



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
"МИРЭА - Российский технологический университет"

РТУ МИРЭА

Институт кибернетики
Кафедра общей информатики

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №7
«Реализация заданной логической функции от четырех
переменных на дешифраторах 4-16, 3-8 и 2-4»
по дисциплине
«ИНФОРМАТИКА»

Выполнил студент группы ИКБО-29-20

Хан А.А.

Принял
Доцент, к.т.н.

Норица В.М.

Практическая
работа выполнена

«__»_____2020 г.

«Зачтено»

«__»_____2020 г.

Москва 2020

СОДЕРЖАНИЕ

1. Постановка задачи и персональный вариант	3
2. Восстановленная таблица истинности	3
3. Схемы, реализующие логическую функцию на дешифраторах	4
3.1. Дешифратор 4-16.....	4
3.2. Дешифратор 3-8.....	6
3.3. Дешифратор 2-4.....	7
ВЫВОДЫ.....	9
СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ	10

1. Постановка задачи и персональный вариант

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».

Исходная функция: $F(a, b, c, d) = 3AE7_{16}$

Преобразуем ее в двоичную запись: 0011 1010 1110 0111₂– получаем столбец значений логической функции, который необходим для восстановления полной таблицы истинности (таблица 1).

2. Восстановленная таблица истинности

Таблица 1.

a	b	c	d	F
0	0	0	0	0
0	0	0	1	0
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	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

3. Схемы, реализующие логическую функцию на дешифраторах

3.1. Дешифратор 4-16

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: переменную «d» на младший адресный вход, переменную «a» - на старший адресный вход, прочие переменные – аналогично. В процессе работы на выходах дешифратора будут последовательно возникать единичные значения в соответствии с поступающей на адресные входы комбинацией значений переменных. Выберем лишь те выходы, 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).

3.2. Дешифратор 3-8

Реализуем функции, используя дешифраторы 3-8 и необходимую дополнительную логику. Количество выходов у дешифратора 3-8 в два раза меньше, чем количество значений функции, поэтому понадобятся два таких дешифратора. Подадим на каждый дешифратор значения трех переменных «b», «c» и «d». Переменная «a» будет использоваться для управления дешифраторами. Когда «a» равна нулю, то должен работать только первый дешифратор, отвечающий за первую половину таблицы истинности. Когда «a» равна единице, то должен работать только второй дешифратор, отвечающий за вторую половину таблицы истинности. Переменная «a» будет подаваться на разрешающий вход без инверсии на второй дешифратор и с инверсией на первый.

Объединим выбранные выходы обоих дешифраторов через «или» и получим требуемую реализацию (рис. 2).

