

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Болдырев Антон Сергеевич  
Должность: Директор  
Дата подписания: 24.02.2026 21:49:38  
Уникальный программный ключ:  
9c542731014dd7196f5752b7fa57c524495323a0



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)  
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО  
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
В Г. ТАГАНРОГЕ РОСТОВСКОЙ ОБЛАСТИ  
ПИ (филиал) ДГТУ в г. Таганроге**

**ЦМК «ПРИКЛАДНАЯ ИНФОРМАТИКА»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ  
ВЫПОЛНЕНИЯ ПРАКТИЧЕСКИХ РАБОТ  
по МДК.01.01 «Цифровая схемотехника»  
профессионального модуля  
«Участие в проектировании архитектуры  
интеллектуальных интегрированных систем»**

Таганрог

2026

Составители: Андрян И.В., Андрян О.В.

Методические указания по выполнению практической работы по МДК.01.01 «Цифровая схемотехника» ПИ (филиал) ДГТУ в г. Таганроге, 2026 г.

В методических указаниях кратко изложены теоретические вопросы, необходимые для успешного выполнения практической работы, рабочее задание и контрольные вопросы для самопроверки.

Предназначено для обучающихся по специальности 09.02.08 Интеллектуальные интегрированные системы.

Ответственный за выпуск:

Зав. кафедрой (председатель ЦМК) (руководитель структурного подразделения, ответственного за реализацию ОПОП): О.В. Андрян  
Ф.И.О.

Издательский центр ДГТУ, 2026 г.

## **Введение**

Данное учебно-методическое пособие предназначено для сопровождения практических занятий по курсу "Цифровая схемотехника". В нем представлены материалы, необходимые для успешного выполнения практических работ, углубляющих знания, полученные на лекциях, и формирующих практические навыки работы с цифровыми схемами.

Пособие содержит информацию о применяемых инструментах, таких как программы моделирования и синтеза цифровых схем, а также о методиках работы с ними. В нем описаны типовые задачи, встречающиеся в курсе "Цифровая схемотехника", и приведены различные подходы к их решению. Обучающийся должен

**Знать:**

методы проведения эффективных интервью;

принципы создания программно-аппаратных интерфейсов системы;

инфраструктуры проектируемой системы ПО;

инсталляции необходимого для создания информационной структуры

проектируемой системы ПО;

**Уметь:**

создавать инженерную документацию;

создавать макеты программно-аппаратных интерфейсов системы;

применять методы приемочных испытаний;

проводить демонстрацию функций системы

**Владеть навыками:**

взаимодействия с пользователями системы для выявления их требований к свойствам системы;

создания макетов программно-аппаратных интерфейсов системы;

проведения тестирования систем, аналогичных проектируемой;

работы с сетевыми модулями для подключения к веб-ресурсам в процессе проведения приемочных испытаний системы;

## Практическая работа №1-2.

### Представление информации в различных системах счисления

**Цель работы:** научиться переводить числа из одной системы счисления в другую.

**Теоретическая часть:**

Позиционная система счисления	Используемые символы						
Двоичная	0 и 1						
Троичная	0, 1, 2						
Восьмеричная	от 0 до 7						
Десятичная	от 0 до 9						
16-ричная	от 0 до 9	A	B	C	D	E	F
		10	11	12	13	14	15

Рисунок 1. Виды систем счисления

### Перевод в десятичную с.с.

1) Пронумеровать разряды числа справа налево  
 2) Записать сумму произведений цифр на соответствующие степени основания

Пример 1.  
 Перевести  $110110_2$  в десятичную систему счисления.

5 4 3 2 1 0  
 $110110_2 = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 32 + 16 + 4 + 2 = 54_{10}$ .

Ответ:  $110110_2 = 54_{10}$ .

$2^0=1$   
 $2^1=2$   
 $2^2=4$   
 $2^3=8$   
 $2^4=16$   
 $2^5=32$   
 $2^6=64$   
 $2^7=128$   
 $2^8=256$   
 $2^9=512$   
 $2^{10}=1024$

### Перевод из 10-тичной с.с. в с.с. с любым основанием

1) Последовательно выполнять деление на основание системы  
 2) Записать полученные остатки справа налево

Пример 2.  
 Перевести из десятичной с.с. в 2-ичную с.с.

$19 = 10011_2$

Рисунок 2. Пример перевода

Системы счисления - это способы представления чисел с помощью символов. В каждой системе счисления используется определенное количество символов, называемых основанием системы счисления.

Десятичная система счисления (десятичная) - самая распространенная система счисления, использующая 10 символов (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) и основание 10.

Двоичная система счисления (двоичная) - система счисления с основанием 2, использующая только два символа (0 и 1). В двоичной системе каждая цифра представляет степень двойки, начиная с  $2^0$  для младшего разряда.

Восьмеричная система счисления (восьмеричная) - система счисления с основанием 8, использующая 8 символов (0, 1, 2, 3, 4, 5, 6, 7). В восьмеричной системе каждая цифра представляет степень восьмерки, начиная с  $8^0$  для младшего разряда.

Шестнадцатеричная система счисления (шестнадцатеричная) - система счисления с основанием 16, использующая 16 символов (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). В шестнадцатеричной системе каждая цифра представляет степень шестнадцати, начиная с  $16^0$  для младшего разряда.

Перевод между системами счисления:

Из десятичной в двоичную: делим исходное число на 2, записываем остаток (0 или 1) в двоичную запись, повторяем процедуру с полученным частным, пока частное не станет 0. Двоичная запись составляется из полученных остатков, начиная с последнего.

Из двоичной в десятичную: умножаем каждую цифру двоичного числа на соответствующую степень двойки ( $2^0$  для младшего разряда) и суммируем полученные произведения.

Из десятичной в восьмеричную: делим исходное число на 8, записываем остаток (от 0 до 7) в восьмеричную запись, повторяем процедуру с полученным частным, пока частное не станет 0. Восьмеричная запись составляется из полученных остатков, начиная с последнего.

Из восьмеричной в десятичную: умножаем каждую цифру восьмеричного числа на соответствующую степень восьмерки ( $8^0$  для младшего разряда) и суммируем полученные произведения.

Из десятичной в шестнадцатеричную: делим исходное число на 16, записываем остаток (от 0 до F) в шестнадцатеричную запись, повторяем процедуру с полученным частным, пока частное не станет 0. Шестнадцатеричная запись составляется из полученных остатков, начиная с последнего.

Из шестнадцатеричной в десятичную: умножаем каждую цифру шестнадцатеричного числа на соответствующую степень шестнадцати ( $16^0$  для младшего разряда) и суммируем полученные произведения.

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

**Рабочее задание:**

Задание 1. Переведите в 10-тичную с.с.

№ варианта	Из двоичной
1	100011
2	11011
3	101011
4	111011
5	110101
6	101001
7	100100
8	1011101
9	110101
10	101101

Задание 2.

Какое из чисел 1100112 , 1111 , является:

- а) наибольшим;
- б) наименьшим?

Задание 3. Ответить на вопросы

Контрольные вопросы

1. Как производится перевод чисел из двоичной системы в десятичную
2. Как производится перевод чисел из десятичной системы в двоичную.

1. Перевести данное число из десятичной системы счисления в двоичную, восьмеричную

и шестнадцатеричную системы счисления.

2. Перевести данное число в десятичную систему счисления.

Вариант 1

1. а) 777; б) 305; в) 153,25; г) 162,25; д) 248,46.

2. а) 11001110112; б) 100000001112; в) 10110101,12; г) 100000110,101012; д) 671,248;

е) 41А,616.

Вариант 2

1. а) 164; б) 255; в) 712,25; г) 670,25; д) 11,89.

2. а) 10011100112; б) 10010002; в) 1111100111,012; г) 1010001100,1011012; д) 413,418; е) 118,8С16.

Вариант 3

1. а) 273; б) 661; в) 156,25; г) 797,5; д) 53,74.

2. а) 11000000002; б) 11010111112; в) 1011001101,000112; г) 1011110100,0112; д) 1017,28; е) 111,В16.

Вариант 4

1. а) 105; б) 358; в) 377,5; г) 247,25; д) 87,27.

2. а) 11000010012; б) 11001001012; в) 1111110110,012; г) 11001100,0112; д) 112,048; е) 334,А16.

Вариант 5

1. а) 500; б) 675; в) 810,25; г) 1017,25; д) 123,72.

2. а) 11010100012; б) 1000111002; в) 1101110001,0110112; г) 110011000,1110012; д) 1347,178 (8); е) 155,6С16 (16).

Вариант 6

1. а) 218; б) 808; в) 176,25; г) 284,25; д) 253,04.

2. а) 1110001002; б) 10110011012; в) 10110011,012; г) 1010111111,0112; д) 1665,38; е) FА,716.

Вариант 7

1. а) 306; б) 467; в) 218,5; г) 667,25; д) 318,87.

2. а) 11110001112; б) 110101012; в) 1001111010,0100012; г) 1000001111,012; д) 465,38; е) 252,3816.

Вариант 8

1. а) 167; б) 113; в) 607,5; г) 828,25; д) 314,71.
2. а) 1100100012; б) 1001000002; в) 1110011100,1112; г) 1010111010,11101112; д) 704,68; е) 367,3816.

#### Вариант 9

1. а) 342; б) 374; в) 164,25; г) 520,375; д) 97,14.
2. а) 10001101102; б) 1111000012; в) 1110010100,10110012; г) 1000000110,001012; д) 666,168; е) 1С7,6816.

#### Вариант 10

1. а) 524; б) 222; в) 579,5; г) 847,625; д) 53,35.
2. а) 1011111112; б) 11111001102; в) 10011000,11010112; г) 1110001101,10012; д) 140,228; е) 1DE,5416.

#### Вариант 11

1. а) 113; б) 875; в) 535,1875; г) 649,25; д) 6,52.
2. а) 111010002; б) 10100011112; в) 1101101000,012; г) 1000000101,010112; д) 1600,14; е) 1E9,416.

#### Вариант 12

1. а) 294; б) 723; в) 950,25; г) 976,625; д) 282,73.
2. а) 100000110012; б) 101011002; в) 1101100,012; г) 1110001100,12; д) 1053,28; е) 200,616.

#### Вариант 13

1. а) 617; б) 597; в) 412,25; г) 545,25; д) 84,82.
2. а) 1101111012; б) 11100111012; в) 111001000,012; г) 1100111001,10012; д) 1471,178; е) 3EC,516.

#### Вариант 14

1. а) 1047; б) 335; в) 814,5; г) 518,625; д) 198,91.
2. а) 11011000002; б) 1000010102; в) 1011010101,12; г) 1010011111,11012; д) 452,638; е) 1E7,0816.

#### Вариант 15

1. а) 887; б) 233; в) 801,5; г) 936,3125; д) 218,73.

2. а) 10101000012; б) 100000101012; в) 1011110000,1001012; г) 1000110001,10112;  
д) 1034,348; е) 72,616.

Вариант 16

1. а) 969; б) 549; в) 973,375; г) 508,5; д) 281,09.

2. а) 101000102; б) 11100101112; в) 110010010,1012; г) 1111011100,100112;  
д) 605,028; е) 3С8,816.

Вариант 17

1. а) 163; б) 566; в) 694,375; г) 352,375; д) 288,61.

2. а) 10011010012; б) 1100111012; в) 1000001101,012; г) 1010001001,110112;  
д) 247,18; е) 81,416.

