

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**"МИРЭА - Российский технологический университет"**

# РТУ МИРЭА

Институт искусственного интеллекта

Кафедра общей информатики

**ОТЧЕТ**

**ПОПРАКТИЧЕСКОЙРАБОТЕ№ 7**

Реализация заданной логической функции от четырех переменных на дешифраторах 4-16, 3-8 и 2-4

**по дисциплине**

**«**ИНФОРМАТИКА**»**

|  |  |
| --- | --- |
| Выполнил студент группы *ИКБО-09-22* | *Гришин А. В.* |
| Принял  *Старший преподаватель кафедры ОИ* | *Смирнов С. С.* |
| Практическая «\_\_»\_\_\_\_\_\_\_\_\_\_2022 г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| работа выполнена |  |
| «Зачтено» «\_\_»\_\_\_\_\_\_\_\_\_\_2022 г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Москва 2022

# СОДЕРЖАНИЕ

[1 ПОСТАНОВКА ЗАДАЧИ НА ПРАКТИЧЕСКУЮ РАБОТУ 3](#_Toc118642200)

[2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ 4](#_Toc118642201)

[2.1 Построение таблицы истинности 4](#_Toc118642202)

[2.2 Реализация функции, используя дешифратор 4-16 и одну дополнительную схему «или» 5](#_Toc118642203)

[2.3 Реализация функции, используя два дешифратора 3-8 и необходимую дополнительную 6](#_Toc118642204)

[2.4 Реализация функции, используя пять дешифраторов 2-4 и одну дополнительную схему «или» 9](#_Toc118642205)

3 [ВЫВОДЫ 12](#_Toc118642206)

4 [СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 13](#_Toc118642207)

# 1 ПОСТАНОВКА ЗАДАЧИ НА ПРАКТИЧЕСКУЮ РАБОТУ

Логическая функция от четырех переменных задана в 16-теричной векторной форме. Восстановить таблицу истинности. По таблице истинности реализовать в лабораторном комплексе логическую функцию на дешифраторах тремя способами:

– используя дешифратор 4-16 и одну дополнительную схему «или»;

– используя два дешифратора 3-8 и необходимую дополнительную логику;

– используя пять дешифраторов 2-4 и одну дополнительную схему «или».

Протестировать работу схем и убедиться в правильности их работы. Подготовить отчет о проделанной работе и защитить её.

F (a, b, c, d) = FAD216

# 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

# Построение таблицы истинности

Функция, заданная в 16-теричной форме имеет следующий вид: F (a, b, c, d) = FAD216

Преобразуем ее в двоичную запись: 1111 1010 1101 00102 – получили столбец значений логической функции, который необходим для восстановления полной таблицы истинности (см. табл.1).

Таблица 1 – Таблица истинности логической функции F

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | c | d | F |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

# Реализация функции, используя дешифратор 4-16 и одну дополнительную схему «или»

Реализуем функцию, используя дешифратор 4-16 и одну дополнительную схему «или». Количество выходов дешифратора соответствует количеству значений логической функции, поэтому требуется только один такой дешифратор. Подадим значения переменных функции на адресные входы дешифратора: младшую переменную «d» – на младший адресный вход, старшую переменную «a» – на старший адресный вход, прочие переменные – аналогично (на схеме далее переменные подаются на адресные входы дешифратора при помощи шины). В процессе работы на выходах дешифратора (с нулевого по пятнадцатый) будут последовательно возникать единичные значения в соответствии с поступающей на адресные входы комбинацией значений переменных. Выберем лишь те выходы дешифратора, номера которых совпадают с номерами наборов значений переменных, на которых функция равна единице. Объединим эти выходы дешифратора через «или» и получим требуемую реализацию (рис. 1).

Сразу после добавления дешифратора на рабочую область необходимо настроить ему некоторые свойства:

– «выбирающие биты» (адресные входы) установить равными 4;

– «три состояния» — нет;

– «на отключенном выходе» — установить равным 0;

– «разрешающий вход» — нет.

