ВОЕННАЯ АКАДЕМИЯ СВЯЗИ

Кафедра №20

Курсовой проект

На тему: «Программа виртуального синтеза цифровых схем с учётом особенностей эммуляции процессорного устройства»

Исполнил: курсант 351 учебной группы Тютрин Н.Ю.

Проверил: преподаватель кафедры №20

подполковник Мордовин В.Н.

Санкт-Петербург 2009

Содержание

Перечень принятых сокращений

Введение

1. Возможности применения программы в учебном процессе

2. Технические характеристики программы

3. Описание тела программы

3.1 Обоснование выбора языка программирования

3.2 Структура программы

3.3 Принцип работы программы

Заключение

Литература

Приложение 1

Приложение 2

Приложение 3

Приложение 4

Приложение 5

Приложение 6

Приложение 7

Приложение 8

Приложение 9

Приложение 10

Приложение 11

Приложение 12

Перечень принятых сокращений

1. ВАС – Военная академия связи.

2. ВТ – вычислительная техника.

3. ПО – программное обеспечение.

4. ПК – персональный компьютер.

5. УГО – условное графическое обозначение.

6. ООП – объектно-ориентированное программирование.

7. ЯП – язык программирования.

Введение

Целью предлагаемой разработки является создание программного обеспечения (ПО), используемого в учебном процессе в военных и гражданских ВУЗах. Программа позволяет эмулировать виртуальную рабочую среду для сборки, отладки, а также проверки функционирования устройств на базе цифровых интегральных микросхем. В работе произведена неполная эмуляция элементной базы ТТЛ и ТТЛШ, а не готовых технических устройств, что позволяет учащимся по мере возможностей самостоятельно производить проектирование, сборку и отладку исследуемых схем.

Программа также может использоваться при написании курсовых и дипломных проектов, связанных с аппаратными разработками на базе цифровых микросхем. Она может стать хорошим подспорьем при проведении самых разнообразных лабораторных работ. Преподаватель получает дополнительные возможности контроля знаний учащихся посредством имеющейся электронной библиотеки исправных и неисправных схем. Составлены методические разработки для проведения лабораторных работ, предполагающие отдельные варианты заданий для каждой бригады курсантов. Подобный подход позволяет учесть как индивидуальную подготовленность учащихся, так и специализацию всей группы в целом.

В Военной академии связи (ВАС) предлагаемое ПО уже применяется в учебном процессе при изучении таких специальных предметов, как «Микропроцессорные системы» (ДД – 420 – 14) и «Архитектура вычислительных машин» (ДД – 420 – 24). Использование данной программы также рекомендуется при изучении предметов «Схемотехника ЭВМ» (ДД – 420 – 10) и «Организация ЭВМ и систем» (ДД – 420 – 4). Кроме указанных дисциплин, предлагаемая разработка может найти применение при изучении любого предмета, связанного с цифровой схемотехникой.

Проблема эмуляции рабочей среды для построения и отладки всевозможных электронных устройств возникла давно и до сего дня не теряет своей актуальности. Практическая значимость решения данной задачи с годами только растёт, поскольку с развитием средств вычислительной техники (ВТ), при постоянном совершенствовании программного обеспечения (ПО) ускоренно расширяются возможности компьютерной эмуляции. Задачи, ещё недавно казавшиеся невыполнимыми, такие как, например, компьютерное моделирование работы процессорного устройства, сегодня уже не кажутся особо сложными.

Проблема компьютерного моделирования работы электронных устройств в последние годы вызывала живой интерес программистов, как в нашей стране, так и за рубежом. Оригинальными и популярными вариантами её решения стали такие известные в мире программы, как Simulink, WorkBench и LabView. Указанные типы ПО являются мощной и надёжной базой для решения множества инженерных задач.

Однако универсальность подобных разработок может создать определённые неудобства при их использовании в педагогических целях. Отображение элементов составленной схемы посредством условных графических обозначений (УГО) не создаёт эффекта живого контакта с реальным техническим устройством и не способствует лучшему усвоению материала. С другой стороны, универсальность и сложность указанного выше ПО требует длительной подготовки учащихся к работе с подобной программой. Затраты времени, отводимого на данную подготовку, ведут к сокращению числа остальных учебных часов. Использование программы, оперирующей с фотографически точными изображениями микросхем, по мнению авторов, способствует лучшему осмыслению изучаемого материала.