**Время работы: 2 часа.**

### **Практическое занятие №3-4**

#### **Кодирование целых, дробных и смешанных чисел в различных системах счисления**

**Цель:** научиться кодировать числа в различных системах счисления.

#### **Теоретические сведения**

Арифметические действия, выполняемые в двоичной системе, подчиняются тем же

правилам, что и в десятичной системе. Только в двоичной системе счисления перенос единиц в

старший разряд возникает чаще, чем в десятичной. Вычитание многоразрядных двоичных чисел

производится с учетом возможных займов из старших разрядов. Действия умножения и деления в

двоичной арифметике можно выполнять по общепринятым для позиционных систем правилам.

**Сложение двоичных чисел.** Способ сложения столбиком такой же, как и для десятичного

числа. То есть, сложение выполняется поразрядно, начиная с младшего. Если получается больше 1,

то записывается 0 и 1 добавляется к старшему разряду. Вот так выглядит таблица сложения в

двоичной системе:

$0+0=0$	$0+1=1$
$1+0=1$	$1+1=10$ (перенос в старший разряд)

Пример 1:  $10011 + 10001$ .

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

Вычитание двоичных чисел. Вычитание выполняется поразрядно и если в разряде не хватает единицы, то она занимается в старшем. Первый разряд.  $1 - 0 = 1$ . Записываем 1. 3 Второй разряд  $0 - 1$ . Не хватает единицы. Занимаем её в старшем разряде. Единица из старшего разряда переходит в младший, как две единицы (потому что старший разряд представляется двойкой большей степени)  $2 - 1 = 1$ . Записываем 1. Третий разряд. Единицу этого разряда мы занимали, поэтому сейчас в разряде 0 и есть необходимость занять единицу старшего разряда.  $2 - 1 = 1$ . Записываем 1. Проверим результат в десятичной системе:  $1101 - 110 = 13 - 6 = 7$  (111) Верное равенство.

Умножение в двоичной системе счисления. Таблица умножения для двоичных чисел ещё

проще, чем для сложения.

Для того, чтобы умножить двоичное число на 2 (десятичная двойка это 10 в двоичной

системе) достаточно к умножаемому числу слева приписать один ноль.

Пример 3:  $10101 * 10 = 101010$

Проверка.

$$10101 = 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 16 + 4 + 1 = 21$$

$$101010 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 32 + 8 + 2 = 42$$

$$21 \cdot 2 = 42$$

Умножение - это ряд последовательных сдвигов. Общее правило таково: как и для

десятичных чисел, умножение двоичных выполняется поразрядно. И для каждого разряда второго множителя к первому множителю добавляется один ноль справа. Пример

4:(пока не столбиком):

$1011 \cdot 101$  Это умножение можно свести к сумме трёх поразрядных умножений

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующего занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);

- выполнение
- заключение

### **Рабочее задание:**

1. Выполнить арифметическую операцию сложения в двоичной системе.
2. Выполнить арифметическую операцию вычитания в двоичной системе.
3. Выполнить арифметическую операцию умножения в двоичной системе.
4. Ответить на контрольные вопросы.

### **Контрольные вопросы**

1. С какого разряда начинается сложение?
2. В каком разряде занимается единица при вычитании?
3. Как производится перевод чисел из десятичной системы в двоичную.

**Время работы:** 2 часа.

## **Практическая работа 5-6**

### **Построение схем и таблиц истинности для заданных логических функций**

#### **Цель:**

Научиться строить схемы логических функций, используя стандартные логические элементы (И, ИЛИ, НЕ, XOR).

Научиться составлять таблицы истинности для заданных логических функций.

Развивать навыки логического мышления и анализа.

#### **Теоретическая часть**

1. Логические операции и их представление

Логические операции - это операции над истинностными значениями (истина или ложь), которые возвращают истинностное значение. Основные логические операции:

Конъюнкция (И):  $F(A, B) = A \text{ AND } B$  - истинна только тогда, когда оба аргумента истинны.

Дизъюнкция (ИЛИ):  $F(A, B) = A \text{ OR } B$  - истинна, если хотя бы один из аргументов истинен.

Отрицание (НЕ):  $F(A) = \text{NOT } A$  - истинна, если аргумент ложен, и ложна, если аргумент истинен.

Исключающее ИЛИ (XOR):  $F(A, B) = A \text{ XOR } B$  - истинна, если один из аргументов истинен, а другой ложен.

Импликация:  $F(A, B) = A \text{ IMPLICATION } B$  - ложна только тогда, когда А истинно, а В ложно.

Представление логических операций:

Таблицы истинности: показывают все возможные комбинации входных значений и соответствующие выходные значения.

Логические выражения: записывают логические функции с использованием символов операций и переменных.

Логические схемы: представляют логические функции с помощью графических элементов (вентилей), которые реализуют логические операции.

## 2. Логические элементы (вентили)

Логические элементы - это электронные схемы, которые реализуют логические операции. Основные типы вентилей:

И-вентиль: выдает "1" только тогда, когда все его входы "1".

ИЛИ-вентиль: выдает "1", если хотя бы один из его входов "1".

НЕ-вентиль (инвертор): выдает "1", если на его входе "0", и наоборот.

XOR-вентиль: выдает "1", если на его входах разные значения.

IMPLICATION-вентиль: выдает "0" только тогда, когда первый вход "1", а второй "0".

### 3. Построение схем

Логические схемы строятся с помощью соединения вентиляей.

Каждый вентиль представляет собой определенную логическую операцию.

Входы и выходы вентиляей соединяются между собой, чтобы реализовать заданную логическую функцию.

Схемы можно строить по таблице истинности или по логическому выражению.

### 4. Составление таблиц истинности

Таблица истинности - это сводка всех возможных комбинаций входных значений и соответствующих выходных значений для заданной логической функции.

Таблица содержит столбец для каждой входной переменной и столбец для выходного значения.

Количество строк в таблице равно  $2^n$  в степени количества входных переменных.

### 5. Применение логических функций

Логические функции широко используются в электронике, информатике, математической логике и других областях. Они применяются для:

Реализации цифровых схем: процессоры, память, логические контроллеры.

Программирования: условные операторы, циклы, логические выражения в языках программирования.

Математической логики: доказательства теорем, построение формальных систем.

Искусственного интеллекта: логические системы, экспертные системы, машинное обучение.

## 6. Основные понятия

Входные переменные: переменные, которые задают значения на входе логической функции.

Выходная переменная: переменная, которая представляет результат работы логической функции.

Логическое значение: может быть истинным (1) или ложным (0).

Логическое выражение: представление логической функции в виде формулы с использованием символов логических операций.

Логическая схема: графическое представление логической функции с помощью логических элементов (вентилей).

### Рабочие задания:

#### 1. Построение схем логических функций:

Задание 1.1: Для каждой из следующих функций построить схему, используя только логические элементы И, ИЛИ и НЕ.

$$F(A, B, C) = (A \text{ AND } B) \text{ OR } (\text{NOT } C)$$

$$F(A, B, C) = (\text{NOT } A) \text{ AND } (B \text{ XOR } C)$$

$$F(A, B, C) = (A \text{ OR } B) \text{ AND } (C \text{ XOR NOT } B)$$

Задание 1.2: Для каждой функции из Задания 1.1 построить схему, используя только логические элементы И, ИЛИ и НЕ, но с минимальным количеством элементов.

Задание 1.3: Построить схему для функции  $F(A, B, C, D) = (A \text{ AND } B) \text{ OR } (C \text{ XOR } D) \text{ OR } (\text{NOT } A \text{ AND NOT } D)$ .

## 2. Составление таблиц истинности:

Задание 2.1: Для каждой из функций из Задания 1.1 составить таблицу истинности.

Задание 2.2: Составить таблицу истинности для функции  $F(A, B, C) = (A \text{ XOR } B) \text{ AND } (C \text{ OR NOT } A)$ .

\*Задание 2.3: Составить таблицу истинности для функции  $F(A, B, C, D) = (A \text{ AND } B) \text{ OR } (C \text{ XOR } D) \text{ OR } (\text{NOT } A \text{ AND NOT } D)$ .

**Время работы:** 2 часа.

## Практическая работа 7-8

### Исследование цифровых логических элементов в Multisim

#### Цель

Получить практические навыки работы с программой Multisim.

Изучить основные типы цифровых логических элементов и их характеристики.

Построить и смоделировать различные логические схемы, используя изученные логические элементы.

Научиться анализировать работу логических схем и интерпретировать полученные результаты.

## **Теоретическая часть**

Цифровая электроника — это область электроники, занимающаяся обработкой и передачей информации в двоичном коде (0 и 1). Основными элементами цифровых схем являются логические элементы, или вентили, которые реализуют логические операции над сигналами.

Основные типы вентиляей:

**И-вентиль (AND):** выдает высокий уровень сигнала (логическая "1") только тогда, когда все его входы находятся на высоком уровне.

**ИЛИ-вентиль (OR):** выдает высокий уровень сигнала, если хотя бы один из его входов находится на высоком уровне.

**НЕ-вентиль (NOT, инвертор):** выдает высокий уровень сигнала, если на его входе низкий уровень, и наоборот.

**Исключающее ИЛИ-вентиль (XOR):** выдает высокий уровень сигнала, если на его входах разные уровни (один высокий, другой низкий).

**Импликация-вентиль (IMPLICATION):** выдает низкий уровень сигнала только тогда, когда первый вход высокий, а второй низкий.

Логические схемы строятся путем соединения вентиляей. Каждый вентиль представляет собой определенную логическую операцию. Входы и выходы вентиляей соединяются между собой, чтобы реализовать заданную логическую функцию.

Multisim - это популярный программный пакет для моделирования и анализа электронных схем. В нем можно создавать, моделировать и анализировать цифровые схемы, используя виртуальные логические элементы.

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующего занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

### **Рабочие задания**

Часть 1: Знакомство с Multisim

1. Создать новый проект в Multisim.
2. Ознакомиться с основными элементами интерфейса:

Панель инструментов.

Окно схемы.

Окно анализа.

Окно компонентов.

3. Добавить на схему логические элементы (И, ИЛИ, НЕ, XOR, IMPLICATION).

4. Изменить параметры логических элементов (название, цвет, размер).

5. Добавить на схему источники питания (DC Power Supply) и измерительные приборы (Voltmeter, Logic Analyzer).

6. Создать соединительные провода (Wire).

7. Изучить основные функции моделирования (Run, Single Step).

8. Сохранить проект.

Часть 2: Исследование логических элементов

1. И-вентиль:

Создать схему с И-вентилем с двумя входами.

Задать различные комбинации входных сигналов (высокие/низкие) с помощью источников питания.

Проанализировать выходной сигнал И-вентиля с помощью вольтметра.

Построить таблицу истинности И-вентиля.

2. ИЛИ-вентиль:

Повторить задание 1 для ИЛИ-вентиля.

3. НЕ-вентиль:

Создать схему с НЕ-вентилем (инвертором).

Задать различные входные сигналы (высокие/низкие).

Проанализировать выходной сигнал НЕ-вентилля с помощью вольтметра.

Построить таблицу истинности НЕ-вентилля.

#### 4. XOR-вентилль:

Повторить задание 1 для XOR-вентилля.

#### 5. IMPLICATION-вентилль:

Повторить задание 1 для IMPLICATION-вентилля.

### Часть 3: Создание и анализ логических схем

#### 1. Реализация простой логической функции:

Создать схему, реализующую логическую функцию  $F(A, B, C) = (A \text{ AND } B) \text{ OR } (C \text{ AND NOT } A)$ .

