



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 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 ИНБО-03-22

Иолович Е.А.

Принял доцент кафедры ОИ

Смолянинова В.А.

Практическая «\_\_» \_\_\_\_\_ 2022 г.

\_\_\_\_\_  
(подпись студента)

работа выполнена

«Зачтено» «\_\_» \_\_\_\_\_ 2022 г.

\_\_\_\_\_  
(подпись руководителя)

Москва 2022

## СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ .....	3
2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ .....	4
2.1 Восстановление таблицы истинности из 16-ричной векторной формы.....	4
2.2 Схема, реализующая логическую функцию при помощи дешифратора 4-16 и одной дополнительной схемы «ИЛИ» .....	5
2.3 Схема, реализующая логическую функцию при помощи дешифраторов 3-8 и одной дополнительной схемы «ИЛИ» .....	6
2.4 Схема, реализующая логическую функцию при помощи дешифраторов 2-4 и одной дополнительной схемы «ИЛИ» .....	8
3 ВЫВОДЫ .....	10
4 СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ .....	11

## 1 ПОСТАНОВКА ЗАДАЧИ

Логическая функция от четырёх 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 правильности их работы. Подготовить отчёт о проделанной работе и защитить её.

Персональный вариант:

$$F(a,b,c,d) = 36FA_{16} \quad (1)$$

## 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

### 2.1 Восстановление таблицы истинности из 16-ричной векторной формы

Для восстановления таблицы истинности логической функции из 16-ричной векторной формы необходимо каждый символ 16-ричной векторной формы представить в виде тетрады. Выполним этот перевод для 16-ричного векторного вида функции (1):

$$36FA_{16} = 0011\ 0110\ 1111\ 1010_2 \quad (2)$$

Получили столбец значений логической функции. С помощью него восстановим полную таблицу истинности (см. табл. 1).

Таблица 1 – Восстановленная таблица истинности

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	0
0	1	0	1	1
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	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

## 2.2 Схема, реализующая логическую функцию при помощи дешифратора 4-16 и одной дополнительной схемы «ИЛИ»

Реализуем функцию, используя дешифратор 4-16 и одну дополнительную схему «ИЛИ». Так как количество выходов дешифратора соответствует количеству значений логической переменной, потребуется только один такой дешифратор. Подадим значения переменных функции на адресные входы дешифратора, причём младшую переменную «d» подадим на младший адресный вход, а старшую переменную «a» - на старший адресный вход, переменные «b» и «c» подадим аналогично (на схеме далее переменные подаются на адресные входы дешифратора при помощи разветвителя и шины).

В процессе работы на выходах дешифратора (с 0 по 15) будут последовательно появляться единичные значения в соответствии с поступающей на адресные входы комбинаций переменных. Те выходы, номера которых совпадают с номерами наборов значений переменных, на которых функция равна единице, объединим при помощи схемы «или» и получим требуемую реализацию (рис. 1).

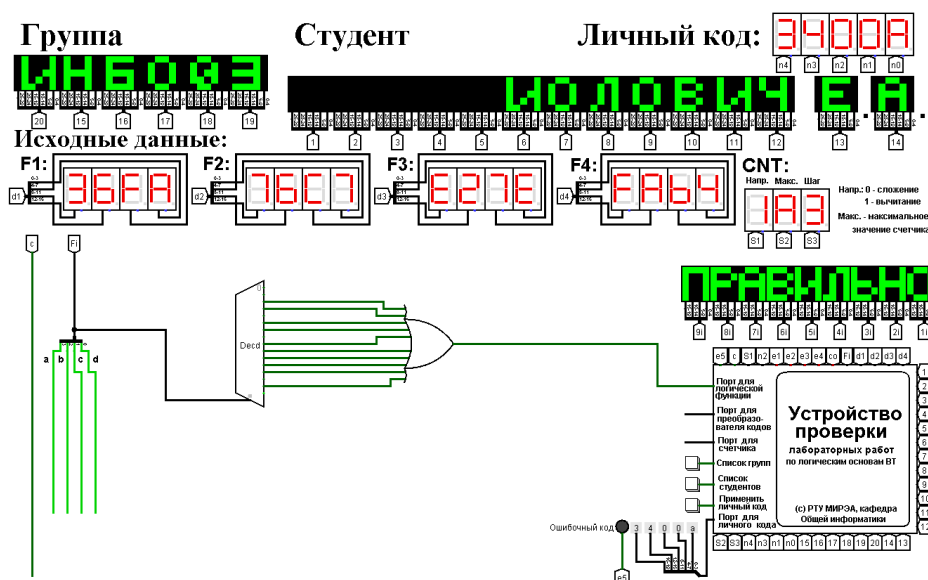


Рисунок 1 - Тестирование схемы, реализующей логическую функцию на дешифраторе 4-16

