Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра радиотехнических систем (РТС)

СИСТЕМАТИЧЕСКОЕ ЦИКЛИЧЕСКОЕ КОДИРОВАНИЕ

Учебно-методическое пособие
по лабораторной и самостоятельной работе и практическим занятиям
для студентов направления
«Инфокоммуникационные технологии и системы связи»

Томск 2018

Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра радиотехнических систем (РТС)

Утверждаю:
Зав. каф. РТС, проф., д.т.н.
_____Мелихов С.В.
___ _____2018 г.

СИСТЕМАТИЧЕСКОЕ ЦИКЛИЧЕСКОЕ КОДИРОВАНИЕ

Учебно-методическое пособие по лабораторной и самостоятельной работе и практическим занятиям
для студентов направления
«Инфокоммуникационные технологии и системы связи»

Разработчики:
Доц. каф. РТС Кологривов В.А._____
Студентка гр. 1В4 Гердт К.А._____

Томск 2018

Кологривов В.А., Гердт К.А.

«Систематическое циклическое кодирование»: Учебно-методическое пособие по лабораторной и самостоятельной работе и практическим занятиям для студентов направления «Инфокоммуникационные технологии и системы связи». – Томск: ТУСУР. Образовательный портал, 2018.- 20 с.

Учебно-методическое пособие содержит описание функциональной модели кодека, основанного на блочном систематическом циклическом кодировании, выполненного в среде функционального моделирования *Simulink* системы для инженерных и научных расчетов *MatLab*.

АННОТАЦИЯ

Лабораторная работа «Систематическое циклическое кодирование» посвящена экспериментальному исследованию функциональной модели кодека, основанного на блочном циклическом кодировании с использованием пакета функционального моделирования *Simulink* системы для инженерных и научных расчетов *MatLab.*

Работа «Систематическое циклическое кодирование» относится к циклу лабораторных работ по разделу «Помехоустойчивое кодирование», входящему в дисциплины по направлению «Инфокоммуникационные технологии и системы связи».

В описании сформулирована цель работы, приведены краткие теоретические сведения о циклическом кодировании, краткая характеристика пакета *Simulink* системы *MatLab,* описание лабораторного макета и используемых блоков библиотеки *Simulink,* а также требования к экспериментальному исследованию и контрольные вопросы, ответы на которые необходимы для успешной защиты лабораторной работы.

СОДЕРЖАНИЕ

1 ЦЕЛЬ РАБОТЫ. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ ПО СИСТЕМАТИЧЕСКОМУ ЦИКЛИЧЕСКОМУ КОДИРОВАНИЮ

Цель работы: изучение структуры и принципа организации кодека с использованием систематического циклического кодирования при использовании пакета функционального моделирования *Simulink.*

Теоретические сведения систематического циклического кодирования

Циклические коды составляют большую группу наиболее широко используемых на практике кодов. Их основное свойство, давшее им название, состоит в том, что каждый вектор, получаемый из исходного кодового вектора путём циклической перестановки его символов, также является разрешённым кодовым вектором. Принято описывать циклические коды при помощи порождающих полиномов $g(x)$ степени $r = n — k$, где r — число проверочных символов в кодовом слове.

При циклической перестановке символы кодового слова перемещаются слева направо на одну позицию, причем, крайний справа символ переносится на место крайнего левого.

Операции кодирования и декодирования циклических кодов сводятся к известным процедурам умножения и деления полиномов. Согласно определению все многочлены, соответствующие его кодовым комбинациям, должны делиться на $g(x)$ без остатка [1].

На практике используются два способа кодирования. Оба используют циклические коды с порождающим полиномом
$g(x) = a_0 + a_1 x + \cdots + a_{r-1} x^{r-1} + x^r$, который удовлетворяет двум условиям:

1) $g(x)$ является нетривиальным делителем полинома $x^n - 1$;

2) его степень r связана с числом информационных разрядов k равенством: $k = n - r$.

В данной лабораторной работе рассматривается только систематическое циклическое кодирование.

