МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ**

КАФЕДРА РАДИОТЕХНИЧЕСКИХ СИСТЕМ

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

К ЛАБОРАТОРНОЙ РАБОТЕ

«ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ

СВЕРТОЧНЫХ КОДОВ»

Для студентов по дисциплинам: радиоэлектронные системы передачи информации и РЭЗИ

Минск 2014

**1 ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ**

Целью работы является изучение принципов построения, технической реализации и оценки корректирующих способностей сверточных кодов, применяемых в системах передачи цифровой информации для повышения достоверности воспроизводимого сообщения.

**2 ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ**

Одним из эффективных способов борьбы с ошибками, возникающими при передаче сообщений по каналам с шумами, является применение методов помехоустойчивого кодирования [4]. К числу основных задач помехоустойчивого кодирования относятся: отыскание кодов, наилучшим образом исправляющих ошибки требуемого вида, нахождение методов кодирования и декодирования, способов их реализации. Коды, исправляющие ошибки, часто называют корректирующими. В настоящее время чаще всего применяются двоичные равномерные корректирующие коды, обладающие хорошими корректирующими свойствами и сравнительно простой реализацией [1,2, 3,4].

Двоичные равномерные коды делятся на блочные и сверточные (цепные, непрерывные). При использовании блочных кодов цифровая информация передаётся ввиде отдельных кодовых комбинаций равной длины, состоящих из двух различающихся (как правило) частей: информационной и проверочной. Кодирование и декодирование каждой комбинации (блока) осуществляется независимо друг от друга. Наиболее известны среди блочных кодов - циклические коды (коды Хемминга, коды Боуза - Чаудхури и др.)

В сверточных кодах формирование проверочных символов осуществляется по рекуррентным правилам и они перемежаются с информационными по всей длине кодовой последовательности. В настоящее время сверточные коды широко - используются в различных системах связи и передачи данных благодаря простоте реализации схем кодеров и декодеров по сравнению с блочными кодами той же корректирующей способности.

Развитие теории сверточных кодов [1] происходило в трёх направлениях в соответствии с тремя важнейшими методами декодирования кодов: метода декодирования по максимуму правдоподобия, метода последовательного декодирования и метода порогового декодирования. Метод декодирования по максимуму правдоподобия (алгоритм Витерби) теоретически наиболее эффективен, чем два других, однако сложность устройств, необходимых для его реализации, возрастает экспоненциально с ростом длины кода.

Метод последовательного декодирования, приближающийся по своим характеристикам к методу декодирования по максимуму правдоподобия, требует для декодирования одного символа тем меньше операций, чем меньше уровень шума в канале, в отличие от постоянного количества операций при декодировании по максимуму правдоподобия. Это свойство последовательного декодирования позволяет упростить требующиеся для его реализации устройства.

Основной недостаток последовательного метода декодирования состоит в явлении переполнения буфера, которое возникает при сильном шуме. Принятые, но ещё не декодированные символы накапливаются в буферной памяти декодера. В тех случаях, когда ожидающая обработки последовательность символов переполняет буферную память, декодер перестаёт работать правильно. Вероятность переполнения буфера часто оказывается гораздо больше, чем вероятность ошибочного декодирования и именно она определяет возможности последовательного декодирования.

Метод порогового декодирования, аналогичный методу мажоритарного декодирования, состоит в том, что при приёме вычисляются синдромы и эти синдромы или последовательности, получение в результате линейного преобразования синдромов, подаются на входы порогового элемента. Символы с выхода порогового элемента осуществляют исправление ошибок. Методы порогового декодирования имеют простые реализующие их устройства и поэтому находят широкое применение на практике.

В данной лабораторной работе изучаются сверточные коды, допускающие пороговое декодирование.

**2.1 Сверточные коды, исправляющие случайные ошибки**

При построении свёрточных кодов информационная последовательность подвергается обработке без предварительного разбиения на независимые блоки, в отличие от блочных кодов.

Проверочные символы получаются в результате проведения линейных операций над определёнными информационными символами.

Сверточные коды, как и циклические, описываются с помощью полиномов и матриц [1,2].

