Министерство науки и высшего образования Российской Федерации

Национальный исследовательский университет «МИЭТ»

**В.И. Ухандеев, М.И. Собченко, А.С. Корявина**

# Методические указания к лабораторным работам по курсу «Интерфейсы вычислительных систем»

Утверждено редакционно-издательским советом университета

Москва 2022

УДК 004.5:004.22 (076.5)

Рецензент канд. техн. наук *Д.В. Вертянов*

**Ухандеев В.И., Собченко М.И., Корявина А.С.**

Методические указания к лабораторным работам по курсу «Интерфейсы вычислительных систем». М.: МИЭТ, 2022. 28 с.: ил.

Состоят из двух лабораторных работ, посвященных изучению принципов работы интерфейсов и кодирования информации на канальном уровне. Включают в себя необходимые теоретические сведения по предмету «Интерфейсы вычислительных систем».

Для студентов 3-го курса Института МПСУ, обучающихся по специальности 09.03.01 «Информатика и вычислительная техника».

**© МИЭТ, 2022**

**Предисловие**

Проектирование любого устройства вычислительной техники неразрывно связано с реализацией взаимодействия между его составными частями и с другими устройствами посредством интерфейсов. Причем на разном уровне конструкторской иерархии это могут быть интерфейсы различного назначения и сложности. При проектировании устройства на уровне ячеек и модулей в первую очередь применяются межкомпонентные интерфейсы - интерфейсы между микросхемами в пределах печатной платы, длина линий связи которых не превышает десятков сантиметров. Для связи с внешними устройствами применяют периферийные интерфейсы, длина линий связи которых составляет единицы-десятки метров. Это интерфейсы с последовательным принципом передачи информации, в которых для решения задачи битовой и кадровой синхронизации применяют различные способы представления и кодирования логических сигналов. Общая тенденция совершенствования различных по функциональному назначению интерфейсов сегодняшнего дня - переход на последовательные принципы передачи информации с повышением пропускной способности.

Значимость интерфейсов передачи информации подтверждает тот факт, что в федеральную программу подготовки студентов по направлению 09.03.01 «Информатика и вычислительная техника» введена обязательная учебная дисциплина «Интерфейсы вычислительных систем».

Система автоматизированного проектирования (САПР) для программируемых логических интегральных схем (ПЛИС) является наглядным инструментом для работы с цифровыми сигналами и визуализацией временны́х диаграмм реализации кодирования, протоколов информационного обмена по интерфейсам.

В лабораторных работах настоящих методических указаний закрепляются знания обучающихся по кодам, применяемым в последовательных асинхронных интерфейсах, и по последовательным синхронным межкомпонентным интерфейсам на примере интерфейсов I2C и SPI. Программно-аппаратная реализация позволяет визуализировать временны́е диаграммы информационных сигналов и на практике реализовать и применить интерфейсы.

Работы выполняются с применением САПР для ПЛИС на языках высокого уровня Verilog и VHDL.

# Лабораторная работа № 1

# Изучение и реализация процедур арифметического кодирования. Функциональное тестирование

***Цель работы*:** изучение, реализация и функциональное тестирование методов битового кодирования информации.

***Продолжительность работы*:** 3 часа.

## Теоретические сведения

***Интерфейс вычислительной системы*** - совокупность программных, аппаратных и конструкторских средств, реализующих алгоритмы взаимодействия устройств. Одним из таких средств является кодирование информации при ее передаче. Кодирование представляет собой набор правил изменения информационной последовательности для передачи в канале. Использование кодирования при передаче информации дает следующие преимущества:

* повышение скорости передачи информации;
* отсутствие необходимости в дополнительном синхросигнале - имеется собственная битовая синхронизация кодера и декодера;
* уменьшение полосы занимаемых сигналом частот (спектра);
* удаление постоянной составляющей спектра частот передаваемого сигнала - гальваническая развязка устройств (обязательна в цифровой телефонии);
* возможность обнаружения ошибок передачи информации;
* простота аппаратуры для кодирования;
* сглаживание до равномерного спектра сигнала - уменьшение электромагнитных помех.

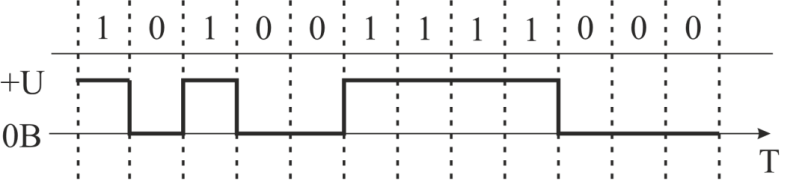
Физические методы кодирования информации основаны на изменении характеристик передаваемого сигнала - амплитуды, частоты и фазы напряжения. В настоящей лабораторной работе рассмотрим подсемейство физических кодов.

***Алфавитный код*** - это код, в котором исходной битовой последовательности (символу) ставится в соответствие другая битовая последовательность (необязательно той же длины). При ограниченном наборе передаваемых символов длину каждого символа сокращают, это повышает скорость передачи. Кроме того, алфавитный код легко декодируем. Недостаток данного кода - при большом объеме передаваемой информации она становится несжимаемой, и всем устройствам необходима таблица перевода символов, что требует больших затрат памяти.

*Униполярный NRZ-код*

***Униполярный код без возвращения к нулю*** (nonreturn to zero) образуется из двоичной последовательности по таким правилам:

* логический «0» передается уровнем 0 В в течение битового интервала (длительностью в 1 бит);
* логическая «1» передается уровнем +*U* В в течение битового интервала (рис.1).



*T*

+*U*

0 B

Рис.1. Униполярный NRZ-код

Достоинства униполярного *NRZ*-кода:

простота реализации: уровень 0 В = «1», уровень +*U* В = «0».

