**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ (ИТТСУ)**

**Кафедра «Управление и защита информации»**

А.И. САФРОНОВ

**Проектирование и создание виртуальных приборов National Instruments LabView**

**Сборник типовых задач**

**для проведения аудиторных занятий по Учебной практике**

**МОСКВА – 2021**

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»

ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ (ИТТСУ)

Кафедра «Управление и защита информации»

А.И. САФРОНОВ

Проектирование и создание виртуальных

приборов National Instruments LabView

Сборник типовых задач

для магистрантов направления

27.04.04 «Управление в технических системах»

**МОСКВА – 2021**

**УДК 004**

**С 21**

Сафронов А.И. Проектирование и создание виртуальных приборов *National Instruments LabView*: Сборник типовых задач для проведения аудиторных занятий по Учебной практике. – М.: РУТ, 2021. – … с.

Сборник типовых задач содержит постановки к работам, которые обучающиеся должны выполнить для приобретения навыков использования пакета прикладных программ и среды разработки виртуальных приборов *National Instruments LabView. NI LabView* является передовой разработкой в области информационных технологий, позволяющей выполнять моделирование как чисто программных, так и программно-аппаратных звеньев автоматизированных систем управления на базе типовых микропроцессорных устройств, а также специализированных микропроцессорных модулей, предназначенных для сложных оптимизационных расчётов, проводимых в реальном времени. Сборник задач ориентирован на рассмотрение программных реализаций, предоставляя обучающимся возможность понять внутреннее устройство решений, внедрённых в учебный процесс на кафедре «Управление и защита информации». Приобретённые навыки способствуют закреплению основ программирования и алгоритмизации, развитию образного мышления, эффективному восприятию принципиальных технических схем, пониманию потоковых вычислений и элементов параллельного программирования.

© РУТ (МИИТ), 2021

# Введение

Данный сборник типовых задач содержит постановки к пятнадцати инженерным, практическим заданиям, необходимым для развития первичных профессиональных умений и навыков, в том числе, первичных умений и навыков научно-исследовательской деятельности обучающихся.

К числу первичных умений и навыков научно-исследовательской деятельности стоит отнести отработку на «Учебной практике»:

– аналитического вывода формул,

– построения алгоритмов решения инженерных задач,

– поиска способов решения инженерных задач,

– подбора соответствующих методов, обоснования принятых решений,

– формулировки развёрнутого и содержательного вывода о проделанной работе.

Сборник типовых задач является учебно-методическим дополнением действующей и утверждённой программы практик, проводимых на кафедре «Управление и защита информации», а также за её пределами, на предприятиях [1].

Счёт заданий, изложенных в данном сборнике начинается с девятки («9»). Подразумевается, что в качестве первого задания «Учебной практики» обучающиеся выполняют отчётную документацию к решённой задаче алгоритмизации и программирования, постановка которой изложена учебно-методическом пособии [2], а задания со второго по восьмое содержатся в сборнике задач [3] и позволяют обучающимся закрепить навыки работы в электронных таблицах *Microsoft Office Excel*. Обучающимся необходимо внимательно следить за нумерацией, указанной в постановочных частях заданий сборника типовых задач, поскольку реальные номера этих заданий, согласно программе «Учебной практики», отличаются от номеров пунктов, под которыми они изложены в настоящем издании.

Задания, входящие в состав сборника типовых задач, систематизированы согласно следующей опорной схеме:

– наименование задания (указано в заголовке раздела первого уровня);

– постановочный раздел задания, содержащий формулировку общей части работы для группы обучающихся (общая часть задания выполняется всеми обучающимися без исключения до выполнения индивидуальных задач и должна быть оформлена в соответствующем разделе отчёта; в некоторых заданиях сборника общая часть может отсутствовать);

– формулировка цели работы;

– полезные соотношения, необходимые для выполнения работы (актуально не для всех задач сборника);

– пример выполнения общей части работы или её фрагмента (актуально не для всех задач сборника);

– варианты индивидуального задания;

– образец компоновки графического пользовательского интерфейса виртуального прибора (далее – ВП);

– образец базовой разметки исходных данных на блок-диаграмме ВП;

– схему алгоритма функционирования типового ВП, создаваемого в рамках решения текущей рассматриваемой задачи.

Сборник типовых задач разработан с целью повышения познавательной активности обучающихся, достигаемой за счёт уместных отсылок к справочной литературе, а также ранее пройденному материалу в курсах:

– «Высшей математики»,

– «Программирования и основ алгоритмизации»,

– «Информатики»,

– «Физики»,

– «Теоретических основ электротехники»,

– «Электроники»,

– «Микропроцессорных устройств систем управления»,

– «Численных методов в инженерных задачах»,

– «Компьютерной математики»,

– и других.

Отчёты, оформляемые по результатам выполнения задач программирования и алгоритмизации, должны соответствовать структуре, представленной в Приложении 1.

# 1 Начало работы с пакетом прикладных программ *National Instruments LabView*

Для решения всех поставленных задач сборника предлагается работа в пакете прикладных программ *National Instruments LabView* версии 6.1[[1]](#footnote-1). В разделах, содержащих полезные соотношения для выполнения задач, рассматриваются интерфейс, основные разделы меню, а также элементы управления именно этой версии программного обеспечения.

В качестве альтернативы для обучающихся, привыкших решать задачи не по шаблонам, предлагается работа в пакете прикладных программ *National Instruments LabView* выпуска2016 года[[2]](#footnote-2).

Представленные по сноске гиперссылки содержат апробированное программное обеспечение, укомплектованное всеми необходимыми компонентами для его установки на персональные компьютеры, работающими под управлением современных операционных систем семейства *Microsoft Windows*.

Процесс установки программного обеспечения на персональный компьютер в сборнике не рассматривается. Дальнейшее изложение материала предполагает, что *NI LabView* версии 6.1 уже предустановлен на персональном компьютере, но ещё ни разу не запускался.

После запуска исполняемого файла пакета прикладных программ *National Instruments LabView 6.1* открывается (если настройками не предусмотрено иного) диалоговое окно (Рисунок 1.1), в котором можно:

– приступить к созданию нового ВП (*«New VI»*);

– продолжить разработку ранее созданного ВП (*«Open VI»*);

– перейти к готовым решениям по сбору данных (*«DAQ (Data Acquisition) Solutions»*; при первом обращении запускает «мастер настройки устройств сбора информации» – *DAQ hardware Wizard*);

– перейти к справочной подсистеме поиска примеров реализации ВП (*«Find Examples»*), входящих в состав пакета прикладных программ *NI LabView*;

– приступить к изучению справочных материалов (*«LabView Tutorial»*), позволяющих быстро начать работу с пакетом прикладных программ *National Instruments LabView 6.1* при использовании другой справочной подсистемы;

– завершить работу с диалоговым окном (*«Exit»*).

Представленное описание относится к набору кнопок, расположенных в правой части диалогового окна. Левая часть диалогового окна (Рисунок 1.1) содержит логотип *NI LabView 6.1*, быстрые подсказки по работе с пакетом прикладных программ, а также область настройки вывода диалогового окна начала работы.

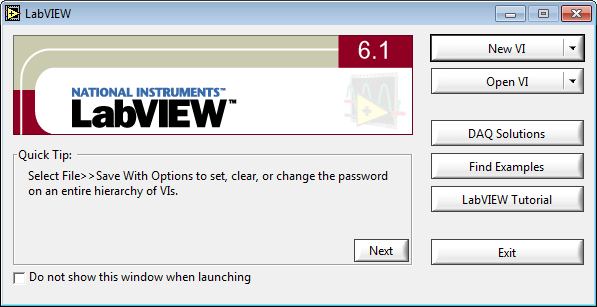


Рисунок 1.1 – Начальное диалоговое окно пакета прикладных программ *National Instruments LabView 6.1*. Первый запуск программного обеспечения

Диалоговое окно начала работы, запущенное впервые на персональном компьютере, отличает возможность выставления логического признака, запрещающего вывод аналогичного диалогового окна при последующих запусках программного обеспечения («*Do not show this window when launching*»).

Если логический признак не был выставлен при первом запуске программного обеспечения, аналогичное диалоговое окно будет демонстрироваться каждый раз при обращении к исполняемому файлу *NI LabView*. Вместо логического признака на нём будет размещаться логический тумблер, переключающий развёрнутый диалог (*«Large Dialog»*) на краткий диалог (*«Small Dialog»*) и обратно (Рисунок 1.2).

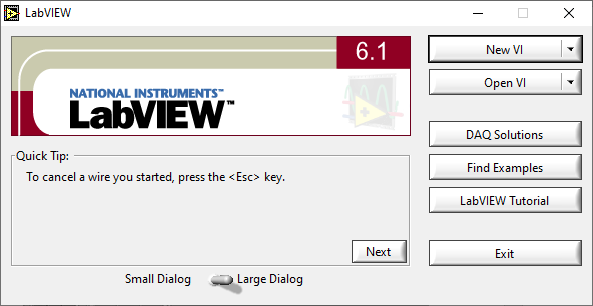


Рисунок 1.2 – Начальное диалоговое окно пакета прикладных программ *National Instruments LabView 6.1*. Последующие запуски программного обеспечения. Развёрнутый диалог

Краткий диалог (Рисунок 1.3) отличается от развёрнутого тем, что из него исключены все элементы справочных подсистем, а также элементы связи со внешними системами сбора данных (датчиками).

Вообще говоря, материалы, представленные далее в сборнике типовых задач, всецело ориентированы на использование краткого диалога. В нём рассматривается разработка программного, а не программно-аппаратного обеспечения, а также в качестве справочного материала выполнено целевое описание всех необходимых элементов управления. Подразумевается, что обращение обучающихся к справочным подсистемам *NI LabView 6.1* при решении задач сборника является избыточным.

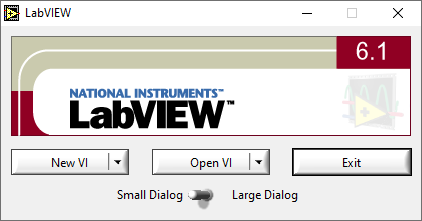


Рисунок 1.3 – Начальное диалоговое окно пакета прикладных программ *National Instruments LabView 6.1*. Последующие запуски программного обеспечения. Краткий диалог

Для начала работы по созданию нового ВП необходимо нажать на кнопку *«New VI»*. Сокращение *«VI»* в данном случае является обозначением самого ВП (то есть *«Virtual Instrument»* в англоязычной литературе, «Виртуальный Прибор» – в русскоязычной литературе).

Выбор данного режима автоматически генерирует две рабочих области, каждая из которых имеет собственное устойчивое наименование. Как правило, сверху расположена передняя панель (*«Front Panel»*) – графический пользовательский интерфейс ВП, под ней – блок-диаграмма кода (*«Block Diagram»*). Область передней панели нового ВП показана на Рисунке 1.4.

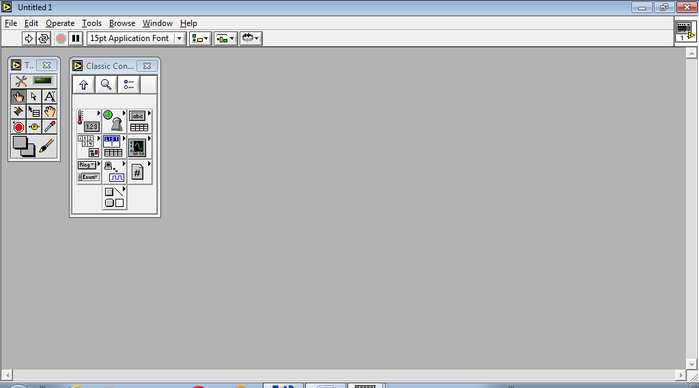


Рисунок 1.4 – Область проектирования передней панели нового ВП в пакете прикладных программ *National Instruments LabView 6.1*

Для перехода от передней панели к блок-диаграмме опытные пользователи применяют сочетание клавиш *«Ctrl» + «E»*, менее опытные – используют переход через меню *«Window > Show Diagram»*, как это показано на Рисунке 1.5.

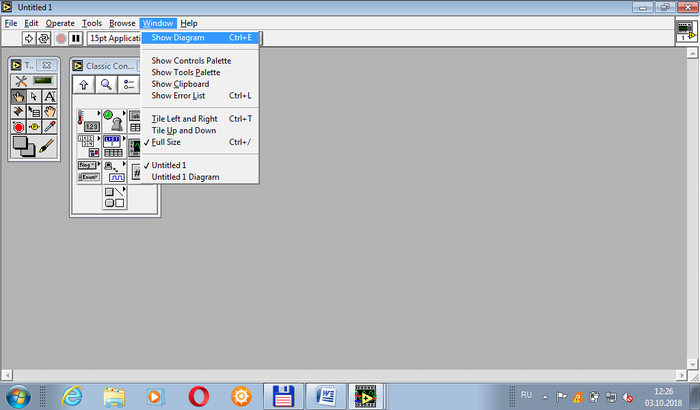
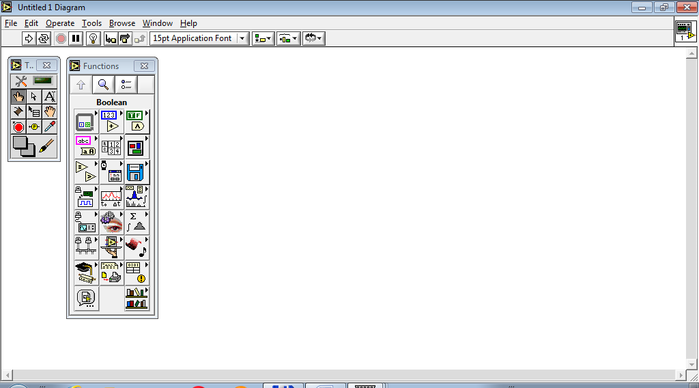


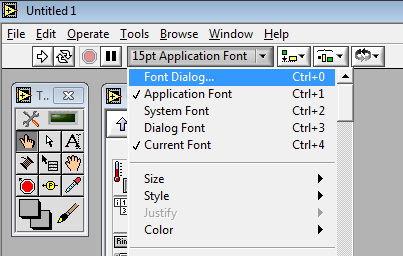
Рисунок 1.5 – Переход к блок-диаграмме ВП через пункт меню *«Window»*

На Рисунке 1.6 показана область блок-диаграммы нового ВП.

  
Рисунок 1.6 – Область блок-диаграммы нового ВП *National Instruments LabView 6.1*

На сегодняшний день известна одна существенная проблема, препятствующая комфортной работе в *National Instruments LabView 6.1*. Связана проблема с некорректным отображением кириллических шрифтов при настройках, заданных по умолчанию. Данная проблема имеет решение.

На панели инструментов, расположенной строкой ниже по отношению к главному меню, необходимо нажать на самую большую кнопку (она предназначена для выбора и настройки параметров шрифта). Переход к основным настройкам производится путём нажатия на первую строку выпадающего списка – это вызов диалогового окна шрифтов (*«Font Dialog…»*). Расположение пункта меню показано на Рисунке 1.7.

Рисунок 1.7 – Переход к диалоговому окну настройки шрифтов

В раскрывшемся диалоговом окне необходимо проследить за тем, чтобы обе галочки «Умолчания для передней панели» (*«Panel Default»*) и «Умолчания для блок-диаграммы» (*«Diagram Default»*) были выставлены. После этого в комбинированном списке шрифтов (*«Font»*) следует выбрать любой кириллический шрифт. Для такого шрифта характерен постфикс «*Cyr*». Все остальные умолчания подбираются на усмотрение автора-разработчика ВП, либо оставляются им без изменений.

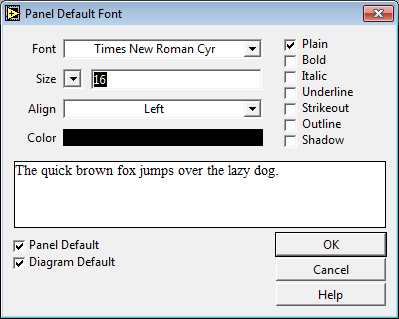


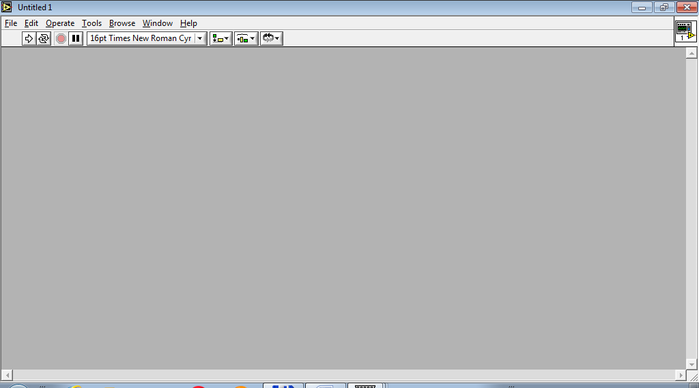
Рисунок 1.8 – Рекомендации по настройке шрифта, заданного в *National Instruments LabView* по умолчанию

При выставленных флагах *«Panel Default»* и *«Diagram Default»* указанные настройки шрифта будут применены по умолчанию для текущего разрабатываемого ВП, а также для последующих создаваемых ВП.

Стоит отметить, что при всём многообразии функций и возможностей пакета прикладных программ *National Instruments LabView* в первой каждой задаче сборника рассматривается и закрепляется лишь небольшой их фрагмент, что соответствует дозированному изучению.

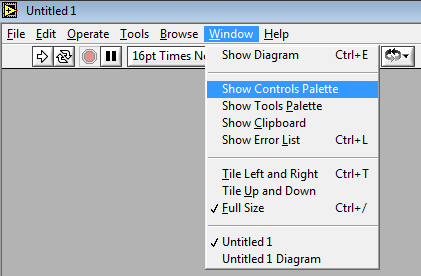
После устранения проблем со шрифтами необходимо проверить готовность среды разработки к нормальному и удобному функционированию.

Допустим, при первом запуске *National Instruments LabView* все панели инструментов скрыты от пользователя, как это показано на Рисунке 1.9.

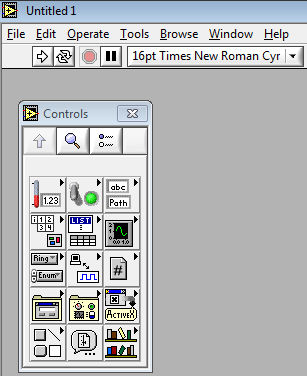
Рисунок 1.9 – Состояние *National Instruments LabView 6.1*, при котором основные панели инструментов скрыты

Для отображения на рабочей области основных панелей инструментов достаточно помнить места расположения пунктов меню, связанных с отображением этих программных компонентов. Они находятся в том же пункте меню *«Window»*, через который осуществляется переход между областями передней панели и блок-диаграммы.

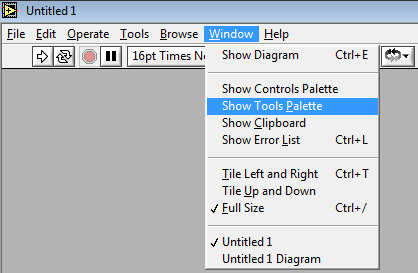
Сначала отобразим панель интерфейсных элементов управления, которые позволяют обеспечить взаимодействие конечного пользователя ВП с внутренними механизмами ВП. Для этого в главном меню следует пройти путь *«Window > Show Controls Palette»* как это показано на Рисунке 1.10.

Рисунок 1.10 – Переход по меню к отображению панели интерфейсных элементов управления

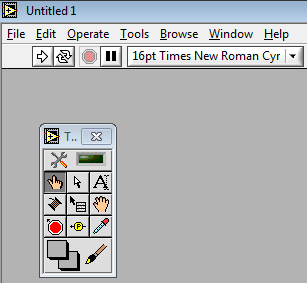
В результате выбора указанного пункта меню над передней панелью ВП отобразится структура, представленная на Рисунке 1.11 или аналогичная ей.

  
Рисунок 1.11 – Отображение панели интерфейсных элементов управления

Далее отобразим панель основных инструментов, предоставляющих возможность манипулировать элементами интерфейса, а также элементами в составе графического кода. Для этого в главном меню следует пройти путь *«Window > Show Tools Palette»*, как показано на Рисунке 1.12.

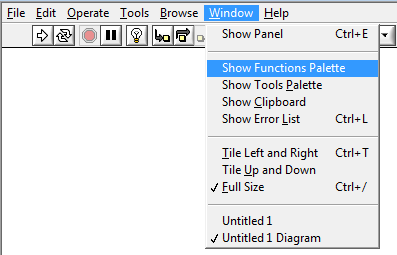
Рисунок 1.12 – Переход по меню к отображению панели основных инструментов

В результате выбора указанного пункта меню над передней панелью ВП отобразится структура, представленная на Рисунке 1.13.

  
Рисунок 1.13 – Отображение панели основных инструментов

Панель основных инструментов является общей как для передней панели, так и для блок-диаграммы ВП.

К настоящему моменту без рассмотрения осталась только одна панель из числа часто используемых – панель функций среды разработки. Она доступна для вызова только в режиме составления блок-диаграммы ВП. В главном меню следует пройти путь *«Window > Show Functions Palette»* (Рисунок 1.14).

  
Рисунок 1.14 – Переход по меню к отображению панели функций среды

В результате выбора указанного пункта меню над блок-диаграммой ВП отобразится структура, представленная на Рисунке 1.15.

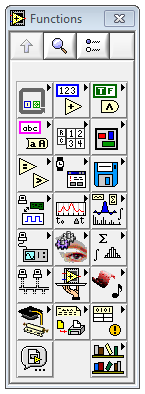
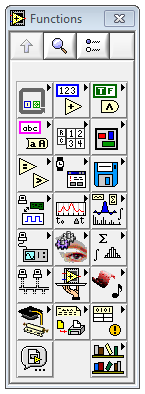
 

Рисунок 1.15 – Отображение панели функций среды

После вызова всех упомянутых панелей считается, что пакет прикладных программ *National Instruments LabView* готов к нормальному взаимодействую в части разработки нового ВП.

# 2 Задание «Разработка виртуального прибора "Простой калькулятор"»

В пакете прикладных программ *National Instruments LabView* разработать виртуальный прибор «Простой Калькулятор», позволяющий оператору по итогам работы этого виртуального прибора наблюдать результаты выполнения следующих арифметических операций:

– сложение двух операндов,

– вычитание одного операнда из другого,

– умножение двух операндов,

– деление одного операнда на другой (с индикацией ошибки при делении на ноль).

В рамках разработки графического пользовательского интерфейса предусмотреть:

– две ячейки (контроллера) для ввода числовой информации (по количеству операндов, над которыми выполняются арифметические операции);

– четыре ячейки (индикатора) для вывода числовой информации (по количеству используемых арифметических операций);

– лампочку (логический индикатор) для вывода логической информации с целью реагирования на ошибочный ввод исходных данных – ноль в знаменателе.

Все интерфейсные элементы управления должны содержать значащие имена, отражающие назначение этих элементов, например, ячейки для ввода информации могут быть названы:

– «Операнд 1», «Операнд 2»;

– «Слагаемое 1 / Уменьшаемое / Множитель 1 / Делимое», «Слагаемое 2 / Вычитаемое / Множитель 2 / Делитель».

Необходимо продумать индивидуальный авторский дизайн графического пользовательского интерфейса «Простого Калькулятора». Двух одинаковых дизайнерских решений для виртуального прибора «Простой Калькулятор» в одной учебной студенческой группе быть не должно.

Изменения, вносимые в исходные данные, должны сказываться на отображаемых результатах в реальном времени.

При подготовке блок-диаграммы графического кода программы минимизировать количество пересечений линий связи, а также попытаться создать максимально компактную и читаемую схему.

Выше сформулированы требования для выполнения общей части работы.

По итогам выполнения работы сдаются строго четыре файла:

- отчёт, выполненный в текстовом редакторе Microsoft Office Word (\*.doc или \*.docx);

- файл виртуального прибора National Instruments LabView (\*.vi) по общей части работы;

- файл виртуального прибора National Instruments LabView (\*.vi) по индивидуальной части работы, связанной с закреплением навыков использования переменных (контроллеров и индикаторов);

- файл виртуального прибора National Instruments LabView (\*.vi) по индивидуальной части работы, связанной с закреплением навыков использования констант и некоторых стандартных функций.

Отправленные поодиночке файлы проверке не подлежат. При отсутствии одного из упомянутых файлов зачёт по заданию не выставляется.

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «9».

## 2.1 Цель работы

Закрепление навыков работы в пакете прикладных программ *National Instruments LabView*, развитие навыков алгоритмизации и графического программирования.

## 2.2 Полезные соотношения для выполнения задания

**Термины**

Операнд –

Операция –

**Организация однозначной связи между численными элементами управления**

Нет ничего проще в *NI LabView*, чем задать однозначное соответствие между входными и выходными данными одного и того же типа. Решим эту задачу для численных элементов управления.

В перечне доступных интерфейсных элементов управления необходимо перейти в раздел «Численные» (*«Numeric»*), как показано на Рисунке 2.2.1.

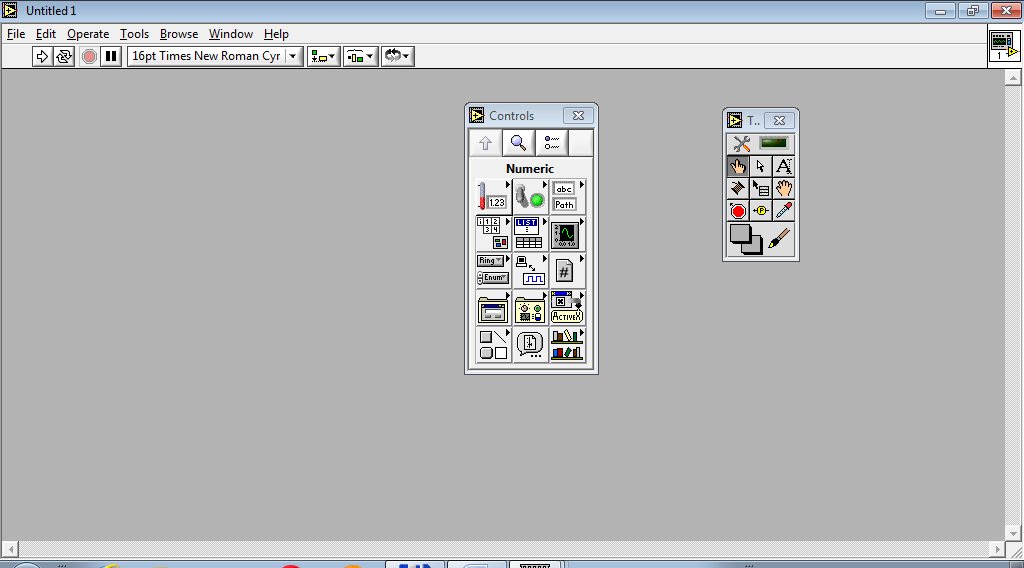


Рисунок 2.2.1 – Отметка в панели интерфейсных элементов управления (*«Controls»*) раздела численных элементов (Numeric)

В раскрывшемся перечне выбрать «числовой контроллер» (Рисунок 2.2.2), необходимый для ввода каких-либо значений в программу, написанную на графическом языке программирования.

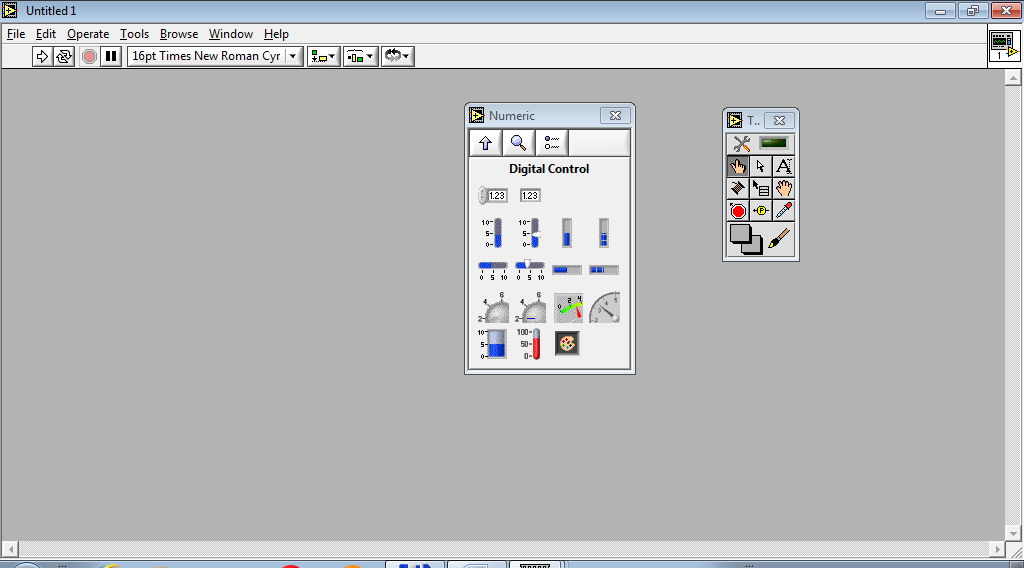


Рисунок 2.2.2 – Содержимое раздела численных элементов (Numeric) с подсветкой контроллера (Digital Control)

И разместим выбранный контроллер на передней панели виртуального прибора (Рисунок 2.2.3). Аналогичным способом на передней панели может быть размещено сколько угодно таких же численных контроллеров. Одновременно с размещением на панели образа элемента, на блок-диаграмме размечается его представление в виде графической переменной (это будет показано позже).

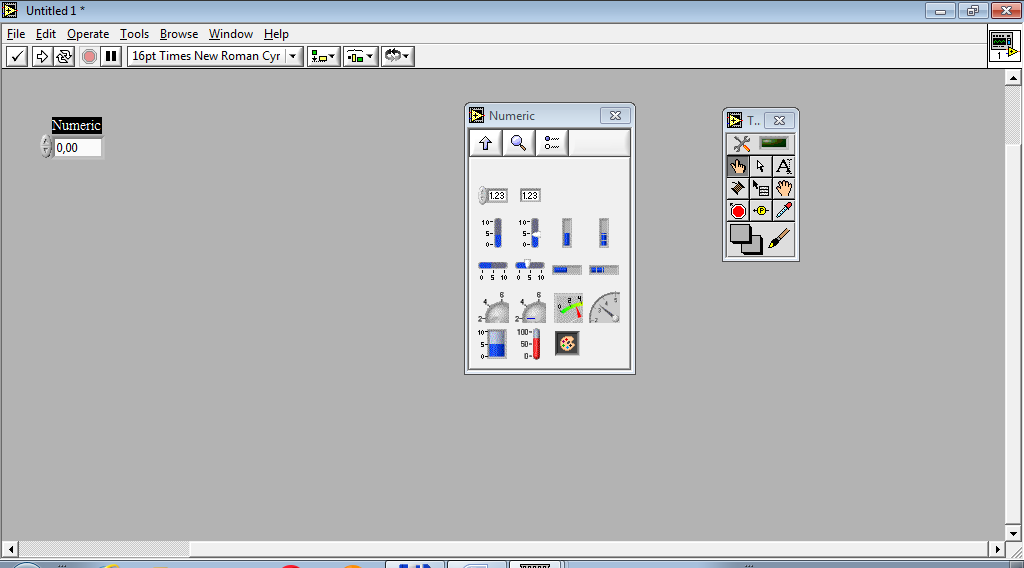


Рисунок 2.2.3 – Размещение контроллера (Digital Control) на передней панели

На панели раздел численных элементов нами не закрывался, потому выберем в нём теперь соседний элемент – численных индикатор (Рисунок 2.2.4), необходимый для вывода результирующей или промежуточной расчётной информации.

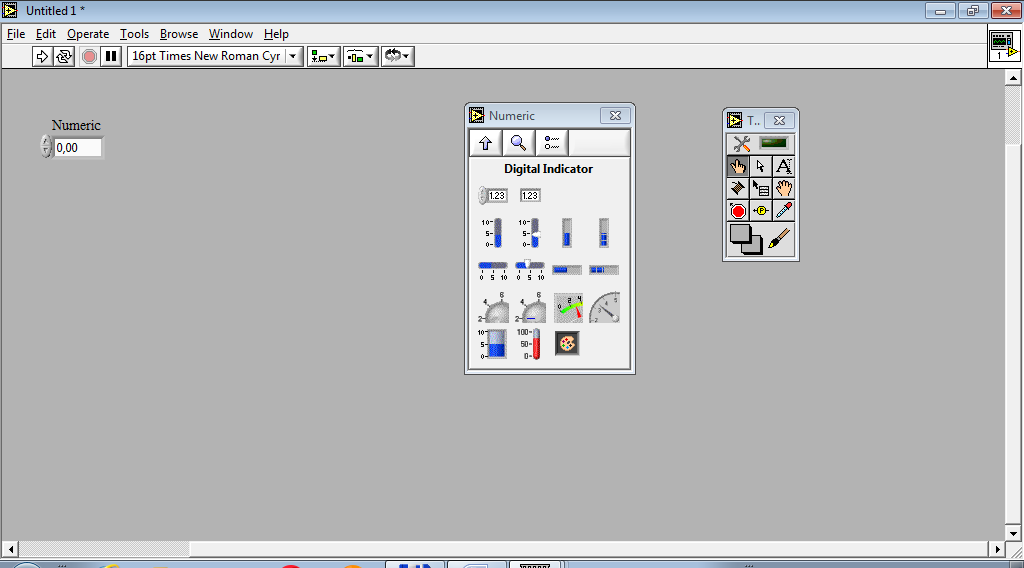


Рисунок 2.2.4 – Содержимое раздела численных элементов (Numeric) с подсветкой индикатора (Digital Indicator)

Разместим выбранный индикатор на передней панели рядом с ранее размещённым контроллером (Рисунок 2.2.5).

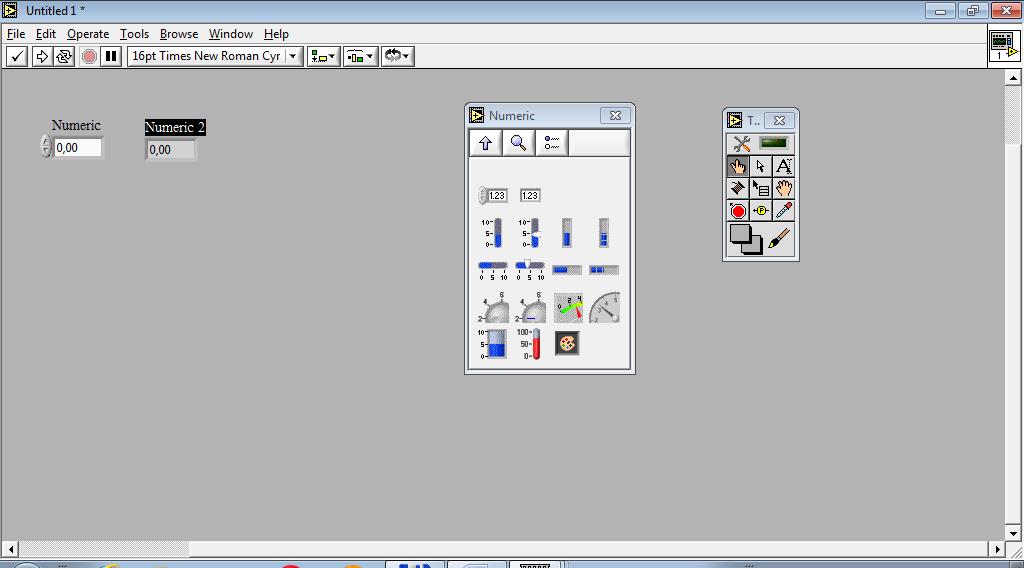


Рисунок 2.2.5 – Размещение индикатора (Digital Indicator) на передней панели

После щелчка в свободной области передней панели снимаем выделение с последнего размещённого на графическом пользовательском интерфейсе компонента и в этот момент считаем заготовку/болванку для виртуального прибора выполненной. Согласно одному из правил вежливости программиста, переменным необходимо давать значащие имена, которые бы позволили лицу, намеренному или поставленному продолжить написание ранее составленного кода, быстрее в этом коде разобраться. В LabView, что приятно, имена переменных однозначно связаны с именами элементов на передней панели, потому проявляя заботу о конечном пользователе одновременно получается заботиться и о программисте. Для выдачи значащих имён интерфейсным элементам управления используется инструмент редактирования текста (Рисунок 2.2.6).

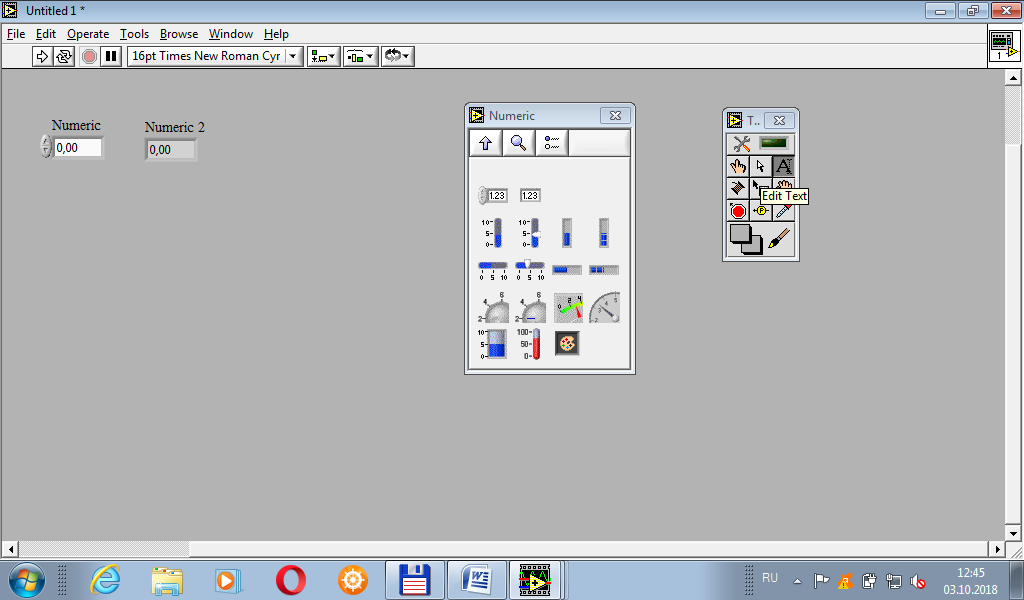


Рисунок 2.2.6 – Инструмент для редактирования текста

Программный продукт, не являющийся дружественным для пользователя, является непригодным для использования. Итого, руководствуясь вышесказанным, назовём контроллер входом, а индикатор – выходом (можно назвать синонимично вводом и выводом, соответственно). Результат показан на Рисунке 2.2.7.

