



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт радиоэлектроники и информатики
Кафедра геоинформационных систем

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-30-23*

Павлов Н.С.

Принял ассистент кафедры ГИС

Корчемная А.И.

Практическая
работа выполнена

«__»_____ 2023 г.

«Зачтено»

«__»_____ 2023 г.

Москва 2023

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ	3
2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ	4
2.1 Персональный вариант	4
2.2 Восстановление таблицы истинности	4
2.3 Схема, реализующая логическую функцию на дешифраторе 4-16	5
2.4 Схема, реализующая логическую функцию на дешифраторе 3-8	6
2.5 Схема, реализующая логическую функцию на дешифраторе 2-4	8
3 ВЫВОДЫ	11
4 СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ	12

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.

2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

2.1 Персональный вариант

Вариант (личный код): 11015

В соответствии с вариантом функция, заданная в 16-теричной форме имеет следующий вид:

$$F(a, b, c, d) = 4C77_{16}$$

2.2 Восстановление таблицы истинности

Преобразуем функцию в двоичную запись: 0100 1100 0111 0111₂ – получили столбец значений логической функции, который необходим для восстановления полной таблицы истинности (см. таблицу 1)

Таблица 1 – Таблица истинности функции F

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

2.3 Схема, реализующая логическую функцию на дешифраторе 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: 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.

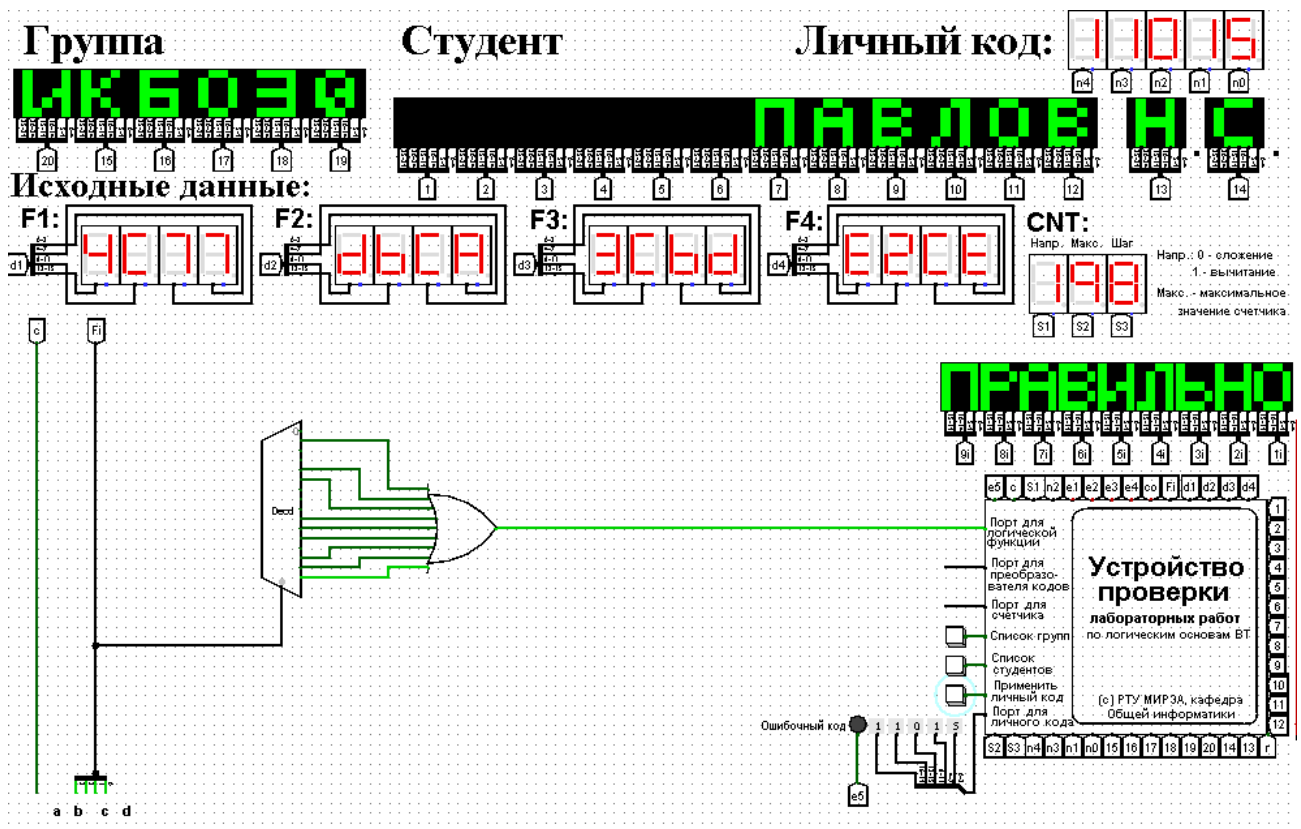


Рисунок 1 – Тестирование схемы, реализующей логическую функцию на дешифраторе 4-16

Тестирование показало, что схема работает правильно.