Недостатки униполярного *NRZ*-кода:

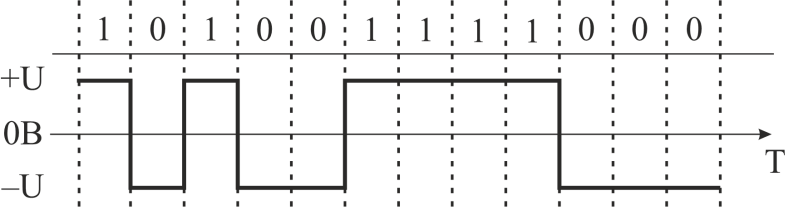
* при передаче последовательностей из одинаковых символов в канале нарастает паразитная емкость (постоянная составляющая). Это увеличивает время и ухудшает качество работы устройств и канала без помех, а также не обеспечивает гальваническую развязку устройства;
* при передаче последовательностей из одинаковых символов устройства рассинхронизируются, так как долго нет фронтов сигнала;
* нет возможности обнаружить ошибку передачи.

Сам по себе униполярный *NRZ*-код используется редко, однако он применяется в простом интерфейсе RS-232.

*Биполярный NRZ-код*

***Биполярный код без возвращения к нулю*** образуется из двоичной последовательности по таким правилам:

* логический «0» передается уровнем –*U* В в течение битового интервала;
* логическая «1» передается уровнем +*U* В в течение битового интервала (рис.2).



–*U*

*T*

0 B

+*U*

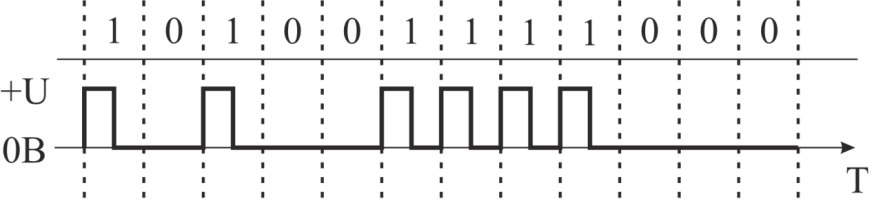
Рис.2. Биполярный NRZ-код

Данный метод кодирования выгодно отличается от предыдущего уменьшением в 2 раза мощности, рассеиваемой при передаче. Остальные достоинства и недостатки биполярного и униполярного *NRZ*-кодов совпадают.

*Униполярный RZ-код*

***Униполярный код с возвращением к нулю*** (return-to-zero) образуется из двоичной последовательности по таким правилам:

* логический «0» передается уровнем 0 В в течение битового интервала;
* логическая «1» передается уровнем +*U* В в течение первой половины битового интервала и уровнем 0 В в течение второй половины битового интервала (рис.3).



*T*

0 B

+*U*

Рис.3. Униполярный RZ-код

Достоинства униполярного *RZ*-кода:

* фронт в середине битового интервала обеспечивает частичную синхронизацию;
* простота реализации.

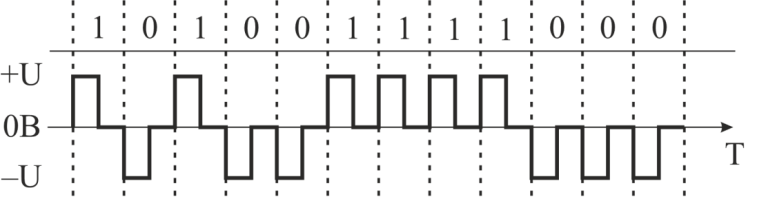
Недостатки униполярного *RZ*-кода:

* присутствует постоянная составляющая;
* при передаче последовательностей из логических нулей устройства рассинхронизируются;
* нет обнаружения ошибок.

*Биполярный RZ-код*

***Биполярный код с возвращением к нулю*** образуется из двоичной последовательности по таким правилам:

* логический «0» передается уровнем –*U* В в течение первой половины битового интервала и уровнем 0 В в течение второй половины битового интервала;
* логическая «1» передается уровнем +*U* В в течение первой половины битового интервала и уровнем 0 В в течение второй половины битового интервала (рис.4).



*T*

–*U*

0 B

+*U*

Рис.4. Биполярный RZ-код

Достоинства биполярного *RZ*-кода:

* полная синхронизация (фронт для синхронизации присутствует в каждом битовом интервале);
* уменьшение количества низкочастотных компонентов в аппаратуре.

Недостатки биполярного *RZ*-кода:

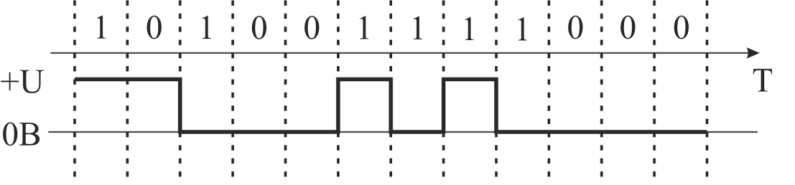
* присутствует постоянная составляющая;
* нет обнаружения ошибок.

## ***NRZI****-код*

Для ***кода без возвращения к нулю и с инверсией*** (Non Return to Zero with ones Inverted, NRZI) различают два противоположных варианта правил формирования сигнала.

Вариант № 1:

* логическая «1» передается сменой полярности напряжения в канале связи;
* при поступлении логического «0» уровень напряжения в линии связи постоянный (рис.5).



*T*

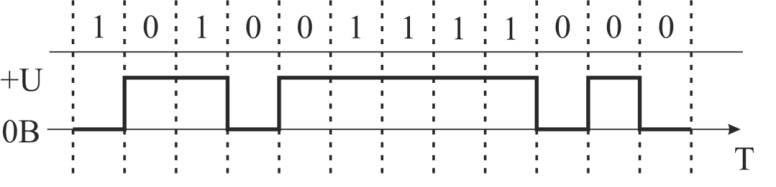
+*U*

0 B

Рис.5. Вариант № 1 правил формирования сигнала NRZI-кода

Вариант № 2:

* логический «0» передается сменой полярности напряжения в канале связи;
* при поступлении логической «1» уровень напряжения в линии связи не меняется (рис.6).



*T*

+*U*

0 B

Рис.6. Вариант № 2 правил формирования сигнала NRZI-кода

Достоинства и недостатки *NRZI*-кода такие же, как достоинства и недостатки униполярного *NRZ*-кода. Модифицированный вариант *NRZI*-кода используется в интерфейсе FDDI.