Систематические коды характеризуются тем, что сумма по модулю 2 двух разрешённых кодовых комбинаций кодов этого класса снова даёт разрешённую кодовую комбинацию.

Кроме того, в систематических кодах информационные символы, как правило, не изменяются при кодировании и занимают определённые заранее заданные места.

Биты чётности (проверочные символы) образуются с помощью деления, предварительно сдвинутых на r разрядов информационных символов, на порождающий полином. Эти избыточные биты выталкиваются последующими символами из кодера, вслед за информационными.

При данном виде кодирования в кодовом символе содержится исходный код [2].

Главным плюсом циклических кодов является возможность, при небольшой избыточности и простой аппаратурной реализации, получение *высокой вероятности обнаружения ошибок*.

Обнаружение одиночных ошибок

Любая принятая по каналу связи кодовая комбинация $h(x)$, возможно содержащая ошибку, может быть представлена в виде суммы по модулю два неискаженной комбинации кода $f(x)$ и вектора ошибки $\mathrm{E}(x)$:

$$h(x) = f(x) \oplus E(x).$$

При делении $h(x)$ на образующий многочлен $g(x)$ остаток, указывающий на наличие ошибки, обнаруживается только в случае, если многочлен, соответствующий вектору ошибки, не делится на $g(x)$: $f(x)$ — неискаженная комбинация кода, следовательно, делится на $g(x)$ без остатка [3].

Вектор одиночной ошибки имеет единицу в искаженном разряде и нули во всех остальных разрядах. Таким образом, при любом числе информационных разрядов необходим только один проверочный разряд. Значение символа этого разряда как раз и обеспечивает четность числа единиц в любой разрешенной кодовой комбинации, а, следовательно, и делимость ее на $x^n + 1$.

2 КРАТКОЕ ОПИСАНИЕ ФУНКЦИОНАЛЬНОЙ SIM-МОДЕЛИ КОДЕКА ОСНОВАННОЙ НА БЛОЧНОМ ЦИКЛИЧЕСКОМ КОДИРОВАНИИ

Вариант реализации SIM-модели кодека, при использовании систематического циклического кодирования, представлен на рисунке 2.1.

Исходный сигнал формируется с помощью блоков **«Random Number»** и блока сравнения с нулем**.** Далее информационный символ формируется блоком **«Buffer»** и блоком **«DeMux»** разделяется на биты. К информационной части добавляются дополнительные нулевые биты, реализующие сдвиг степени полинома, получаемый в процессе последующего умножения на образующий полином. Блок **«Multiport Switch»** используется для изменения скорости потока с избыточными битами.

После ключа, биты поступают на кодер, выполненный на основе регистра сдвига с обратной связью, задаваемый порождающим полиномом и одновременно проходят дальше на выход кодера. Далее с помощью блоков **«Pulse Generator»** и **«XOR»** реализуется имитатор ошибок.

Декодирование происходит путем деления принятых битов на порождающий полином в аналогичном регистре сдвига с обратными связями. На выходе декодера образуется синдром в виде остатка от деления.

Блок **«Combination Logic»** устанавливает соответствие синдромов и векторов ошибок. Эти вектора ошибок используются для исправления

блоком **«XOR»** принятых символов. В результате на выходе декодера после обратного изменения скорости потока получаем информационные биты.

