МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Радиофизический факультет

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП
к.ф.-м.н., доцент
_____ В.А. Мещеряков
«23» января 2023 г.

ДИПЛОМНАЯ РАБОТА

РАЗРАБОТКА СИСТЕМЫ ДЛЯ ИЗУЧЕНИЯ КОМБИНАЦИОННЫХ ЦИФРОВЫХ УСТРОЙСТВ НА БАЗЕ LabVIEW и NI ELVIS III

по основной образовательной программе подготовки специалиста
по специальности 11.05.01 «Радиоэлектронные системы и комплексы»

Горовой Антон Андреевич

Руководитель ВКР
канд. физ. мат. наук, доцент
_____ С.Н.Торгаев
«___» января 2023 г.

Автор работы
студент группы № 779
_____ А.А. Горовой

Томск-2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Радиофизический факультет

УТВЕРЖДАЮ
Руководитель ООП
к.ф-м.н., доцент
_____ В.А. Мещеряков
«20» декабря 2022 г.

ЗАДАНИЕ
на выпускную квалификационную работу
студенту 07709 группы Горовому Антону Андреевичу

1. Тема ВКР: Разработка системы для изучения комбинационных устройств на базе LabView и NI ELVIS III

2. Срок сдачи студентом выполненной ВКР:
а) на кафедре 26.01.2023,
б) в ГЭК 03.02.2023.

3. Краткое содержание работы:

Работа посвящена разработке систем для изучения комбинационных цифровых устройств на базе LabVIEW и NI ELVIS III.

4. Календарный график учебной практики:

а) Анализ литературы и оформление литературного обзора	13.10.2022 – 01.10.2022
б) Обзор основных видов комбинационных микросхем	14.10.2022– 20.10.2022
в) Освоение комплексной работы LabVIEW и NI ELVIS III.	20.10.2022– 01.11.2022
г) Разработка программы в системе LabVIEW.	15.12.2022– 31.11.2022
д) Написание ВКР и сдача в ГЭК	12.01.2023– 31.01.2023
е) Размещение текста ВКР в электронной библиотеке ТГУ	26.01.2023 – 01.02.2023

5. Дата выдачи задания «20» декабря 2022 г.

Руководитель ВКР – кандитат физ.-мат. наук, доцент С.Н. Торгаев

Задание принял к исполнению А.А. Горовой

АННОТАЦИЯ

Отчет о дипломной работе 3 главы, 41 стр., 43 рис., источников 7.

КОМБИНАЦИООНАЯ ЛОГИКА, NI ELVIS III, LABVIEW, ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ, МУЛЬТИПЛЕКСОРЫ, ДЕМУЛЬТИПЛЕКСОРЫ, ШИФРАТОРЫ, ДЕШИФРАТОРЫ, ТАБЛИЦА ИСТИННОСТИ

Объект исследования – комбинационные цифровые устройства

Цель работы – разработка виртуальных приборов для работы с комбинационными цифровыми устройствами на базе NI ELVIS III.

В результате работы:

а) проведен обзор современных публикаций по теме работы;

б) изучены принципы совместной работы LabVIEW и NI ELVIS III;

в) проведен анализ комбинационных цифровых устройств;

г) разработаны виртуальные приборы в среде LabVIEW для работы аппаратных средствах NI ELVIS III с логическими элементами, мультиплексорами.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 4

1 Обзор программы LabView 5

1.1 История разработки LabVIEW 5

1.2 Программа LabVIEW 9

1.3 Принцип работы LabVIEW 9

1.4 NI ELVIS III 11

2 Комбинационная логика 16

2.1 Базовая логика. Таблицы истинности. Условные графические обозначения элементов. 16

2.2 Мультиплексоры и демультиплексоры 19

2.3 Шифраторы и дешифраторы 21

2.4 Сумматоры. Полусумматоры и полные сумматары. 25

3 Описание программ 28

3.1 Разработка программы для лабораторной работы по изучению логических элементов с использование базовой логики 28

3.2 Разработка программы для лабораторной работы по изучению логических функций с использованием базовой логики 31

3.3 Разработка программы для лабораторной работы по изучению мультиплексора 35

ЗАКЛЮЧЕНИЕ 39

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 40

ВВЕДЕНИЕ

Виртуальные приборы, являющиеся неотъемлемой частью продуктов National Instruments, таких как Labview и Elvis III, который позволяют усовершенствовать учебный процесс студентов. Они позволяют экспериментам быть гибкими и адаптируемыми, комбинировать операции управления, автоматизации и управления. Совокупная работа Labview и NI Elvis III расширяют возможности пользователя в проведении экспериментов, в частности лабораторных работ.

В образовательном процессе особенно важно учитывать его доступность, а также удобство для студента. С помощью LabVIEW и NI Elvis III можно реализовывать виртуальные приборы с различными сценариями выполнения, где для удобства наглядность процесса вынесена на лицевую панель.

Актуальность работы заключается в адаптации комплекса лабораторных работ под LabVIEW и NI Elvis III для более доступного освоения цифровых устройств, основой которых является комбинационная логика.

Целью работы является разработка виртуальных приборов на базе LabVIEW и NI ELVIS III для изучения комбинационных цифровых устройств.

Задачи, которые ставит перед собой данная работа:

- разработка виртуального прибора для работы с логическими элементами;
- разработка виртуального прибора для работы с логическими функция с использованием базовой логики;
- разработка виртуального прибора для работы с мультиплексорами

1 Обзор программы LabView

Компания National Instruments представлена на мировом рынке контрольно-измерительных приборов и средств автоматизации уже 30 лет. Сегодня у National Instruments 30 тысяч заказчиков в более чем 90 странах мира. Компания имеет представительства в 41 стране. Общемировой объем продаж в 2006 году составил 660 млн. долл. В разработки компания инвестирует 16% от своего оборота. По данным журнала FORTUNE, в течение последних восьми лет National Instruments входит в 100 лучших компаний США и занимает 14-е место в рейтинге Business Week "Компании, где лучше всего начинать карьеру". Технологии, предлагаемые National Instruments, позволяют создавать законченные измерительные и контрольно-управляющие системы. National Instruments является разработчиком технологии виртуальных приборов – революционной концепции, изменившей подходы и методику проведения измерений и разработки систем автоматизации. В ее основе лежит LabVIEW – открытая среда графического программирования приложений. Благодаря ей для построения высокопроизводительных распределенных систем измерения, сбора данных и управления можно использовать обычные персональные компьютеры. Используя гибкую среду программирования и стандартные интерфейсы, LabVIEW позволяет очень быстро вывести предприятие на новый технологический уровень.

1.1 История разработки LabVIEW

В 1983 году компания National Instruments начала поиски способов сокращения времени, необходимого для программирования измерительных систем. В результате появилась концепция виртуального прибора LabVIEW - сочетания интуитивного пользовательского интерфейса лицевой панели с передовой методикой блок-диаграммного программирования, позволяющего создавать эффективные измерительные системы на основе графического программного обеспечения. Первая версия LabVIEW увидела свет в 1986 году. Она была предназначена только для компьютеров Macintosh. Несмотря на то, что Macintosh довольно редко использовались в задачах измерений и автоматизации, графическая оболочка их операционной системы MacOS наилучшим образом соответствовала технологии LabVIEW. Довольно скоро и другие наиболее распространенные

операционные системы перешли на графический пользовательский интерфейс и начали поддерживать эту технологию. К 1990 году разработчики National Instruments полностью переделали LabVIEW, сочетая новые компьютерные технологии с анализом отзывов пользователей. И, что более важно, вторая версия LabVIEW включала компилятор, делающий скорость исполнения виртуальных приборов сравнимой со скоростью выполнения программ, созданных на языке программирования С. Патентное ведомство США (United States Patent Office) выдало National Instruments несколько патентов, признающих новизну технологии LabVIEW. С появлением новых графических операционных систем, подобных MacOS, компания National Instruments перенесла ставшую уже признанной технологию LabVIEW на другие платформы - персональные компьютеры и рабочие станции. В 1992 году благодаря новой открытой архитектуре появились версии LabVIEW для Windows и Sun.

Третья версия LabVIEW появилась в 1993 году сразу для трех операционных систем: Macintosh, Windows и Sun. Виртуальные приборы версии 3, разработанные на одной из платформ, могли без изменений запускаться на другой. Эта межплатформенная совместимость дала пользователям возможность выбора платформы разработки и уверенность, что созданные ВП будут функционировать и на других платформах (обратите внимание, что это было реализовано за пару лет до появления Java). В 1994 году список платформ, поддерживающих LabVIEW, увеличился и стал включать Windows NT, Power Macs, рабочие станции Hewlett Packard и в 1995 году - Windows 95.

LabVIEW 4 была выпущена в 1996 году и обеспечивала большую гибкость оболочки среды разработки, которая позволила пользователям создавать программы, подходящие типу их деятельности, уровню опыта и навыкам разработки. Кроме этого, LabVIEW 4 включала в себя мощные инструменты редактирования и отладки более совершенных измерительных систем, в том числе обмен данными на основе OLE-технологии и распределенных средств исполнения. Версии LabVIEW 5 и 5.1 (в 1999 году) продолжают наращивать возможности системы: появляется встроенный Internet-сервер, подсистема динамического программирования и управления (сервер виртуального прибора), интеграция с ActiveX и особый протокол для упрощения обмена данными через Internet -

