



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

**РТУ МИРЭА**

---

Институт информационных технологий (ИТ)  
Кафедра Общей информатики

**ОТЧЕТ**  
**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №6 :**  
Реализация заданной логической функции от  
четырёх переменных на дешифраторах 4-16, 3-8 и 2-4  
**по дисциплине**  
**«ИНФОРМАТИКА»**

Выполнил студент группы ИНБО-15-20

Ло Ван Хунг

Принял  
Старший преподаватель

Шагалин Я.В

Практическая  
работа выполнена

«\_\_»\_\_\_\_\_2020 г.

«Зачтено»

«\_\_»\_\_\_\_\_2020 г.

Москва 2020

## **Содержание**

1.Постановка задачи и персональный вариант .....	3
2.Восстановленная таблица истинности .....	4
3.Схемы, реализующие логическую функцию на дешифраторах требуемыми способами (должны быть приведены фотографии экрана, на которых видны: группа, ФИО студента, индикаторы исходных данных, разработанные схемы с подключением к устройству проверки, а также положительный результат проверки).....	5
<b>ВЫВОДЫ .....</b>	<b>10</b>
<b>СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ .....</b>	<b>11</b>

## **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: 1. Используя дешифратор 4-16 и одну дополнительную схему «или». 2. Используя два дешифратора 3-8 и необходимую дополнительную логику. 3. Используя пять дешифраторов 2-4 и одну дополнительную схему «или». Протестировать работу схем и убедиться в их правильности. Подготовить отчет о проделанной работе и защитить ее. Персональный вариант: 1E81A

## 2. Восстановленная таблица истинности

$$F(a,b,c,d) = D56E_{16}$$

Преобразуем ее в двоичную запись: 1101 0101 0110 1110<sub>2</sub> – получили столбец значений логической функции, который необходим для восстановления полной таблицы истинности (см. табл.1).

Таблица 1: Таблица истинности для F

a	b	c	d	F	
0	0	0	0	1	D
0	0	0	1	1	
0	0	1	0	0	
0	0	1	1	1	
0	1	0	0	0	5
0	1	0	1	1	
0	1	1	0	0	
0	1	1	1	1	
1	0	0	0	0	6
1	0	0	1	1	
1	0	1	0	1	
1	0	1	1	0	
1	1	0	0	1	E
1	1	0	1	1	
1	1	1	0	1	
1	1	1	1	0	

**3. Схемы, реализующие логическую функцию на дешифраторах требуемыми способами (должны быть приведены фотографии экрана, на которых видны: группа, ФИО студента, индикаторы исходных данных, разработанные схемы с подключением к устройству проверки, а также положительный результат проверки).**

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.