Изучение сверточного кодирования и декодирования рассмотрим на примере систематического кода со скоростью передачи R = k/n = 1/2.

В рассматриваемом случае кодер содержит один вход и два выхода (см. рисунок 1).

На вход кодера поступает информационная последовательность А(х), а с выхода снимаются две последовательности, одна из которых является повторением входной, а вторая - проверочной последовательностью С(х). Далее с помощью ключа "К” путём

чередования информационных символов и проверочных формируется сверточный код.

Проверочная последовательность С(х) находится путём умножения входной последовательности

А(х) = a0(1)+a1(1)x+ а2(1)х2+...

на генераторный (порождающий) полином

G(x) = g0(1)+g1x+ g2x2+...grxr

следующим образом:

С(х) = А(х) G(x) = Со(2)+ c1(2)x + С2(2)х2+ ...

Если степень полинома G(x) равна r, то любой фиксированный информационный символ a1 может оказывать влияние на выходную последовательность С(х) в течение не более чем r +1 тактов. За это время с выхода кодера будет считано: т = n(r + 1) = 2(r + 1) символов. Величину т называют кодовым ограничителем кода, а n - количество последовательностей на выходе кодера.

Кодирование, т.е. нахождение последовательности С(х) осуществляется с помощью линейных фильтров без обратных связей (см. рисунок 2). При этом значения проверочных символов определяется выражением

С1 = g0Aj + g1Aj-1 + ... + gr Аj-r .

На рисунке 3 показана реализация кодирующего устройства сверточного кода со скоростью R=l/2 и кодовым ограничением m = 2(5 +1) = 12, задаваемого генераторным полиномом

G(x) = 1 + х3 + х4 + х5 .

и исправляющего любые две случайные ошибки в пределах кодового ограничения. Сверточный код является систематическим и имеет следующий проверочный треугольник

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

[H∆,Ie]= .

где *Н -* проверочная матрица; *1е -* единичная матрица.

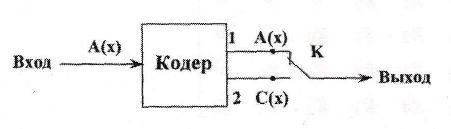


Рисунок 1 –Схема формирования сверточного кода

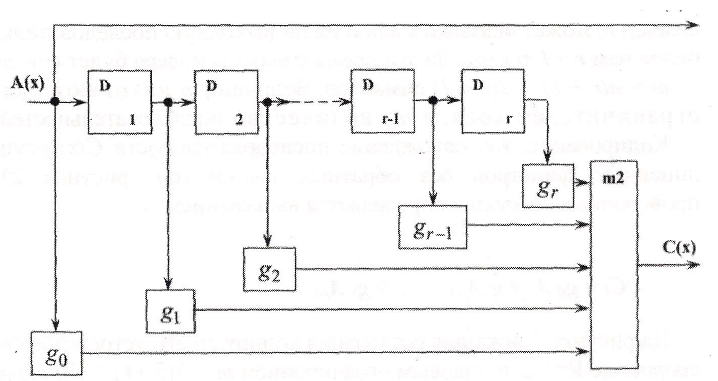


Рисунок 2 – Схема кодирования

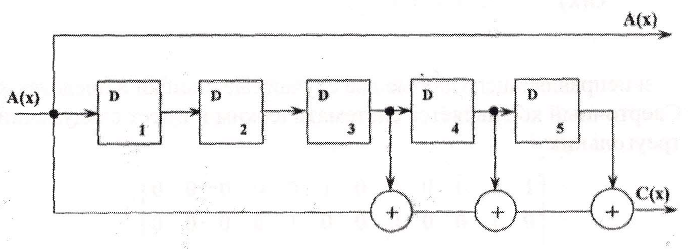
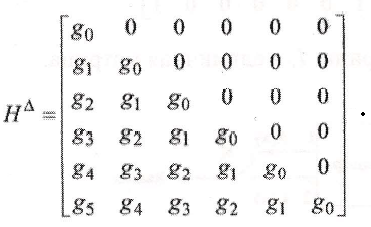


Рисунок 3 – Схема формирования кодовой последовательности

Проверочная матрица описывается коэффициентами порождаемого полинома

****

Рассмотрим теперь задачу декодирования сверт очных кодов. Вследствие помех принятые последовательности отличаются от переданных, т. е.