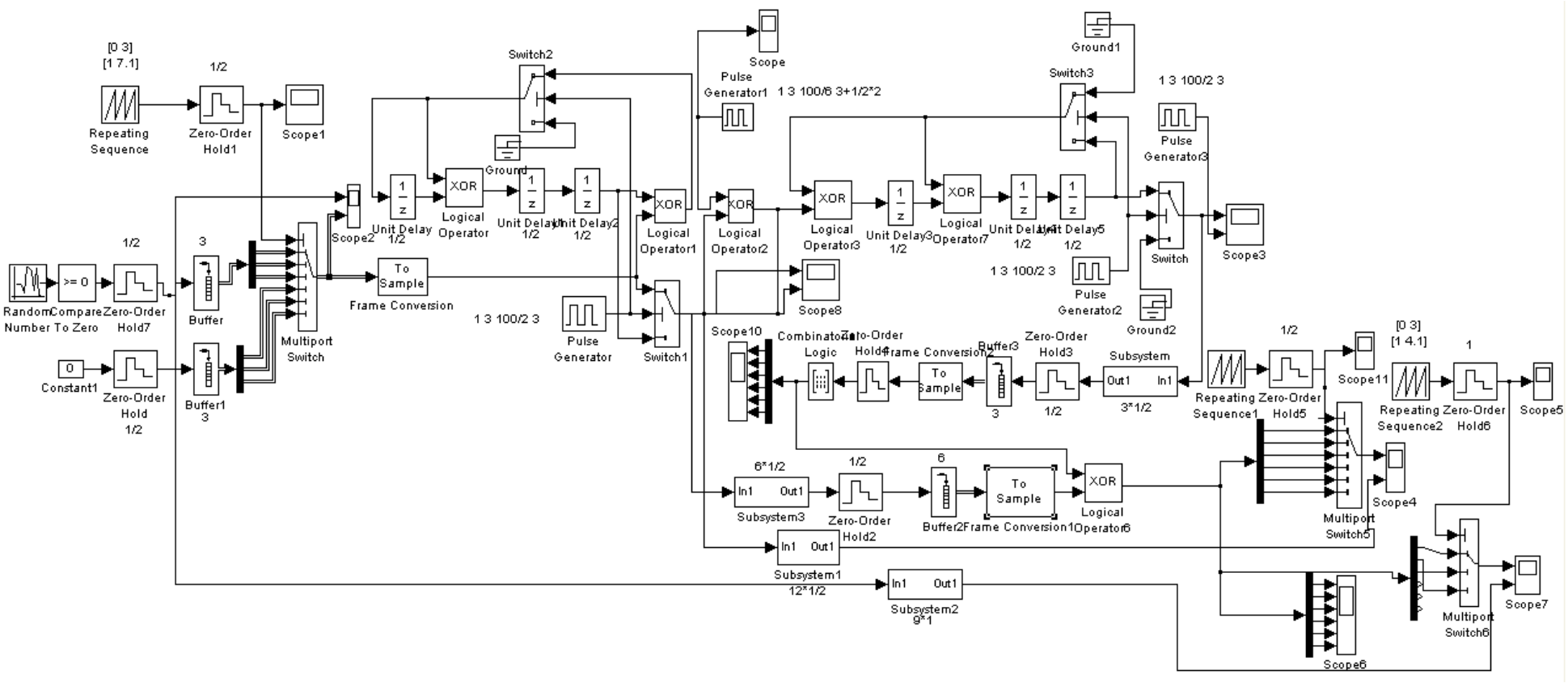


Рисунок 2.1 – Функциональная SIM-модель систематического циклического кодека (6,3)

Принцип работы схемы:

Сигнал, поступающий на вход, проходит на выход кодера и одновременно подаётся в цепь обратной связи регистра сдвига, где происходит кодирование (деление на порождающий полином), получая в качестве остатка в регистрах *биты чётности*. При поступлении нового символа, эти биты чётности подаются на выход. Далее кодовый символ поступает на имитатор ошибок и после на декодер. После декодирования (деления на порождающий полином) в регистрах образуется остаток, называемый синдром, который выталкивается на выход следующим символом. Синдром определяет вектор ошибки. Вектор ошибки, суммируясь с информационной частью принятого символа, исправляет ее. В итоге на выходе получаем исходные информационные биты.

3 КРАТКОЕ ОПИСАНИЕ ПАКЕТА SIMULINK И ИСПОЛЬЗУЕМЫХ БЛОКОВ

Для запуска системы Simulink необходимо предварительно выполнить запуск системы MatLab. После открытия командного окна системы MatLab нужно запустить систему Simulink. Это можно сделать одним из трех способов:

- нажать кнопку (Simulink) на панели инструментов системы MatLab;
- в строке командного окна MatLab напечатать Simulink и нажать клавишу Enter;
- выполнить опцию Open в меню File и открыть файл модели (mdl-файл).

