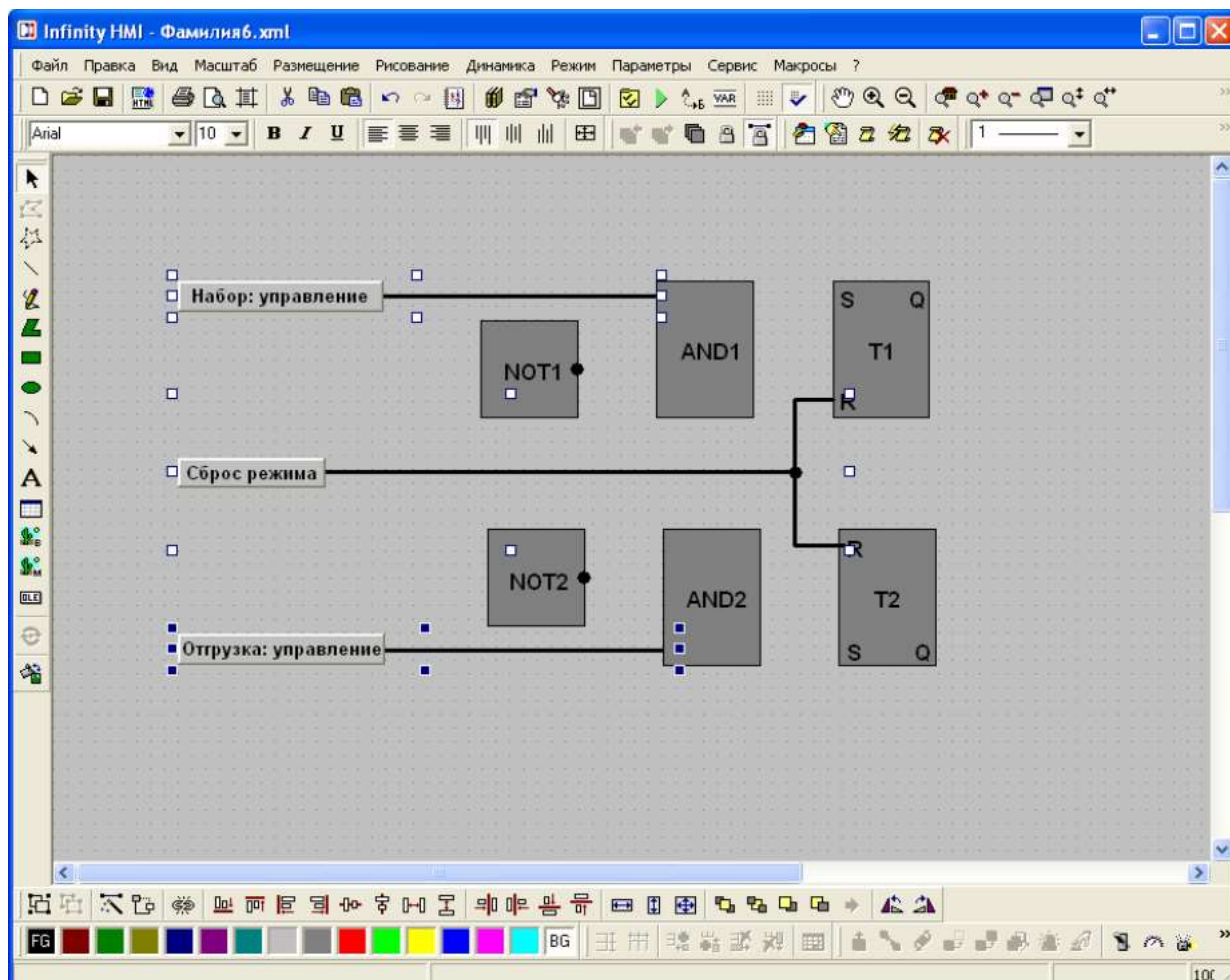


Рис. 24

5. Для визуализации значения тега примените к верхнему символу, куда входит кнопка «**Набор: управление**», динамику **Цвет**. Изменение цвета применяйте только к линии, тогда динамика будет отображаться собственно на линии, соединяющей кнопку с элементом **AND1**, и на тексте внутри кнопки. Цветовая политика прежняя.

