|  |  |
| --- | --- |
| Министерство науки и высшего образования Российской Федерации  ОРСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)  ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  «ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  Факультет среднего профессионального образования | |
| **Курсовая работа**  по междисциплинарному курсу «Технология разработки программного обеспечения»  профессионального модуля «Осуществление интеграции программных модулей»    **Разработка программного обеспечения для исследования работы логических устройств компьютера**  Пояснительная записка  ОГУ 09.02.07. 3024. 083 ПЗ | |
|  | Руководитель работы  преподаватель высшей категории  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ж. В. Михайличенко  «\_\_\_» \_\_\_\_\_­\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.  Студент группы 22ИСП-1  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К. Г. Бекеша  «\_\_\_» \_\_\_\_\_­\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |
| Орск 2024 | |

|  |
| --- |
| Утверждаю  Председатель предметно-цикловой комиссии дисциплин профессионального цикла |
| \_\_\_\_\_\_\_\_\_\_\_\_ Ж. В. Михайличенко  подпись |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |

**ЗАДАНИЕ**

**на выполнение курсовой работы**

студенту Бекеша Кириллу Григорьевичу\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ фамилия, имя, отчество

по специальности \_\_\_09.02.07 Информационные системы и программирование \_\_\_\_\_\_\_\_\_\_\_\_\_

по междисциплинарному курсу \_\_Технология разработки программного обеспечения \_\_\_\_\_\_\_\_

1. Тема работы: \_\_\_Разработка программного обеспечения для исследования работы логических устройств компьютера\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Срок сдачи студентом работы «10» \_июня\_\_\_\_ 2024 г.
3. Цель и задачи работы \_\_Разработать программный продукт для реализации не менее пяти логических устройств компьютера, исследовать их работу\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Исходные данные к работе: \_\_Учебники и интернет-источники по технологии разработки программного обеспечения и архитектуре вычислительных систем, включающие описание функционирования логических компьютерных схем\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. Перечень вопросов, подлежащих разработке\_ а) изучить предметную область, выполнить анализ требований к программному обеспечению, составить техническое задание на разработку; б) выполнить проектирование системы с помощью CASE-средств; в) для решения поставленной задачи реализовать оконное приложение на языке C# и протестировать его; г) сформулировать предложения по внедрению, эксплуатации и сопровождению разработанного программного обеспечения. Сделать выводы по результатам проделанной работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. Перечень графического (иллюстративного) материала: таблицы, графики, рисунки, схемы, отражающие теоретический материал и программную реализацию поставленной задачи\_\_\_\_\_\_\_

Руководитель «19» \_февраля\_\_\_\_\_ 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Ж. В. Михайличенко

подпись инициалы, фамилия

Студент «19» \_февраля\_\_\_\_\_ 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_ К.Г. Бекеша\_\_\_\_\_\_

подпись инициалы, фамилия

**Аннотация**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

ОГУ 09.02.07. 3024. 083 ПЗ

Разраб.

Бекеша К. Г.

Провер.

Михайличенко Ж

Реценз.

Н. Контр.

Утверд.

Разработка программного обеспечения для исследования работы логических устройств компьютера

Лит.

Листов

\*

21ИСП-2

В курсовой работе по междисциплинарному курсу «Технология разработки программного обеспечения» профессионального модуля «Осуществление интеграции программных модулей» проведена разработка \*\*\*\*\*.

В первой главе курсовой работы \*\*\*

Во второй главе курсовой работе \*\*\*

В третьей главе курсовой работы \*\*\*

Пояснительная записка содержит \*\* страницы, в том числе \*\* рисунков, \*\* таблиц, \*\* источников, 1 приложение.

Разработка приложения выполнена \*\*\*.

**Содержание**

[Введение 5](#_Toc168497851)

[1 Анализ требований и проектирование программного продукта 6](#_Toc168497852)

[1.1 Анализ предметной области 6](#_Toc168497853)

[1.2 Техническое задание на разработку 13](#_Toc168497854)

[1.3 Построение моделей 17](#_Toc168497855)

[2 Реализация и тестирование программного продукта 19](#_Toc168497856)

[2.1 Обоснование программных средств реализации 19](#_Toc168497857)

[2.2 Разработка пользовательского интерфейса 20](#_Toc168497858)

[2.3 Алгоритмизация и программирование 21](#_Toc168497859)

[2.4 Тестирование 25](#_Toc168497860)

[3 Рекомендации по внедрению, эксплуатации и сопровождению программного обеспечению 26](#_Toc168497861)

[3.1 Руководство пользователя 26](#_Toc168497862)

[3.2 План внедрения и сопровождения 29](#_Toc168497863)

[Заключение 31](#_Toc168497864)

[Список использованных источников 32](#_Toc168497865)

[Приложение А 33](#_Toc168497866)

# Введение

Логическое устройства компьютера – это часть электронной схемы, которая выполняет элементарную логическую функцию. Логическая функция – это функция, которая принимает одно или несколько логических значений (обычно истинность или ложность) и возвращает логическое значение в зависимости от заданных условий. Я должен узнать какие существуют логические устройства, их виды и применение каждого логического устройства в компьютере.

Цифровой компьютер работает на базе двоичной системы счисления, где данные представлены в виде битов – минимальных единиц информации, которые могут принимать значения 0 или 1.

Знание логических устройств компьютера существенно для разработчиков программного обеспечения, поскольку позволяет им понимать, как устроена цифровая техника и как она выполняет различные операции. Понимание принципов работы логических устройств также позволяет оптимизировать процессы обработки данных в компьютерных системах и повысить их эффективность.

В целом, изучение логических устройств компьютера является важной частью образования в сфере информационных технологий, помогая специалистам разрабатывать и совершенствовать программное обеспечение для различных областей применения.

Цель курсовой работы:

Изучить принципы работы логического устройства, виды устройств, их логические схемы, таблицы истинности и создание программного обеспечения.