Провести моделирование схемы с использованием различных комбинаций входных сигналов.

Проанализировать выходной сигнал схемы с помощью Logic Analyzer.

Составить таблицу истинности для данной функции.

#### 2. Создание схемы, реализующей логический контроллер:

Предложить простой пример логического контроллера (например, система сигнализации).

Разработать логическую схему контроллера с использованием изученных логических элементов.

Провести моделирование схемы и проанализировать её работу.

**Время работы:** 2 часа.

## Практическая работа 9-10

### Минимизация логических функций методом Карно

#### Цель:

Освоить метод Карно для минимизации логических функций.

Научиться строить карты Карно для функций с различным количеством переменных.

Научиться находить минимальные ДНФ и КНФ для заданных логических функций.

Развить навыки логического мышления и умения анализировать логические выражения.

#### Теоретическая часть

##### 1. Логические функции и их представление

Логические функции -это математические функции, которые принимают на вход логические значения (истина/ложь, 1/0) и возвращают логическое значение. Они используются для описания отношений между логическими переменными.

Способы представления логических функций:

Таблицы истинности: показывают все возможные комбинации входных значений и соответствующие выходные значения.

Логические выражения: записывают логические функции с использованием символов логических операций (И, ИЛИ, НЕ, XOR, IMPLICATION) и переменных.

Логические схемы: представляют логические функции с помощью графических элементов (вентилей), которые реализуют логические операции.

##### 2. Минимизация логических функций

Минимизация логических функций - это процесс преобразования логической функции в эквивалентную функцию с минимальным количеством логических операций. Минимизация позволяет:

Упростить логические схемы: схемы, реализующие минимизированные функции, состоят из меньшего количества вентиляей.

Снизить стоимость реализации логических схем: меньшее количество вентиляей значит меньше компонентов и меньше стоимость.

Повысить быстродействие логических схем: схемы с меньшим количеством вентиляей работают быстрее.

Улучшить надежность логических схем: проще отладить схем с меньшим количеством компонентов.

### 3. Метод Карно

Метод Карно - это графический метод минимизации логических функций, основанный на использовании карт Карно. Карта Карно - это таблица, где каждая ячейка соответствует одной комбинации входных значений, а значение в ячейке определяет значение выходной функции для этой комбинации.

Преимущества метода Карно:

Простота и наглядность: метод легко понять и применить.

Эффективность для минимизации функций с небольшим числом переменных: метод дает хорошие результаты для функций с 2-4 переменными.

Возможность учитывать не только логические операции, но и физические ограничения: метод позволяет учитывать ограничения на число вентиляей и их тип.

### 4. Основные понятия

Дизъюнктивная нормальная форма (ДНФ): представление логической функции в виде суммы произведений (конъюнкций) переменных.

Конъюнктивная нормальная форма (КНФ): представление логической функции в виде произведения сумм (дизъюнкций) переменных.

Минимальный член (минтерм): произведение переменных, где каждая переменная входит в виде прямой или инверсной формы.

Максимальный член (макситерм): сумма переменных, где каждая переменная входит в виде прямой или инверсной формы.

## 5. Применение минимизации

Минимизация логических функций широко используется в различных областях:

Цифровая электроника: для создания упрощенных и эффективных схем.

Программирование: для оптимизации логических выражений в коде.

Искусственный интеллект: для создания упрощенных и эффективных логических систем.

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

### **Рабочие задания:**

Часть 1: Построение карт Карно

1. Заданные логические функции:

$$F(A, B, C) = A'BC + AB'C' + ABC + AB'C$$

$$F(A, B, C, D) = A'B'C'D + A'B'CD + A'BCD + AB'C'D + AB'CD + ABCD$$

$$F(A, B, C, D, E) = A'B'C'D'E + A'B'C'DE + A'B'CDE + A'BC'D'E + A'BC'DE + ABC'D'E + A'BCD'E$$

Для каждой из этих функций:

Построить карту Карно.

Описать порядок расположения переменных в столбцах и строках карты.

Заполнить карту Карно значениями функции (0 или 1) для каждой комбинации входных значений.

Часть 2: Минимизация логических функций

1. Минимизация ДНФ:

Для каждой функции из задания 1 найти минимальную ДНФ, используя метод Карно.

Объединить максимальное количество ячеек на карте Карно, содержащих "1".

Записать минимальную ДНФ в виде суммы минимальных членов.

## 2. Минимизация КНФ:

Для каждой функции из задания 1 найти минимальную КНФ, используя метод Карно.

Объединить максимальное количество ячеек на карте Карно, содержащих "0".

Записать минимальную КНФ в виде произведения максимальных членов.

## Часть 3: Анализ и сравнение

### 1. Сравнение исходной функции и минимальной ДНФ/КНФ:

Для каждой функции из задания 2 сравнить исходную формулу с минимальной ДНФ и КНФ.

Проанализировать, как изменилось количество логических операций в результате минимизации.

Обсудить преимущества использования минимальной формы.

**Время работы:** 2 часа.

## Практическая работа 11-12

### Минимизация логических функций методом Карно в Multisim

#### Цель:

Изучить метод Карно для минимизации логических функций.

Научиться строить карты Карно для функций с различным количеством переменных.

Научиться находить минимальные ДНФ и КНФ для заданных логических функций.

Использовать программное обеспечение Multisim для моделирования и проверки работы логических схем, реализующих минимизированные функции.

Развить навыки логического мышления, анализа и решения задач, связанных с минимизацией логических функций.

#### Общие положения

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

## Форма отчета:

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

## Рабочие задания

Часть 1: Построение карт Карно и поиск минимальных ДНФ/КНФ

### 1. Простые функции:

Построить карты Карно для следующих функций:

$$F(A,B,C) = \Sigma m(0,1,2,3,4,5)$$

$$F(A,B,C,D) = \Sigma m(0,1,3,5,7,9,11,13,15)$$

$$F(A,B,C) = \text{ПМ}(1,3,5,7)$$

Найти минимальную ДНФ и КНФ для каждой функции.

Проверить полученные результаты, используя свойства логических операций и алгебраические преобразования.

### 2. Функции с дополнительными условиями:

$F(A,B,C,D) = \Sigma m(0,1,3,5,7,9,11,13,15) + d(2,4,6,8,10,12)$  (d - "незначащие" переменные). Найти минимальную ДНФ и КНФ для этой функции, учитывая "незначащие" переменные.

$F(A,B,C,D) = \Sigma m(0,1,3,5,7,9,11,13,15) + \Sigma d(2,4,6,8,10,12) + \Sigma m(14,15)$  (d - "незначащие", m - "запрещённые"). Найти минимальную ДНФ и КНФ для этой функции, учитывая "незначащие" и "запрещённые" переменные.

Дополнительно:

Объяснить, как "незначащие" и "запрещённые" переменные влияют на минимизацию функции.

Построить карты Карно для функций с "незначащими" и "запрещёнными" переменными и сравнить их с картами для функций без таких переменных.

### 3. Сложные функции:

$F(A,B,C,D) = (A+B) \cdot (C+D) + (A \cdot B) \cdot (C \cdot D)$ . Найти минимальную ДНФ и КНФ для этой функции.

$F(A,B,C,D,E) = (A \cdot B) \cdot (C+D) + (E \cdot A) \cdot (B+C)$ . Найти минимальную ДНФ и КНФ для этой функции.

## Часть 2: Моделирование логических схем в Multisim

### 1. Реализация минимальных ДНФ/КНФ:

Создать логические схемы в Multisim для реализованных в предыдущих заданиях минимальных ДНФ и КНФ.

Использовать логические элементы (И, ИЛИ, НЕ) для построения схем.

Проверить работу схем с помощью генератора сигналов и осциллографа.

Проанализировать влияние изменения числа логических элементов на время работы схемы.

### 2. Сравнение схем:

Сравнить схемы, реализующие исходную функцию и её минимальную ДНФ/КНФ.

Проанализировать влияние минимизации функции на количество логических элементов и время работы схемы.

Сделать выводы о преимуществах и недостатках минимизации логических функций.

**Время работы:** 2 часа.

## **Практическая работа 13-14**

### **Построение логических схем в заданном базисе**

#### **Цель задания:**

- Освоить основы построения логических схем в заданном базисе.
- Научиться преобразовывать логические выражения в схемы, используя заданные логические элементы.
- Развивать навыки анализа и синтеза логических схем.
- Повысить уровень понимания работы логических элементов и их комбинаций.

#### **Теоретическая часть:**

##### 1. Логические схемы и базисы

Логическая схема - это графическое представление логической функции. Она состоит из логических элементов, соединенных линиями, которые представляют сигналы.

Логический элемент - это устройство, реализующее определенную логическую функцию (например, инверсия, конъюнкция, дизъюнкция).

Базис - это набор логических элементов, из которых можно построить любую логическую схему.

Примеры базисов:

- Базис Шеффера: Состоит из элемента "НЕ-И" (NAND).
- Базис Пирса: Состоит из элемента "НЕ-ИЛИ" (NOR).
- Базис "И-НЕ": Состоит из элементов "И" и "НЕ".
- Базис "ИЛИ-НЕ": Состоит из элементов "ИЛИ" и "НЕ".

## 2. Построение логических схем в заданном базисе

Процесс построения логической схемы в заданном базисе включает в себя следующие этапы:

1. Получение логического выражения: Логическое выражение может быть задано в виде таблицы истинности, словесного описания или уравнения.

2. Минимизация логического выражения: Минимизация упрощает схему и сокращает количество элементов.

3. Преобразование логического выражения в схему: Логическое выражение разбивается на подвыражения, которые реализуются с помощью элементов из заданного базиса.

4. Соединение элементов: Элементы соединяются между собой в соответствии с логическим выражением.

Примеры преобразований:

- Преобразование "ИЛИ" в базисе "И-НЕ":

$$- 'A + B = \text{NOT}(\text{NOT}(A) * \text{NOT}(B))'$$

- Преобразование "И" в базисе "ИЛИ-НЕ":

$$- \text{`A * B = NOT(NOT(A) + NOT(B))`}$$

### 3. Методы минимизации логических выражений

- Метод Карно: Использует диаграммы Карно для визуального представления логических функций и упрощения выражений.

- Метод Квайна-Мак-Класки: Использует алгоритм для минимизации выражений.

- Метод минимизации методом алгебры логики: Использует законы алгебры логики для преобразования выражений в более простую форму.

### 4. Примеры практических задач

1. Построение схемы сумматора: Сумматор - это устройство, которое складывает два числа.

2. Построение схемы дешифратора: Дешифратор - это устройство, которое преобразует двоичный код в сигнал, соответствующий одному из возможных состояний.

3. Построение схемы регистра: Регистр - это устройство, которое хранит информацию в виде двоичных цифр.

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

### **Рабочие задания**

Задание 1: Преобразование логических выражений

1. Преобразуйте следующие логические выражения в базис Шеффера (NAND):

$$A + B$$

$$A * B$$

$$\neg A$$

$$A \oplus B$$

2. Преобразуйте следующие логические выражения в базис Пирса (NOR):

$$A + B$$

$$A * B$$

$$\neg A$$

$$A \oplus B$$

Задание 2: Построение логических схем

1. Постройте логическую схему для функции  $F(A,B,C) = (A * B) + (\neg C)$  в базисе "И-НЕ".