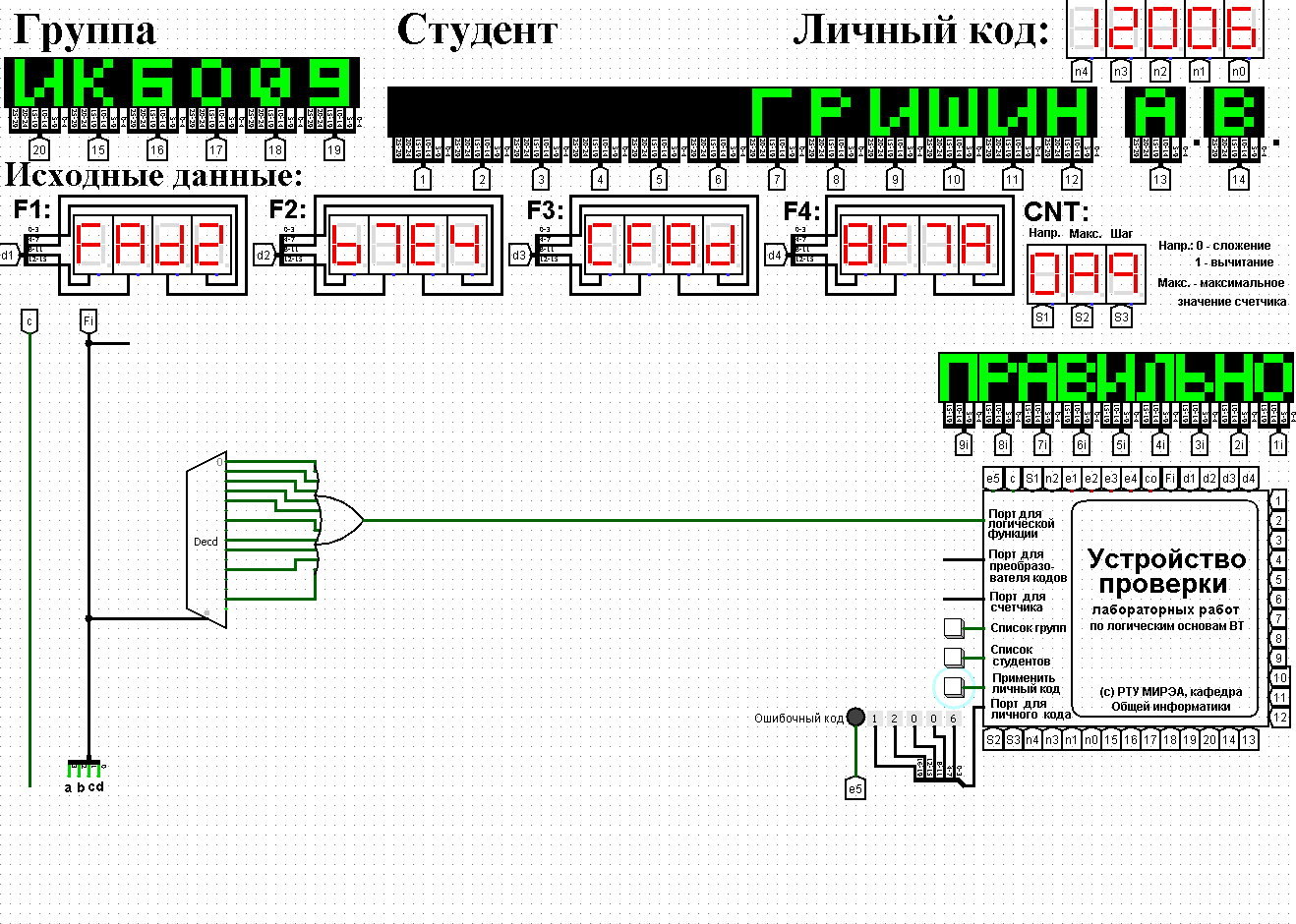


Рисунок 1 – Тестирование схемы,

реализующей логическую функцию на дешифраторе 4-16

Тестирование показало, что схема работает правильно.

# Реализация функции, используя два дешифратора 3-8 и необходимую дополнительную логику

Реализуем функцию, используя дешифраторы 3-8 и необходимую дополнительную логику. Количество выходов у дешифратора 3-8 в два раза меньше количества значений логической функции, поэтому нам потребуется разместить на рабочей области лабораторного комплекса два дешифратора 3-8. Также следует обратить внимание, что количество адресных входов дешифратора меньше, чем количество переменных функции.

Поэтому подадим значения трех младших переменных функции на адресные входы обоих дешифраторов: младшую переменную «d» – на младший адресный вход, старшую переменную «b» – на старший адресный вход, переменную «с» – аналогично (на схеме далее переменные подаются на адресные входы дешифраторов при помощи разветвителя и шины).