Для изучения поставленной цели можно выполнить следующие задачи:

Изучить основы двоичной системы счисления.

Изучить различные виды логических устройств и их принципы работы.

Изучить способы представления логических схем и таблиц истинности.

Освоить специализированные языки программирования для создания программного обеспечения для логических устройств.

Практиковаться в создании логических схем и таблиц истинности с использованием программного обеспечения.

# Анализ требований и проектирование программного продукта

## Анализ предметной области

Логическое устройства компьютера – это часть электронной схемы, которая выполняет элементарную логическую функцию. Логическая функция – это функция, которая принимает одно или несколько логических значений (обычно истинность или ложность) и возвращает логическое значение в зависимости от заданных условий. Я должен узнать какие существуют логические устройства, их виды и применение каждого логического устройства в компьютере.

К некоторым логическим устройствам относятся:

1. Шифратор — логическое устройство, выполняющее преобразование позиционного кода в n-разрядный двоичный код.

Таблица истинности устройства «Шифратор» показана на рисунке 1

Изображение выглядит как прямоугольный, число, текст, кроссворд

Автоматически созданное описание

Рисунок 1 – Таблица истинности устройства «Шифратор»

Логическая схема устройства «Шифратор» показана на рисунке 2.

Изображение выглядит как диаграмма, План, Технический чертеж, зарисовка

Автоматически созданное описание

Рисунок 2 – Логическая схема устройства «Шифратор»

1. Демультиплексор — это логическое устройство, предназначенное для переключения сигнала с одного информационного входа на один из информационных выходов. Таким образом, демультиплексор в функциональном отношении противоположен мультиплексору.

Таблица истинности устройства «Демультиплексор» показана на рисунке 3

Изображение выглядит как текст, число, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 – Таблица истинности устройства «Демультиплексор»

Логическая схема устройства «Демультиплексор» показана на рисунке 4.

Изображение выглядит как диаграмма, Технический чертеж, План, зарисовка

Автоматически созданное описание

Рисунок 4 – Логическая схема устройства «Демультиплексор»

1. Полусумматор — это логическая схема, способная складывать два одноразрядных двоичных числа

Таблица истинности устройства «Полусумматор» показана на рисунке 5

Изображение выглядит как число, текст, кроссворд

Автоматически созданное описание

Рисунок 5 – Таблица истинности устройства «Полусумматор»

Логическая схема устройства «Полусумматор» показана на рисунке 6.

Изображение выглядит как текст, Шрифт, снимок экрана, символ

Автоматически созданное описание

Рисунок 6 – Логическая схема устройства «Полусумматор»

1. **Сумматор** - узеларифметико-логическое устройство, посредством которого осуществляется суммирование чисел. Все математические операции в микропроцессорах сводятся к одной – к сложению.

Таблица истинности устройства «**Сумматор**» показана на рисунке 7

Изображение выглядит как текст, кроссворд, прямоугольный, число

Автоматически созданное описание

Рисунок 7 – Таблица истинности устройства «**Сумматор**»

Логическая схема устройства «**Сумматор**» показана на рисунке 8.

Изображение выглядит как текст, Шрифт, диаграмма, снимок экрана

Автоматически созданное описание

Рисунок 8 – Логическая схема устройства «**Сумматор**»

1. **Триггер** – это основной элемент электронной вычислительной машины устройство, имеющее два устойчивых состояния, 0 и 1.

Таблица истинности устройства «**Триггер**» показана на рисунке 9

Изображение выглядит как текст

Автоматически созданное описание со средним доверительным уровнем

Рисунок 9 – Таблица истинности устройства «**Триггер**»

Логическая схема устройства «**Триггер**» показана на рисунке 10.

Изображение выглядит как зарисовка, диаграмма, рисунок, линия

Автоматически созданное описание

Рисунок 10 – Логическая схема устройства «**Триггер**»

1. Инвертор - простейший логический элемент, выполняющий функцию отрицания. Если на вход поступает сигнал, соответствующий единице, то на выходе будет ноль. И наоборот.

Таблица истинности устройства «Инвертор» показана на рисунке 11

Изображение выглядит как снимок экрана, текст, число, прямоугольный

Автоматически созданное описание

Рисунок 11 – Таблица истинности устройства «Инвертор»

Логическая схема устройства «Инвертор» показана на рисунке 12.

Изображение выглядит как зарисовка, диаграмма, линия, дизайн

Автоматически созданное описание

Рисунок 12 – Логическая схема устройства «Инвертор»

1. Дизъюнктор - выдает на выходе значение логической суммы входных сигналов. Он имеет один выход и не менее двух входов.

Таблица истинности устройства «Дизъюнктор» показана на рисунке 13

Изображение выглядит как прямоугольный, число, линия, текст

Автоматически созданное описание

Рисунок 13 – Таблица истинности устройства «Дизъюнктор»

Логическая схема устройства «Дизъюнктор» показана на рисунке 14.

Изображение выглядит как текст, Шрифт, линия, диаграмма

Автоматически созданное описание

Рисунок 14 – Логическая схема устройства «Дизъюнктор»

1. Логическое умножение (Коньюктор) - операция, соединяющая два или более высказываний при помощи логической связки или. Результат операции может быть истинным только в том случае, если одновременно истинны исходные высказывания

Таблица истинности устройства «Коньюктор» показана на рисунке 15

Изображение выглядит как линия, белый

Автоматически созданное описание

Рисунок 15 – Таблица истинности устройства «Коньюктор»

Логическая схема устройства «Коньюктор» показана на рисунке 16.

Изображение выглядит как диаграмма, Шрифт, текст, линия

Автоматически созданное описание

Рисунок 16 – Логическая схема устройства «Коньюктор»

1. Дешифратор — это комбинационное логическое устройство, которое предназначено для преобразования двойного двоичного кода в необходимый сигнал управления в какой-либо системе исчисления на одном из выходов.