Проблемы педагогического характера способствовали принятию решения о самостоятельной разработке ПО, узкоспециализированного для виртуальной сборки цифровых схем. Можно сказать, что на текущий момент прямого аналога данное ПО не имеет. С учётом объёма, занимаемого программой, её скоростных качеств, а также личных предпочтений авторов, в качестве платформы программирования был избран Borland C++ Builder 6.

Необходимость компьютерного моделирования учебной лабораторной базы обусловливается тем, что используемая в настоящее время во многих ВУЗах лабораторная база для изучения цифровой электроники не всегда удовлетворяет современным требованиям. Стенды, выпускаемые ранее отечественной промышленностью, устарели. В тоже время новых стендов, отражающих современное развитие вычислительной техники, не создаётся, что часто и вполне оправданно вынуждает прибегать к собственноручному изготовлению макетов (подобный подход практикуется в некоторых учебных заведениях).

Но это не может решить существующих проблем:

- Чаще всего исследуется лишь одна микросхема или один тип микросхем, что сужает диапазон проводимых исследований.

- В условиях жёсткой сборки отсутствует возможность изменения конфигурации схемы.

- Отсутствие возможности самостоятельной сборки схемы, отладки её работы и устранения найденных неисправностей.

- Не предусматривается самостоятельное проектирование цифровых схем (по мнению авторов – главный недостаток).

- Не предусмотрено выполнение различными бригадами различных вариантов лабораторных работ.

- Отсутствие возможности контроля знаний посредством электронной библиотеки неисправных схем – самостоятельного нахождения неисправности и её устранения.

- Любая сборка (и в первую очередь нефабричная) не исключает вероятности частых поломок, что может стать особенно нежелательно с учётом возможной неподготовленности студентов или курсантов.

Всё сказанное относится как к самодельным стендам, так и к стендам, выпускаемым промышленностью.

Решение не всех, но многих вопросов (по крайней мере, всех перечисленных) даёт использование компьютерного моделирования учебной лабораторной базы.

Возможности применения предлагаемой разработки выходят за рамки учебного процесса. Она может эффективно использоваться в научно-исследовательских учреждениях и на производстве. Для выпуска электронной продукции предполагаются стадии разработки, сборки и отладки выпускаемого изделия. Все эти стадии требуют наличия квалифицированного персонала. Они также связаны со значительными временными затратами. Применение предлагаемой программы требует лишь наличия соответствующего компьютера и одного оператора. Таким образом, сводятся к минимуму как аппаратные, так и временные затраты, связанные со всеми тремя перечисленными стадиями производства. Так, например, виртуальная сборка устройства, включающего 5 – 6 цифровых микросхем средней и даже большой степени интеграции обычно занимает всего лишь 15 – 20 минут, чего, конечно, нельзя ожидать в условиях сборки реальной.

Включение в тело программы таких элементов, как виртуальный логический анализатор и многофункциональный виртуальный генератор – устройств, по своим параметрам во многом превосходящих свои реальные аналоги, делают разработку самодостаточной, не требующей никаких дополнительных элементов.

1. Возможности применения программы в учебном процессе

Предлагаемая разработка может активно использоваться в учебном процессе в ВАС. Так, например, при изучении предмета «Микропроцессорные системы» (ДД – 420 – 14) она полностью обеспечивает проведение практических занятий на персональных компьютерах (ПК) по темам: «Проектирование на ПК и виртуальная отладка устройства преобразования сигналов» (8 учебных часов), а также «Проектирование на ПК и виртуальная отладка вычислительного устройства» (8 учебных часов).

При изучении предмета «Архитектура вычислительных машин» (ДД – 420 – 24) данное ПО полностью обеспечивает выполнение курсового проекта с использованием средств ВТ (8 учебных часов). Кроме того, оно обеспечивает проведение практического занятия на ПК по теме: «Проектирование на ПК и виртуальная отладка вычислительного устройства» (4 учебных часа).