DataSocket. Также введена долгожданная возможность отката действий пользователя (undo), которая уже присутствовала в большинстве компьютерных программ. Вышедшей в 2000 году новой версии - LabVIEW 6 (известной также как 6i) - сделали «подтяжку лица»: в нее встроили новый комплект объемных элементов управления и индикации, поскольку в то время компьютерная индустрия обнаружила, что внешний вид программного продукта имеет весьма серьезное значение (чему способствовало появление систем Apple iMac и G4). В LabVIEW 6 воплотилась очень серьезная работа по обеспечению как простого и интуитивного интерфейса среды программирования (особенно для непрограммистов), так и поддержки множества передовых технологий программирования, например объектно-ориентированного программирования, многопоточности (multithreading), распределенных вычислений (distributed computing) и т.д. И пусть простота графической оболочки LabVIEW не вводит вас в заблуждение: LabVIEW - это инструмент, который является достойным соперником систем программирования C++ или Visual Basic, да еще и превосходит их в удобстве работы, как отмечают тысячи пользователей. В версии LabVIEW 6.1, вышедшей в 2001 году, было введено событийно-управляемое (event-oriented) программирование, удаленное управление LabVIEW через Internet и другие улучшения. Совершенно особой разновидностью LabVIEW, на которую следует обратить внимание, является LabVIEW RT. RT означает Real Time - реальное время. LabVIEW RT представляет собой совокупность аппаратного и программного обеспечения, которая позволяет выделять части кода LabVIEW и загружать их для выполнения на отдельном контроллере, работающем под управлением собственной операционной системы реального времени. Таким образом гарантируется, что выделенные участки LabVIEW-приложения будут выполняться в точно определенные моменты времени, даже если Windows «зависнет» и компьютер перестанет работать. LabVIEW является мощным инструментом программирования, пригодным для решения практически любых задач, например компьютерного моделирования, тем не менее он чаще всего используется для сбора экспериментальных данных и управления приборами и установками, и поэтому содержит множество виртуальных приборов, разработанных специально для этой цели. Например, LabVIEW может управлять встраиваемыми многофункциональными устройствами сбора данных (plug-in data

acquisition - DAQ), которые предназначены для ввода и/или вывода аналоговых и цифровых сигналов. Например, вы можете совместно использовать многофункциональные платы и LabVIEW для мониторинга температуры, формирования управляющих сигналов для экспериментальной установки или определения частоты неизвестного сигнала. LabVIEW также обеспечивает передачу команд и данных по каналу общего пользования (КОП) или через стандартный последовательный порт компьютера. Канал общего пользования часто применяется для взаимодействия с осциллографами, сканерами и мультиметрами, а также для дистанционного управления подобными приборами. С помощью программного обеспечения LabVIEW допустимо управлять сложными измерительными системами стандарта VXI, приборами с сетевым интерфейсом Ethernet или через порт USB. Получив со встроенной платы или внешнего прибора массив данных, вы можете использовать множество содержащихся в LabVIEW виртуальных приборов анализа для всесторонней обработки этих данных и их преобразования. Часто полезен обмен данными не только с измерительными приборами, но и с другими программными продуктами и удаленными компьютерами. В LabVIEW встроены функции, которые упрощают этот процесс, поддерживая несколько сетевых протоколов, вызов внешнего программного кода или динамических библиотек (DLL) и автоматизацию ActiveX. Оставшаяся часть этой главы посвящена обсуждению типичных задач, для решения которых и был создан LabVIEW.

Рисунок 1 – Штаб-квартира National Instrumets

1.2 Программа LabVIEW

Это принципиально иной язык программирования, или если хотите целая «философия» программирования. Функциональный язык, заставляющий несколько иначе мыслить и порой предоставляющий совершенно фантастические возможности для разработчика. Является ли LabVIEW языком программирования вообще? Это спорный вопрос — здесь нет стандарта, как, например, ANSI C. В узких кругах разработчиков говорят, что пишем на языке «G». Формально такого языка не существует, но в этом и заключается прелесть этого средства разработки: от версии к версии в язык вводятся всё новые конструкции. Сложно представить, что в следующей реинкарнации C появится, например, новая структура для for-цикла. А в LabVIEW такое вполне возможно.

Впрочем, надо заметить, что LabVIEW входит в рейтинг языков программирования TIOBE, занимая на данный момент тридцатое место — где-то между Прологом и Фортраном.

1.3 Принцип работы LabVIEW

Программы LabVIEW называются виртуальными приборами (ВП, virtual instruments - VI), так как они функционально и внешне подобны реальным (традиционным) приборам. Однако они столь же подобны программам и функциям на популярных языках программирования, таких как C или Basic. Здесь и далее мы будем называть программы LabVIEW виртуальными приборами или ВП, причем вне зависимости от того, соотносится их вид и поведение с реальными приборами или нет. Виртуальный прибор состоит из трех основных частей.

Лицевая панель (Front Panel) представляет собой интерактивный пользовательский интерфейс виртуального прибора и названа так потому, что имитирует лицевую панель традиционного прибора. На ней могут находиться ручки управления, кнопки, графические индикаторы и другие элементы управления (controls), которые являются средствами ввода данных со стороны пользователя, и элементы индикации (indicators) - выходные данные из программы. Пользователь вводит данные, используя мышь и клавиатуру, а затем видит результаты действия программы на экране монитора;

Блок-диаграмма (Block Diagram) является исходным программным кодом ВП, созданным на языке графического программирования LabVIEW, G (Джей). Блок-диаграмма представляет собой реально исполняемое приложение. Компонентами блок-диаграммы являются: виртуальные приборы более низкого уровня, встроенные функции LabVIEW, константы и структуры управления выполнением программы. Для того чтобы задать поток данных между определенными объектами или, что то же самое, создать связь между ними, вы должны нарисовать соответствующие проводники (wires). Объекты на лицевой панели представлены на блок-диаграмме в виде соответствующих терминалов (terminals), через которые данные могут поступать от пользователя в программу и обратно;

Для того чтобы использовать некоторый ВП в качестве подпрограммы (подприбора) в блок-диаграмме другого ВП, необходимо определить его иконку (icon) и соединительную панель (connector). Виртуальный прибор, который применяется внутри другого ВП, называется виртуальным подприбором (ВПП, SubVI), который аналогичен подпрограмме в традиционных алгоритмических языках. Иконка является однозначным графическим представлением ВП и может использоваться в качестве объекта на блок-диаграмме другого ВП. Соединительная панель представляет собой механизм передачи данных в ВП из другой блок-диаграммы, когда он применяется в качестве подприбора - ВПП. Подобно аргументам и параметрам подпрограммы, соединительная панель определяет входные и выходные данные виртуального прибора. Виртуальные приборы являются иерархическими и модульными (modular). Вы можете использовать их как самостоятельные приложения (top-level programs), так и в качестве виртуальных подприборов. Согласно этой логике, LabVIEW следует концепции модульного программирования (modular programming). Вначале вы разделяете большую прикладную задачу на ряд простых подзадач. Далее создаете виртуальные приборы для выполнения каждой из подзадач, а затем объединяете эти ВП на блок-диаграмме прибора более высокого уровня, который выполняет прикладную задачу в целом. Технология модульного программирования очень хороша, потому что вы можете работать с каждым ВПП по отдельности, что облегчает отладку приложения. Более того, ВПП низкого уровня часто выполняют задачи, типичные для нескольких

приложений, и поэтому могут использоваться независимо во многих отдельных приложениях.

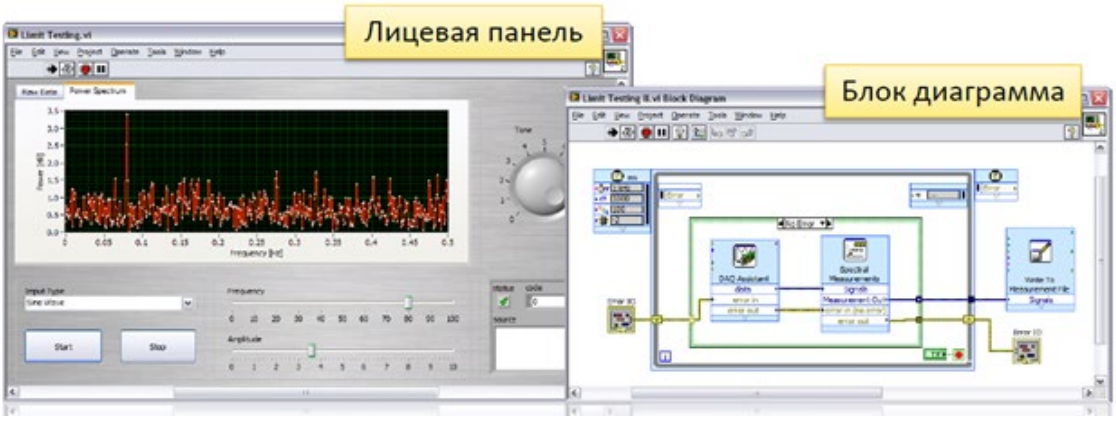


Рисунок 2 – Лицевая панель и блок-диаграмма ВП

1.4 NI ELVIS III

Комплект виртуальных измерительных приборов для учебной лаборатории NI Educational Laboratory Virtual Instrumentation Suite III (NI ELVIS™ III) представляет собой инженерное решение для проектного обучения.

Он объединяет в себе типовые лабораторные измерительные приборы, гибкий аналоговый и цифровой ввод и вывод, а также высокопроизводительный встраиваемый контроллер. Открытая программная архитектура поддерживает широкий спектр экспериментальных исследований, позволяя студентам быстро изучать основные принципы с помощью готовых приложений и базового программирования API, или переходить к низкоуровневому программированию встраиваемого процессора и FPGA. Съемная макетная плата может быть заменена широким набором прикладных плат для конкретной прикладной области, вместе с ресурсами обучающих курсов, охватывающими такие дисциплины, как электротехника, машиностроение и др

NI ELVIS III обеспечивает функциональность и удобное непосредственное подключение к прикладной плате с помощью разъемов с высокой плотностью контактов для управления вводом и выводом, стандартных промышленных разъемов для входов и выходов измерительных приборов и стандартных коммуникационных портов для управления NI ELVIS.

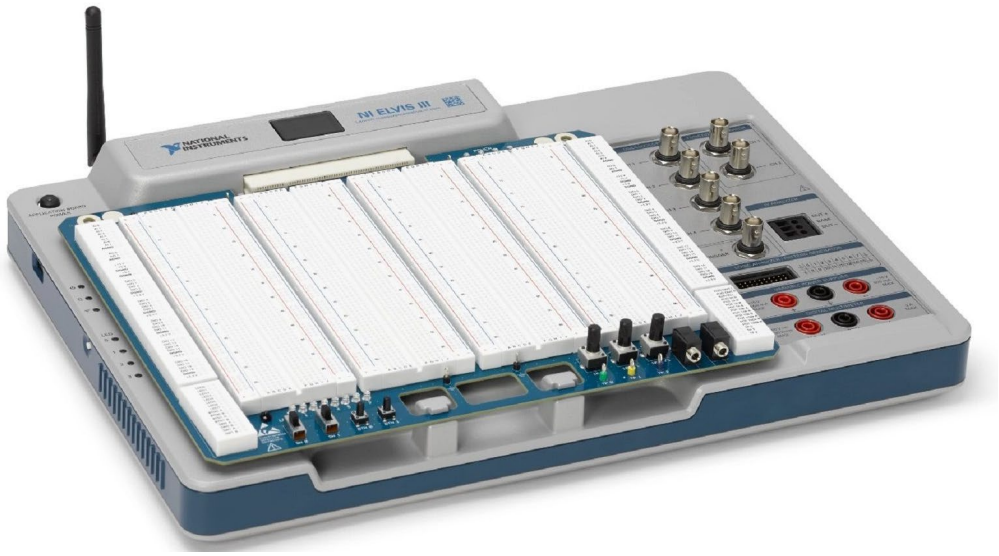


Рисунок 3 – NI ELVIS III

1.5 Основные возможности и архитектура аппаратных средств NI ELVIS III

Рабочая станция для инженерных лабораторий, управляемая через USB, Ethernet или WiFi;

Полный набор лабораторных измерительных приборов:

- осциллограф;
- генератор сигналов стандартной и произвольной формы;
- цифровой мультиметр;
- регулируемые источники питания;
- логический анализатор и генератор цифровых последовательностей;
- анализатор вольтамперных характеристик;
- анализатор амплитудно-частотных характеристик (анализатор Боде).

