

|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ "МИРЭА - РОССИЙСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ" |

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

по дисциплине ―Интерфейсы измерительных приборов и систем безопасности‖

Направление подготовки 12.03.01 «Приборостроение»

*(код и наименование)*

Профиль "**Аналитическое приборостроение и интеллектуальные системы безопасности**"

Институт КБСП «Комплексной безопасности и специального приборостроения»

*(краткое и полное наименование)*

Форма обучения очная

*(очная, очно-заочная, заочная)*

Программа подготовки академический бакалавриат

*(академический, прикладной бакалавриат)*

Квалификация выпускника Бакалавр

Кафедра КБ6 «Приборы и информационно-измерительные системы»

*(краткое и полное наименование кафедры, разработавшей методические указания)*

Москва 2021г

## Цель работы

Целью работы является:

1. изучить принципы синтеза логических схем на комбинационных и последовательностных элементах;
2. научиться синтезировать схемы на базе комбинационных элементов (дешифратор, мультиплексор и др.);
3. научиться синтезировать схемы на базе последовательностных элементов (регистры, счетчики, триггеры);
4. изучить особенности схем подавления дребезга, глитча и научиться проектировать устройства с такими схемами.

## Подготовка к выполнению работы

По конспекту лекций и рекомендуемым источникам изучить базовые логические элементы, их условное графическое отображение, таблицу истинности, булево выражение и принцип работы. Рассмотреть способы построения схем комбинационных устройств по таблице истинности их работы. Изучить принципы построения схем на базе последовательностных элементов и временных диаграмм для описания их работы.

## Краткие теоретические сведения

Любая логическая схема (без элементов памяти) полностью описывается таблицей истинности. Эта таблица является исходной информацией для синтеза схемы на основе логических элементов «И», «ИЛИ», «НЕ». Для разработки требуемого цифрового устройства сначала на основе таблицы истинности записывают его логическое выражение. Затем с целью упрощения цифрового устройства минимизируют его логическое выражение и далее разрабатывают схему, реализующую полученное логическое выражение. Логические выражения можно получить двумя способами: в совершенной дизъюнктивной нормальной форме (СДНФ) и в совершенной конъюнктивной нормальной форме (СКНФ).

**Совершенная дизъюнктивная нормальная форма (СДНФ).** Функция представляется суммой групп. Каждая группа состоит из произведения, в которую входят все переменные. Например:

f(x1,x2,x3)=~~x1~~·x2·x3 + x1·~~x2~~·x3 + x1·x2·~~x3~~.

**Совершенная конъюнктивная нормальная форма (СКНФ).** Функция представляется произведением групп. Каждая группа состоит из суммы, в которую входят все переменные. Например:

f(x1,x2,x3)=(~~x1~~+x2+x3) · (x1+~~x2~~+x3) · (x1+x2+~~x3~~).

Если схема имеет несколько выходов, то каждый выход описывается своей функцией. Такая система функций называется системой собственных

функций. СДНФ составляется на основе таблицы истинности по следующему правилу: *для каждого набора переменных, при котором функция равна 1, записывается произведение, в котором с отрицанием берутся переменные, имеющие значение «0».* СКНФ составляется на основе таблицы истинности по правилу: *для каждого набора переменных, при котором функция равна 0, записывается сумма, в которой с отрицанием берутся переменные, имеющие значение 1.*

Дешифратор (декодер).

Служит для преобразования n-разрядного позиционного двоичного кода в единичный выходной сигнал на одном из 2n выходов. При каждой входной комбинации сигналов на одном из выходов появляется 1 . Таким образом, по единичному сигналу на одном из выходов можно судить о входной кодовой комбинации. Таблица истинности для декодера с двумя входами изображена в таблице 1.

Таблица 1 – Таблица истинности двухразрядного дешифратора

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **x1** | **x2** | **y0** | **y1** | **y2** | **y3** |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

Для построения схемы декодера по таблице истинности воспользуемся методикой, изложенной выше и получим схему, представленную на рисунке 1.



а) б)

Рисунок 1 – Схема и условное графическое обозначение дешифратора

Мультиплексор.

Мультиплексор – устройство, которое позволяет коммутировать один из 2n информационных входов X на один выход Y под действием n управляющих (адресных) сигналов. Входы A0 и A1 являются управляющими входами мультиплексора, определяющими адрес информационного входного сигнала, который будет соединѐн с выходным выводом мультиплексора Y. Информационные входные сигналы обозначены: X0, X1, X2 и X3. На рисунке 2 представлена схема и условное графическое обозначение мультиплексора.