Таким образом, только по двум указанным дисциплинам использование предлагаемой разработки способно обеспечить проведение практических занятий с применением ВТ в объёме 28 часов. Сначала курсанты под руководством преподавателя разрабатывают исследуемый компьютерный узел. Затем виртуально «собирают» его на экране ПК, отлаживают и убеждаются в работоспособности. Для наиболее подготовленных учащихся предполагаются задания повышенной сложности. Фотографически точное (вплоть до бликов и теней) воспроизведение цифровых микросхем, а также вспомогательных и индикаторных элементов создаёт иллюзию работы с реальными техническими устройствами.

Помимо указанных двух предметов программа может быть применена при изучении любой дисциплины, связанной с аппаратной частью ВТ («Схемотехника ЭВМ», «Организация ЭВМ и систем» и других).

2. Технические характеристики программы

Технические характеристики применяемого на сегодняшний день программного и аппаратного обеспечения вполне приемлемы для широкого использования компьютерной имитации лабораторных стендов. Программное обеспечение для проведения компьютерного моделирования учебной лабораторной базы, как правило, не требует мощных вычислительных ресурсов. Используются аппаратные средства, не обладающие высокой производительностью, доступные любому учебному заведению. Примером программы, позволяющей имитировать создание и отладку цифровых схем в процессе обучения, является разработка, используемая в ВАС.

Программа называется «ICBuilder», исполнительный файл – «ICBuilder.exe». Она инсталлируется на диск C: в каталог Program Files в подкаталог ICBuilder. Программа написана с использованием платформы С++ Builder, её использование требует персонального компьютера на базе центрального процессора не ниже Intel Pentium I, с установленной операционной системой не ниже Windows 95. Подобные технические требования не являются на сегодняшний день хоть сколько-нибудь обременительными.

В разархивированном виде используемая программа совместно со всеми библиотеками, а также заранее составленными примерами занимает не более 3 Мбайт, в самораспаковывающемся архивном файле – около 1,8 Мбайта. Подобный объём легко вмещается практически на любой носитель и, таким образом, установка программы на новый компьютер не сопряжена ни с какими сложностями. При разработке аналогичной программы, с использованием другого языка программирования – Delphi – предполагается занимаемый объём примерно такого же порядка.

Процесс виртуальной сборки цифровых схем не представляет сложности для учащихся. Пользовательский интерфейс предполагает развёрнутое окно, в котором посредством соответствующих манипуляций левой кнопкой мыши вызываются необходимые для виртуальной сборки проектируемого устройства цифровые микросхемы. Большинство операций с уже вызванными микросхемами ориентировано на левую кнопку мыши и лишь отдельные из них используют правую кнопку.

На данный момент в программе используются микросхемы двадцати трёх типов, и производится работа по дальнейшему увеличению их количества. Число вызываемых микросхем одного типа ограничено лишь размерами рабочего поля программы.

3. Описание тела программы

3.1 Обоснование выбора языка программирования

При выборе языка программирования для написания проекта на ЭВМ, возникает необходимость формулировки требований к программе:

- графическая точность воспроизводимых объектов (все эмулированные в программе устройства должны максимально соответствовать оригиналам);

- удобство использования программы (простота доступа к создаваемым устройствам, справке, проектам и пр.);

- возможность получения справочной информации;

- функциональное соответствие реальных и виртуальных микросхем;

- функциональное разнообразие виртуальных устройств;

- возможность загрузки файлов проектов;

- высокая скорость работы при передаче сигналов между устройствами;

- малый объем программы.

Необходимость создания небольшого, хорошо структурированного кода, имеющего высокую скорость работы, требует применения двух подходов: объектно-ориентированного программирования (ООП) и использования языка программирования «С».

ООП является в настоящее время естественным путём к построению сложных программ и систем. Его упрощенное описание сводится к следующему:

При запуске любой программы в ОС Windows, на экране появляется окно с множеством кнопок, разделов меню, окон редактирования, списков и т.п. Всё это – объекты. Они ожидают событий – нажатия пользователем клавиши, кнопки мыши, перемещения курсора и т.д. При совершении подобного события, объект получает сообщение и может как-то отреагировать: выполнить некоторые вычисления, свернуть или закрыть окно, занести символ в окно редактирования.