Гибкий ввод-вывод для управления и измерения:

- аналоговый ввод;
- аналоговый вывод;
- цифровой ввод/вывод;
- нерегулируемые источники питания.

Высокопроизводительный RIO контроллер:

- поддержка программирования в LabVIEW и в LabVIEW FPGA;
- модульные прикладные платы;
- макетная плата, по умолчанию;

- множество прикладные плат для различных областей применения.

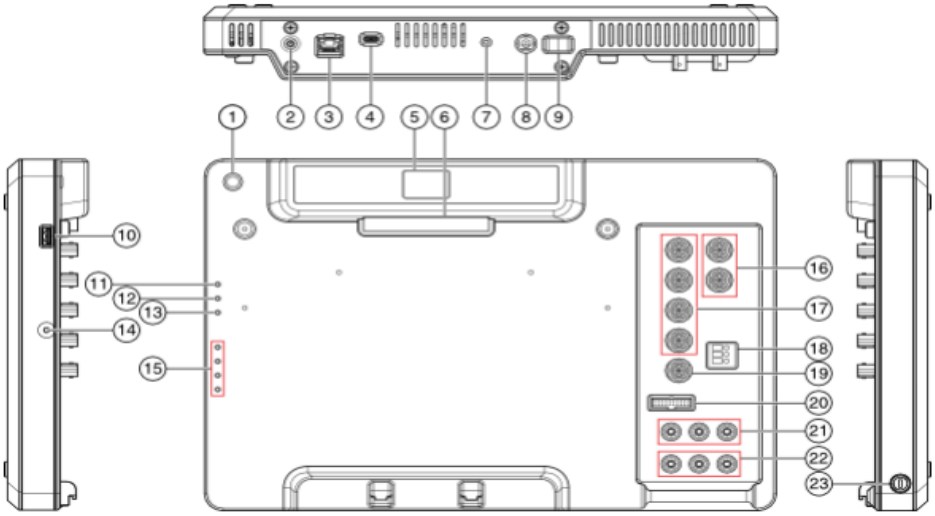


Рисунок 4 – Рабочая станция NI ELVIS III

1. кнопка/светодиод питания макетной платы;
2. разъем для антенны WiFi;
3. порт Ethernet;
4. порт устройства USB;
5. светодиодный дисплей;
6. разъем для прикладной платы;
7. кнопка RESET (сброс);
8. разъем для источника питания;
9. выключатель питания рабочей станции;
10. хост-порт USB;
11. светодиод питания рабочей станции;
12. светодиод Status (состояние);
13. светодиод WiFi;
14. кнопка, программируемая пользователем (Button 0);
15. светодиод, программируемый пользователем;
16. разъемы BNC генератора сигналов;
17. разъемы BNC осциллографа;
18. винтовые клеммы анализатора вольтамперных характеристик;
19. разъем BNC сигнала запуска;

20. 20-контактный разъем логического анализатора/генератора цифровых последовательностей;

21. гнезда типа "банан" регулируемого источника питания;

22. гнезда типа "банан" цифрового мультиметра;

23. предохранитель цифрового мультиметра;

NI ELVIS III предлагает множество вариантов программного обеспечения для управления оборудованием. Доступ к приборам может осуществляться интерактивно с помощью программных лицевых панелей (SFP), которые запускаются из Measurements Live в вашем браузере. Доступ к каналам ввода-вывода измерительных приборов и к каналам ввода-вывода управления NI ELVIS III может осуществляться программно из приложения LabVIEW RT, выполняющегося на встраиваемом процессоре. Доступ низкого уровня к каналам ввода-вывода управления NI ELVIS III можно также получить программно из приложения LabVIEW FPGA, выполняющегося во встроенной FPGA.

По умолчанию к устройству подключена макетная плата NI ELVIS III, которая обеспечивает простое подключение к доступным входам/выходам через разъемы платы, а также дополнительную пользовательскую периферию, часто используемую в лабораторных работах

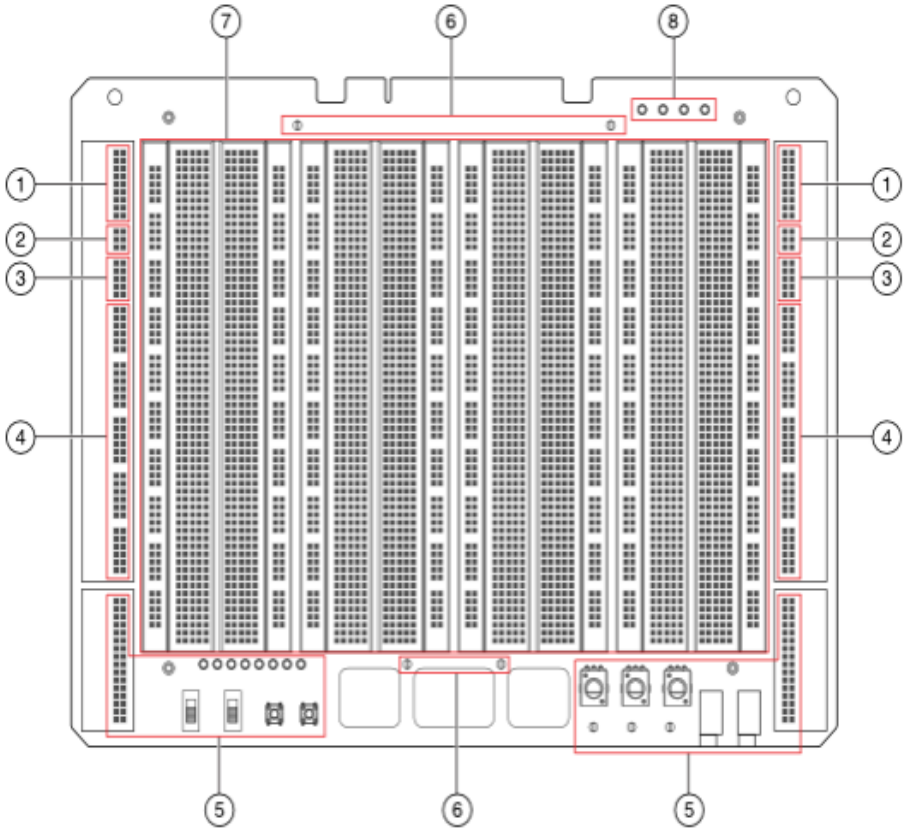


Рисунок 5 – Макетная плата NI ELVIS III

1. Аналоговый ввод
2. Аналоговый вывод
3. Нерегулируемые источники питания
4. Цифровой ввод-вывод
5. Пользовательская периферия
6. Общая точка для цифровых сигналов (цифровая земля)
7. Центральная область для сборки схем (область макетирования)
8. Светодиоды нерегулируемых источников питания

2 Комбинационная логика

Цифровые схемы разделяются на комбинационные и последовательные. Главной особенностью комбинационных схем является то, что сигнал на выходе комбинационных схем зависит только от комбинации сигналов на входах схемы и не зависит ни от времени, ни от предыдущих входных сигналов. Таким образом, при многократном поступлении одного и того же входного сигнала на выходах комбинационной схемы будет формироваться один и тот же сигнал.

2.1 Базовая логика. Таблицы истинности. Условные графические обозначения элементов.

Абсолютно все цифровые микросхемы состоят из одних и тех же логических элементов – «кирпичиков» любого цифрового узла.

Логический элемент – схема, у которой несколько входов и один выход. Каждому из сигналов на входах соответствует определенный сигнал на выходе. Итак, какие же бывают элементы?

Логическая функция «НЕ» (инверсия) выполняет операцию логического отрицания над значением логической переменной. Если значение переменной ложно (эквивалентно нулю), то значением функции будет истина (эквивалентно единицы) и, наоборот, если значение логической переменной ноль, то значение логической функции «НЕ» будет единица.

Логический элемент «НЕ», осуществляющий эту функцию в цифровой электронике с помощью электронной схемы, называется инвертором. Он является простейшим элементом с одним входом и одним выходом.

Элемент «НЕ» изменяет значение напряжения входного логического сигнала так, что на входе элемента появляется напряжение с логически противоположенным значением, то есть инвертирует логическое значение входной переменной. Его условное изображение предоставлено на рисунке 6. Таблица истинности для логического элемента «НЕ» предоставлена в таблице 1.

Таблица 1 – Таблица истинности элемента «НЕ»

X	Y
0	1
1	0

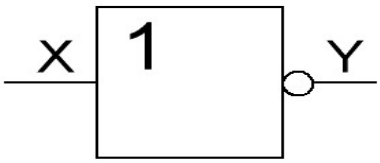


Рисунок 6 – Условное графическое отображение элемента «НЕ»

Логическая функция И (конъюкция) выполняет операцию логичского умножения переменных. Значение этой функции эквивалентно еденице тогда и только тогда, когда все логические переменные эквивалентны еденице. Значение функции И равно нулю,если хотя бы одна из логических переменных равно нулю.

Логический элемент «И», электрически выполняющий операцию логического умножения, иногда называют схемой вопадения. Напряжение, соответсвующее логичской еденицы появляется на его выходе только при совпадении значений напряжений, соответсвующих еденице всех входных логических переменных. Графическое обозначение представлено на рисунке 7. Таблица истинности предоставлена в таблице 2.