6. Примените динамику **Цвет** к среднему и нижнему символам с кнопками. Условия применения динамики такие же, как в п. 6, теги соответствующие.

7. Сохраните файл и проверьте мнемосхему в режиме **Режим-Исполнение**. При записи в сигнал единицы должны менять цвет с красного на зеленый соответствующая линия и текст внутри кнопки.

8. Нарисуйте связь от выхода триггера **T1** к входу инвертирующего элемента **NOT2**. При рисовании ломаной линии руководствуйтесь рекомендациями пунктов 6 и 7 упражнения 2. Сгруппируйте отрезки линии в символ.

9. Добавьте к этому символу линию выхода и точку пересечения как показано на Рис. 25.

10. Сгруппируйте три объекта (ломанную, выход и точку пересечения выхода с ломаной) в один символ.

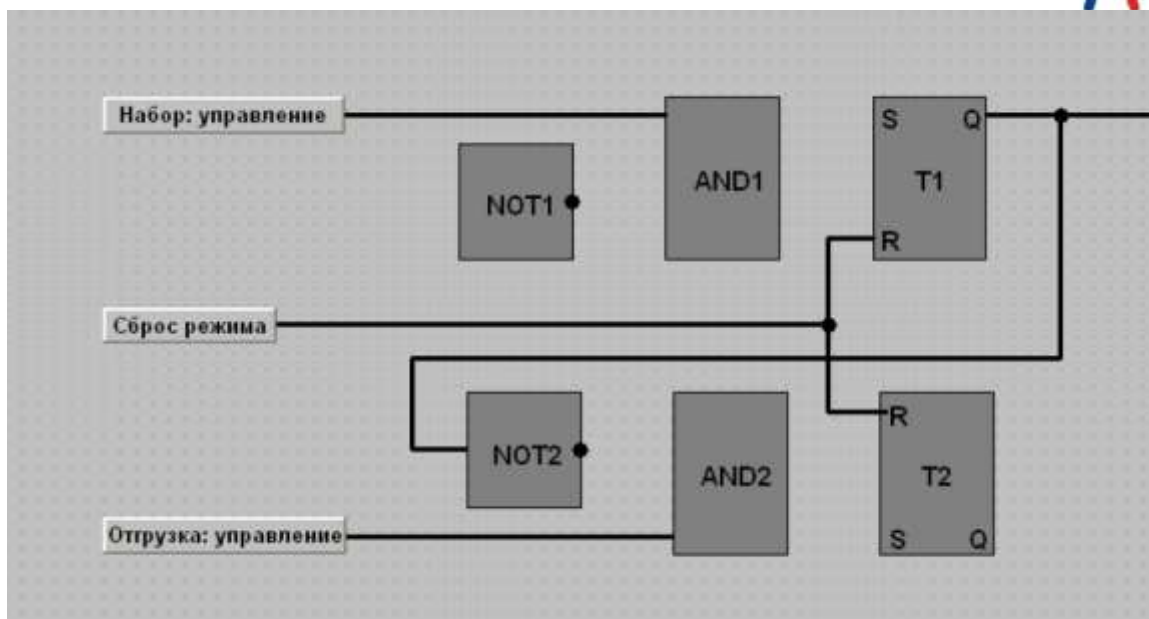


Рис. 25

11. Примените для сгруппированного символа динамику **Цвет** для визуализации значения тега **Infinity.OPCServer\Teach.Work.Valve.Control** (задвижка, управление).

12. Повторите действия, описанные в пунктах 9-12 применительно к связи выхода триггера **T2** с входом инвертора **NOT1**. Тег соответственно: **Infinity.OPCServer\Teach.Work.Pump.Control** (насос. управление). Экранная форма в режиме **Разработка** будет иметь вид, как на Рис. 26.