А\*(х) = А(х) + Е1 (х),

С\*(х) = С(х)+Е2(х),

где

E1(x) = e0(1)+e1(1)x+e2(1)x2+ … ,

Е2(х) = e0(2)+e1(2)x+e2(2)x2+ …

шумовые последовательности, наложенные на информационную и проверочную последовательности.

Если принятую информационную последовательность А(х) вновь умножить на неизменный генераторный полином G(X) и сложить по модулю 2 с принятой проверочной последовательностью С(х), то получим корректирующую последовательность - синдром:

S(x) = А\*(х) ∙ G(x) + С\*(х) = А(х) ∙ G(x) + Е(1)(х) ∙ G(x) + С(х) + Е(2)(x) = = Е(1) (х) ∙ G(x) + Е(1)(х)

Отсюда видно, что значения синдрома

S(x) = s1 + s1x + s2x2 + ...

определяются исключительно шумами.

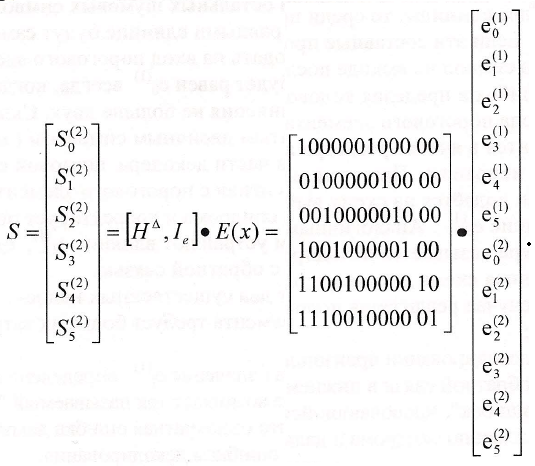
Схема декодера для сверточного кода, порождаемого полиномом

G(x) = 1 + х3 + х4 + х5 представлена на рисунке 4.

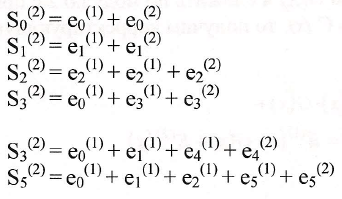
Декодер по принятой информационной последовательности сначала вычисляет точно так же, как и при кодировании, проверочные символы. Для вычисления этих прове­рочных символов декодер содержит *r -* разрядный регистр сдвига. Далее полученная с помощью этого кодера проверочная последовательность складывается покомпонентно по модулю 2 с принятой проверочной последовательностью, и в результате получается синдром

ST = (s0(2), s1(2), s2(2),…, s5(2))

Символы этого синдрома подаются в регистр сдвига, показанный на рисунке 4 в нижней части декодера. Синдром *S* и шумовые символы *е'и U=* 0,....,5, i = 1,2 связаны между собой следующим матричным соотношением:

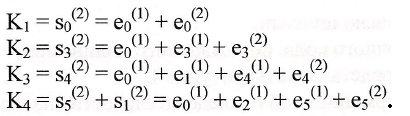


или, то же самое



.

С помощью линейного преобразования синдрома *S* находятся величины K1, К2, К3, К4 : (1)



Линейное преобразование, с помощью которого по синдрому *S* находятся величина К1 , называется правилом ортогонализации, а получаемое с его помощью величины К1 называются составными проверкам и. Составные проверки обладают следующим свойством: шумовой символ е0(1) , воздействующий на информационный символ в момент 0, входит во все составные проверки *K*1 ,( i = 1,2,3,4 ) и в то же время любой другой шумовой символ входит не более чем в одну из этих проверок. Такая совокупность составных проверок {K1} называется ортогональной относительно e0(1).