Таблица 2 – Таблица истинности элемента «И»

X1	X2	Y
0	0	0
1	0	0
0	1	0
1	1	1

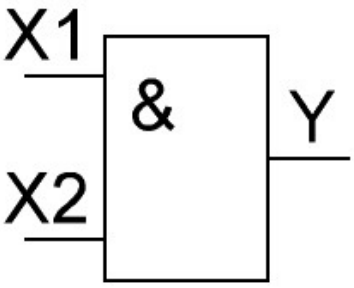


Рисунок 7 – Графическое отображение элемента «И»

Логическая функция «ИЛИ» (дизъюнкция) выполняет логическое сложение переменных. Значение этой функции эквивалентно нулю тогда и только тогда, когда все логические переменные эквивалентны нулю. Значение функции «ИЛИ» равно еденице если хотя бы одна из логических переменных равна единице. Условное графическое отображение показано на рисунке 8. Таблица истинности предоставлена в таблице 3

Таблица 3 – Таблица истинности «ИЛИ»

X1	X2	Y
0	0	0
1	0	1
0	1	1
1	1	1

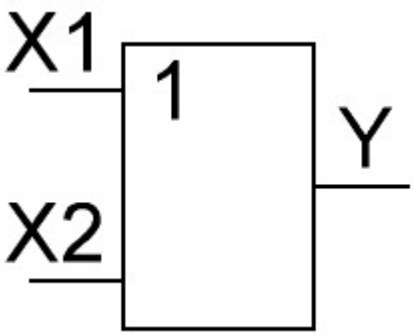


Рисунок 9 – Условное обозначение «ИЛИ»

Сложные логические операции, осуществляемые микросхемами цифровой техники, строятся обычно на основе функциональных базисов, содержащих одну операцию. Функция «ИЛИ-НЕ» сначала выполянет функцию логического сложения любого числа логических переменных, а затем инверсию результата сложения. Графичесое обозначения и таблица истинности предоставлены на рисунке 10 и таблице 4.

Таблица 4 – Таблица истинности «ИЛИ-НЕ»

X1	X2	Y
0	0	1
1	0	0
0	1	0
1	1	0

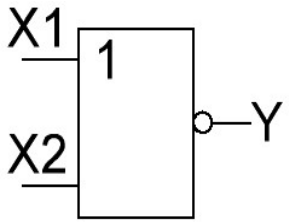


Рисунок 11 – Условное обозначение «ИЛИ-НЕ»

Функция «И-НЕ» последовательно выполняет сначала функцию логического умножения переменных, а затем инверсию результата умножения для любого числа логических переменных. Условное графическое обозначение и таблица истинности функции «И-НЕ» предосталвены на рисунке 12 и таблице 5.

Таблица 5 – Таблица истинности «И-НЕ»

X1	X2	Y
0	0	1
1	0	1
0	1	1
1	1	0

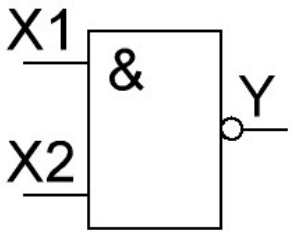


Рисунок 12 – Условное графическое обозначение «И-НЕ»

2.2 Мультиплексоры и демультиплексоры

Мультиплексор комбинационное цифровое устройство, которое обеспечивает передачу на единственный выход F одного из нескольких входных сигналов D_j в соответствии с поступающим адресным кодом A_i. При наличии n адресных входов можно реализовать $M=2^n$ комбинаций адресных сигналов, каждая из которых обеспечивает выбор одного из M входов. Чаще всего используются мультиплексоры «из 4 в 1» (n=2, M=4), «из 8 в 1» (n=3, M=8), «из 16 в 1» (n=4, M=16). Правило работы мультиплексора «из 4 в 1» обозначено на таблице 6

Таблица 6 – Таблица истинности для мультиплексора «из 4 в 1»

Входы		Выход
A_1	A_0	F
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Для реализации мультиплексора можно использовать логические элементы «И», «ИЛИ», «НЕ». Синтезированная структурная схема мультиплексора показана на рисунке 13

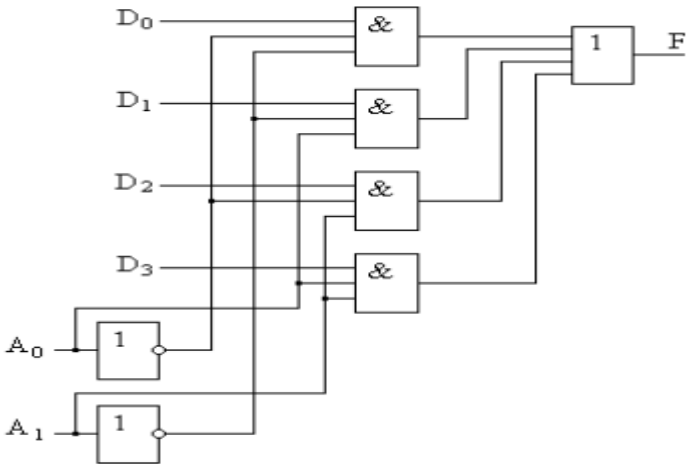


Рисунок 13 – Структура мультиплексора «из 4 в 1»

Демультиплексор выполняет функцию, обратную мультиплексору, то есть в соответствии с принятой адресацией A_i направляет информацию с единственного входа D на один из M выходов F_j. При этом на остальных выходах будут логические нули (единицы). Принцип работы демультиплексора «из 1 в 4» иллюстрируется таблицей 7.

Таблица 7 – Таблица истинности демультиплексора «из 1 в 4»

Входы		Выходы			
A_1	A_0	F_3	F_2	F_1	F_0
0	0	0	0	0	D
0	1	0	0	D	0
1	0	0	D	0	0
1	1	D	0	0	0

Структурная схема, реализующая демультиплексор «из 1 в 4» приведена на рисунке 14.

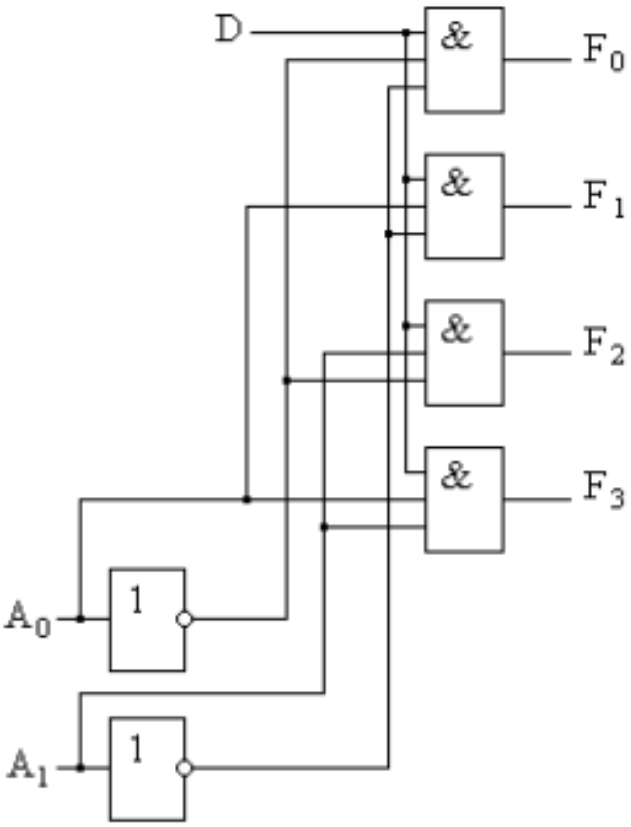


Рисунок 14 – Структура демультиплексора «из 1 в 4»

2.3 Шифраторы и дешифраторы

Шифраторы и дешифраторы используются в качестве кодирующих и декодирующих устройств (рисунок 15).

Рисунок 15 – Использование шифратора и дешифратора

Эти микросхемы относятся к комбинационным, то есть представляют собой комбинацию из простых логических микросхем, но в отличие от последовательных они не обладают памятью и их выходные сигналы никак не зависят от комбинации предыдущих входных сигналов.

Шифратор или кодер, это электронное устройство (микросхема), которая преобразует код одной системы счисления в код другой системы. Наибольшее распространение в электронике получили шифраторы, преобразующие позиционный десятичный код, в параллельный двоичный.

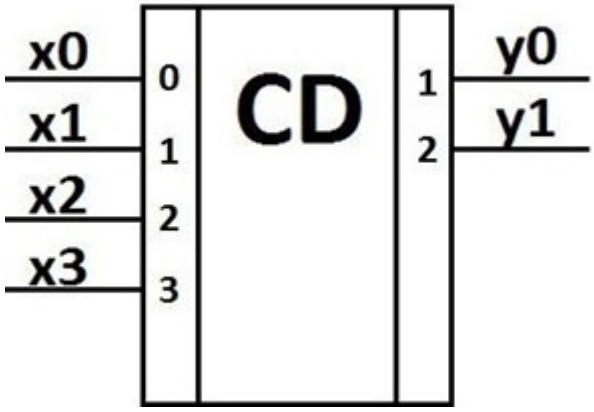


Рисунок 16 – Условно графическое обозначение шифратора

Микросхемы шифраторов обозначаются на схемах буквами «CD» (от англ. Coder – кодирующее устройство). Особенностью данной микросхемы является то, что в каждый момент времени, напряжение может быть только на одном входе.

Шифраторы можно разделить на полные и неполные. Полный двоичный шифратор имеет 2^n входов и n выходов. В неполном шифраторе, при n выходах – число входов будет меньше чем 2^n.

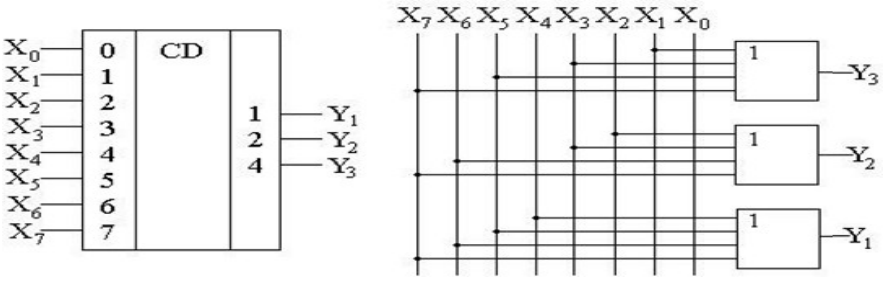


Рисунок 17 – Шифратор 8х3 построенный на логический элементах «ИЛИ»

Таблица 8 – Таблица истинности для шифратора 8х3

Вход								Выход		
X0	X1	X2	X3	X4	X5	X6	X7	Y1	Y2	Y3
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Дешифраторы преобразуют параллельный двоичный код в позиционный десятичный. Особенностью является то, что в любой момент времени только на одном выходе будет напряжение или единица.