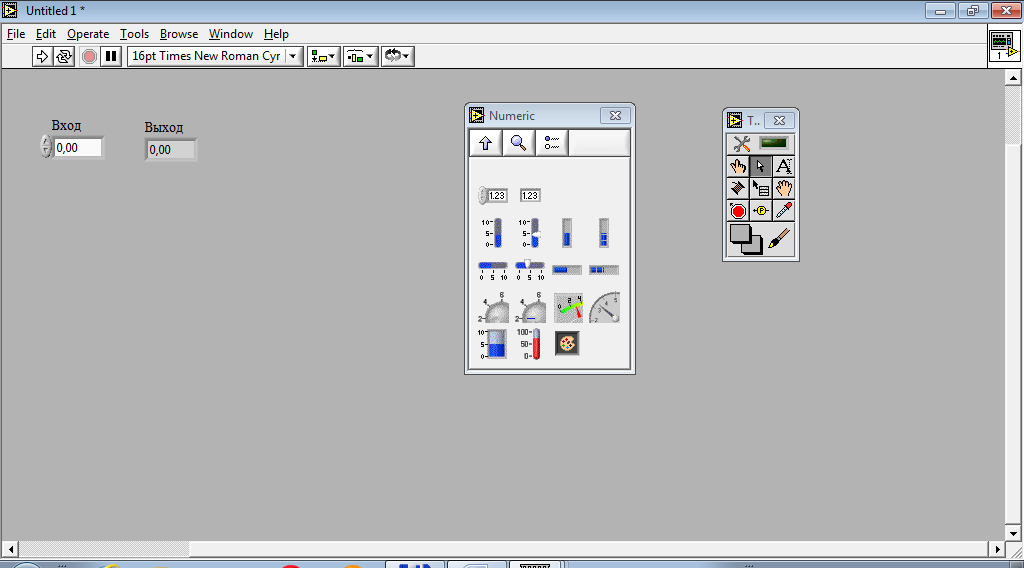


Рисунок 2.2.7 – Присвоение значащих имён переменным, связанным с контроллером и индикатором

Если с эстетической точки зрения начинает казаться, что элементы расположены криво, можно произвести их выравнивание относительно друг друга, предварительно выделив их и выбрав одну из операций выравнивания, как это показано на Рисунке 2.2.8.

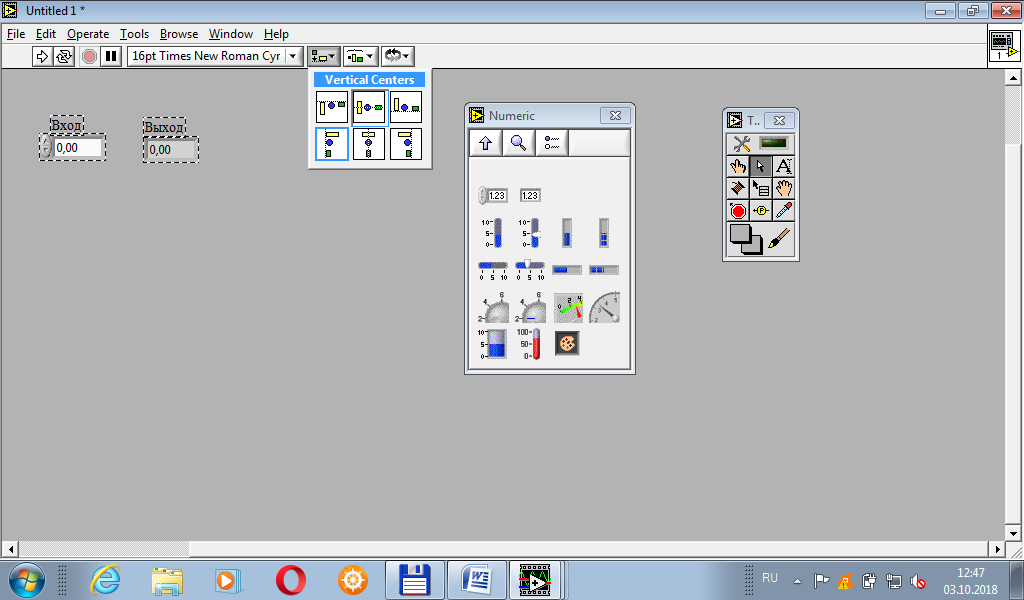


Рисунок 2.2.8 – Выравнивание контроллера и индикатора относительно друг друга по общей для них центральной и горизонтальной оси

После наведения порядка на передней панели переходим к кодированию, на блок-диаграмму (Рисунок 2.2.9).

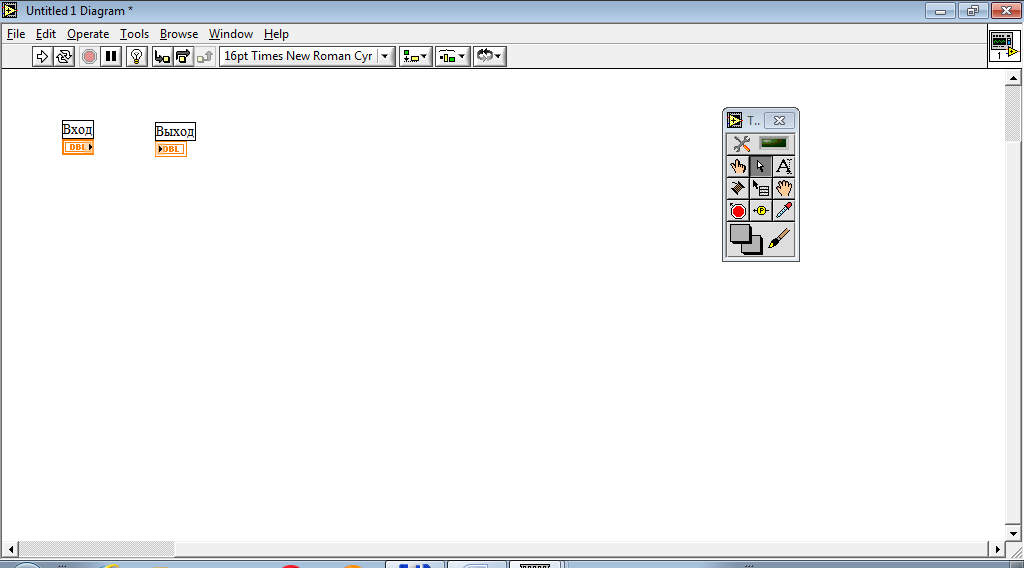


Рисунок 2.2.9 – Отображение тех же контроллера и индикатора на блок-диаграмме

Хорошо видно, что графические переменные однозначно соответствуют графическим элементам, размещённым ранее на передней панели. Решим поставленную простую задачу – соединим переменные между собой. Для чего выберем соединительный провод (Рисунок 2.2.10) и последовательно нажмём левой кнопкой мыши на мигающей правой границе входа и затем на мигающей левой границе выхода.

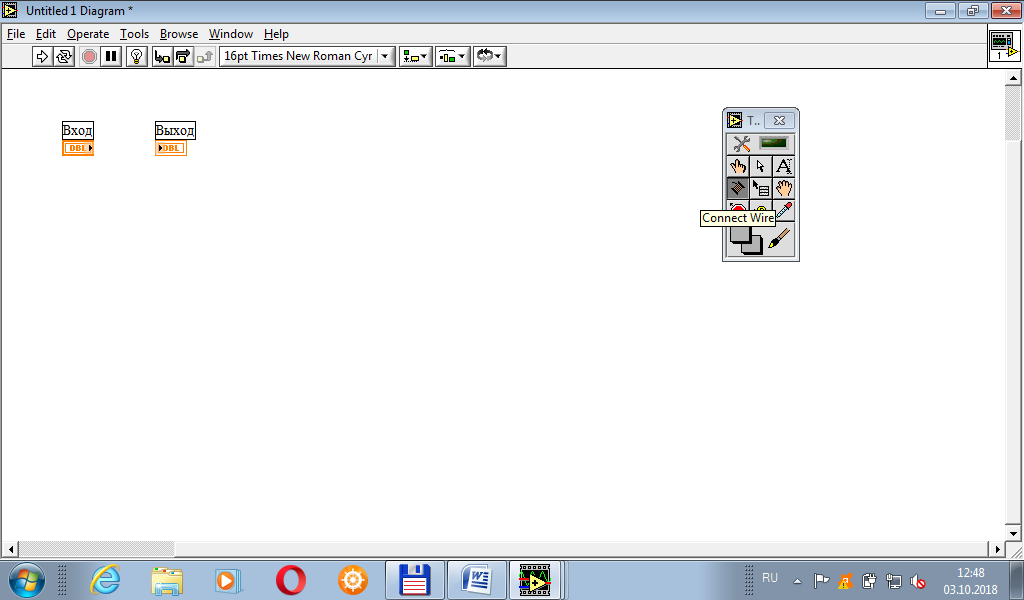


Рисунок 2.2.10 – Выбор инструмента соединительный провод (Connect Wire)

Результат соединения можно наблюдать на Рисунке 2.2.11.

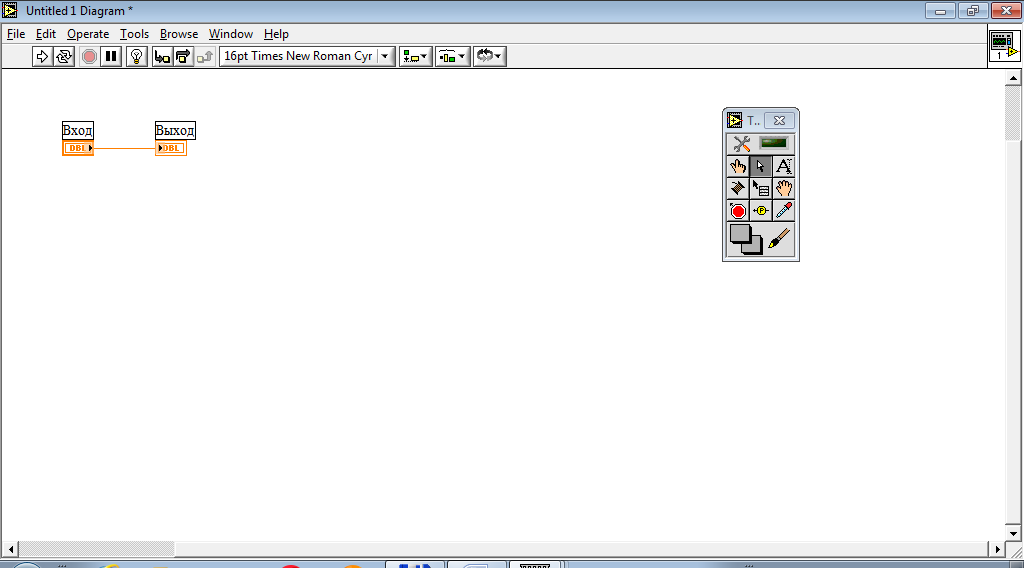


Рисунок 2.2.11 – Результат соединения контроллера и индикатора

Для обеспечения возможности изменения значения в контроллере, сменим соединительный провод на инструмент изменения значений, выглядящий как указательный палец правой руки (Рисунок 2.2.12).

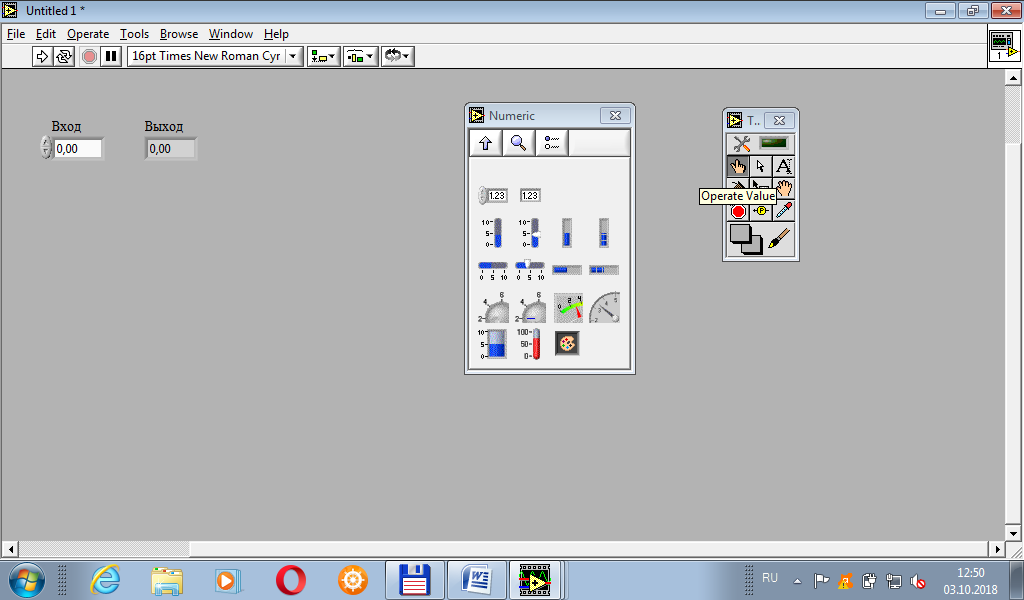


Рисунок 2.2.12 – Выбор инструмента изменения значений элементов (Operate Value)

С использованием данного инструмента увеличим значение на входе до «5,00», как показано на Рисунке 2.2.13 и запустим программу в режиме однократного исполнения.

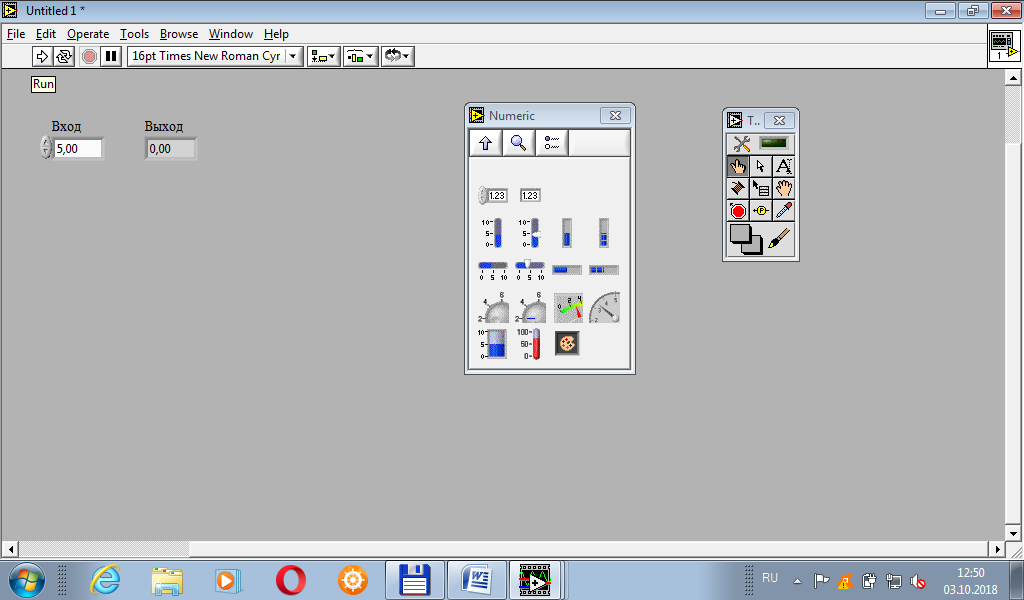


Рисунок 2.2.13 – Изменение значения, хранимого в контроллере и переход к разовому запуску программы на исполнения

Хорошо видно (Рисуное 2.2.14), что результатом работы программы становится настройка однозначного соответствия выхода со входом.

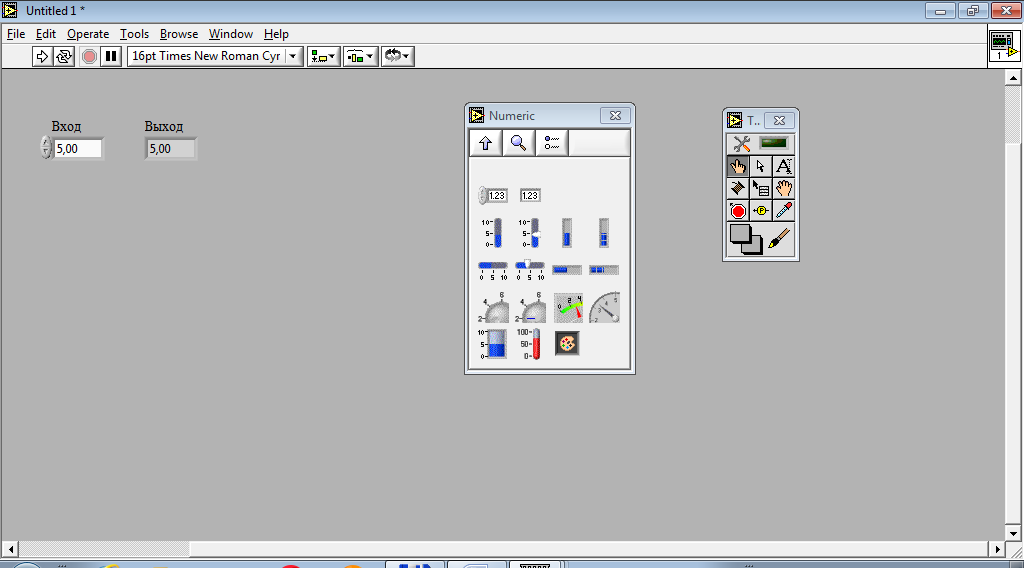


Рисунок 2.2.14 – Результат разового запуска программы на исполнение

Попробуйте проделать самостоятельно серию экспериментов, изменяя значения на входе и запуская программу в режиме однократного исполнения, а также запустив программу в режиме циклического исполнения и меняя значения на входе.

**Организация связи между численным и логическим элементами**

Далее рассмотрено решение несколько иной задачи, в которой проверяется логика наличия соответствия между входной информацией и некоторой наперёд заданной константой (пусть, например, её значение равно «5»). Соответствие, как известно, может быть, а также его может и не быть. Так наш ответ на вход должен получиться логическим. Это означает, что мы имеем возможность модифицировать ранее созданную программу виртуального прибора, исключив из неё численный индикатор и заменив его логическим индикатором. Все логические элементы содержатся в разделе логических или булевских (Рисунок 2.2.15).

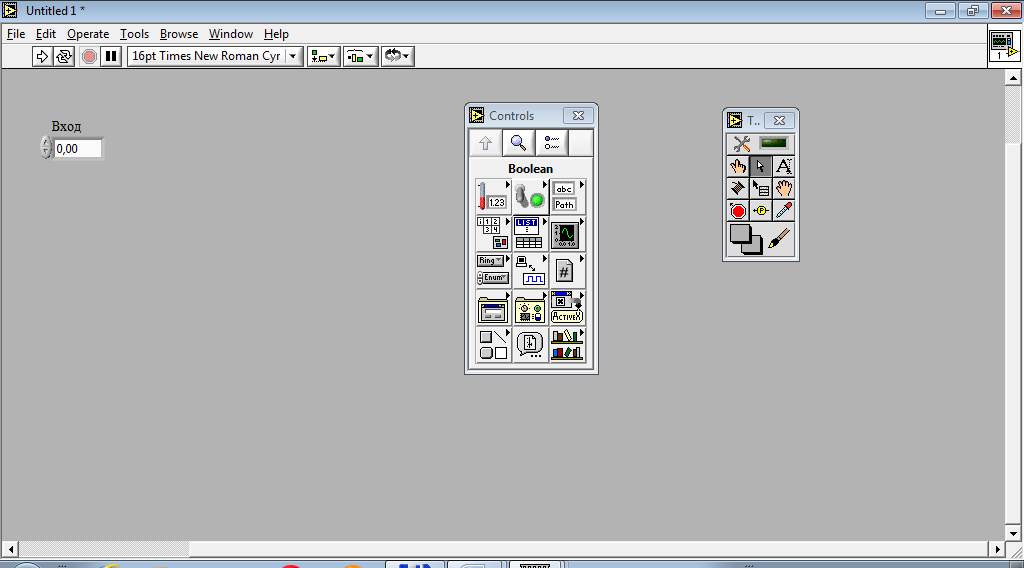


Рисунок 2.2.15 – Подсветка в панели элементов управления раздела логических элементов (Boolean)

В перечне логических элементов существуют два наиболее примечательных индикатора: квадратный (это только название и на деле – элемент прямоугольной формы) светодиод и круглый светодиод. По смысловой нагрузке эти элементы равноценны, потому размещаются на графическом пользовательском интерфейсе по усмотрению автора виртуального прибора (Рисунок 2.2.16).

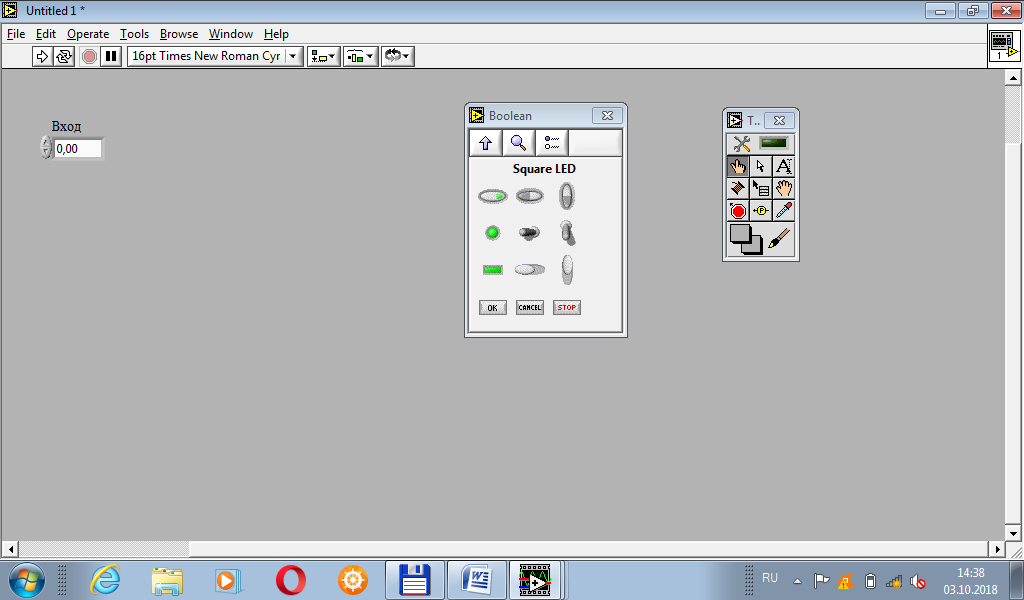


Рисунок 2.2.16 – Содержимое раздела логических элементов (Boolean) с подсветкой логического индикатора типа «прямоугольный светодиод» (Square LED)

Как правило, круглые элементы менее строги к вниманию оператора, прямоугольные элементы – более требовательны к вниманию (правило из области психологии человека). В то время как Информатика – строгая практическая дисциплина, требующая не менее строгой отчётности, применим к графическому пользовательскому интерфейсу логический индикатор прямоугольной формы (Рисунок 2.2.17).

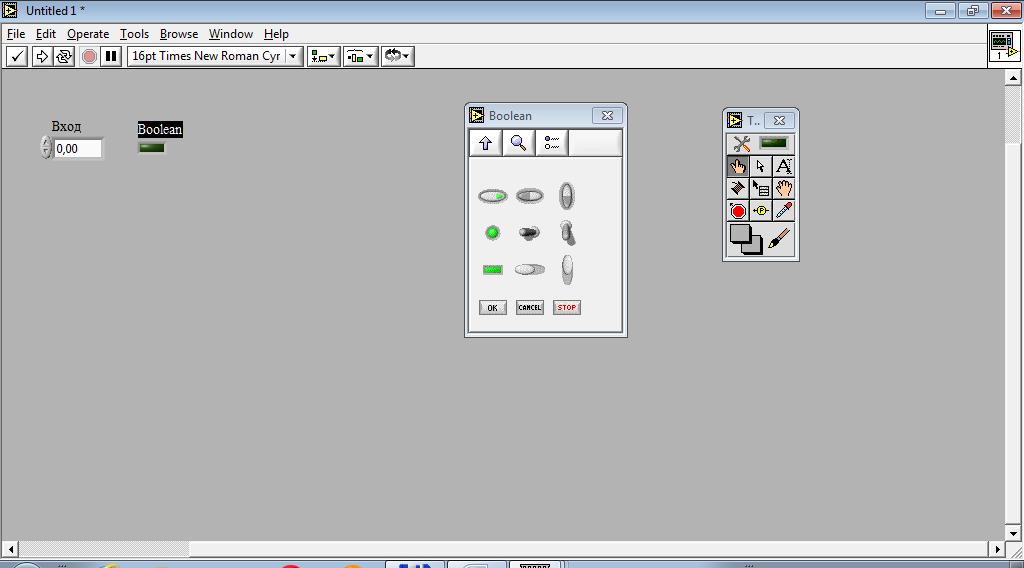


Рисунок 2.2.17 – Размещение прямоугольного светодиода на передней панели

Как и для случая с численным индикатором не стоит забывать о существовании правил вежливости программиста, а также правил обеспечения дружественности графического пользовательского интерфейса, потому введём значащее имя для нового логического индикатора (Рисунок 2.2.18).

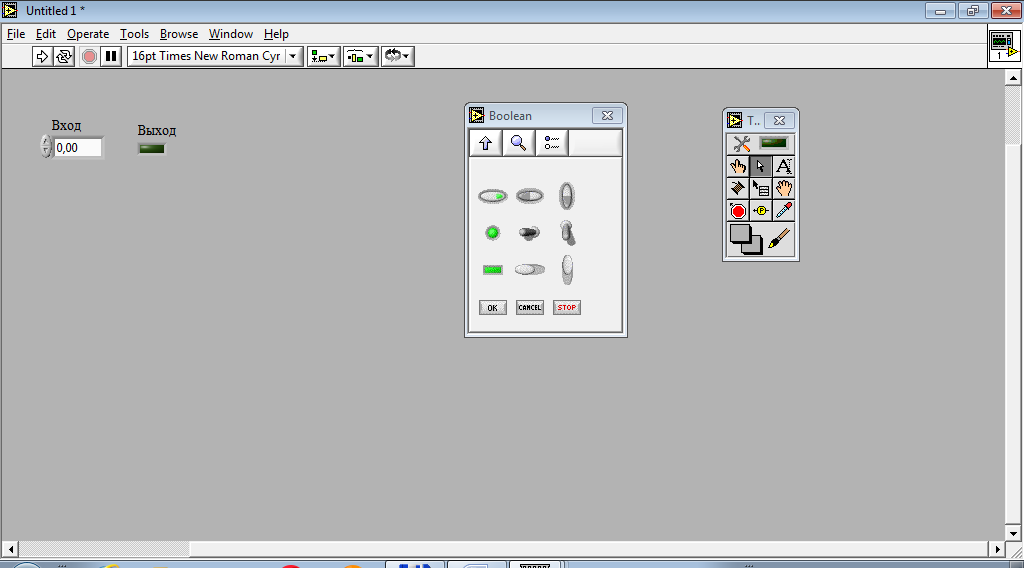


Рисунок 2.2.18 – Формирование значащего имени для переменной, связанной с логическим индикатором

Далее потребуется ещё раз обратиться к разделу численных элементов, но уже в рамках блок-диаграммы создаваемого виртуального прибора (Рисунок 2.2.19).

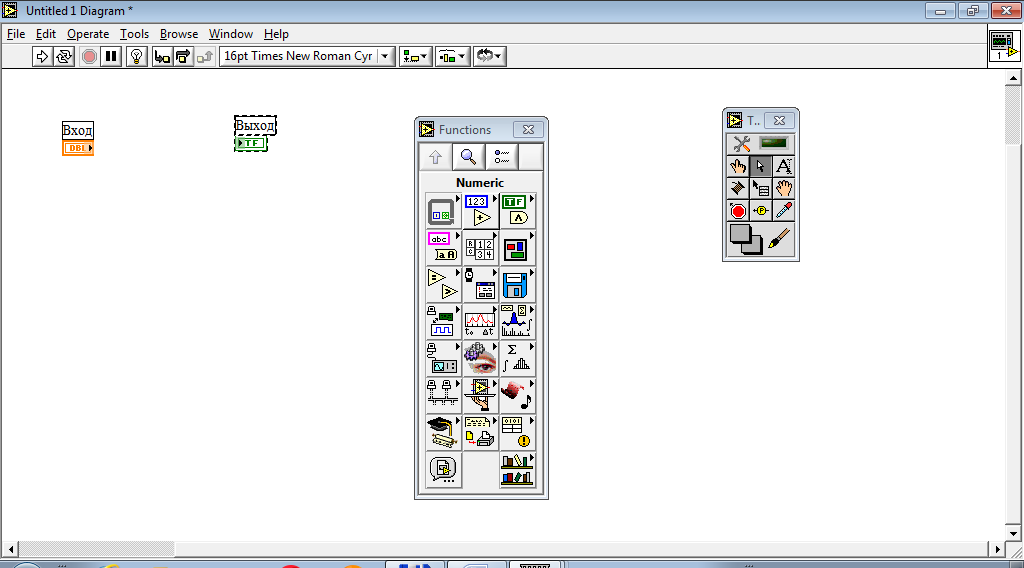


Рисунок 2.2.19 – Подсветка в панели функций раздела численных элементов (Numeric)

Из раздела понадобится численная константа, расположенная в левом нижнем углу окна с перечнем доступных для использования элементов данного раздела (Рисунок 2.2.20).

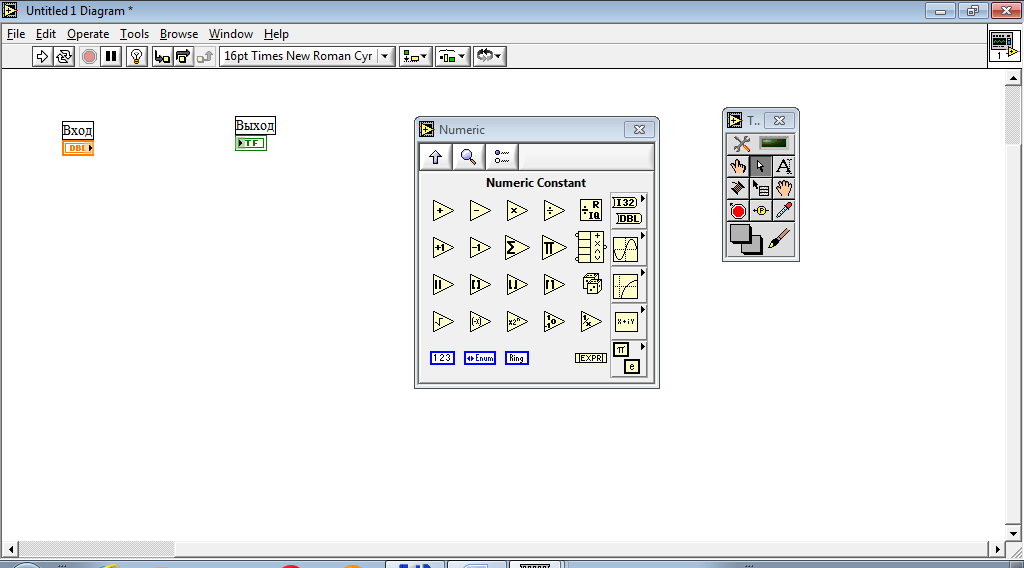


Рисунок 2.2.20 – Содержимое раздела численных функций (Numeric) с подсветкой численной константы (Numeric Constant)

Напомним, что любая константа характеризуется типом и значением, причём значение константы определяет её тип. Таким образом, пока в рамке указано целое число – это будет целочисленная константа (Рисунок 2.2.21), как только там появится значение с плавающей запятой – это будет вещественная константа. Причём, существует возможность задать константу «NaN» (Not a Number – не числовое значение), позволяющую в отдельных случаях отлавливать ошибки.

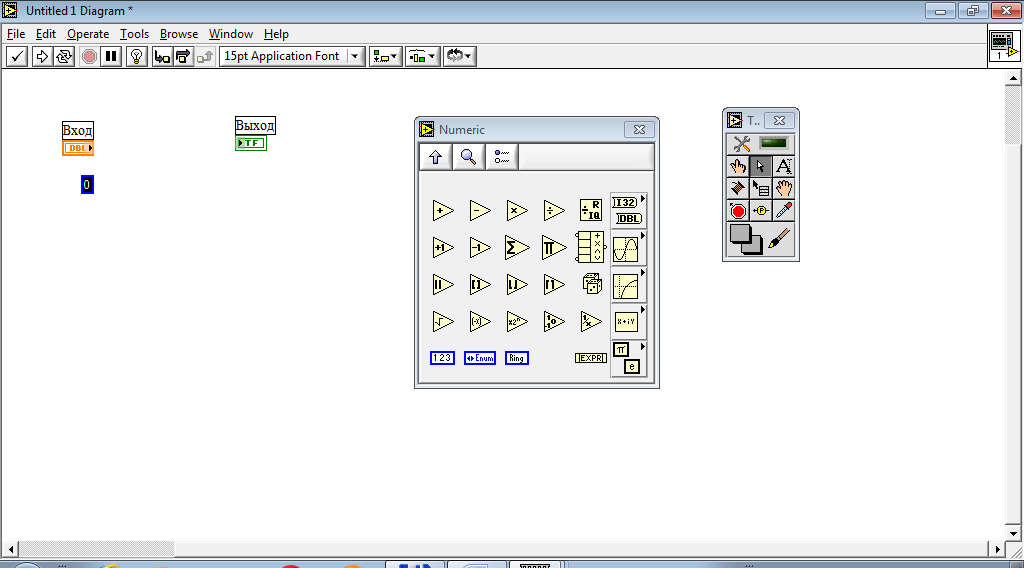


Рисунок 2.2.21 – Размещение целочисленной константы на блок-диаграмме

Не будем выходить за область целых чисел, потому в качестве константы в данном примере выставим значение равное «5» (Рисунок 2.2.22).

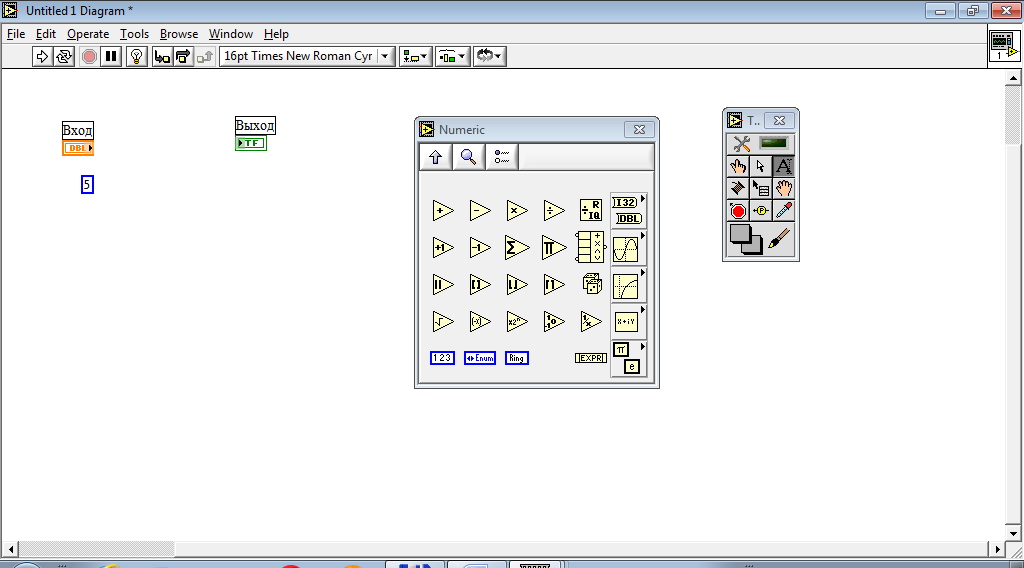


Рисунок 2.2.22 – Изменение значения целочисленной константы с «0» на «5»

После этого следует перейти к сравнению значения переменной, задаваемому численным контроллером, с константой, для чего обратиться в панели функций области блок-диаграммы к разделу элементов сравнения (Рисунок 2.2.23).

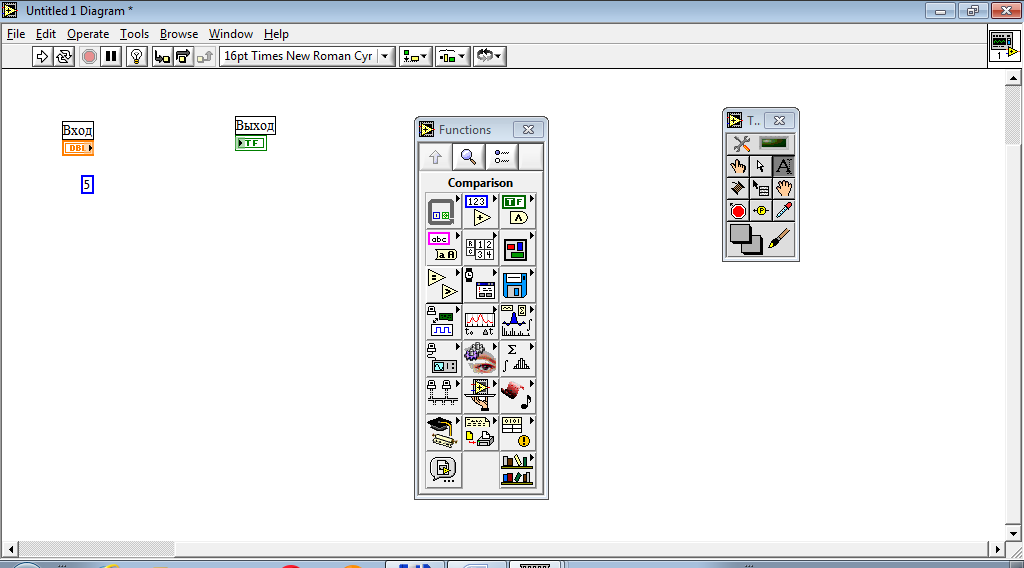


Рисунок 2.2.23 – Подсветка в панели функций раздела элементов сравнения (Comparison)

Для всех случаев, где сравнение требуется со значениями, отличными от нуля требуются на входе два операнда, для случаев сравнения с нулём – один операнд. Далее рассматривается простейший случай сравнения на точное соответствие элементов – их эквивалентность (Рисунок 2.2.24). Соответствующий элемент располагается в левом верхнем углу окна, содержащего элементы, доступные для размещения на блок-диаграмме из этого раздела.