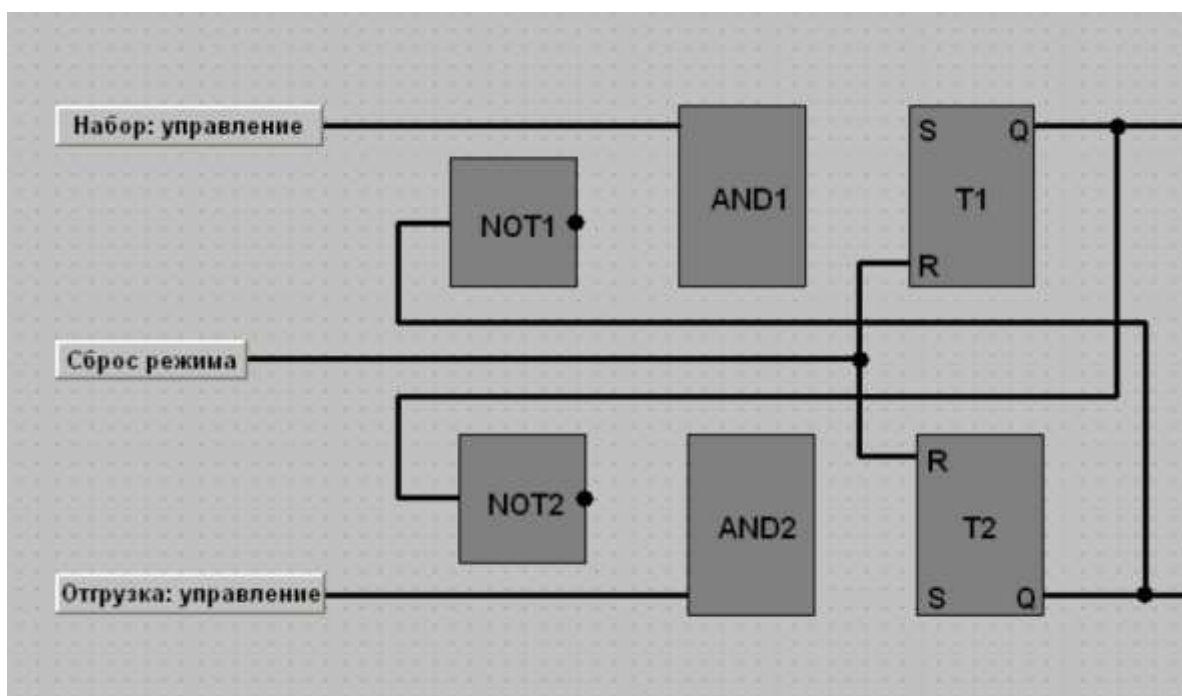


Рис. 26

13. Сохраните файл и проверьте мнемосхему. После перевода мнемосхемы в режим **Режим-Исполнение**, линии, связанные с сигналами управления должны быть красными.

14. Создайте подложки для блока кнопок и блока логики: Рис. 27 (Заливка выключена!). Внешний вид мнемосхемы после создания подложек, отделяющих блок кнопок от блока логики, показан на Рис. 28.