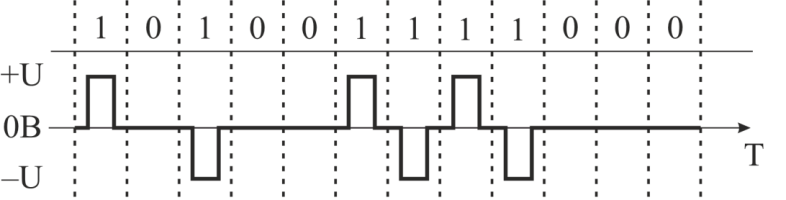
Для некоторых сред передачи, например оптических кабелей, введение трех уровней напряжения (+*U*, 0, –*U*) нежелательно, так как они предназначены для двухуровневой передачи (различают состояния: носитель заряда, отсутствие носителя). В этом случае предпочтительнее использование двухуровневых кодов.

## *Код* ***AMI***

***AMI*** (alternative mark inversion)-код, предложенный Э. Баркером для цифровой передачи речевых сигналов (1962 г.). Код формируется по таким правилам:

* логический «0» передается нулевым уровнем напряжения в течение битового интервала;
* логическая «1» передается импульсом напряжения либо +*U* В, либо –*U* В (текущая полярность напряжения противоположна предыдущей при логической «1»).

Импульс напряжения длительностью *Т*вх. бит/2 расположен в середине битового интервала (рис.7).



*T*

0 B

–*U*

+*U*

Рис.7. Код AMI

Достоинства кода *AMI*:

* частичная синхронизация;
* нет постоянной составляющей - возможна гальваническая развязка;
* обнаружение ошибок при нарушении правил следования единичных импульсов;
* меньшая полоса частот, чем у *NRZ*-кода.

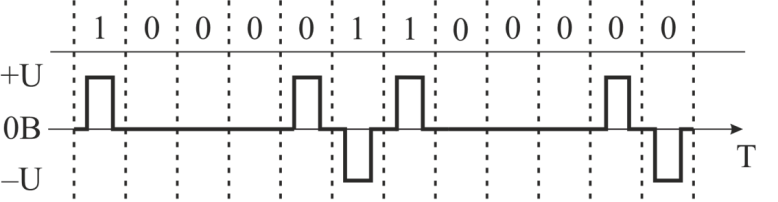
Недостатки кода *AMI*:

* усложнение реализации по сравнению с двухуровневыми кодами;
* при передаче последовательностей логических нулей устройства рассинхронизируются, в связи с чем на практике код Баркера сам по себе не используется.

В линиях цифровой телефонии применяется модифицированный код *AMI*: ***код биполярный с замещением четырех нулей*** (Bipolar with 4-Zero Substitution - B4ZS)или ***код высокой плотности биполярный порядка 3*** (High Density Bipolar 3 - HDB3).

## *Модификация AMI (HDB3)*

В непрерывной последовательности, состоящей из четырех и более логических нулей, первая тетрада (0000) заменяется двоичной последовательностью 000V, где V - логическая «1». При этом нарушается правило кодирования *AMI* (HDB3). Полярность напряжения не меняется при появлении следующего единичного импульса. Вторая и последующие тетрады, состоящие только из нулей, заменяются двоичной последовательностью B00V, где B - логическая «1» с соблюдением правил кодирования *AMI*, т.е. импульс с изменением полярности предшествующей логической «1», а V - логическая «1» с нарушением правил кодирования *AMI*, т.е. импульс с сохранением полярности предшествующей логической «1» (рис.8).



*T*

–*U*

+*U*

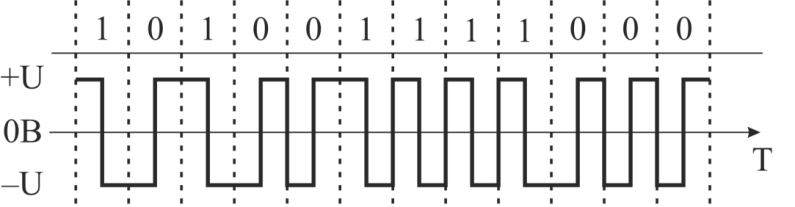
0 B

Рис.8. Код HDB3

## *Код* ***Манчестер-2***

Данный код представляет собой разновидность модуляционного кода. Полярность напряжения в середине каждого битового интервала в этом коде изменяется на противоположную. Например, в интерфейсе МИК (ГОСТ Р 52070-2003) действуют следующие правила формирования сигнала:

* логический «0» передается сменой уровня напряжения –*U* В на уровень +*U* В;
* логическая «1» передается сменой уровня напряжения +*U* В на уровень –*U* В (рис.9).



0 B

*T*

+*U*

–*U*

Рис.9. Код Манчестер-2 (МИК)

Возможно и инверсное представление такого кодирования, например, как в ЛВС Ethernet (технологии 10Base-2, 10Base-5, 10Base-TX).

Достоинства кода Манчестер-2:

* полная синхронизация;
* нет постоянной составляющей - возможна гальваническая развязка;
* обнаружение ошибок;
* хорошая помехозащищенность за счет двух выделенных частот в спектре сигнала.

Недостатки кода Манчестер-2:

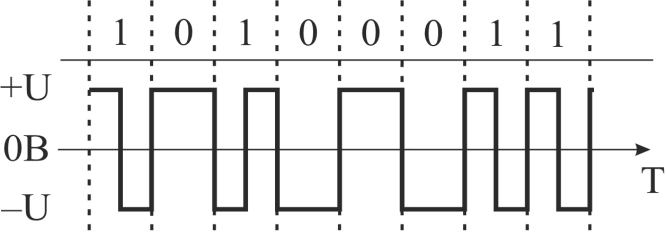
* необходима большая полоса частот;
* высокие требования к точности измерения времени (для определения половины интервала).

Код Манчестер-2 широко применяется в медных и оптоволоконных кабелях, используется протоколом локальной вычислительной сети (ЛВС) Ethernet 10 Мбит/с.

## *Частотно-модуляционный код*

В данном коде уровень напряжения в начале каждого такта изменяется на противоположный, а также действуют такие правила:

* при передаче логического «0» не изменяется уровень напряжения в середине битового интервала;
* при передаче логической «1» добавляется изменение уровня напряжения в середине битового интервала (рис.10).