2. Постройте логическую схему для функции  $F(A,B,C,D) = (A + B) * (\neg C + D)$  в базисе "ИЛИ-НЕ".

3. Постройте схему сумматора на двух битах в базисе Шеффера. Для суммирования используйте таблицу истинности.

4. Постройте схему дешифратора на 2 входа (4 выхода) в базисе Пирса.

Задание 3: Минимизация логических выражений

1. Минимизируйте следующее логическое выражение методом Карно:

$$* F(A,B,C,D) = (\neg A * B * C * D) + (A * \neg B * C * D) + (A * B * \neg C * D) + (A * B * C * \neg D)$$

2. Минимизируйте следующее логическое выражение методом Квайна-Мак-Класки:

$$F(A,B,C,D) = (\neg A * B * C * D) + (A * \neg B * C * D) + (A * B * \neg C * D) + (A * B * C * \neg D) + (A * B * C * D)$$

3. Минимизируйте следующее логическое выражение методом алгебры логики:

$$F(A,B,C) = (A * B) + (\neg C * B) + (A * \neg C)$$

**Время работы:** 2 часа

## Практическая работа 15

### Моделирование и анализ дешифраторов

**Цель работы:**

Изучить принцип работы дешифратора и его основные типы.

Создать модель дешифратора с помощью логических элементов.

Провести анализ полученного дешифратора, проверив его работоспособность на конкретном примере.

### **Теоретическая часть:**

Дешифратор - это логическая схема, которая преобразует двоичный код на своем входе в единственный активный выходной сигнал, соответствующий этому коду. Дешифраторы используются для выбора одной из нескольких линий данных, например, для адресации памяти, управления периферийными устройствами или для управления различными компонентами в цифровых системах.

Типы дешифраторов:

Дешифратор с двоичным кодом: преобразует двоичный код на своем входе в единственный активный выходной сигнал, соответствующий этому коду. Например, дешифратор 2-к-4 имеет два входных сигнала и четыре выходных сигнала.

Дешифратор с BCD кодом: преобразует код BCD (Binary-Coded Decimal) на своем входе в единственный активный выходной сигнал, соответствующий десятичной цифре, закодированной в BCD коде.

Дешифратор с приоритетом: выбирает только один выходной сигнал, соответствующий входному коду с наивысшим приоритетом.

Мультиплексор: комбинирует несколько входных сигналов на один выходной, выбирая активный входной сигнал по адресу.

Логические элементы:

НЕ: инвертирует входной сигнал.

И: дает логическую 1 на выходе только тогда, когда все входные сигналы равны 1.

ИЛИ: дает логическую 1 на выходе, если хотя бы один входной сигнал равен 1.

И-НЕ: инвертирует выходную логическую операцию И.

ИЛИ-НЕ: инвертирует выходную логическую операцию ИЛИ.

Реализация дешифраторов:

Дешифраторы могут быть реализованы с помощью таблиц истинности и логических схем, которые представляют собой комбинацию логических элементов.

Пример:

Дешифратор 2-к-4:

| Вход А | Вход В | Выход 0 | Выход 1 | Выход 2 | Выход 3 |

|---|---|---|---|---|

| 0 | 0 | 1 | 0 | 0 | 0 |

| 0 | 1 | 0 | 1 | 0 | 0 |

| 1 | 0 | 0 | 0 | 1 | 0 |

| 1 | 1 | 0 | 0 | 0 | 1 |

Логическая схема дешифратора 2-к-4 может быть реализована с помощью четырех элементов И-НЕ, каждый из которых получает комбинацию входных сигналов А и В, соответствующую определенному выходу.

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

### **Рабочие задания:**

#### 1. Моделирование простого дешифратора:

Спроектируйте дешифратор 2-в-4 с использованием логических вентилей (ИЛИ-НЕ, И).

Опишите принцип работы дешифратора и покажите таблицу истинности.

Разработайте схему дешифратора в любой программе для проектирования схем (например, Logisim, Multisim).

#### 2. Анализ дешифратора:

Задан дешифратор 3-в-8.

Определите количество выходов дешифратора.

Опишите принцип работы дешифратора.

Постройте таблицу истинности для дешифратора.

### 3. Реализация дешифратора с использованием микросхем:

Используйте микросхему CD4511 (дешифратор 2-в-4) для реализации схемы дешифратора.

Подключите к микросхеме светодиоды для визуализации работы дешифратора.

Проведите эксперимент, варьируя входные сигналы и наблюдая за изменением состояния светодиодов.

**Время работы:** 2 часа

## **Практическая работа 16**

### **Моделирование и анализ шифраторов**

#### **Цель работы:**

Изучить принципы работы различных типов шифраторов: освоить основные понятия криптографии, алгоритмы шифрования и дешифрования, а также способы анализа криптографической стойкости.

Разработать и реализовать модели шифраторов: получить практический опыт в разработке программного обеспечения для шифрования данных.

Провести анализ стойкости разработанных шифраторов: оценить уровень защищенности шифраторов и выявить их уязвимости.

#### **Теория:**

Шифрование – это процесс преобразования информации в нечитаемый вид, чтобы защитить ее от несанкционированного доступа. Шифраторы – это

алгоритмы, которые используются для выполнения этого процесса. Моделирование и анализ шифраторов играют важную роль в криптографии, помогая нам понять их сильные и слабые стороны, а также их безопасность.

Основные этапы моделирования и анализа шифраторов:

#### 1. Модель шифратора:

Описание алгоритма шифрования и дешифрования.

Определение входных и выходных данных.

Описание ключей и их роли.

Определение криптографических операций, используемых в шифраторе.

#### 2. Анализ стойкости шифратора:

Статистический анализ: Анализ частоты символов, биграмм, триграмм и т. д., чтобы выявить закономерности, которые могут помочь злоумышленнику дешифровать сообщение.

Анализ дифференциальных характеристик: Изучение влияния небольших изменений в входных данных на выходные данные, чтобы определить, насколько легко отличить зашифрованные сообщения друг от друга.

Линейный криптоанализ: Поиск линейных зависимостей между ключами, входными данными и выходными данными, чтобы сократить пространство поиска ключей.

Атака с использованием известных открытых текстов: Использование известных пар открытых текстов и соответствующих зашифрованных текстов для восстановления ключа.

Атака с использованием выбранных открытых текстов: Возможность выбора открытых текстов для шифрования и использования полученных зашифрованных текстов для восстановления ключа.

Атака с использованием выбранных зашифрованных текстов: Возможность выбора зашифрованных текстов для дешифрования и использования полученных открытых текстов для восстановления ключа.

### 3. Оптимизация и улучшение шифратора:

Увеличение сложности алгоритма, чтобы сделать его более устойчивым к криптоанализу.

Использование более сложных ключей и алгоритмов для повышения безопасности.

Введение дополнительных криптографических операций, чтобы затруднить анализ шифратора.

Преимущества моделирования и анализа шифраторов:

Повышение уверенности в безопасности шифратора.

Обнаружение и устранение уязвимостей в шифраторах.

Разработка более надежных и эффективных шифраторов.

Инструменты для моделирования и анализа шифраторов:

Программные инструменты для моделирования криптографических алгоритмов.

Специализированные библиотеки для анализа шифраторов.

Программное обеспечение для проведения криптоаналитических атак.

Примеры моделей и анализа шифраторов:

Модель DES (Data Encryption Standard) и его анализ методом дифференциального криптоанализа.

Модель AES (Advanced Encryption Standard) и его анализ методом линейного криптоанализа.

Модель RSA (Rivest-Shamir-Adleman) и его анализ методом факторизации больших чисел.

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующего занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

### **Рабочие задания:**

#### 1. Изучение теории:

Изучить основные понятия криптографии: шифрование, дешифрование, ключ, алгоритм, стойкость, криптоанализ.

Изучить алгоритмы симметричного шифрования:

Шифр Цезаря

Шифр простой замены

Шифр Виженера

Шифр Playfair

Изучить методы криптоанализа:

Частотный анализ

Статистический анализ

Атака по известному открытому тексту

Атака по выбранному открытому тексту

Атака по выбранному шифротексту

2. Разработка моделей шифраторов:

Выбрать два алгоритма шифрования (по одному симметричному и асимметричному) для реализации.

Разработать программное обеспечение (на языке Python или другом языке программирования), реализующее выбранные алгоритмы шифрования и дешифрования.

Осуществить тестирование разработанных шифраторов на произвольных тестовых данных.

3. Анализ стойкости шифраторов:

Провести криптоанализ разработанных шифраторов, используя известные методы и инструменты.

Оценить стойкость шифраторов к различным типам атак:

Атака по известному открытому тексту

Атака по выбранному открытому тексту

Атака по выбранному шифротексту

Определить уязвимости разработанных шифраторов и предложить варианты их устранения.

Проанализировать результаты анализа и сформулировать выводы о стойкости разработанных шифраторов.

**Время работы:** 2 часа

## **Практическая работа 17**

### **Моделирование и анализ мультиплексоров**

#### **Цель работы:**

Изучить принципы работы мультиплексоров и их применение в цифровых системах.

Овладеть навыками моделирования мультиплексоров в специализированном программном обеспечении.

Провести анализ характеристик различных типов мультиплексоров.

Получить практические навыки реализации мультиплексоров в различных логических семействах.

#### **Теория:**

##### **Введение:**

Мультиплексор (MUX) - это электронное устройство, которое выбирает один из нескольких входных сигналов и направляет его на единственный выходной сигнал. MUX широко используются в цифровых системах для реализации таких задач, как:

**Выбор данных:** Выбор одного из нескольких источников данных для передачи.

**Переключение каналов:** Переключение между несколькими входами, например, в системах связи.

Адресная дешифрация: Преобразование адресного кода в сигнал выбора для доступа к памяти или другому устройству.

Принцип работы:

Мультиплексор состоит из следующих основных элементов:

Входные сигналы ( $n$ ): Количество входных сигналов, которые могут быть выбраны.

Сигнал выбора ( $m$ ): Управляет выбором одного из входных сигналов. Число бит в сигнале выбора равно  $\log_2(n)$ , где  $n$  - количество входных сигналов.

Выходной сигнал ( $1$ ): Выходной сигнал, который представляет выбранный входной сигнал.

Логическая реализация:

Мультиплексор может быть реализован с помощью логических вентилях AND, OR и NOT. Логика работы заключается в следующем:

Вентили AND: Каждый входной сигнал связывается с отдельным вентилям AND.

Вентили NOT: Сигнал выбора преобразуется в набор сигналов NOT, соответствующих каждому входному сигналу.

Вентиль OR: Выходные сигналы от вентилях AND объединяются в единый выходной сигнал с помощью вентилях OR.

Моделирование и анализ:

Для моделирования и анализа мультиплексоров используются различные методы, в том числе:

Логические схемы: Для представления логической структуры мультиплексора.

Таблицы истинности: Для определения выходного сигнала для каждой возможной комбинации входных и сигналов выбора.

Симуляция: Для проверки функциональности мультиплексора в различных условиях работы.

Анализ временных характеристик: Для оценки задержек и времени переключения мультиплексора.

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующего занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

## **Рабочие задания**

### **1. Теоретический обзор:**

Изучить принцип работы мультиплексора, его основные элементы и характеристики.

Исследовать различные типы мультиплексоров (аналоговые, цифровые, мультиплексированные).

Ознакомиться с применением мультиплексоров в различных цифровых системах (например, выбор данных, управление сигналами).