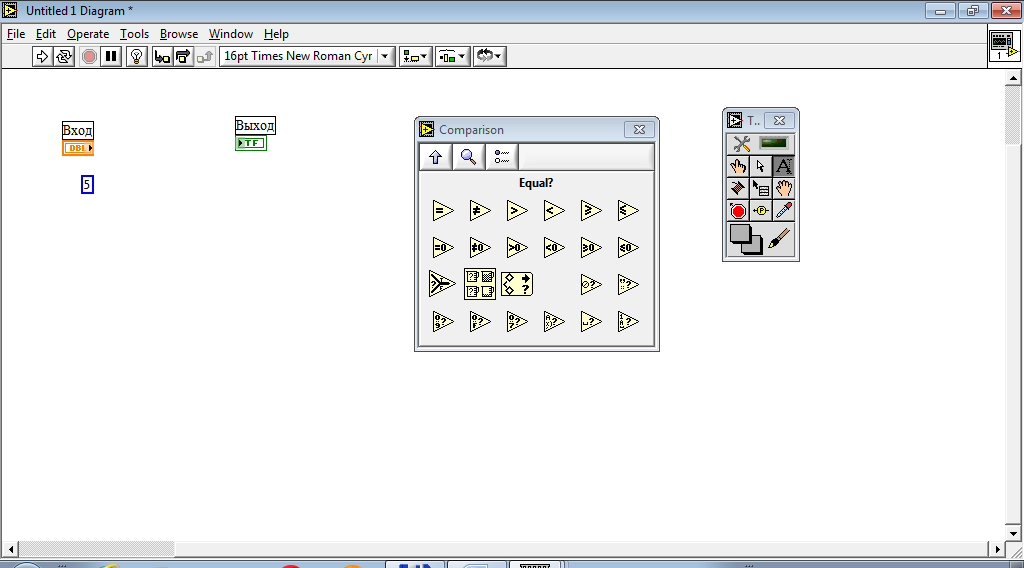


Рисунок 2.2.24 – Содержимое раздела функций сравнения (Comparison) с подсветкой оператора однозначного соответствия / проверки на эквивалентность (Equal?)

Итого размещаем элемент на блок-диаграмме и видим, что на ней теперь всего достаточное, остаётся лишь настроить связи между элементами (Рисунок 2.2.25).

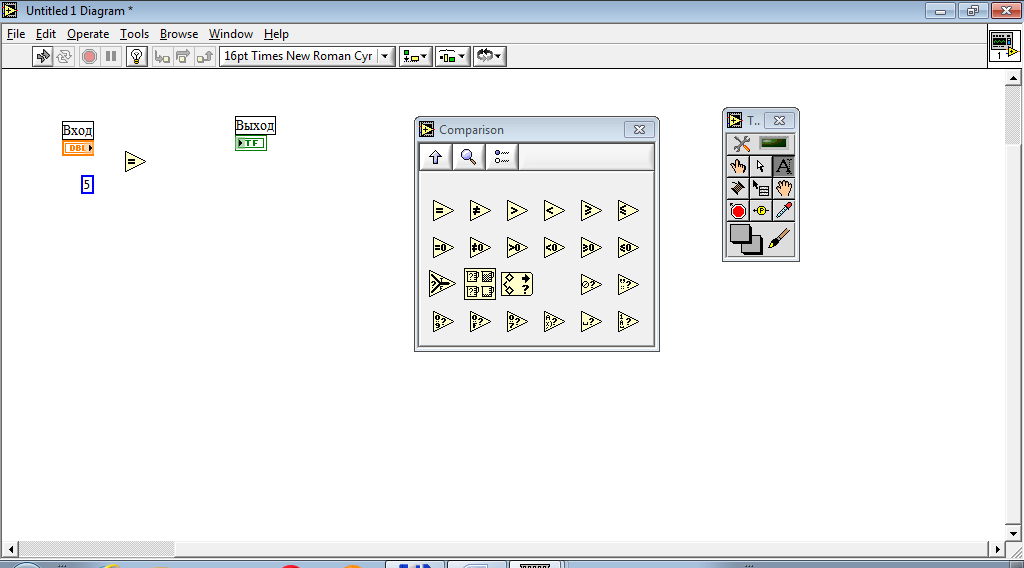


Рисунок 2.2.25 – Размещение оператора однозначного соответствия на блок-диаграмме

Не стоит забывать о правилах вежливости программиста: написанный код должен обладать читаемостью. Для графического кода в этом смысле существует куда больше дополнений, ограничивающих понятие читаемости. Так, например, необходимо стремиться к минимизации изломов линий связи – они отвлекают внимание на себя, а также стремиться к минимизации занимаемого графическим кодом пространства. Последнее пришло, скорее, из электроники, где при проектировании печатных плат стараются минимизировать количество использованного проводника, поскольку каждый сантиметр потраченного медного провода повышает стоимость разработки. В работе, естественно, создаётся только модель, но о минимизации затрат на токопроводящий материал нужно привыкать думать и помнить уже на данном этапе (Рисунок 2.2.26). Если выработается привычка минимизации занимаемого пространства элементами схемы в рамках Информатики – это значимый шаг для решения практических инженерных задач в будущем.

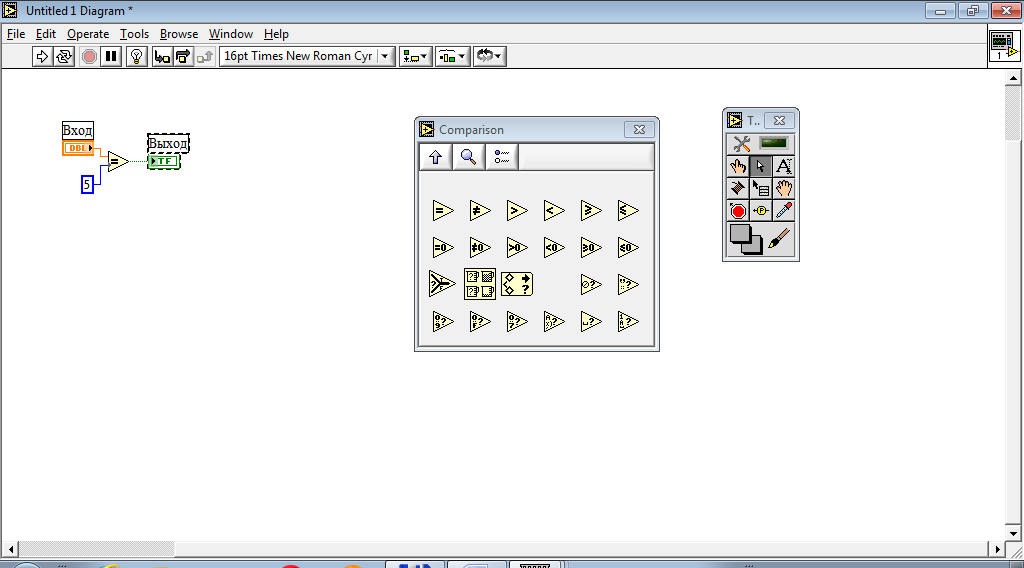


Рисунок 2.2.26 – Соединение размещённых элементов между собой

Протестируем созданный, очень простой виртуальный прибор. В алгоритме имеет место разветвляющийся вычислительный процесс, в котором существуют лишь два исхода, потому для доказательства работоспособности нового программного обеспечения необходимо подобрать и отработать два тестовых примера. Подадим для начала на вход численного контроллера виртуального прибора значение равное «2». В этих условиях логическое выражение отвечает ложным результатом и лампочка не загорается (Рисунок 2.2.27).

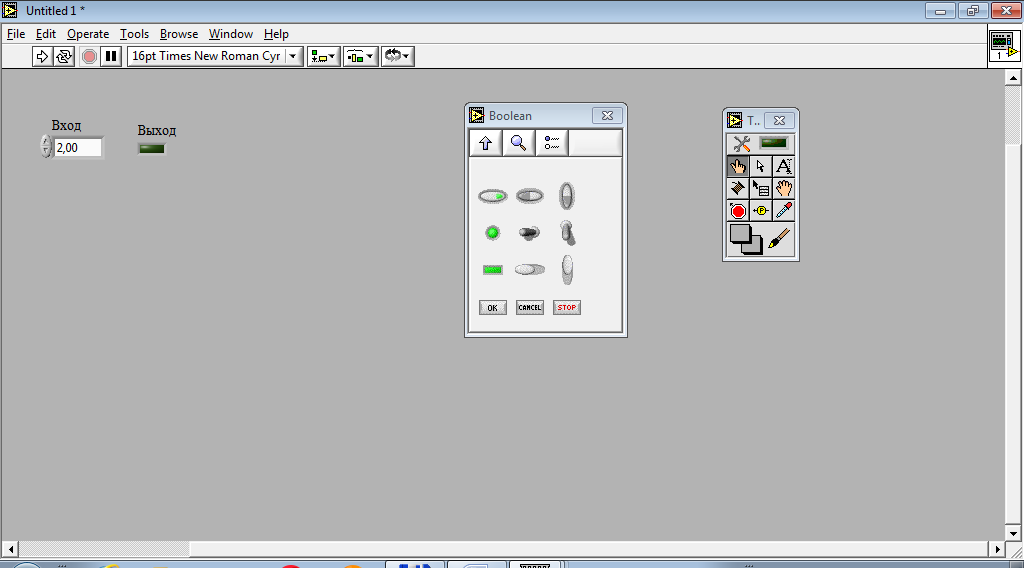


Рисунок 2.2.27 – Отклик лампы на значение «2,00», поданное на вход

Затем подадим на вход численного контроллера виртуального прибора значение, равное «5». Здесь логическое выражение отвечает истинным результатом – лампочка загорается.

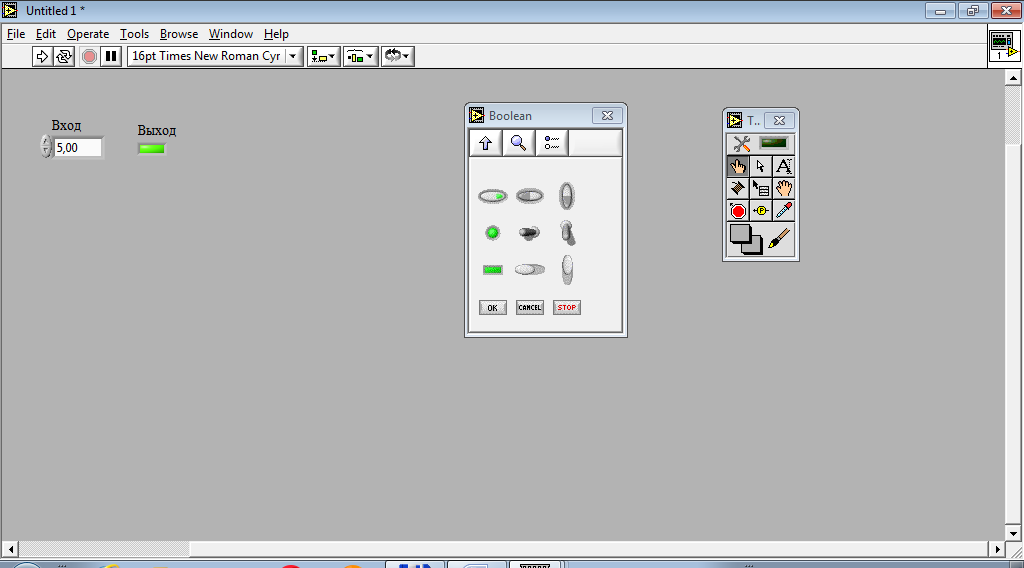


Рисунок 2.2.28 – Отклик лампы на значение «5,00», поданное на вход

Рассмотренных выше примеров достаточно для предоставления обучающимся базовых умений и навыков для решения поставленной задачи.

## 2.3 Пример выполнения задания

### 2.3.1 Общая часть

### 2.3.2 Индивидуальная часть (переменные)

### 2.3.3 Индивидуальная часть (константы)

## 2.4 Варианты индивидуального задания

### 2.4.1 Переменные

В дополнительном файле \*.vi создать развитие структуры «Простого Калькулятора» посредством добавления в неё элементов ввода и вывода информации, а также индикации (в случае необходимости), достаточных для демонстрации результата расчёта выражения, заданного по варианту.

Без острой на то необходимости графический код не усложнять! Например, операция возведения в степень не должна вызывать потребности у автора виртуального прибора в использовании специальной функции возведения числа в некоторую степень N.

Таблица 2.4.1.1 –

|  |  |  |
| --- | --- | --- |
| **Вар.** | **Выражение** | **Логический индикатор** |
| **1** | (a + b + c + d) \* (– a – b – c – d) | Не требуется |
| **2** | (a + b + c)2 | Не требуется |
| **3** | (a – b + c – d) \* (– a + b – c – d) | Не требуется |
| **4** | (b / a) – (d / c) | Требуется |
| **5** | (a + (b – d) \* c)2 | Не требуется |
| **6** | (d / (b + a)) – (c / (d + a)) | Требуется |
| **7** | 3 \* a + b2 – c + 5 \* d3 | Не требуется |
| **8** | –((b + c) / (a + d)) | Требуется |
| **9** | (a + (b \* c) / d) / (a / (b \* d) – c) | Требуется |
| **10** | ((b + 2) \* (a + 4) / (d + 6) \* (c + 8)) | Требуется |
| **11** | –((b + c) \* (a + d) / c2) | Требуется |
| **12** | 10 \* a + 15 \* b + 20 \* c + 25 \* d | Не требуется |
| **13** | 7 \* (a + b + c + d) \* (– a – 7 \* b – c – d) | Не требуется |
| **14** | (a + b + c + d) / (– a – b – c – d) | Требуется |
| **15** | (a + b + c + d)3 | Не требуется |
| **16** | (a – b + c – d) / (– a + b – c – d) | Требуется |
| **17** | (a / b) + (c / d) | Требуется |
| **18** | (a \* (b – d) + c)2 | Не требуется |
| **19** | (a / (c + a)) + (b / (d \* a)) | Требуется |
| **20** | ((b / a) / (d / c)) – ((a / b) / (c / d)) | Требуется |
| **21** | 5 \* a + b2 + c – 3 \* d3 | Не требуется |
| **22** | –((a + b) / (d – c)) | Требуется |
| **23** | (a – (b \* c) / d) \* (a / (b \* d) + c) | Требуется |
| **24** | 10 \* d – 5 \* c + 20 \* a – 15 \* b | Не требуется |
| **25** | –((b – c) \* (a + d) / (c + 4)) | Требуется |
| **26** | 3 \* (a + c + d) \* (–7 \* b – d) | Не требуется |
| **27** | (a + b + c + d)4 | Не требуется |
| **28** | (a \* (c + a)) / (b \* (d – a)) | Требуется |
| **29** | (a / b)2 + (c / d)2 | Требуется |
| **30** | (a2 + b2 + c2) | Не требуется |
| **31** |  |  |
| **32** |  |  |
| **33** |  |  |
| **34** |  |  |
| **35** |  |  |

### 2.4.1 Константы

В ещё одном дополнительном файле \*.vi создать независимый от двух предыдущих виртуальных приборов иной виртуальный прибор, который смоделировал бы правильный ответ на сложное выражение, заданное по варианту.

Графический пользовательский интерфейс должен содержать в своём составе в удобной для чтения форме исходное сложное выражение, заданное по варианту, а также индикатор, выдающий правильный ответ на упомянутое сложное выражение.

## 2.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

## 2.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 2.7 Блок-схемы алгоритмов для типовых решений

**3 Задание «Разработка виртуального прибора "Конвертор температуры"»**

В пакете прикладных программ *National Instruments* *LabView* создать учебный виртуальный прибор «Термометры», моделирующий конвертацию температуры, заданной в градусах Цельсия, к другим единицам измерения, таким как градусы:

* Кельвина,
* Фаренгейта,
* Ранкина,
* Реомюра,
* Рёмера,
* Ньютона,
* Делиля.

Настроить шкалы деления виртуальных термометров таким образом, чтобы предельные значения шкалы основного термометра, задающего градусы Цельсия, совпадали с предельными значениями дополнительных термометров, показывающих иные единицы измерения **по графическому уровню**.

Для повышения индивидуальности выполняемых обучающимися работ по созданию виртуальных приборов диапазоны шкал термометров в градусах Цельсия задаются согласно таблице вариантов.

И, поскольку, прибор является лабораторной установкой – он должен содержать на передней панели заголовок, авторство и год выпуска.

**Внимание!** Для случаев, когда стандартных размеров термометра недостаточно для нормального отображения на шкале заданного диапазона, необходимо принудительно увеличить размеры термометра посредством использования инструмента позиционирования элементов (*Position/Size/Select*).

В процессе проектирования виртуального прибора необходимо решить вопрос рационального размещения функциональных элементов, как на графическом пользовательском интерфейсе – передней панели, так и на блок-диаграмме с кодом. Всё должно быть читаемым, обладать минимальным количеством пересечений линий связи, а также быть понятным не только разработчику виртуального прибора, но и самому-самому неопытному пользователю программного обеспечения.

На блок-диаграмме все функциональные, то есть значащие пересечения линий связи (узлы) обозначить точками. Для этого в меню перейти к настройкам «*Tools > Options…*», далее в выпадающем списке перейти к настройкам блок-диаграммы (*Block Diagram*) и выставить галочку напротив пункта «*Show dots at wire junctions*».

Правила конвертации одних температурных величин в другие представлены в разделе «соотношения, необходимые для выполнения работы».

Чётные варианты как в общей, так и в индивидуальной части работы используют в качестве задающего элемента термометр опорной измерительной системы (в общей части опорной системой измерения являются градусы Цельсия);

нечётные варианты используют в качестве задающего элемента – числовой контроллер опорной системы.

По итогам выполнения работы сдаются строго три файла:

- отчёт, выполненный в текстовом редакторе *Microsoft Office Word* (*\*.doc* или *\*.docx*);

- файл виртуального прибора *National Instruments LabView* (*\*.vi*) по общей части работы;

- файл виртуального прибора *National Instruments LabView* (*\*.vi*) по индивидуальной части работы.

Отправленные поодиночке файлы проверке не подлежат. При отсутствии одного из упомянутых файлов зачёт по заданию не выставляется.

\*Для выполнения всех работ рекомендуется скачать старую версию (6.1) пакета прикладных программ *National Instruments LabView* (). Обучающимся,

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «10».

## 3.1 Цель работы

ознакомление с интерфейсом пакета прикладных программ *National Instruments* *LabView*, его элементами, настройками и арифметическими операциями. Закрепление навыков составления блок-схем алгоритмов к авторским прикладным программам с разбиением этих блок-схем по процедурам и функциям.

## 3.2 Полезные соотношения для выполнения задания

1. Перевод из градусов Цельсия в градусы Кельвина:;

2.1. Перевод из градусов Фаренгейта в градусы Цельсия: ;

2.2. Перевод из градусов Цельсия в градусы Фаренгейта: ;

3. Соответствие градусов Цельсия градусам Ранкина: 

4. Соответствие градусов Цельсия градусам Реомюра: 

5. Соответствие градусов Цельсия градусам Рёмера: ;

6. Соответствие градусов Цельсия градусам Ньютона: ;

7. Перевод из градусов Цельсия в градусы Делиля: 

**Пример настройки шкалы к единому уровню**

Рассмотрим задачу градуировки виртуального термометра. Переместим значение управляющего воздействия к нижнему уровню. Хорошо видно, что на втором термометре при этом получено значение, равное «100». Достигнута в точности верхняя граница. Это обстоятельство является, своего рода, везением и ничего менять здесь не требуется (Рисунок 23).

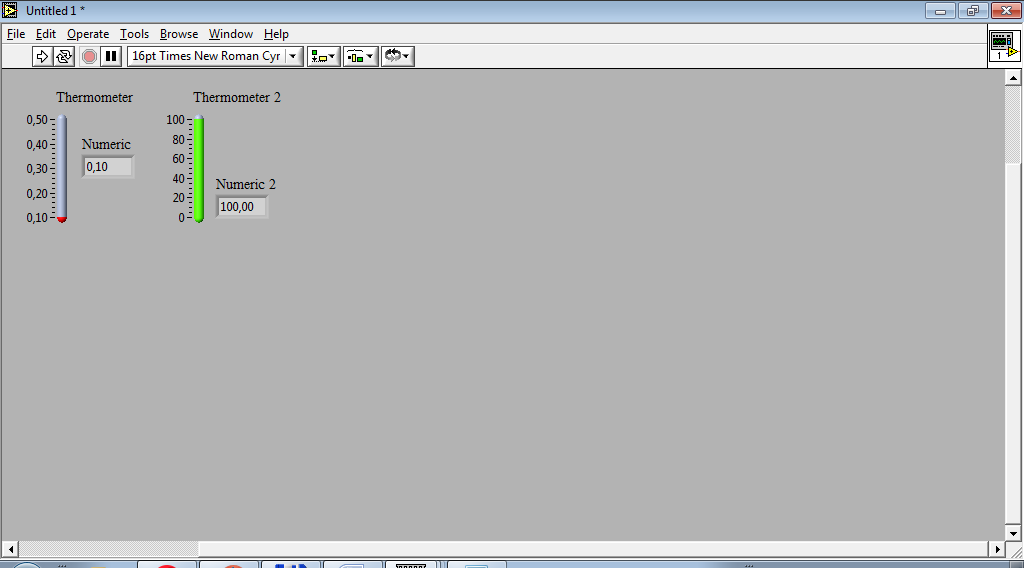
****

Рисунок 23 – Проверка нижнего уровня диапазона по активному параметру – прибор не зашкаливает. Это совпадение, случайность, что не потребовалось вносить какие-либо изменения

Теперь рассмотрим поведение системы при достижении управляющим сигналом верхней границы. Хорошо видно, что на втором, зелёном термометре остался участок шкалы, который никогда не будет достигнут при заданном диапазоне входного сигнала. Так нам удалось обнаружить наличие избыточности (Рисунок 24). Избыточность необходимо устранять.

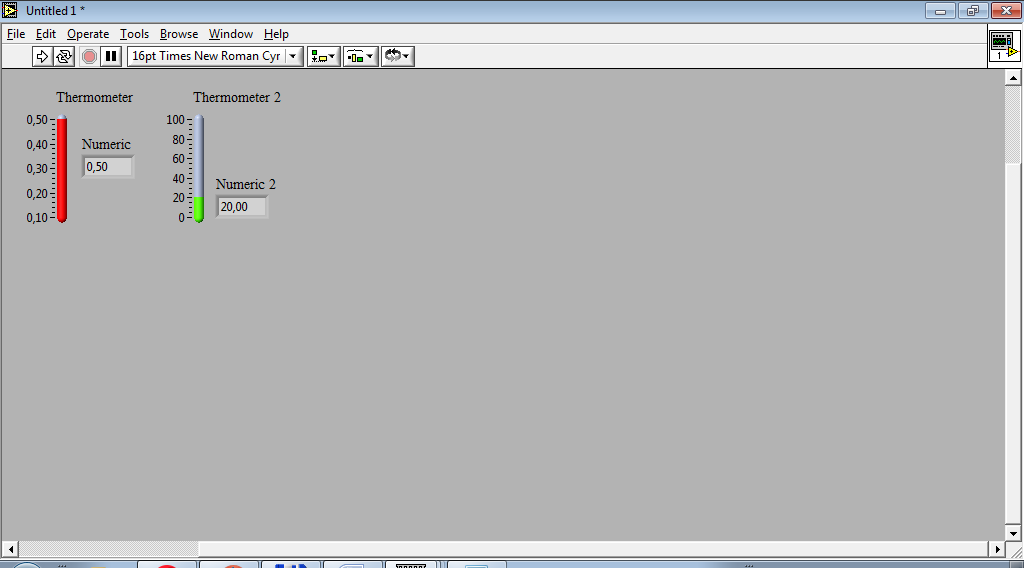
****

Рисунок 24 – Проверка верхнего уровня диапазона по активному параметру – прибор обладает избыточностью по шкале

Выполненная корректировка гарантирует отсутствие зашкаливания, а также предоставляет неизбыточность проводимых измерений (Рисунок 25).

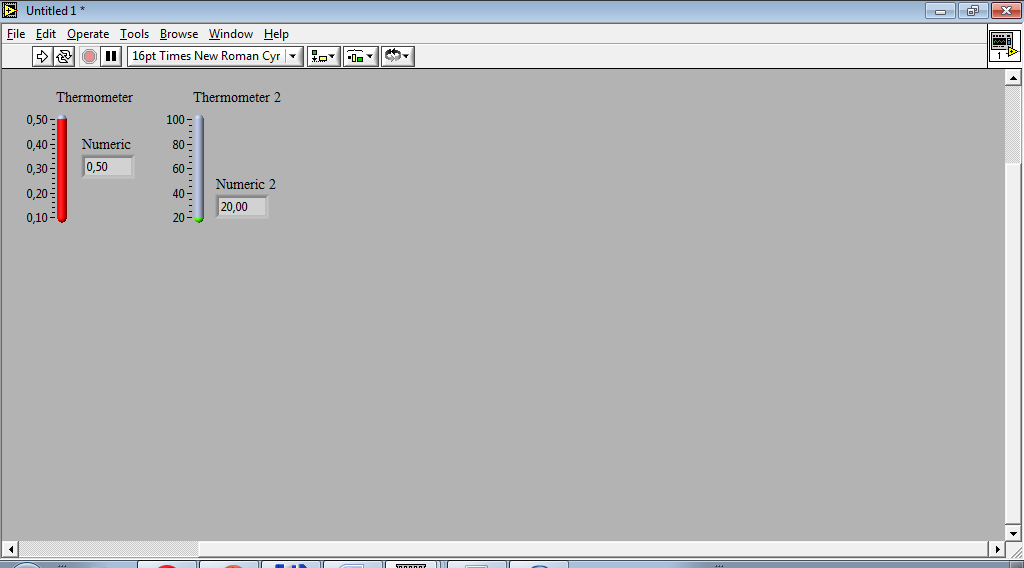
****

Рисунок 25 – Настроено соответствие шкал, иначе выполнена градуировка виртуального прибора

Для особо пытливых инженерных умов в *National Instruments LabView* присутствует возможность настройки заполнителя шкалы (*Fill Options*). Потому если настроить заполнение к максимуму (*Fill To Maximum*), показанное на Рисунке 26.

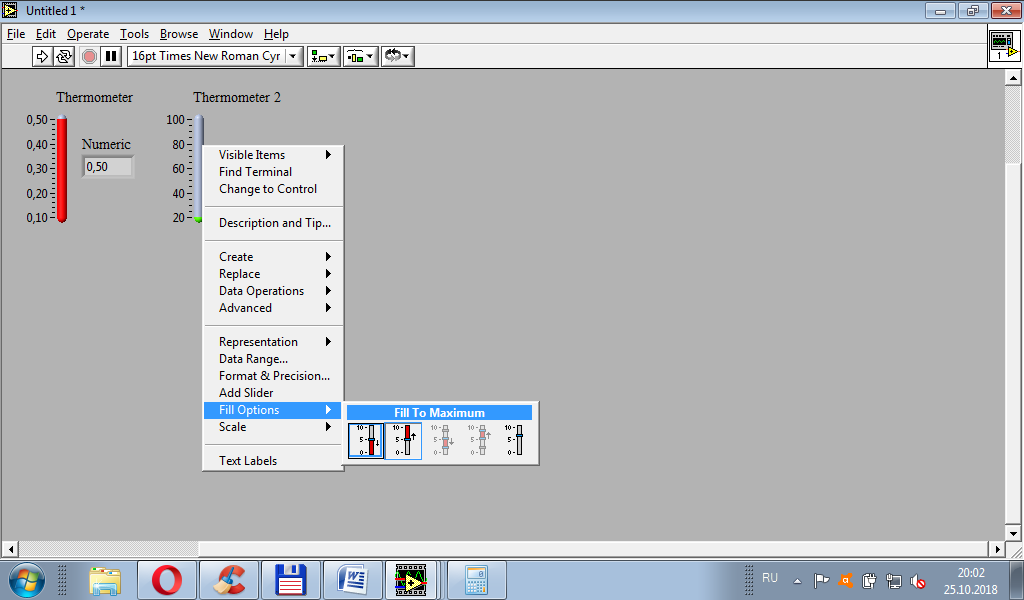
****

Рисунок 26 – Для настройки соответствия уровней заполнения при обратно пропорциональной зависимости в контекстном меню выбрана иная настройка заполняемости (*Fill Options*)

Можно получить в каком-то смысле идентичные по заполнителю термометры, объединённые обратной пропорциональной зависимостью (Рисунок 27).

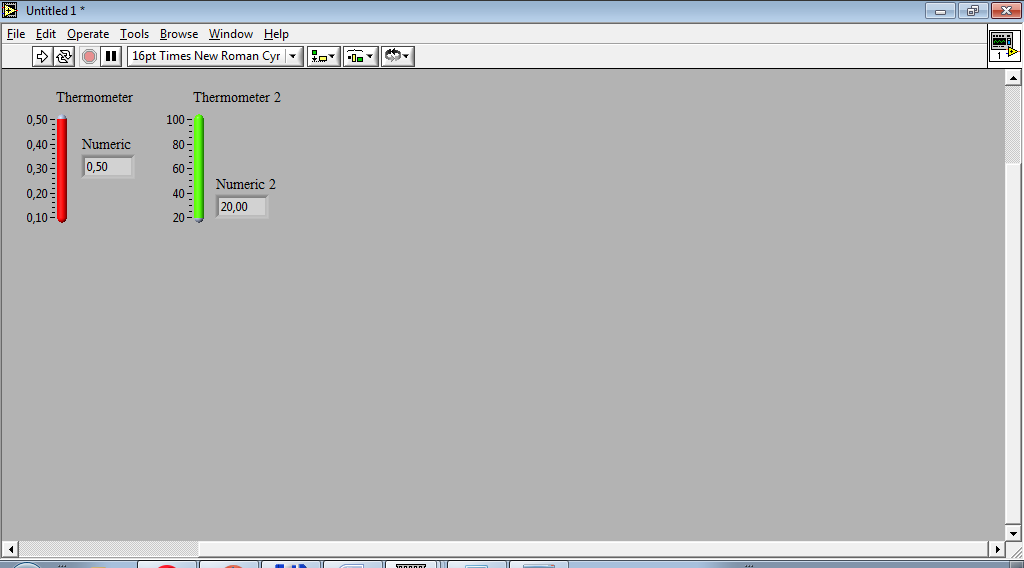
****

Рисунок 27 – Результат полноценной градуировки шкал элементов виртуального прибора

**Пример настройки начального значения задающего элемента:**

Часто для удобства работы с виртуальными приборами требуется их начальное приближение, отличное от минимального или нулевого значения. Подход удобен при отладке, чтобы каждый раз при новом открытии файла виртуального прибора не задавать одно и то же, нужное для тестирования значение. Этот подход в состоянии существенно сэкономить время.

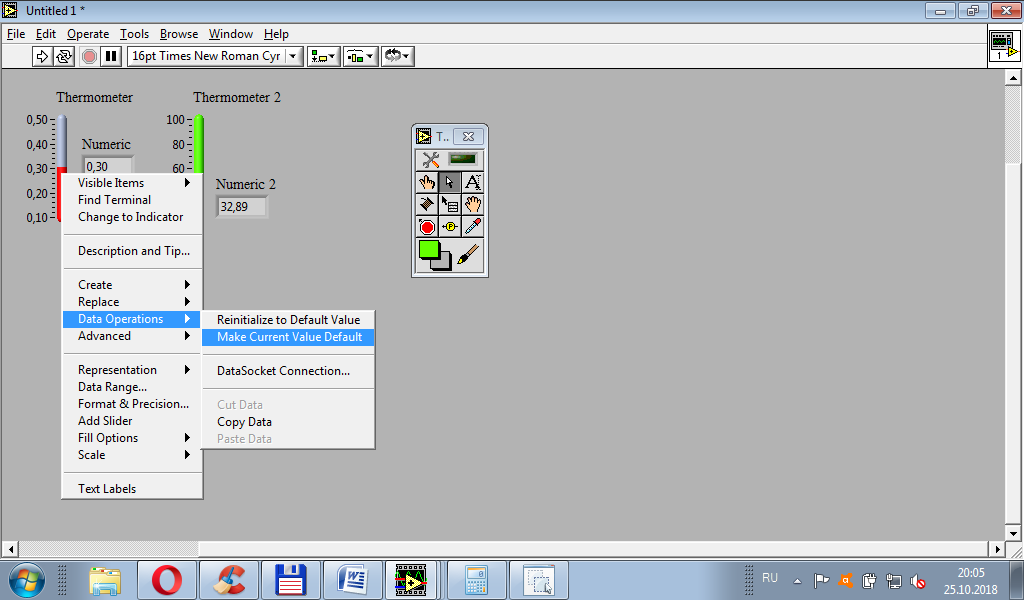
****

Рисунок 28 – Контекстное меню красного термометра. Фиксация выставленного значения заданным по умолчанию (*Make Current Value Default*)

Результат открытия файла виртуального прибора после изменения значения, установленного по умолчанию показано на Рисунке 29. Все значения сброшены в минимум или ноль, кроме значения красного термометра.

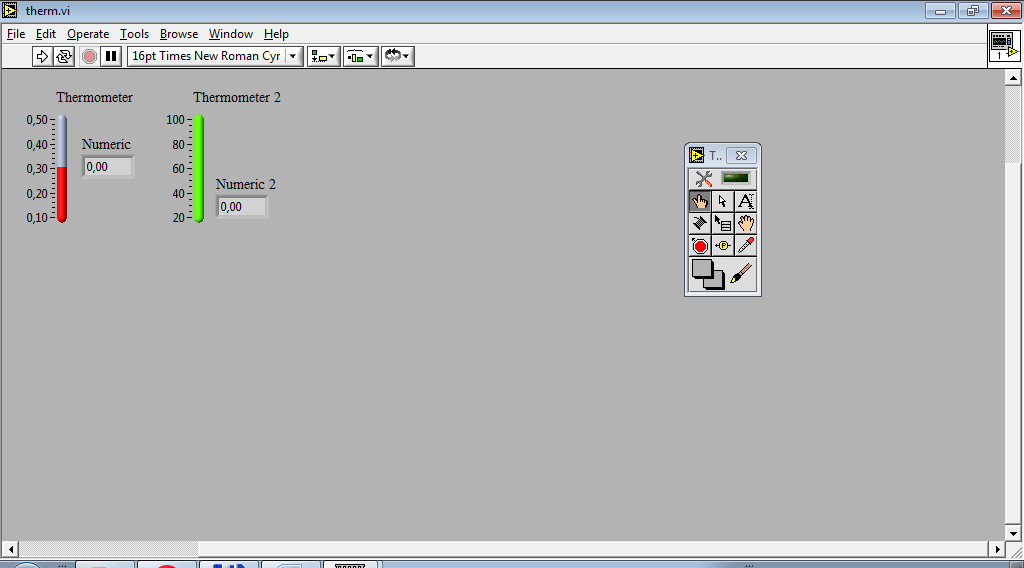
****

Рисунок 29 – Результат отображения по

Представленных в примерах сведений в сочетании с ранее приобретёнными навыками проектирования виртуальных приборов достаточно для выполнения обучающимися поставленной задачи.

## 3.3 Пример выполнения задания

Элемент «термометр» в пакете прикладных программ *National Instruments LabView* размещается разделе численных элементов (*Numeric*). Его место показано на Рисунке 1.

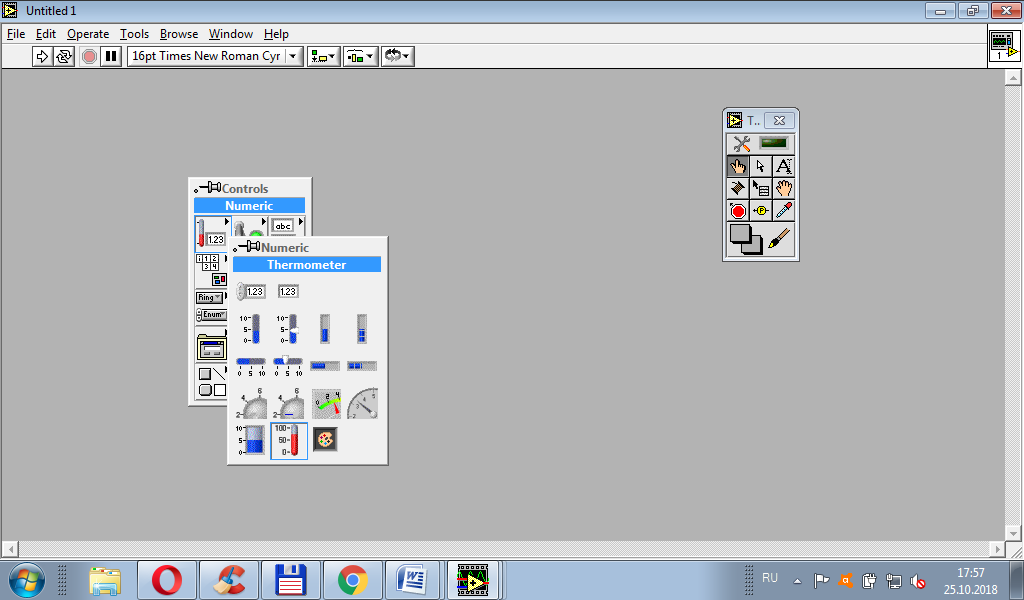
****

Рисунок 1 – Демонстрация с маркировкой места расположения термометра в разделе численных интерфейсных элементов управления (*Numeric Controls*)

По умолчанию шкала термометра выставлена от 0 до 100 (Рисунок 2).