Переменная «а» используется для управления дешифраторами. Когда «а» равна нулю, то должен работать первый дешифратор – он отвечает за первую половину таблицы истинности. Когда «а» равна единице, то должен работать второй дешифратор – он отвечает за вторую половину таблицы истинности. Чтобы это реализовать, переменная «а» должна подаваться на разрешающий вход первого дешифратора через инверсию, а на вход второго – без инверсии.

Для большей наглядности проиллюстрируем сказанное выше (рис. 2).

**Когда “a” равна единице, работает второй дешифратор**

**Когда “a” равна нулю, работает первый дешифратор**

**Область ответственности первого дешифратора**

**Область ответственности второго дешифратора**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | c | d | F |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

Рисунок 2 – Распределение областей таблицы истинности между дешифраторами 3-8

Для того чтобы у дешифраторов появился разрешающий вход, нам потребуется в их свойствах активировать соответствующую опцию.

Прочие настройки дешифраторов должны быть аналогичны предыдущей реализации.

В процессе работы на выходах всех дешифраторов будут последовательно возникать единичные значения в соответствии с поступающей на адресные входы комбинацией значений переменных. У первого дешифратора выберем лишь те выходы, чьи номера совпадают с номерами наборов значений переменных, на которых функция равна единице, из первой половины таблицы. У второго дешифратора выберем лишь те выходы, чьи номера совпадают с номерами наборов значений переменных за вычетом 8, на которых функция равна единице, из второй половины таблицы.

Объединим выбранные выходы обоих дешифраторов через «или» и получим требуемую реализацию (рис. 3).

Тестирование подтвердило правильность работы схемы.