Таблица истинности устройства «Дешифратор» показана на рисунке 17

Изображение выглядит как текст, число, Шрифт, линия

Автоматически созданное описание

Рисунок 17 – Таблица истинности устройства «Дешифратор»

Логическая схема устройства «Дешифратор» показана на рисунке 18.

Изображение выглядит как диаграмма, Технический чертеж, План, текст

Автоматически созданное описание

Рисунок 18 – Логическая схема устройства «Дешифратор»

1. Мультиплексор - устройство, имеющее несколько сигнальных входов, один или более управляющих входов и один выход. Мультиплексор позволяет передавать сигнал с одного из входов на выход; при этом выбор желаемого входа осуществляется подачей соответствующей комбинации управляющих сигналов.

Таблица истинности устройства «Мультиплексор» показана на рисунке 19

Изображение выглядит как текст, число, кроссворд

Автоматически созданное описание

Рисунок 19 – Таблица истинности устройства «Мультиплексор»

Логическая схема устройства «Мультиплексор» показана на рисунке 20.

Изображение выглядит как текст, диаграмма

Автоматически созданное описание

Рисунок 20 – Логическая схема устройства «Мультиплексор»

В программе «Логические устройства компьютера» ставятся следующие задачи:

* Ознакомление с логическими устройствами: в рамках этой задачи требуется изучить различные логические устройства, их таблицы истинности и логические схемы.
* Проверка работы устройства: в рамках этой задачи требуется проверить работу конкретного устройства. Для этого пользователю необходимо ввести входные данные и проверить правильность вывода выходных данных. При этом выбор желаемого входа на логическое устройство осуществляется подачей соответствующей комбинации, вписанной в программе.

## 1.2 Техническое задание на разработку

Техническое задание на разработку программного обеспечения – документ, который содержит подробное содержание требований к программному продукту. Техническое задание на разработку программного обеспечении для исследования работы логических устройств компьютера составлена согласно ГОСТ 34.602–2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной систем» и представлено ниже.

1 Общие сведения

а) Название автоматизированной системы (АС): «Логические устройства компьютера»

б) Наименование заказчика: факультет среднего профессионального образования Орского гуманитарно-технологического института (филиала ОГУ) в лице преподавателя высшей категории Михайличенко Ж. В.;

в) Наименование разработчика: студент второго курса группы 22ИСП-1 Бекеша К.Г.;

г) Документ, на основании которого создаётся АС: протокол закрепления тем курсовой работы по дисциплине «Технология разработки программного обеспечения» от 19.02.2024 г.;

д) Дата начало работ: 19.02.2024;

е) Дата окончания работ: 10.06.2024;

2 Цели и назначение создания АС;

а) Цели создания автоматизированной системы:

* изучение принципа работы логического устройства, виды логических устройств, их логические схемы, таблицы истинности и создание программного обеспечения.

б) Назначение автоматизированной системы:

* Умение работать с логическими устройствами

3 Характеристика объекта автоматизации.

Объектом автоматизации является работа логических устройств компьютера.

4 Требования к АС.

а) Требования к функциям, выполняемым АС:

* Выполнение команд и операций
* Защита данных от неверного ввода
* Управление вводом

б) Требования к видам обеспечения АС.

* Требования к математическому обеспечению: должно быть способно выполнять математические операции, обеспечивать точность вычислений и обрабатывать данные в соответствии с заданными алгоритмами.
* Требования к информационному обеспечению: приложение должно быть создано для работы на операционной системе Windows. Документация по игре, инструкции для пользователей также должны быть представлены в удобной форме.
* Требование к лингвистическому обеспечению: интерфейс системы должен быть русскоязычным, общение с пользователем реализуется с помощью диалоговых окон.
  + - Требование к программному обеспечению: создаваемая система должна быть совместима с операционной системой Windows, а также интегрирована с языком программирования C#. Использование операционной системы Windows, программирование на языке C# с использованием интегрированной среды разработки Microsoft Visual Studio 2019, Microsoft Word для документирования; а для документирования следует использовать Microsoft Word средство построения бизнес-процессов Ramus Educational, приложение для быстрого создания файлов справки Dr.Explain;
* Требования к техническому обеспечению: система должна быть легкой и масштабируемой, чтобы работать на различных конфигурациях ПК. Это важно для обеспечения доступности и удобства использования для широкого круга пользователей. Также это позволяет снизить затраты на оборудование и обеспечить более широкое распространение системы.
* Требования к организационному обеспечению: взаимодействие с системой будет осуществляться пользователем, который может вводить исходные данные и получать результаты путём вывода на экран монитора или записи в файл. Таким образом, система будет обеспечивать удобство использования и гибкость в обработке данных для пользователя.

в) Общие технические требования к АС.

* Требования к численности и квалификации пользователей: система предназначена для одного пользователя и должна быть проста в эксплуатации для человека;
* Требования к эргономике технической эстетике: лёгкое управление всеми функциями и понятный, удобный интерфейс для пользователей.
* Требования к защите информации: написать

5 Состав и содержание работ по созданию АС;

В таблице 1 показаны этапы разработки программного обеспечении для исследования работы логических устройств компьютера.

Таблица 1 – Этапы разработки АС

|  |  |  |
| --- | --- | --- |
| Номер и название этапа разработки | Сроки выполнения | Содержание |
| 1 Анализ требований | 19.02.2024 – 05.03.2024 | Анализ предметной области, изучение программных аналогов, разработка технического задания на создание АС |
| 2 Проектирование | 06.03.2024 – 20.03.2024 | Разработка диаграмм IDEF различного уровня, диаграмм вариантов использования. Выбор и обоснование технологий и инструментов разработки. Проектирование пользовательского интерфейса. |
| 3 Программирование | 21.03.2024 – 14.05.2024 | Разработка и программная реализация алгоритмов и методов ввода, хранения, обработки и вывода данных. Разработка пользовательского интерфейса, организация диалога с пользователем. Отладка работы системы. Документирование программного кода. |
| 4 Тестирование | 15.05.2024 – | Тестирование кода, поиск и исправление ошибок. |
| 5 Внедрение | 01.06.2024 –  10.06.2024 | Система разворачивается на реальном оборудовании и начинает использоваться. |
| 6 Эксплуатация и сопровождение | с 10.06.2024 | Поддержка стабильной работы системы. Внесение изменений и улучшений. |

6 Порядок разработки АC.