Второй подход – использование языка программирования (ЯП) «С», яваляющегося ЯП среднего уровня. Применение ЯП высокого уровня приводит к росту объёма программы. В свою очередь ЯП низкого уровня, при небольшом объёме исходных кодов, является сложным для написания подобной программы в связи с тем, что он непосредственно работает с командами процессора с учетом логики его работы. ЯП среднего уровня использует преимущества двух других типов ЯП, в этом причина его популярности.

Объектно-ориентированный язык (ООЯ) С++ объединяет в себе перечисленные подходы. Данный ЯП выбран для написания программы. Удобную среду программирования для С++ предоставляет фирма производитель ПО Borland. Среда называется С++ Builder 6.

3.2 Структура программы

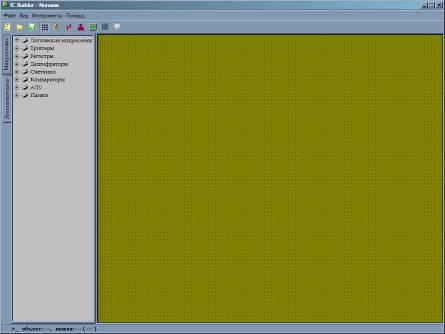


Рис.2 Окно программы

Программа включает следующие объекты:

1) окна:

- окно-заставка;

- главное окно;

- окно логического анализатора;

- окно «О программе»;

2) объекты окна (кнопки, меню, списки и пр.);

3) объекты устройств (интегральные микросхемы (ИМС), дополнительные устройства, соединения и т.п.).

Физически все перечисленные объекты находятся в модулях. Модуль включает два файла с расширениями «.h» и «.cpp», первый из них объявляет класс, а второй описывает его.

Базовый класс

Класс линии

Класс ИМС

Класс ВУ

Класс сумматора

Класс кнопки

Базовый класс

Класс линии

Класс ИМС

Класс ВУ

Класс сумматора

Класс кнопки

Рис.3 Иерархия классов программы

Каждый объект является экземпляром класса, т.е системой, созданной по «правилам» класса. Класс имеет свойства и методы. Свойства служат для изменения закрытых переменных класса, а методы являются функциями, имеющими доступ к закрытым переменным своего класса.

3.3 Принцип работы программы

Принцип работы программы наглядно иллюстрируется примером передачи кнопкой сигнала индикатору (предполагается, что эти два устройства между собой связаны). Нажатие кнопки левой клавишей мыши означает срабатывание события кнопки. Обработка этого события включает следующие этапы:

- изменение рисунка кнопки (создание визуального эффекта утопленной кнопки);

- изменение внутреннего состояния кнопки (с 0 на 1);

- отправка сигнала от кнопки по всем имеющимся соединениям;

- отправка сигнала от соединения к индикатору;

- «оценка» индикатором типа полученного сигнала (0 или 1);

- изменение внутреннего состояния индикатора, затем – обновление рисунка.

Заключение

Результаты произведённой работы свидетельствует о достаточной полноте решения поставленных задач. Использование предлагаемой разработки в учебном процессе уже охватывает 28 учебных часов по двум изучаемым дисциплинам. Возможно дальнейшее увеличение количества учебного времени, обеспечиваемого применением данной программы, за счёт использования её при изучении других предметов. Курсанты с искренним интересом относятся к занятиям, позволяющим самостоятельно составить принципиальную схему исследуемого устройства и тут же, на практике, убедиться в его работоспособности.

Отличительной особенностью последнего варианта разработанной программы явилась возможность проверки работы исследуемых устройств не в статическом режиме, посредством подачи одиночных импульсов, а в динамике. Решение данной задачи осуществилось посредством введения в тело программы виртуального генератора импульсов и 128-канального логического анализатора с памятью на 256 логических перепадов.

Проявленный учащимися интерес к используемому программному обеспечению облегчил работу преподавателя, позволил повысить качество знаний, получаемых курсантами.

Возможности применения предлагаемого ПО не ограничиваются учебными заведениями. Оно может использоваться в научно-исследовательских институтах военного и гражданского профиля, а также на производстве.