### **2. Моделирование:**

Выбрать подходящее программное обеспечение для моделирования (например, Logisim, Multisim, Vivado).

Разработать модель мультиплексора заданной конфигурации (число входов, выходов, управление).

Провести моделирование работы мультиплексора с различными входными сигналами и вариантами управления.

Записать и проанализировать результаты моделирования.

### **3. Анализ характеристик:**

Сравнить характеристики различных типов мультиплексоров (скорость, задержка, потребление энергии).

Определить факторы, влияющие на выбор мультиплексора для конкретного приложения.

### **4. Практическая реализация:**

Реализовать мультиплексор заданной конфигурации с использованием микросхем логических вентилей (например, 74LS00, 74LS138).

Провести тестирование работоспособности реализованного мультиплексора.

Описать результаты реализации и сравнить их с результатами моделирования.

**Дополнительные задания:**

Разработать схему и провести моделирование мультиплексора с использованием различных логических семейств (TTL, CMOS).

Изучить применение мультиплексоров в системах передачи данных и их роль в сетевых устройствах.

Разработать алгоритм и программу для автоматического синтеза мультиплексора заданной конфигурации.

**Время работы:** 2 часа

## **Практическая работа 18**

### **Моделирование и анализ демультиплексоров**

**Цель работы:**

Цель работы - изучение принципов работы демультиплексора, моделирование различных типов демультиплексоров с использованием логических элементов и анализ их функциональных характеристик.

**Теоретическая часть:**

Демультиплексор - это логическое устройство, которое позволяет передать один входной сигнал на один из нескольких выходов. Демультиплексор работает по принципу выбора одного из нескольких выходов с помощью управляющего сигнала.

**Принцип работы:**

Демультимплексор имеет один входной сигнал (D) и несколько выходных сигналов ( $Y_0, Y_1, Y_2, \dots, Y_n$ ). Выбор выходного сигнала осуществляется с помощью управляющего сигнала (S). Управляющий сигнал имеет  $n$  битов, где  $n$  - число выходов.

Логика работы демультимплексора заключается в том, что на выбранный выходной сигнал передается входной сигнал, а на остальные выходы подается логический "0".

Типы демультимплексоров:

Существуют различные типы демультимплексоров, которые отличаются по способу реализации:

Демультимплексор на основе логических вентилей: В этом случае демультимплексор реализуется с помощью комбинации логических вентилей, таких как AND, NOT и OR.

Демультимплексор на основе транзисторов: Этот тип демультимплексора использует транзисторы для переключения входного сигнала на выбранный выход.

Демультимплексор на основе микросхем: В этом случае демультимплексор реализован в виде микросхемы, которая содержит все необходимые логические элементы.

Применение демультимплексоров:

Демультимплексоры широко применяются в различных электронных системах, например:

Коммутация сигналов: Демультимплексор может использоваться для переключения сигнала на один из нескольких устройств.

Адресация памяти: В системах памяти демультиплексор используется для выбора конкретной ячейки памяти.

Передача данных: Демультиплексор может использоваться для передачи данных от одного источника к нескольким приемникам.

Разделение данных: Демультиплексор может использоваться для разделения потока данных на несколько потоков.

Основные характеристики демультиплексора:

Количество выходов (n): Количество выходов демультиплексора.

Скорость работы: Скорость, с которой демультиплексор может переключаться между выходами.

Потребление мощности: Мощность, потребляемая демультиплексором.

Задержка сигнала: Время, которое требуется сигналу для прохождения через демультиплексор.

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующего занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

### **Рабочие задачи**

Изучить теоретические основы работы демультиплексора.

Построить логическую схему демультиплексора на основе логических вентилей.

Моделировать работу демультиплексора в выбранной программной среде (например, Logisim, Multisim).

Провести анализ функциональных характеристик моделируемого демультиплексора, таких как скорость работы, количество выходов, потребление мощности.

Сравнить различные типы демультиплексоров по их характеристикам.

**Время работы:** 2 часа

## **Практическая работа 19**

### **Моделирование и анализ сумматоров**

#### **Цель работы:**

Цель данной работы – изучение принципов работы различных типов сумматоров, моделирование их функционирования с использованием программного обеспечения и анализ полученных результатов с целью сравнительной оценки их эффективности и выявления ключевых особенностей.

Теоретическая часть:

Сумматоры – это основные логические элементы цифровых схем, предназначенные для выполнения операции сложения двух или более двоичных чисел. Они широко используются в процессорах, контроллерах, арифметико-логических устройствах (АЛУ) и других цифровых системах. Существует множество типов сумматоров, отличающихся по своей структуре, скорости работы и сложности реализации.

#### Полусумматор

Простейший тип сумматора, выполняющий сложение двух одноразрядных двоичных чисел.

Имеет два входа:  $A$  и  $B$ , и два выхода:  $S$  (сумма) и  $C$  (перенос).

Сумма  $S$  формируется по модулю 2 (XOR), а перенос  $C$  – по логическому "И" (AND).

#### Полный сумматор:

Выполняет сложение двух одноразрядных чисел, а также учитывает перенос из предыдущего разряда.

Имеет три входа:  $A$ ,  $B$  и  $C_{in}$  (входящий перенос), и два выхода:  $S$  (сумма) и  $C_{out}$  (выходящий перенос).

Выходная сумма формируется с использованием XOR, а выходной перенос – с использованием комбинации XOR и AND.

#### Многоразрядные сумматоры:

Состоят из нескольких полных сумматоров, соединенных каскадом.

Каждый полный сумматор обрабатывает один разряд входных чисел, а перенос передается в следующий разряд.

Существуют различные схемы соединения полных сумматоров, например, параллельная и последовательная.

#### Быстрые сумматоры:

Используют различные оптимизированные архитектуры для ускорения операции сложения.

Примеры быстрых сумматоров: сумматоры с переносом по перескокам (Carry Lookahead), сумматоры с переносом по блокированию (Carry-Block), сумматоры с переносом по строкам (Carry-Select).

#### Критерии сравнения сумматоров:

Скорость работы: время выполнения операции сложения.

Сложность реализации: количество логических элементов, требуемых для реализации сумматора.

Потребляемая мощность: количество энергии, потребляемой сумматором.

Стоимость: цена реализации сумматора.

Моделирование сумматоров:

Моделирование позволяет изучать работу сумматоров без необходимости их физической реализации.

Для моделирования можно использовать специализированные программы для проектирования электронных схем (например, Logisim, Multisim, Proteus).

Моделирование позволяет проводить эксперименты с различными типами сумматоров, изменять параметры и наблюдать за результатами.

Анализ результатов моделирования:

Анализ полученных результатов моделирования позволяет сравнить различные типы сумматоров по критериям, указанным выше.

Можно определить наиболее эффективный тип сумматора для конкретного приложения.

Анализ позволяет выявлять ключевые особенности и преимущества различных архитектур сумматоров.

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

### **Рабочие задания**

#### 1. Моделирование и анализ одноразрядного сумматора:

Спроектируйте и смоделируйте одноразрядный сумматор с помощью логических вентилях (И, ИЛИ, НЕ, Исключающее ИЛИ).

Проведите анализ работы сумматора, проверив его на всех возможных комбинациях входных сигналов.

Постройте таблицу истинности и логическую схему сумматора.

Определите максимальную задержку сигнала через сумматор.

#### 2. Моделирование и анализ многоразрядного сумматора:

Спроектируйте и смоделируйте многоразрядный сумматор (например, 4-разрядный) с использованием одноразрядных сумматоров в качестве строительных блоков.

Используйте разные методы каскадирования одноразрядных сумматоров:

Последовательное соединение.

Параллельное соединение с использованием полных сумматоров.

Проведите анализ работы многоразрядного сумматора, проверив его на различных входных данных.

Постройте таблицу истинности и логическую схему многоразрядного сумматора.

Определите максимальную задержку сигнала через сумматор.

### 3. Анализ сумматоров с использованием языка Verilog/VHDL:

Разработайте описание одноразрядного и многоразрядного сумматора на языке Verilog/VHDL.

Проведите симуляцию работы сумматора с помощью симулятора Verilog/VHDL (например, ModelSim, Icarus Verilog).

Проанализируйте результаты симуляции, проверив корректность работы сумматора.

Определите время работы сумматора и влияние задержек вентиляей.

### 4. Исследование сумматоров с использованием различных архитектур:

Изучите различные архитектуры сумматоров (например, сумматоры carry-lookahead, carry-skip, ripple carry).

Спроектируйте и смоделируйте сумматор с использованием одной из выбранных архитектур.

Сравните производительность сумматоров с различными архитектурами по времени работы, занимаемому пространству и энергопотреблению.

### 5. Практическое применение сумматоров

Разработайте систему, использующую сумматор для выполнения определенной задачи (например, калькулятор, счетчик, таймер).

Проведите симуляцию работы системы и проверьте ее корректность.

Документируйте проект, описав использованные технологии, архитектуру и результаты симуляции.

**Время работы:** 2 часа

## **Практическая работа 20**

### **Моделирование и анализ RS-триггера**

#### **Цели:**

Понимание принципов работы RS-триггера: изучить логическую структуру и принцип функционирования RS-триггера, разобраться в типах RS-триггеров (с прямым/обратным управлением).

Разработка и анализ схем RS-триггеров: уметь самостоятельно проектировать RS-триггер, используя логические вентиля, и анализировать его поведение с помощью таблицы истинности и временных диаграмм.

Понимание особенностей RS-триггера: выявить ограничения RS-триггера, например, возможность возникновения запрещенного состояния, и разобраться в причинах его возникновения.

Применение RS-триггера в практических задачах: научиться использовать RS-триггер в качестве элемента памяти в более сложных цифровых схемах, например, в счетчиках, регистрах и т.д.

### **Теория:**

RS-триггер - это базовый элемент цифровой логики, который реализует функцию памяти. Он состоит из двух взаимно связанных логических элементов, обычно инверторов или NAND/NOR вентилей. RS-триггер может находиться в одном из двух устойчивых состояний:

Состояние 0 ( $S = 0, R = 0$ ): выходы триггера сохраняют свое предыдущее значение.

Состояние 1 ( $S = 1, R = 0$ ): выходы триггера устанавливаются в значение 1.

Состояние 2 ( $S = 0, R = 1$ ): выходы триггера устанавливаются в значение 0.

Запрещенное состояние ( $S = 1, R = 1$ ): поведение триггера неопределенно, так как оба входа одновременно стремятся установить разные значения на выходах.

Типы RS-триггеров:

RS-триггер с прямым управлением: выходы триггера устанавливаются в 1 или 0 непосредственно при подаче сигнала на вход S или R.

RS-триггер с обратным управлением: выходы триггера устанавливаются в 1 или 0 при снятии сигнала с входа S или R.

Анализ RS-триггера:

Таблица истинности: показывает, как меняются выходные сигналы триггера в зависимости от состояний входных сигналов.

Временные диаграммы: визуализируют изменение сигналов на входах и выходах триггера во времени

Моделирование RS-триггера:

Логические схемы: используются для описания структуры RS-триггера с помощью логических вентилей.

Программы моделирования: позволяют создавать и анализировать схемы RS-триггера, моделировать их поведение и получать временные диаграммы.

## Применения RS-триггера:

Элементы памяти: используются для хранения информации в цифровых схемах.

Счетчики: RS-триггеры могут использоваться для построения счетчиков, которые фиксируют количество импульсов.