а) б)

Рисунок 2 – Схема и условное графическое обозначение мультиплексора

Сумматор.

Сумматор предназначен для сложения двоичных чисел. Построение двоичных сумматоров обычно начинается с сумматора по модулю 2, таблица истинности которого представлена в таблице 2.

Таблица 2. – Таблица истинности сумматора по модулю 2

|  |  |  |
| --- | --- | --- |
| **x1** | **x2** | **y** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Сумматор по модулю 2 выполняет суммирование без учѐта переноса. В обычном двоичном сумматоре требуется учитывать перенос, поэтому требуются схемы, позволяющие формировать перенос в следующий двоичный разряд. Таблица истинности такой схемы, называемой полусумматором,

приведена в таблице 3. Здесь A и B – слагаемые, S – сумма, P0 – перенос в старший разряд (выход переноса Pout).

Таблица 3 – Таблица истинности полусумматора

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **S** | **P0** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Принципиальная схема, реализующая таблицу истинности сумматора по модулю 2 и полусумматора представлена на рисунке 3.



а) б)

Рисунок 3 – Схема а) сумматора по модулю 2 и б) полусумматора

Регистры.

Регистры предназначены для хранения и преобразования многоразрядных двоичных чисел. Для запоминания отдельных разрядов числа могут применяться триггеры различных типов. Одиночный триггер можно считать одноразрядным регистром. Занесение информации в регистр называется операцией записи. Операция выдачи информации из регистра – считывание.

По способу ввода/вывода информации различают:

* параллельные (регистры хранения) – информация вводится и выводится одновременно по всем разрядам;
* последовательные (регистры сдвига) – информация бит за битом

«проталкивается» через регистр и выводится также последовательно.

Принципиальная схема, для параллельного и последовательного регистра представлена на рисунке 4.

Параллельные регистры осуществляют прием и выдачу информации в параллельном коде, а это значит, что для передачи каждого разряда используется отдельная линия. Для записи информации в регистр на его входных выводах (**D0-D3**) нужно установить логические уровни, после чего на

вход синхронизации (**C**) подать разрешающий импульс — логическую единицу. После этого на выходах **Q0-Q3** появится записанное слово. Регистры запоминают входные сигналы только в момент времени, определяемый сигналом синхронизации.



а) б)

Рисунок 4 – Схема а) параллельного и б) последовательного регистра

Последовательный регистр (регистр сдвига) обычно служит для преобразования последовательного кода в параллельный и наоборот. Рассмотрим работу этого регистра (рисунок 5). Можно предположить, что в начале все триггеры регистра находятся в состоянии логического нуля, т.е. Q0=0, Q1=0, Q2=0, Q3=0. Если на входе D-триггера Т1 имеет место логический 0, то поступление синхроимпульсов на входы «С» триггеров не меняет их состояния.

Как следует из рисунка 4, синхроимпульсы поступают на соответствующие входы всех триггеров регистра одновременно и записывают в них то, что имеет место на их информационных входах. На информационных входах триггеров Т2, Т3, Т4 – уровни логического «0», т.к. информационные входы последующих триггеров соединены с выходами предыдущих триггеров, находящихся в состоянии логического «0», а на вход «D» первого триггера, по

условию примера, подается «0» из внешнего источника информации. При подаче на вход «D» первого триггера «1», с приходом первого синхроимпульса, в этот триггер запишется «1», а в остальные триггеры – «0», т.к. к моменту поступления фронта синхроимпульса на выходе триггера Т1 ещѐ присутствовал логический «0». Таким образом, в триггер Т1 записывается та информация (тот бит), которая была на его входе «D в момент поступления фронта синхроимпульса и т.д. При поступлении второго синхроимпульса логическая

«1» с выхода первого триггера, запишется во второй триггер, и в результате происходит сдвиг первоначально записанной «1» с триггера Т1 в триггер Т2, из триггера Т2 в триггер Т3 и т.д. Таким образом, производится последовательный сдвиг поступающей на вход регистра информации (в последовательном коде) на один разряд вправо в каждом такте синхроимпульсов. После поступления четырѐх синхроимпульсов регистр оказывается полностью заполненным разрядами числа, вводимого через последовательный ввод «D». В течение следующих четырѐх синхроимпульсов производится последовательный поразрядный вывод из регистра записанного числа, после чего регистр оказывается полностью очищенным (регистр окажется полностью очищенным только при условии подачи на его вход уровня «0» в режиме вывода записанного числа).

Рисунок 5 – Временные диаграммы, поясняющие работу регистра сдвига Счѐтчик.