Литература

1. Архангельский А. Я. Программирование в C++ Builder 6., М.: Бином, 2004. 1152 с.

2. Шилдт Герберт. Полный справочник по С++, 4-е издание. : Пер. с англ. М. : Издательский дом “Вильямс”, 2004. 704 с.

3. Шило В. Л. Популярные микросхемы ТТЛ. М: “Аргус”, 1993. – 64 с.: ил.

4. Лебедев О. Н. и др. Изделия электронной техники. Цифровые микросхемы. Микросхемы памяти: Справочник. М.: Радио и связь, 1999. 248 с.: ил.

5. Бунтов В. Д., Макаров С. Б. Цифровые и микропроцессорные радиотехнические устройства: Учеб. Пособие. СПб.: Изд-во Политехн. Ун-та, 2005. 399 с.

Приложение 1

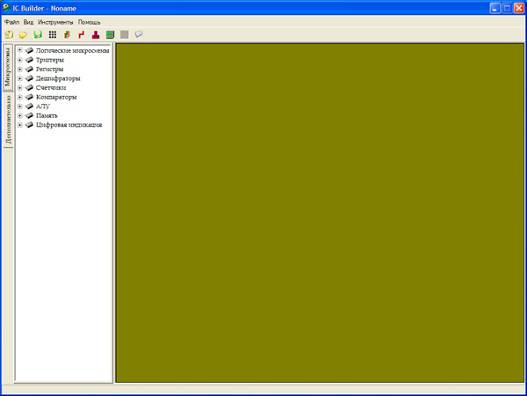


Рис.4 Рабочее поле программы

Приложение 2

Главное меню

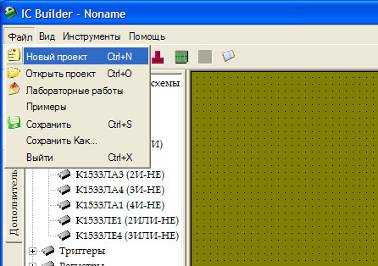


Рис.5 Позиция «Файл».

Включает в себя следующие опции:

- Новый проект

- Открыть проект

- Лабораторные работы (открывается соответствующая директория)

- Примеры (заранее собранные схемы)

- Сохранить

- Сохранить Как…

- Выйти

(Все опции имеют стандартное назначение).

Приложение 3

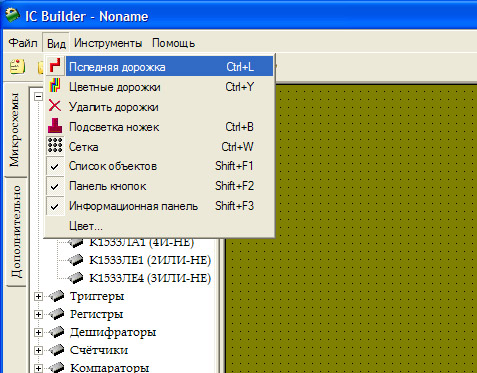


Рис.6 Позиция «Вид».

Включает в себя следующие опции:

- Последняя дорожка (соединение)

(Отображается лишь оно – рекомендуется при составлении сложных схем, когда в связи с большим числом соединений бывает трудно наблюдать сам факт проведения соединения).

- Цветные дорожки

(Соседние соединения отображаются разными цветами. В программе задано шестнадцать цветов, однако, возможно увеличение их количества. Использование данной опции удобно для наглядного просмотра соединений при составлении сложных схем.)

- Удалить дорожки

(Удаляются все соединения).

- Подсветка ножек

(Выделяются красным цветом все выходы, питание и «земля». Рекомендуется для предотвращения попытки соединения двух выходов микросхем, не обладающих Z-состоянием.)

- Сетка

(Может быть вызвана на рабочее поле – рекомендуется для более точного позиционирования элементов и придания схеме более эстетичного вида).

- Список объектов

(Возможность его удаления позволяет расширить рабочее поле).

- Панель кнопок

(Возможность удаления пиктографического меню – также ведёт к расширению рабочего поля).

- Информационная панель

(Возможность удаления подсказки в низу экрана).

- Цвет

(Выбор цвета рабочего поля).