Последний способ предпочтителен при запуске уже готовой и отлаженной модели, когда требуется лишь провести моделирование и не нужно добавлять новые блоки в модель. При применении двух первых способов открывается окно обозревателя библиотеки блоков (Simulink Library Browser) [4].

Работа Simulink происходит на фоне открытого окна системы MatLab, закрытие которого приведёт к выходу из Simulink.

Краткое описание используемых блоков:

Ниже описаны основные блоки базовых разделов библиотеки Simulink, используемые в функциональной схеме:

- 

Random Number – источник случайного сигнала с нормальным распределением уровня. Назначение: формирование сигнала с равномерным распределением уровня. Параметры блока: Mean – среднее значение; Variance – дисперсия; Initial seed – Начальное значение генератора случайного сигнала; Sample time – такт дискретности; флажок

Interpreted vector parameters as 1 – D интерпретировать вектор как массив скаляров.

- 

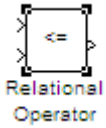
Sign – блок определения знака сигнала. Назначение: определяет знак входного сигнала, при этом, если - входной сигнал, то сигнал на выходе определяется выражением

$$sign(x)=\begin{cases}-1 & при\ \ x<0,\\ 0 & при\ \ x=0,\\ 1 & при\ \ x>0.\end{cases}$$

Параметры блока: флажок - Enable zero crossing detection позволяет фиксировать прохождение сигнала через нулевой уровень.

- 

Scope – блок осциллографа. Назначение: построение графиков исследуемых сигналов как функций времени. Открытие окна осциллографа производится двойным щелчком ЛКМ на пиктограмме блока. В случае векторного сигнала каждая компонента вектора отображается отдельным цветом. Настройка окна осциллографа выполняется с помощью панелей инструментов, позволяющих: осуществить печать содержимого окна осциллографа; установить параметры, в частности, Number of axes - число входов осциллографа, Time range – отображаемый временной интервал и другие; изменить масштабы графиков; установить и сохранить настройки; перевести в плавающий режим и так далее.

- 

Relational Operator – блок выполнения операций отношения. Назначение: сравнение текущих значений входных сигналов поступающих на входы. Параметры блока: Relational Operator –тип операции отношения выбираемый из списка:

== - тождественно равно;

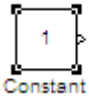
~ = - не равно;

< - меньше;

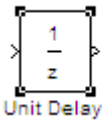
<= - меньше или равно;

>= - больше или равно;

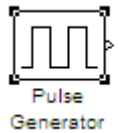
>- больше.

- 

Constant – источник постоянного сигнала. Назначение: задает сигнал постоянного уровня. Параметры блока: Constant value – постоянная величина, значение которой может быть задано действительным или комплексным числом, вычисляемым выражением, вектором или массивом; флажок Interpret vector parameters as 1 - D – интерпретировать вектор как массив скаляров; флажок Show additional parameters – показать дополнительные параметры, в нашем случае не используется.

- 

Unit delay – блок дискретной задержки. Назначение: выполняет задержку дискретного сигнала на заданный такт дискретности. Параметры блока: Initial conditions – начальное значение выходного сигнала; Sample time – такт дискретности (при задании значения параметра равного -1 такт дискретности наследуется от предшествующего блока).

- 

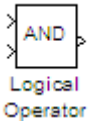
Pulse generator – блок источника импульсного сигнала. Назначение: формирование сигнала в форме прямоугольных импульсов. Параметры блока: Pulse Type – способ формирования сигнала, может принимать два значения: Time-based – по текущему времени; Samplebased – по величине такта дискретности и количеству шагов моделирования.