0 B

*T*

–*U*

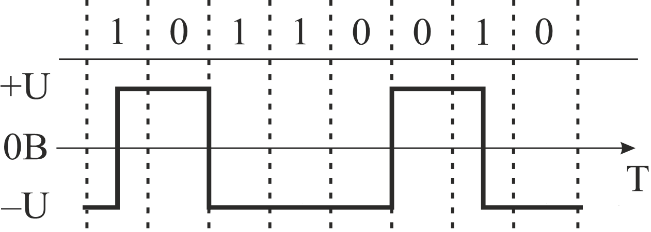
+*U*

Рис.10. Частотно-модуляционный код

***Частотно-модуляционный код*** (ЧМ-код) в основном используется при записи на магнитные носители.

## *Код RLL*

Другое название кода - *RLL*-код с ограничением длины поля записи (Run Length Limit - *RLL*). Это семейство методов кодирования. Отличия внутри семейства определяются двумя параметрами: run length - ***минимальное*** количество битовых интервалов ***между сменами*** полярности напряжения, и run limit - ***максимальное*** количество битовых интервалов ***без смены*** полярности. Рассмотрим популярный вариант *RLL* (рис.11): *RLL* (2,7). Для него составляется словарь кодирования (табл.1).



*T*

0 B

+*U*

–*U*

Рис.11. Вариант кода RLL (2,7)

**Таблица 1**

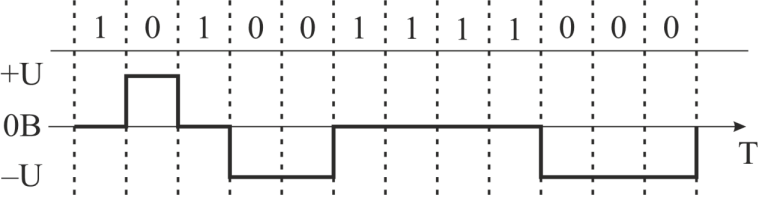
**Словарь кодирования RLL (2,7)**

|  |  |  |  |
| --- | --- | --- | --- |
| Битовая последовательность | Кодированная последовательность | Количество смен полярности на бит | Вероятность встречи в случайном потоке данных, % |
| 11 | RNNN | 1/2 | 25 |
| 10 | NRNN | 1/2 | 25 |
| 011 | NNRNNN | 1/3 | 12,5 |
| 010 | RNNRNN | 2/3 | 12,5 |
| 000 | NNNRNN | 1/3 | 12,5 |
| 0010 | NNRNNRNN | 2/4 | 6,25 |
| 0011 | NNNNRNNN | 1/4 | 6,25 |
| Примечание: R - смена полярности в начале полутакта; N - полярность не меняется. | | | |

## *Код биполярный трехуровневый*

Сигнал формируется по таким правилам:

* логическая «1» передается нулевым уровнем напряжения;
* логический «0» передается уровнем напряжения +*U* В или –*U* В, причем полярность меняется на противоположную всегда, когда перед логическим «0» следует нечетное количество логических «1» (рис.12).



*T*

0 B

+*U*

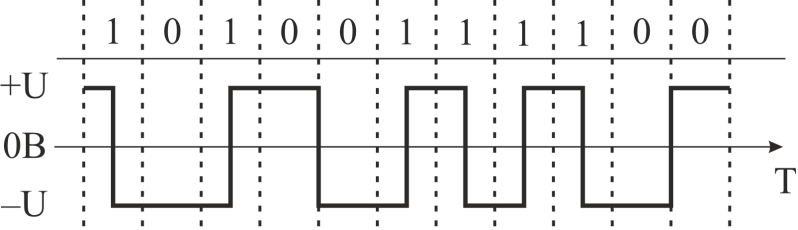
–*U*

Рис.12. Код биполярный трехуровневый

## ***Код Миллера***

Сигнал формируется по таким правилам:

* логическая «1» передается сменой полярности импульса в середине битового интервала;
* при передаче логического «0» нет смены полярности, если предыдущий бит был логической «1»;
* при передаче логического «0» происходит смена полярности в конце битового интервала, если предыдущий бит был логическим «0» (рис.13).



*T*

+*U*

–*U*

0 B

Рис.13. Код Миллера

Достоинства кода Миллера:

* полная синхронизация;
* обнаружение ошибок;
* полоса частот меньше, чем полоса частот, требуемая для кода Манчестер-2.

Недостатки кода Миллера:

* есть постоянная составляющая;
* высокие требования к точности измерения времени.

## *Код 4B5S*

Исходная логическая последовательность разбивается на тетрады. Каждой двоичной тетраде ставится в соответствие двоичный код из 5 бит по таблице кодирования (табл.2). В этом случае можно закодировать 32 различные кодовые комбинации. Для передачи данных используют только содержащие не менее двух единиц комбинации, остальные зарезервированы.

**Таблица 2**

**Словарь кодирования 4B5S**

|  |  |  |  |
| --- | --- | --- | --- |
| Исходный код | Результирующий код | Исходный код | Результирующий код |
| 0000 | 11110 | 1000 | 10010 |
| 0001 | 01001 | 1001 | 10011 |
| 0010 | 10100 | 1010 | 10110 |
| 0011 | 10101 | 1011 | 10111 |
| 0100 | 01010 | 1100 | 11010 |
| 0101 | 01011 | 1101 | 11011 |
| 0110 | 01110 | 1110 | 11100 |
| 0111 | 01111 | 1111 | 11101 |

Достоинства кода *4B5S*:

* битовая синхронизация;
* дополнительные возможности управления за счет назначения сервисных команд зарезервированным комбинациям.

Недостатки кода *4B5S*:

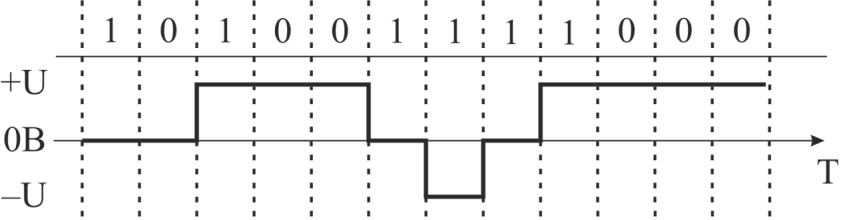
* необходима таблица кодирования;
* уменьшение скорости передачи или повышение тактовой частоты передачи по сравнению с исходной последовательностью.