Приложение 4

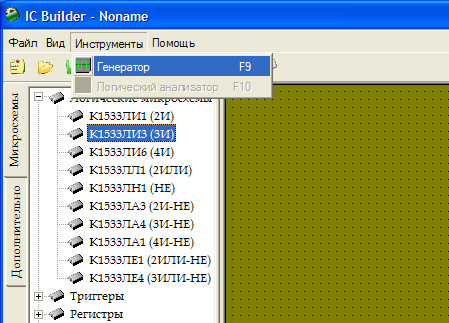


Рис.7 Позиция «Инструменты».

Включает следующие опции:

- Генератор

(Вызов генератора. Используется виртуальный четырёхфазный генератор со сдвигом фаз на 1800. Имеется возможность дискретного изменения его частоты от 0,5 Гц до 10 МГц. Также допускается генерация конечного числа импульсов, количество которых задаётся в специально вызываемом меню.)

- Логический анализатор

(Вызывается только после вызова генератора. Имеет практически неограниченное число каналов (из-за ограничений рабочего поля – до 128). Имеет внутреннюю память – запоминает до 256 перепадов импульсов. При необходимости данный параметр легко увеличить.)

Приложение 5

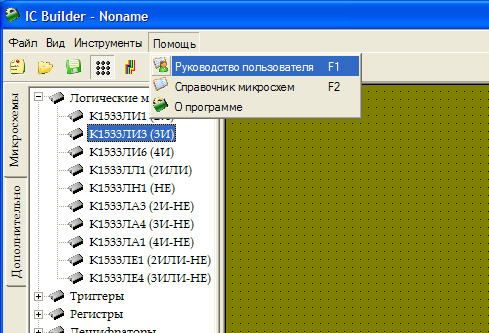


Рис.8 Позиция «Помощь».

- Справочник микросхем

- О программе

(Все опции имеют стандартное назначение. В программу введён электронный справочник микросхем, включающий микросхемы, используемые разработкой. Введение данной опции существенно экономит учебное время, поскольку отпадает необходимость выдачи справочного материала.)

Приложение 6



Рис.9 Пиктографическое меню

Все основные команды главного меню представлены в пиктографическом меню под ним. В пиктографическом меню представлены лишь те команды, использование которых происходит наиболее часто:

- Новый проект

- Открыть проект

- Сохранить

- Сетка

- Цветные линии

- Последнее соединение

- Подсветка выходов

- Генератор

- Логический анализатор

- Справочник микросхем

Приложение 7

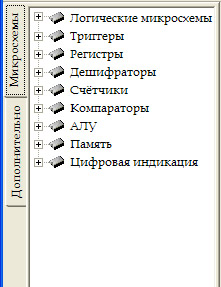


Рис.10 Вкладки

В работе программы используются две основные вкладки: «Микросхемы» и «Дополнительно».

- Вкладка «Микросхемы» позволяет вывести на рабочее поле любую из микросхем, представленных деревом объектов.

Для этого открывается соответствующий класс микросхем, нужная микросхема помечается левой кнопкой мыши и, не отпуская её, «перетаскивается» на рабочее поле. Допускаются неоднократные вызовы одной и той же микросхемы.

- Вкладка «Дополнительно» позволяет вывести на рабочее поле переключатели (из 0 в 1), кнопки, индикаторы четырёх цветов, надпись, напряжения логического нуля и логической единицы.

Приложение 8

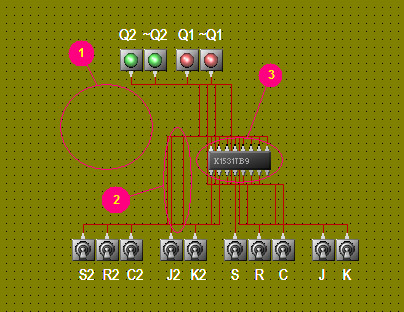


Рис.11 Вызванная микросхема совместно с индикаторными и вспомогательными устройствами: 1. – рабочая область (поле); 2 – соединение (дорожка) – графическая линия, соединяющая ножки устройств. 3 – объект (интегральная микросхема или вспомогательное устройство).

