



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-08-21*

*Пономарев М.Д.*

Принял  
*Старший преподаватель*

*Смирнов С.С.*

Практическая  
работа выполнена  
«Зачтено»

«\_\_»\_\_\_\_\_2021 г.

\_\_\_\_\_  
(подпись студента)

«\_\_»\_\_\_\_\_2021 г.

\_\_\_\_\_  
(подпись руководителя)

Москва 2021

## СОДЕРЖАНИЕ

<b>1</b>	<b>ПОСТАНОВКА ЗАДАЧИ .....</b>	<b>3</b>
<b>2</b>	<b>ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ.....</b>	<b>4</b>
	<b>2.1 Построение таблицы истинности.....</b>	<b>4</b>
	<b>2.2 Схемы, реализующие логическую функцию на дешифраторах требуемыми способами.....</b>	<b>4</b>
<b>3</b>	<b>ВЫВОДЫ .....</b>	<b>12</b>
<b>4</b>	<b>ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ.....</b>	<b>13</b>

## 1 ПОСТАНОВКА ЗАДАЧИ

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 ее.

Заданная функция имеет вид:

$$F(a, b, c, d) = A6FC_{16}$$

## 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

### 2.1 Построение таблицы истинности

Функция, заданная в 16-теричной форме, имеет следующий вид:

$$F(a, b, c, d) = A6FC_{16}$$

Преобразуем ее в двоичную запись: 1010 0110 1111 1100<sub>2</sub> – получили столбец значений логической функции, который необходим для восстановления полной таблицы истинности (см. табл. 1).

Таблица 1 – Таблица истинности для функции  $F$

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

### 2.2 Схемы, реализующие логическую функцию на дешифраторах требуемыми способами

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 (рисунок 1).