## 2.3 Схема, реализующая логическую функцию при помощи дешифраторов 3-8 и одной дополнительной схемы «ИЛИ»

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

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	0
0	1	0	1	1
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	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Когда "а" равна нулю, работает первый дешифратор (область ответственности первого дешифратора)

Когда "а" равна единице, работает второй дешифратор (область ответственности второго дешифратора)

Реализуем функцию, используя дешифраторы 3-8 и необходимую дополнительную логику. Так как количество выходов дешифратора 3-8 в два раза меньше количества значений логической функции, нам потребуется использовать два таких дешифратора. Также следует заметить, что количество адресных входов дешифратора 3-8 меньше, чем количество переменных функции. Подадим значения трёх младших переменных функции на адресные входы обоих дешифраторов, причём младшую переменную «d» подадим на младший адресный вход, а старшую переменную «b» - на старший адресный вход, переменные «с» подадим аналогично (на схеме далее переменные подаются на адресные входы дешифратора при помощи разветвителя и шины).

Переменную «а» используем для управления дешифраторами. Когда «а» равна 0, то должен работать первый дешифратор, который будет отвечать за первую половину таблицы истинности. Когда же «а» равна 1, то должен работать второй дешифратор, отвечающий за вторую половину таблицы

истинности. Чтобы реализовать это, будем подавать переменную «а» на разрешающий вход первого дешифратора через инверсию, а на разрешающий вход второго дешифратора – без инверсии.

В процессе работы на выходах всех дешифраторов будут последовательно появляться единичные значения в соответствии с поступающей на адресные входы комбинаций переменных. Выберем те выходы первого дешифратора, номера которых совпадают с номерами наборов значений переменных, на которых функция равна единице, из первой половины таблицы истинности. У второго дешифратора выберем те входы, номера которых совпадают с номерами наборов значений переменных, на которых функция равна единице, из второй половины таблицы истинности. Объединим выбранные выходы первого и второго дешифратора при помощи схемы «или» и получим требуемую реализацию (рис. 2).

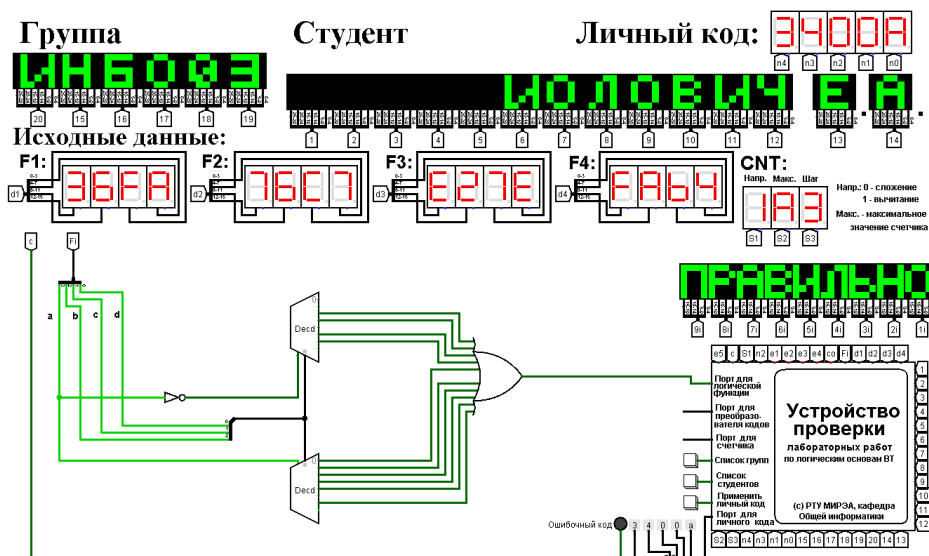


Рисунок 2 – Схема, реализующая логическую функцию при помощи дешифраторов 3-8

## 2.4 Схема, реализующая логическую функцию при помощи дешифраторов 2-4 и одной дополнительной схемы «ИЛИ»

Таблица 3 – Распределение областей таблицы истинности между дешифраторами 2-4

	a	b	c	d	F	
Первый операционный дешифратор включается, когда на адресных входах управляющего дешифратора комбинация 00	0	0	0	0	0	Область ответственности первого операционного дешифратора
	0	0	0	1	0	
	0	0	1	0	1	
	0	0	1	1	1	
Второй включается, когда на адресных входах управляющего 01	0	1	0	0	0	Область ответственности второго операционного дешифратора
	0	1	0	1	1	
	0	1	1	0	1	
	0	1	1	1	0	
Третий включается, когда на адресных входах управляющего 10	1	0	0	0	1	Область ответственности третьего операционного дешифратора
	1	0	0	1	1	
	1	0	1	0	1	
	1	0	1	1	1	
Четвертый включается, когда на адресных входах управляющего 11	1	1	0	0	1	Область ответственности четвертого операционного дешифратора
	1	1	0	1	0	
	1	1	1	0	1	
	1	1	1	1	0	