Код *4B5S* используется при передаче информации по оптоволокну в ЛВС FDDI или по витой паре в сети 100Base-TX.

## ***Код MLT-3***

*Код MLT-3* (Multi Level Transmission - 3), или ***код трехуровневой передачи***, использует три уровня напряжения: +*U*, 0, –*U*. Сигнал формируется по следующим правилам:

* логический «0» передается отсутствием изменения уровня напряжения в линии;
* логическая «1» передается циклическим изменением уровня напряжения: +*U*, 0, –*U*, 0, +*U* (рис.14).



*T*

+*U*

–*U*

0 B

Рис.14. Код MLT-3

Достоинства кода *MLT-3*:

* частичная синхронизация;
* нет постоянной составляющей - возможна гальваническая развязка;
* обнаружение ошибок при нарушении правил следования единичных импульсов;
* меньшая полоса частот, чем у *NRZ*-кода.

Недостатки кода *MLT*-*3*:

* усложнение реализации по сравнению с *NRZ*-кодом;
* если необходимо передать тетраду «1111» за один такт (т.е. обеспечить четыре перехода уровня за один такт), то частоту передачи придется сделать в 4 раза меньше максимальной частоты среды;
* при передаче последовательностей логических «0» устройства рассинхронизируются.

Код *MLT-3* используется в ЛВС Ethernet 100 Мбит (100Base-TX).

# Порядок выполнения лабораторной работы

1. Разработать в любой цифровой среде граф состояний процедуры битового кодирования.
2. Реализовать кодер (варианты - в табл.3) на языке Verilog.
3. Разработать test bench для проверки модулей.
4. Проверить работу кодера на временнόй диаграмме, где входная последовательность - номер студенческого билета (например, 811031810 = = 1110010010000110011112‬).‬‬‬‬‬‬‬‬‬‬‬‬‬
5. Разработать граф состояний декодера.
6. Реализовать декодер с возможностью распознавания ошибок передачи на языке Verilog.
7. Проверить работу декодера на временнόй диаграмме, на которой входная последовательность - выходная для кодера.
8. (Дополнительное задание.) Реализовать кодер на языке VHDL.
9. Ответить на контрольные вопросы.

***Таблица 3***

**Варианты заданий**

|  |  |
| --- | --- |
| Номер варианта | Задание |
| 1, 9, 17, 25 | *NRZI*-код (два варианта, мультиплексирование кодера) |
| 2, 10, 18, 26 | Код *AMI* |
| 3, 11, 19, 27 | Код «Манчестер-2» (мультиплексирование кодера) |
| 4, 12, 20, 28 | ЧМ-код (мультиплексирование кодера) |
| 5, 13, 21, 29 | Код *RLL* (2,7) |
| 6, 14, 22, 30 | Код биполярный трехуровневый |
| 7, 15, 23, 31 | Код Миллера (мультиплексирование кодера) |
| 8, 16, 24, 32 | Код *MLT-3* |

# Контрольные вопросы

1. Где и в каких случаях применяется кодирование информации?
2. В чем заключаются преимущества и недостатки алфавитных кодов?
3. Как происходит кодирование информации методом *NRZ*?
4. Достоинства и недостатки кода *NRZ*.
5. Как происходит кодирование информации методом *RZ*?
6. Перечислите достоинства и недостатки кода *RZ*.
7. Опишите способ кодирования информации методом *NRZI*.
8. Опишите способ кодирования информации методом *AMI*.
9. Перечислите достоинства и недостатки кода *AMI*.
10. Опишите способ кодирования информации методом Манчестер-2.
11. Перечислите достоинства и недостатки кода Манчестер-2.
12. Опишите способ кодирования информации частотно-модуляционным методом.
13. Связь частотно-модуляционного и *RLL*-кодов.
14. Как происходит кодирование информации методом Миллера?
15. В чем заключаются достоинства и недостатки кода Миллера?
16. Каковы преимущества и недостатки избыточного кода?
17. Опишите способ кодирования информации методом биполярным трехуровневым.
18. Опишите способ кодирования информации методом *MLT-3*.
19. Каковы достоинства и недостатки кода *MLT-3*?
20. Какой код называется двухуровневым?
21. Какой код называется трехуровневым?
22. Перечислите примеры двухуровневых кодов.
23. Перечислите примеры трехуровневых кодов.
24. В чем заключаются преимущества и недостатки многоуровневых кодов?

# Лабораторная работа № 2

# Изучение и реализация интерфейсов I2C, SPI

***Цель работы*:** изучить интерфейсы I2C и SPI.

***Продолжительность работы*:** 3 часа.

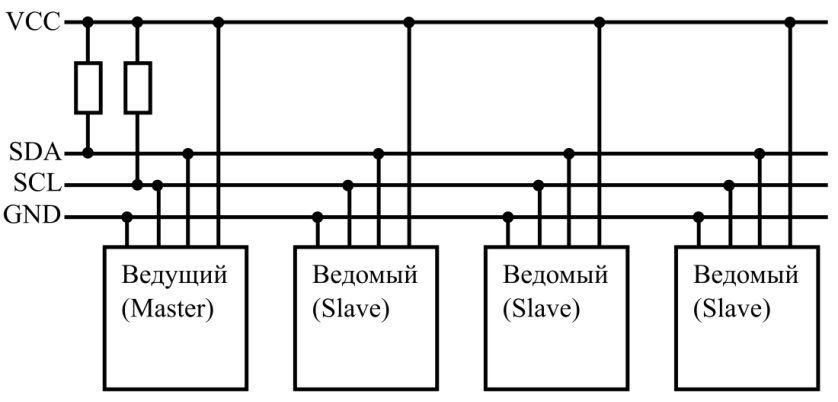
# Теоретические сведения

### ***Интерфейс I2C***

***I²C*** (Inter-Integrated Circuit) - межкомпонентный двухпроводной интерфейс, представляющий собой синхронную последовательную шину, которая обеспечивает двустороннюю передачу данных между подключенными устройствами. Интерфейс I²C состоит из двух однобитных сигнальных линий:

1. данные (Serial Data, SDA);
2. синхронизация (Serial Clock, SCL).