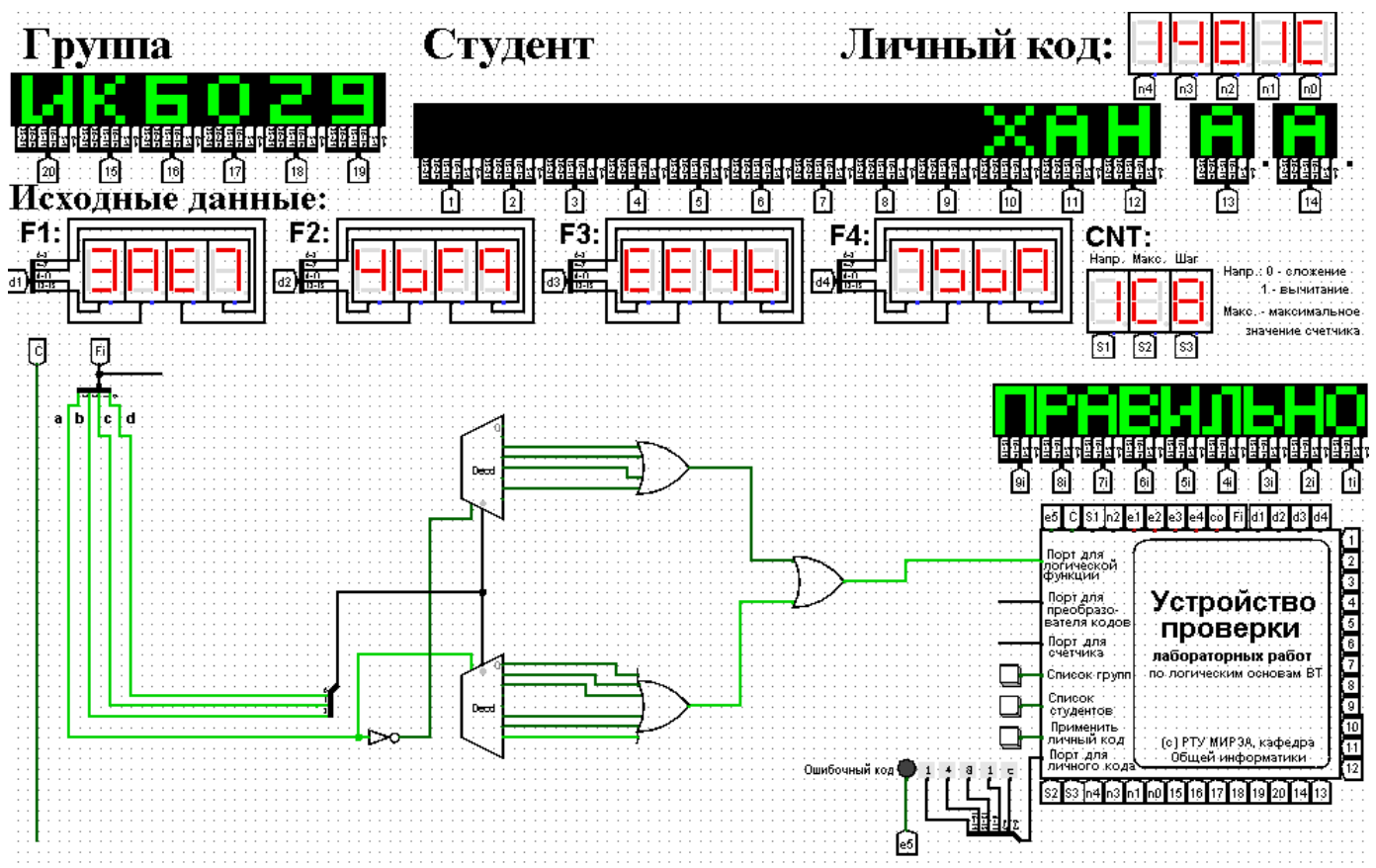


Рис. 2. Тестирование схемы, реализующей логическую функцию на дешифраторе 3-8 и дополнительной логике

3.3. Дешифратор 2-4

Реализуем функцию, используя дешифраторы 2-4 и необходимую дополнительную логику. Количество выходов у дешифратора 2-4 в 4 раза меньше чем значений логической функции, поэтому потребуются 4 дешифратора 2-4, которые назовем операционными. Также понадобится еще один дешифратор 2-4, который будет управлять остальными, который назовем управляющим. Так же понадобится одна дополнительная схема «или». Так как входов у дешифраторов 2-4 в два раза меньше, чем количество переменных функции, то каждый дешифратор будет отвечать лишь за четверть исходной таблицы истинности.

Переменные «с» и «d» будут подаваться на операционные дешифраторы, а переменные «a» и «b» - на управляющий. Выходы управляющего дешифратора должны быть подключены к разрешающим входам операционных дешифраторов.

Когда «a» и «b» равны нулю, на нулевом выходе управляющего дешифратора образуется единица, которая подается на разрешающий вход первого операционного дешифратора. И так далее, аналогично.

Таким образом каждый операционный дешифратор отвечает за свою двоичную тетраду в исходной логической функции. Выберем у каждого операционного дешифратора лишь те выходы, где у двоичной тетрады стоят единицы. Выбранные выходы объединяются через «или». Получаем требуемую реализацию (рис. 3).