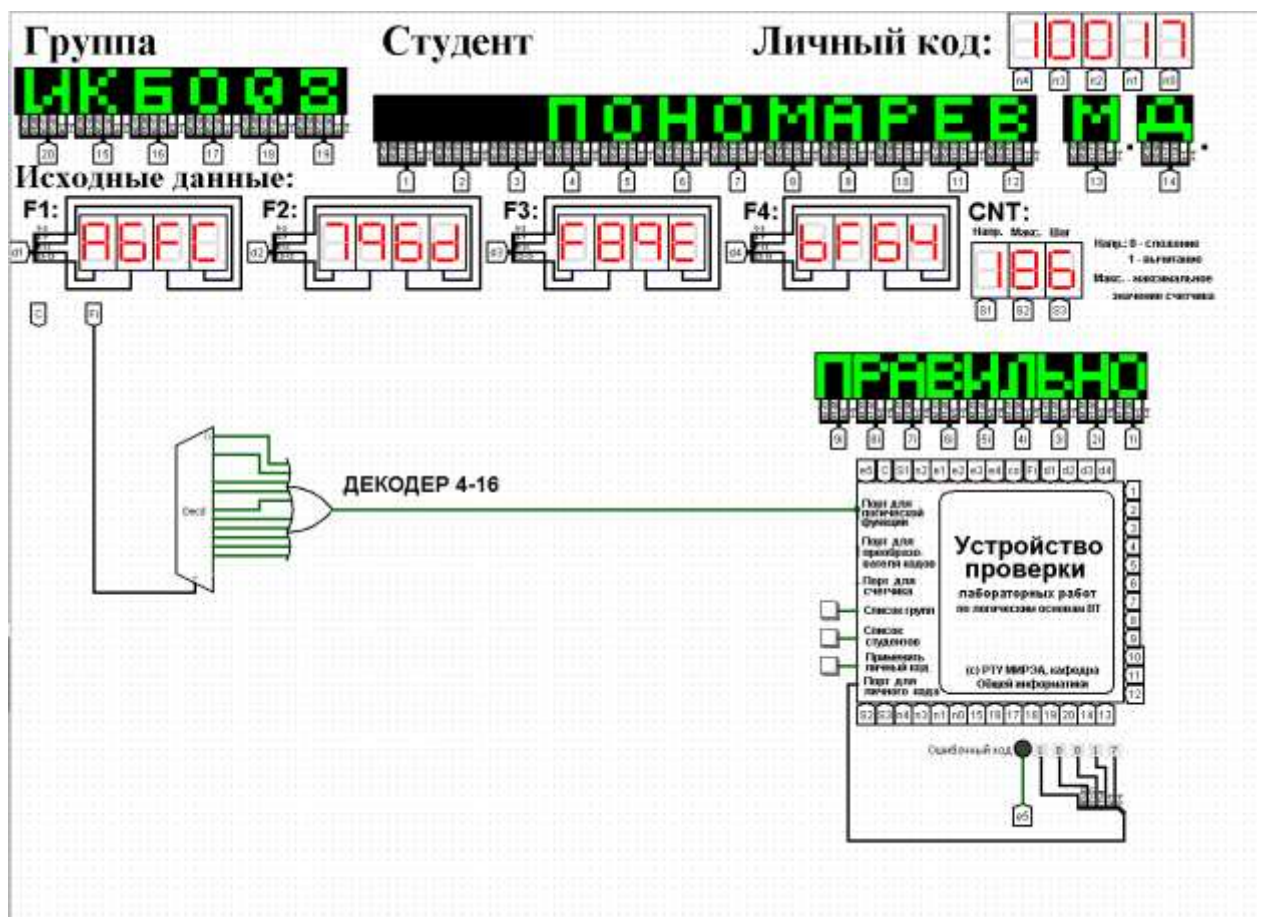


Рисунок 1 – Тестирование схемы, реализующей логическую функцию на дешифраторе 4-16

Тестирование показало, что схема работает правильно.

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
Когда 'a' равна нулю, работает первый дешифратор	0	0	0	0	1
	0	0	0	1	0
	0	0	1	1	1
	0	0	1	1	0
	0	1	0	0	0
	0	1	0	1	1
	0	1	1	0	1
	0	1	1	1	0
Когда 'a' равна единице, работает второй дешифратор	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	1
	1	1	1	0	0
	1	1	1	1	0

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

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 (рисунок 3).