Приложение 9

Работа с программой

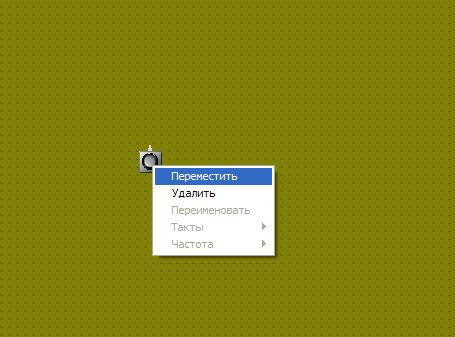


Рис.12 Перемещение и удаление

Вызванный объект (микросхему, вспомогательный элемент или надпись) можно переместить или удалить, наведя на него курсор, нажав правую кнопку мыши, и, выбрав левой кнопкой нужное действие. При перемещении возникает передвигаемый мышью контурный образ объекта. Фиксация нового положения объекта осуществляется передвижением курсора (не нажимая кнопок мыши) на новое место с последующим нажатием левой кнопки.

Приложение 10

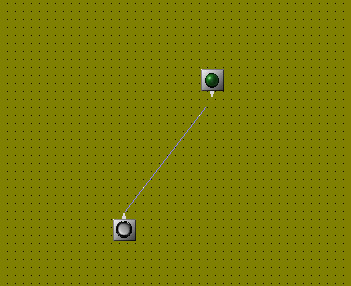


Рис.13 Выполнение соединений

Производится левой кнопкой мыши при нажатии и удержании кнопки. Соединение производится от выхода ко входу.

Удаление соединений аналогично удалению объекта. При наведении курсора на соединение оно маркируется (становится чёрным). Удаление объекта возможно только после удаления всех его соединений.

Приложение 11

Создание надписи.

Во вкладке «Дополнительно» выбирается позиция «Подпись». Не отпуская левой кнопки мыши, выбранная позиция перетаскивается на рабочее поле программы. В той точке рабочего поля, где будет отпущена кнопка мыши, появится надпись. В появившееся при отпускании кнопки окно ввести текст надписи и нажать кнопку «ОК».

Перемещение надписи аналогично перемещению микросхемы.

Приложение 12

В качестве практических рекомендаций по применению предлагаемого ПО приведём лишь одну из методических разработок для проведения лабораторных работ (разработка приводится с некоторыми сокращениями).

Лабораторное занятие №1

Логический узел на элементах ТТЛ

Предмет исследования: Логические схемы на элементах ТТЛ (серии: К 155, К 531, К 555, КР 1533, КР 1531)

Цели занятия:

Уяснить процессы прохождения сигналов в логических схемах на элементах ТТЛ.

Приобрести необходимые навыки в сборке цифровых схем.

Практически ознакомиться с работой логических вентилей.

Порядок выполнения работы:

1. Ознакомление с работой стенда:

1.1. Проделать все действия, предусмотренные в разделе «Краткие теоретические сведения» (раздел не приводится).

1.2. Ознакомиться с работой электронного справочника микросхем, изучить назначение выводов исследуемых логических вентилей ознакомиться с УГО изучаемых микросхем.

1.3. Вызвав одну из микросхем на экран и, удобно расположив её (например, микросхему 155ЛИ1 – 2И), виртуально «собрать» на экране схему включения логического элемента (Рис.14). Желательно в верхней части рабочего поля помещать индикаторные элементы, под ними – микросхемы, а в нижней части – вспомогательные элементы.

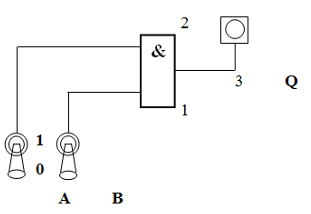


Рис.14

1.4. Продемонстрировать преподавателю работу данной схемы.

1.5. Составить таблицу истинности рассмотренного вентиля:

Табл.1

|  |  |  |
| --- | --- | --- |
| A | B | Q |
|  |  |  |

1.5. Занести схему включения и таблицу в отчёт.

1.6. Добавив к вентилю микросхемы 155ЛИ1 логический инвертор (микросхема 155ЛН1), построить схему инвертирующего вентиля:

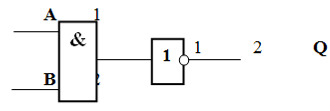


Рис.15

1.7. Продемонстрировать преподавателю работу данной схемы.