Счѐтчик – устройство для подсчѐта числа входных импульсов. Различают синхронные и асинхронные, суммирующие и вычитающие счетчики. Основным параметром счетчика является модуль счета (т.е. количество устойчивых состояний).

Асинхронный суммирующий счѐтчик представляет собой несколько последовательно включенных счѐтных триггеров (рисунок 6). По каждому входному импульсу счѐтный триггер изменяет своѐ состояние на противоположное.



Рисунок 6 – Схема асинхронного суммирующего счѐтчика на базе Т-триггеров, срабатывающих по переходу от 1 к 0 (задний фронт)

Для того чтобы разобраться, как работает схема двоичного счѐтчика, воспользуемся временными диаграммами сигналов на входе и выходах этой схемы, приведѐнными на рисунке 7.



Рисунок 7 – Временные диаграммы, поясняющие работу асинхронного суммирующего счѐтчика

Пусть первоначальное состояние всех триггеров счѐтчика будет нулевым. Это состояние видно на временных диаграммах. Запишем его в таблицу 4. После поступления на вход счѐтчика тактового импульса (который воспринимается по заднему фронту) первый триггер изменяет своѐ состояние на противоположное, то есть единицу. Запишем новое состояние выходов счѐтчика в ту же самую таблицу. Так как по приходу первого импульса изменилось состояние первого триггера, то этот триггер содержит младший разряд двоичного числа (единицы). Подадим на вход счѐтчика ещѐ один тактовый импульс. Значение первого триггера снова изменится на прямо противоположное. На этот раз на выходе первого триггера, а значит и на входе второго триггера сформируется задний фронт. Это означает, что второй триггер

тоже изменит своѐ состояние на противоположное. Это отчѐтливо видно на временных диаграммах, приведѐнных на рисунке 7. Запишем новое состояние выходов счѐтчика в таблицу 4. В этой строке таблицы образовалось двоичное число 2. Оно совпадает с номером входного импульса. Продолжая анализировать временную диаграмму, можно определить, что на выходах приведѐнной схемы счѐтчика последовательно появляются цифры от 0 до 7. Эти цифры записаны в двоичном виде. При поступлении на счѐтный вход счѐтчика очередного импульса, содержимое его триггеров увеличивается на 1. Поэтому такие счѐтчики получили название суммирующих двоичных счѐтчиков. Если информацию снимать с инверсных выходов триггеров, то получится вычитающий счѐтчик.

Таблица 4 – Изменение уровней на выходе суммирующего двоичного счѐтчика при поступлении на его вход импульсов

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер****входного импульса** | **Q2** | **Q1** | **Q0** |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 |
| 8 | 0 | 0 | 0 |

Счѐтчик с произвольным модулем счѐта.

Для построения такого счѐтчика можно использовать двоичный счѐтчик, у которого модуль счѐта М должен быть больше модуля счѐта разрабатываемого счѐтчика с произвольным модулем счѐта. Пусть нужно сделать счѐтчик с М= 10. У 4-х разрядного счѐтчика модуль счѐта равен 16 (больше 10). Схема счѐтчика представляет собой 4 последовательно включѐнных счѐтных триггера, у которых есть вход сброса R. Число 10 в двоичной системе счисления представляется 1010. Когда на выходах счетчика будет код 1010, на выходе элемента «И» появится логическая единица, которая запустит схему гашения. Длительность импульса на выходе схемы гашения должна быть достаточна для надѐжного сброса всех триггеров счѐтчика в 0. Разряды числа 1010, равные 1 подаются на схему «И» с прямых выходов триггеров, а равные 0 - с инверсных. Таким образом, как только счѐтчик досчитает до 10, произойдѐт обнуление всех триггеров и счѐт продолжится с кода 0000.



Рисунок 6 – Схема счетчика с модулем 10

## Порядок выполнения лабораторной работы

1. Составить таблицу истинности для дешифратора, имеющего заданное количество выходов.
2. Начертить схему дешифратора, имеющего заданное количество выходов.
3. Включить стенд и собрать схему цифрового устройства дешифратора.
4. Подключить к выходу схемы одноразрядный светодиодный индикатор для визуального отображения выходного сигнала схемы.
5. Подать на вход схемы все возможные комбинации сигналов и заполнить таблицу истинности для разработанного цифрового устройства.
6. Начертить схему для регистра, имеющего заданный тип и разрядность.
7. Включить стенд и собрать схему цифрового устройства регистра.
8. Составить таблицу состояний для регистра, имеющего заданный тип и разрядность.
9. Начертить схему для счетчика, имеющего заданный тип и модуль счета.
10. Включить стенд и собрать схему цифрового устройства счетчика.
11. Составить таблицу состояний для регистра, имеющего заданный тип и модуль счета.
12. Продемонстрируйте результаты работы преподавателю. Оформите отчет, ответьте на контрольные вопросы и сделайте выводы по проделанной работе.