2.4 Схема, реализующая логическую функцию на дешифраторе 3-8

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 «c» — 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	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

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).

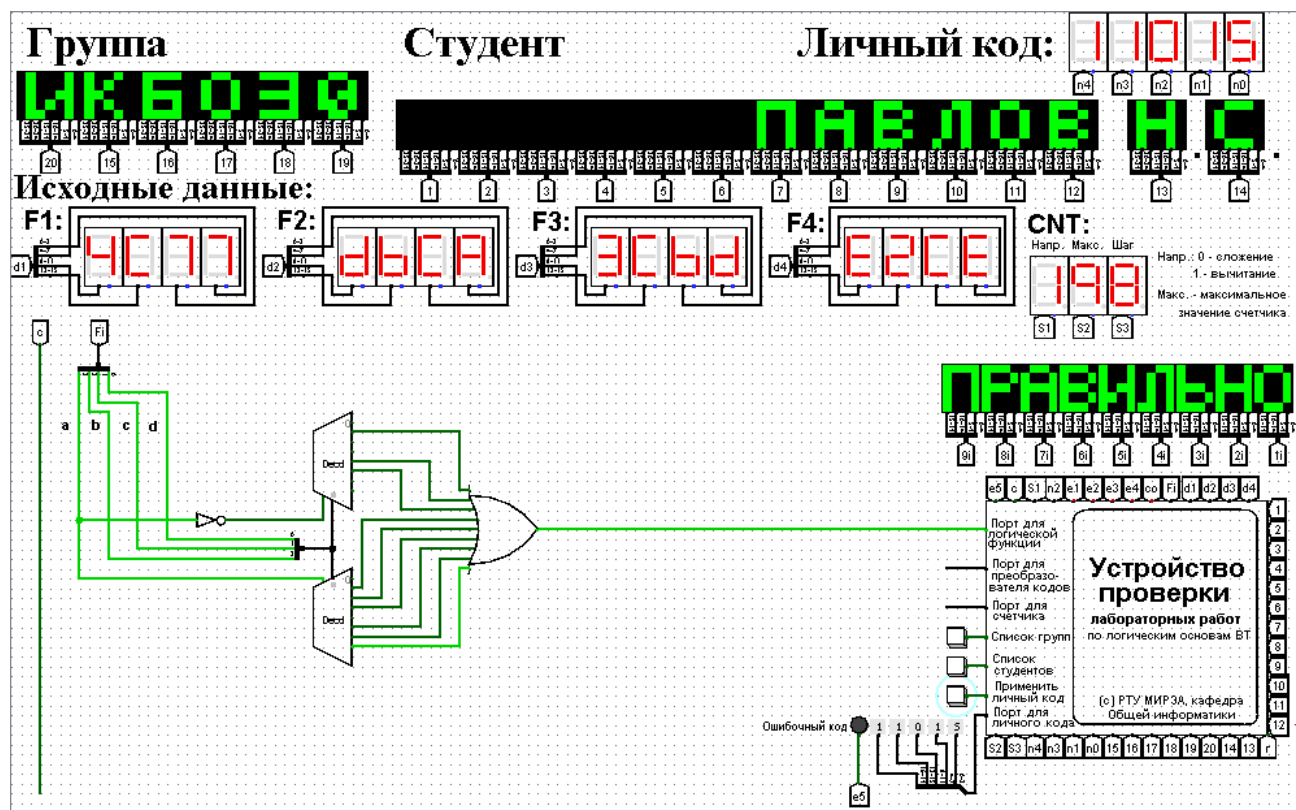


Рисунок 3 – Тестирование схемы, реализующей логическую функцию на дешифраторах 3-8 и дополнительной логике

Тестирование подтвердило правильность работы схемы.