Регистры: RS-триггеры используются в регистрах для хранения и обработки данных.

## **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующего занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

## **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

## **Рабочие задания**

1. Логическое моделирование:

Задание 1:

С помощью таблицы истинности и логических уравнений опишите поведение RS-триггера (с использованием вентилей НЕ-ИЛИ).

Проведите логическое моделирование RS-триггера в программе логического моделирования (например, Logisim, Multisim) для различных комбинаций входных сигналов (R, S).

Проанализируйте состояния выхода триггера (Q, Q') в зависимости от входных сигналов и предыдущего состояния.

#### Задание 2:

С помощью таблицы истинности и логических уравнений опишите поведение RS-триггера (с использованием вентилей И-НЕ).

Проведите логическое моделирование RS-триггера в программе логического моделирования (например, Logisim, Multisim) для различных комбинаций входных сигналов (R, S).

Проанализируйте состояния выхода триггера (Q, Q') в зависимости от входных сигналов и предыдущего состояния.

Сравните полученные результаты с моделированием RS-триггера с использованием вентилей НЕ-ИЛИ.

#### 2. Анализ временных диаграмм:

##### Задание 3:

Используя программу логического моделирования, создайте временную диаграмму для RS-триггера (с использованием вентилей НЕ-ИЛИ или И-НЕ).

Примените различные входные сигналы (R, S) с различными временными интервалами.

Проанализируйте временные диаграммы сигналов Q, Q' и входных сигналов R, S.

Определите, как изменения входных сигналов влияют на выходные сигналы триггера.

##### Задание 4:

С помощью временных диаграмм определите время установления и время сброса для RS-триггера (с использованием вентилей НЕ-ИЛИ или И-НЕ).

Измерьте эти временные характеристики для разных схем RS-триггеров и сравните результаты.

Объясните, как временные характеристики влияют на работу триггера в реальных схемах.

### 3. Практическое моделирование (на макетной плате):

#### Задание 5:

Соберите макетную схему RS-триггера (с использованием вентилей НЕ-ИЛИ или И-НЕ) на макетной плате.

Подключите к схеме переключатели (для имитации входных сигналов R, S) и светодиоды (для отображения состояний Q, Q').

Экспериментально проверьте работу схемы для различных комбинаций входных сигналов.

Сравните полученные результаты с результатами логического моделирования и анализом временных диаграмм.

**Время работы:** 2 часа

## Практическая работа 21

### Моделирование и анализ D-триггера

#### Цели:

1. Построить модель D-триггера: изучить принцип работы D-триггера, его основные характеристики, и реализовать его модель на основе логических элементов.

2. Анализировать поведение D-триггера: провести анализ работы D-триггера, включая его временные диаграммы, режимы работы, и характеристики задержки.

3. Применить полученные знания для проектирования цифровых систем: продемонстрировать, как D-триггер может быть использован для реализации различных цифровых схем, таких как регистры, счетчики и память.

#### Теоретическая часть

##### 1. Основные понятия

Триггер: это элементарный элемент цифровой электроники, который может находиться в одном из двух устойчивых состояний (0 или 1).

D-триггер: тип триггера, который записывает значение сигнала D на своем выходе Q в момент прихода тактового импульса.

Тактовый сигнал: специальный сигнал, который синхронизирует работу триггера.

Синхронные триггеры: триггеры, которые изменяют свое состояние только при приходе тактового импульса.

## 2. Принцип работы D-триггера

D-триггер состоит из комбинационной логики, которая управляет переключением между двумя состояниями, и запоминающего элемента, который хранит состояние.

Схема D-триггера:

D-триггер может быть построен из нескольких логических элементов, например, из двух SR-триггеров. Основными элементами D-триггера являются:

D-вход: входной сигнал, который определяет значение, которое будет записано в триггер.

Q-выход: выходной сигнал, который представляет текущее состояние триггера.

Q'-выход: инвертированный выходной сигнал.

Тактовый вход: вход, который управляет записью значения D в триггер.

## 3. Режимы работы D-триггера

D-триггер имеет два основных режима работы:

Режим записи: когда тактовый сигнал активен, значение D записывается в триггер.

Режим хранения: когда тактовый сигнал неактивен, триггер сохраняет свое предыдущее состояние.

## 4. Характеристики задержки

Время установления: время, которое требуется D-триггеру для перехода в новое состояние после прихода тактового импульса.

Время удержания: время, которое требуется D-входу для сохранения своего значения после прихода тактового импульса.

## 5. Применение D-триггеров

D-триггеры широко используются в цифровых системах для реализации следующих функций:

Регистры: последовательность D-триггеров, которые могут хранить данные.

Счетчики: схемы, которые могут подсчитывать импульсы.

Память: для хранения информации.

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

### **Рабочие задания**

1. С помощью логического симулятора (например, Logisim, Multisim, Quartus) создать схему D-триггера на основе базовых логических элементов (И, ИЛИ, НЕ).

Построить таблицу истинности для схемы, показывающую состояния триггера в зависимости от входных сигналов D и CLK.

Провести симуляцию работы схемы с различными комбинациями входных сигналов, наблюдая за изменениями состояния триггера.

Зафиксировать на диаграмме временных диаграмм сигналы D, CLK, Q и Q', иллюстрирующие работу триггера в различных режимах.

Проанализировать работу триггера, выявив его основные характеристики: время установления, время сброса, время задержки распространения.

## 2. Моделирование D-триггера с помощью языка описания аппаратуры (VHDL/Verilog)

Разработать VHDL/Verilog код D-триггера.

Провести симуляцию разработанного кода с помощью симулятора VHDL/Verilog.

Проверить работоспособность триггера, подавая различные комбинации входных сигналов и анализируя состояние выходов.

Сравнить полученные результаты с результатами моделирования в логической схеме (задание 1).

## 3. Анализ работы D-триггера с использованием временных диаграмм

Построить диаграмму временных диаграмм работы D-триггера с учетом задержки распространения сигнала.

Наблюдать, как задержка распространения влияет на состояние выходов триггера в зависимости от частоты тактового сигнала.

Проанализировать влияние задержки распространения на скорость работы схемы, построенной на основе D-триггеров.

## 4. Реализация D-триггера с использованием микросхем

Задание:

Составить схему D-триггера на основе доступных микросхем (например, 74LS74, 4013, CD4013).

Собрать схему на макетной плате и протестировать ее работу с помощью осциллографа.

Провести сравнительный анализ полученных результатов с результатами моделирования (задание 1 и 2).

## 5. Практическое применение D-триггера

Выбрать практическое применение D-триггера (например, регистр сдвига, счетчик, таймер).

Разработать схему на основе D-триггера, реализующую выбранную функцию.

Собрать схему на макетной плате и проверить ее работоспособность.

Проанализировать полученные результаты и сделать выводы о практическом применении D-триггера в выбранной схеме.

**Время работы:** 2 часа

## **Практическая работа 22**

### **Моделирование и анализ JK-триггера**

#### **Цели**

1. Познакомиться с принципом работы JK-триггера. Изучить его логическую структуру, входные и выходные сигналы, функции переключения.
2. Овладеть методами моделирования JK-триггера. Изучить различные инструменты моделирования, такие как логические схемы, временные диаграммы, таблицы истинности.
3. Провести анализ характеристик JK-триггера. Исследовать его режимы работы, временные параметры, возможности синхронизации.
4. Приобрести навыки построения и анализа схем с использованием JK-триггеров. Понять, как JK-триггеры применяются в цифровых системах для реализации различных функций.

#### **Теоретическая часть**

##### **1. Введение в JK-триггеры**

JK-триггер - это синхронный элемент памяти, который может сохранять логическое значение 0 или 1. В отличие от других типов триггеров (SR, D), JK-триггер позволяет реализовать все возможные комбинации переключения состояний, включая сохранение текущего значения.

##### **2. Логическая структура JK-триггера**

JK-триггер обычно реализуется на основе двух RS-триггеров, соединенных в цепь обратной связи. Он имеет следующие входы:

J (set): Устанавливает выход в логическую 1.

K (reset): Сбрасывает выход в логический 0.

Clock (CP): Тактовый сигнал, управляющий переключением триггера.

И один выход:

Q: Выходной сигнал, отражающий текущее состояние триггера.

### 3. Режимы работы JK-триггера

JK-триггер может работать в нескольких режимах:

Нормальный режим: Переключение происходит на положительном фронте тактового сигнала.

Инверсный режим: Переключение происходит на отрицательном фронте тактового сигнала.

### 4. Временные параметры JK-триггера

Время задержки установки ( $t_{su}$ ): Время, которое должно пройти между установкой сигнала на вход J и подачей тактового сигнала.

Время задержки сброса ( $t_{sh}$ ): Время, которое должно пройти между установкой сигнала на вход K и подачей тактового сигнала.

Время задержки переключения ( $t_{pd}$ ): Время, которое проходит от подачи тактового сигнала до момента установления нового значения на выходе.

### 5. Моделирование JK-триггера

Для моделирования JK-триггера используются различные инструменты:

Логические схемы: Визуальное представление триггера с использованием стандартных логических элементов.

Временные диаграммы: Графическое отображение изменения сигналов во времени.

Таблицы истинности: Описание поведения триггера в зависимости от входных сигналов.

Специальные программные среды: (например, Multisim, LTspice) для создания и анализа цифровых схем.

### 6. Применение JK-триггеров

JK-триггеры применяются в различных цифровых системах, таких как:

Счетчики: Для подсчета импульсов.

Регистры: Для хранения и передачи данных.

Автоматы: Для реализации логических функций и принятия решений.

Цифровые часы: Для отображения времени.

Другие логические схемы: Для реализации различных функций.

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

### **Рабочие задания**

Задание 1: Моделирование JK-триггера с помощью логических элементов

1. Схема:

Создайте схему JK-триггера с помощью логических элементов (И-НЕ, ИЛИ-НЕ).

Включите в схему сигналы J, K, Clock, Reset (если требуется).

2. Моделирование:

С помощью моделирования (например, в программе Multisim, LTSpice) проследите за изменением состояния триггера в зависимости от входных сигналов J, K, Clock.

Проведите тестирование для различных комбинаций сигналов J, K и состояния Clock.

Задокументируйте результаты, описав состояние триггера для каждого временного интервала.

### Задание 2: Анализ временных диаграмм JK-триггера

#### 1. Временные диаграммы:

Загрузите или создайте временные диаграммы с различными комбинациями входных сигналов (J, K, Clock).

#### 2. Анализ:

Определите, в каком состоянии находится триггер в каждом временном интервале.

Объясните, как изменение входных сигналов влияет на состояние триггера.

Определите, какие состояния триггера являются стабильными, а какие – неустойчивыми.

Выясните, какое влияние оказывает сигнал Clock на работу триггера.

Проанализируйте поведение триггера при изменении состояния сигнала Reset (если имеется).

### Задание 3: Реализация JK-триггера с помощью микросхемы

#### 1. Выбор микросхемы:

Выберите микросхему, реализующую JK-триггер (например, 74LS76).

#### 2. Схема подключения:

Составьте схему подключения микросхемы, используя соответствующие выводы для сигналов J, K, Clock, Reset (если требуется) и выходного сигнала Q.

#### 3. Моделирование (опционально):

Если доступно, промоделируйте работу схемы с помощью выбранной микросхемы.

#### 4. Практическая реализация (опционально):

Постройте схему с использованием выбранной микросхемы и проверьте ее работу экспериментально.