Реализуем функцию, используя дешифраторы 2-4 и необходимую дополнительную логику. Так как количество выходов дешифратора 2-4 в четыре раза меньше количества значений логической функции, нам потребуется использовать четыре дешифратора 2-4, которые будем называть операционными, и один дешифратор 2-4, который будет управлять первыми четырьмя, назовём его управляющим. Значит, что в общей сложности потребуется пять дешифраторов 2-4 и дополнительная схема «ИЛИ».

Также следует заметить, что количество адресных входов дешифратора 2-4 в два раза меньше, чем количество переменных функции, поэтому каждый из операционных дешифраторов будет отвечать лишь за одну четверть таблицы истинности. Подадим значения двух младших переменных функции



на адресные входы всех операционных дешифраторов, причём младшую переменную «d» подадим на младший адресный вход, а старшую переменную «с» - на старший адресный вход (на схеме далее переменные подаются на адресные входы дешифратора при помощи разветвителя и шины).

Переменные «а» и «b» используются для управления операционными дешифраторами, и аналогичным образом подаются на адресные входы управляющего дешифратора. Выходы управляющего дешифратора подключим к разрешающим входам операционных дешифраторов. Таким образом, когда «а» и «b» равны нулю, будет работать первый операционный дешифратор, отвечающий за первую четверть таблицы истинности. 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 те выходы, где у двоичной тетрады стоят единицы, причём будем считать, что нулевой выход дешифратора соответствует старшему двоичному разряду тетрады. Объединим выбранные выходы при помощи схемы «ИЛИ», и получим требуемую реализацию (рис. 3).

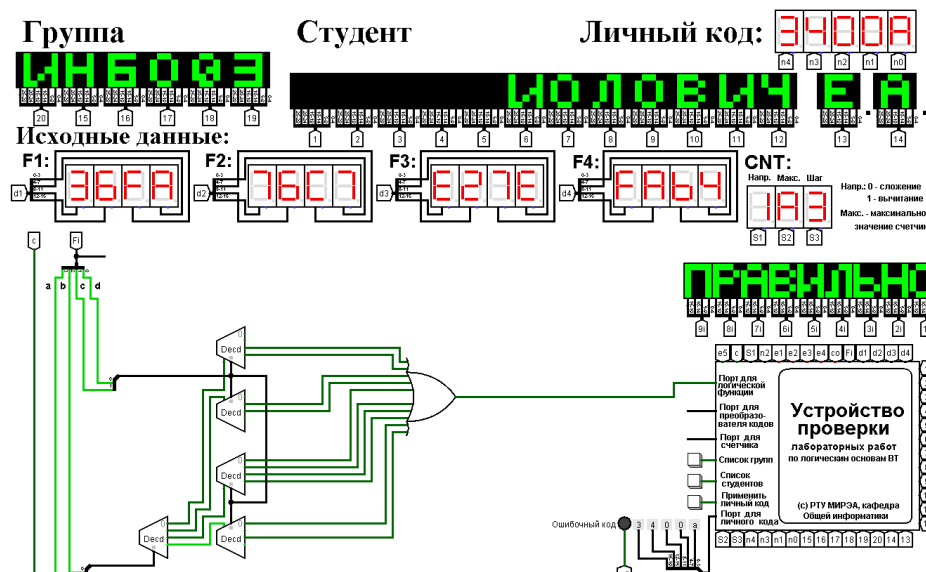


Рисунок 3 – Схема, реализующая логическую функцию при помощи дешифраторов 2-4

### **3 ВЫВОДЫ**

В этой практической работе были освоены навыки построения комбинационных схем, реализующих логическую функцию на дешифраторах 4-16, 3-8 и 2-4 в лабораторном комплексе, используя необходимую дополнительную логику и один дополнительный элемент «или», а также навыки тестирования работы схем в среде схемотехнического моделирования Logisim.

#### 4 СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Информатика: Методические указания по выполнению практических работ / С.С. Смирнов, Д.А. Карпов; Федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА - Российский технологический университет». – Москва : РТУ МИРЭА, 2020. – 102 с. – Текст: непосредственный.

2. Программа построения и моделирования логических схем Logisim: – Текст: электронный // Карл Берч: [сайт] – 2011. – URL: <http://www.cburch.com/logisim/ru/index.html>.