Микросхемы дешифраторов обозначаются на схемах буквами «DC» (от англ. Decoder – декодирующее устройство).

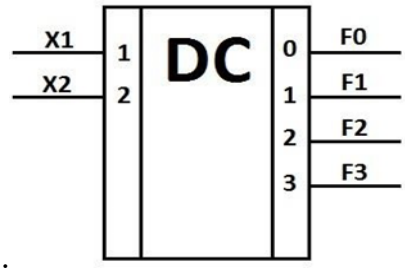


Рисунок 18 – Условно графическое обозначения дешифратора

Количество выходов полного дешифратора равно 2 n, где n-число входов. Если количество выходов меньше – значит дешифратор неполный.

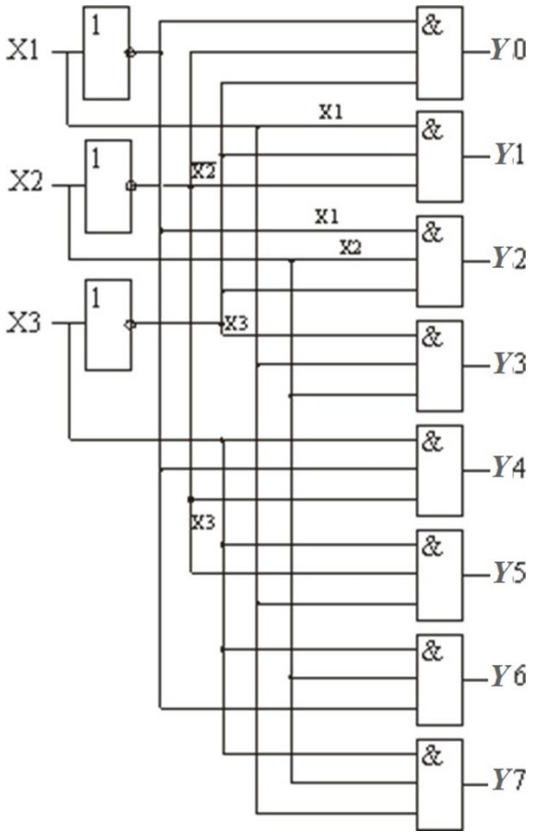


Рисунок 19 – Схема дешифратора на собранная на логических элементах «И» и «НЕ»

Ниже приведены условное графическое обозначение для дешифратора типа 3x8 и таблица истинности для данного дешифратора.

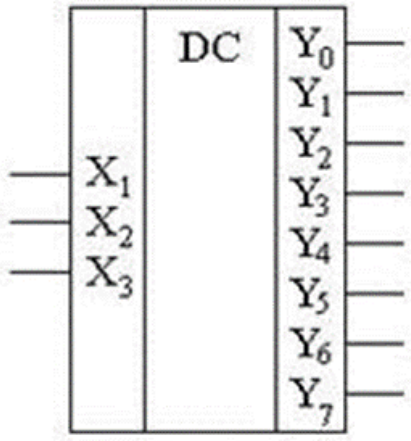


Рисунок 20 – Условно графическое обозначение дешифратора типа 3x8

Таблица 9 – Таблица истинности для дешифратора типа 3x8

Вход			Выход							
X3	X2	X1	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

2.4 Сумматоры. Полусумматоры и полные сумматары.

Наиболее распространенной операцией в устройствах обработки цифровой информации является суммирование, т. е. арифметическое сложение двух чисел, имеющих одинаковое число разрядов. Операция сложения выполняется в соответствии с правилами арифметики, используемой в данном конкретном случае системы счисления.

Сумматором называют функциональный узел, выполняющий сложение одно- или многоразрядных двоичных чисел в соответствии с правилами двоичного сложения.

Арифметические сумматоры выполняют операцию сложения слов. В соответствии с правилами двоичной арифметики сложение двух n – разрядных чисел

$A = a_0a_1...a_{n-1}$ и $B = b_0b_1...b_{n-1}$ сводится к вычислению поразрядной суммы $S = s_0s_1...s_{n-1}$ с учётом переносов.

На рисунке 21 приведена классификация арифметических сумматоров.

Рисунок 21 – Классификация арифметический сумматоров

Одноразрядные сумматоры выполняют сложение слагаемых одного разряда с учётом переноса из младшего разряда.

Многоразрядные сумматоры выполняют сложение n – разрядных слагаемых A и B с учётом переноса.

Многоразрядные сумматоры делятся на последовательные, в которых обработка данных ведётся поочерёдно разряд за разрядом на одном и том же оборудовании, и параллельные, в которых слагаемые обрабатываются одновременно по всем разрядам и для каждого разряда имеется своё оборудование.

По способу организации межразрядных переносов параллельные сумматоры подразделяются на схемы с последовательным переносом, с параллельным переносом и с групповой структурой. В последних разрядная сетка поделена на поля, обрабатываемые группами разрядных схем. В группах и между ними могут применяться разные способы переносов, причём в наименованиях сумматоров вначале указывается вид переноса внутри группы.

В зависимости от способа построения суммирующих схем различают комбинационные и накапливающие сумматоры. Комбинационные сумматоры не содержат запоминающих элементов и реализуют операцию сложения в виде S:= A + B. Накапливающие сумматоры содержат регистр, в котором аккумулируют результат суммирования. Накапливающие сумматоры реализуют операцию сложения в виде S = S + A.

По способу тактирования различают синхронные и асинхронные сумматоры. Синхронные сумматоры имеют постоянное время, отводимое для суммирования, независимо от значения слагаемых. В асинхронных сумматорах вырабатывается признак завершения операции. При этом среднее время суммирования уменьшается.

В зависимости от системы счисления различают двоичные, двоично-десятичные и другие сумматоры.

Полусумматорами называют устройство с двумя входами и двумя выходами, на которых вырабатываются сигналы суммы и переноса согласно формулам: $s_i = a_i$ v и b_i, $p_i = a_i * b_i$, где a_i, b_i – разряды слагаемых А и В; s_i – сумма по данному разряду; p_i – перенос по данному разряду.

Полусумматор реализует лишь часть задачи суммирования, так как не учитывает входной величины – переноса из младшего разряда.

Одноразрядные сумматоры имеют три входа и два выхода и обеспечивают сложение разрядов слагаемых a_i и b_i с переносом из предыдущего разряда p_{i-1}.

Таблица истинности сумматора, реализующего сложение трёх входных величин и условное графическое обозначение приведена в табл. 10. и рисунке 22.

Таблица 10 – Таблица истинности полного сумматора

Входы			Выходы	
a_i	b_i	pi-1	s_i	p_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

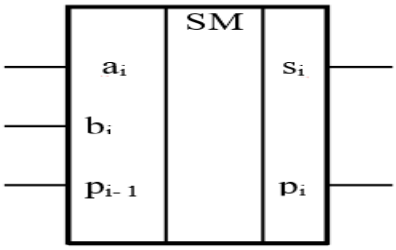


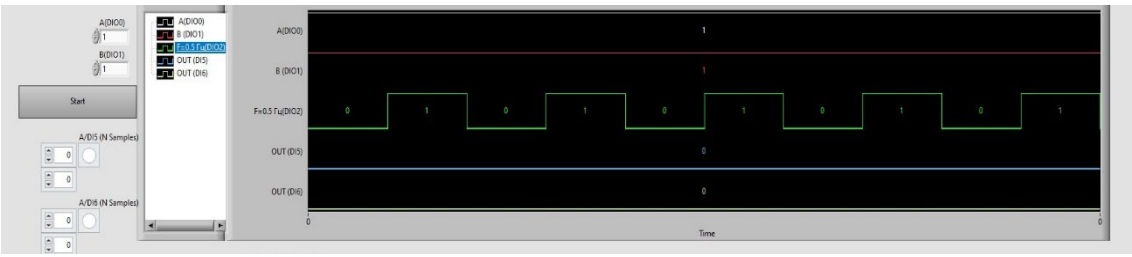
Рисунок 22 – Условное графическое обозначение полного сумматора

3 Описание программ

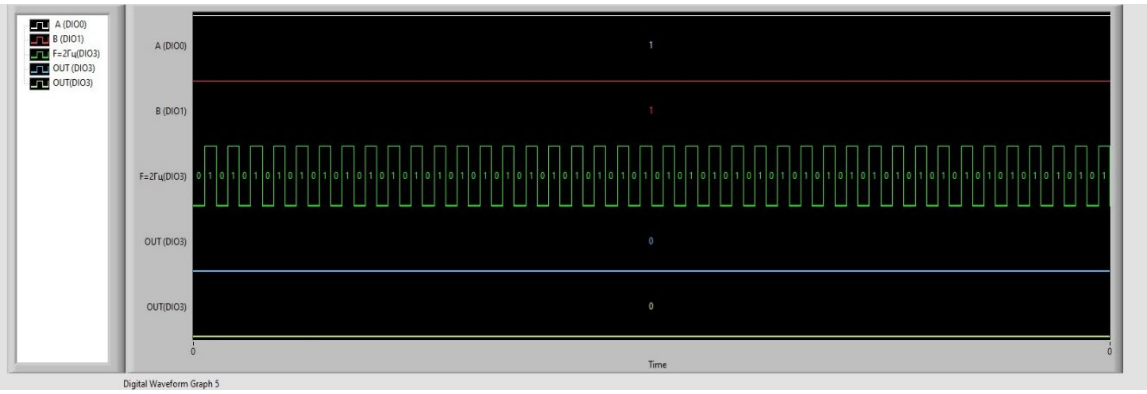
3.1 Разработка программы для лабораторной работы по изучению логических элементов с использование базовой логики

Для каждой из лабораторных работ была сделана индивидуальная программа в LabVIEW.