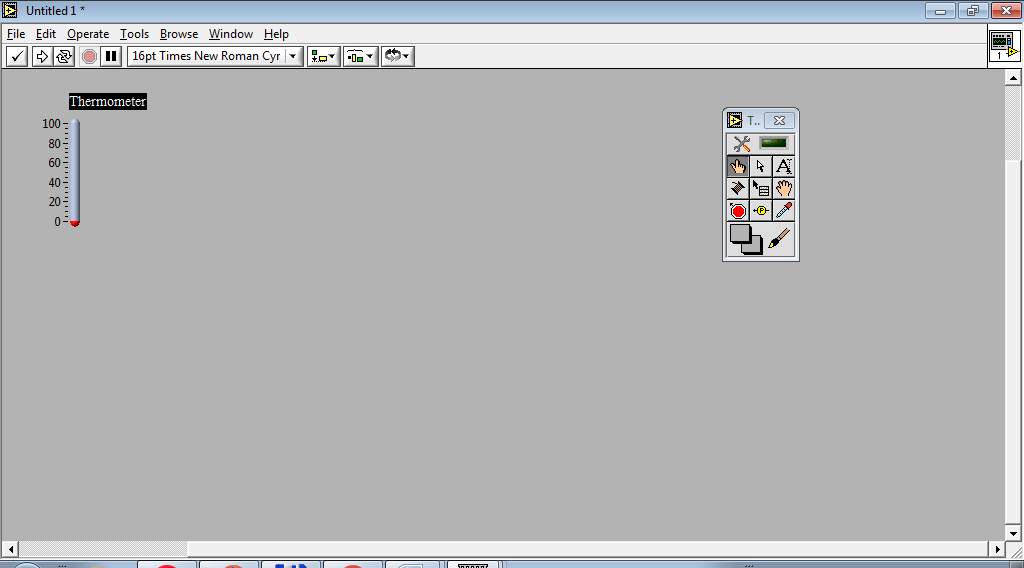
****

Рисунок 2 – Начальное представление интерфейсного элемента «Термометр» (*Thermometer*)

На блок-диаграмме термометр по умолчанию – элемент индикации (Рисунок 3). Это не лишено логики и вполне естественно, однако, в *LabView* разработчику и конечному пользователю предоставлена возможность организовывать управление посредством самого термометра.

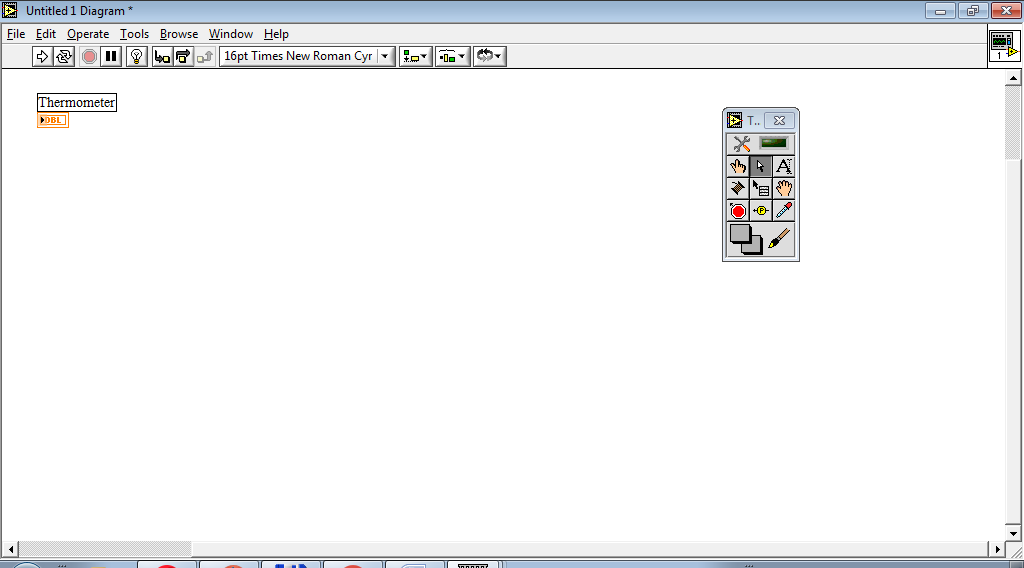
****

Рисунок 3 – По умолчанию на блок-диаграмме термометр является индикатором

Для перевода термометра в режим контроллера, на соответствующий ему элемент на блок-диаграмме необходимо нажать правой кнопкой мыши с целью вызовы контекстного меню (Рисунок 3), в котором присутствует позиция «Изменить на контроллер» (*Change to Control*). Его необходимо выбрать.

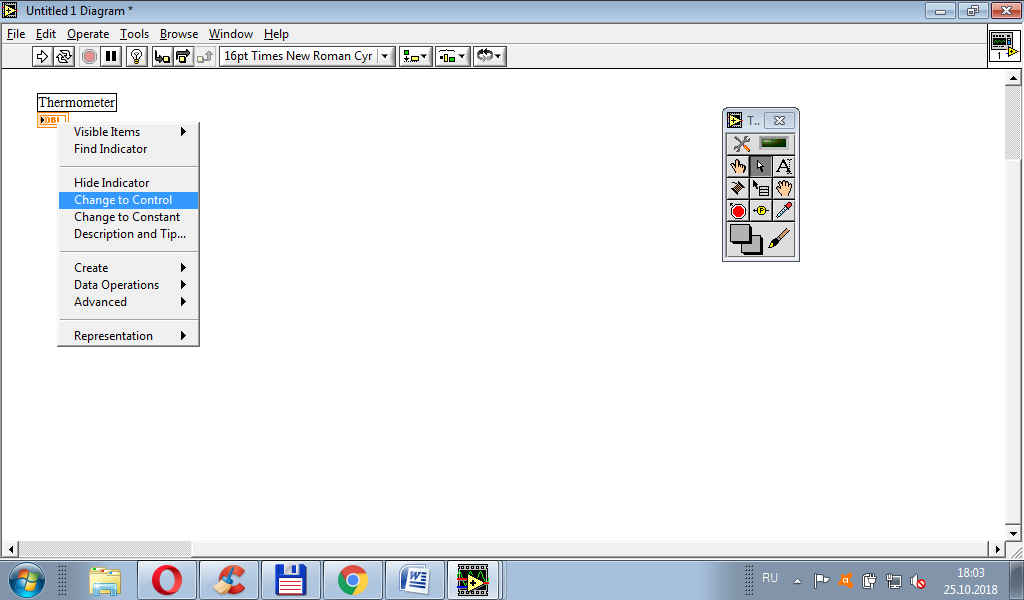
****

Рисунок 4 – Возможность смены «полярности» виртуального термометра с индикатора на контроллер (*Change to Control*) в контекстном меню

Вслед за выбором операции в контекстном меню образ элемента на блок-диаграмме изменит «полярность» (Рисунок 5).

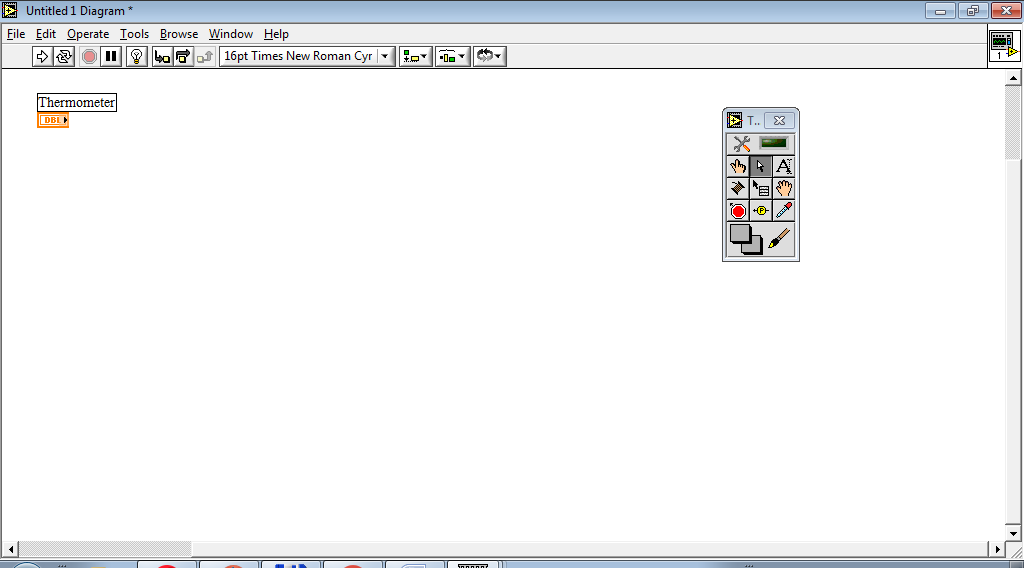
****

Рисунок 5 – Результат перехода термометра к состоянию контроллера на блок-диаграмме

Далее на передней панели виртуального прибора для удобства отслеживания изменений в численных значениях, проводимых по шкале термометра, разместим численный индикатор, который однозначно свяжем с термометром (Рисунок 6).

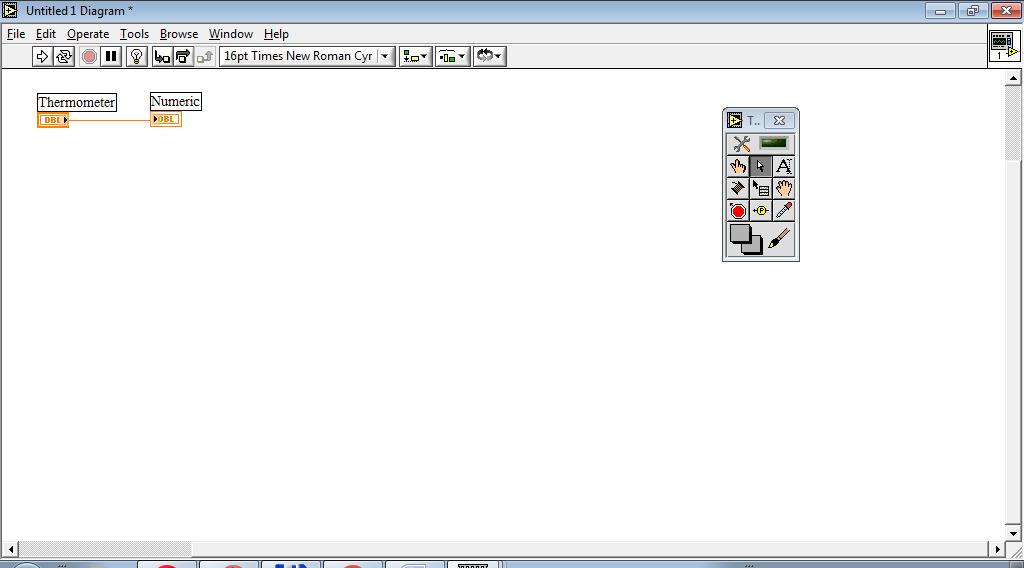
****

Рисунок 6 ­– Термометр-контроллер, однозначно связанный с числовым индикатором

Посредством использования инструмента для изменения значения (*Operate Value*) сместим значение заполнителя термометра к середине и запустим программу в режиме однократного исполнения (*Run*). Результат данной манипуляции отображён на Рисунке 7. Хорошо видно, что в отсутствии достаточной детализации по шкале, можно получать вывод довольно точных значений.

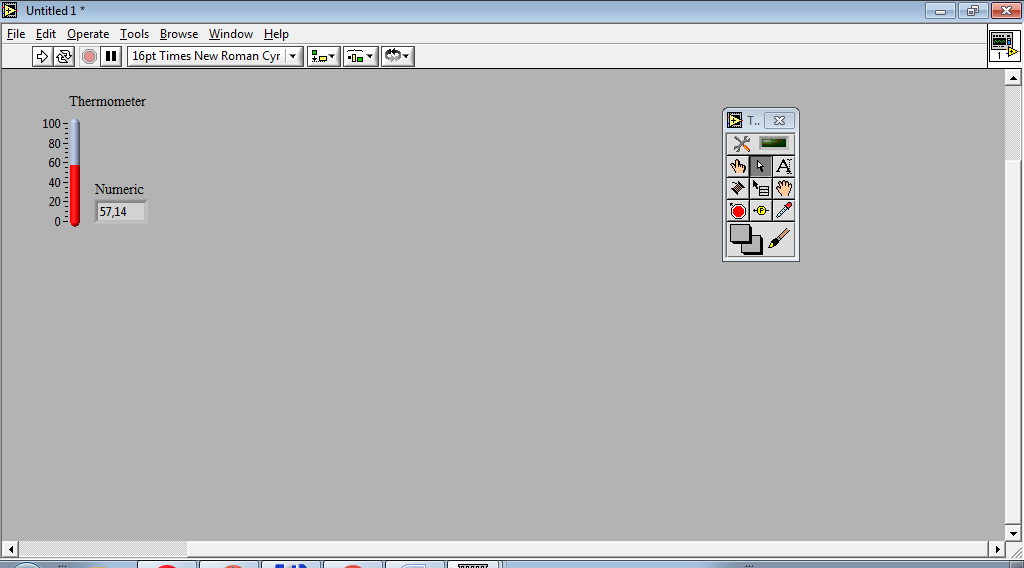
****

Рисунок 7 – Результат запуска виртуального прибора, когда предварительно на шкале термометра выставлено некоторое значение

Покажем, что изменение диапазона по шкале прибора не является сложной задачей. Она решается с использованием инструмента редактирования текста (*Edit Text*). Выберем при активном инструменте нижнее значение по диапазону и заменим его, например, с нуля на «-50» (Рисунок 8).

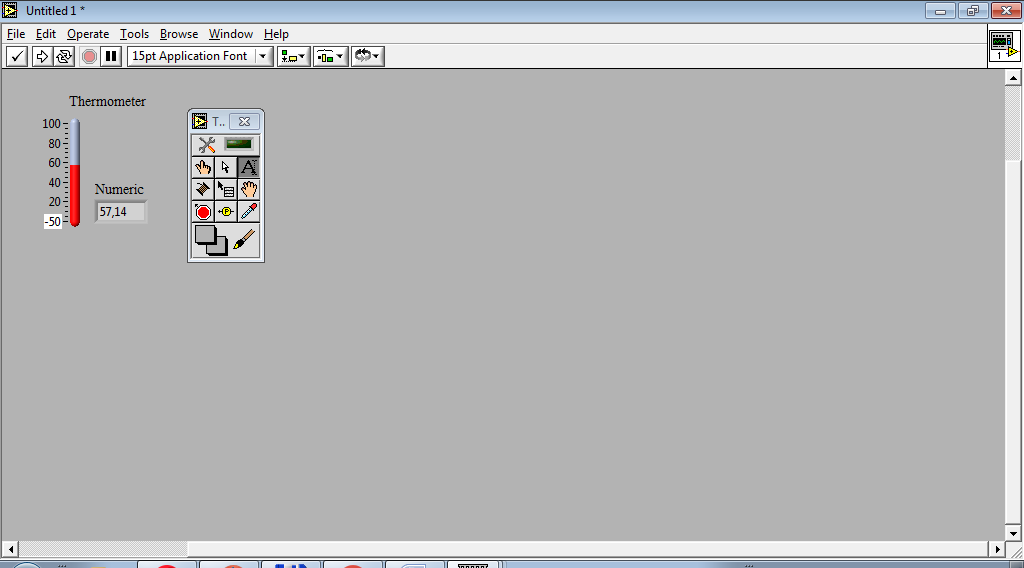
****

Рисунок 8 – Демонстрация изменения значения нижней границы диапазона термометра с использованием инструмента редактирования текста (*Edit Text*)

После подтверждения выполненного изменения посредством нажатия на клавишу «*Enter*» на клавиатуре изменённое значение вступит в силу (Рисунок 9).

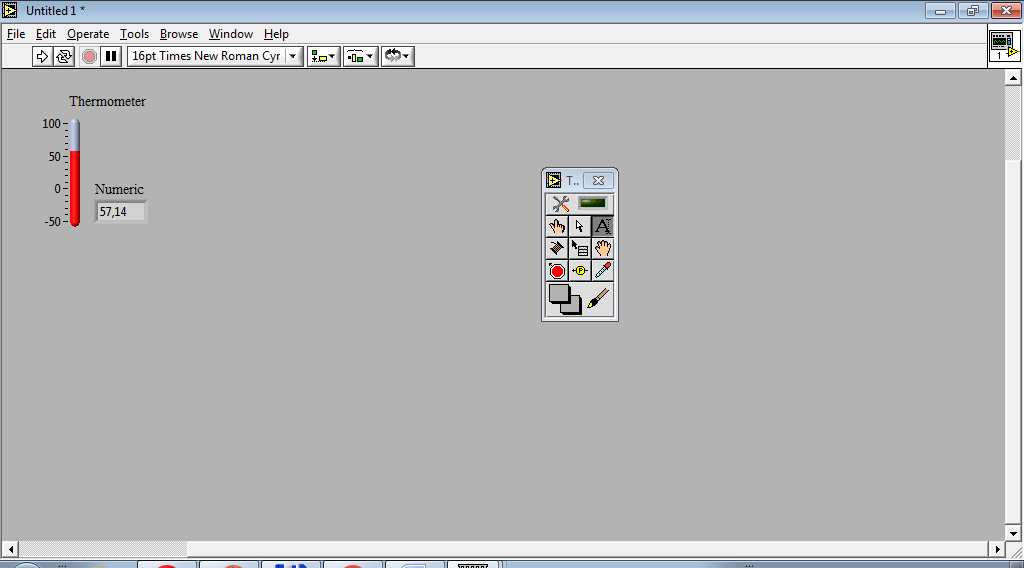
****

Рисунок 9 – Результат изменения нижней границы диапазона для термометра

Часто у начинающих разработчиков виртуальных приборов возникает проблема с тем, чтобы задать шкалу с дробными значениями в *National Instruments LabView*. Разберём на примере настройки шкалы термометра в диапазоне от нуля до одного градуса (Рисунок 10).

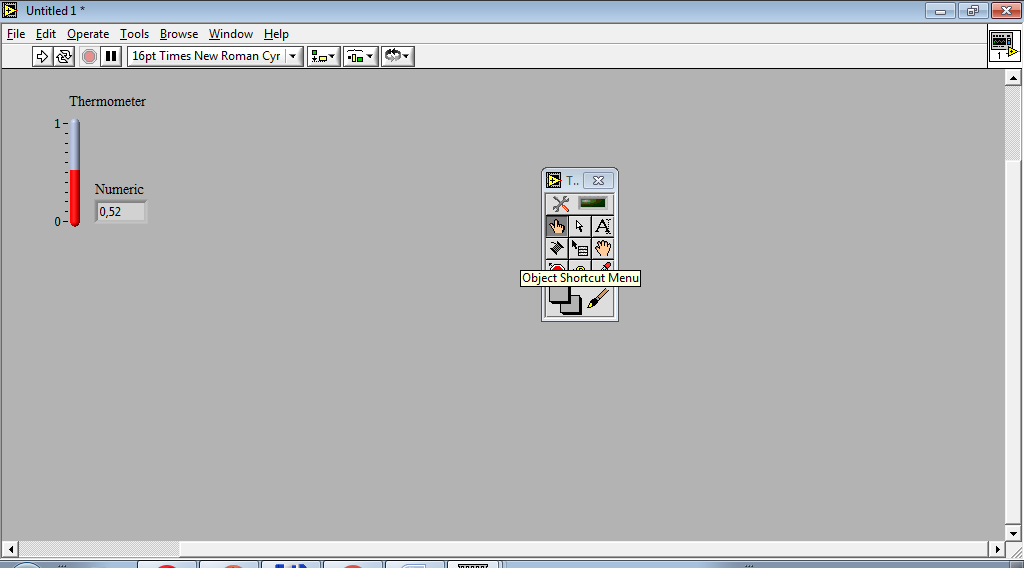
****

Рисунок 10 – Результат изменения одновременно верхней и нижней границ диапазона термометра для последующей демонстрации настройки дробных значений шкалы

Эта проблема имеет довольно несложное решение. Для начала необходимо обратиться к контекстному меню термометра и в разделе шкала (*Scale*) перейти к диалоговому окну, скрывающемуся за пунктом «Формат и Точность» (*Format & Precision*). Переход по контекстному меню показан на Рисунке 11.

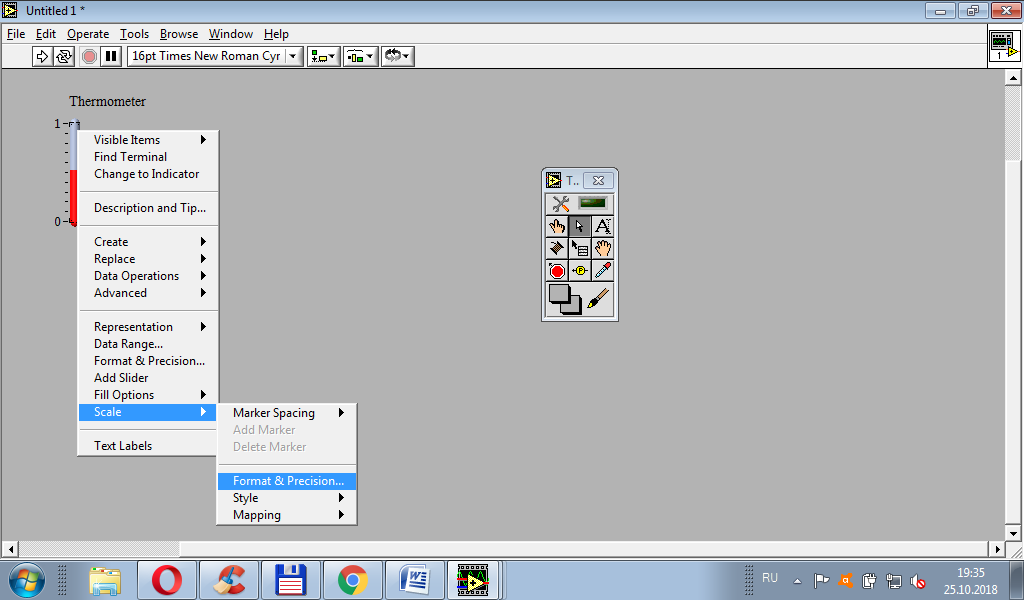
****

Рисунок 11 – В контекстном меню термометра показан раздел настройки шкалы (*Scale*) с маркировкой перехода к диалоговому окну «Формат и Точность» (*Format & Precision*)

В открывшемся диалоговом окне необходимо выставить требуемую точность по количеству знаков после плавающей запятой (*Digits of Precision*). В данном примере рассматривается точность с двумя знаками после плавающей запятой (Рисунок 12).

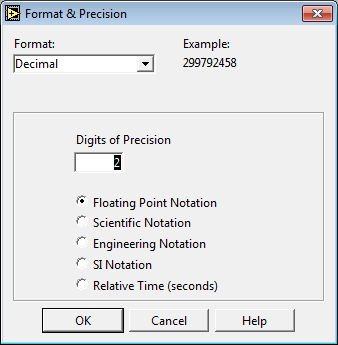
****

Рисунок 12 – Диалоговое окно «Формат и Точность» с изменённым значением точности до двух знаков после запятой

После подтверждение настроек нажатием на кнопку «*OK*» шкала виртуального термометра изменится – на ней появятся промежуточные значения, а также возможность редактирования цифр, расположенных справа от плавающей запятой (Рисунок 13). Предыдущие настройки не позволили бы реализовать подобные ввод и корректировку – после подтверждения результат бы округлялся до ближайшего целого.

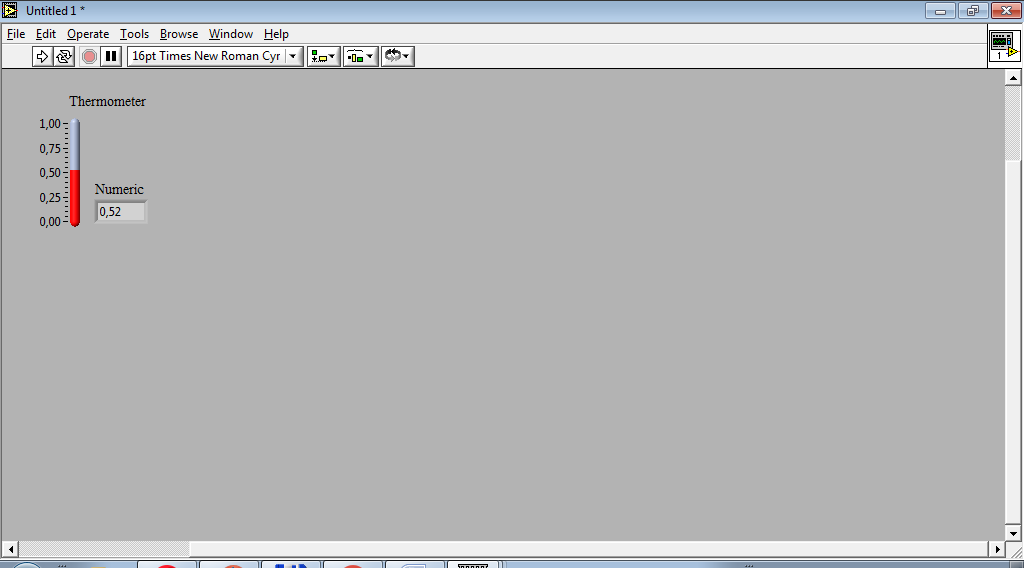
****

Рисунок 13 – Демонстрация шкалы, учитывающей два знака после плавающей запятой

Соответственно, как только мы отказались от округления до ближайшего целого, появилась возможность для ещё большего сокращения диапазона по шкале виртуального термометра. Такой элемент (Рисунок 14) позволяет в полном объёме исследовать обратно пропорциональные температурные зависимости.

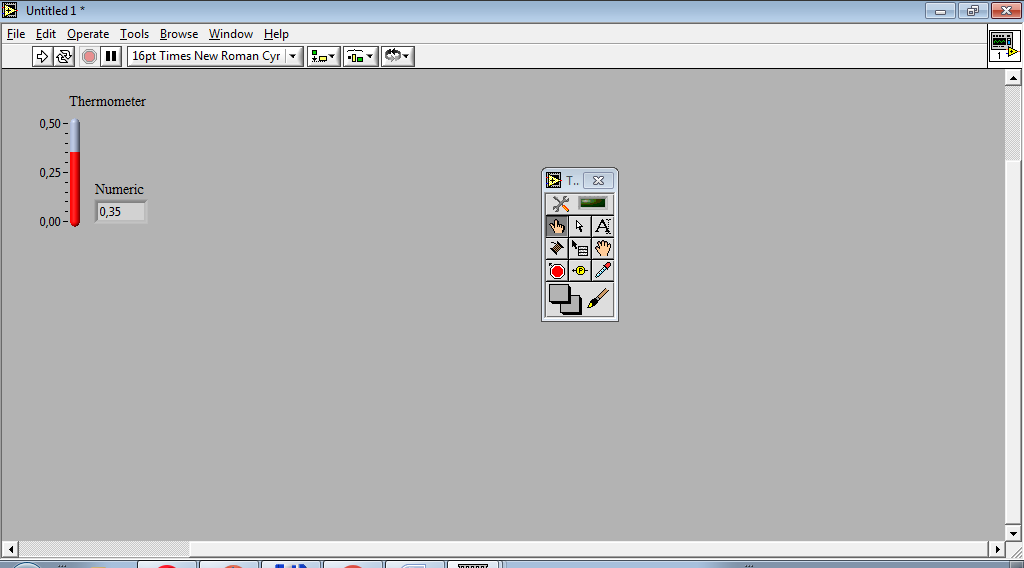
****

Рисунок 14 – В таком состоянии шкала подготовлена для демонстрации обратной пропорциональной зависимости между единицами измерения температуры

Для упомянутого исследования на графическом пользовательском интерфейсе понадобится второй термометр (Рисунок 15).

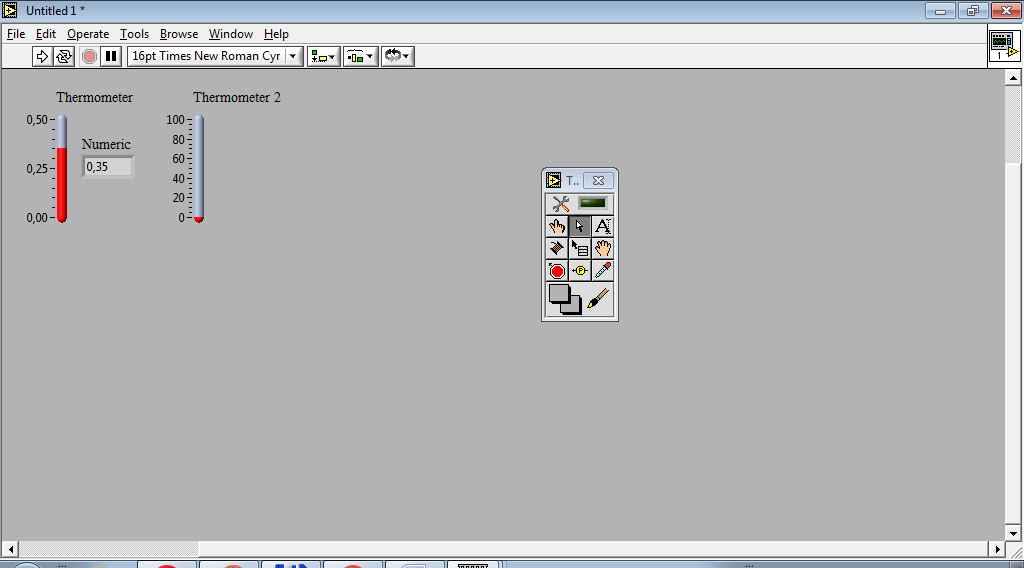
****

Рисунок 15 – Дополнение передней панели виртуального прибора вторым термометром

На Рисунке 16 показан код, позволяющий организовать обратно пропорциональную произвольно выдуманную температурную зависимость с коэффициентом усиления, равным «10» для организации большего масштабирования.

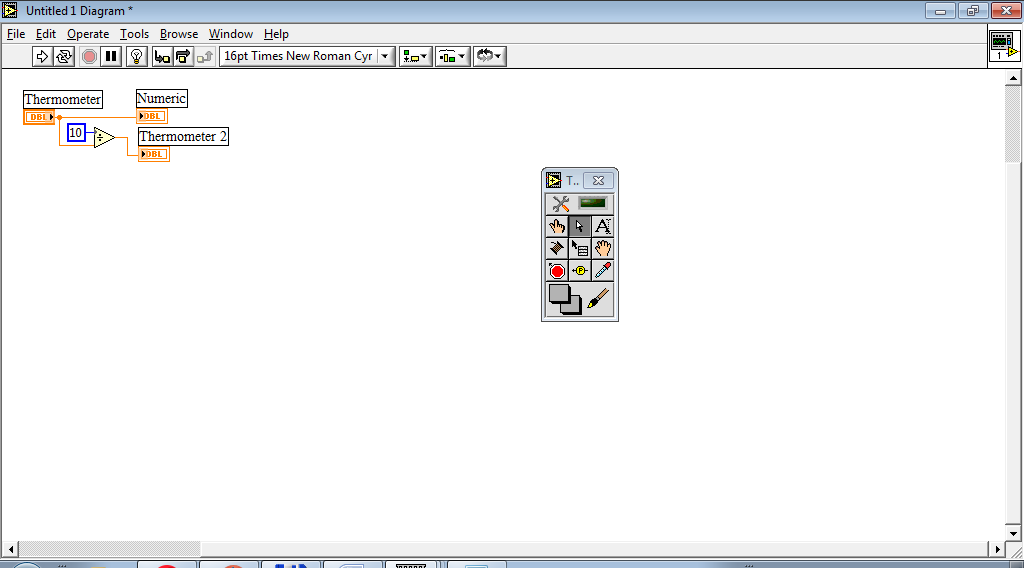
****

Рисунок 16 – Обратная пропорциональная зависимость единиц измерения температуры с коэффициентом масштабирования равным «10» единицам на блок-диаграмме

Так большему входному значению соответствует меньшее выходное (Рисунок 17).

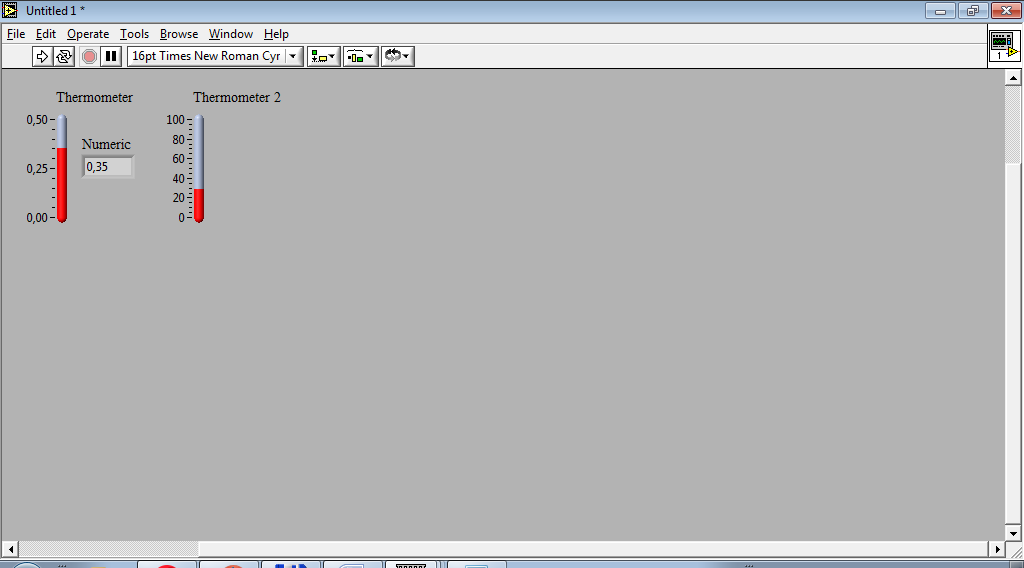
****

Рисунок 17 – Результат работы программы конвертации единиц измерения температуры

Прежде, чем двигаться дальше, разделим идентичные термометры цветовым признаком (это важно знать, поскольку данная настройка является обязательным атрибутом индивидуальной части задания). Для удобства повествования, исключительно. Подход к изменению цвета проиллюстрирован Рисунком 18.

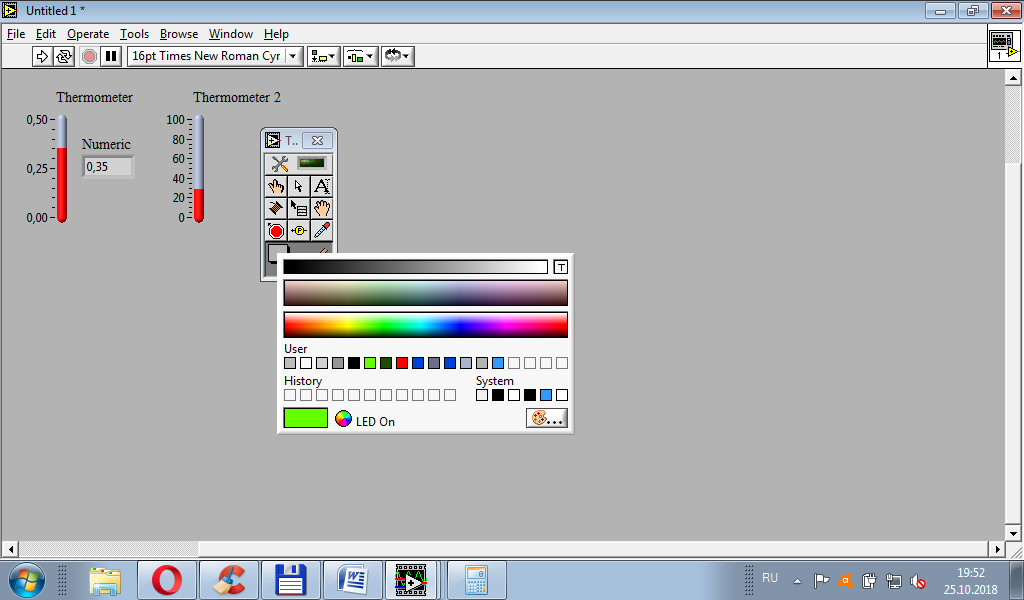
****

Рисунок 18 – Изменение цветовой настройки для окрашивания самих элементов в зелёный цвет; цветовая настройка для окрашивания подложки элементов оставлена тёмно-серой

После выбора активной части термометра инструментом установки цвета (*Set Color*) он изменит своё состояние по аналогии с тем, как это проиллюстрировано на Рисунке 19.

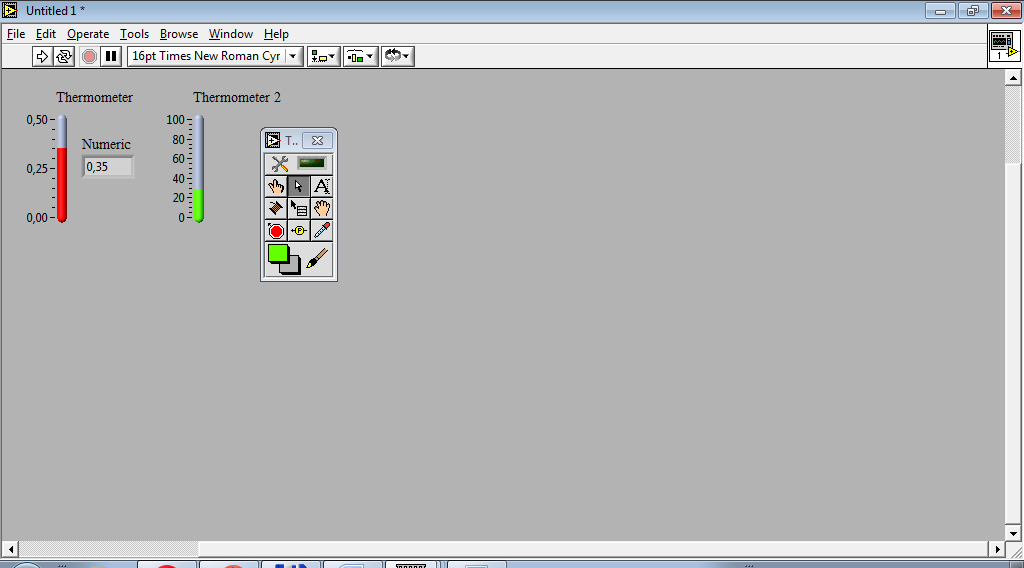
****

Рисунок 19 – Правый термометр окрашен в зелёный цвет для удобства визуального различия имеющихся термометров

Так вот при обратно пропорциональной зависимости меньшему значению на входе соответствует большее значение на выходе (Рисунок 20).