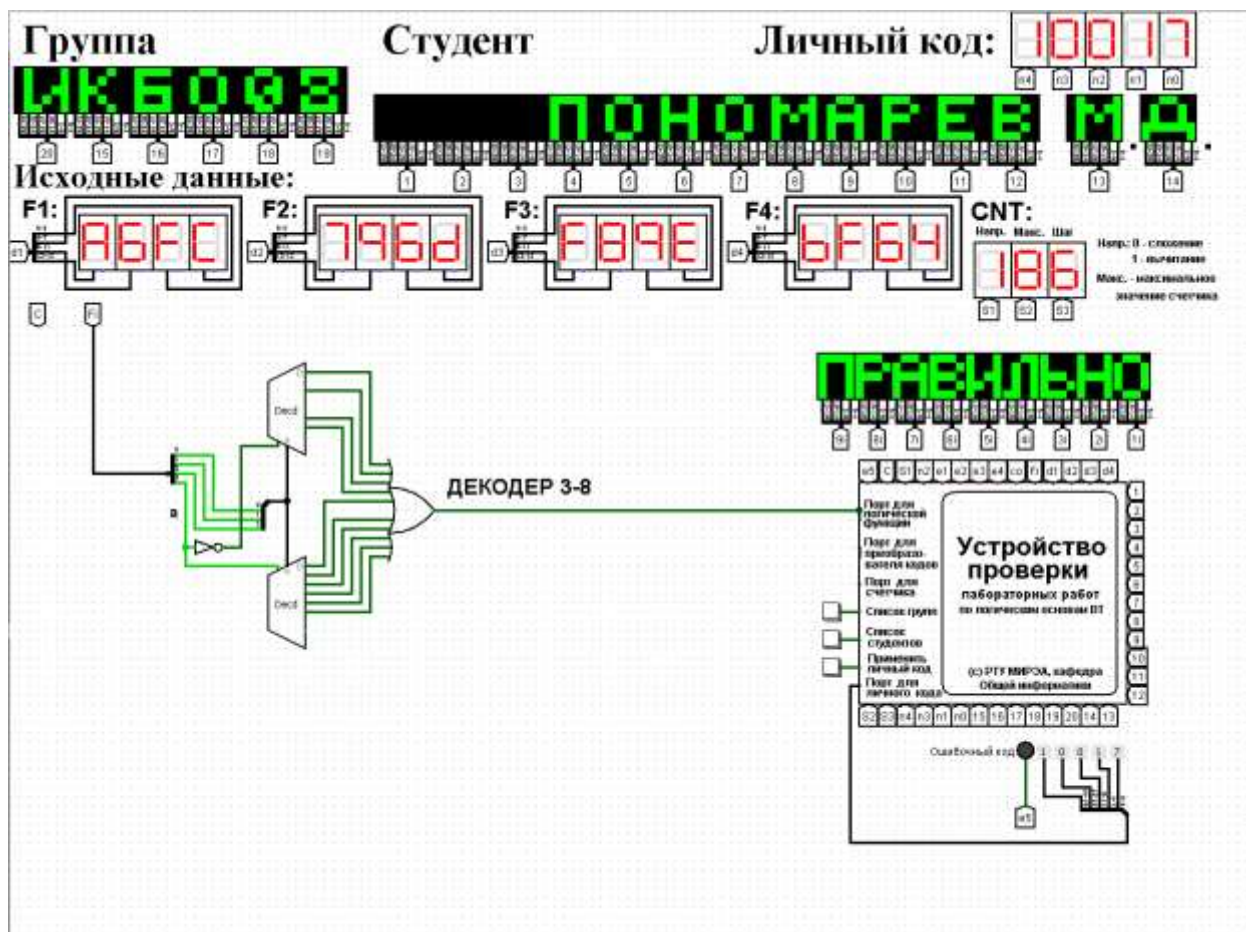


Рисунок 3 – Тестирование схемы, реализующей логическую функцию на дешифраторах 3-8 и дополнительной логике

Тестирование показало, что схема работает правильно.

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 4.

	a	b	c	d	F
Первый операционный дешифратор включается, когда на адресных входах управляющего дешифратора - 'a' равна нулю - 'b' равна нулю	0	0	0	0	1
	0	0	0	1	0
	0	0	1	1	1
	0	0	1	1	0
Второй операционный дешифратор включается, когда на адресных входах управляющего дешифратора - 'a' равна нулю - 'b' равна единице	0	1	0	0	0
	0	1	0	1	1
	0	1	1	0	1
	0	1	1	1	0
Третий операционный дешифратор включается, когда на адресных входах управляющего дешифратора - 'a' равна единице - 'b' равна нулю	1	0	0	0	1
	1	0	0	1	1
	1	0	1	0	1
	1	0	1	1	1
Четвертый операционный дешифратор включается, когда на адресных входах управляющего дешифратора - 'a' равна единице - 'b' равна единице	1	1	0	0	1
	1	1	0	1	1
	1	1	1	0	0
	1	1	1	1	0

Рисунок 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 «c» - на старший адресный вход (на схеме далее переменные подаются на адресные входы дешифраторов при помощи разветвителя и шины). Переменные «a» 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 «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 (рисунок 5).