Этапы должны выполняться разработчиком в прямой последовательности в соответствии с приведенной в пункте пять таблицей этапов разработки АС с обязательным контролем и консультацией с заказчиком.

1. Порядок контроля и приёмки АС.

Контроль за разработкой осуществляется заказчиком и принимается в указанные сроки – 10.06.2024.

1. Требования к составу и содержанию подготовки объекта автоматизации к вводу АС в действие

* установка программного обеспечения: разработчик устанавливает программное обеспечение компьютере пользователя в соответствии с требованиями системы;
* тестирование и отладка: после установки и настройки системы проводится тестирование на работоспособность, отладка возможных ошибок и сбоев;
* поддержка и сопровождение: после ввода системы в действие разработчик оказывает поддержку и сопровождение для решения возможных проблем и вопросов пользователей.

9 Требования к документированию.

Вместе с программным продуктом должна поставляться следующая документация:

* документирование программного кода (комментарии)
* техническое задание
* пояснительная записка к курсовой работе;
* руководство пользователя

10 Источники разработки.

* Протокол закрепления тем курсовой работы по дисциплине «Технология разработки программного обеспечения» от 19.02.2024 г.;
* ГОСТ 34.602–2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы»;
* Работы студенческие. Общие требования и правила оформления. СТО 02069024.101–2015. – Оренбург: Изд-во ОГУ, 2015. – 89 с.;
* Добавить гост на блок схемы
* Добавить гост на стадии разработки АС

## Построение моделей

Функциональная модель является необходимым инструментом на этапе проектирования ПО для обеспечения эффективной и качественной разработки программного продукта.

Для более наглядного представления этапов работы и понимания производственных функций информационной системы «Логические устройства компьютера», разработана функциональная модель с использованием методологии IDEF0 в программном инструменте Ramus Educatio, который предназначен для анализа, моделирования и разработки систем.

IDEF0 - это метод, который используется для функционального моделирования в системном анализе и проектировании. Он помогает описывать и анализировать функции системы, их взаимосвязи и взаимодействие.

Для начала построения модели мы создали контекстную диаграмму. Входными данными будут тип логического устройства, теория о его структуре и функционировании, графическое представление логического устройства, а также таблица истинности для его работы. Управление работой информационной системы будет осуществляться алгоритмом функционирования логических устройств. Пользователь и персональный компьютер будут выступать в роли механизмов, а результатом работы системы будут результаты исследования логических устройств.

Контекстная диаграмма IDEF0 представлена на рисунке 21.

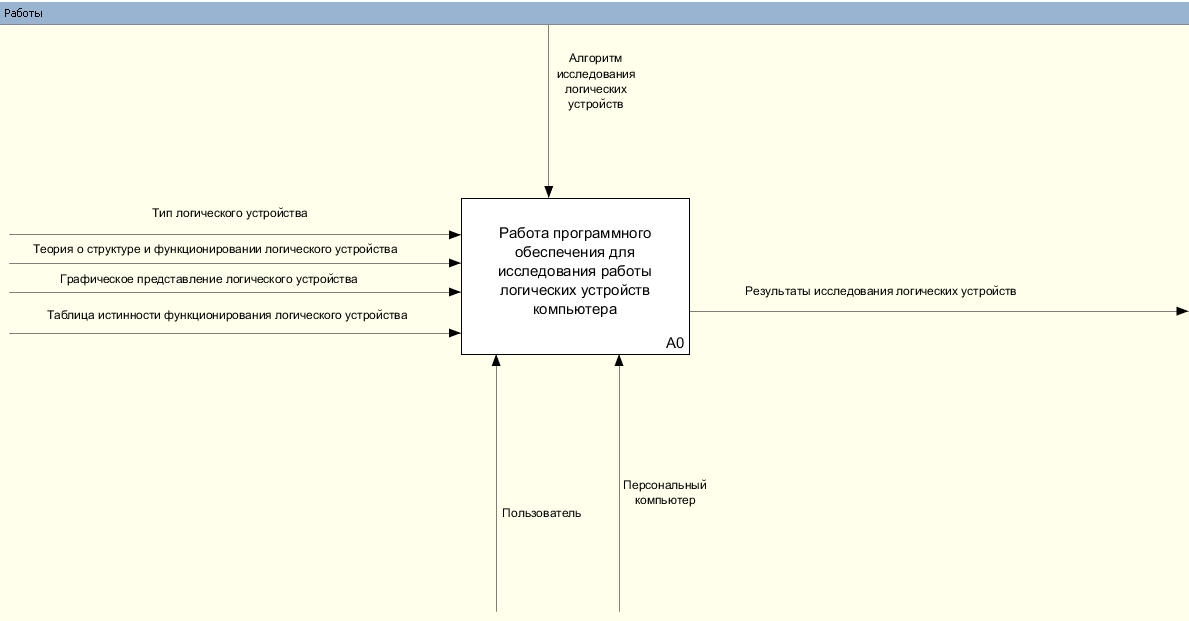


Рисунок 21 – Контекстная диаграмма IDEF0

Для более детального изучения компонентов системы можно определить следующие дополнительные функции:

A1 «Выбор типа логического устройства» входными данными этой функции является тип логического устройства, механизмами - пользователь и ПК, а выходом – Выбранный тип логического устройства.

A2 «Вывод основных характеристик логического устройства» входными данными этой функции является теория о структуре и функционировании логического устройства, механизмами - пользователь и ПК, а выходом – Характеристика логического устройства.