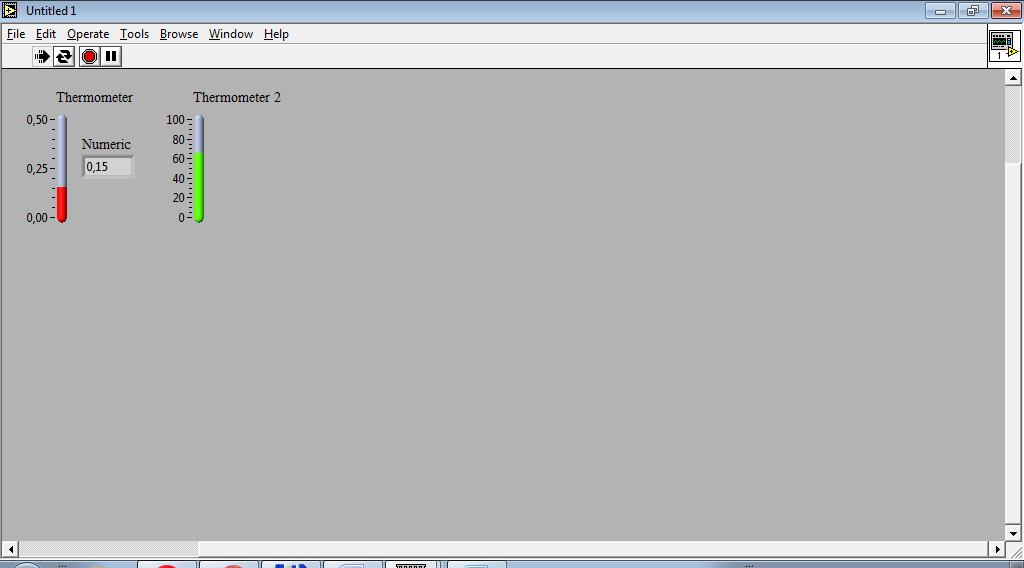
****

Рисунок 20 – Проверка реакции: при снижении активного параметра выходной сигнал возрастает

По аналогии с управляющим термометром индикаторный термометр снабдим дополнительным числовым индикатором, роль которого состоит в том, чтобы отображать более точные значения температуры, которые могут быть не очевидны по шкале с выбранным по умолчанию масштабом (Рисунок 21).

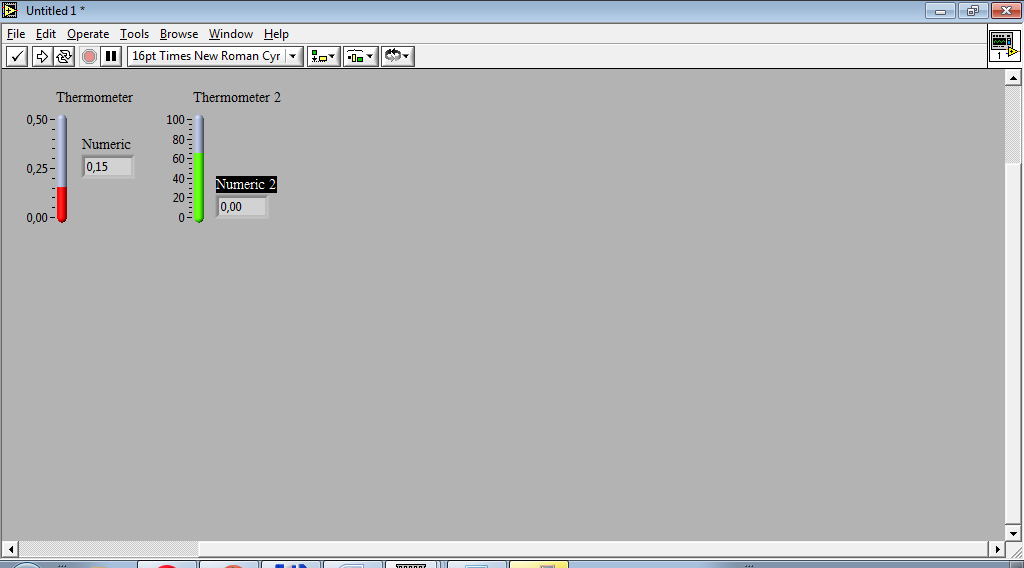
****

Рисунок 21 – Добавим численный индикатор, позволяющий точно отслеживать значения шкалы зелёного термометра

На блок-диаграмме показано, что упомянутое дополнение не сильно отягощает разработанную ранее схему виртуального прибора (Рисунок 22).

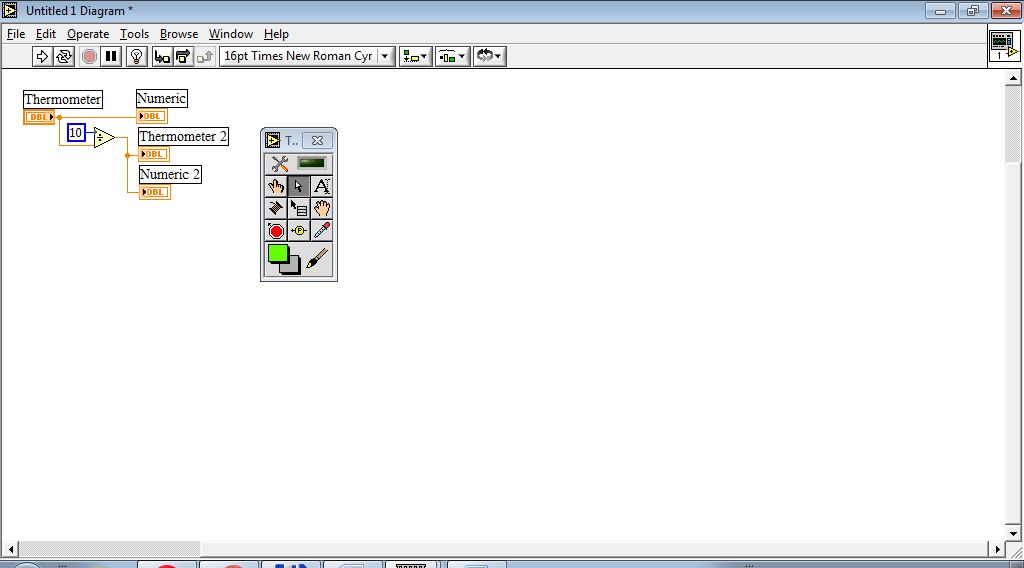
****

Рисунок 22 – Подключение численного индикатора к зелёному термометру на блок-диаграмме

В таком состоянии виртуальный прибор становится идеальным объектом для изучения такой процедуры, как «градуировка», изучаемой в метрологии. Градуировка шкал виртуальных термометров, проводящих одни и те же измерения по различным правилам (системам измерений) рассматривается далее, в последующем разделе.

## 3.4 Варианты индивидуального задания

Величины предельных значений каждого термометра могут и должны отличаться друг от друга. Например, если для градусов Цельсия задан диапазон от -100 до 100, то для градусов Кельвина должен быть задан диапазон от 173,15 до 373,15.

Таблица 3.4.1 – Диапазоны шкал виртуальных термометров

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** |
| -30…30 | 2.9…6.1 | -1…99 | 0.12…3.45 | -1000…1000 |
| **6** | **7** | **8** | **9** | **10** |
| 5…55 | 0.1…0.9 | -70.77…15.58 | 0…10 | -50…0 |
| **11** | **12** | **13** | **14** | **15** |
| 0…80 | 12.77…40.92 | -500…300 | -15…15 | -20…-10 |
| **16** | **17** | **18** | **19** | **20** |
| -78.9…-12.3 | -5…55 | -45…10 | -0.9…0.1 | 0…100 |
| **21** | **22** | **23** | **24** | **25** |
| 300…500 | -22…22 | -10000…10 | 0.001…0.1 | -0.001…0 |
| **26** | **27** | **28** | **29** | **30** |
| 1…101 | 0.9…30.9 | 0.0001…1 | 0…4 | -23.4…5.67 |

Выполнить второй виртуальный прибор для исследования конвертации температур, в котором за основу взять единицы измерения, заданные по варианту.

|  |  |
| --- | --- |
| **№ вар.** | **Содержание задания** |
| 1. | Выполнить прямой перевод из градусов Кельвина в градусы Делиля.  Цвет входного термометра оранжевый выходного – фиолетовый. |
| 2. | Выполнить прямой перевод из градусов Фаренгейта в градусы Реомюра.  Цвет входного термометра зелёный выходного – красный. |
| 3. | Выполнить прямой перевод из градусов Делиля в градусы Кельвина.  Оба термометра оливкового цвета. |
| 4. | Выполнить прямой перевод из градусов Ньютона в градусы Реомюра.  Оба термометра фиолетового цвета. |
| 5. | Выполнить прямой перевод из градусов Фаренгейта в градусы Рёмера.  Цвет входного термометра серый выходного – розовый. |
| 6. | Выполнить прямой перевод из градусов Реомюра в градусы Рёмера.  Оба термометра зелёного цвета. |
| 7. | Выполнить прямой перевод из градусов Кельвина в градусы Ранкина.  Цвет входного термометра синий выходного – зелёный. |
| 8. | Выполнить прямой перевод из градусов Делиля в градусы Цельсия.  Оба термометра серые. |
| 9. | Выполнить прямой перевод из градусов Фаренгейта в градусы Ранкина.  Цвет входного термометра фиолетовый выходного – синий. |
| 10. | Выполнить прямой перевод из градусов Ньютона в градусы Кельвина.  Цвет входного термометра салатовый выходного – алый. |
| 11. | Выполнить прямой перевод из градусов Делиля в градусы Реомюра.  Цвет входного термометра голубой выходного – серый. |
| 12. | Выполнить прямой перевод из градусов Кельвина в градусы Рёмера.  Цвет входного термометра чёрный выходного – салатовый. |
| 13. | Выполнить прямой перевод из градусов Ньютона в градусы Делиля.  Цвет входного термометра бирюзовый выходного – лайм. |
| 14. | Выполнить прямой перевод из градусов Фаренгейта в градусы Цельсия.  Цвет входного термометра синий выходного – красный. |
| 15. | Выполнить прямой перевод из градусов Делиля в градусы Рёмера.  Цвет входного термометра красный выходного – коричневый. |
| 16. | Выполнить прямой перевод из градусов Кельвина в градусы Реомюра.  Цвет входного термометра розовый выходного – салатовый. |
| 17. | Выполнить прямой перевод из градусов Делиля в градусы Ранкина.  Цвет входного термометра чёрный выходного – красный. |
| 18. | Выполнить прямой перевод из градусов Фаренгейта в градусы Кельвина.  Цвет входного термометра жёлтый выходного – белый. |
| 19. | Выполнить прямой перевод из градусов Ньютона в градусы Цельсия.  Цвет морской волны для входного термометра выходного – красный. |
| 20. | Выполнить прямой перевод из градусов Делиля в градусы Фаренгейта.  Цвет входного термометра голубой выходного – оранжевый. |
| 21. | Выполнить прямой перевод из градусов Кельвина в градусы Цельсия.  Цвет входного термометра белый выходного – чёрный. |
| 22. | Выполнить прямой перевод из градусов Рёмера в градусы Реомюра.  Цвет входного термометра синиё выходного – жёлтый. |
| 23. | Выполнить прямой перевод из градусов Фаренгейта в градусы Делиля.  Оба термометра салатового цвета. |
| 24. | Выполнить прямой перевод из градусов Реомюра в градусы Ньютона.  Оба термометра оранжевого цвета. |
| 25. | Выполнить прямой перевод из градусов Кельвина в градусы Ньютона.  Цвет входного термометра оливковый выходного – оранжевый. |
| 26. | Выполнить прямой перевод из градусов Фаренгейта в градусы Ньютона.  Цвет входного термометра синий выходного – жёлтый. |
| 27. | Выполнить прямой перевод из градусов Рёмера в градусы Цельсия.  Цвет входного термометра серый выходного – розовый. |
| 28. | Выполнить прямой перевод из градусов Делиля в градусы Ньютона.  Оба термометра белого цвета. |
| 29. | Выполнить прямой перевод из градусов Кельвина в градусы Фаренгейта.  Цвет входного термометра красный выходного – синий. |
| 30. | Выполнить прямой перевод из градусов Ньютона в градусы Рёмера.  Цвет входного термометра алый выходного – бледно-зелёный. |

## 3.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

## 3.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 3.7 Блок-схемы алгоритмов для типовых решений

**4 Задание «Разработка виртуального прибора "Умный калькулятор (*CASE*-калькулятор)"»**

В пакете прикладных программ *National Instruments* *LabView* разработать виртуальный прибор «Умный калькулятор», в состав которого входят операции, заданные согласно таблице вариантов (Таблица 3).

Операции должны быть размещены на графическом пользовательском интерфейсе строго в том порядке, в котором они упоминаются в таблице исходных вариантов. Определение именно той операции, которую пожелал реализовать пользователь, должно выполняться в соответствии с некоторой логикой, задаваемой булевскими элементами управления, например, кнопками или тумблерами.

В процессе разработки графического пользовательского интерфейса предусмотреть:

* числовые контроллеры в необходимом количестве для ввода значений операндов,
* один единственный числовой индикатор для вывода результата/ответа,
* логические контроллеры в необходимом количестве для однозначного определения системой выполняемой операции.

Результат/ответ выводить на индикатор только при единственном включённом логическом контроллере.

Исключить одновременное включение нескольких логических контроллеров и либо не реагировать на подобный входной сигнал – оставлять на выходе ноль, либо при включении нескольких контроллеров выдавать ошибки, формируемые в виде числового кода, например, «9999», «8888», которые следует расшифровать для пользователя на графическом пользовательском интерфейсе дополнительной таблицей соответствия.

Далее приведены примеры расшифровок кодов, которые могут быть размещены на графическом пользовательском интерфейсе:

* «9999» – недопустимо одновременное умножение и деление,
* «8888» – недопустимо одновременное сложение и вычитание,
* «7777» – недопустимо деление на ноль,
* и так далее.

Автору требуется продумать структуру других возможных ошибок самостоятельно.

На блок-диаграмме все функциональные, то есть значащие пересечения линий связи (узлы) обозначить точками. Для этого в меню перейти к настройкам «*Tools > Options…*», далее в выпадающем списке перейти к настройкам блок-диаграммы (*Block Diagram*) и выставить галочку напротив пункта «*Show dots at wire junctions*».

По итогам выполнения работы сдаются строго два файла:

- отчёт, выполненный в текстовом редакторе *Microsoft Office Word* (*\*.doc* или *\*.docx*);

- файл виртуального прибора *National Instruments LabView* (*\*.vi*) по индивидуальной части работы.

Отправленные поодиночке файлы проверке не подлежат. При отсутствии одного из упомянутых файлов зачёт по заданию не выставляется.

Необходимо продумать индивидуальность подхода к решению логического наполнения «Умного калькулятора» на базе *CASE-*структуры. Двух одинаковых работ в студенческой группе быть не должно.

\*Для выполнения всех работ рекомендуется скачать старую версию (6.1) пакета прикладных программ *National Instruments LabView* (). Обучающимся,

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «11».

## 4.1 Цель работы

закрепление навыков работы и навыков графического программирования в пакете прикладных программ *National Instruments* *LabView*. Приобретение навыков работы c *CASE*-структурой, входящей в состав пакета прикладных программ *National Instruments* *LabView*.

## 4.2 Полезные соотношения для выполнения задания

**Для выполнения индивидуального задания потребуются следующие функции:**

Раздел числовых элементов содержит следующие графические функции, подразделы и константы, изображённые на Рисунке 1 (чтение ведётся слева направо, сверху вниз):

- операция сложения (*Add*);

- операция вычитания (*Subtract*);

- операция умножения (*Multiply*);

- операция деления (*Divide*);

- операции целочисленного деления и получения остатка от деления (*Quotient & Remainder*);

- подраздел конвертации (*Conversion*);

- операция инкрементирования / добавления единицы (*Increment*);

- операция декрементирования / вычитания единицы (*Decrement*);

- операция суммирования элементов массива (*Add Array Elements*);

- операция перемножения элементов массива (*Multiply Array Elements*);

- элемент, объединяющий арифметические операции (*Compound Arithmetic*);

- подраздел тригонометрии (*Trigonometric*);

- операция взятия абсолютной величины / модуля (*Absolute Value*);

- операция математического округления до ближайшего целого (*Round To Nearest*);

- операция округления в меньшую сторону (*Round To -Infinity*);

- операция округления в большую сторону (*Round To +Infinity*);

- операция выдачи реализации псевдослучайной величины (*Random Number (0-1)*);

- подраздел логарифмов (*Logarithmic*);

- операция взятия квадратного корня (*Square Root*);

- операция инверсии / взятия значения с обратным знаком (*Negate*);

- операция масштабирования по степени двойки / умножение на 2 в степени *N* (*Scale By Power Of 2*);

- операция сигнум (определения знака): пришло любое положительное – в ответе «1», пришло любое отрицательное – в ответе «-1», пришёл ноль – в ответе «0» (*Sign*);

- операция обращения / деления единицы на заданное значение *(Reciprocal*);

- подраздел операций над комплексными числами (*Complex*);

- численная константа (*Numeric Constant*);

- перечислимая константа (*Enum Constant*);

- константа списка значений (*Ring Constant*);

- узел математических выражений (*Expression Node*);

- подраздел дополнительных констант (*Additional Numeric Constants*).

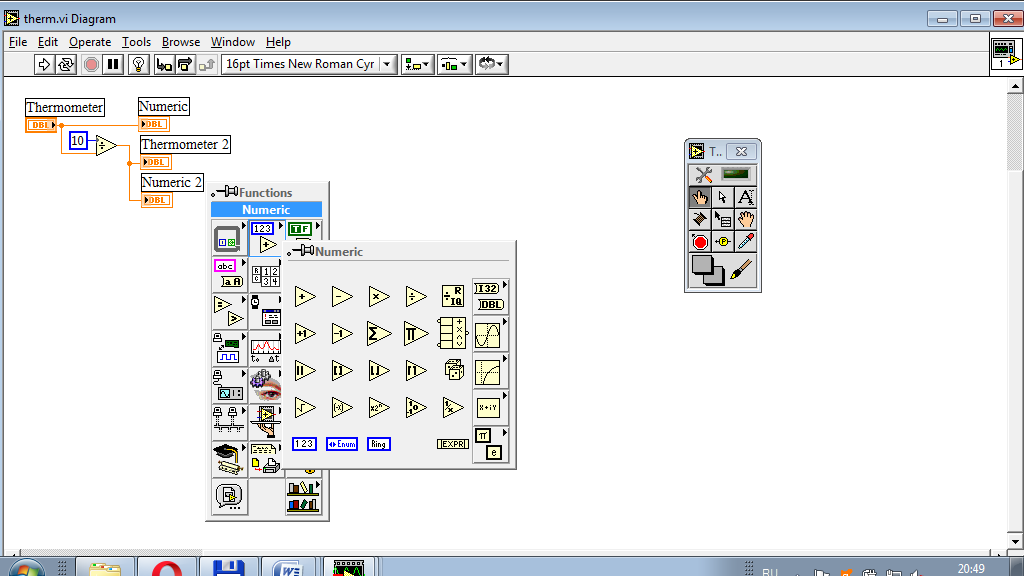


Рисунок 1 – Раздел численных (*Numeric*) функций на блок-диаграмме

Подчёркнутые выше элементы в данной работе использованы не будут. Далее рассмотрен подраздел тригонометрических функций (Рисунок 2):

- синус (*Sine*);

- косинус (*Cosine*);

- тангенс (*Tangent*);

- арксинус (*Inverse Sine*);

- арккосинус (*Inverse Cosine*);

- арктангенс (*Inverse Tangent*);

- гиперболический синус (*Hyperbolic Sine*);

- гиперболический косинус (*Hyperbolic Cosine*);

- гиперболический тангенс (*Hyperbolic Tangent*);

- гиперболический арксинус (*Inverse Hyperbolic Sine*);

- гиперболический арккосинус (*Inverse Hyperbolic Cosine*);

- гиперболический арктангенс (*Inverse Hyperbolic Tangent*);

- косеканс (*Cosecant*);

- секанс (*Secant*);

- котангенс (*Cotangent*);

- объединённых синуса и косинуса (*Sine & Cosine*);

- арктангенс двух аргументов (*Inverse Tangent (2 Input)*);

- отсчётов (*Sinc*).

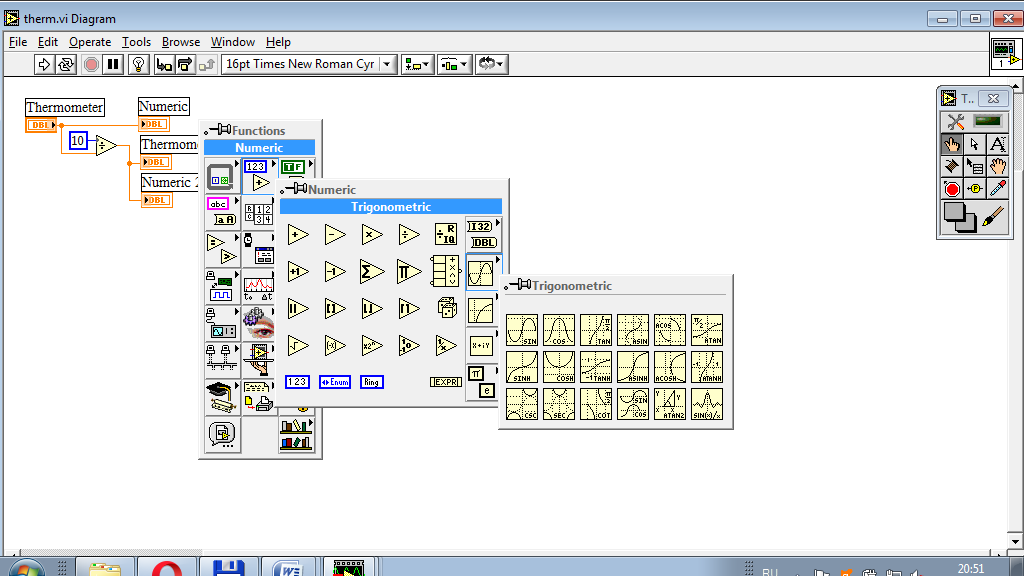


Рисунок 2 – Подраздел тригонометрических (*Trigonometric*) функций раздела численных (*Numeric*) функций на блок-диаграмме

Подчёркнутые выше элементы в данной работе использованы не будут. Далее рассмотрен подраздел логарифмических функций (Рисунок 3):

- экспонента (*Exponential*);

- степень десяти (*Power Of 10*);

- степень двойки (*Power Of 2*);

- возведение *X* в степень *Y* (*Power Of X*);

- экспонента, проходящая через начало координат (*Exponential (Arg) – 1*);

- натуральный логарифм (*Natural Logarithm*);

- логарифм по основанию десяти (*Logarithm Base 10*);

- логарифм по основанию двойки (*Logarithm Base 2*);

- логарифм по основанию *X* (*Logarithm Base X*);

- натуральный логарифм, проходящий через начало координат (*Natural Logarithm (Arg + 1)*).

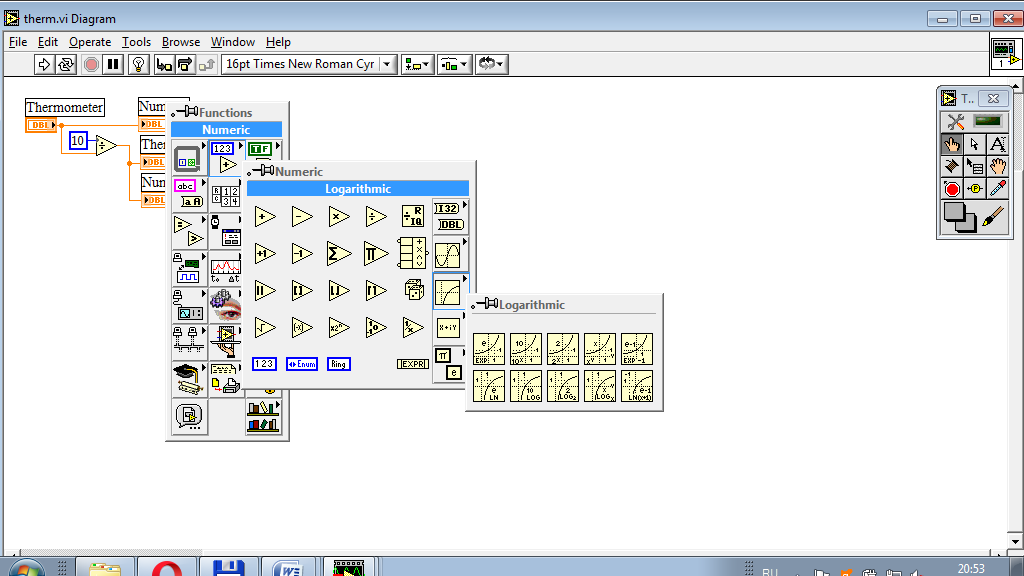


Рисунок 3 – Подраздел логарифмических (*Logarithmic*) функций раздела численных (*Numeric*) функций на блок-диаграмме

Подчёркнутые выше элементы в данной работе использованы не будут. Далее рассмотрен подраздел дополнительных численных констант (Рисунок 4):

- цветовая константа (*Color Box Constant*);

- символьная константа списка значений для интерфейсного элемента «список» (*Listbox Symbol Ring Constant*);

- константа списка ошибок (*Error Ring Constant*);

- Пифагорова константа (*Pi*);

- Пифагорова константа, умноженная на два (*Pi Multiplied By 2*);

- Пифагорова константа, делённая на два (*Pi Divided By 2*);

- обращённая Пифагорова константа (*Reciprocal Of Pi*);

- натуральный логарифм Пифагоровой константы (*Natural Logarithm Of Pi*);

- минус бесконечность (*Negative Infinity*);

- основание натурального логарифма (*Natural Logarithm Base*);

- обращённое основание натурального логарифма (*Reciprocal Of e*);

- логарифм экспоненты по основанию десяти (*Base 10 Logarithm Of e*);

- натуральный логарифм десяти (*Natural Logarithm Of 10*);

- натуральный логарифм двойки (*Natural Logarithm Of 2*);

- плюс бесконечность (*Positive Infinity*);

- постоянная Планка (*Planck Constant (J/Hz)*);

- элементарный заряд (*Elementary Charge (C)*);

- скорость света в вакууме (*Speed Of Light In Vacuum (m/sec)*);

- гравитационная постоянная (*Gravitational Constant (N m2/kg2)*);

- постоянная Авогадро (*Avogadro Constant (1/mol)*);

- Постоянная Ридберга (*Rydberg Constant (/m)*);

- молярная газовая постоянная (*Molar Gas Constant (J / (mol K))*).

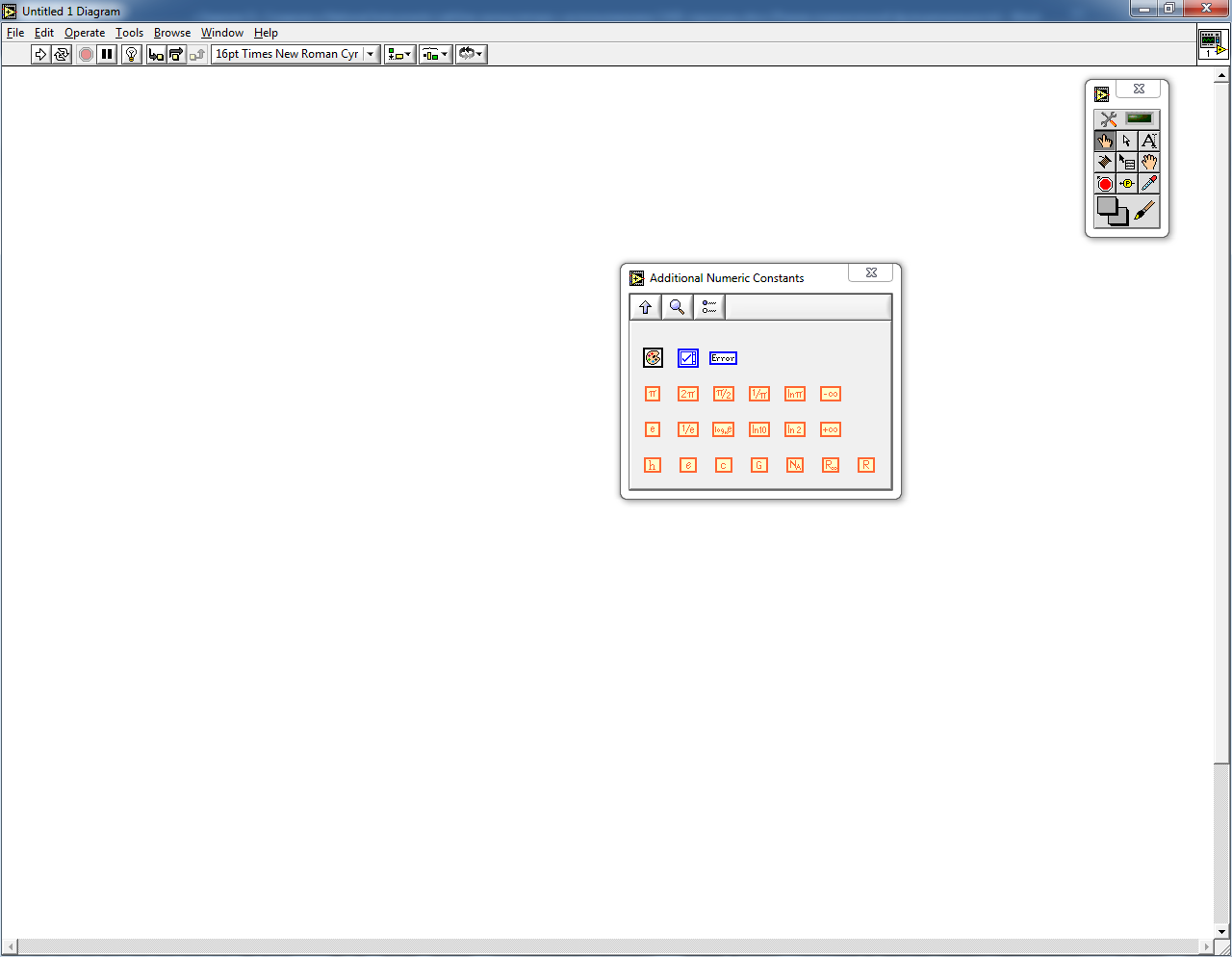


Рисунок 4 – Подраздел дополнительных численных констант (*Additional Numeric Constants*) раздела численных (*Numeric*) функций на блок-диаграмме

Подчёркнутые выше элементы в данной работе использованы не будут.

**Использование объединённой арифметики:**

Объединённая арифметика – функция, относящаяся одновременно к двум разделам: числовому и логическому. Функция объединяет в себе следующие операции: сложения, умножения, логического сложения (ИЛИ), логического умножения (И), исключающего ИЛИ. На Рисунке 5 показано расположение объединённой арифметики в разделе численных функций блок-диаграммы.

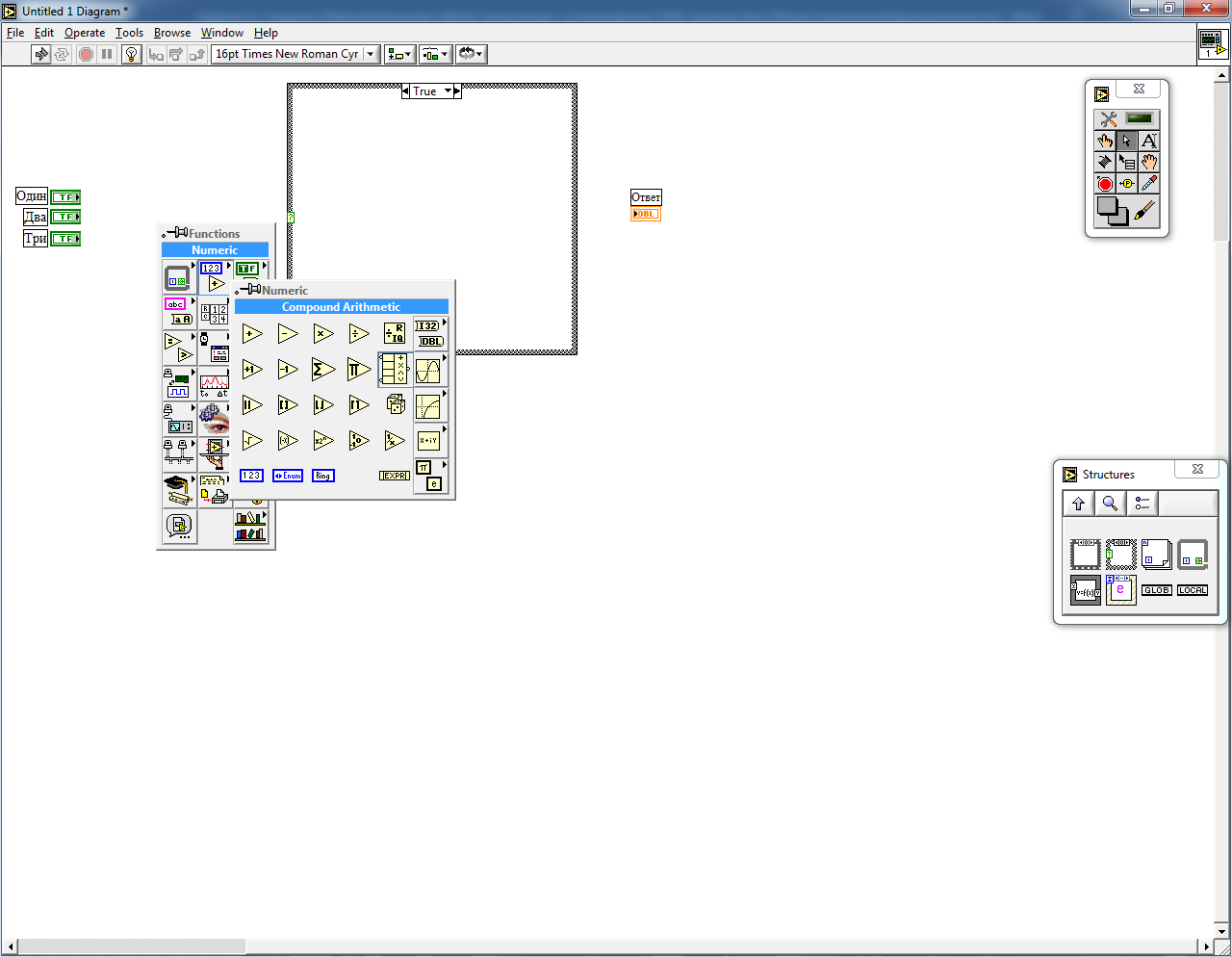


Рисунок 5 – Маркировка расположения функции объединённой арифметики (*Compound Arithmetic*) раздела численных (*Numeric*) функций, размещаемых на блок-диаграмме

При размещении объединённой арифметики на блок-диаграмме по умолчанию доступно только два входа, а также выбрана операция сложения. Для изменения количества входов необходимо выбрать среди инструментов элемент позиционирования / изменения размерности / выбора (*Position / Size / Select*) и, ухватившись за нижнюю границу размещённой на блок-диаграмме функции, вытянуть её вниз на нужное количество входов. Для изменения выполняемой операции функцией объединённой арифметики необходимо вызвать контекстное меню этой функции нажатием на неё правой кнопкой мыши и далее в пункте изменения режима (*Change Mode*) выбрать нужную операцию. В рамках решаемой задачи объединённая арифметика используется в режиме логического умножения (*AND*). Смена режима объединённой арифметики показана на Рисунке 6.

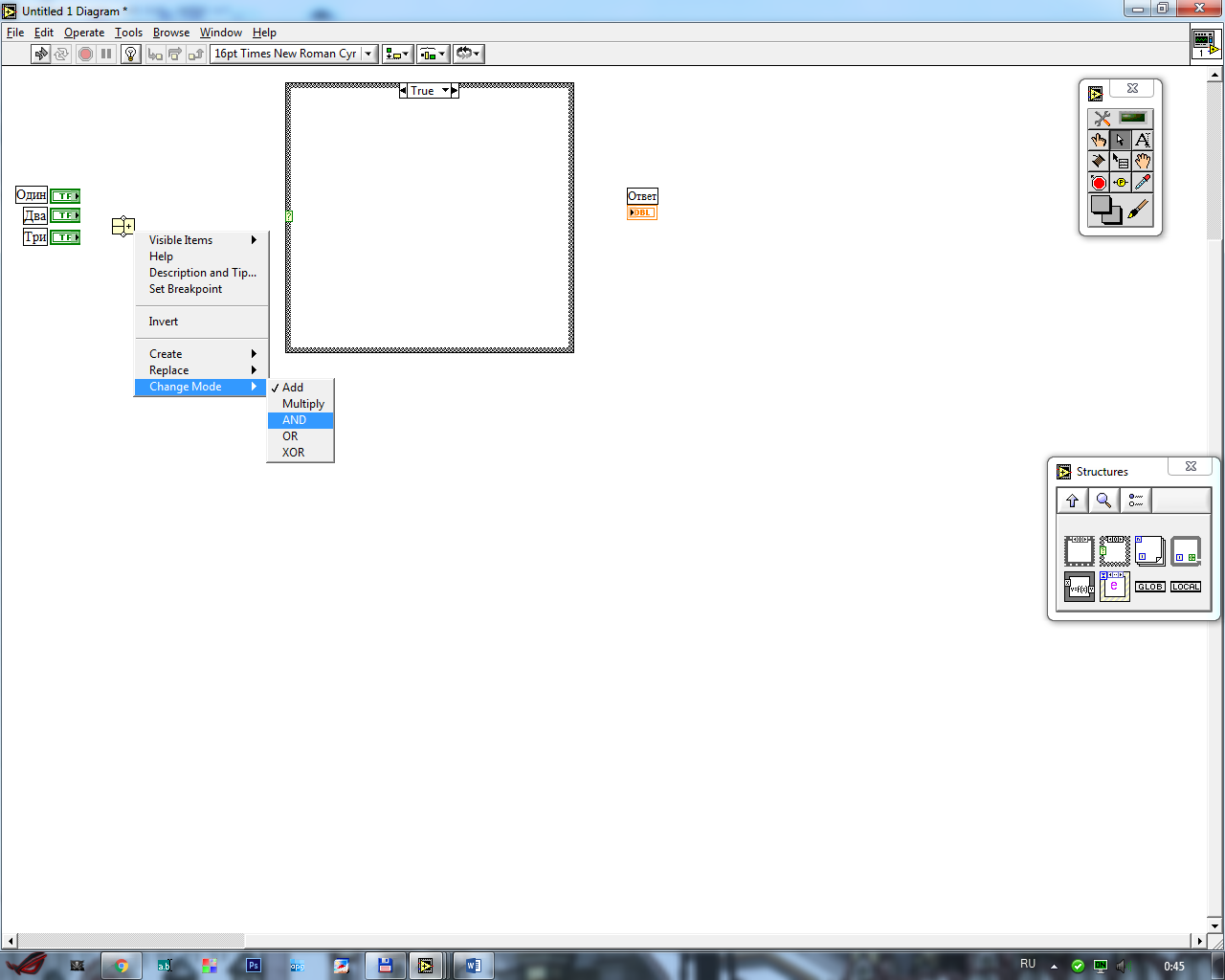


Рисунок 6 – Замена операции в рамках функции объединённой арифметики с численного сложения «+» (*Add*) на логическое умножение «И» («AND»)

С целью сокращения количества применяемых логических элементов на блок-диаграмме можно воспользоваться одной из полезных настроек функции объединённой арифметики. В контекстном меню функции объединённой арифметики присутствует возможность инвертирования (*Invert*) полученного значения. Так, например, в решаемой задаче может быть эффективно использована операция отрицания логического умножения («И-НЕ»). Такая операция доступна посредством размещения на блок-диаграмме одного единственного, но особым образом настроенного блока функции объединённой арифметики. Настройка инверсии логического умножения представлена на Рисунке 7.