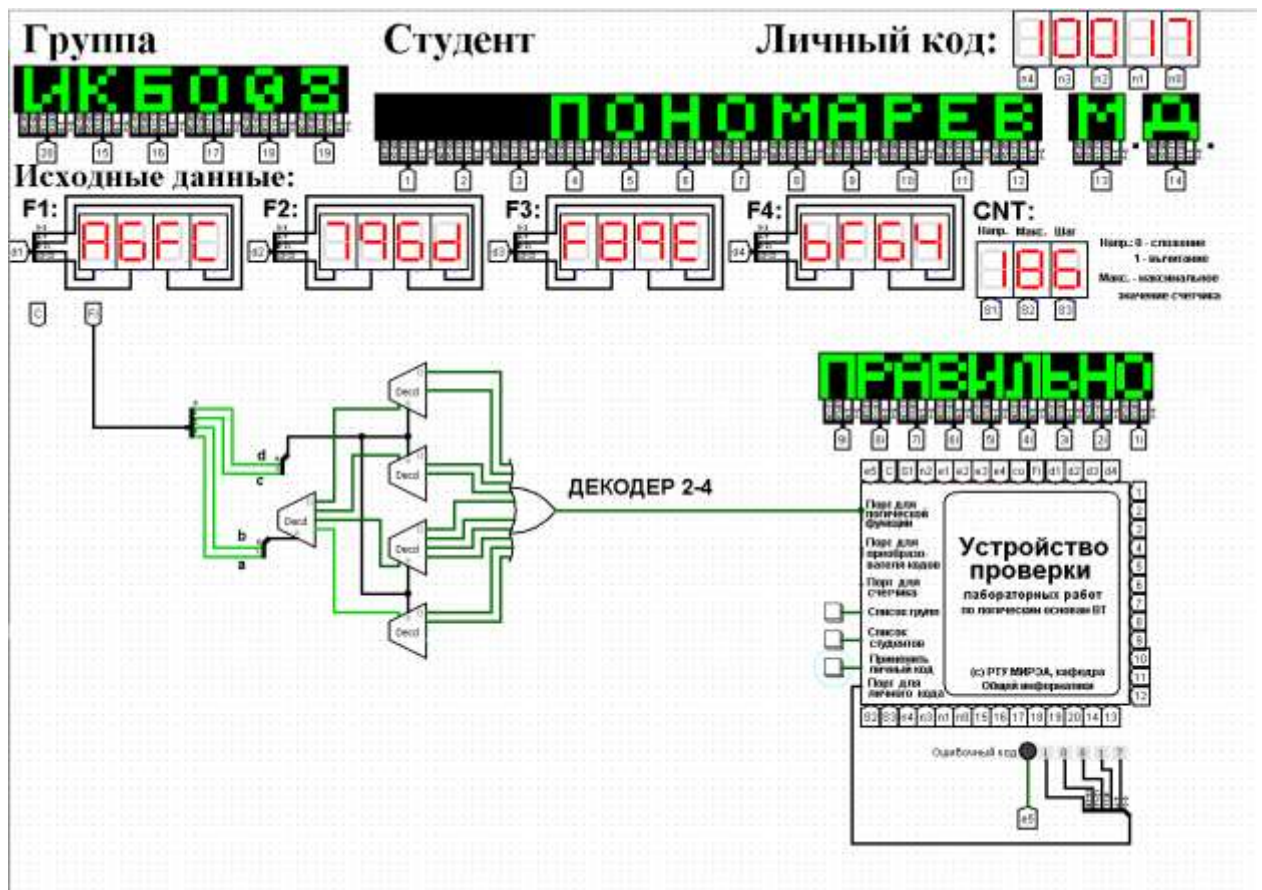


Рисунок 5 – Тестирование схемы, реализующей логическую функцию на дешифраторах 2-4 и дополнительной логике

Тестирование показало, что схема работает правильно.

### **3 ВЫВОДЫ**

В ходе практической работы была восстановлена таблица истинности рассматриваемой функции. По таблице истинности были построены в лабораторном комплексе комбинационные схемы, реализующие логическую функцию на дешифраторах требуемыми способами. Тестирование показало, что построенные схемы работают корректно.

#### **4 ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ**

1. Информатика: Методические указания по выполнению практических работ / С.С. Смирнов, Д.А. Карпов — М., МИРЭА — Российский технологический университет, 2020. — 102 с