A3 «Вывод таблицы истинности» входными данными этой функции является таблица истинности функционирования логического устройства, механизмами - пользователь и ПК, а выходом – таблица истинности.

А4 «Анализ результатов работы логических устройств» входными данными этой функции является графическое представление логического устройства, механизмами - пользователь и ПК, а выходом – результаты исследования логических устройств.

Контекстная диаграмма IDEF0 первого уровня представлена на рисунке 22.

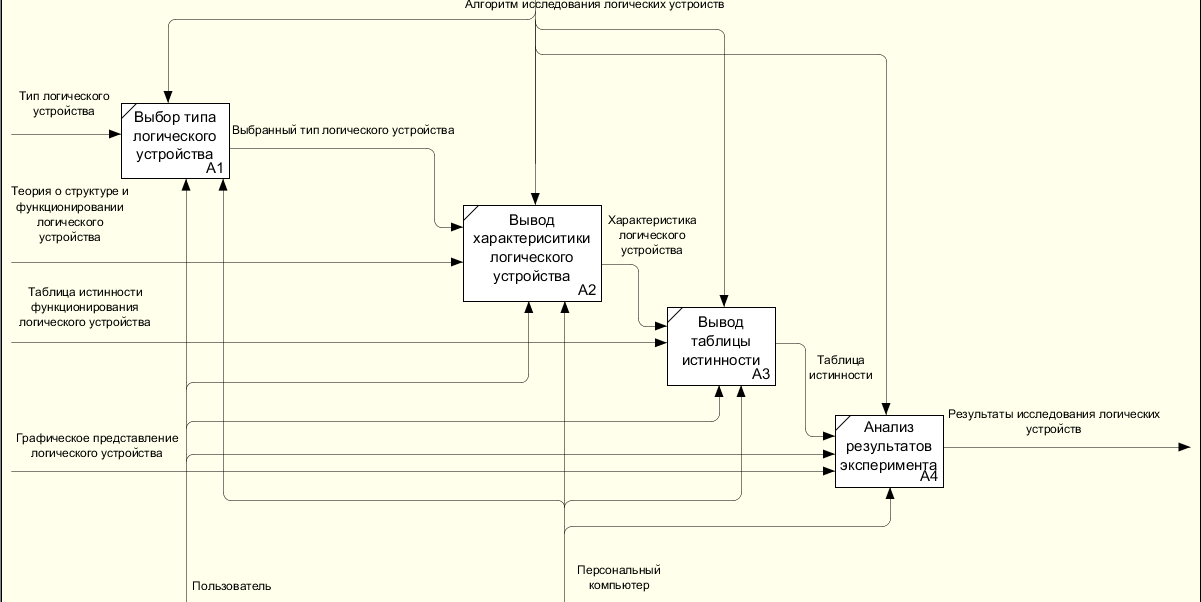


Рисунок 22 – Контекстная диаграмма IDEF0 первого уровня

Все построенные функциональные диаграммы будут использованы при разработке программного обеспечения в системе «Логические устройства компьютера»

# Реализация и тестирование программного продукта

## Обоснование программных средств реализации

Для разработки автоматизированной системы «Логические устройства компьютера» решено использовать следующие программные средства: язык программирования C# и интегрированную среду разработки Visual Studio 2019.

Язык программирования C# является объектно-ориентированным и обладает простотой использования, высокой производительностью и широкими возможностями для создания программ. Он поддерживает объектно-ориентированное программирование, что упрощает разработку сложных систем. C# широко применяется в различных областях, таких как веб-разработка, мобильные приложения и игры.

Язык C# имеет простой и понятный синтаксис, что делает его легким для изучения и быстрой разработки приложений. Он автоматически управляет выделением и освобождением памяти для объектов, что упрощает работу с памятью. C# предоставляет необходимые конструкции для создания объектно-ориентированных приложений и обеспечивает защиту от несанкционированных действий.

Благодаря интеграции с .NET-платформой C# позволяет использовать библиотеки и компоненты, созданные на других языках .NET, что упрощает разработку приложений и обеспечивает совместимость с другими технологиями на платформе.

Интегрированная среда разработки Visual Studio 2019 предоставляет разработчикам мощные инструменты для создания программ на платформе .NET. Она содержит редактор кода, отладчик, дизайнер пользовательского интерфейса и другие инструменты разработки. Visual Studio 2019 поддерживает различные языки программирования, интеграцию с платформами Windows API, Windows Forms, WPF и позволяет создавать как управляемый, так и машинный код.

Visual Studio 2019 обладает широкой поддержкой различных языков программирования, интеграцией с платформами Windows API, Windows Forms, WPF, а также возможностью создания как управляемого, так и машинного кода. Эта среда обеспечивает удобство и эффективность при разработке программных продуктов, а также удобную интеграцию с другими сервисами и инструментами разработки. Visual Studio 2019 также предлагает мощный отладчик, широкий набор инструментов и удобную интеграцию с платформой .NET.

Вместе C# и Visual Studio предоставляют мощные и гибкие инструменты для разработки программного обеспечения. Учитывая все преимущества данного языка программирования C# и интегрированной среды разработки Visual Studio, выбор данного программного обеспечения вполне обоснован для разработки создания автоматизированной системы «Логические устройства компьютера».

## Разработка пользовательского интерфейса

Пользовательский интерфейс — это средство, через которое пользователи взаимодействуют с программным обеспечением. Программный интерфейс — это набор функций, процедур и протоколов, которые позволяют одной программе взаимодействовать с другой. Пользовательский интерфейс в программном обеспечении играет ключевую роль в опыте пользователей и влияет на их удовлетворенность и эффективность при работе с приложением. Хорошо спроектированный интерфейс упрощает взаимодействие пользователя с программой, делая процессы более интуитивными и эффективными.