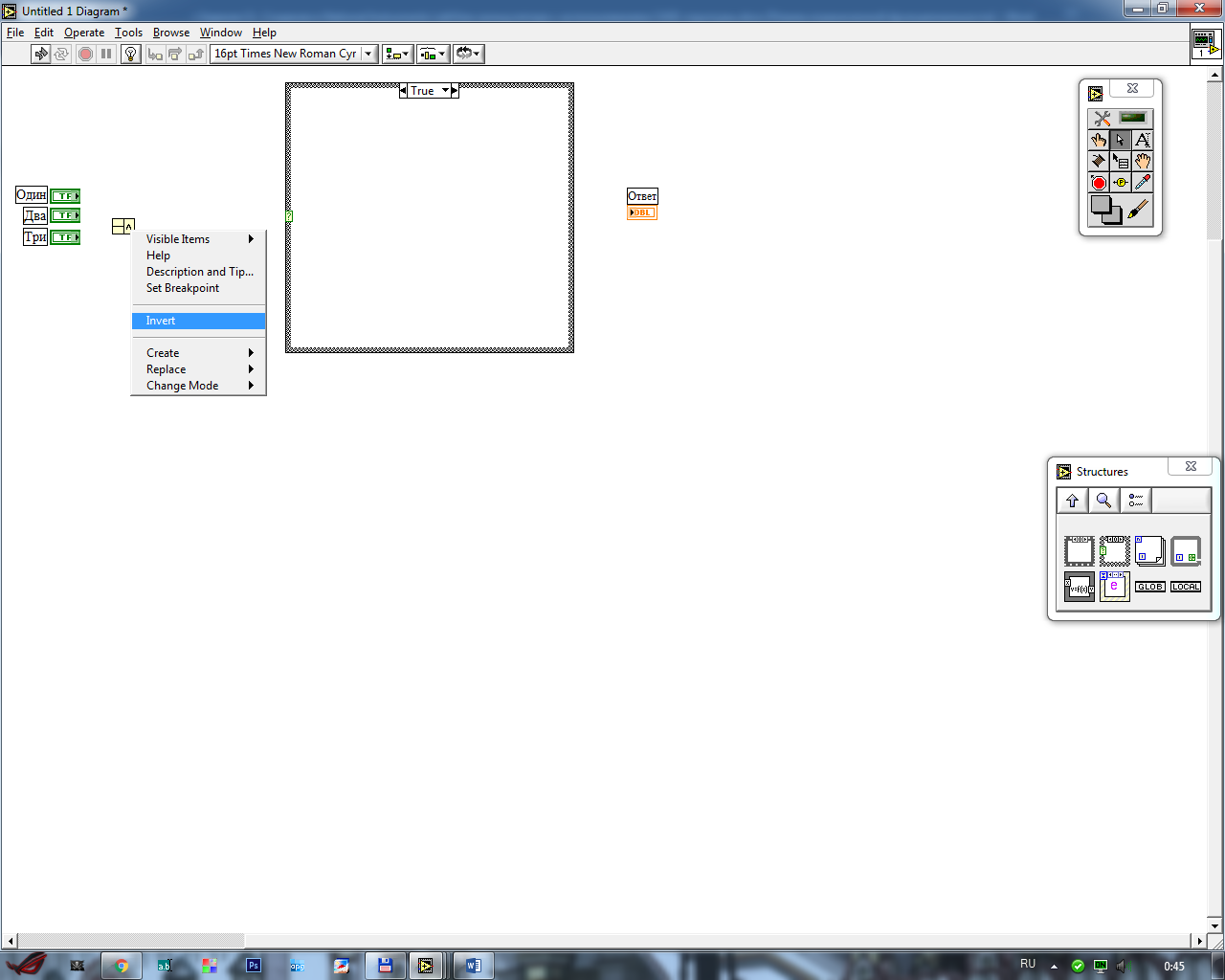


Рисунок 7 – Изменение режима применяемой операции логического умножения с прямой «И» на инверсную «И-НЕ» (*Invert AND*)

По итогам изучения основ настройки функции объединённой арифметики, а также численных и логических элементов, входящих в состав пакета прикладных программ *National Instruments LabView*, можно переходить непосредственно к созданию заготовки для виртуального прибора «Умный калькулятор».

## 4.3 Пример выполнения задания

Как и было заявлено ранее, для демонстрации работы *CASE*-структуры откажемся от операндов «Операнд 1» и «Операнд 2», чтобы сделать акцент на логике работы виртуального прибора (Рисунок 12).

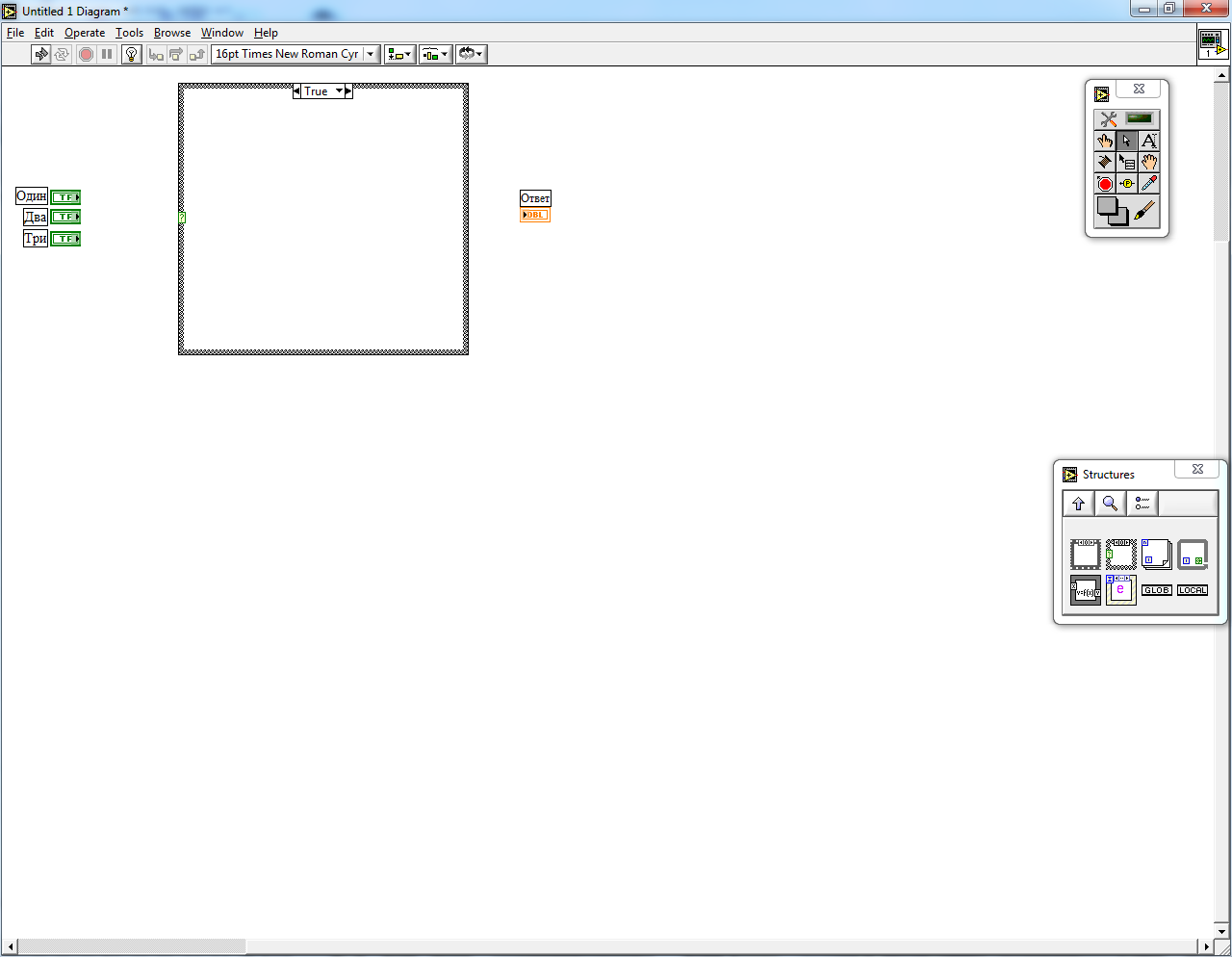


Рисунок 12 – Пример заготовки на блок-диаграмме фрагмента «Умного калькулятора», не требующего участия каких-либо операндов

Для вывода на числовой индикатор только одного из адекватных ответов лишь одна кнопка из трёх должна быть нажата, а другие не нажаты. То есть:

- либо кнопка «Один» нажата, кнопки «Два» и «Три» не нажаты,

- либо кнопка «Два» нажата, кнопки «Один» и «Три» не нажаты,

- либо кнопка «Три» нажата, кнопки «Один» и «Два» не нажаты.

Построим схему на блок-диаграмме в соответствии со сформулированной выше логикой. Начнём с первого условия: «Два» и «Три» логически перемножаются и инвертируются функцией объединённой арифметики и полученный результат логически умножается на «Один» (Рисунок 13).

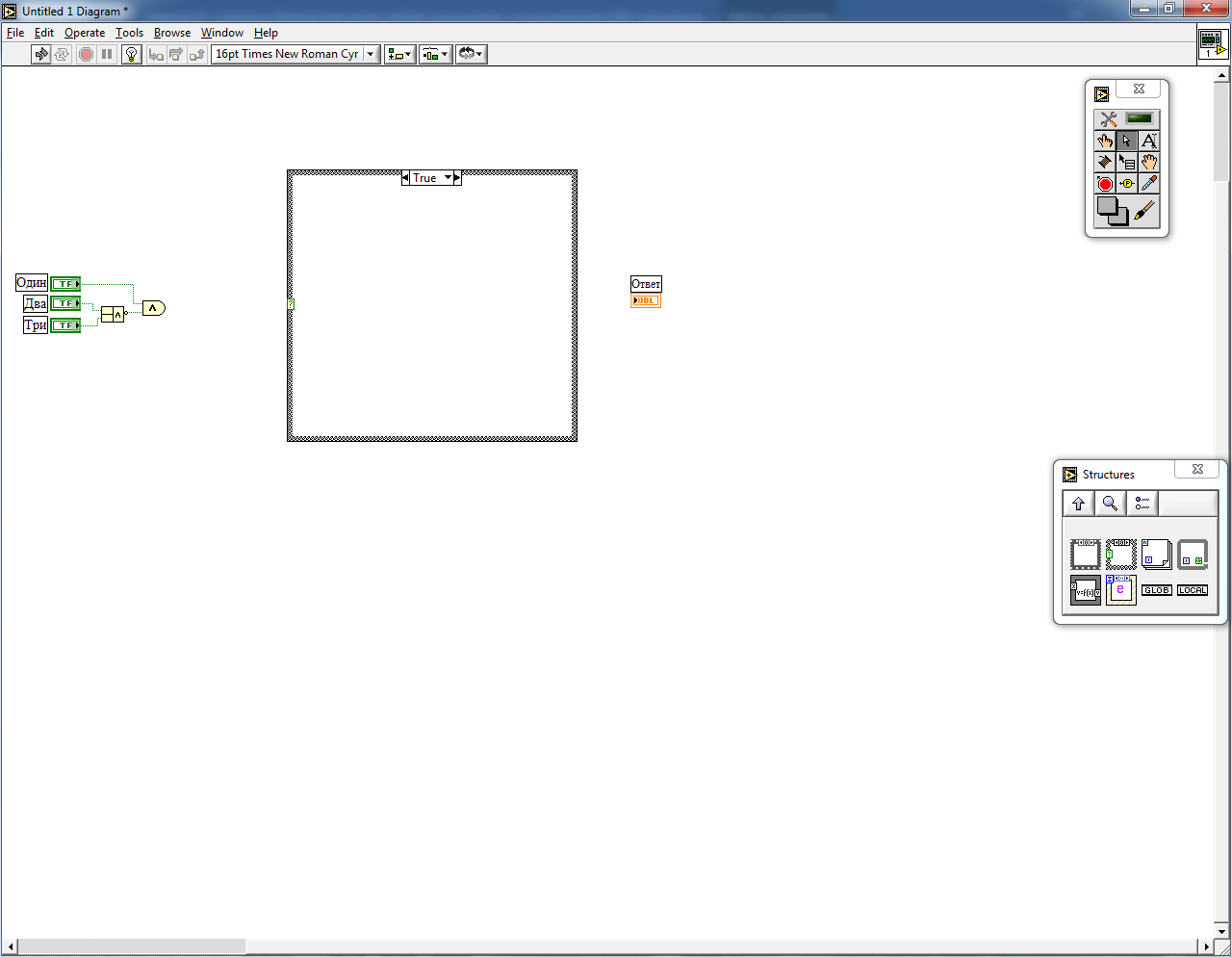


Рисунок 13 – Составление логики для нажатой кнопки «Один» и не нажатых кнопок «Два» и «Три»

По аналогии строится схема для второго условия: «Один» и «Три» логически перемножаются и инвертируются функцией объединённой арифметики и полученный результат логически умножается на «Два» (Рисунок 14).

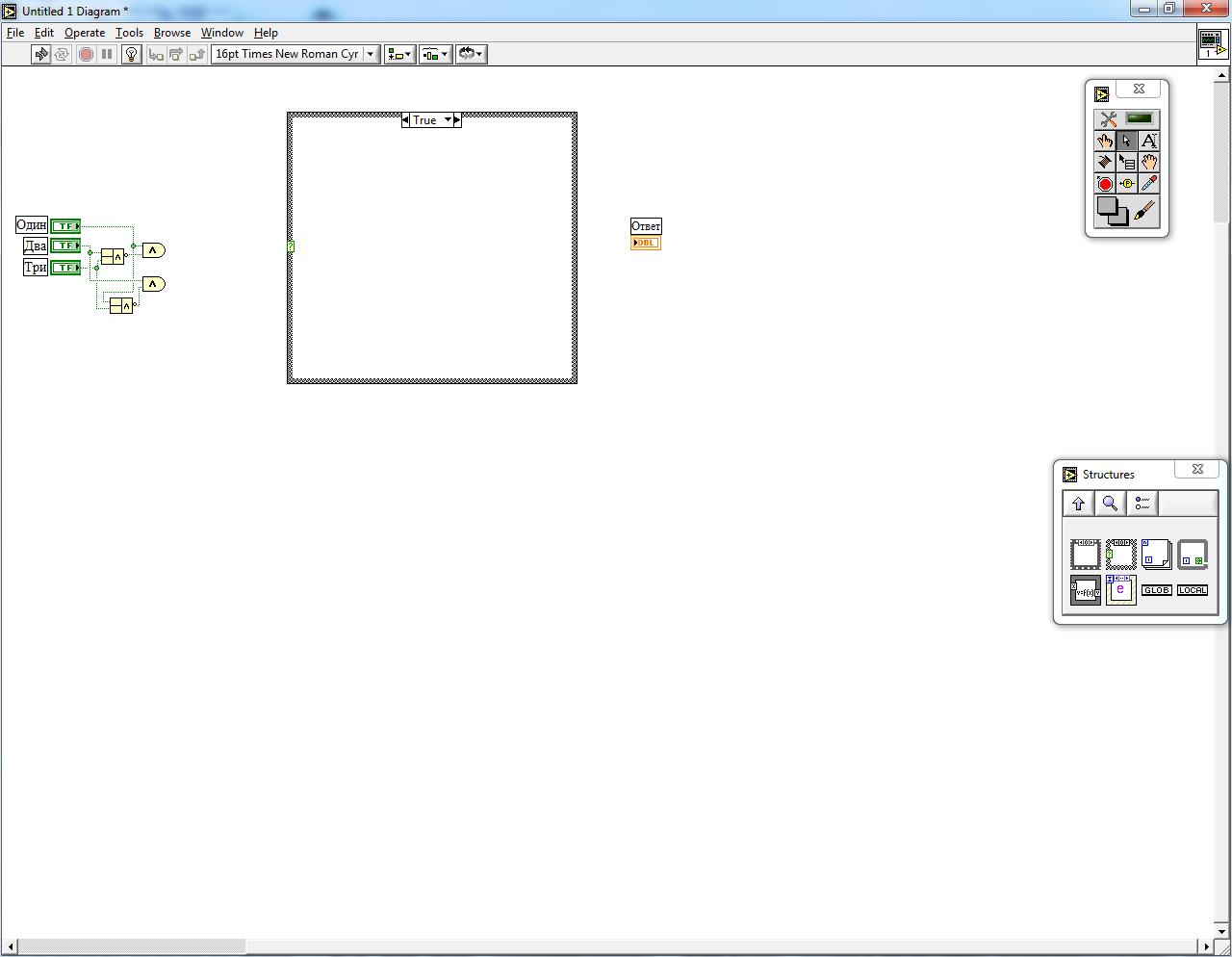


Рисунок 14 – Добавление логики для нажатой кнопки «Два» и не нажатых кнопок «Один» и «Три»

Схема дополняется учётом третьего условия (Рисунок 15).

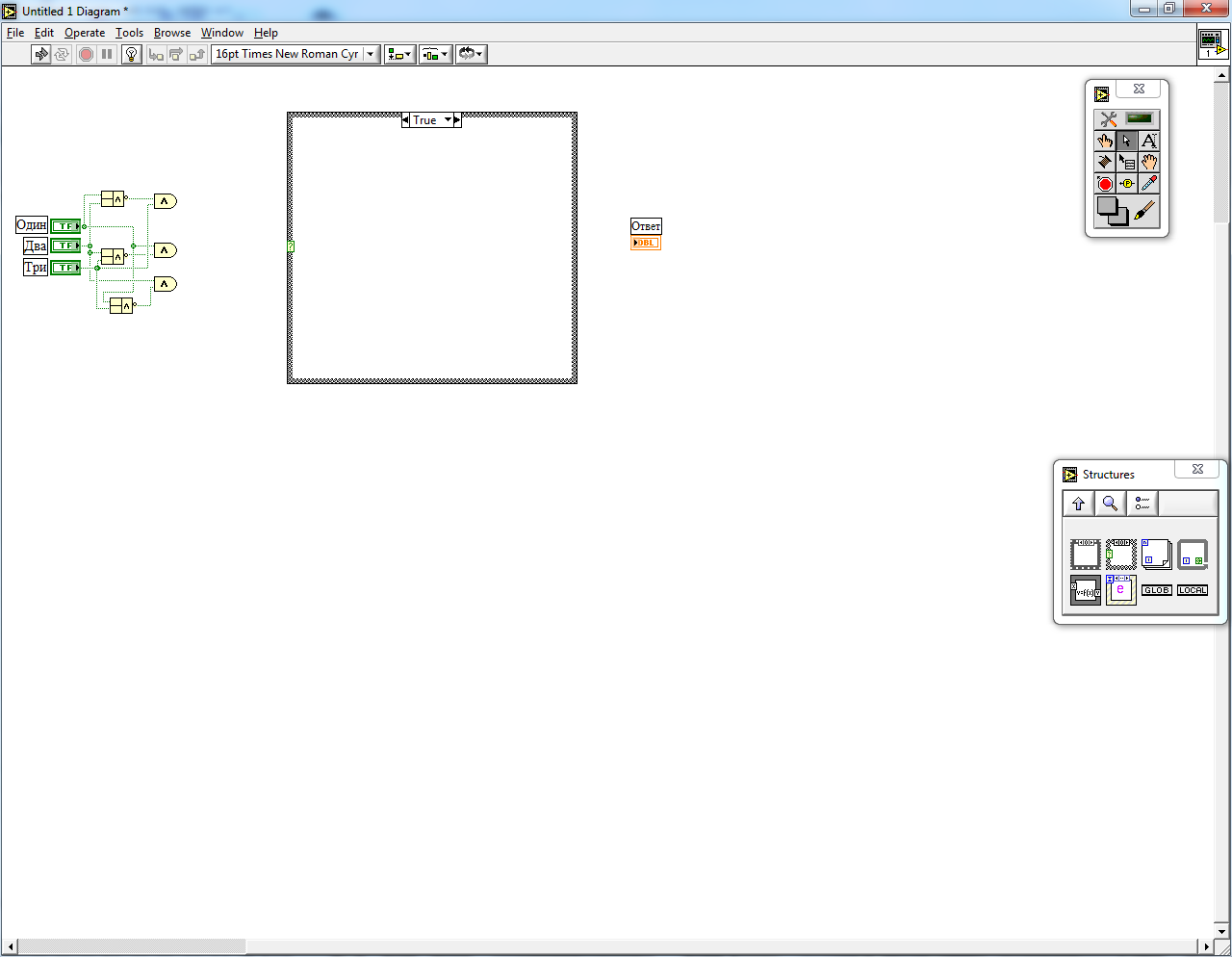


Рисунок 15 – Добавление логики для нажатой кнопки «Три» и не нажатых кнопок «Один» и «Два»

Решение задачи контроля нажатия одной единственной кнопки из трёх при составлении данной логической цепочки – это тот редкий случай, когда уместно применение операции исключающего ИЛИ. Это трудно принять с первого прочтения, поскольку само по себе в рассуждениях напрашивается обыкновенное логическое сложение, но это не так. Для доказательства составим таблицу истинности по логическому сложению (Таблица 1) и таблицу истинности для исключающего ИЛИ (Таблица 2).

Таблица 1 – Истинность объединения трёх операндов по ИЛИ

|  |  |  |  |
| --- | --- | --- | --- |
| **Один** | **Два** | **Три** | **OR** |
| false | false | false | **false** |
| false | false | true | **true** |
| false | true | false | **true** |
| false | true | true | **true** |
| true | false | false | **true** |
| true | false | true | **true** |
| true | true | false | **true** |
| true | true | true | **true** |

Таблица 2 – Истинность объединения трёх операндов по исключающему ИЛИ

|  |  |  |  |
| --- | --- | --- | --- |
| **Один** | **Два** | **Три** | **XOR** |
| false | false | false | **false** |
| false | false | true | **true** |
| false | true | false | **true** |
| false | true | true | **false** |
| true | false | false | **true** |
| true | false | true | **false** |
| true | true | false | **false** |
| true | true | true | **false** |

Хорошо видно, что объединение по ИЛИ – это истинность хотя бы одного из трёх. То есть если хотя бы один элемент отвечает истиной, то всё выражение отвечает истиной – логика не подходит. Исключающее ИЛИ гласит: только один должен ответить истиной, тогда всё выражение ответит истиной.

Объединение ранее сформулированных условий по исключающему ИЛИ объединённой арифметикой представлено на Рисунке 16. Вместе с тем для тестирования виртуального прибора в блок истинного результата всего выражения устанавливается плюс бесконечность.



Рисунок 16 – Объединение условий по исключающему ИЛИ (*XOR*). Проверка работоспособности составленной схемы: если условие выполняется – выводится «бесконечность» (*Inf*)

В блоке ложного результата (Рисунок 17) всего выражения выставляется константа не числового значения (*NaN*). Для создания константы не числового значения на блок-диаграмме размещается обыкновенная целочисленная константа и в ней вручную записывается с клавиатуры *NaN* при строгом учёте верхнего и нижнего регистра символов (две заглавных «*N*» объединяются между собой строчной «*a*»).

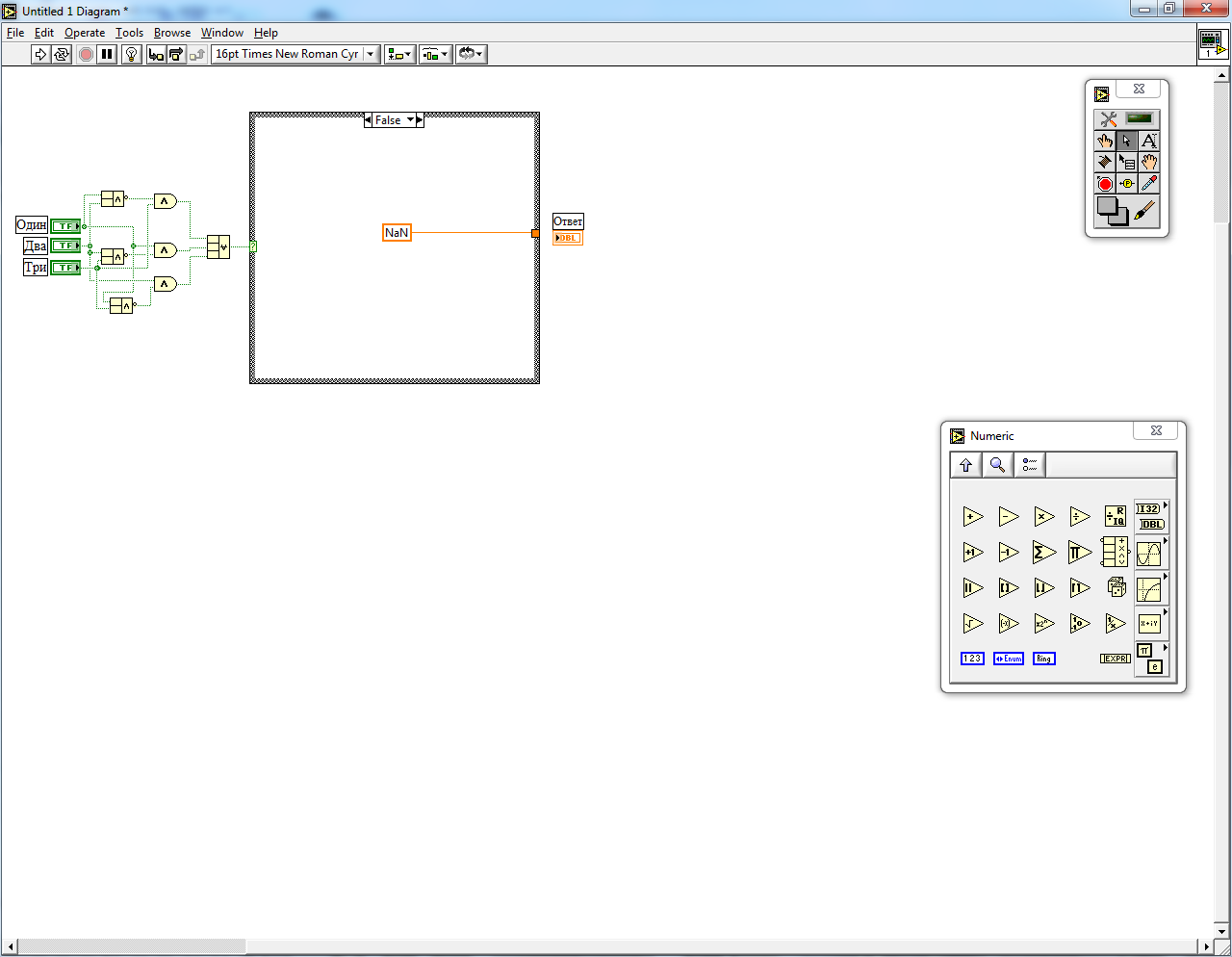


Рисунок 17 – Проверка работоспособности составленной схемы: если условие не выполняется – выводится «не числовое значение» (*NaN*)

После настройки содержимого всех блоков *CASE*-структуры на забудьте соединить её объединённый выход с численным индикатором «Ответ», как это показано на Рисунке 18. Это довольно частая ошибка, которая препятствует запуску виртуального прибора на исполнение.

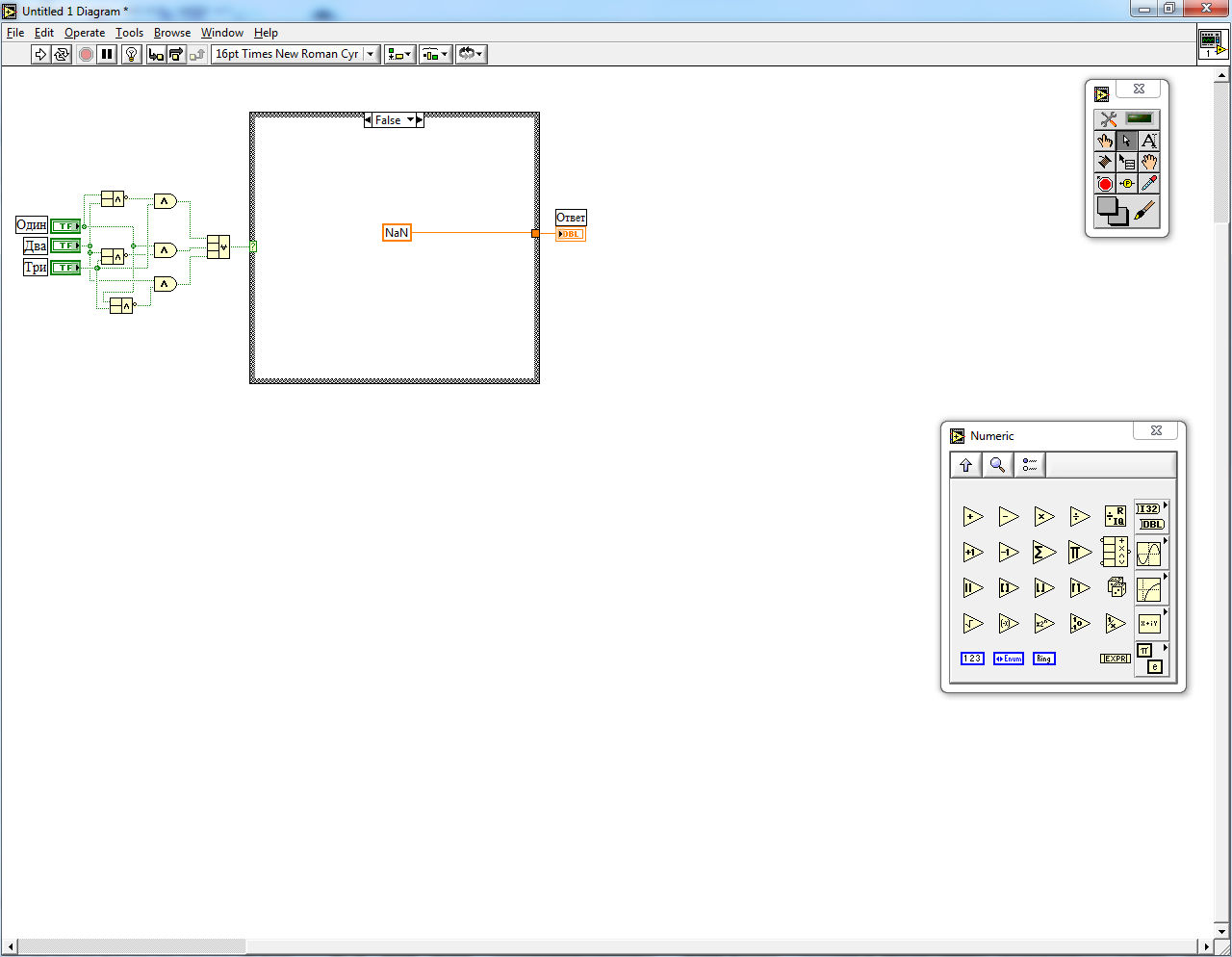


Рисунок 18 – Передача сформированного *CASE*-структурой результата на интерфейсный элемент, выводящий ответ

Запуск виртуального прибора при всех не нажатых кнопках даёт в результате не числовое значение – произошло попадание в блок ложного результата. Согласно составленной ранее таблице истинности так и должно быть (Рисунок 19).

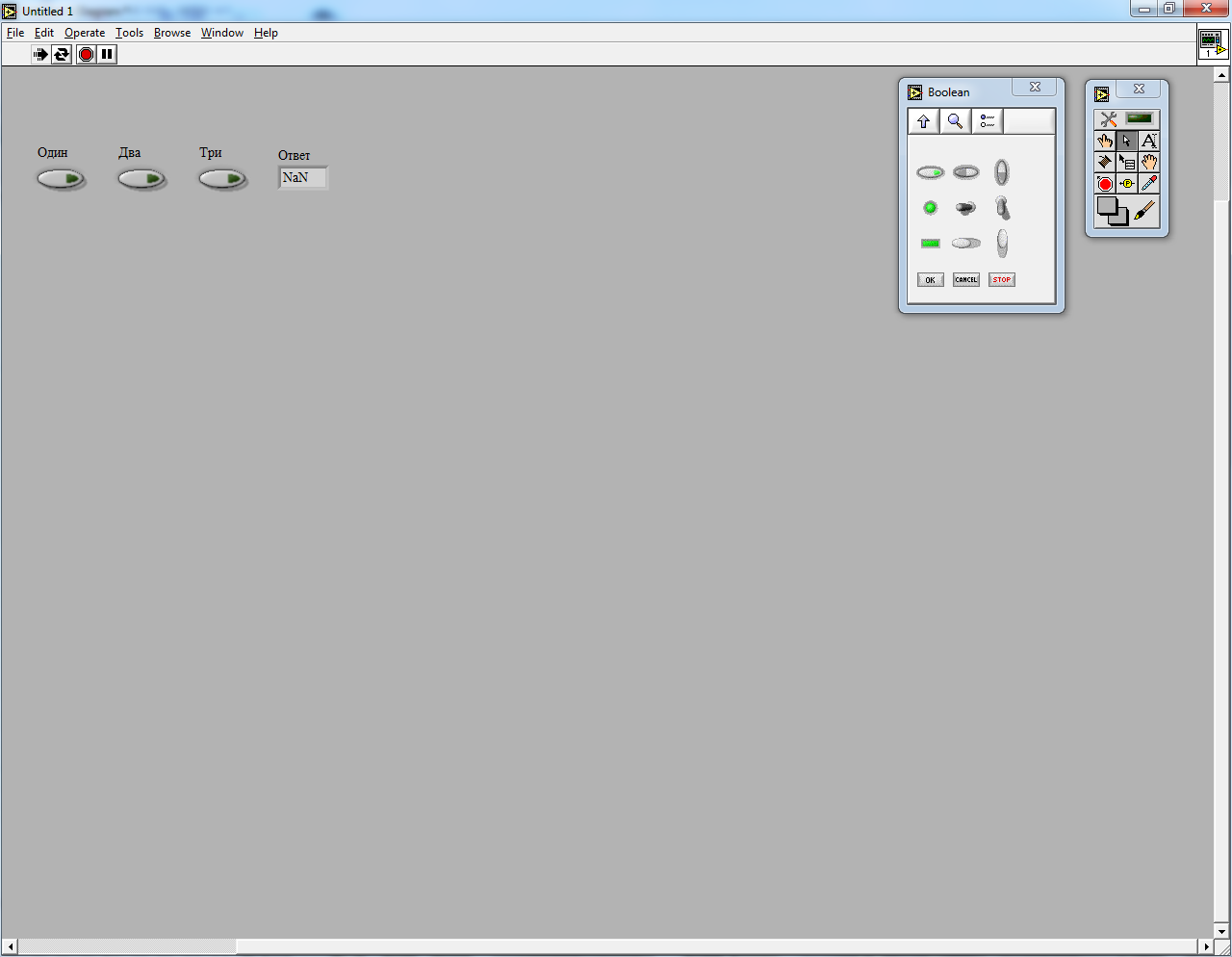


Рисунок 19 – Проверка работоспособности составленной логики: ничего не нажато – результат отрицательный

Запуск виртуального прибора при всех нажатых кнопках тоже даёт в результате не числовое значение в связи с попаданием в блок ложного результата *CASE*-структуры по результату выполнения объединённых логических операций. Согласно составленной ранее таблице истинности, опять же, так и должно быть (Рисунок 20).

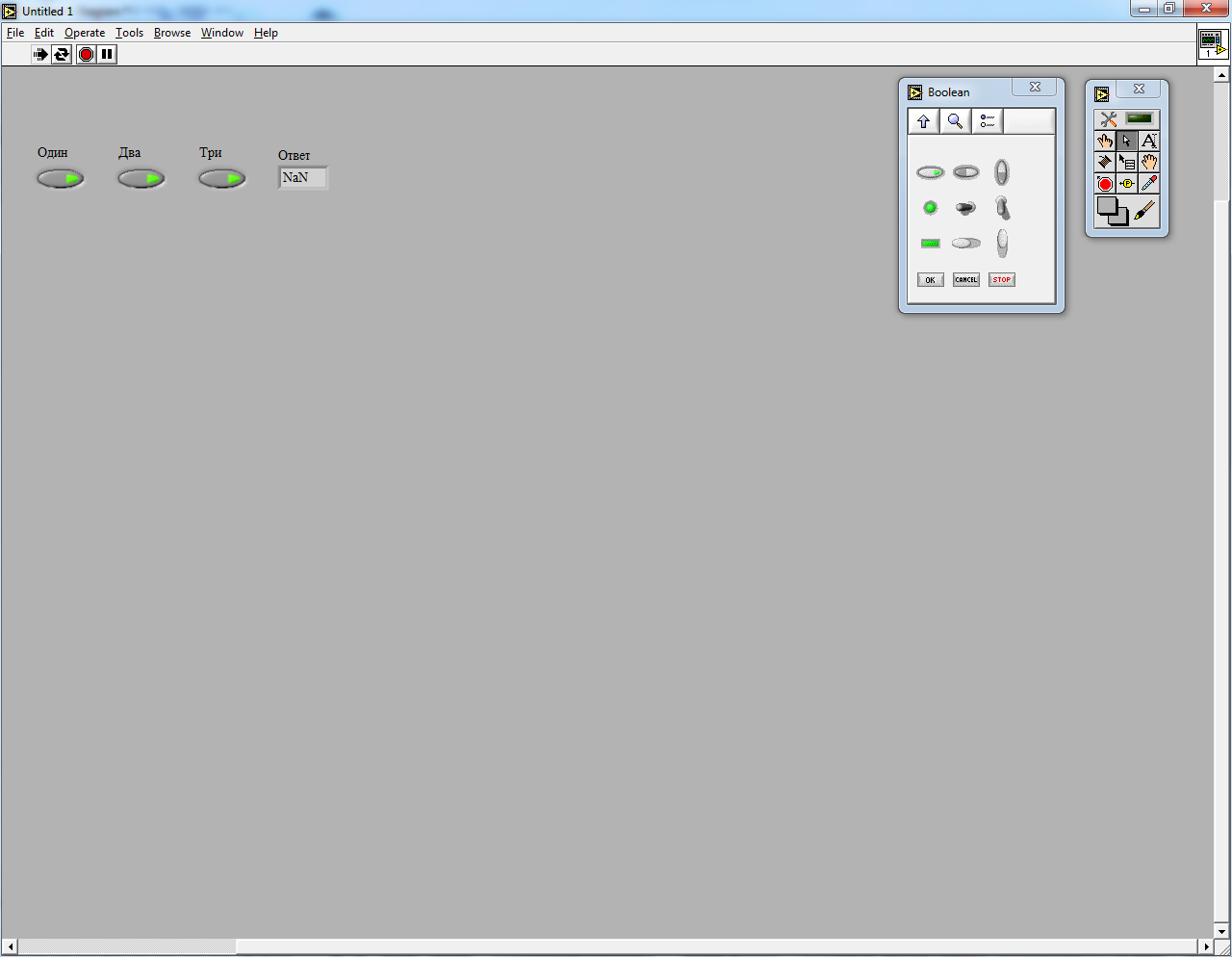


Рисунок 20 – Проверка работоспособности составленной логики: нажаты все кнопки сразу – результат отрицательный

Запуск с нажатыми «Один» и «Три» – шестая строчка таблицы (не считая заголовка). В ответе ложный результат, но так и должно быть (Рисунок 21).