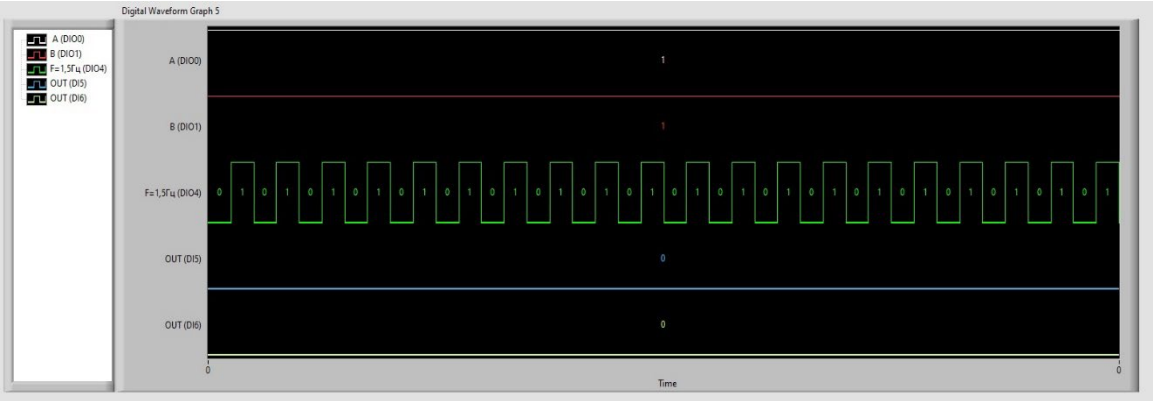
Рассмотрим программу для работы с микросхемами 74HC04, 74HC08, 74HC32, 74HC32, 74HC00, 74HC02.



а)



б)



в)

Рисунок 23 – Лицевая панель ВП
а – При частоте 0,5Гц;
б – При частоте 2Гц;
в – При частоте 1,5гц.

На лицевую панель вынесены три графика для рассмотрения трех случаев (рисунок 23). В каждой из них на входы микросхем пользователь может выбрать, что подать, ноль или единицу, а на третий вход поедается индивидуальный сигнал разной частоты. Так же подключены два выхода с микросхем, для отображения результата. Для удобства каждое отображение на графике подписано и имеет свой индивидуальный цвет. Работа программы начинается при нажатии кнопки «Start».

Рассмотрим программу на блок-диаграмме.

В первую очередь мною был добавлен генератор Digital Pattern Generator (рисунок 24), с его помощью подается импульс на логическую микросхему. Данный генератор позволяет нам выбирать режимы импульсов: Single Value, Ramp, Toggle, Random, Marching Value. Так же мы можем задать количество сигналов, подаваемые с нашего генератора.

Для реализации этой лабораторной работы было подключено пять генераторов, работающих в разных режимах.

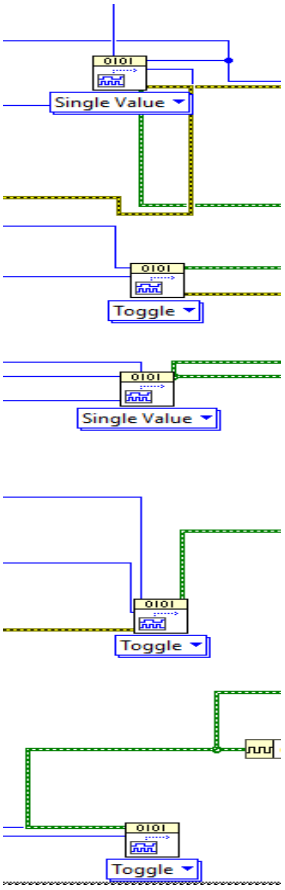


Рисунок 24 – Используемые Digital Pattern Generate в режимах Toggle и Signal Value

Режим работы генератора Single Value позволяет пользователю выбирать отдельные значения с лицевой панели самому, ноль или единицу. Три других работают в режиме Toggle, который чередует единицу и ноль.

Для генераторов, работающих в режиме Toggle, я добавил делитель частоты (рисунок 25), который изменяет исходную частоту, подаваемую с генератора. В обычном случае генератор подает импульсы с частотой 1Гц. Нам же нужны три разных величины.

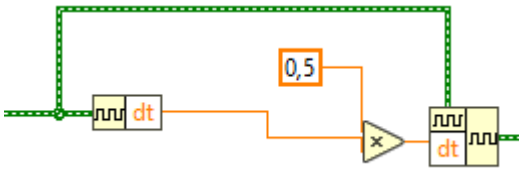


Рисунок 25 – Пример делителя частоты

Для вывода сгенерированного импульса на макетную плату NI ELVIS III добавлен элемент, с помощью которого цифровой сигнал преобразуется в двумерный массив нулей и единиц (рисунок 26).

Рисунок 26 – элемент преобразования цифрового сигнала в массив

Для того, чтобы вывести импульсы на макетную плату была создана структура FOR LOOP. Она позволяет провести необходимое количество итераций, где N – это количество наших импульсов (рисунок 27).

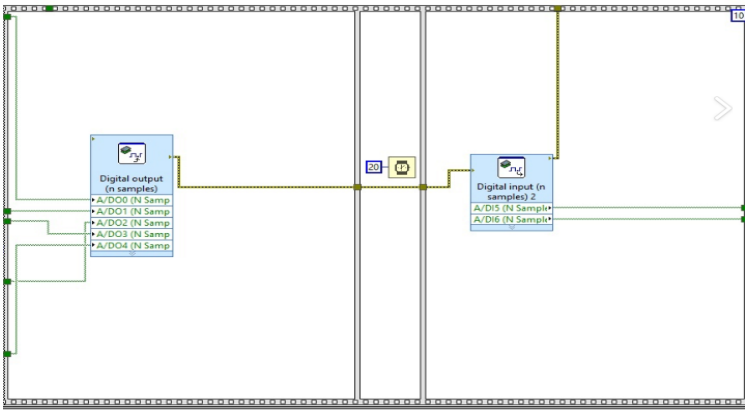


Рисунок 27 – структура FOR LOOP для вывода сигнала

Полученный массив выводится на выход макетной платы, а затем поступают на используемую микросхему. Далее, с микросхемы через соединительные провода импульс поступает на вход NI ELVIS III. Для получения более корректного результата была добавлена клиповая структура WAIT, с помощью которого реализована задержка в 20 мс.

Выходные данные с микросхем представляют из себя массив нулей и единиц. Для того, чтобы увидеть результат с выходов микросхемы нужно перевести наш массив обратно в цифровой сигнал. Для этого был добавлен элемент противоположенный начальному (рисунок 28).

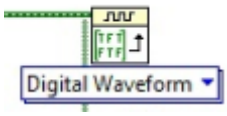


Рисунок 28 – Элемент преобразования массива в цифровой сигнал

Для удобства отображения были добавлены три кластера, для объединения данных рассмотренных в трех случаях (рисунок 29).

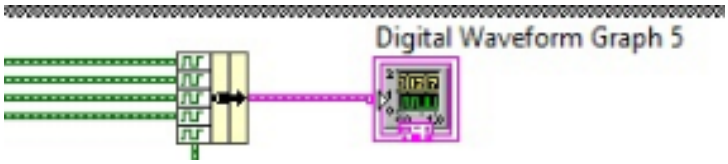


Рисунок 29 – Кластер и вывод данных на график

3.2 Разработка программы для лабораторной работы по изучению логических функций с использованием базовой логики

Вторая лабораторная реализована следующим образом: на основе элементов, которые использовались в предыдущей работе были разработаны две программы, которые можно использовать поочередно.

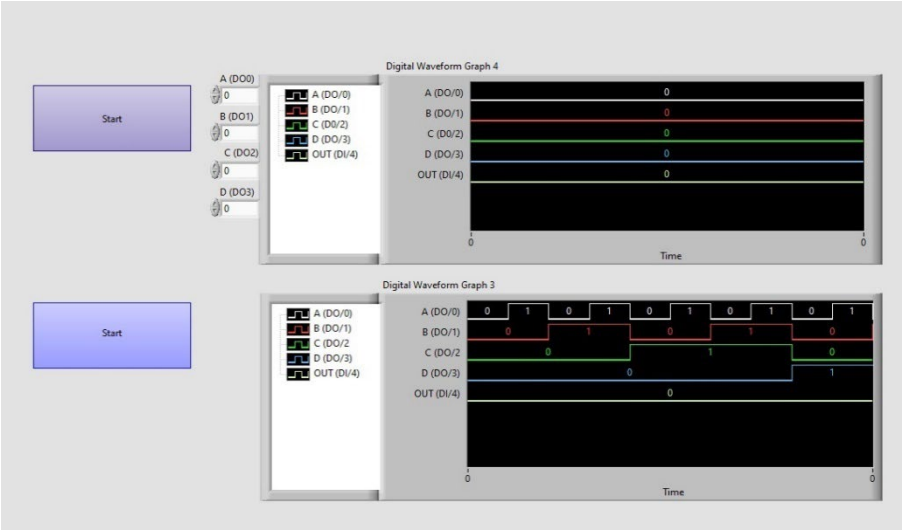


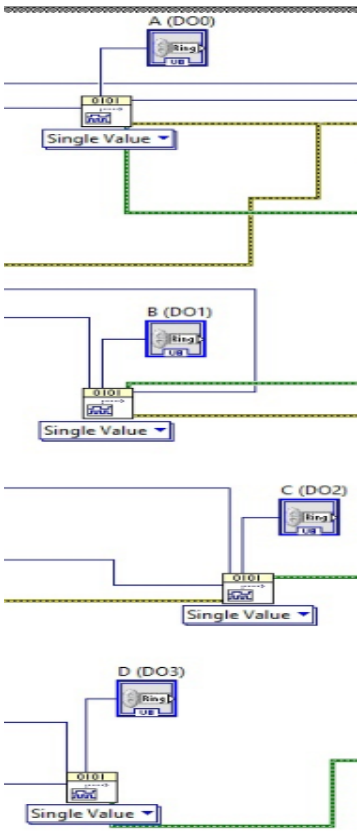
Рисунок 30 – Лицевая панель ВП

На лицевую панель выведены два графика (рисунок 30). На первый график приходят результаты статической работы генераторов. Пользователь сам выбирает, что ему отправить на входы микросхем (DO/0, DO/1, DO/2, DO/3) ноль или единицу. Результат выводится отдельным обозначением, к которому подключена микросхема (DI/4)

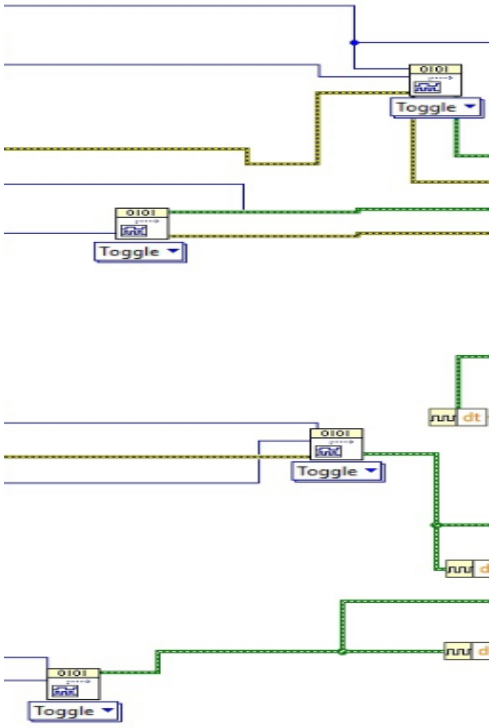
На второй график выводятся динамические значения генераторов. Заданные частоты уже заданы на блок диаграмме. Результат так же выводится отдельным обозначением.