Если e0(1) = *1* и остальные шумовые символы равны нулю, то составные проверки, задаваемые формулами (1), равны 1, а именно K1 = К2 = К3 = К4 = 1. В случае когда один из шумовых символов, отличных от e0(1) , равен 1, то среди проверок K1 по крайней мере три будут равны 1. Если e0(1) = 0 и среди остальных шумовых символов имеется не более чем две единицы, то среди проверок равными единице будут самое большое две проверки. Если эти составные проверки подать на вход порогового элемента с порогом 2,5 то символ на выходе последнего будет равен e0(1) всегда, когда число случайных ошибок в пределах кодового ограничения не больше двух. Складывая символ с выхода порогового элемента с принятым двоичным символом (a0(1) + e0(1)), который хранится в ячейке регистра в верхней части декодера, шумовой символ e0(1) всегда можно исправить. При этом выходной сигнал с порогового элемента по цепи обратной связи подаётся на схему вычисления синдрома и корректирует последний, устраняя влияние e0(1) . Аналогичным образом устраняют влияние e1(1) , е2(1) и т. д. Такая процедура называется декодированием с обратной связью.

Рассмотренная схема декодирования имеет два существенных недос-. татка:

* в общем случае реализация порогового элемента требует больших затрат оборудования;
* если при декодировании произошла ошибка (значение е1(1) определено неверно), то из-за наличия обратной связи в нижнем регистре возникает так называемый "эффект размножения ошибок", заключающийся в том, что однократная ошибка декодирования приводит к искажению синдрома и дальнейшим ошибкам декодирования.

Для устранения первого недостатка были построены сверточные коды, допускающие мажоритарное декодирование. Порождающий полином G(x) этих кодов выбирается таким, чтобы получить систему разделённых относительно символа e0(1) проверочных соотношении как у вышеприведённого примера с G(x) *=* 1 + х3 + х4 + х5.

Помимо введённого ранее понятия кодового ограничения m для сверточных кодов существует понятие эффективного кодового ограничения m*\*,* которое определяется как число различных шумовых символов, входящих в систему проверочных соотношений. Имеет место неравенство m\* < m*.* Для кодов с разделенными проверками при скорости *R = 1/2* получаем m =i/2 ∙(i+ 1)+1*.* Смысл понятия эффективного кодового ограничения ясен из утверждения: если количество возникших ошибок на длине *т* разрядов не превосходит [i/2], то символ е0(1) может быть правильно декодирован с помощью мажоритарного элемента. Здесь значение *i* эквивалентно понятию реализуемого кодового расстояния 𝛿(𝛿 ≤ *d,* где *d -* минимальное кодовое расстояние).

Схема на рисунке 4 позволяет исправить две случайные ошибки на длине m*\** *=*11 при замене порогового элемента на мажоритарный элемент.

Использование мажоритарного декодирования позволяет упростить реализацию декодера, однако в этой схеме возможен эффект размножения ошибок. Для исключения этого эффекта предложены три следующих метода:

1.После передачи n0 информационных символов в течение следующих т тактов кодируются нули. При этом после декодирования n(n0+τ) символов декодер возвращается в исходное нулевое состояние и начинается процесс декодирования для следующей группы символов. Таким образом, размножение ошибок возможно только на длине n(n0+τ) символов.

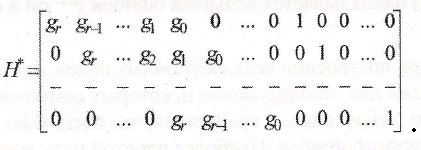
Скорость передачи в этом случае уменьшается и равна

R = n0k / n(n0+τ) .