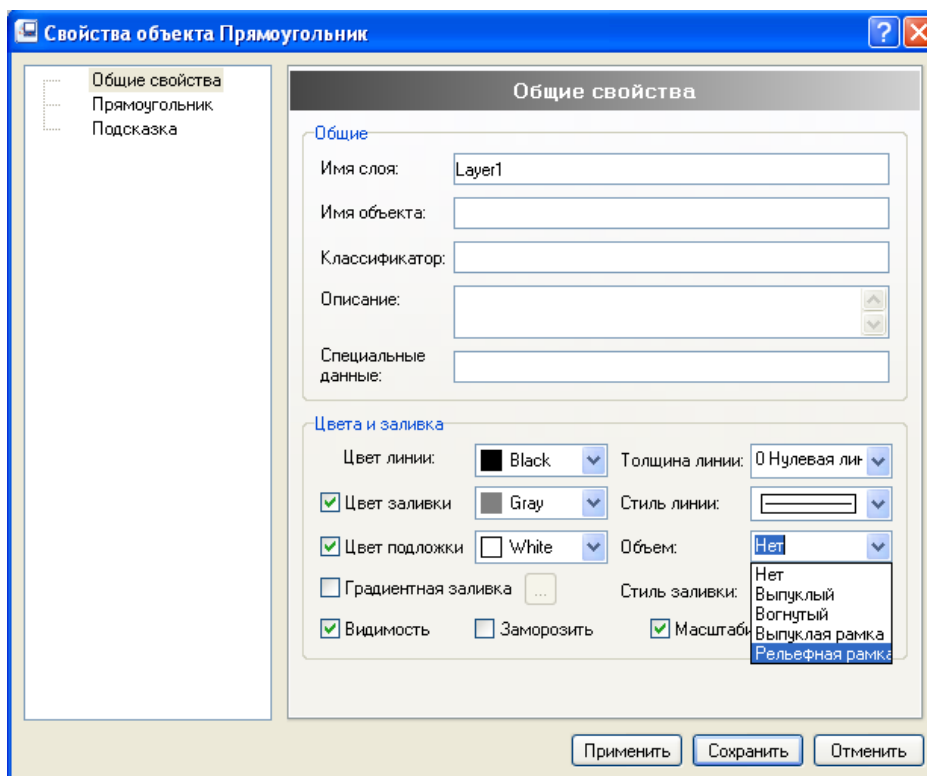


Рис. 27

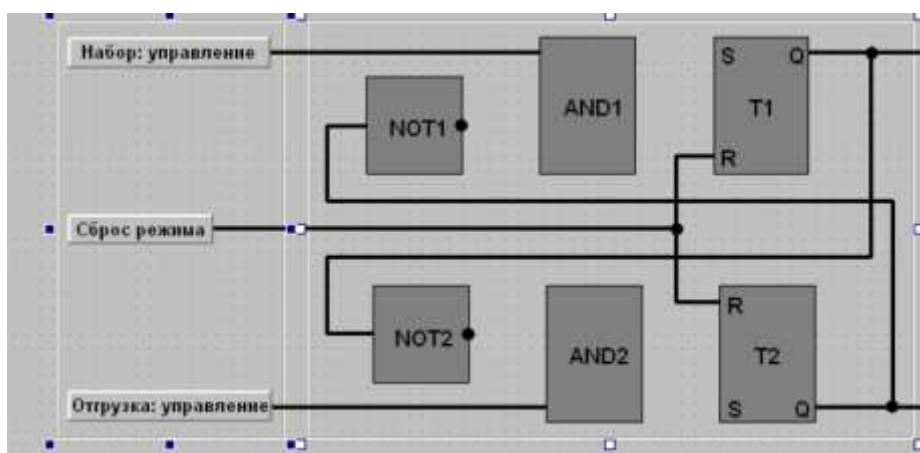


Рис. 28

15. Пронумеруйте входные и выходные сигналы, как это показано на Рис. 29.

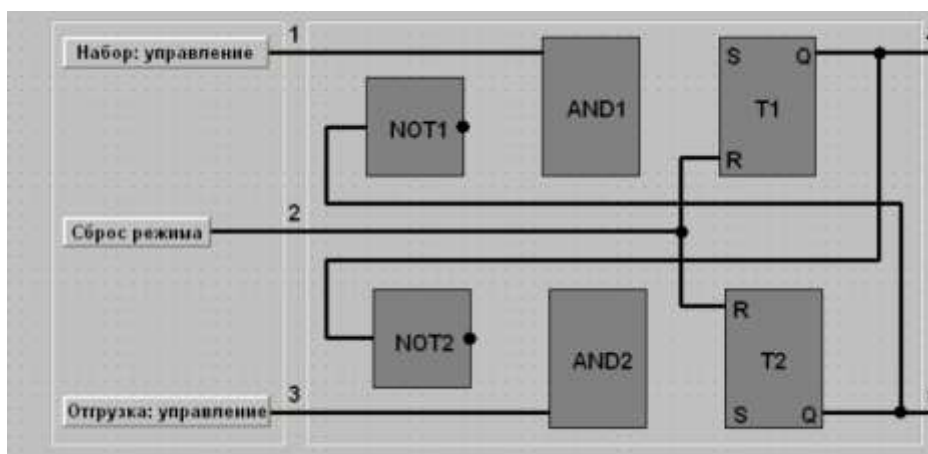


Рис. 29



Представьте, что в инверторах **NOT1** и **NOT2** мнемосхемы действительно происходит вычисление выражения (инверсии входного для инвертора сигнала). Значит должен быть сигнал, куда выражение будет записывать свое вычисленное значение (поле **Источник** в динамике **Динамическое действие**). Создавать новые сигналы, а новых нужно 4 сигнала (два на инверторы и два на элементы И), не целесообразно, так как они являются внутренними переменными. Рациональным выходом (но не единственным) из такой ситуации является объявление **локальной переменной**.