Вид окна параметров зависит от выбранного способа формирования сигнала. Amplitude – амплитуда; Period – период, задается в секундах при способе Time-based или количеством тактов при способе Sample-based; Pulse width – ширина импульса, задается в процентах от периода при способе Timebased или количеством тактов при способе Sample-based; Phase delay – фазовая задержка, задается в секундах при способе Time-based или количеством тактов при способе Sample-based; Sample time – такт дискретности; флажок Interpret vector parameters as 1 - D – интерпретировать вектор как массив скаляров.

-  *Demux* – блок демультиплексора. Назначение: разделение входного векторного сигнала на составляющие (последовательного представления в параллельное). Параметры блока: Number of output – количество выходов; Display option – способ отображения выбирается из списка: bar – вертикальный узкий прямоугольник черного цвета; none – прямоугольник с белым фоном без отображения меток входных сигналов. Флажок Bus Selection Mode – режим разделения векторных сигналов в шине, используется для разделения сигналов, объединенных в шину.

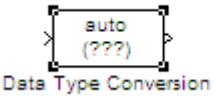
- 

Combinatorial Logic – блок комбинаторной логики. Назначение: преобразует входной векторный сигнал в соответствии с таблицей истинности. Таблица истинности представляет собой список возможных выходных значений блока. Каждому состоянию входного векторного сигнала соответствует определенное логическое состояние выходного сигнала. Параметры блока: Truth table – таблица истинности.

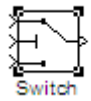
- 

Logical Operation - блок выполнения логических операций. Назначение: реализует одну из базовых логических операций. Параметры

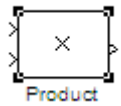
блока: Operator - вид реализуемой логической операции – выбирается из списка: AND - логическое умножение (операция логическое И), OR-логическое сложение (операция логическое ИЛИ), NAND- операция И-НЕ, NOR - операция ИЛИ-НЕ, XOR -операция сложения по модулю 2 (операция ИСКЛЮЧАЮЩЕЕ ИЛИ), NOT - логическое отрицание (логическое НЕ); Number of input ports-количество входных портов; Флажок Show additional parameters – показать дополнительные параметры (в нашем случае не используется); Флажок Require all inputs to have same data type- установить одинаковый тип входных данных; Output data type mode - выбор типа выходных данных из списка: Boolean (двоичный), Logical (логический), Specify via dialog (задаваемый дополнительным списком). В последнем случае появится окно списка Output data type- тип выходных данных. Входные сигналы могут быть как действительного, так и логического типа (Boolean). Выходным сигналом блока является 1, если результат вычисления логической операции есть ИСТИНА, и 0, если результат –ЛОЖЬ.

- 

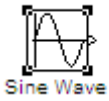
Data Type Conversion – блок преобразователя типа сигнала. Назначение: преобразует тип входного сигнала. Параметры блока: Data type – тип данных выходного сигнала, может принимать значения из писка: auto; double; single; int8; int16; int32/ uint8; uint16; uint32; Boolean. Saturate on integer – подавлять переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно. Значение auto параметра Data type используется при наследовании выходным сигналом типа входного сигнала. Входной сигнал блока может быть действительным или комплексным, в случае последнего выходной сигнал также комплексный. Входные сигналы могут быть в скалярной, векторной и матричной формах.

- 

Switch – блок переключателя. Назначение: переключение входных сигналов по сигналу управления. Параметры блока: Criteria for passing first input – условие прохождения сигнала с первого входа, значение выбирается из списка: u2>=Threshold – сигнал управления больше или равен пороговому значению; u2>Threshold – сигнал управления больше порогового значения; u2~=Threshold – сигнал управления не равен пороговому значению. Threshold – порог; флажок Show additional parameters – показать дополнительные параметры. При выставленном флажке отображаются дополнительные окна списков, в нашем случае флажок не используется.

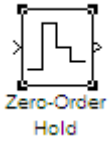
- 

Product – блок умножения и деления. Назначение: вычислени произведения текущих значений сигналов. Параметры блока: Mumber of inputs – количество входов, может задаваться как число или как список знаков. В списке знаков можно использовать знаки: * - умножить и / - разделить. Multiplication – способ выполнения операции, может принимать значения из списка: Element-wise – поэлементный; Matrix – матричный. Флажок Show additional parameters – показать дополнительные параметры. При выставленном флажке отображается окно списка Output data type mode, в нашем случае флажок не используется.