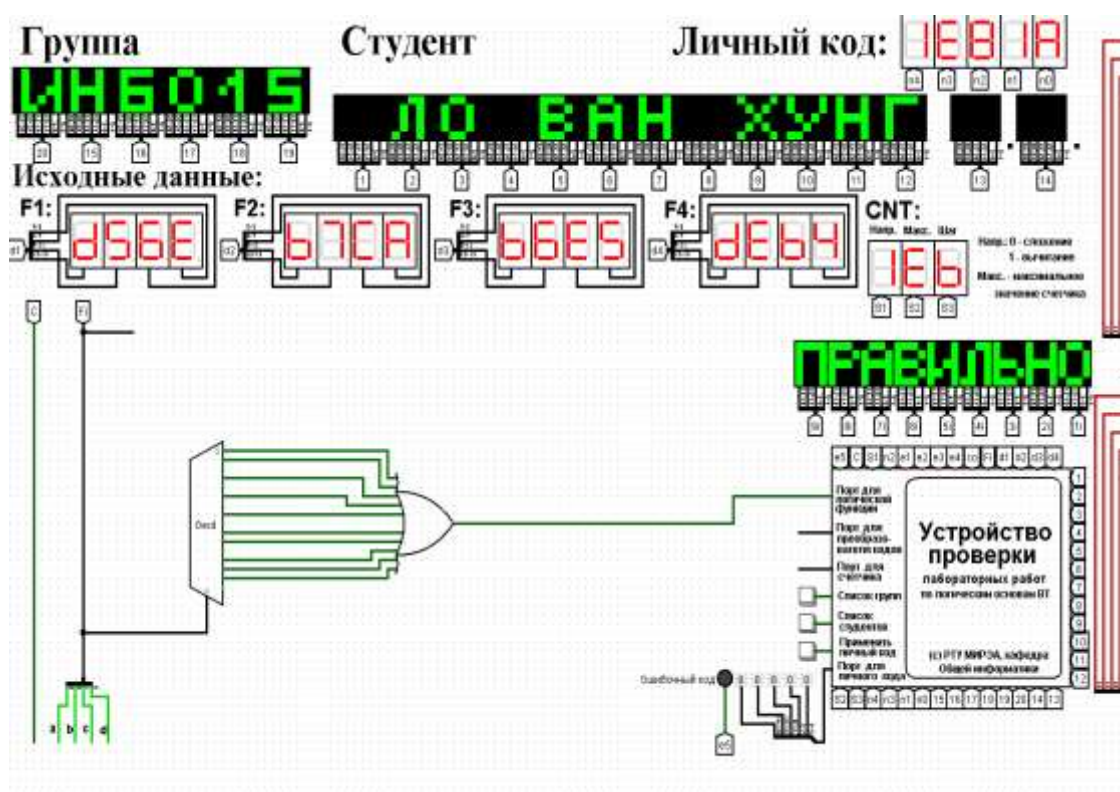


Рис. 1 Тестирование схемы, реализующей логическую функцию на дешифраторе 4-16

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 рисунком 2.

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

Рис. 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).

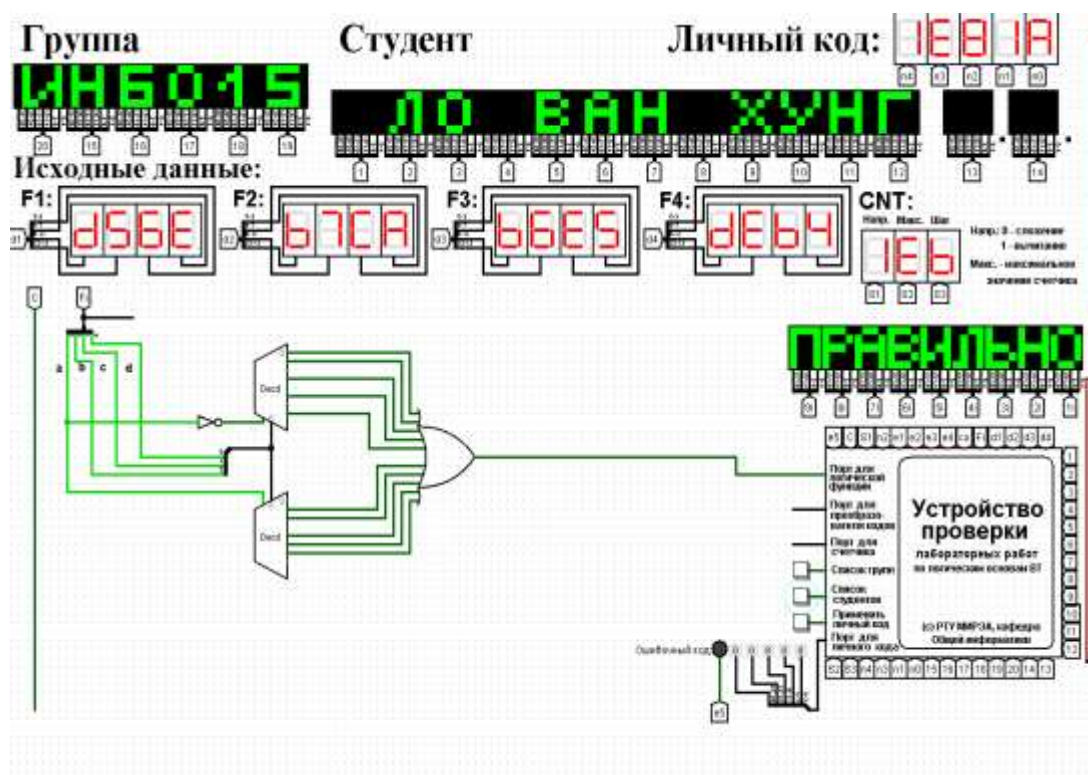


Рис. 3 Тестирование схемы, реализующей логическую функцию на дешифраторах 3-8 и дополнительной логике

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 рисунком 3.