Локальная переменная в Infinity HMI – это переменная, область видимости которой ограничена экранной формой, в которой она объявлена. Имя локальной переменной должно заключаться в двойные волновые скобки: **~~Имя_локальной_переменной~~**. В имя локальной переменной могут входить любые символы, за исключением следующих:

- пробел;
- ~ (волна);
- ? (знак вопроса);
- * (звездочка);
- + (плюс).

Никаких прочих ограничений на имя переменной не существует (например, имя переменной может начинаться с цифры). Локальные переменные объявляются в поле **Источник**. После применения динамики (кнопка **Применить** в **Свойствах объекта**) локальная переменная будет считаться объявленной.

16. Выделите инвертор **NOT1** и примените к нему динамику **Динамическое действие**. В поле **Источник** введите: **~~invert_Pump.Control~~**. В эту локальную переменную будет записываться инвертированное значение тега **Infinity.OPCServer\Teach.Work.Pump.Control**. Настройка динамики для инвертора **NOT1** показана на Рис. 30.

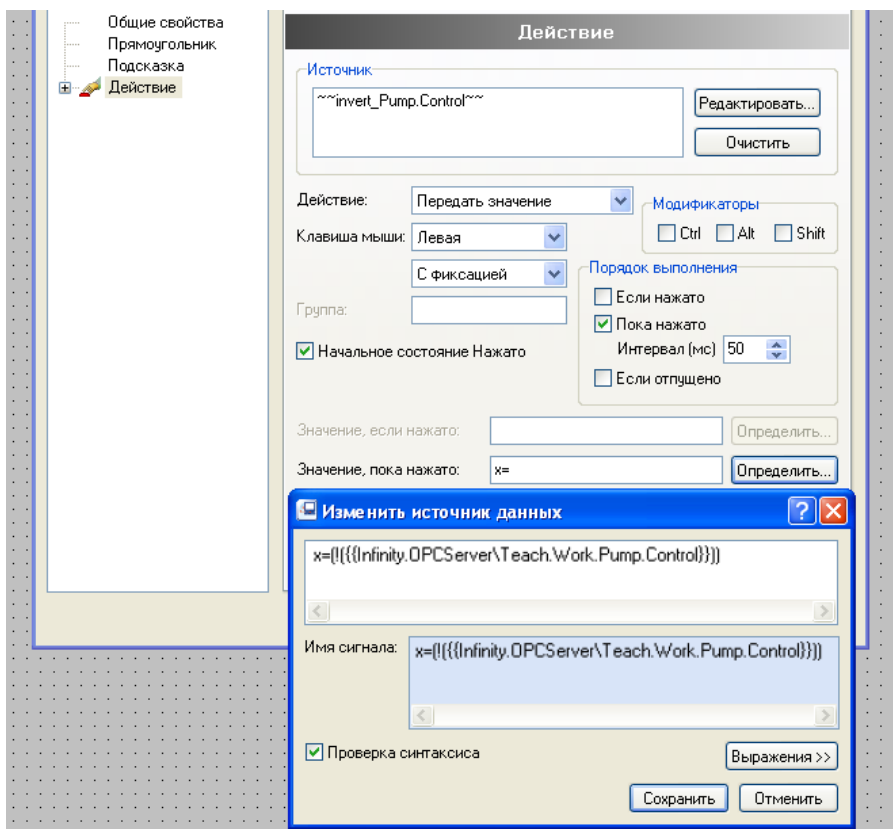


Рис. 30



17. Примените аналогичную динамику для инвертора **NOT2**. Локальную переменную назовите **invert_Valve.Control**. Выражение в редакторе должно быть, естественно, свое.

18. Нарисуйте связи между выходами инверторов и соответствующими входами элементов **AND1** и **AND2**.

19. Выделите верхнюю линию и примените к ней инструмент динамики **Цвет** для визуализации значения **invert_Pump.Control**. В настройке изменения нет ничего нового. В поле **Источник** локальная переменная вводится как тег, только нужно выбирать **Выражения – Локальные переменные** (Рис. 31).