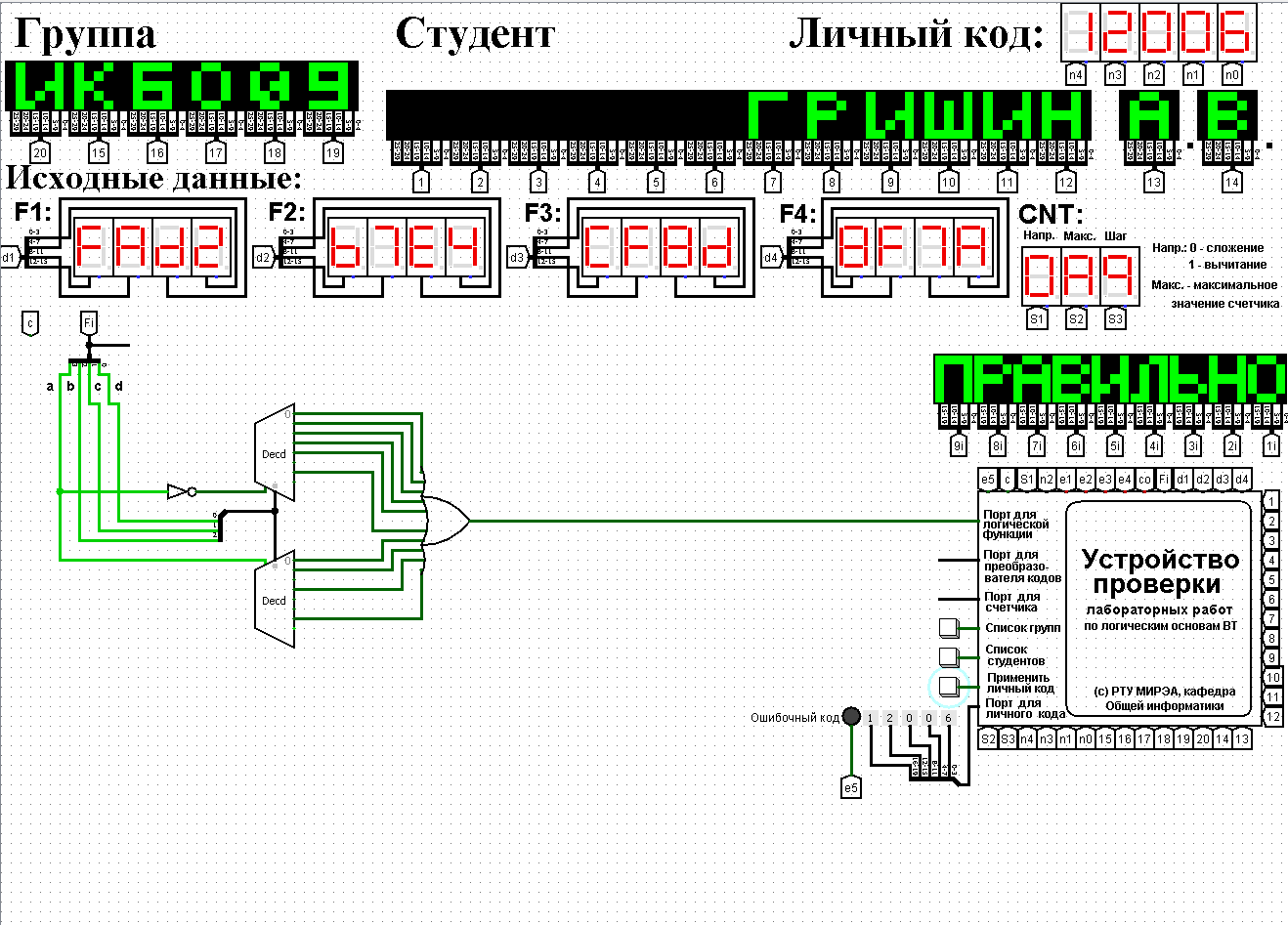


Рисунок 3 – Тестирование схемы, реализующей логическую функцию

на дешифраторах 3-8 и дополнительной логике

# Реализация функции, используя пять дешифраторов 2-4 и одну дополнительную схему «или»

Реализуем функцию, используя дешифраторы 2-4 и необходимую дополнительную логику. Количество выходов у дешифратора 2-4 в четыре раза меньше количества значений логической функции, поэтому нам потребуется разместить на рабочей области лабораторного комплекса четыре дешифратора 2-4, которые мы будем называть операционными, а также еще один дешифратор 2-4, который будет управлять первыми четырьмя – назовем его управляющим.

Итого всего потребуется пять дешифраторов 2-4 и дополнительная схема «или».

Следует обратить внимание, что количество адресных входов у каждого дешифратора в два раза меньше, чем количество переменных функции, поэтому каждый операционный дешифратор будет отвечать лишь за одну четверть исходной таблицы истинности. Для большей наглядности проиллюстрируем сказанное выше (рис. 4).

**Когда “a” равна нулю, работает первый дешифратор**

**Область ответственности первого дешифратора**

**Когда “a” равна единице, работает второй дешифратор**

**Область ответственности второго дешифратора**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | c | d | F |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

Рисунок 4 – Распределение областей таблицы истинности между дешифраторами 2-4

**Четвертый включается, когда на адресных входах управляющего 11**

**Область ответственности первого операционного дешифратора**

**Область ответственности второго операционного дешифратора**

**Область ответственности третьего операционного дешифратора**

**Область ответственности четвертого операционного дешифратора**

**Третий включается, когда на адресных входах управляющего 10**

**Второй включается, когда на адресных входах управляющего 01**

**Первый операционный дешифратор включается, когда на адресных входах управляющего дешифратора комбинация 00**

Значения двух младших переменных функции используются для адресации четырех операционных дешифраторов: младшая переменная «d» – подается на младший адресный вход, старшая переменная «с» – на старший адресный вход (на схеме далее переменные подаются на адресные входы дешифраторов при помощи разветвителя и шины).

Переменные «а» и «b» используется для управления операционными дешифраторами и аналогичным образом подаются на адресные входы управляющего дешифратора. Выходы управляющего дешифратора должны быть подключены к разрешающим входам операционных дешифраторов. Таким образом, когда «а» и «b» равны нулю, то на нулевом выходе управляющего дешифратора образуется единица, которая подается на разрешающий вход первого операционного дешифратора. И так далее, аналогично.

Теперь фактически каждый операционный дешифратор отвечает за свою двоичную тетраду в исходной векторной записи логической функции. Выберем у каждого операционного дешифратора лишь те выходы, где у двоичной тетрады стоят единицы. При этом необходимо считать, что нулевой выход соответствует старшему двоичному разряду тетрады.

Объединим выбранные выходы всех операционных дешифраторов через «или» и получим требуемую реализацию (рис. 5).

Тестирование подтвердило правильность работы схемы.