**Время работы: 2 часа**

## Практическая работа 23

### Моделирование и анализ суммирующего счетчика

#### Цель:

Цель работы - изучить принципы работы суммирующего счетчика, разработать модель его функционирования в среде моделирования (например, в программе Multisim), провести анализ полученных результатов и сопоставить их с теоретическими данными.

#### Теоретическая часть:

##### 1. Суммирующий счетчик: определение и принцип работы

Суммирующий счетчик - это устройство, которое подсчитывает количество входных импульсов и выдает на выходе двоичный код, соответствующий суммарному количеству импульсов.

##### Принцип работы:

Счетчик состоит из набора триггеров, соединенных в цепочку.

Каждый триггер представляет собой элемент, способный находиться в одном из двух устойчивых состояний (0 или 1).

При поступлении входного импульса на первый триггер, его состояние переключается на противоположное.

Переключение состояния первого триггера может инициировать переключение следующего триггера, если первый триггер находится в состоянии 1.

Таким образом, по цепочке триггеров распространяется сигнал, а количество переключений каждого триггера соответствует количеству входных импульсов.

##### 2. Типы суммирующих счетчиков:

Существуют различные типы суммирующих счетчиков, отличающиеся типом используемых триггеров:

**Синхронные счетчики:** все триггеры переключаются одновременно по фронту тактового сигнала.

**Асинхронные счетчики:** триггеры переключаются последовательно, каждый триггер управляется выходным сигналом предыдущего.

##### 3. Декодирование выходных данных:

Для удобства отображения подсчитанных импульсов выходной код счетчика часто декодируется с помощью дешифратора. Дешифратор преобразует двоичный код счетчика в десятичный или другой удобный для восприятия формат.

#### 4. Моделирование суммирующего счетчика:

Моделирование счетчика в среде моделирования позволяет:

Проверить работоспособность схемы.

Оптимизировать её параметры.

Визуализировать процесс подсчета.

#### 5. Анализ полученных результатов:

Анализ моделирования позволяет:

Определить скорость счета.

Определить максимальное значение счетчика.

Определить максимальную частоту входных импульсов, при которой счетчик работает корректно.

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

### **Рабочие задания**

Задание 1: Моделирование суммирующего счетчика с использованием логических элементов.

Описание:

Создайте модель суммирующего счетчика на основе логических элементов (И, ИЛИ, НЕ, XOR) в любой среде моделирования, например, Logisim, Multisim или Digital.

Используйте заданное количество разрядов (например, 4 разряда).

Обеспечьте возможность сброса счетчика в начальное состояние.

Моделируйте работу счетчика при различных входных сигналах (например, при подаче единичных импульсов на тактовый вход, при изменении направления счета).

Задание 2: Анализ работы суммирующего счетчика.

Описание:\* Получите временную диаграмму работы счетчика при различных входных сигналах.

Проведите анализ работы счетчика:

Определите период счета.

Выявите особенности работы счетчика при различных входных сигналах.

Проанализируйте влияние изменения направления счета на работу счетчика.

Определите максимальное значение, которое может принимать счетчик.

Задание 3: Моделирование суммирующего счетчика с использованием языка описания аппаратуры (HDL).

Описание:

Используя язык описания аппаратуры (например, Verilog или VHDL), разработайте описание суммирующего счетчика с заданным количеством разрядов.

Обеспечьте возможность сброса счетчика в начальное состояние.

Загрузите описание счетчика в среду моделирования (например, ModelSim, Vivado) и проведите симуляцию работы.

Проанализируйте результаты симуляции, сравните их с результатами моделирования на основе логических элементов.

**Задание 4: Разработка контроллера для суммирующего счетчика.**

Описание:

Разработайте контроллер для суммирующего счетчика, который будет управлять его работой.

Контроллер должен:

Устанавливать начальное значение счетчика.

Изменять направление счета.

Останавливать и запускать счет.

Выводить информацию о текущем значении счетчика.

Моделируйте работу счетчика с контроллером, используя среду моделирования.

**Время работы: 2 часа**

## **Практическая работа 24**

### **Моделирование и анализ вычитающего счетчика**

**Цели:**

1. Изучение принципа работы вычитающего счетчика.
2. Разработка модели вычитающего счетчика с использованием языка Verilog.
3. Анализ и симуляция разработанной модели в среде моделирования.
4. Определение характеристик счетчика, таких как скорость счета, максимальное значение, возможность переполнения.
5. Сравнение вычитающего счетчика с обычным счетчиком.

## Теоретическая часть:

### 1. Введение:

Вычитающий счетчик - это электронное устройство, которое уменьшает свое значение на единицу при каждом импульсе входного сигнала. В отличие от обычного счетчика, который увеличивает свое значение, вычитающий счетчик работает в обратном направлении.

### 2. Принцип работы:

Вычитающий счетчик обычно реализуется с использованием комбинации триггеров JK и логических вентилях.

Триггер JK: В качестве основы для вычитающего счетчика обычно используются JK-триггеры. JK-триггер переключает свое состояние на противоположное при поступлении импульса на вход J или K в зависимости от состояния другого входа.

Логические вентили: Для управления переключением состояний триггеров JK используются логические вентили.

### 3. Архитектура вычитающего счетчика:

Архитектура вычитающего счетчика может быть различной в зависимости от конкретной реализации.

Счетчик с декрементом на единицу: Простейший вариант, использующий JK-триггер с  $J=K=1$ , который переключает свое состояние на противоположное при каждом импульсе.

Счетчик с декрементом на n единиц: Более сложные варианты могут использовать комбинации JK-триггеров и логических вентилях для декремента на более чем одну единицу.

### 4. Характеристики вычитающего счетчика:

Скорость счета: Скорость, с которой счетчик уменьшает свое значение. Определяется тактовой частотой.

Максимальное значение: Максимальное значение, которое может быть достигнуто счетчиком.

Переполнение: Счетчик может переполняться, если он достигает своего минимального значения и продолжает уменьшаться.

### 5. Применение вычитающих счетчиков:

Управление таймерами и контроллерами.

Реализация счетчиков обратного отсчета.

Процессоры и цифровые системы.

6. Сравнение с обычным счетчиком:

Вычитающий счетчик работает в обратном направлении по сравнению с обычным счетчиком.

Оба типа счетчиков могут быть реализованы с использованием JK-триггеров и логических вентилях.

Выбор между вычитающим и обычным счетчиком зависит от конкретного применения.

7. Моделирование и анализ:

Для моделирования и анализа вычитающего счетчика можно использовать язык Verilog, который позволяет описывать цифровые схемы. С помощью симулятора Verilog можно протестировать работу счетчика, определить его характеристики и найти возможные ошибки.

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

– титульный лист;

- введение (цель и задачи);
- выполнение
- заключение

### **Рабочие задания**

Рассчитать время задержки распространения сигнала через логические элементы, составляющие счетчик.

Определить критическую цепочку в схеме счетчика и ее время задержки.

Определить максимальную частоту работы счетчика, исходя из времени задержки критической цепочки.

Провести симуляцию работы счетчика с различными частотами тактового сигнала.

Проанализировать результаты симуляции и сравнить их с рассчитанными значениями максимальной частоты.

**Время работы:** 2 часа.

## **Практическая работа 25**

### **Моделирование и анализ параллельных регистров**

#### **Цели:**

Изучить принципы работы параллельных регистров.

Освоить методы моделирования параллельных регистров в различных программных средах.

Провести анализ характеристик параллельных регистров, таких как скорость, емкость и потребляемая мощность.

Развить навыки оптимизации работы параллельных регистров для повышения их производительности.

Приобрести знания о применении параллельных регистров в различных электронных устройствах.

#### **Теоретическая часть:**

1. Основные понятия:

Регистр: Электронное устройство, которое хранит и обрабатывает битовую информацию.

Параллельный регистр: Тип регистра, который позволяет одновременно записывать и считывать все биты данных.

Тактовая частота Скорость работы регистра, измеряемая в Гц.

Емкость регистра: Количество битов, которое может хранить регистр.

Входы: Сигналы, управляющие записью данных в регистр.

Выходы: Сигналы, выходящие из регистра, содержащие хранимую информацию.

Логические элементы: Элементы, используемые для построения регистра, например, триггеры, логические вентили.

## 2. Типы параллельных регистров:

SR-триггер: Простейший тип регистра, основанный на использовании двух логических элементов (S и R).

D-триггер: Тип регистра, который принимает информацию на своем входном D-контакте и сохраняет ее при активизации тактового сигнала.

JK-триггер: Тип регистра, который может переключать свой выходной сигнал при подаче импульса на вход J или K в зависимости от состояния выходного сигнала.

T-триггер: Тип регистра, который переключает свой выходной сигнал при подаче импульса на вход T.

Регистры сдвига: Тип регистра, который позволяет сдвигать данные на один или несколько разрядов.

## 3. Методы моделирования:

Логическое моделирование: Использует логические операции для описания поведения регистра.

Цифровое моделирование: Использует специализированные программы для моделирования работы регистра на уровне цифровых схем.

Физическое моделирование: Использует программное обеспечение для моделирования поведения регистра на уровне физических элементов.

## 4. Анализ характеристик:

Скорость: Время, которое требуется регистру для выполнения операции записи или считывания данных.

Емкость: Количество битов, которое может хранить регистр.

Потребляемая мощность: Количество энергии, потребляемой регистром во время работы.

#### 5. Оптимизация работы:

Выбор оптимального типа регистра: Подбор типа регистра, соответствующего требованиям конкретной задачи.

Минимизация задержек: Сокращение времени, необходимого для выполнения операции записи или считывания.

Снижение потребления энергии: Применение схемных решений для уменьшения потребления энергии.

#### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

#### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

#### **Рабочие задания**

## 1. Моделирование работы параллельного регистра

Задание:

Используя инструмент моделирования (например, Logisim, Multisim, LTSpice), разработать схему параллельного регистра на 4 бита.

Реализовать следующие операции:

Запись данных в регистр

Считывание данных из регистра

Сброс регистра (обнуление всех битов)

Инверсия всех битов регистра

Провести моделирование работы регистра, используя различные комбинации входных данных и управляющих сигналов.

Проанализировать результаты моделирования, убедившись в правильной работе регистра.

## 2. Анализ временных характеристик параллельного регистра

Задание:

\*Используя инструмент моделирования, измерить время задержки записи и считывания данных в регистр.

Проанализировать влияние времени задержки на общую скорость работы схемы.

Построить графики временных диаграмм, иллюстрирующие работу регистра.

Сформулировать выводы о влиянии времени задержки на быстродействие схемы.

## 3. Реализация параллельного регистра с использованием триггеров D

Задание:

Разработать схему параллельного регистра на 8 бит с использованием триггеров D.

Реализовать операции записи, считывания, сброса и инверсии данных.

Провести моделирование работы регистра.

Проанализировать схему и сравнить ее с реализацией на основе других типов триггеров (например, JK, T).

**Время работы:** 2 часа

## **Практическая работа 26**

### **Моделирование и анализ последовательных регистров**

**Цели:**

Понимание принципов работы последовательных регистров: Изучить основные типы последовательных регистров (сдвиговые, регистры сдвига с обратной связью, счетчики), их архитектуру, принципы работы и функции.

Разработка моделей последовательных регистров: Изучить методы и инструменты моделирования, позволяющие описать работу последовательных регистров, включая моделирование логических схем, функциональное моделирование, временные диаграммы и т.д.