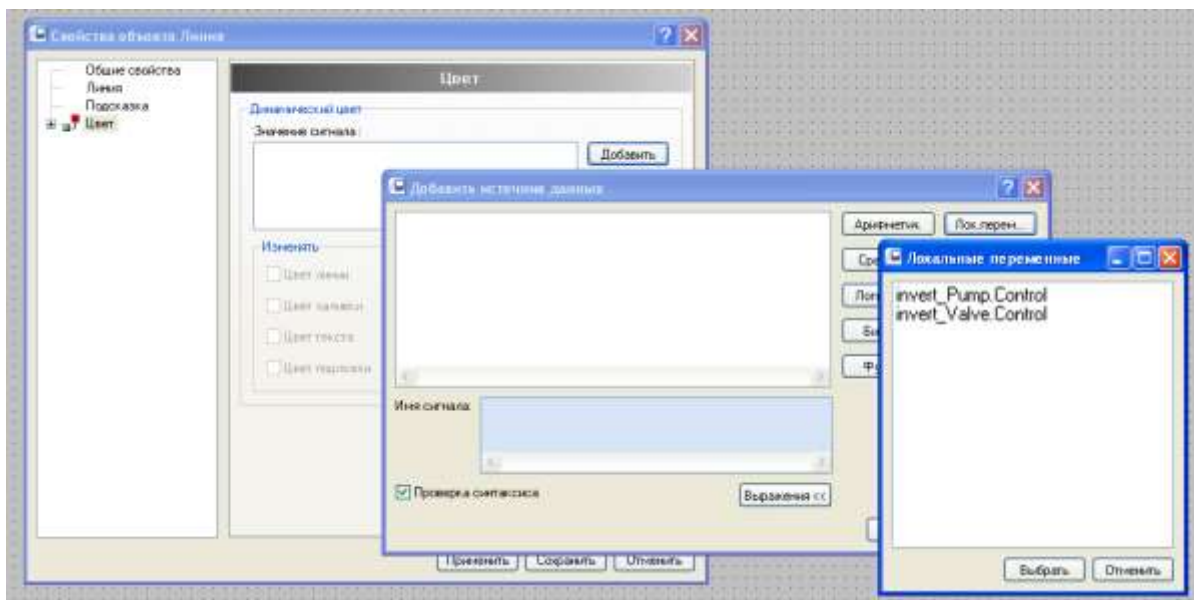


Рис. 31

20. По аналогии примените динамику **Цвет** для нижней линии, соединяющей выход инвертора **NOT2** с входом элемента **AND2**.

21. В верхнем меню нажмите кнопку **VAR** - **Редактирование локальной переменной** (или выберите в меню пункт **Динамика**, затем **Редактирование локальной переменной**) и в появившемся окне измените, тип данных для каждой переменной на **BOOL**, как на Рис. 32, после чего проверьте мнемосхему. На вход каждого инвертора должна приходиться линия красного цвета, выходная линия будет зеленой.

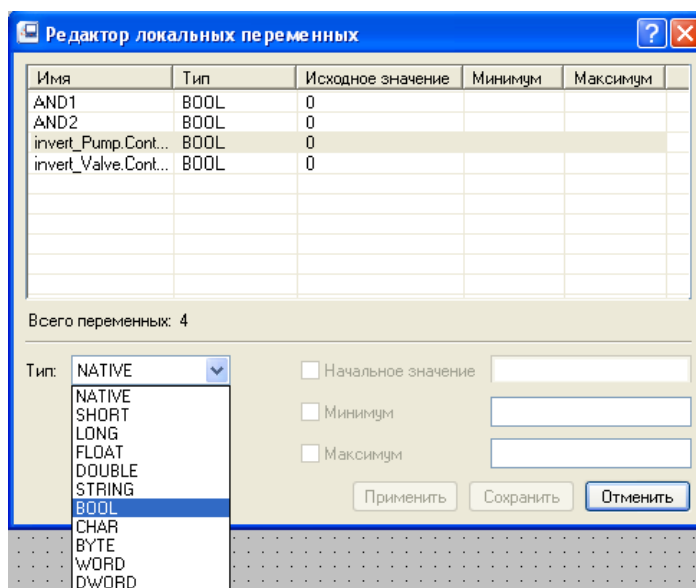


Рис. 32



22. Примените необходимую динамику для элемента **AND1**, где будет происходить логическое умножение тега **Infinity.OPCServer\Teach.Work.Valve.In** с локальной переменной **~~invert_Pump.Control~~**. Локальную переменную, в которую будет записываться вычисленное значение, назовите **~~AND1~~** и измените ее тип на **BOOL**.