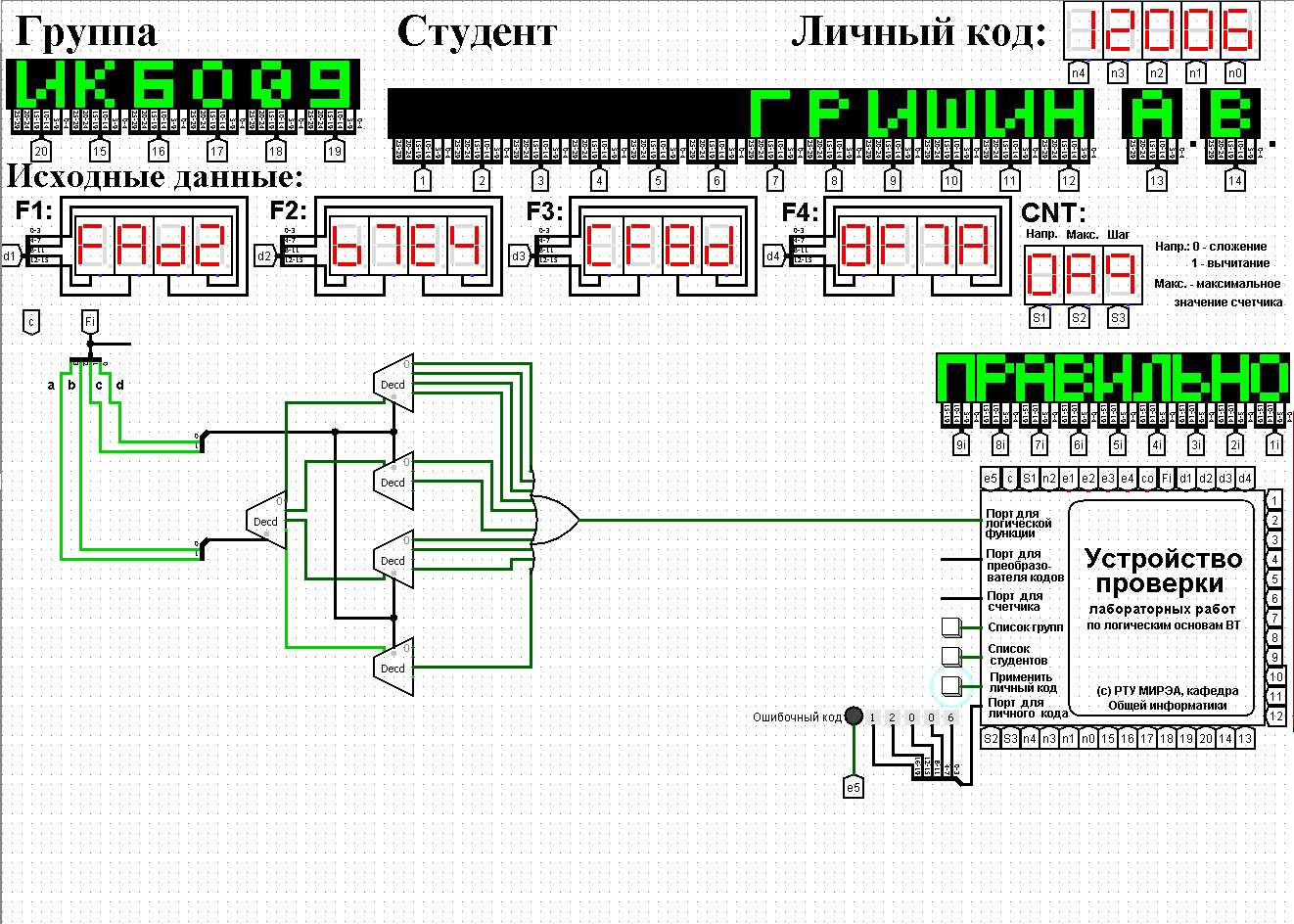


Рисунок 5 – Тестирование схемы, реализующей логическую функцию на дешифраторах 2-4 и дополнительной логике

# ВЫВОДЫ

В ходе практической работы, была восстановлена таблица истинности, и по ее значениям реализована в лабораторном комплексе логическая функция на дешифраторах тремя способами, с помощью: дешифратора 4-16 и одной дополнительной схемы «или», двух дешифраторов 3-8 и необходимой дополнительной логики и пяти дешифраторов 2-4 и одной дополнительной схемы «или». Тестирование показало, что все схемы работают правильно.

# СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Смирнов С.С., Карпов Д.А. Информатика: Методические указания по выполнению практических работ / Смирнов С.С., Карпов Д.А. – Москва: МИРЭА Российский технологический университет, 2020. – 102с.
2. Смирнов С. С. Лекционные материалы по информатике – Москва: МИРЭА Российский технологический университет, 2022 – лекция № 6. <https://cloud.mirea.ru/index.php/s/BXK49cca73ocX6P>.