Анализ работы последовательных регистров: Развить навыки анализа, позволяющие прогнозировать поведение последовательных регистров в различных режимах работы, определять их характеристики (скорость работы, задержки, точность) и выявлять возможные ошибки.

Применение знаний для решения практических задач: Разработать и реализовать простые схемы на основе последовательных регистров, например, счетчики, генераторы импульсов, и использовать их в простых цифровых устройствах.

#### **Теоретическая часть:**

1. Введение:

Определение последовательных регистров: Последовательные регистры — это цифровые схемы, которые хранят и обрабатывают биты данных последовательно, по одному за раз.

Типы последовательных регистров:

Сдвиговые регистры: Регистры, которые перемещают данные по битам, как по цепочке.

Регистры сдвига с обратной связью: Сдвиговые регистры, у которых выходной бит подается обратно на вход.

Счетчики: Регистры, которые последовательно изменяют свое состояние, реализуя счетные функции.

## Принципы работы:

**Тактирование:** Последовательные регистры работают под управлением тактового сигнала.

**Входные и выходные сигналы:** Входной сигнал содержит данные, которые записываются в регистр. Выходной сигнал выдает хранимые данные.

**Запоминающие элементы:** В качестве элементов памяти в последовательных регистрах используются триггеры (D-триггеры, JK-триггеры).

## Применение:

**Цифровые системы:** Последовательные регистры широко используются в цифровых системах, например, в счетчиках, таймерах, генераторах импульсов, системах связи.

**Микропроцессоры:** Последовательные регистры используются для хранения и передачи данных между различными компонентами микропроцессора.

## 2. Моделирование последовательных регистров:

**Логическое моделирование:** Использование схемных диаграмм для описания логической структуры последовательных регистров.

**Функциональное моделирование:** Описание поведения последовательных регистров в виде таблиц истинности или логических уравнений.

**Временное моделирование:** Использование временных диаграмм для визуализации работы последовательных регистров во времени, включая тактовый сигнал и изменения значений данных.

**Инструменты моделирования:** Использование специализированных программных средств для моделирования, таких как логические симуляторы, языки описания аппаратуры (Verilog, VHDL).

## 3. Анализ последовательных регистров:

**Анализ временных диаграмм:** Построение и анализ временных диаграмм для определения поведения регистров в различных режимах работы.

**Определение характеристик:** Определение характеристик регистров, таких как скорость работы, задержки, максимальное количество хранимых данных.

**Поиск ошибок:** Анализ работы регистров для выявления потенциальных ошибок и неполадок.

**Оптимизация:** Изучение методов оптимизации работы регистров, например, минимизация задержек, повышение быстродействия.

#### 4. Практические приложения:

Разработка простых цифровых схем: Применение знаний о последовательных регистрах для создания счетчиков, таймеров, генераторов импульсов и других простых устройств.

Использование в более сложных системах: Встраивание последовательных регистров в более сложные системы, например, системы связи, микропроцессорные устройства.

#### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

#### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

#### **Рабочие задания**

##### Задание 1:

Описание: Разработать модель 4-битного параллельного регистра с возможностью сброса (Reset) и загрузки (Load).

Требования:

Регистр должен иметь входные сигналы для данных (D0-D3), тактовый сигнал (CLK) и управляющие сигналы Reset и Load.

Выходные сигналы должны представлять текущее состояние регистра (Q0-Q3).

Моделировать работу регистра с помощью логической схемы (например, в Verilog/VHDL) или графа состояний.

Провести симуляцию и проанализировать поведение регистра при различных комбинациях входных сигналов.

Задание 2:

Описание: Создать модель 8-битного последовательного регистра сдвигового типа (Shift Register).

Требования:

Регистр должен иметь входные сигналы для данных (D), тактовый сигнал (CLK) и сигнал управления режимом сдвига (Shift).

Режим сдвига может быть "вправо" (Right) или "влево" (Left).

Выходные сигналы должны представлять текущее состояние регистра (Q0-Q7).

Моделировать работу регистра с помощью логической схемы или графа состояний.

Провести симуляцию и проанализировать поведение регистра при различных режимах сдвига и входных данных.

**Время работы: 2 часа**

## **Практическая работа 27**

### **Моделирование и анализ ЦАП**

#### **Цели**

Понимание работы ЦАП: Моделирование позволяет визуализировать и понять принципы работы ЦАП, а также его основные характеристики и ограничения.

Оптимизация производительности: Моделирование позволяет оптимизировать характеристики ЦАП, такие как скорость преобразования, точность, разрешение, уровень шума и т.д., для достижения желаемого результата.

## **Теория**

Основные этапы моделирования и анализа ЦАП:

1. Выбор модели: Существуют различные модели ЦАП, такие как идеальный ЦАП, ЦАП с конечным разрешением, ЦАП с шумом и т.д. Выбор модели зависит от конкретной задачи и требуемого уровня точности.

2. Создание модели: Моделирование ЦАП может быть выполнено с помощью различных инструментов, таких как программное обеспечение для моделирования систем (например, MATLAB, Simulink) или языки программирования (например, Python, C++).

3. Анализ модели: Анализ модели ЦАП включает в себя измерение и оценку различных параметров, таких как:

Точность преобразования: насколько близки выходные значения ЦАП к входным значениям.

Разрешение: минимальный шаг, который ЦАП может измерить.

Скорость преобразования: максимальная частота, с которой ЦАП может выполнять преобразования.

Уровень шума: уровень случайных колебаний выходного сигнала ЦАП.

Нелинейные искажения: искажения сигнала, вызванные нелинейностью работы ЦАП.

4. Оптимизация модели: Анализ модели может выявить области для улучшения, например, за счет изменения архитектуры ЦАП, параметров фильтрации, методов коррекции ошибок и т.д.

5. Вариация модели: Важно изучить влияние различных факторов на характеристики ЦАП, например, изменения температуры, напряжения питания, частоты дискретизации и т.д.

Основные методы анализа ЦАП:

Частотный анализ: позволяет определить частотные характеристики ЦАП, такие как полоса пропускания, частота среза и т.д.

Анализ временных характеристик: позволяет определить временные характеристики ЦАП, такие как время установления, время нарастания и т.д.

Анализ спектра: позволяет определить спектр выходного сигнала ЦАП и выявить наличие гармоник и других нежелательных составляющих.

Анализ искажений: позволяет определить уровень нелинейных искажений, вызванных работой ЦАП.

Анализ шума: позволяет определить уровень шума, присутствующего в выходном сигнале ЦАП

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

### **Рабочие задания**

#### **1. Описание принципа работы ЦАП:\***

Описать, как ЦАП преобразует цифровой сигнал в аналоговый.

Назвать основные типы ЦАП и их отличия.

Объяснить, как работает ЦАП на примере конкретного типа (например, ЦАП с цифровым квантованием).

## 2. Характеристики ЦАП:

Назвать основные характеристики ЦАП (разрешение, скорость преобразования, нелинейность, динамический диапазон).

Объяснить, как эти характеристики влияют на качество аналогового сигнала.

Дать примеры применения ЦАП с различными характеристиками.

## 3. Моделирование работы ЦАП:

Используя программное обеспечение для моделирования (например, MATLAB, Simulink), создать модель ЦАП.

Сгенерировать цифровой сигнал и подать его на вход модели ЦАП.

Проанализировать полученный аналоговый сигнал, сравнить его с исходным цифровым сигналом.

## 4. Анализ искажений сигнала:

Определить основные типы искажений, возникающих при работе ЦАП.

Проанализировать влияние характеристик ЦАП на уровень искажений.

Предложить способы снижения искажений.

**Время работы:** 2 часа

## **Практическая работа 28**

### **Моделирование и анализ АЦП**

**Цель работы:** Изучить принципы работы аналого-цифровых преобразователей (АЦП). Ознакомиться с основными типами АЦП и их характеристиками. Научиться моделировать работу АЦП в среде программирования. Провести анализ характеристик моделируемого АЦП.

### **Теоретическая часть:**

Аналого-цифровые преобразователи (АЦП):

АЦП преобразуют аналоговые сигналы (непрерывные по времени и амплитуде) в цифровые сигналы (дискретные по времени и амплитуде).

АЦП выполняют квантование и кодирование входного сигнала.

Разбиение диапазона входных значений на дискретные уровни.

Представление каждого уровня квантования двоичным кодом.

Основные типы АЦП:

Последовательные: Используют последовательное сравнение сигнала с эталонным напряжением.

Пример АЦП с наклонным преобразованием (Ramp ADC).

Параллельные: Используют параллельное сравнение сигнала с несколькими эталонными напряжениями.

Пример: АЦП с последовательным приближением (Successive Approximation ADC).

Сигма-дельта: Используют интегрирование сигнала и последующее преобразование в цифровой сигнал.

Пример: Сигма-дельта АЦП (Sigma-Delta ADC).

Характеристики АЦП:

Разрешение: Число бит, используемых для кодирования сигнала.

Скорость преобразования: Число преобразований в секунду.

Точность: Максимальная ошибка преобразования.

Динамический диапазон: Разница между максимальным и минимальным значениями, которые может преобразовать АЦП.

Частота дискретизации: Частота, с которой АЦП дискретизирует входной сигнал.

Моделирование АЦП:

Использование программных сред: Существуют специализированные программные средства для моделирования АЦП, например, MATLAB, Simulink, Multisim.

Выбор модели: Необходимо выбрать подходящий тип АЦП для моделирования в зависимости от задачи.

Настройка параметров: Нужно задать необходимые параметры модели, такие как разрешение, скорость преобразования, тип сигнала.

Анализ результатов: Моделирование позволяет исследовать характеристики АЦП, например, точность преобразования, искажения, скорость отклика.

### **Общие положения**

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

### **Форма отчета:**

- титульный лист;
- введение (цель и задачи);
- выполнение

– заключение

**Рабочее задание:**

1. Выбор типа АЦП для моделирования:

Выберите тип АЦП (например, АЦП с последовательным приближением) и определите его характеристики (разрешение, скорость преобразования).

2. Моделирование АЦП в выбранной среде:

Создайте модель выбранного АЦП в выбранной среде программирования (MATLAB, Simulink, Multisim).

Задайте входной сигнал для моделирования (например, синусоидальный сигнал).

3. Анализ характеристик моделируемого АЦП:

Исследуйте характеристики моделируемого АЦП, такие как:

Точность преобразования.

Скорость преобразования.

Искажения сигнала.

Влияние шума на результат преобразования.

4. Сравнение результатов моделирования с теоретическими данными:

Сравните полученные результаты моделирования с теоретическими значениями для выбранного типа АЦП.

Объясните любые различия между результатами моделирования и теоретическими значениями.

5. Представление результатов:

Подготовьте отчет о проделанной работе, включающий:

Описание выбранного типа АЦП.

Описание моделирования и настройки параметров.

Анализ полученных результатов.

Сравнение результатов моделирования с теоретическими данными.

Выводы о работе АЦП.

## Список литературы:

1. Степина, В. В. Архитектура ЭВМ и вычислительные системы: учебник / В.В. Степина. — Москва: КУРС : ИНФРА-М, 2021. — 384 с. — (Среднее профессиональное образование). - ISBN 978-5-906923-07-3.
2. Миленина, С. А. Электроника и схемотехника : учебник и практикум для среднего профессионального образования / С. А. Миленина ; под редакцией Н. К. Миленина. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2021. — 270 с. — (Профессиональное образование). — ISBN 978-5-534-06085-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/472059>