## Варианты заданий

Номер варианта соответствует последней цифре в студенческом билете студента. В таблице 5 указаны: количество выходов для проектируемого дешифратора, тип и разрядность схемы проектируемого регистра, тип и счетчика и его модуль счета.

Таблица 5 – Варианты заданий на лабораторную работу

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **кол-во выходов дешифратора** | **тип и разрядность регистра** | **тип и модуль счета счетчика** |
| 0 | 3 | параллельный,4 разряда | суммирующий,по модулю 7 |
| 1 | 4 | параллельный,3 разряда | вычитающий,по модулю 7 |
| 2 | 5 | параллельный,2 разряда | суммирующий,по модулю 5 |
| 3 | 6 | последовательный,2 разряда | вычитающий,по модулю 5 |
| 4 | 7 | последовательный,3 разряда | вычитающий,по модулю 3 |
| 5 | 3 | последовательный,4 разряда | вычитающий,по модулю 6 |
| 6 | 4 | параллельный,4 разряда | суммирующий,по модулю 7 |
| 7 | 5 | последовательный,3 разряда | вычитающий,по модулю 6 |
| 8 | 6 | параллельный,2 разряда | суммирующий,по модулю 3 |
| 9 | 7 | последовательный,2 разряда | суммирующий,по модулю 6 |

## Содержание отчета

Отчет оформляется в отдельной тетради, демонстрируется преподавателю по окончании лабораторной работы и должен содержать:

* цель работы;
* исходный вариант;
* схемы разработанных цифровых устройств;
* ответы на контрольные вопросы;
* выводы.

## Контрольные вопросы

1. Принцип работы шифратора?
2. Таблица истинности шифратора?
3. Принцип работы демультиплексора?
4. Таблица истинности демультиплексора?
5. Чем определяется разрядность регистров?
6. По каким принципам классифицируются регистры?
7. Объяснить принцип работы вычитающего счетчика.
8. Изобразить временные диаграммы работы вычитающего счетчика.

## Рекомендуемые источники

1. [www.iqlib.ru](http://www.iqlib.ru/) (Дмитриев Н.А., Ёхин М.Н., Иванов М.А., Ковригин Б.Н. и др.

«Схемотехника ЭВМ. Сборник задач»)

1. [www.iqlib.ru](http://www.iqlib.ru/) (Никитин В.А. «Схемотехника интегральных схем ТТЛ, ТТЛШ и КМОП»)
2. [www.iqlib.ru](http://www.iqlib.ru/) (Аверченков О.Е. «Схемотехника: аппаратура и программы»)

# МЕТОДИЧЕСКИЕ УКАЗАНИЯ

К ЛАБОРАТОРНОЙ РАБОТЕ №2 НА ТЕМУ:

# «РЕАЛИЗАЦИЯ ПРОТОКОЛА MODBUS-RTU МИКРОКОНТРОЛЛЕРОМ С ЯДРОМ AVR»

## Цель работы

Целью работы является:

1. изучить интегрированную среду разработки AVR Studio от компании Atmel и получить навыки работы с ней;
2. изучить особенности работы последовательного асинхронного интерфейса USART, встроенного в микроконтроллер с ядром AVR;
3. освоить методику расчета скорости для обмена по последовательному интерфейсу, настройки формата посылки и задания режимов работы модуля USART;
4. изучить особенности обмена по протоколу ModBus-RTU интерфейса RS-485 промышленных сетей;
5. изучить способы отладки программ с использованием стартового набора разработчика STK500 фирмы Atmel и программы терминала.

## Подготовка к выполнению работы

По конспекту лекций и рекомендуемым источникам изучить архитектуру, систему команд и особенности работы последовательного асинхронного интерфейса USART, встроенного в микроконтроллер с ядром AVR. Изучить формат посылки, расчет контрольной суммы и порядок обмена по протоколу ModBus-RTU интерфейса RS-485 промышленных сетей. Проанализировать рассмотренные в источниках примеры программ.

Используя методические указания [1] и интегрированную среду разработки AVR Studio от компании Atmel получить навыки работы с текстовым редактором, программными проектами, транслятором программ и отладчиком. Изучить порядок подключения и программирования стартового набора разработчика STK500 фирмы Atmel.