На рис.1 представлен пример соединения устройств по шине I²C: один ведущий (Master) микроконтроллер и три ведомых (Slave) устройства (АЦП, ЦАП, МК).



*Рис.1.* ШинаI2C

В число I²C-совместимых устройств входят ИС памяти, видеопроцессоров и модулей обработки аудио- и видеосигналов, аналогоцифровых и цифро-аналоговых преобразователей, драйверы ЖК-индикаторов, процессоры со встроенным аппаратным контроллером I²C шины и т.д. Интерфейс I²C позволяет работать и в режиме «мультимастер» - когда на шине имеется несколько Master-устройств.

Перед началом передачи данных на ведущее устройство поступает информация о том, что шина свободна, т.е. сигналы SCL и SDA имеют высокий уровень. После этого Master-устройство формирует условие СТАРТ (Start) - перевод сигнала SDA в низкий уровень при сохранении высокого уровня SCL. Завершается передача данных введением ведущим устройством условия СТОП (Stop) - сигнал SDA переводится в высокий уровень при сохранении высокого уровня SCL. На рис.2 изображены условия СТАРТ и СТОП.

2.2..tif

*Рис.2.* Временнáя диаграмма обмена по шине I2C

На рис.3 и 4 представлены запись и чтение по шине I2C.

2.3..tif

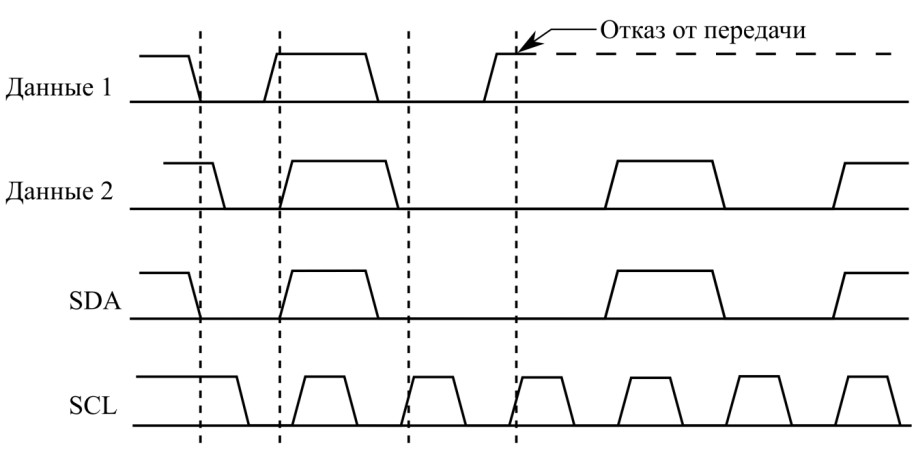
*Рис.3.* Запись по шине I2C

Slave адрес - адрес, указанный в технической документации ведомого устройства. После адреса указывается бит чтения («1») или записи («0»). Ответ - ACK («0») или NACK («1»), после битов адреса и бита операции ведомое устройство выставляет ответ ACK, чтобы проинформировать Master-устройство о том, что Slave-устройство работает и готово к обмену. Чтение отличается от записи тем, что после каждого считанного байта ответ на шину выставляет Master-устройство - ACK для продолжения чтения, NACK для остановки.

*2.4..tif*

*Рис.4.* Чтение по шине интерфейса I2C

Арбитраж в интерфейсе I2C выполняется на линии SDA при высоком уровне SCL (рис.5). Устройство, которое формирует на линии SDA высокий уровень, в то время как другое передает низкий, теряет право быть ведущим и должно перейти в режим ведомого.



*Рис.5.* Арбитраж интерфейса I2C

Достоинства интерфейса I2C:

* требуется только две линии - линия данных (SDA) и линия синхронизации (SCL). Каждое устройство, подключенное к шине, может быть программно-адресовано по уникальному адресу. В каждый момент времени существует простое отношение ведущий/ведомый: ведущие устройства могут работать как ведущий-передатчик и ведущий-приемник;
* шина позволяет иметь несколько ведущих устройств, предоставляя средства для определения коллизий и арбитраж для предотвращения повреждения данных в ситуации, когда два или более ведущих одновременно начинают передачу данных;
* относительно высокие скорости обмена - от 100 Кбит/с до 3,4 Мбит/с, соединение друг с другом устройств с различными напряжениями питания.

Недостатки интерфейса I2C:

* максимально допустимое количество микросхем, подсоединенных к одной шине, ограничивается максимальной емкостью шины 400 пФ;
* трудность локализации неисправности, если одно из подключенных устройств ошибочно устанавливает на шине состояние низкого уровня.

### ***Интерфейс SPI***

***SPI*** (Serial Peripheral Interface) - синхронный последовательный периферийный интерфейс.

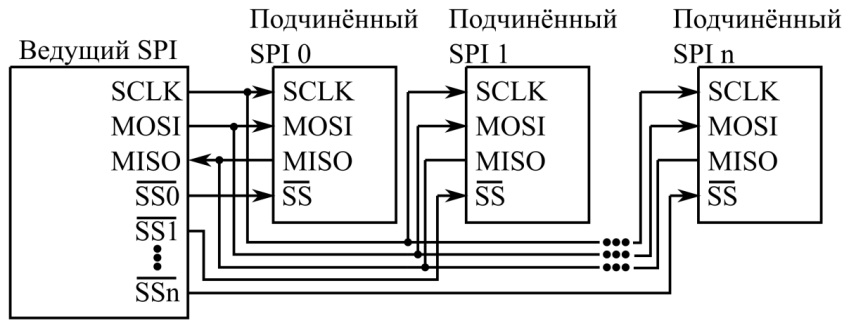
В SPI используется четыре цифровых сигнала:

1. MOSI - выход ведущего, вход ведомого (Master Out Slave In). Служит для передачи данных от ведущего устройства ведомому;
2. MISO - вход ведущего, выход ведомого (Master In Slave Out). Служит для передачи данных от ведомого устройства ведущему;
3. SCLK - последовательный тактовый сигнал (Serial Clock). Служит для передачи тактового сигнала для ведомых устройств;
4. CS или SS - выбор микросхемы, выбор ведомого (Chip Select, Slave Select).