23. Нарисуйте связь между выходом **AND1** и входом **S** триггера **T1**. 24. Примените к линии необходимую динамику для визуализации значения **~~AND1~~**.

25. Прделайте необходимые действия для элемента **AND2**. Локальную переменную назовите **~~AND2~~**.

26. Нарисуйте связь между выходом **AND2** и входом **S** триггера **T2** и примените к линии динамику **Цвет** для визуализации значения **~~AND2~~**.

27. Сохраните файл. Внешний вид экранной формы на этом этапе показан на Рис.33.

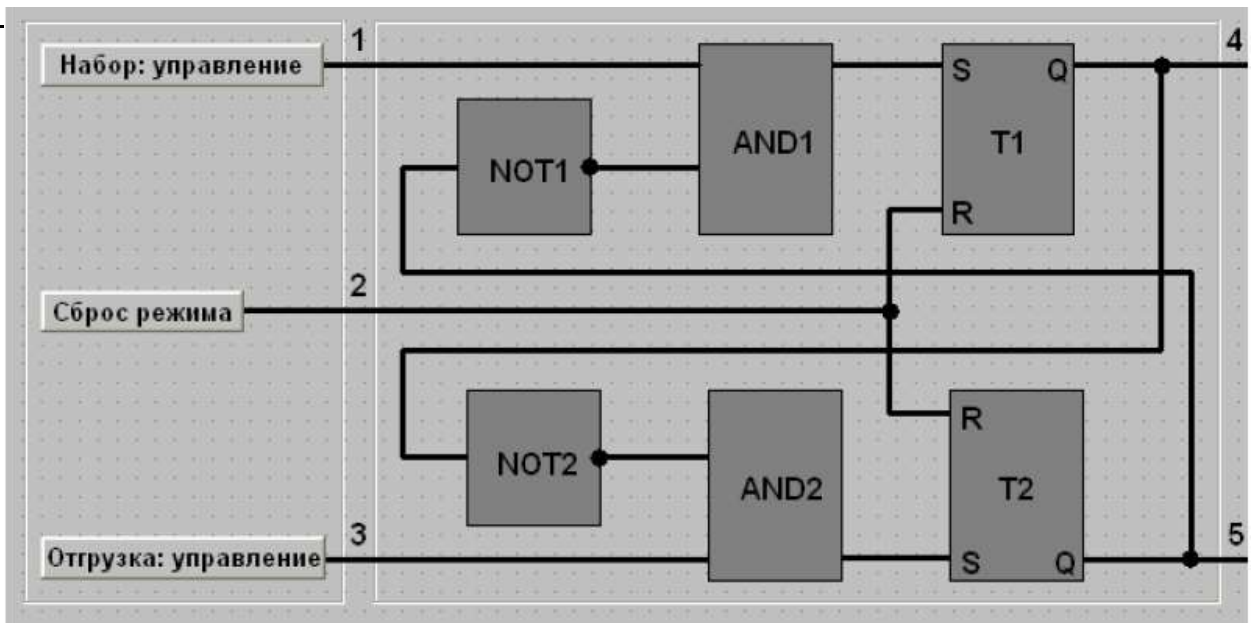


Рис. 33

28. Проверьте мнемосхему. Любой ноль на входе элементов **AND1** и **AND2** даст ноль на выходе (линия красного цвета). Только две единицы на входе дадут единицу на выходе. Для проверки логики нажимайте соответствующие кнопки.

29. Примените к элементу **T1** динамику для обсчета формулы, которая показана в общем виде:

$$\text{Valve.Control} = \text{~~AND1~~} \ \&\& \ \text{Valve.Control} \ \&\& \ \text{Reset}$$

30. Примените к элементу **T2** необходимую динамику для обсчета формулы, которая показана в общем виде:

$$\text{Pump.Control} = \text{~~AND1~~} \ \&\& \ \text{Pump.Control} \ \&\& \ \text{Reset}$$

31. В главном меню **Правка** выберите **Выделить все**.

32. Сгруппируйте объекты в один символ. Проверьте мнемосхему. Активным может быть только один из выходов. Переключение между режимами (выходы 4 и 5) осуществляйте через кнопку «Сброс режима». На этом лабораторная работа закончена.