1.8. Составить аналогичную п.1.3. таблицу и занести схему включения и таблицу в отчёт

2. Построение логических функций с использованием заданных микросхем:

2.1. Используя заданные микросхемы: 1533ЛИ1 (2И), 1533ЛЛ1 (2ИЛИ), 1533ЛН1 (НЕ), 1533ЛА3 (2И-НЕ), составить схемы, реализующие логические функции, соответствующие своему варианту.

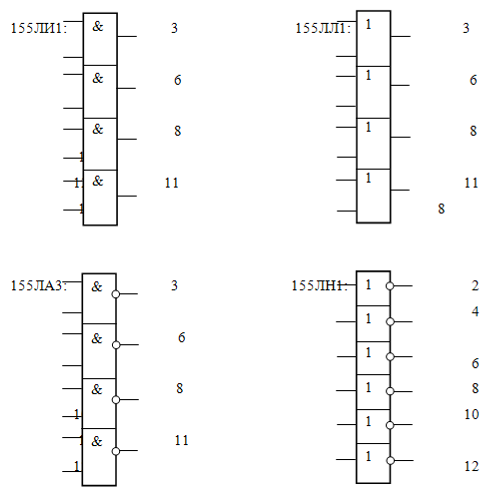
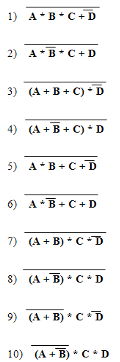


Рис.16

Варианты логических функций (по № рабочего места):



2.2. Построить в отчёте логическую схему, соответствующую своему варианту и показать её преподавателю.

2.3. «Собрать» на экране схему, соответствующую своему варианту.

2.4. Составить таблицу истинности «собранной» схемы и занести её в отчёт. Результаты показать преподавателю.

3. Проверка теоремы де Моргана.

3.1. Для чётных вариантов:

Используя вентили И и НЕ построить схему вентиля ИЛИ, «собрать» её и составить таблицу истинности.

Для нечётных вариантов:

Используя вентили ИЛИ и НЕ построить схему вентиля И, «собрать» её и составить таблицу истинности.

3.2. Поменять чётные и нечётные варианты и повторить действия.

Таблицы истинности и схемы занести в отчёт.

4. Построение одноразрядного управляемого инвертора.

(В зависимости от управляющего сигнала информационный сигнал либо поступает со входа на выход без изменения, либо инвертируется)

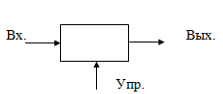


Рис.17

4.1. Используя три логических вентиля: И-НЕ, ИЛИ, И, спроектировать указанную схему и показать её преподавателю.

4.2. «Собрать» составленную схему.

4.3. Составить таблицу истинности управляемого инвертора и вместе со схемой занести в отчёт.

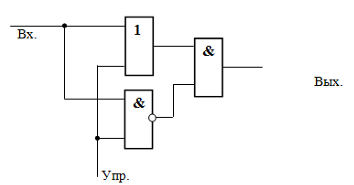


Рис.18 Схема управляемого инвертора (составляется курсантами самостоятельно)

Таблица истинности управляемого инвертора

|  |  |  |
| --- | --- | --- |
| Упр. | Вх. | Вых. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

5. Построение элементов с двумя устойчивыми состояниями на базе логических вентилей И-НЕ.

5.1. Используя микросхему 155ЛА3, построить схему:

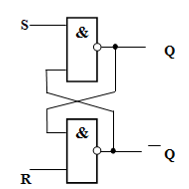


Рис.19

5.2. Составить таблицу истинности элемента с двумя устойчивыми состояниями и занести схему и таблицу истинности в отчёт. Объяснить принцип работы данной схемы.

Контрольные вопросы:

Что называется логическим элементом?

Нарисовать схему логического вентиля, запрещающую прохождение импульсов по сигналу «0» и по сигналу «1».

Какой уровень помехи: в сторону увеличения напряжения или в сторону его уменьшения более опасен (в плане искажения информации) при нулевом уровне сигнала, а какой – при единичном уровне сигнала?