## Порядок выполнения лабораторной работы

1. Включите ЭВМ и откройте интегрированную среду разработки AVR Studio.
2. Создайте новый проект, укажите имя и поместите его в отдельную папку.
3. Напишите текст программы в соответствии с вариантом задания. Оттранслируйте проект и исправьте ошибки, если они есть. При отсутствии ошибок убедитесь, что в папке проекта появился файл с расширением \*.hex.
4. Запустите отладчика. Выполните пошаговую отладку программы и запишите значения переменных в заданных точках программы.
5. Включите стартовый набор разработчика STK500 и загрузите программу (файл с расширением \*.hex) в установленный на стенде микроконтроллер с ядром AVR, используя программатор в среде разработки AVR Studio.
6. Проверьте работоспособность программы, используя различные варианты комбинаций тестовых воздействий с ПК и отклик микроконтроллера с ядром AVR через программу терминала, настроенную в соответствии с заданием.
7. Продемонстрируйте преподавателю работоспособный вариант программы с помощью STK500 и среды разработки AVR Studio. Оформите отчет, ответьте на контрольные вопросы и сделайте выводы по проделанной работе.

## Варианты заданий

В лабораторной работе необходимо написать и отладить программу, в которой расшифровываются данные, поступившие с ПК, формируется ответная посылка, содержащая не только измерительную информацию, но и контрольную сумму.

Номер варианта соответствует последней цифре в студенческом билете студента. В таблице указаны формат и скорость обмена данными по последовательному асинхронному интерфейсу. В формате обмена данными оговаривается количество бит данных в посылке, наличие и вид контроля, количество стоп-бит. Частота тактовых импульсов для микроконтроллера составляет 3,686 МГц.

Таблица 1 – Варианты заданий на лабораторную работу

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Скорость** | **Бит данных** | **Вид контроля** | **Стоп-бит** |
| 0 | 600 бод | 8 | нет контроля | 1 |
| 1 | 1200 бод | 7 | по четности | 2 |
| 2 | 2400 бод | 6 | по нечетности | 1 |
| 3 | 4800 бод | 5 | нет контроля | 2 |
| 4 | 9600 бод | 8 | по четности | 1 |
| 5 | 14400 бод | 7 | по нечетности | 1 |
| 6 | 19200 бод | 6 | нет контроля | 2 |
| 7 | 28800 бод | 5 | по четности | 1 |
| 8 | 38400 бод | 8 | по нечетности | 2 |
| 9 | 57600 бод | 7 | нет контроля | 1 |

## Содержание отчета

Отчет оформляется в электронном виде с помощью текстового редактора, демонстрируется преподавателю по окончании лабораторной работы и должен содержать:

* цель работы;
* заданный формат посылки и расчет параметров синхронизации;
* исходный текст программы (листинг) с комментариями;
* содержимое загрузочного файла (файл с расширением \*.hex);
* скриншоты для состояний внутренних регистров микроконтроллера с ядром AVR на момент передачи каждого бита посылки;
* ответы на контрольные вопросы;
* выводы.

## Контрольные вопросы

1. Появление какого сигнала на линии RxD, или TxD для последовательного асинхронного интерфейса USART, встроенного в микроконтроллер с ядром AVR означает начало посылки?
2. Через какие регистры ввода/вывода настраивается скорость обмена данными для последовательного асинхронного интерфейса USART, встроенного в микроконтроллер с ядром AVR?
3. Через какие регистры ввода/вывода настраивается формат обмена данными для последовательного асинхронного интерфейса USART, встроенного в микроконтроллер с ядром AVR?
4. В посылке передаются биты данных 10111011. Какое значение примет бит контроля, если задан контроль по четности?
5. В посылке передаются биты данных 10001011. Какое значение примет бит контроля, если задан контроль по нечетности?
6. При каком максимальном расхождении (в %) между реальной и желаемой скоростью, обмен данными будет возможным?
7. Последовательный асинхронный интерфейс USART, встроенный в микроконтроллер с ядром AVR является полнодуплексным, или полудуплексным? Поясните ответ.
8. Какова глубина буферного регистра-приемника данных в модуле USART, встроенном в микроконтроллер с ядром AVR?

## Рекомендуемые источники

1. С.А. Канаев, О.В. Москаленко Методические указания для лабораторного практикума «Микропроцессорная измерительная техника. Порядок работы с пакетом AVR Studio»
2. [www.atmel.com/images/2512s.pdf](http://www.atmel.com/images/2512s.pdf)