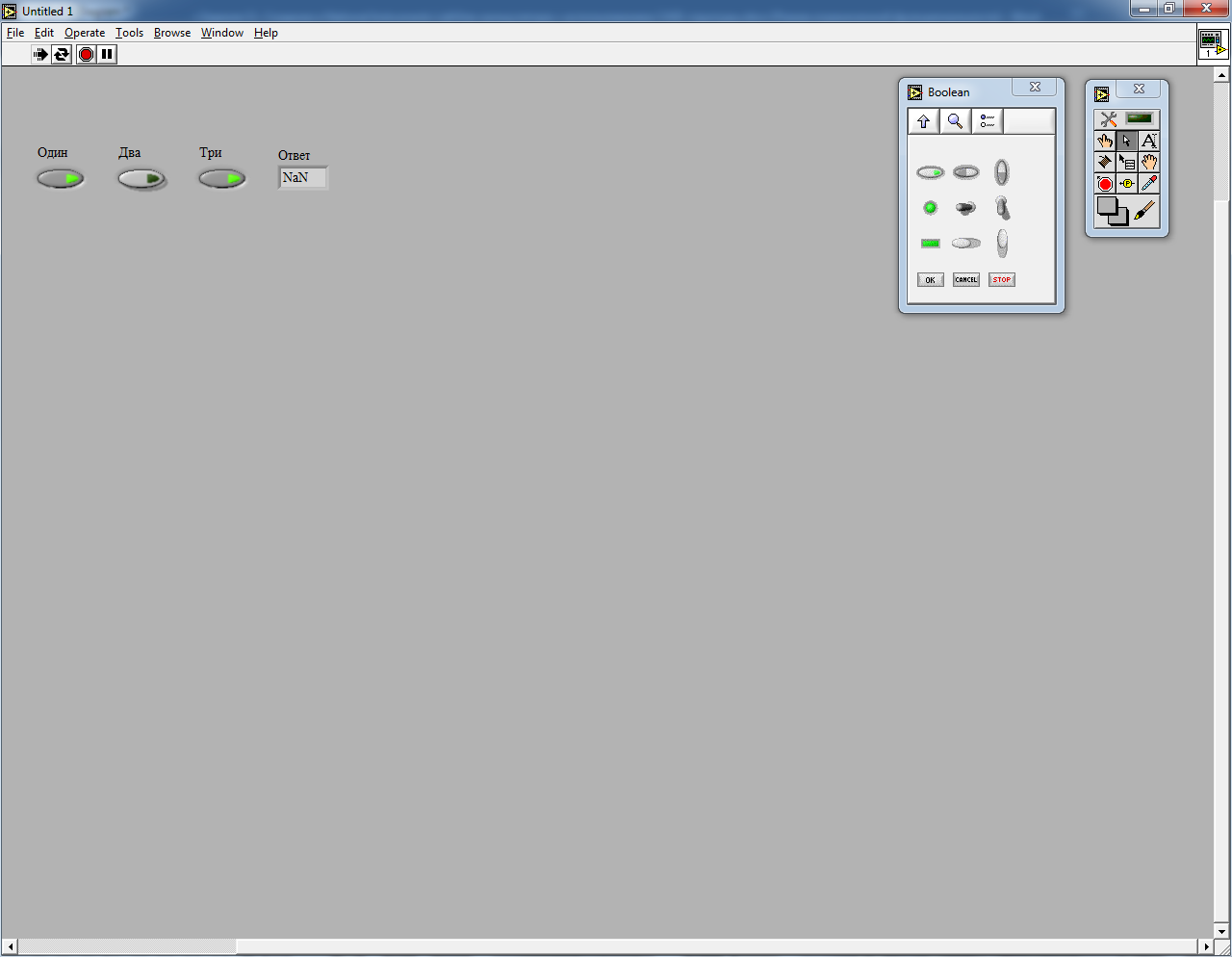


Рисунок 21 – Проверка работоспособности составленной логики: нажаты сразу две кнопки – результат отрицательный

Запуск с нажатыми «Один» и «Два» – седьмая строчка таблицы (не считая заголовка). В ответе ложный результат. Адекватно (Рисунок 22).

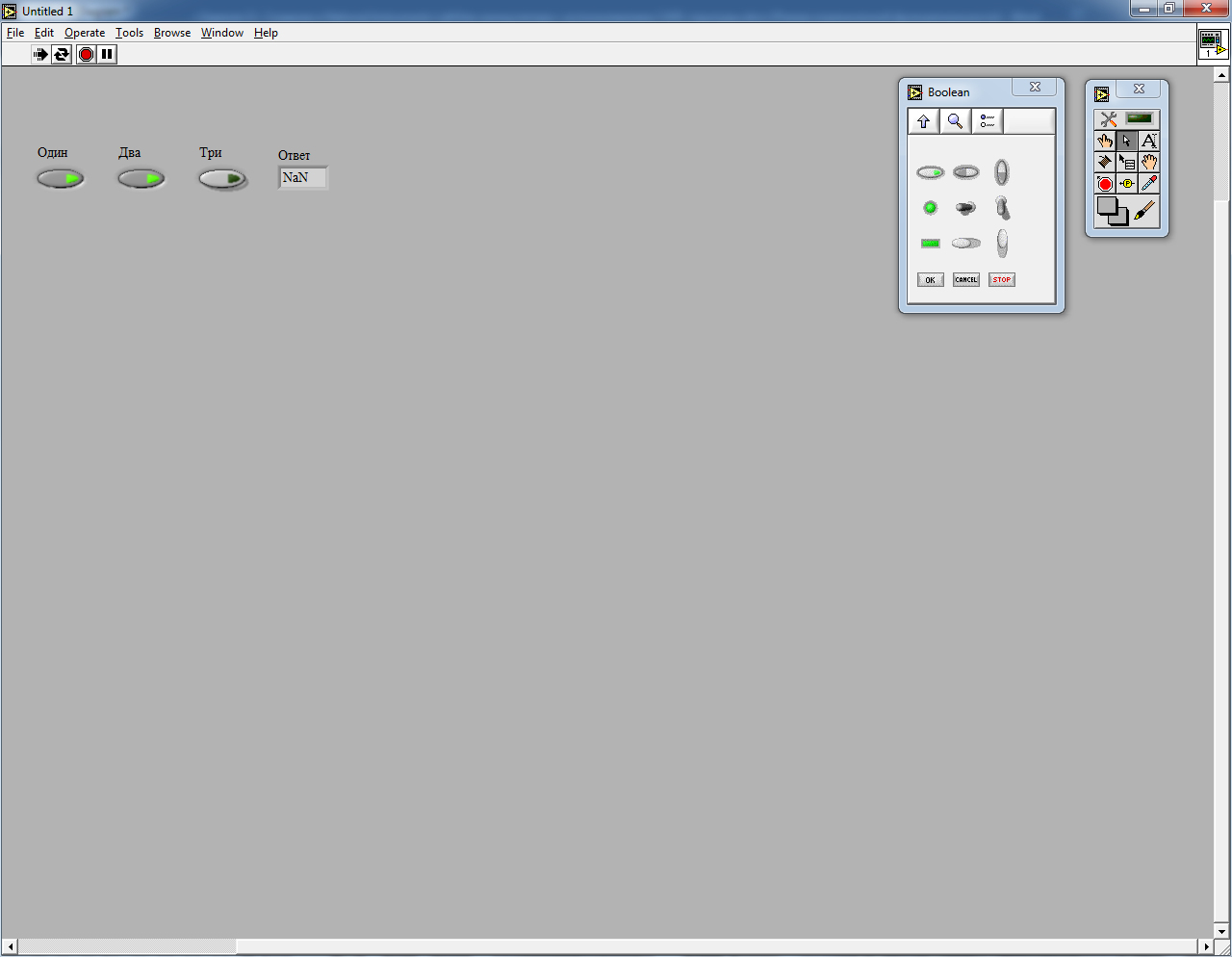


Рисунок 22 – Проверка работоспособности составленной логики: нажаты сразу две другие кнопки – результат отрицательный

Наконец, встречается только «Один» – это пятая строчка таблицы. В ответе плюс бесконечность вместо не числового значения – попали в блок истинного результата. Адекватно (Рисунок 23).

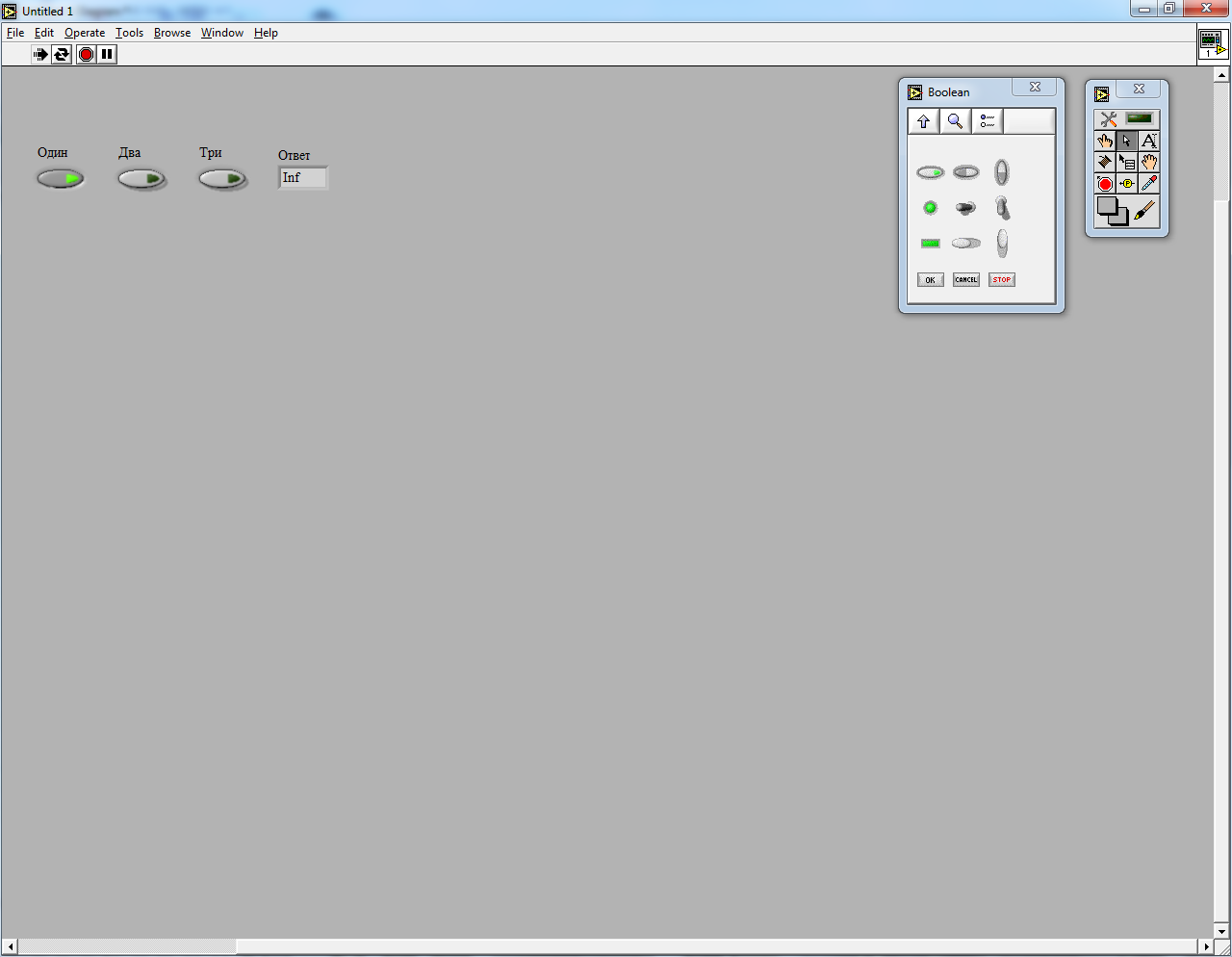


Рисунок 23 – Проверка работоспособности составленной логики: нажата только одна кнопка – результат положительный

После нехитрого тестирования продолжим развитие кода заготовки виртуального прибора «Умный калькулятор» в направлении желаемого результата. В блоке ложного результата контрольной (внешней) *CASE*-структуры оставляем константу не числового значения, а в блоке истинного результата размещаем ещё одну, вложенную *CASE*-структуру. На логический вход вложенной *CASE*-структуры подаём ветвь логики, отвечающей за нажатие «Один» и не нажатие «Два» и «Три». Итого, если нажата кнопка «Один», в блоке истинного результата должна зарождаться численная константа, равная «1» (Рисунок 24). Но если «Один» не нажата, то остаётся ещё два возможных варианта, которые необходимо проверить (либо нажата «Два», либо нажата «Три»). Это ещё одна вложенная *CASE*-структура.

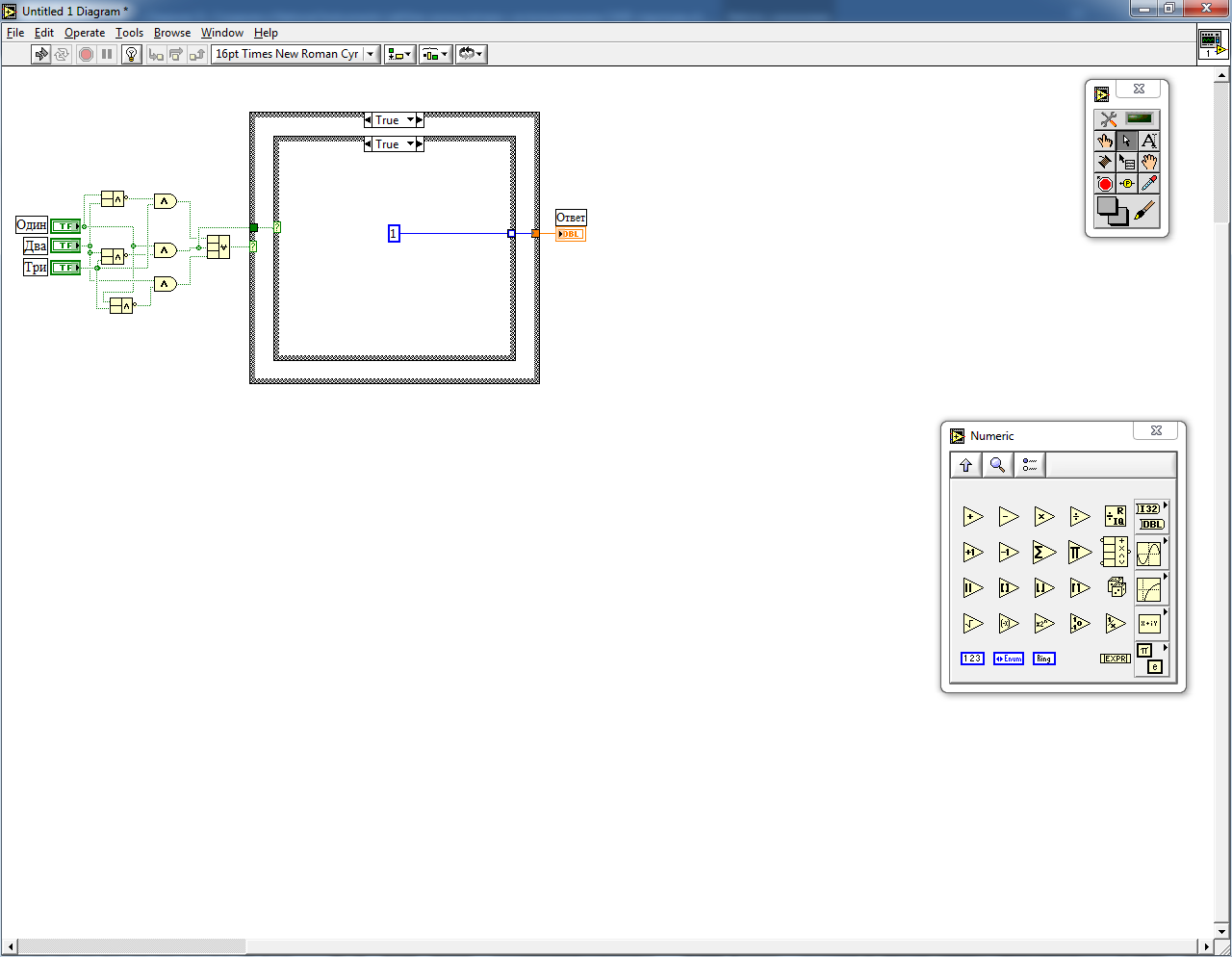


Рисунок 24 – Развитие структуры программы: в случае, если нажата только одна кнопка, и эта кнопка – «Один» должно выводиться в качестве ответа целочисленное значение, равное «1»

На логический вход новой *CASE*-структуры подаём проверку условия нажатия на «Два» (Рисунок 25). Если, действительно, нажата «Два», то должна генерироваться константа, равная «2».

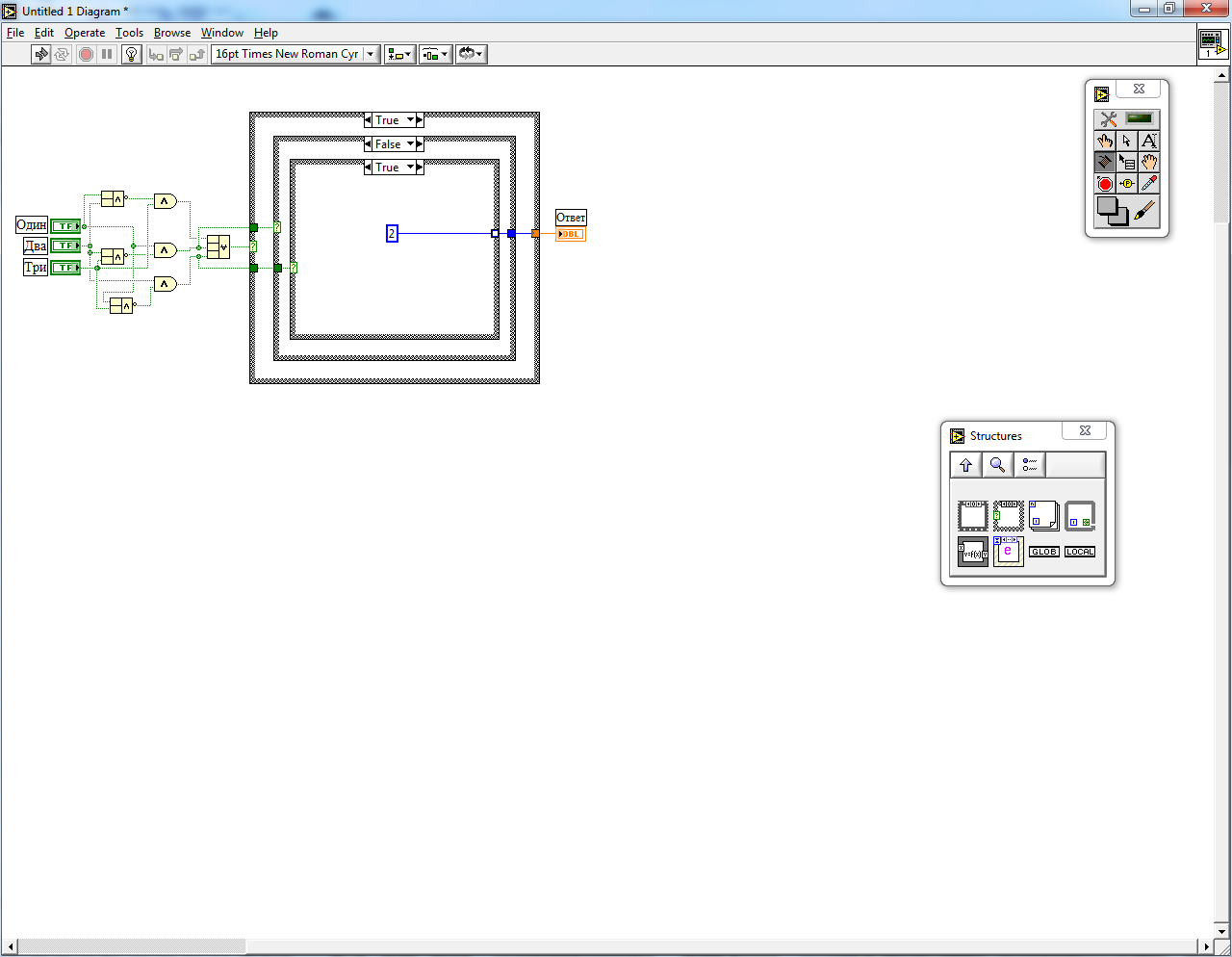


Рисунок 25 – Развитие структуры программы: в случае, если нажата только одна кнопка, и эта кнопка не «Один», а «Два» – должно выводиться в качестве ответа целочисленное значение, равное «2»

Но если не «Два», то остаётся один единственный вариант, что нажата «Три», потому его дополнительной *CASE*-структурой контролировать не нужно, константу со значением «3» генерирует оставшийся незаполненным блок ложного результата (Рисунок 26). Неопытные программисты могут захотеть надстроить ещё одну *CASE*-структуру с целью проверки условия для ветки «Три» по аналогии. Это не будет ошибкой, но в этом случае останется пустой блок ложного результата, в который никогда, ни при каких условиях курсор виртуального прибора не попадёт. Это случай так называемой кодовой избыточности, с которой необходимо бороться.

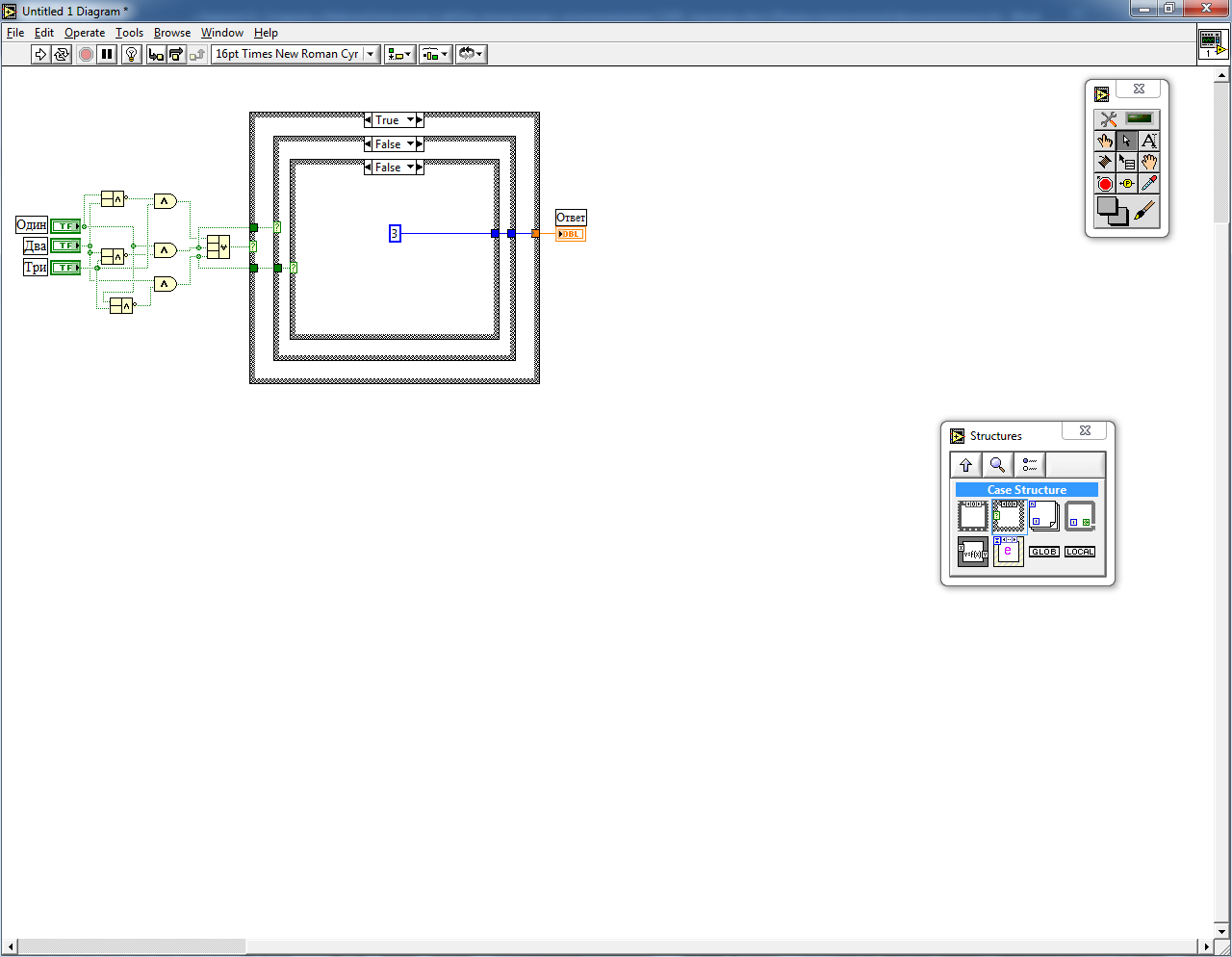


Рисунок 26 – Развитие структуры программы: в случае, если нажата только одна кнопка, и эта кнопка не «Один» и не «Два», то остаётся заключительный вариант – нажата кнопка «Три» и, соответственно, должно выводиться в качестве ответа целочисленное значение, равное «3»

Нижеследующие иллюстрации (Рисунок 27-29) демонстрируют адекватность работы созданного фрагмента виртуального прибора «Умный калькулятор» с учётом ранее проведенных тестов по защите от одновременного выбора нескольких операций.

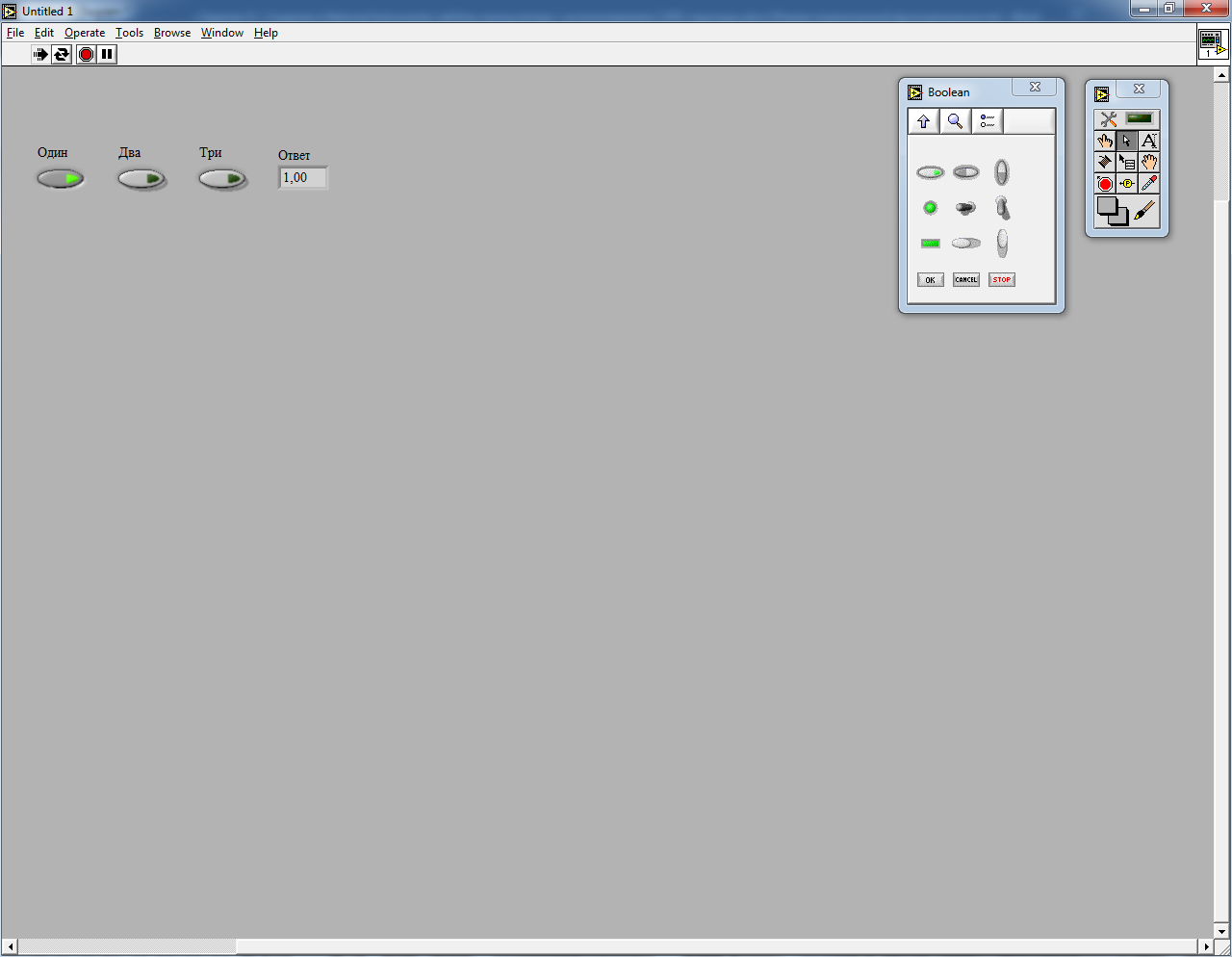


Рисунок 27 – Проверка работоспособности программного обеспечения: нажата кнопка «Один» и в качестве ответа выдаётся значение, равное «1»

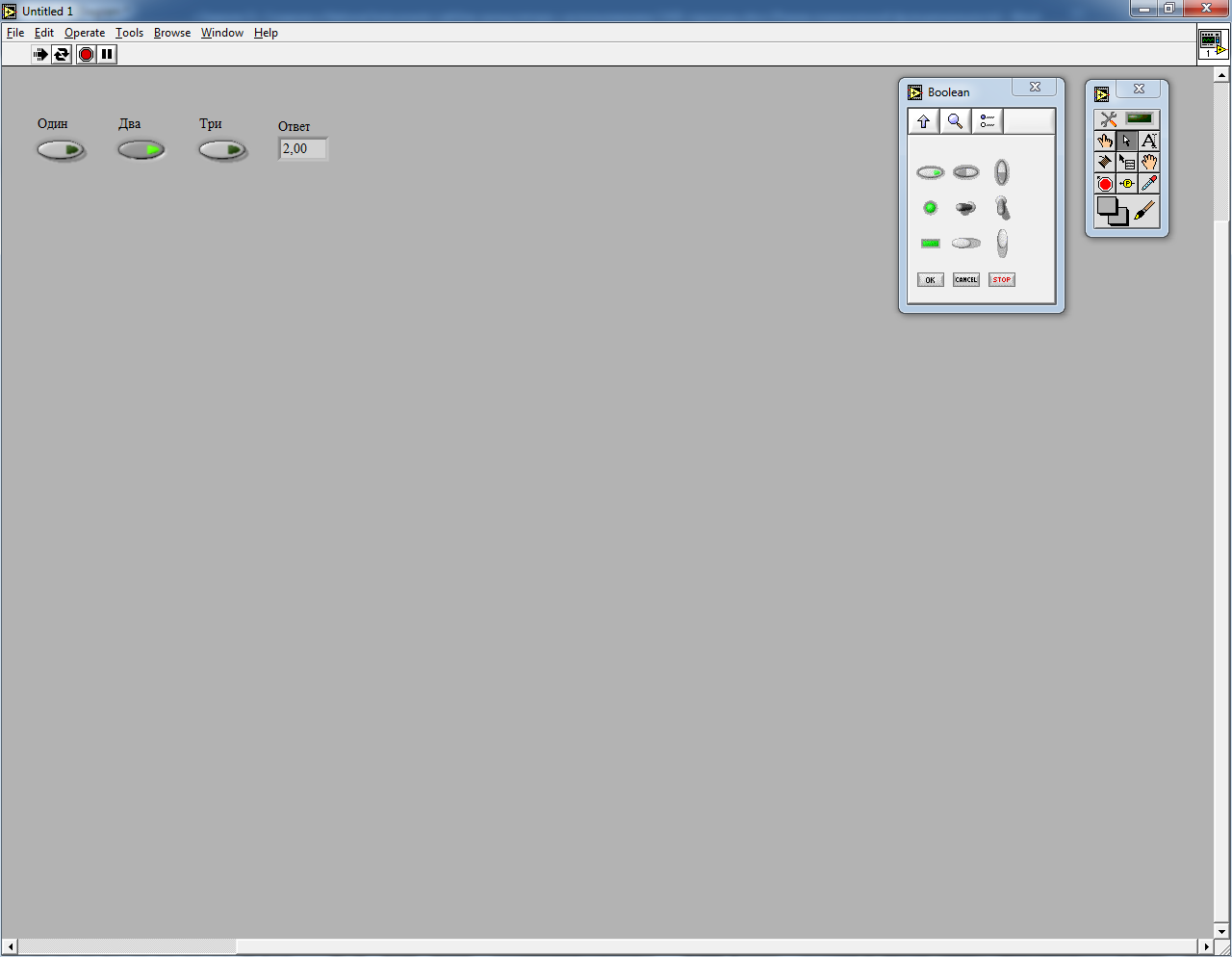


Рисунок 28 – Проверка работоспособности программного обеспечения: нажата кнопка «Два» и в качестве ответа выдаётся значение, равное «2»

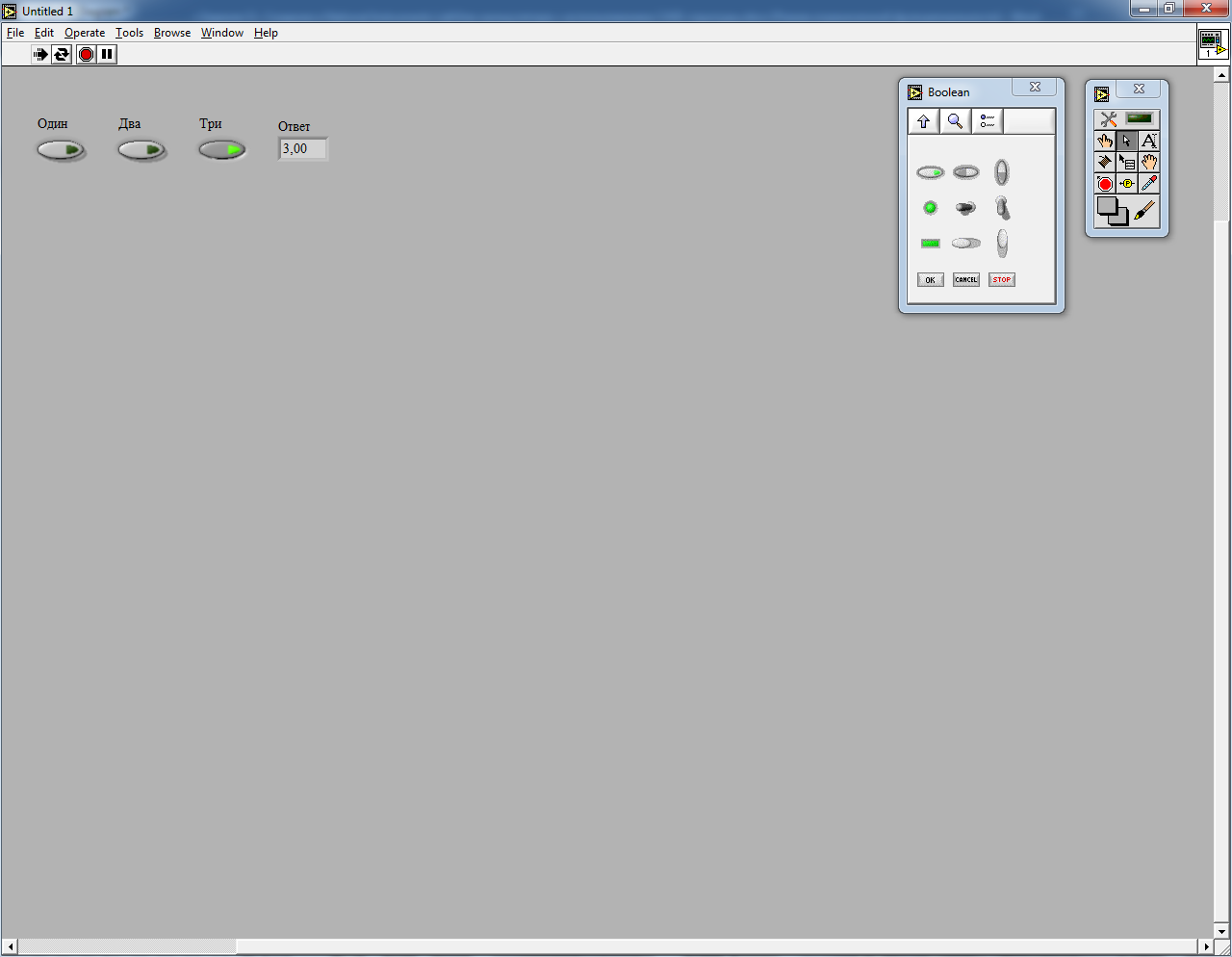


Рисунок 29 – Проверка работоспособности программного обеспечения: нажата кнопка «Три» и в качестве ответа выдаётся значение, равное «3»