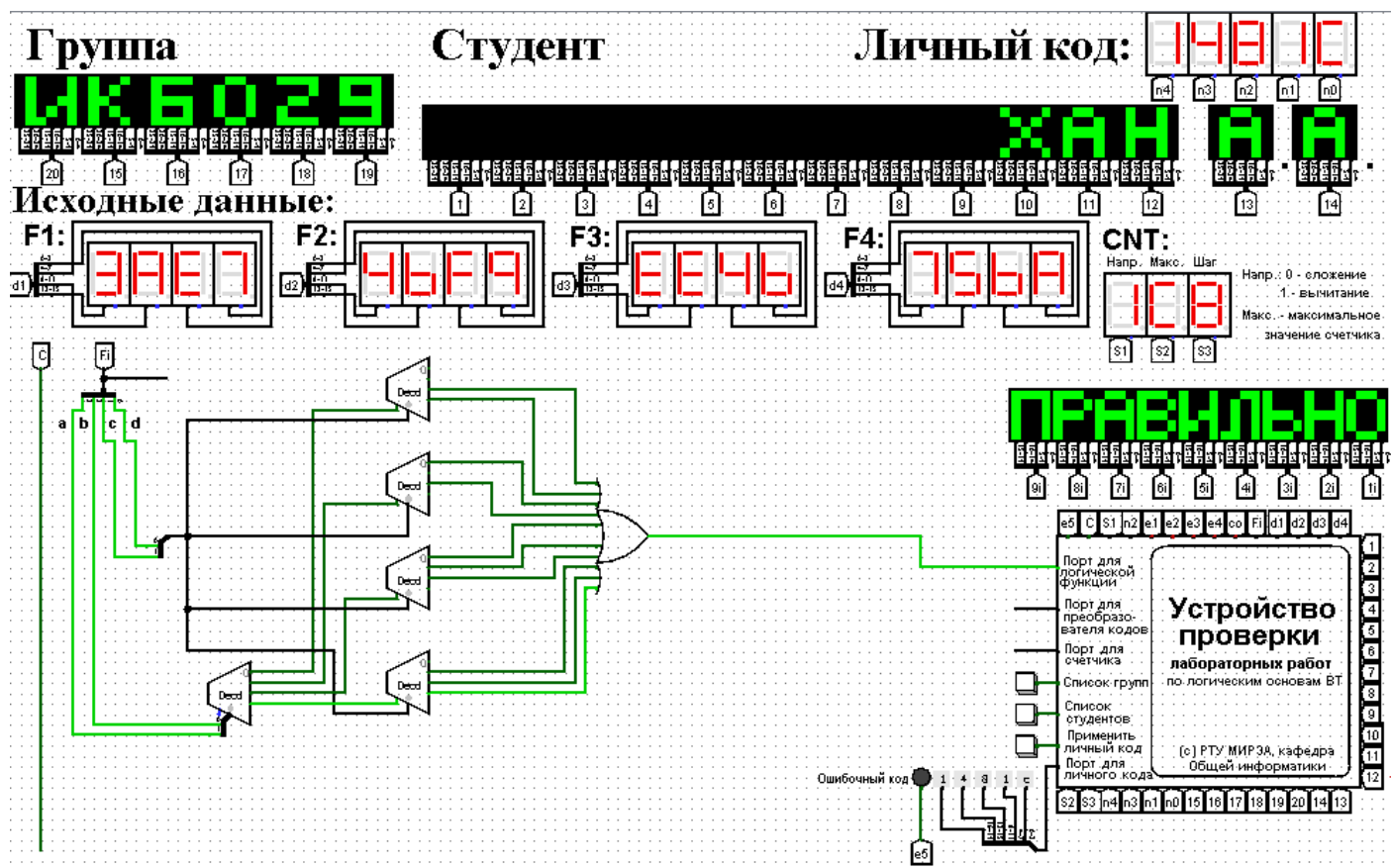


Рис. 3. Тестирование схемы, реализующей логическую функцию на дешифраторах 2-4 и дополнительной логике

Тестирование показало, что схема работает правильно.

ВЫВОДЫ

В ходе практической работы выполнены следующие задачи: восстановление таблицы истинности, построение комбинационной схемы, реализующей логическую функцию на дешифраторах 4-16, и построение комбинационных схем, реализующих логические функции на дешифраторах 3-8 и 2-4 с использованием дополнительной логики. Тестирование подтвердило правильность работы схем.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Программа построения и моделирования логических схем Logisim.
<http://www.cburch.com/logisim/> (Дата обращения 10.11.2020)
2. Справочная система программы Logisim. Устанавливается вместе с программой [1]. Также доступно: <http://www.cburch.com/logisim/ru/docs.html> (Дата обращения 10.11.2020)
3. Описание библиотеки элементов Logisim. Устанавливается вместе с программой [1]. Также доступно: <http://www.cburch.com/logisim/ru/docs.html> (Дата обращения 10.11.2020)
4. Информатика: Методические указания по выполнению практических работ / С.С. Смирнов, Д.А. Карпов М., МИРЭА — Российский технологический университет, 2020. –102с. (Дата обращения 10.11.2020)