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

Рис. 4 Распределение областей таблицы истинности между дешифраторами 2-4



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 «a» и «b» используются 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 «a» 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.4).

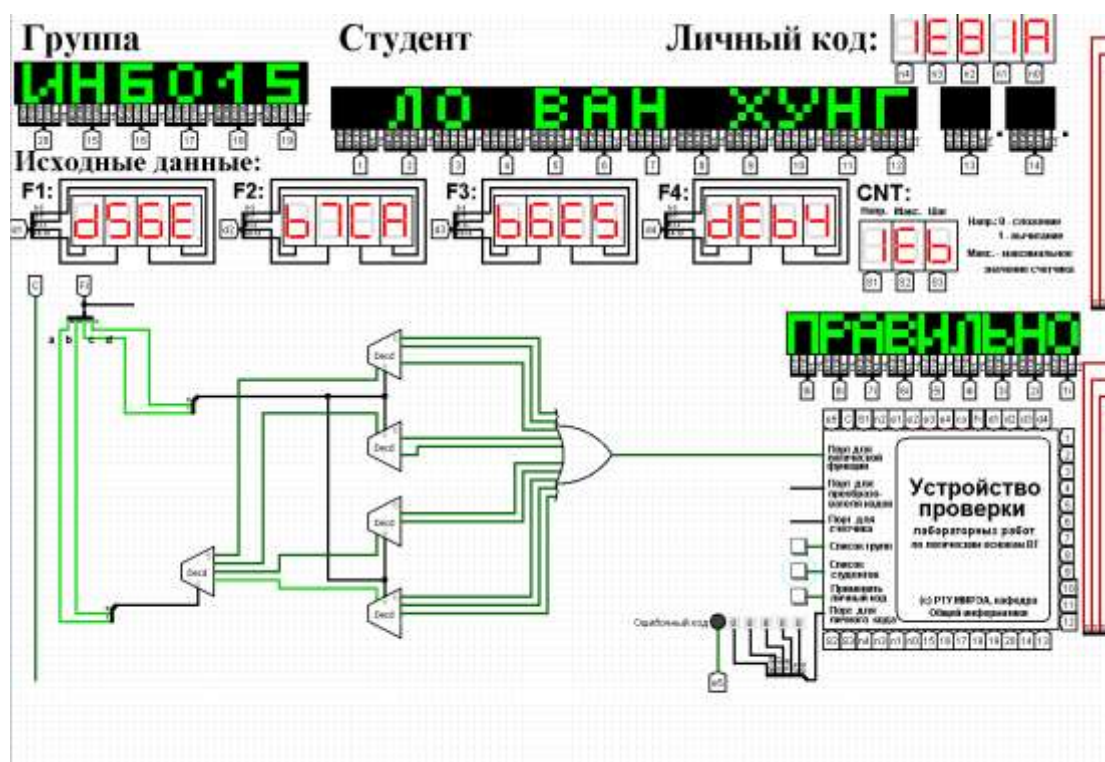


Рис.5 Тестирование схемы, реализующей логическую функцию на дешифраторах 2-4 и дополнительной логике.

## **ВЫВОДЫ**

Реализованная по таблице истинности логическая функция от четырех переменных заданная в 16-теричной векторной форме в лабораторном комплексе на дешифраторах тремя способами: 1. используя дешифратор 4-16 и одну дополнительную схему «или»; 2. используя два дешифратора 3-8 и необходимую дополнительную логику; 3. используя пять дешифраторов 2-4 и одну дополнительную схему «или», работает верно.

## СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Информатика: Методические указания по выполнению практических работ / С.С. Смирнов, Д.А. Карпов —М., МИРЭА — Российский технологический университет, 2020. – 102 с.
2. Справочная система программы Logisim.  
<http://www.cburch.com/logisim/ru/docs.html>.(27/11/2020)
3. Описание библиотеки элементов Logisim.  
<http://www.cburch.com/logisim/ru/docs.html>. .(27/11/2020)