## 4.4 Варианты индивидуального задания

Таблица 3 – Исходные данные для выполнения индивидуального задания

|  |  |
| --- | --- |
| **№ вар.** | **Содержание работы** |
| **1.** | Использовать **три** операнда.  Предусмотреть операции:  - сложения;  - инверсии суммы;  - инверсии суммы объединённой арифметикой;  - сложения объединённой арифметикой. |
| **2.** | Использовать **два** операнда.  Предусмотреть операции:  - умножения;  - деления первого на второй;  - деления второго на первый;  - масштабирования на степень двойки;  - логарифма второго по основанию первого. |
| **3.** | Использовать **один** операнд.  Предусмотреть операции:  - абсолютного значения;  - сигнума;  - синуса;  - экспонирования. |
| **4.** | Использовать **два** операнда.  Предусмотреть операции:  - деления по модулю первого;  - деления по модулю второго;  - инкрементирования первого. |
| **5.** | Использовать **один** операнд.  Предусмотреть операции:  - секанса;  - натурального логарифма;  - декрементирования;  - инверсии;  - получения Пифагоровой константы. |
| **6.** | Использовать **четыре** операнда.  Предусмотреть операции:  - инверсии суммы объединённой арифметикой;  - инверсии разности объединённой арифметикой. |

|  |  |
| --- | --- |
| **7.** | Использовать **два** операнда.  Предусмотреть операции:  - арктангенса с двумя параметрами;  - арктангенса первого;  - арктангенса второго;  - тангенса первого;  - тангенса второго. |
| **8.** | Использовать **один** операнд.  Предусмотреть операции:  - экспонирования;  - взятия логарифма по основанию 2;  - натурального логарифма;  - десятичного логарифма. |
| **9.** | Предусмотреть вывод констант:  - Пифагоровой;  - основания натурального логарифма;  - гравитационной;  - нечислового значения;  - плюс бесконечности;  - минус бесконечности. |
| **10.** | Использовать **два** операнда.  Предусмотреть операции:  - остатка от деления первого на второй;  - остатка от деления второго на первый. |
| **11.** | Использовать **три** операнда.  Предусмотреть операции:  - квадратного корня из суммы квадратов;  - квадратного корня из модуля разности квадратов;  - квадратного корня из модуля разности;  - квадратного корня из модуля суммы. |
| **12.** | Использовать **один** операнд.  Предусмотреть операции:  - арккосинуса;  - округления до ближайшего целого;  - квадратного корня. |
| **13.** | Использовать **три** операнда.  Предусмотреть операции:  - нахождения периметра треугольника;  - нахождения площади треугольника;  - нахождения высоты треугольника. |

|  |  |
| --- | --- |
| **14.** | Использовать **два** операнда.  Предусмотреть операции:  - обращение суммы;  - корень квадратный из суммы;  - деление второго на первый;  - масштабирование на степень двойки;  - логарифм второго по основанию первого. |
| **15.** | Использовать **один** операнд.  Предусмотреть операции:  - нахождения диаметра окружности;  - нахождения площади круга;  - нахождения длины окружности. |
| **16.** | Использовать **четыре** операнда.  Предусмотреть операции:  - нахождения периметра прямоугольника;  - нахождения площади прямоугольника;  - нахождения диагонали прямоугольника. |
| **17.** | Использовать **три** операнда.  Предусмотреть операции:  - нахождения объёма фигуры;  - нахождения площади поперечного сечения фигуры;  - нахождения площади продольного сечения фигуры. |
| **18.** | Использовать **один** операнд.  Предусмотреть операции:  - взятия квадратного корня;  - тангенса;  - обращения;  - возведения двойки в степень. |
| **19.** | Использовать **один** операнд.  Предусмотреть операции:  - математического округления;  - округления в большую сторону;  - округления в меньшую сторону. |
| **20.** | Использовать **два** операнда.  Предусмотреть операции:  - вывода численной константы, равной «8»;  - вычитания второго из первого;  - вычитания первого из второго;  - суммы синуса первого и косинуса второго (синус и косинус в одной операции). |
| **21.** | Использовать **один** операнд.  Предусмотреть операции:  - вычисления значения функции отсчётов;  - декрементирования. |

|  |  |
| --- | --- |
| **22.** | Использовать **три** операнда.  Предусмотреть операции:  - произведения трёх;  - разности суммы первого и третьего из второго;  - суммы трёх;  - частного произведения второго на третье на первое. |
| **23.** | Использовать **один** операнд.  Предусмотреть операции:  - косинуса;  - степени десяти;  - вывода постоянной Планка;  - инкрементирования;  - декрементирования. |
| **24.** | Предусмотреть вывод констант:  - нечисловое значение;  - Авогадро;  - обращённой Пифагорову;  - натурального логарифма десяти. |
| **25.** | Использовать **три** операнда.  Предусмотреть операции:  - вычисления периметра треугольника;  - вычисления высоты треугольника;  - вычисления площади треугольника. |
| **26.** | Использовать **один** операнд.  Предусмотреть операции:  - возведения двойки в степень;  - котангенса;  - вывода обращённого основания натурального логарифма. |
| **27.** | Использовать **два** операнда.  Предусмотреть операции:  - синуса суммы;  - экспонирования суммы;  - обращения суммы;  - возведения двойки в степени суммы. |
| **28.** | Использовать **шесть** операндов.  Предусмотреть операции:  - произведения объединённой арифметикой;  - суммирования объединённой арифметикой. |
| **29.** | Использовать **один** операнд.  Предусмотреть операции:  - косеканса;  - натурального логарифма, проходящего через начало координат. |

|  |  |
| --- | --- |
| **30.** | Использовать **два** операнда.  Предусмотреть операции:  - косинуса разности;  - вычитания из первого квадрата тангенса второго. |

## 4.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

В качестве первого приближения к созданию виртуального прибора «Умный калькулятор» рассмотрим один из его возможных графических пользовательских интерфейсов. Для понимания принципов наращивания (масштабирования) структуры виртуального прибора «Умный калькулятор» достаточно рассмотреть три операции. Каждой операции ставится в соответствие логический контроллер – кнопка. Так на передней панели должны быть размещены три кнопки, которые назовём слева направо «Один», «Два» и «Три», соответственно. В таком случае подразумевается вывод на численный индикатор «Ответ» по нажатию своей, заранее определённой целочисленной константы: либо «1», либо «2», либо «3».

Стоит отметить, что для большинства вариантов данной работы требуется использование операндов для выполнения над ними определённых операций. Под каждый операнд выделяется свой собственный численный контроллер. Заготовка интерфейса, представленная на Рисунке 8, содержит все вышеуказанные элементы, однако, в рассматриваемом примере «Операнд 1» и «Операнд 2» использованы не будут.

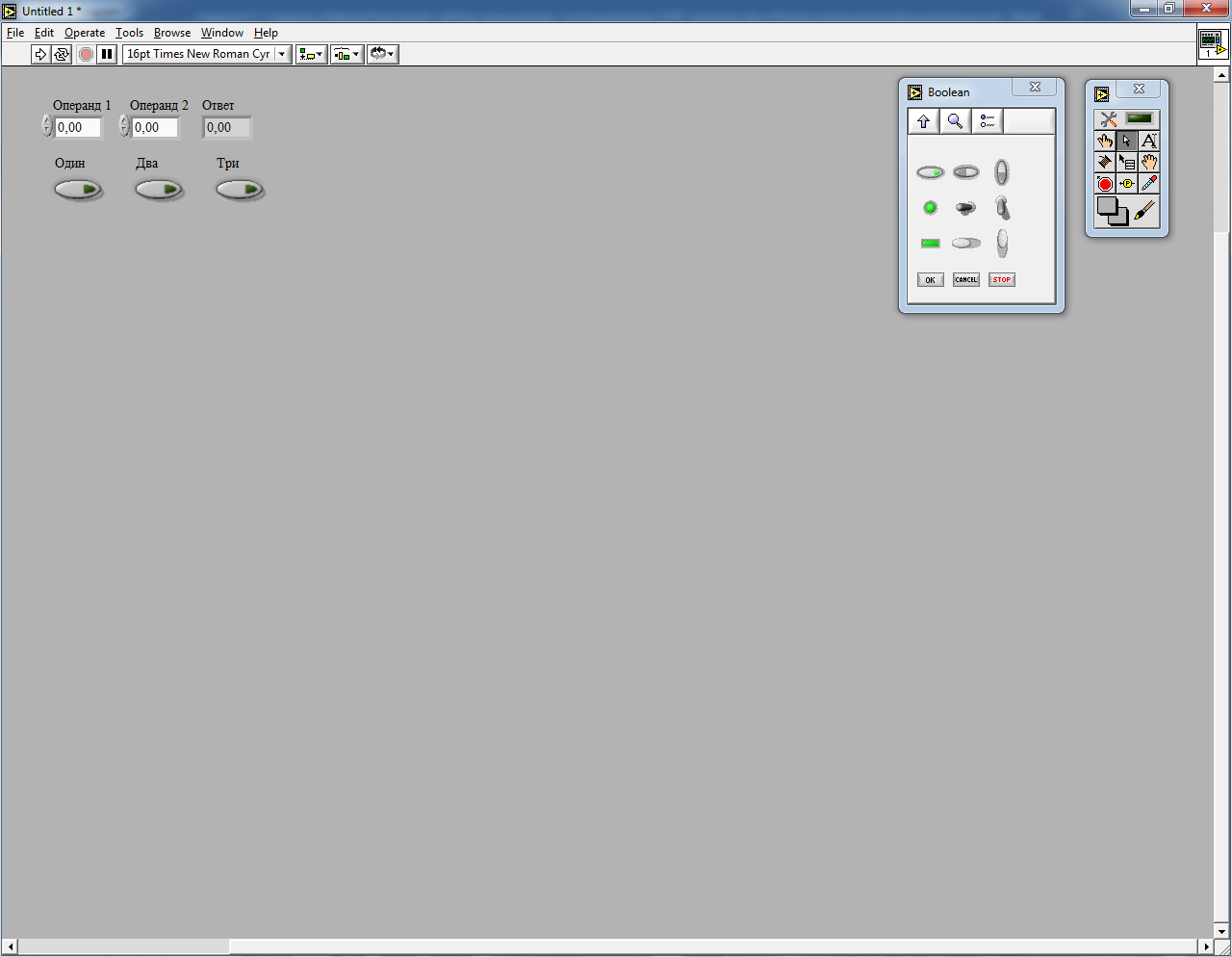


Рисунок 8 – Пример графического пользовательского интерфейса виртуального прибора «Умный калькулятор» с двумя входными операндами, тремя операциями и одним индикатором с численным результатом

При таком наборе интерфейсных элементов блок-диаграмма будет содержать в себе те же элементы, что представлены на Рисунке 9. Помимо них к заготовке добавлена *CASE*-структура. Заметим, что для большей компактности размещения элементов на блок-диаграмме при использовании инструмента позиционирования / изменения размера / выбора (*Position / Size / Select*) наименования этих элементов размещены слева от них.

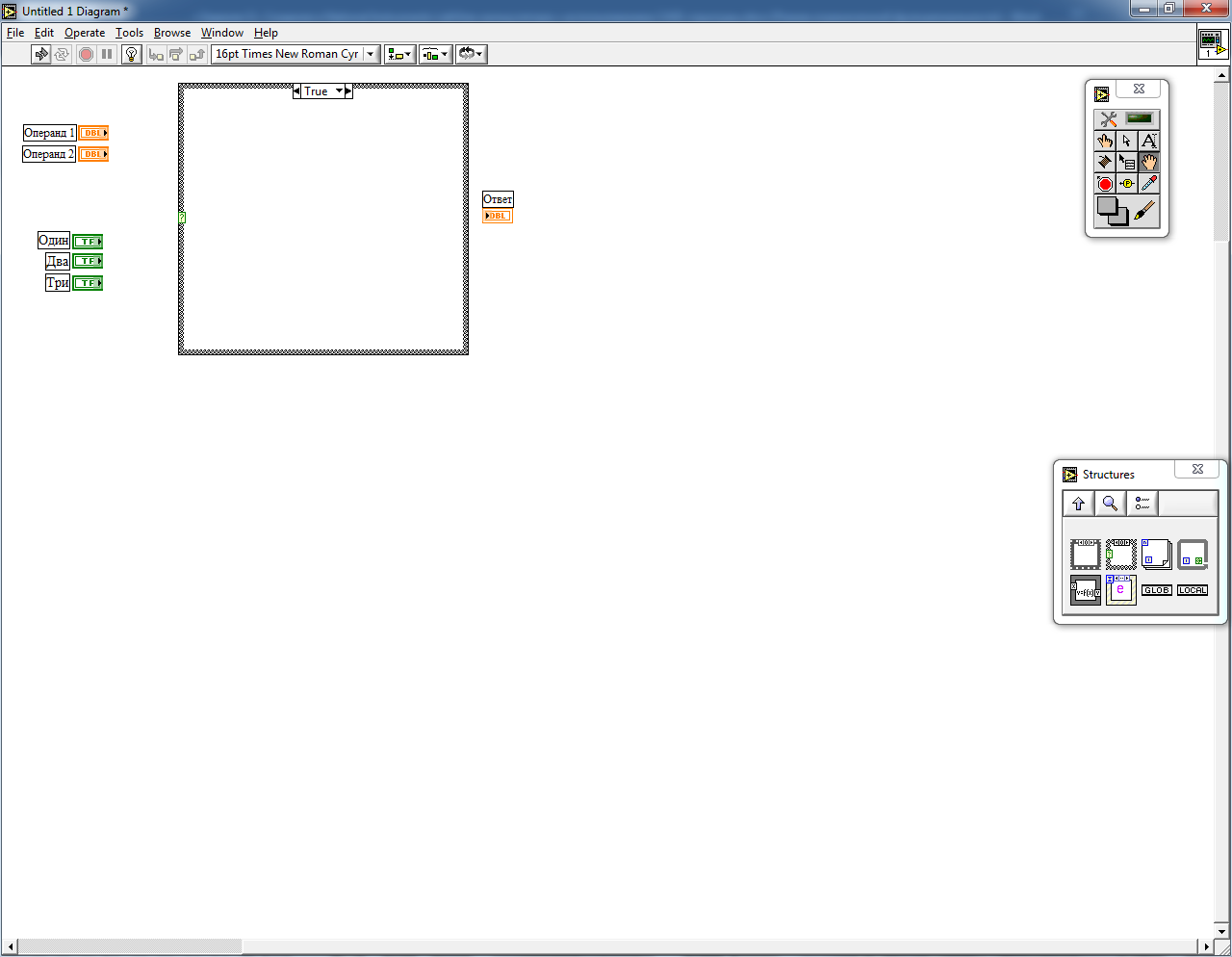


Рисунок 9 – Заготовка блок-диаграммы для последующей настройки рассматриваемого примера

Итого в заготовке использованы элементы из следующих функциональных разделов:

- численные функции;

- логические функции;

- структуры.

Рассмотрение численных и логических уже затрагивалось в более ранних работах Учебной практики, связанных с *National Instruments LabView*. Структуры ранее не использовались. Рассмотрим их расположение в перечне функций, размещаемых на блок-диаграмме (Рисунок 10).

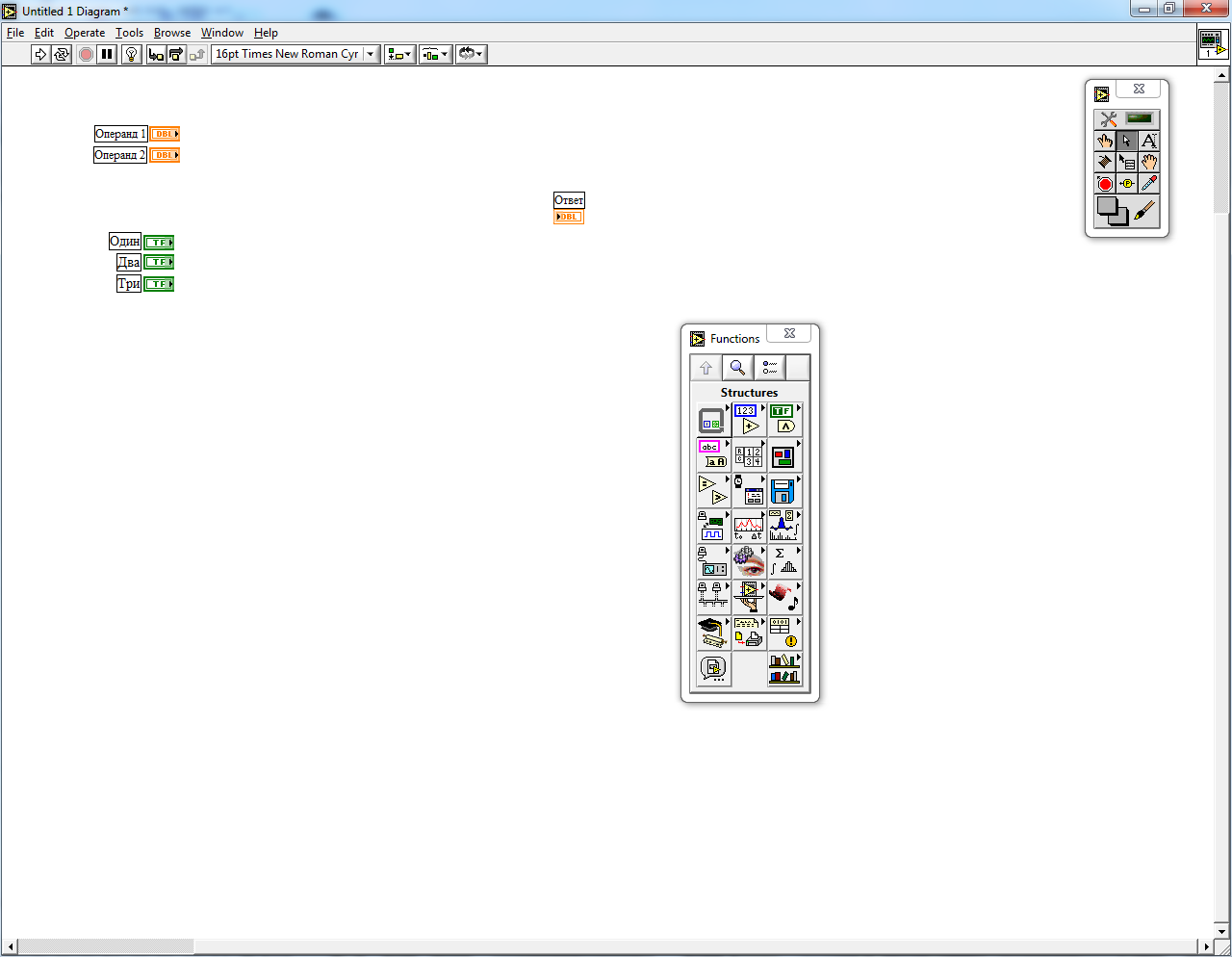


Рисунок 10 – Маркировка раздела структур в перечне функций, размещаемых на блок-диаграмме

Вход в раздел структур сопровождается выводом на экран окна, содержащего все существующие в *National Instruments* *LabView* структуры (их не много), которые при чтении слева направо следующие (Рисунок 11):

- структура последовательности (*Sequence Structure*);

- *CASE*-структура (*Case Structure*);

- структура цикла *For* (*For Loop*);

- структура цикла *While* (*While Loop*);

- узел-формула (*Formula Node*);

- структура событий (*Event Structure*);

- глобальная переменная (*Global Variable*);

- локальная переменная (*Local Variable*).

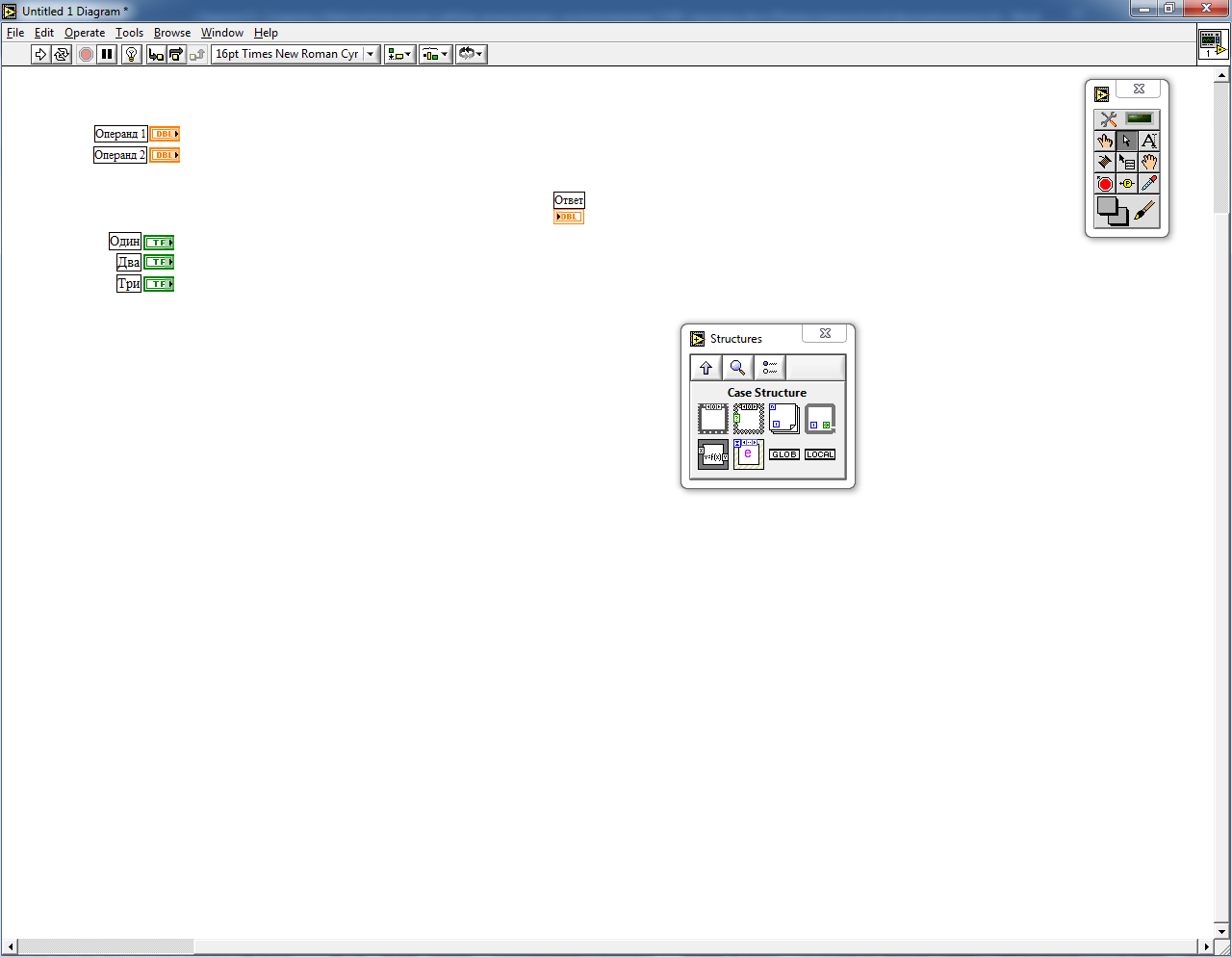


Рисунок 11 – Содержимое раздела структур блок-диаграммы с маркировкой *CASE*-структуры (*Case Structure*)

*CASE*-структура может быть использована в логическом режиме (к знаку вопроса слева подключается связь логического типа), в таком случае её работа аналогична работе условного оператора. Также *CASE*-структура может быть использована в численном режиме (к знаку вопроса слева подключается связь целочисленного типа), и в таком случае структура работает как оператор переключения (*switch* на языке *Visual C#*).

## 4.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 4.7 Блок-схемы алгоритмов для типовых решений

**5 Задание «Разработка виртуального прибора "Полярные координаты"»**

\*Для выполнения всех работ рекомендуется скачать старую версию (6.1) пакета прикладных программ *National Instruments LabView* (). Обучающимся,

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «12».

## 5.1 Цель работы

## 5.2 Пример выполнения задания

## 5.3 Полезные соотношения для выполнения задания

## 5.4 Варианты индивидуального задания

## 5.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

## 5.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 5.7 Блок-схемы алгоритмов для типовых решений

**6 Задание «Разработка игрового виртуального прибора "Угадай число"»**

\*Для выполнения всех работ рекомендуется скачать старую версию (6.1) пакета прикладных программ *National Instruments LabView* (). Обучающимся,

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «13».

## 6.1 Цель работы

## 6.2 Пример выполнения задания

## 6.3 Полезные соотношения для выполнения задания

## 6.4 Варианты индивидуального задания

## 6.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

## 6.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 6.7 Блок-схемы алгоритмов для типовых решений

**7 Задание «Разработка виртуального прибора "Построение ломаной"»**

\*Для выполнения всех работ рекомендуется скачать старую версию (6.1) пакета прикладных программ *National Instruments LabView* (). Обучающимся,

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «14».

## 7.1 Цель работы

## 7.2 Пример выполнения задания

## 7.3 Полезные соотношения для выполнения задания

## 7.4 Варианты индивидуального задания

## 7.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

## 7.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 7.7 Блок-схемы алгоритмов для типовых решений

**8 Задание «Разработка виртуального прибора "Графическое решение системы линейных алгебраических уравнений"»**

\*Для выполнения всех работ рекомендуется скачать старую версию (6.1) пакета прикладных программ *National Instruments LabView* (). Обучающимся,

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «15».

## 8.1 Цель работы

## 8.2 Пример выполнения задания

## 8.3 Полезные соотношения для выполнения задания

## 8.4 Варианты индивидуального задания

## 8.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

## 8.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 8.7 Блок-схемы алгоритмов для типовых решений

**9 Задание «Разработка виртуального прибора "Построение окружности"»**

\*Для выполнения всех работ рекомендуется скачать старую версию (6.1) пакета прикладных программ *National Instruments LabView* (). Обучающимся,

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «16».

## 9.1 Цель работы

## 9.2 Пример выполнения задания

## 9.3 Полезные соотношения для выполнения задания

## 9.4 Варианты индивидуального задания

## 9.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

## 9.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 9.7 Блок-схемы алгоритмов для типовых решений

**10 Задание «Разработка виртуального прибора "Конвертор систем счисления"»**

\*Для выполнения всех работ рекомендуется скачать старую версию (6.1) пакета прикладных программ *National Instruments LabView* (). Обучающимся,

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «17».

## 10.1 Цель работы

## 10.2 Пример выполнения задания

## 10.3 Полезные соотношения для выполнения задания

## 10.4 Варианты индивидуального задания

## 10.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

## 10.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 10.7 Блок-схемы алгоритмов для типовых решений

**11 Задание «Разработка виртуального прибора-подпрограммы»**

\*Для выполнения всех работ рекомендуется скачать старую версию (6.1) пакета прикладных программ *National Instruments LabView* (). Обучающимся,

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «18».

## 11.1 Цель работы

## 11.2 Пример выполнения задания

## 11.3 Полезные соотношения для выполнения задания

## 11.4 Варианты индивидуального задания

## 11.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

## 11.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 11.7 Блок-схемы алгоритмов для типовых решений

**12 Задание «Разработка виртуального прибора "Калькулятор комплексных чисел"»**

\*Для выполнения всех работ рекомендуется скачать старую версию (6.1) пакета прикладных программ *National Instruments LabView* (). Обучающимся,

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «19».

## 12.1 Цель работы

## 12.2 Пример выполнения задания

## 12.3 Полезные соотношения для выполнения задания

## 12.4 Варианты индивидуального задания

## 12.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

## 12.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 12.7 Блок-схемы алгоритмов для типовых решений

**13 Задание «Разработка виртуального прибора "Матричные методы решения систем линейных алгебраических уравнений"»**

\*Для выполнения всех работ рекомендуется скачать старую версию (6.1) пакета прикладных программ *National Instruments LabView* (). Обучающимся,

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «20».

## 13.1 Цель работы

## 13.2 Пример выполнения задания

## 13.3 Полезные соотношения для выполнения задания

## 13.4 Варианты индивидуального задания

## 13.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

## 13.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 13.7 Блок-схемы алгоритмов для типовых решений

**14 Задание «Разработка виртуального прибора "Чтение / Сохранение файлов"»**

\*Для выполнения всех работ рекомендуется скачать старую версию (6.1) пакета прикладных программ *National Instruments LabView* (). Обучающимся,

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «21».

## 14.1 Цель работы

## 14.2 Пример выполнения задания

## 14.3 Полезные соотношения для выполнения задания

## 14.4 Варианты индивидуального задания

## 14.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

## 14.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 14.7 Блок-схемы алгоритмов для типовых решений

**15 Задание «Разработка виртуального прибора "Последовательности вычислений"»**

\*Для выполнения всех работ рекомендуется скачать старую версию (6.1) пакета прикладных программ *National Instruments LabView* (). Обучающимся,

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «22».

## 15.1 Цель работы

## 15.2 Пример выполнения задания

## 15.3 Полезные соотношения для выполнения задания

## 15.4 Варианты индивидуального задания

## 15.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

## 15.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 15.7 Блок-схемы алгоритмов для типовых решений

**16 Задание «Разработка виртуального прибора "Решение систем обыкновенных дифференциальных уравнений"»**

\*Для выполнения всех работ рекомендуется скачать старую версию (6.1) пакета прикладных программ *National Instruments LabView* (). Обучающимся,

Составить отчёт (в текстовом редакторе *Microsoft Office Word*) по выполненной работе с учётом рекомендаций, собранных в Приложении 2. Файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3. В программе «Учебной практики» заданию присвоен номер «23».

## 16.1 Цель работы

## 16.2 Пример выполнения задания

## 16.3 Полезные соотношения для выполнения задания

## 16.4 Варианты индивидуального задания

## 16.5 Пример компоновки графического пользовательского интерфейса виртуального прибора

## 16.6 Рекомендации по размещению информации на блок-диаграмме виртуального прибора

## 16.7 Блок-схемы алгоритмов для типовых решений

**Список литературы:**

1. Монахов, О. И. Методика организации Учебной, Производственной, Научно-исследовательской и Преддипломной практик обучающихся : учебно-методическое пособие для бакалавров направления 27.03.04 «Управление в технических системах» / О. И. Монахов, А. И. Сафронов, Л. Н. Логинова, Н. Н. Зольникова, А. Е. Ермакова. – М.: РУТ (МИИТ). – 2020. – 112 с.

2. Сафронов, А.И. Составление отчётной документации по решённым задачам алгоритмизации и программирования: Учебно-методическое пособие для проведения аудиторных занятий по Учебной практике / А. И. Сафронов, Н. Н. Зольникова, В. Г. Новиков. – М.: РУТ (МИИТ). – 2018. – 83 с.

3. Сафронов, А. И. Получение первичных профессиональных умений и навыков научно-исследовательской деятельности: Сборник задач для проведения аудиторных занятий по Учебной практике / А. И. Сафронов, Н. Н. Зольникова, В. Г. Новиков. – М.: РУТ (МИИТ). – 2019. – 91 с.

**Приложение 1. Типовая структура отчёта по работе, классифицируемой как задача программирования и алгоритмизации**

Отчёт по выполненной работе программирования и алгоритмизации должен содержать:

0. Титульный лист.

1. Формулировку цели работы.

2. Описание задачи согласно выданному варианту.

2.1. Общая часть.

2.2. Индивидуальная часть (если есть).

3. Составление блок-схемы алгоритма программы.

3.1. Общая часть.

3.2. Индивидуальная часть (если есть).

4. Подбор и расчёт тестовых примеров.

4.1. Общая часть.

4.2. Индивидуальная часть (если есть).

5. Листинг кода составленного программного обеспечения (иллюстрация блок-диаграммы *National Instruments LabView*).

5.1. Общая часть.

5.2. Индивидуальная часть (если есть).

6. Графический пользовательский интерфейс программного обеспечения (передняя панель виртуального прибора *National Instruments* *LabView*) и его описание.

6.1. Общая часть.

6.2. Индивидуальная часть (если есть).

7. Расчёт тестовых примеров с использованием составленного программного обеспечения.

7.1. Общая часть.

7.2. Индивидуальная часть (если есть).

8. Формулировку вывода о проделанной работе (обезличено – исключить из вывода местоимения, такие как «я», «мы» и другие).

**Приложение 2. Перечень типовых рекомендаций для демонстрации самостоятельности выполненной работы обучающимися**

Далее сформулирован краткий перечень типовых рекомендаций, уместных для оформления отчёта по каждой из решённых обучающимися задач, изложенных в сборнике:

1. Блок-схема алгоритма: размер для всех блоков по ширине подбирается исходя из ширины наиболее наполненного текстом блока.

2. Блок-схема алгоритма: для случаев, когда в блоке, согласно пункту 1, необходимо разместить довольно большое количество текста, который в итоге обусловит неоправданно громоздкую схему, рекомендуется часть текста оформить в виде комментария к блоку; размеры блоков-комментариев могут превышать принятые размеры основных блоков (комментарии не нормируются).

3. Блок-схема алгоритма: блоки по высоте не нормируются, однако рекомендуется выполнять увеличение их высоты только пропорционально количеству записанных в них действий. **Иначе**: для всей блок-схемы принимается некоторая средняя высота одного блока, достаточная для описания одного действия, и если блок содержит описание двух действий, то его высота может быть увеличена вдвое, трёх действий – втрое, и так далее.

4. Блок-схема алгоритма: в тех случаях, когда блок-схема, размещаемая на странице А4, масштабируется настолько, что текст в блоках становится нечитаемым, рекомендуется распределить блоки на нескольких страницах А4, используя ссылки для демонстрации переходов между страницами по линиям связи.

5. Листинг кода программы: код программы должен содержать значимые имена переменных или таблицу соответствия, раскрывающую смысловое содержание введённых автором переменных, не обладающих значащими именами.

6. Листинг кода программы: для удобства навигации по коду автора программы и проверяющего во время защиты работы, код необходимо снабжать комментариями. Комментарии, выполненные в коде, могут также играть роль связки с блок-схемой алгоритма и упрощать проверку отчёта.

7. Подбор тестовых примеров: минимальное количество тестовых примеров, доказывающих работоспособность составленного программного обеспечения, должно быть достаточным для покрытия всех ветвей разветвляющегося вычислительного процесса.

8. Общее: необходимо выполнение дополнительных скриншотов/иллюстраций для случаев, когда текстовое описание проделанных автором работы действий становится громоздким или трудным для подробного текстового описания.

9. Общее: рекомендуется выполнять нумерацию рисунков (если они есть) с подписями, содержащими наименование этих рисунков, например, «Рисунок 1 – Пользовательский интерфейс калькулятора *Microsoft Windows*».

**Приложение 3. Перечень требований, предъявляемых к именам отчётных файлов, направляемых на удалённую проверку (по электронной почте)**

**Общий вид формата имени файла:** «*Дата. Задание. Фамилия.docx*»

**Формат записи даты:** «*ГГГГММДД*», где *ГГГГ* – четыре цифры текущего года, *ММ* – две цифры текущего месяца, *ДД* – две цифры текущего дня.

**Формат записи задания:** «Задание *NNk*», где *NN* – две цифры номера задания, *k* – обозначение «о», если файл содержит общую часть; обозначение «и», если файл содержит индивидуальную часть; обозначение «ои», если файл содержит как общую, так и индивидуальную части.

**Если устранить замечания по работе удаётся в тот же день:** после фамилии ставится пробел и в круглых скобках записывается номер попытки исправления.

**Примеры правильных имён файлов, которые сдаются на проверку впервые:**

«*20200919. Задание 02о. Иванов.docx*»

«*20200926. Задание 03и. Иванов.xlsx*»

«*20201003. Задание 04ои. Иванов.vi*»

«*20201101. Задание 05ои. Иванов.vsdx*»

**Примеры правильных имён файлов, которые сдаются на проверку повторно в тот же день:**

«*20200919. Задание 02о. Иванов (1).docx*»

«*20200926. Задание 03и. Иванов (1).xlsx*»

«*20201003. Задание 04ои. Иванов (1).vi*»

«*20201101. Задание 05ои. Иванов (1).vsdx*»

**Оглавление**

Введение 3

1 Задание «Приобретение навыков работы с калькулятором *Microsoft Windows*» 6

2 Задание «Приобретение навыков работы с электронной таблицей *Microsoft Office Excel* как с калькулятором» 15

3 Задание «Приобретение навыков форматирования электронных таблиц *Microsoft Office Excel*» 19

4 Задание «Приобретение навыков работы со стандартными функциями электронной таблицы *Microsoft Office Excel*» 33

5 Задание «Логические задачи в *Microsoft Office Excel*» 41

6 Задание «Программирование функции для построения ломаной линии в *Microsoft Office Excel*» 54

7 Задание «Подбор тестовых примеров к программам с разветвлением вычислительного процесса в *Microsoft Office Excel*» 70

Список литературы: 85

Приложение 1. Типовая структура отчёта по работе, классифицируемой как расчётная задача 86

Приложение 2. Типовая структура отчёта по работе, классифицируемой как задача программирования и алгоритмизации 86

Приложение 3. Перечень типовых рекомендаций для демонстрации самостоятельности выполненной работы обучающимися 87

Приложение 4. Перечень требований, предъявляемых к именам отчётных файлов, направляемых на удалённую проверку (по электронной почте) 89

УЧЕБНО-МЕТОДИЧЕСКОЕ ИЗДАНИЕ

Сафронов Антон Игоревич

Получение первичных профессиональных умений и навыков научно-исследовательской деятельности

Сборник задач

для проведения аудиторных занятий по Учебной практике

Тираж 100 экз. Изд. №

1. [*https://yadi.sk/d/JRhS6ITDqHZqC*](https://yadi.sk/d/JRhS6ITDqHZqC) [↑](#footnote-ref-1)
2. [*https://drive.google.com/open?id=18EHK-h-3kCJtAuDELiijaligTm5r8ohQ*](https://drive.google.com/open?id=18EHK-h-3kCJtAuDELiijaligTm5r8ohQ) [↑](#footnote-ref-2)