Каждая программа начинает свою работу по кнопке «Start» расположенной рядом с графиком нужного режима работы.

Обратимся к блок-диаграмме (рисунок 31).



а)



б)

Рисунок 31 – Варианты работы генераторов
а – статическая работа;
б – динамическая работа

Как и в прошлой лабораторной работе для одной из программ нам нужно изменять исходную частоту генератора (рисунок 32).

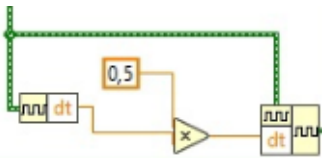


Рисунок 32 – делитель частоты для изменения частот

Следом за ним идет прибор, который преобразует наш сигнал в массив нулей и единиц (рисунок 33).

Рисунок 33 – преобразования цифрового сигнала в массив

Для того, чтобы вывести импульсы на макетную плату была создана структура FOR LOOP. Она позволяет провести необходимое количество итераций, где N – это количество наших импульсов (рисунок 34).

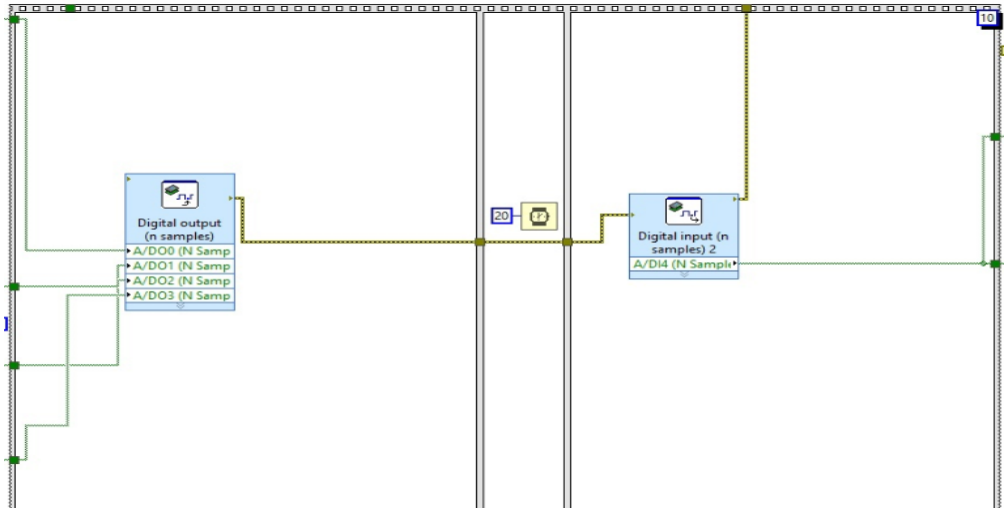


Рисунок - 34 – структура FOR LOOP для вывода сигнала

Полученный массив выводится на выход макетной платы, а затем поступают на используемую микросхему. Далее, с микросхемы через соединительные провода импульс поступает на вход NI ELVIS III. Для получения более корректного результата была добавлена клиповая структура WAIT, с помощью которого реализована задержка в 20мс.

Выходные данные с микросхем представляют из себя массив нулей и единиц. Для того, чтобы увидеть результат с выходов микросхемы нужно перевести наш массив обратно в цифровой сигнал. Для этого был добавлен элемент противоположенный начальному (рисунок 35).

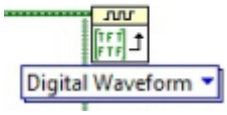


Рисунок 35 – Элемент преобразования массива в цифровой сигнал

В каждой из программ было установлено по одному кластеру, для объединения всех полученных данных, который затем отображается на графиках, которые отображаются на лицевой панели (рисунок 36).

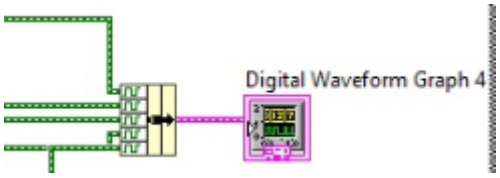
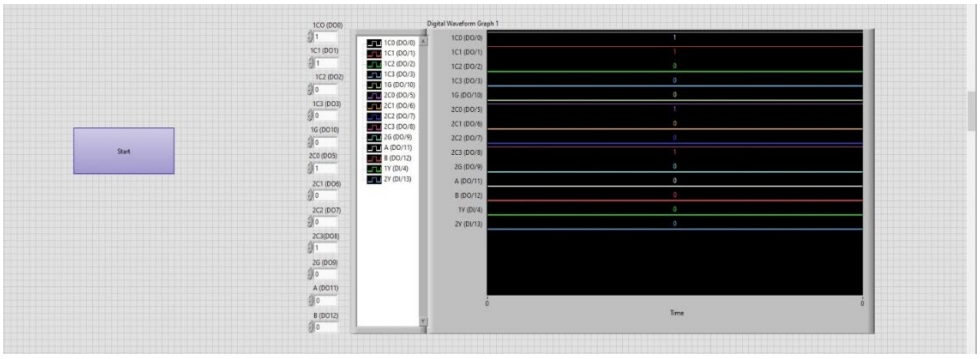


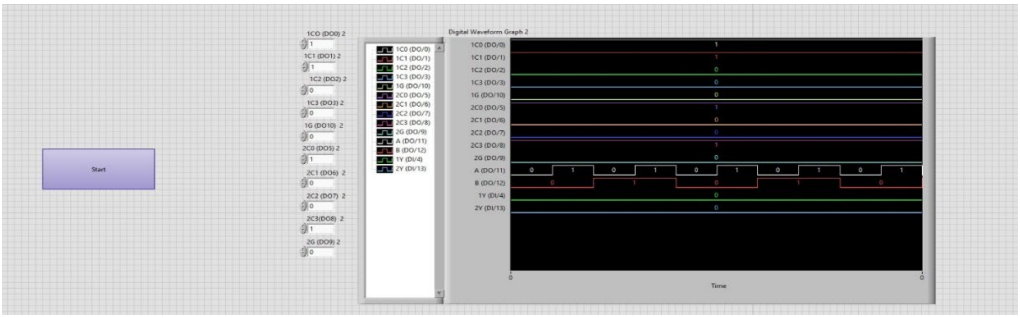
Рисунок 36 – элемент кластер и вывод данных на график

3.3 Разработка программы для лабораторной работы по изучению мультиплексора

Третья лабораторная реализована следующим образом: на основе предыдущих программ было собрана следующая программа.



а)



б)

Рисунок 37 – Лицевая панель ВП
а – значения задаются вручную;
б – установление значения для входов А и Б.

На лицевую панель выведены два графика. На первый график, (рисунок 37а) приходят результаты работы генераторов. Пользователь сам выбирает, что ему

отправить на входы микросхем ноль или единицу. Результат выводится отдельным обозначением, к которому подключена микросхема.

На второй график (рисунок 37б) выводятся значения, который пользователь так же задает вручную, только на указанные входы А и В подаются установленные значения частоты. Заданные частоты уже заданы на блок диаграмме.

Вернемся к блок-диаграмме (рисунок 38).

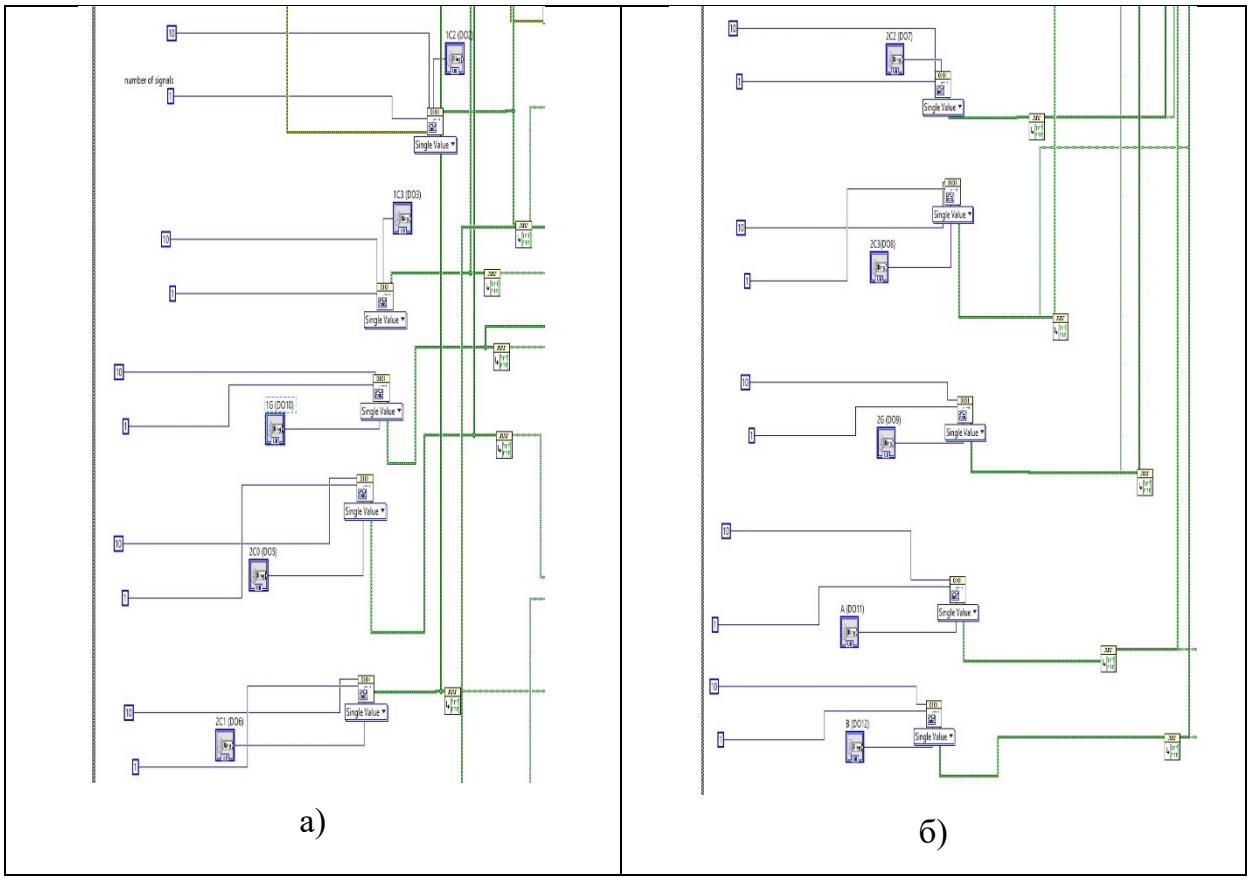


Рисунок 38 – Генераторы для первого режима работы

Для первой программы было создано двенадцать генераторов, работающих в режиме Single Value.