Известны следующие варианты подключения нескольких устройств на шине SPI:

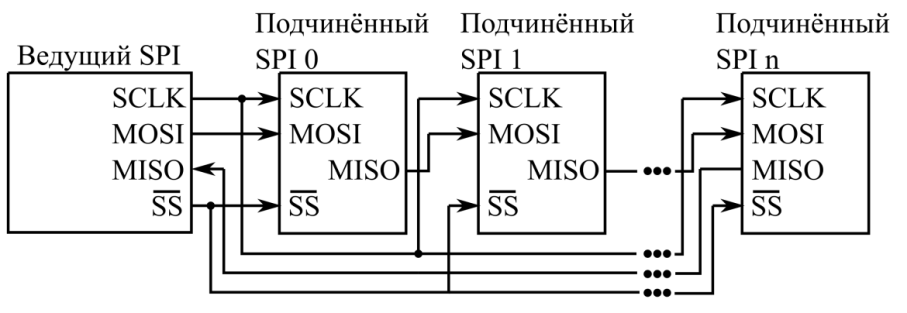
* независимое (параллельное) подключение (рис.6);
* каскадное (последовательное) подключение (рис.7).

При параллельном подключении для каждого Slave-устройства выделяется своя линия SS (CS), Master-устройство устанавливает на линии SS выбранного Slave-устройства высокий уровень сигнала для начала сеанса обмена. Очевидно, что добавление каждого нового устройства на шине сопряжено с необходимостью выделения дополнительной линии SS.



Подчиненный Подчиненный Подчиненный

*Рис.6.* Параллельное подключение SPI



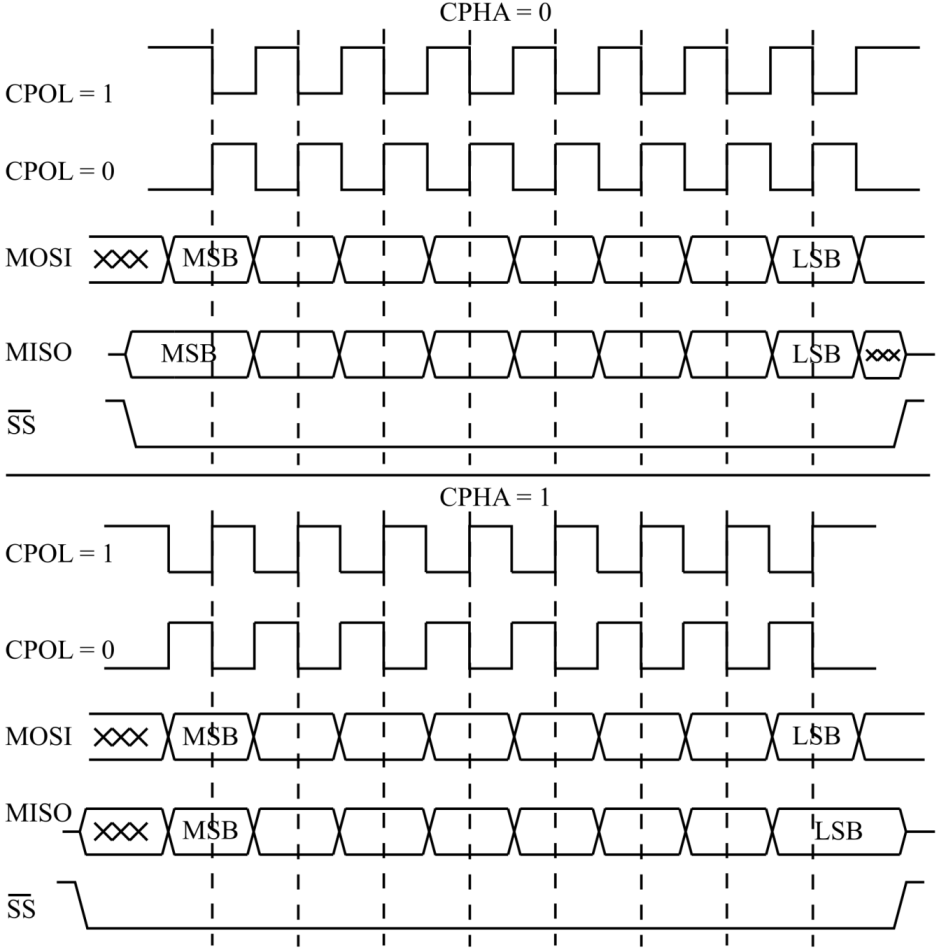
Подчиненный Подчиненный Подчиненный

*Рис.7.* Последовательное подключение SPI

При последовательном подключении используется лишь одна линия SS, но данные поступают поочередно во все Slave-устройства, т.е. первый байт данных предназначается последнему устройству, а последний - первому. Master отправляет большой пакет данных сразу для всех ведомых устройств - первый Slave забирает свою часть данных, а остальные ретранслирует следующему устройству. При таком подключении сдвиговые регистры устройств образуют один большой регистр.

Режим работы интерфейса определяется полярностью (CPOL - Clock Polarity) и фазой (CPHA - Clock Phase) установки данных относительно SCLK. Шина SPI (рис.8) работает в четырех режимах:

* Mode 0 (режим 0): CPOL = 0, CPHA = 0. Тактовая последовательность SCLK начинается с формирования положительного фронта. Считывание данных осуществляется по положительному фронту на линии SCLK. Смена данных по отрицательному фронту;
* Mode 1 (режим 1): CPOL = 0, CPHA = 1. Тактовая последовательность SCLK начинается с формирования отрицательного фронта. Считывание данных осуществляется по отрицательному фронту на линии SCLK. Смена данных по положительному фронту;
* Mode 2 (режим 2): CPOL = 1, CPHA = 0. Тактовая последовательность SCLK начинается с формирования положительного фронта. Считывание данных осуществляется по отрицательному фронту на линии SCLK. Смена данных по положительному фронту;
* Mode 3 (режим 3): CPOL = 1, CPHA = 1. Тактовая последовательность SCLK начинается с формирования отрицательного фронта. Считывание данных осуществляется по положительному фронту на линии SCLK. Смена данных по отрицательному фронту.