Необходимые требования к интерфейсу:

1 Простота использования: Программа должна быть лёгкой в освоении даже для новичков. Это достигается через интуитивное размещение элементов управления, понятные иконки и легкость в навигации. Чем проще и понятнее интерфейс, тем быстрее пользователи смогут освоить программу и начать эффективно ею пользоваться.

2 Скорость работы: Быстрая отзывчивость интерфейса на действия пользователя является ключевым аспектом. Переход по формам должен осуществляться без задержек. Хорошо оптимизированный интерфейс обеспечивает плавную навигацию и мгновенную реакцию на команды пользователя.

3 Чёткое отображение информации: Читаемый текст, достаточно крупные элементы интерфейса, приятная цветовая гамма и логичное размещение информации помогают пользователям быстро ориентироваться в программе и находить необходимую информацию без лишних усилий.

4 Эргономика: Удобство использования также играет важную роль. Это включает в себя удобство расположения элементов интерфейса, удобство взаимодействия с ними и общую эстетику интерфейса.

Хорошо спроектированный пользовательский интерфейс способствует повышению удовлетворенности пользователей, делает работу с программой более эффективной и приятной.

Программный интерфейс автоматизированной системы «Логические устройства компьютера», должен соответствовать всем вышеперечисленным требованиям. Структура интерфейса и его компоненты показаны на рисунке 23.

Интерфейс приложения организован в виде единого окна. Панель видов логических устройств содержит 10 кнопок (элемент radioButton), каждая из которых представляет логическое устройство. При выборе логического устройства отображается следующее:

1 Описание выбранного устройства в элементе richTextBox

2 Схема устройства в pictureBox

3 Таблица истинности устройства в pictureBox

Кроме того, на главной форме предусмотрена функция проверки работы устройства. Пользователь должен ввести данные: 1 или 0 в текстовые поля. При нажатии кнопки «Проверить» в окне «Выходные данные» выводится выходные данные устройства.

Изображение выглядит как текст, снимок экрана, программное обеспечение, электроника

Автоматически созданное описание

Рисунок 23 – Структура интерфейса и его компоненты

## Алгоритмизация и программирование

Алгоритм – четкое описание последовательности действий, которые необходимо выполнить для решения поставленной задачи.

Алгоритм обладает основными свойствами:

1 Конечность: определяет возможность получения результата за конечное число шагов.

2 Дискретность: любой процесс алгоритма можно разбить на определенные этапы или шаги.

3Опредёленность: каждое правило алгоритма должно быть четким и однозначным, а также выполнится в заданной последовательности.

4 Массовость: получение алгоритма при любых исходных данных

Алгоритмы удобно представлять в виде блок схем, согласно ГОСТ 19.701–90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначение условные и правила выполнения. Для автоматизированной системы «Логические устройства компьютера» блок-схема алгоритма показана на рисунках 24,2 5.

Изображение выглядит как текст, диаграмма, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 24 – Блок-схема алгоритма проверки работы дизъюнктора

Алгоритм по блок-схеме, приведенной на рисунке 24:

1 Выбрать логическую операцию: дизъюнкция.

2 Вывести описание, схему и таблицу истинности дизъюнктора.

3 Проверить работу устройства:

* Если входные данные равны 0 и 0, выводится выходные данные 0.
* Если это условие не выполняется, то выводится 1

Изображение выглядит как текст, диаграмма, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 25 – Блок-схема алгоритма проверки работы инвертора

Алгоритм по блок-схеме, приведенной на рисунке 25:

1 Выбрать логическую операцию: инвертор.

2 Вывести описание, схему и таблицу истинности инвертора.

3 Проверить работу устройства:

4 Если входные данные равны 0, вывести выходные данные, если нет, то вывести выходные данные по данным 1.

Для реализации функции системы «Логические устройства компьютера», разработан следующие методы:

* private void radioButton1\_CheckedChanged (object sender, EventArgs e): этот метод используется для обработки события изменения состояния переключателя с именем radioButton1. Когда пользователь выбирает этот переключатель, система отображает информацию о шифраторе, включает соответствующие текстовые поля и изображения.
* private void radioButton2\_CheckedChanged (object sender, EventArgs e): этот метод используется для обработки события изменения состояния переключателя с именем radioButton2. Когда пользователь выбирает этот переключатель, система отображает информацию о демультиплексоре, включает соответствующие текстовые поля и изображения.
* private void radioButton3\_CheckedChanged (object sender, EventArgs e): этот метод используется для обработки события изменения состояния переключателя с именем radioButton3. Когда пользователь выбирает этот переключатель, система отображает информацию о полусумматоре, включает соответствующие текстовые поля и изображения.
* private void radioButton4\_CheckedChanged (object sender, EventArgs e): этот метод используется для обработки события изменения состояния переключателя с именем radioButton4. Когда пользователь выбирает этот переключатель, система отображает информацию о сумматоре, включает соответствующие текстовые поля и изображения.
* private void radioButton5\_CheckedChanged (object sender, EventArgs e): этот метод используется для обработки события изменения состояния переключателя с именем radioButton5. Когда пользователь выбирает этот переключатель, система отображает информацию о триггере, включает соответствующие текстовые поля и изображения.
* private void radioButton6\_CheckedChanged (object sender, EventArgs e): этот метод используется для обработки события изменения состояния переключателя с именем radioButton6. Когда пользователь выбирает этот переключатель, система отображает информацию о инверторе, включает соответствующие текстовые поля и изображения.
* private void radioButton7\_CheckedChanged (object sender, EventArgs e): этот метод используется для обработки события изменения состояния переключателя с именем radioButton7. Когда пользователь выбирает этот переключатель, система отображает информацию о дизъюнкторе, включает соответствующие текстовые поля и изображения.
* private void radioButton8\_CheckedChanged (object sender, EventArgs e): этот метод используется для обработки события изменения состояния переключателя с именем radioButton8. Когда пользователь выбирает этот переключатель, система отображает информацию о конъюкторе, включает соответствующие текстовые поля и изображения.
* private void radioButton9\_CheckedChanged (object sender, EventArgs e): этот метод используется для обработки события изменения состояния переключателя с именем radioButton9. Когда пользователь выбирает этот переключатель, система отображает информацию о дешифраторе, включает соответствующие текстовые поля и изображения.
* private void radioButton10\_CheckedChanged (object sender, EventArgs e): этот метод используется для обработки события изменения состояния переключателя с именем radioButton10. Когда пользователь выбирает этот переключатель, система отображает информацию о мультиплексоре, включает соответствующие текстовые поля и изображения.
* private void button1\_Click\_1(object sender, EventArgs e): этот метод button1\_Click\_1 обрабатывает событие нажатия кнопки. Он содержит логику для различных условий выбора переключателей (radioButtons) в зависимости от того, какие переключатели были выбраны, и основываясь на этих условиях, он устанавливает текст в textBox1. Таким образом, он реагирует на изменения состояний переключателей и обновляет отображаемую информацию соответственно.
* private void button2\_Click (object sender, EventArgs e) обрабатывает событие нажатия кнопки, и очищает текстовые поля.
* private void textBox2\_KeyPress (object sender, KeyPressEventArgs e) защищает текстовое поле от некорректных данных.
* private void textBox3\_KeyPress (object sender, KeyPressEventArgs e) защищает текстовое поле от некорректных данных.
* private void textBox4\_KeyPress (object sender, KeyPressEventArgs e) защищает текстовое поле от некорректных данных.
* private void textBox5\_KeyPress (object sender, KeyPressEventArgs e) защищает текстовое поле от некорректных данных.
* private void textBox1\_KeyPress (object sender, KeyPressEventArgs e) защищает текстовое поле от ввода каких либо данных.