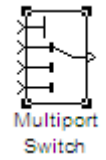
- 

Sine Wave – блок источника синусоидального сигнала. Назначение: формирование синусоидального сигнала с заданной частотой, амплитудой, фазой и смещением. Параметры блока: Sine Type – способ формирования сигнала реализуется двумя алгоритмами: Time-based – по текущему времени (для аналоговых систем) или по значению

сигнала на предыдущем шаге и величине такта дискретности (для дискретных систем); Sample-based – по величине такта дискретности и количеству расчетных шагов на один период синусоидального сигнала. Вид окна задания параметров меняется в зависимости от выбранного способа формирования синусоидального сигнала.

- 

Zero-Older Hold – экстраполятор нулевого порядка. Назначение: экстраполяция входного сигнала на интервале дискретизации. Блок фиксирует значение входного сигнала в начале интервала дискретизации и поддерживает на выходе это значение до окончания интервала дискретизации. Затем выходной сигнал изменяется скачком до величины входного сигнала на следующем шаге дискретизации. Параметры блока: Sample time – такт дискретности. Блок экстраполятора нулевого порядка может использоваться также для согласования работы дискретных блоков, имеющих разные такты дискретности.

- 

Multiport Switch – блок многовходового переключателя. Назначение: выполняет переключение входных сигналов на выход по сигналу управления, задающему номер активного входного порта. Параметры блока: Number of inputs – количество входов; флажок Show additional parameters – показать дополнительные параметры, в нашем случае не используется. Блок Multiport Switch пропускает на выход сигнал с того входного порта, номер которого равен текущему значению управляющего сигнала. Если управляющий сигнал не является сигналом целого типа, то блок Multiport Switch производит округление значения в соответствии со способом, выбранным в графе дополнительного параметра Round integer calculations toward.

4 ЭКСПЕРИМЕНТАЛЬНОЕ ЗАДАНИЕ

1. Собрать SIM-модель для исследования принципа функционирования кодека основанного на систематическом циклическом кодировании в соответствии с рисунком 2.1;
2. Вручную расчитать синдромы для порождающего полинома $g(x) = x^3 + x + 1$ и информационных битов ‘1 0 1′ для систематического кода (6,3) и занести результаты в таблицу соответствий;
3. Выставить параметры блоков SIM-модели, в соответствии с рисунком 2.1, на котором указан один из вариантов параметров схемы;
4. Пронаблюдать, зафиксировать и пояснить основные осциллограммы, иллюстрирующие работу кодека, на основе систематического циклического кодирования.

5 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. На чем основывается циклическое кодирование?
2. В чем заключается систематическое циклическое кодирование?
3. Какие достоинства циклического кодирования?
4. В чем отличие систематического кодирования от несистематического?
5. Как происходит обнаружение одиночных ошибок?
6. Какой вид имеет вектор одиночной ошибки?
7. Что такое *синдром* и как он образуется?
8. В каком блоке хранится таблица соответствия?
9. Как связаны регистр сдвига с обратной связью и порождающий полином?
10. В чем различие подключения регистров сдвига с обратными связями при умножении и делении на образующий полином?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Никитин Г.И. Помехоустойчивые циклические коды: учебное пособие – Санкт-Петербург: СПбГУАП, 2003.- 33 с.
2. Циклические коды [Электронный ресурс]. – Режим доступа: свободный http://pmpu.ru/vf4/codes/cyclic (дата обращения: 15.05.18).
3. Дмитриев В.И. Прикладная теория информации: учебное пособие – М.: Высшая школа, 1989.- 319 с.
4. Черных И.В. Simulink: среда создания инженерных приложений. / Под общ. ред. В.Г. Потёмкина – М.: ДИАЛОГ-МИФИ, 2003.- 496 с.