*Рис.8.* Режимы работы SPI. Пунктиром изображен момент считывания данных

В интерфейсе SPI ведомое устройство не может самостоятельно управлять потоком данных: путем управления линией SS Master-устройство принимает решение о том, какой из Slave-устройств посылает информацию в шину SPI в текущий момент, поэтому арбитраж не нужен.

Достоинства интерфейса SPI:

* полнодуплексный обмен;
* скорость обмена выше, чем у I²C;
* все линии шины SPI являются однонаправленными, что существенно упрощает решение задачи преобразования уровней и гальванической изоляции микросхем.

Недостатки интерфейса SPI:

* необходимо больше выводов, чем для интерфейса I²C;
* ведомое устройство не может управлять потоком данных;
* нет подтверждения приема данных со стороны ведомого устройства;
* нет определенного стандартом протокола обнаружения ошибок;
* отсутствие официального стандарта, что делает невозможным сертификацию устройств.

# Порядок выполнения лабораторной работы

1. Построить структурную схему взаимодействия устройств со всеми используемыми сигналами.
2. Реализовать интерфейс на языке Verilog, согласно варианту задания (I²C/SPI), без использования IP-core, сторонних библиотек и готовых решений.
3. Выполнить задания, указанные в варианте, - написать test bench для построения временнόй диаграммы работы ведущего устройства и симуляции работы ведомых устройств. Информация, принимаемая с датчиков/flash, может быть произвольной, но должна соответствовать диапазону значений, указанному в технической документации.

### Вариант № 1. I2C.

Используемые модули: датчик MPU6000, символьный дисплей LCD1602.

Задания для test bench:

1. Очистить дисплей LCD 1602.
2. На первой строке дисплея вывести название регистра и адрес датчика в виде «’Название регистра’(Пробел) ADDR\_XX». Вместо символов XX вывести адрес датчика на шине.
3. Вывод информации из регистра FIFO\_R\_W датчика MPU6000 на LCD 1602A на второй строке экрана с выравниванием по центру.

### Вариант № 2. SPI.

Используемые модули: Flash-память W25Q16, 7-сегментный индикатор, сдвиговый регистр 74HC595, датчик MPU6000.

Задания для test bench:

1. Считать 4 байта данных из Flash-памяти W25Q16.
2. Вывести полученные данные на восемь 7-сегментных индикаторов, подключенных через сдвиговые регистры 74HC595.
3. Записать данные в Flash-памяти W25Q16 по любому доступному адресу.
4. Считать данные из Flash-памяти W25Q16 по любому доступному адресу, используя команду Fast Read.
5. Вывести полученные данные на 7-сегментные индикаторы.
6. Считать данные из MPU6000 по адресам 114-117.
7. Вывести полученные данные на 7-сегментные индикаторы.

Подключить 7-сегментные индикаторы по последовательной схеме, остальные устройства - по параллельной схеме.

### Вариант № 3. I2C.

Используемые модули: PAJ7620, RFID-reader RCC522, ЖКИ 1602A. (Инициализацию экрана проводить не нужно.)

Задания для test bench:

1. Считать версию датчика жестов PAJ7620 (24 бит).
2. Записать полученный результат в FIFO буфер RFID-reader RCC522.
3. Очистить экран ЖКИ 1602A.
4. Из Interrupt регистра датчика жестов PAJ7620 считать значения двух флагов.
5. На первой строке с выравниванием по центру экрана вывести строку «Lab. 2».
6. Записать полученную информацию на второй строке   
   ЖКИ 1602A с выравниванием по центру экрана в формате «Flag 0x\*\*(Пробел)0x\*\*».

### Вариант № 4. SPI.

Используемые модули: датчик температуры DS1722, АЦП TLA2518, 7-сегментный индикатор, сдвиговый регистр 74HC595.

Задания для test bench:

1. Настроить конфигурацию датчика температуры DS1722 на непрерывный режим с 11-битным разрешением измерений.
2. Считать температуру.
3. Настроить в ручной режим оцифровки температуры (shutdown mode).
4. Считать обновленные данные температуры.
5. Настроить внешний АЦП TLA2518. Считать, что это первая команда после включения устройства. (CPHA = 1, CPOL = 1.)
6. Считать регистр системных статусов АЦП TLA2518.
7. Вывести полученную информацию на два 7-сегментных индикатора.

Подключение 7-сегментных индикаторов осуществить по последовательной схеме, подключение остальных устройств - по параллельной схеме.

**Контрольные вопросы**

1. Что такое интерфейс?

2. Интерфейс I2C. Устройство шины. Open-drain выход.

3. Интерфейс I2C. Протокол обмена. Временны́е диаграммы работы.

4. Интерфейс I2C. Протокол обмена. Арбитраж.

5. Интерфейс I2C. Достоинства и недостатки.

6. Интерфейс SPI. Устройство шины. Варианты подключения.

7. Интерфейс SPI. Протокол обмена. Временны́е диаграммы работы.

8. Интерфейс SPI. Протокол обмена. Режимы работы интерфейса.

9. Интерфейс SPI. Достоинства и недостатки.

10. Что такой битовый интервал? Z-состояние? Конечный автомат состояний?

**Содержание**

**Предисловие** 3

**Лабораторная работа № 1.** Изучение и реализация процедур арифметического кодирования. Функциональное тестирование 5

**Лабораторная работа № 2.** Изучение и реализация интерфейсов   
I2C, SPI 19

Учебное издание

***Ухандеев*** *Владимир Ильич*

***Собченко*** *Максим Иванович*

***Корявина*** *Анастасия Сергеевна*

**Методические указания к лабораторным работам по курсу   
«Интерфейсы вычислительных систем»**

Редактор *Н.А. Кузнецова.* Технический редактор *Л.Г. Лосякова*. Корректор *Л.Г. Лосякова.* Верстка авторов.

Подписано в печать с оригинал-макета 25.02.2022. Формат 60×84 1/16. Печать офсетная. Бумага офсетная. Гарнитура Times New Roman. Усл. печ. л. 1,62.   
Уч.-изд. л. 1,4. Тираж 100 экз. Заказ 11.

Отпечатано в типографии ИПК МИЭТ.  
124498, г. Москва, г. Зеленоград, площадь Шокина, дом 1, МИЭТ.