## Тестирование

# Рекомендации по внедрению, эксплуатации и сопровождению программного обеспечению

## Руководство пользователя

Для установки программы, для пользователя было создано руководство пользователя, перечисленное ниже:

Системные требования

Для стабильной и эфективной работы программы логические устройства компьютера рекомендуется использовать следующую конфигурацию:

Частота процессора (CPU): <1> GHz

Количество ядер процессора (CPU): < 2 Ядра>

Объём оперативной памяти (RAM): <2>GB

Объём свободного места на диске(HDD): <100>MB

Операционная система(OS):<Windows>

Браузер: Google Chrome, Mozilla Firefox, Microsoft Edge, Яндекс Браузер

Начало работы

Данный раздел поможет вам быстро установить, настроить и начать работать с программой логические устройства компьютера.

Перед началом работы, пожалуйста, ознакомьтесь с системными требованиями и лицензионным соглашением.

Содержание Раздела

1. Основные понятия
2. Установка
3. Настройка
4. Запуск

Основные понятия и термины

Перед началом работы в программе логические устройства компьютера рекомендуем ознакомиться с основными понятиями и терминами:

Компьютер — функциональное устройство, способное выполнять значительный объём вычислений, включая многочисленные арифметические и логические операции, без вмешательства человека. Компьютер может быть как отдельным блоком, так и состоять из нескольких взаимосвязанных устройств.

С# — объектно-ориентированный язык программирования общего назначения. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core.

Программа — это набор инструкций, написанных на понятном для компьютера языке, которые указывают компьютеру, что он должен делать. Программы помогают компьютеру выполнять различные задачи, такие как:

Обработка текста, данных и изображений

— Расчёты и анализ информации

— управление различными устройствами

— Представление доступа к различным приложениям и сервисам

— игры и развлечения.

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов.

**Сумматор**— узел арифметико-логическое устройство, посредством которого осуществляется суммирование чисел. Все математические операции в микропроцессорах сводятся к одной – к сложению.

**Триггер** – это основной элемент электронной вычислительной машины устройство, имеющее два устойчивых состояния, 0 и 1

Инвертор — простейший логический элемент, выполняющий функцию отрицания. Если на вход поступает сигнал, соответствующий единице, то на выходе будет ноль. И наоборот.

Дизъюнктор — выдает на выходе значение логической суммы входных сигналов. Он имеет один выход и не менее двух входов.

Логическое умножение (конъюнкция) - операция, соединяющая два или более высказываний при помощи логической связки или. Результат операции может быть истинным только в том случае, если одновременно истинны исходные высказывания

Дешифратор — это комбинационное логическое устройство, которое предназначено для преобразования двойного двоичного кода в необходимый сигнал управления в какой-либо системе исчисления на одном из выходов.

Установка

Для установки программы, пожалуйста, загрузите дистрибутива последней версии программы, доступный по адресу <https://www.product.com/download/>

Перед установкой ознакомьтесь с [системными требованиями](file:///C:\Users\kbeke\AppData\Local\Temp\0001Preview\system_requirements.htm) и лицензионным соглашением.

В процессе установки, пожалуйста, не выключайте программу установки.

Запуск

Для запуска программы логические устройства компьютера нажмите на ярлык программы на рабочем столе, в меню Пуск либо наберите в командной строке «Логические устройства компьютера».

Пользовательский интерфейс

Этот раздел описывает основные элементы пользовательского интерфейса программы логические устройства компьютера: основных режимов работы, предназначение окон и экранов, доступные операции.

Содержание раздела:

1 Главное окно программы

2 Режим работы

3 Горячие клавиши

Главное окно программы

Главное окно программы логические устройства компьютера позволяет выполнять следующие операции:

* Выбор логического устройства
* Вывод описания, схемы и таблицы истинности логического устройства
* Проверка работы логического устройства
* Кнопка Отчистить, для удаления записей в полях
* Кнопка посчитать для вывода выходных данных
* Поля для записей входных данных
* Выход из приложения
* Свернуть приложение

Главное окно программы показано на рисунке 1.