2.5 Схема, реализующая логическую функцию на дешифраторе 2-4

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 комбинация 00	0	0	0	0	0	Область ответственности первого операционного дешифратора
	0	0	0	1	1	
	0	0	1	0	0	
	0	0	1	1	0	
Второй включается, когда на адресных входах управляющего 01	0	1	0	0	1	Область ответственности второго операционного дешифратора
	0	1	0	1	1	
	0	1	1	0	0	
	0	1	1	1	0	
Третий включается, когда на адресных входах управляющего 10	1	0	0	0	0	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	0	Область ответственности четвертого операционного дешифратора
	1	1	0	1	1	
	1	1	1	0	1	
	1	1	1	1	1	

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» 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 «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).

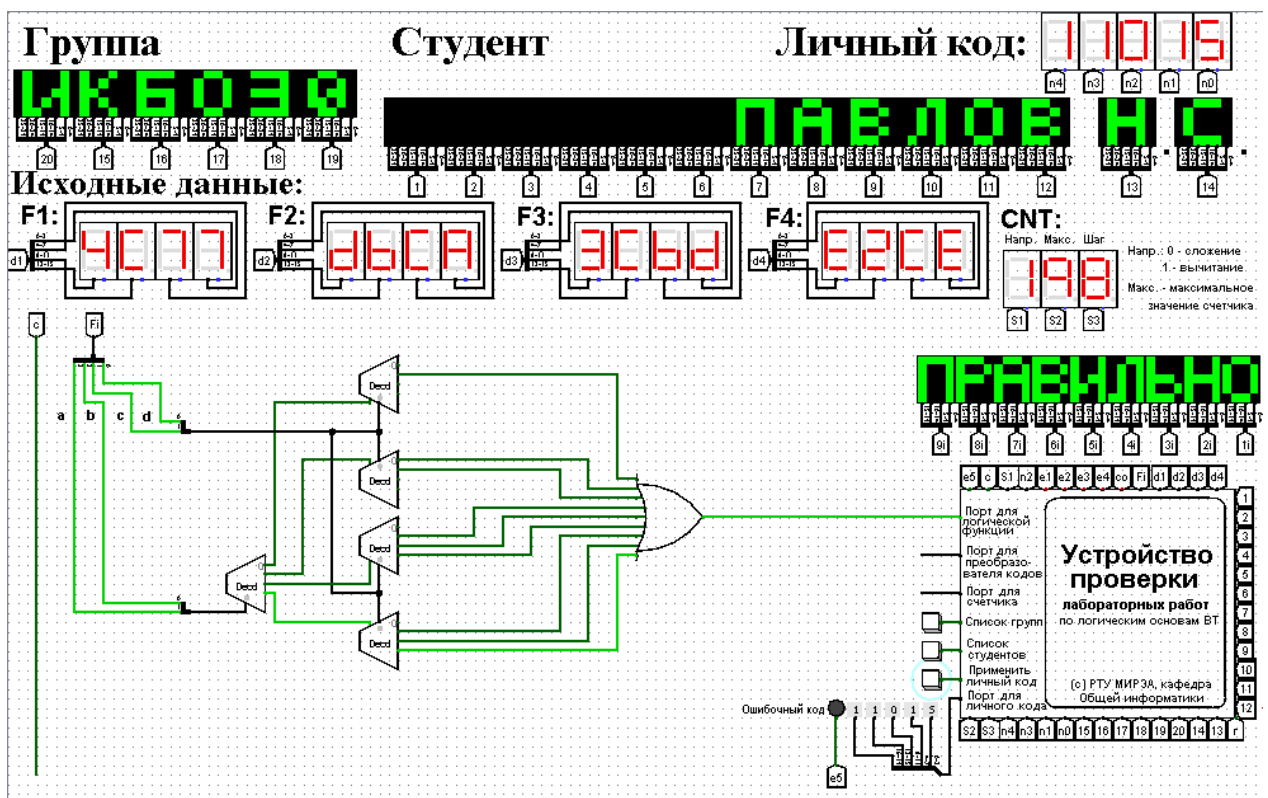


Рисунок 5 – Тестирование схемы, реализующей логическую функцию на дешифраторах 2-4 и дополнительной логике

Тестирование подтвердило правильность работы схемы.

3 ВЫВОДЫ

Восстановлена таблица истинности по логической функции от четырех переменных, заданной в 16-теричной векторной форме. По таблице истинности в лабораторном комплексе реализованы и протестированы схемы логической функции 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»

4 СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Информатика: Методические указания по выполнению практических работ / С.С. Смирнов, Д.А. Карпов, - М., МИРЭА – Российский технологический университет, 2020. – 102
2. Документация «Logisim». Текст: электронный. URL: <http://cburch.com/logisim/ru/docs.html> (дата обращения: 09.10.2023)