Затем установлен элемент, преобразующий наш сигнал в массив нулей и единиц (рисунок 39).

Рисунок 39 – Элемент преобразующий сигнал в массив

Для того, чтобы вывести импульсы на макетную плату была создана структура FOR LOOP. Она позволяет провести необходимое количество итераций, где N – это количество наших импульсов (рисунок 40.

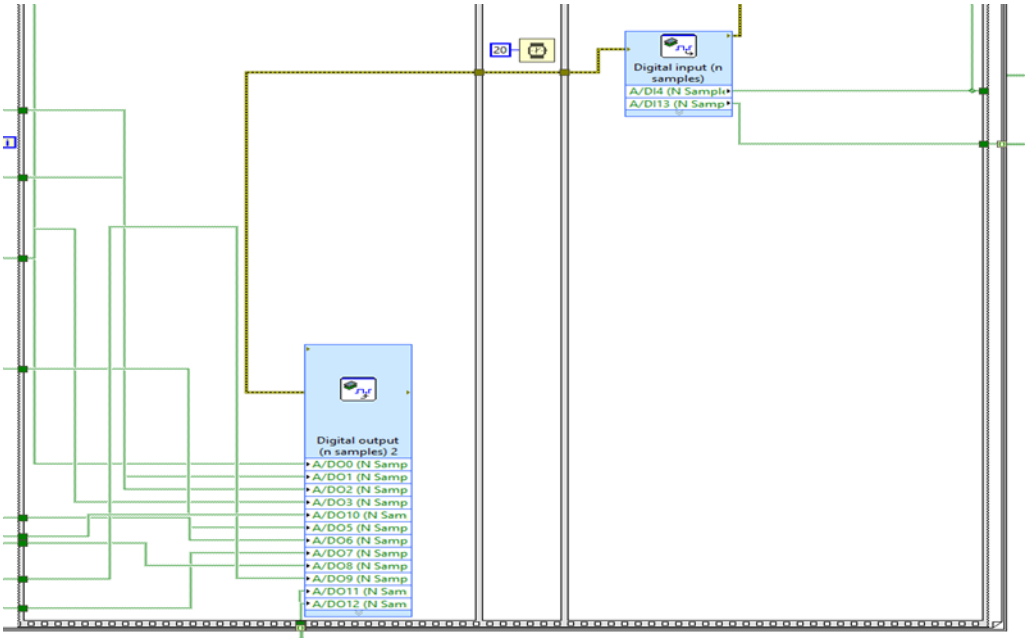


Рисунок 40 - Структура FOR LOOP для вывода сигнала на микросхему и с нее

Выходные данные с микросхем представляют из себя массив нулей и единиц. Для того, чтобы увидеть результат с выходов микросхемы нужно перевести наш массив обратно в цифровой сигнал. Для этого был добавлен элемент противоположенный начальному (рисунок 41).

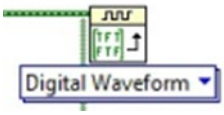


Рисунок 41 – Элемент преобразования массива в цифровой сигнал

В каждой из программ было установлено по одному кластеру, для объединения всех полученных данных, который, затем отображается на графиках, которые отображаются на лицевой панели (рисунок 42).

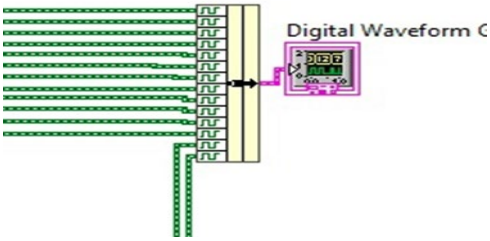


Рисунок – 42 Элемент кластер и вывод данных на график

Вторая программа отличается от первой тем, что генератор с заданной частотой (рисунок 43).

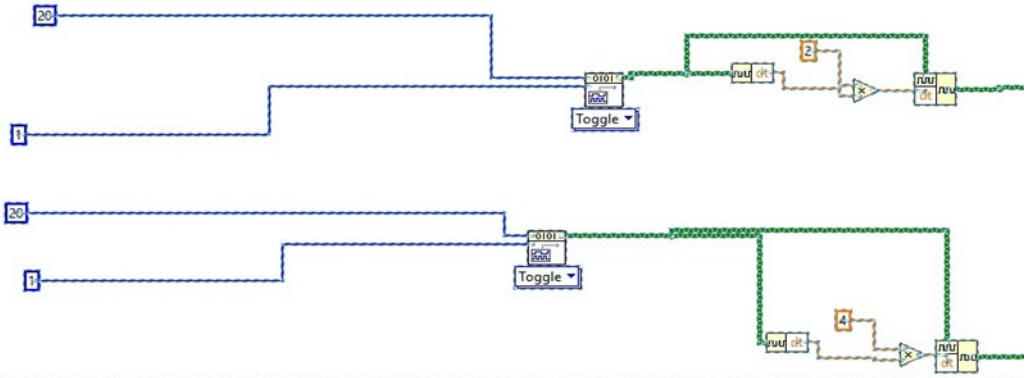


Рисунок 43 – Подключения генераторов во второй программе

ЗАКЛЮЧЕНИЕ

В ходе данной работы были получены следующие результаты:

1. Разработана программа для лабораторной работы по изучению логических элементов с использованием базовой логики. В этой программе реализованна подача на логические микросхему нуля или единицы, которые пользователь выбирает вручную, а так же подается сигнал с установленной частотой в самой программе. Каждый поданный сигнал и результат выводятся на отдельный график.
2. Разработана программа для лабораторной работы по изучению логических функций с использованием базовой логики. Программа состоит из двух блоков. В первом блоке на один из графиков на лицевой панели выведены результаты статической работы генератора, в котором пользователь сам выбирает, какое значение он отправляет на вход микросхемы. На второй график выведены динамический значения генераторов, заранее установленные программой.
3. Разработана программа для лабораторной работы по изучению мультиплексора. В работе на один из графиков поступают заданные пользователем значения на микросхему, ноль или единица. На второй график приходят также заданные пользователем значения, только на обозначенные входа подается сигнал установленный в программе.

Данные лабораторные работы послужат студентам хорошей возможностью еще лучше узнать о прицепе работы комбинационной логики.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1 Евдокимов Ю.К. LabVIEW для радиоинженера: от виртуальной модели до реального прибора / Евдокимов Ю.К., Линдваль В.Р., Щербаков Г.И. // Изд-во ДМК ПРЕСС – 2007. – 400 с.

2 Тревис Д. LavVIEW для всех / Джефри Т. // Изд-во ДМК ПРЕСС – 2005. – 544 с.

3 Волков О. Ю. Практикум по радиоэлектронике. Цифровые схемы / Волков О// Физический факультет МГУ им. М.В. Ломоносова – 2016. – 96 с.

4 Дэвид М. Цифровая схемотехника и архитектура компьютера // Дэвид М., Сара Х. // Изд-во Морган Кауфман – 2013. – 1619 с.

5 Лупов С.Ю. LabVIEW в примерах и задачах / Лупов С.Ю., Муякшин С.И., Шариков В.В. // Нижний Новгород – 2007. – 101 с.

6 Осадченко В.Х. Базовые логические элементы цифровой техники / Осадченко В.Х., Волкова Я.Ю., Германенко А.В., Зеленовский П.С. // Изд-во Урал – 2018. – 120 с.

7 NI ELVIS III руководство по эксплуатации / Центр технологий National Instruments – Новосибирск, 2021. – URL: https://nitec.nstu.ru/upload/lib/2020_Translated/ELVIS%20III_Руководство%20по%20эксплуатации%20(Начало%20работы).pdf

АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

Отчет о проверке на заимствования №1

Автор: Горовой Антон
Проверяющий: Горовой Антон

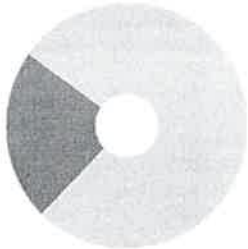
Отчет предоставлен сервисом «Антиплагиат» - http://users.antiplagiat.ru

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 44
Начало загрузки: 30.01.2023 20:28:36
Длительность загрузки: 00:00:01
Имя исходного файла: ДИПЛОМ.pdf
Название документа: ДИПЛОМ
Размер текста: 42 кБ
Символов в тексте: 43176
Слов в тексте: 5181
Число предложений: 312

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Начало проверки: 30.01.2023 17:28:37
Длительность проверки: 00:00:01
Комментарии: не указано
Модули поиска: Интернет Free



СОВПАДЕНИЯ	САМОЦИТИРОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
22,8%	0%	0%	77,2%

Совпадения — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Совпадения, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Источник	Актуален на	Модуль поиска
[01]	7,78%	trevis d. labview dlya vseh. (2005).djvu http://inethub.olvi.net.ua	раньше 2011	Интернет Free
[02]	5,62%	Konspekt_lekcij.doc http://library.mephi.ru	06 Сен 2019	Интернет Free
[03]	0%	Конспект лекций по курсу ТОС- МЕТОДЫ И СРЕДСТВА ЦОС Лектор- доцент В http://samzan.ru	05 Янв 2017	Интернет Free

Еще источников: 7
Еще совпадений: 8,4%

Научный руководитель Торгаев С.Н.