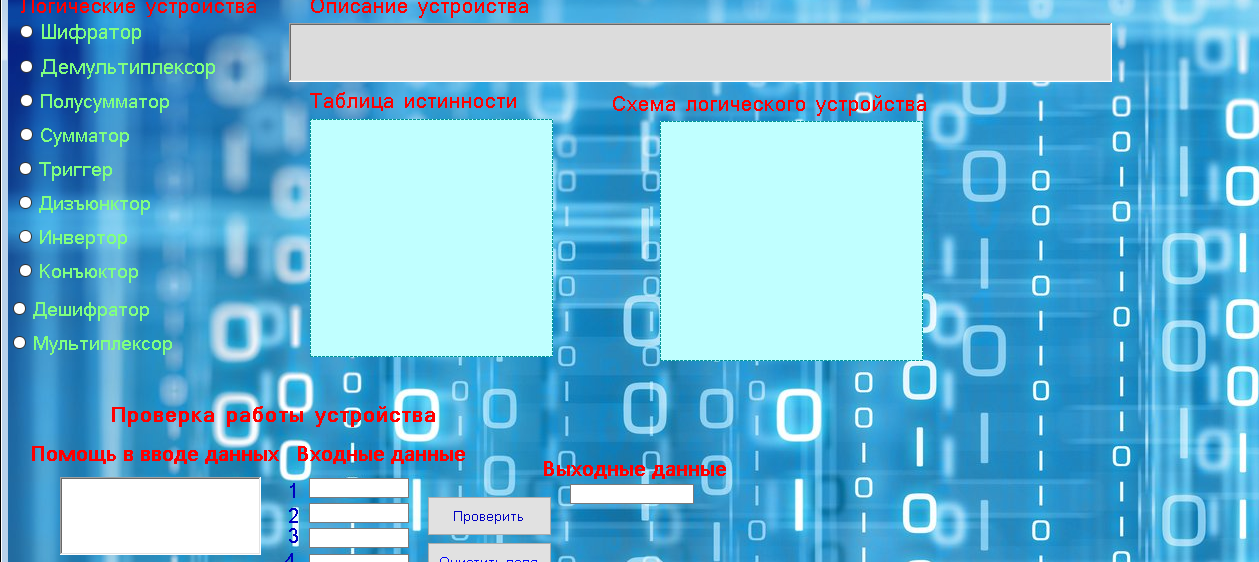


Рисунок 1 – Главное окно программы

Режим работы

Пользовательский интерфейс программы логические устройства компьютера обеспечивает работу только в режиме пользователя.

Пользователю доступны все права.

Горячие клавиши

Следующий раздел содержит все сочетания клавиш и способы управления при помощи мыши, поддерживаемые программой логические устройства компьютера

Общие

F1 – показать контекстную справку

Ctrl+N – создать новый проект

Ctrl+O – открыть проект

Ctrl+S – сохранить открытый текст

Поиск

Ctrl+F – найти

Ctrl+H – найти и заменить

F3 – найти следующее

Редактирование

Alt+BackSpace – отменить

Shift+Delete – вырезать

Shift+Insert – вставить

Ctrl+C – копировать

Ctrl+Insert – копировать

Ctrl+K – вырезать

Ctrl+V – вставить

Частые вопросы (FAQ)

Как запустить приложение?

Ответ: чтобы запустить приложение найдите его в меню Пуск или введите Логические устройства компьютера в поиске.

Для чего нужна это программа?

Ответ - для удобного исследование логического устройства.

Если вы не нашли ответ на свой вопрос, пожалуйста, свяжитесь с нами.

Контактная информация

Программа «Логические устройства компьютера» разрабатывается одиночным разработчиком, являющейся правообладателем.

Сайт продукта

У продукта нет официального сайта.

Техническая поддержка

Вы можете направить вопрос по функциональности программы логические устройства следующими способами:

* Email: kbekesha@mail.ru
* Телефон: +7951035870

Продажи и сотрудничество

По вопросам лицензирования и сотрудничества, пожалуйста используйте следующие каналы:

* Email: kbekesha@mail.ru
* Телефон: +7951035870

Устранение типовых проблем

Не запускается программа

Решение: если у вас не запускается программа, попробуйте перезапустить ее или перезагрузить систему.

Программа зависла

Решение: если у вас зависла программа, попробуйте перезапустить ее.

Если вы не нашли ответа на свой вопрос, пожалуйста, свяжитесь с нами.

## План внедрения и сопровождения

Текст

# Заключение

На 1 страницу, подвести итоги проделанной работы, что я разработал что я оформил чему я научился и т.д

# Список использованных источников

1. Работы студенческие. Общие требования и правила оформления. СТО 02069024.101 – 2015. – Оренбург: Изд-во ОГУ, 2015. – 89 с.
2. 7–10 источников, не меньше, в алфавитном порядке
3. [Скачать ГОСТ 19.701–90 Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения (susu.ru)](https://is20-2019.susu.ru/wp-content/uploads/2021/04/gost_19.701-90.pdf)
4. [Алгоритм - что это такое: виды и типы алгоритмов, применение (skillfactory.ru)](https://blog.skillfactory.ru/glossary/algoritm/?ysclid=lwojnqbuph320391839)
5. [Основные логические устройства компьютера — Студопедия.Нет (studopedia.net)](https://studopedia.net/15_87662_osnovnie-logicheskie-ustroystva-kompyutera.html?ysclid=lugmwm89fj623622924)
6. [IDEF0 — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/IDEF0)
7. Программное обеспечение, которое является компьютерным инструментом для анализа, моделирования, разработки системы - Ramus Education
8. [9.7. Дешифратор. Устройство, принцип работы (studfile.net)](https://studfile.net/preview/5851753/page:42/)
9. [2.3 Мультиплексор (studfile.net)](https://studfile.net/preview/8984582/page:4/)

# Приложение А

(обязательное)

**Текст программы**

Размер 10