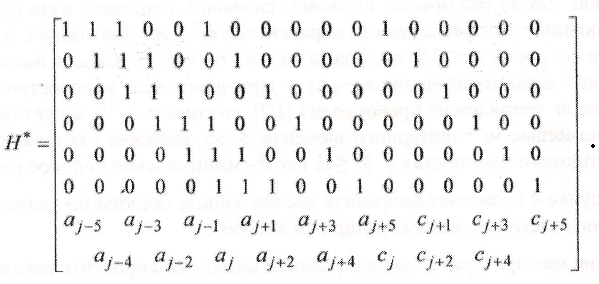
2.В процессе декодирования непрерывно контролируется количество исправляемых ошибок и, если на каком-либо интервале времени окажется, что это количество превышает исправляющую способность кода, то либо при декодировании произошла ошибка, либо из-за возрастания интенсивности шумов в канапе нельзя обеспечить требуемую достоверность приёма. В этом случае производится повторная передача с того момента, когда было зафиксировано отмеченное явление.

3.Переход к бессиндромному мажоритарному декодированию, т. е. к методу декодирования, когда значения информационных символов определяется непосредственно по принятым символам без вычисления синдрома.

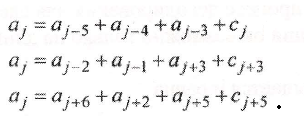
При бессиндромном мажоритарном декодировании размножение ошибок будет исключено, если в декодере будет отсутствовать обратная связь, введённая для коррекции его состояния. Для этого необходимо найти систему уравнений, позволяющих вычислить значение информационного символа а1 по принятой совокупности информационных и проверочных символов. Эти уравнения находятся из проверочной матрицы Н\*, несколько отличающейся от вышеприведённой отсутствием коррекции содержимого декодера, т. е.



Для рассматриваемого примера эта матрица будет иметь следующий вид:



Из первой, четвертой и шестой строк данной матрицы следуют уравнения, правые части которых не содержат ошибок членов:



Добавляя к этим уравнениям тривиальное аj = аj, полним систему из четырех независимых уравнений.

На рисунке 5 представлена схема для бессиндромного декодирования рассматриваемого кода. Задержка при декодировании в этой схеме составляет пять тактов, отсчитываемых с момента прихода первых символов (a0,c0). При бессиндромном декодировании без размножения ошибок декодер содержит r дополнительных ячеек регистра и, кроме того, длина последовательности, на которой реализуется расстояние δ = i, равно:

m = 3 (r+1) -1 = 3r + 2

вместо 2(r +1) символов.

**2.2 Сверочные коды, исправляющие пакеты ошибок**

Пакетом ошибок длины b называется последовательность из b ошибочных символов, первый и последний из которых отличны от нуля. Если в канале связи при передаче информации с помощью свёрточных кодов возникают пакеты ошибок длинны b, то на каждую из последовательностей (имеется в виду к информационных и (n-k) проверочных последовательностей) накладывается вспышка ошибок диной в среднем b/n. Основ­ная идея [1,4], использованная при построении всех свёрточных кодов, исправляющих пакеты ошибок, состоит в применении для декодирования некоторых символов, распределенных во времени так, что только один, в крайнем случае несколько из них, может быть затронут одиночным пакетом ошибок. Наиболее простой путь достижения такой длительности состоит в перемежении информационных и соответствующих им проверочных символов по времени передачи. Поскольку предполагается, что за пачкой ошибок следует определенное количество искаженных символов, то одновременное искажение информационных и зависящих от них проверочных символов считается невозможным. Если же длина пачки ошибок превысит значение, на которое рассчитывался код, или между пачками ошибок не будет необходимого числа неискаженных символов, то свёрточный код не обеспечить исправления ошибок, (в настоящее время известно достаточно много разнообразных свёрточных кодов, исправляющих пакеты ошибок [1, 2, 4]. Рассмотрим исправление пакетов ошибок кодом Хальгельбергера, имеющим скорость передачи R=1/2. Этот код способен исправить пакет ошибок длины < Ъ при условии, что две соседние пачки разделены между собой защитным промежутком ≥3b + 1. Это означает, что между последним символом данного пакета и первым символом последующего пакета должно быть не менее 3b + 1 безошибочных символов. Наибольшая длина пачки b кратна n, т.е. код со скоростью R = 1/2 может исправлять пакеты с длиной 2, 4, 6.

Конкретно рассмотрим код, задаваемый порождающим полиномом

G(x)=x2+x4 .

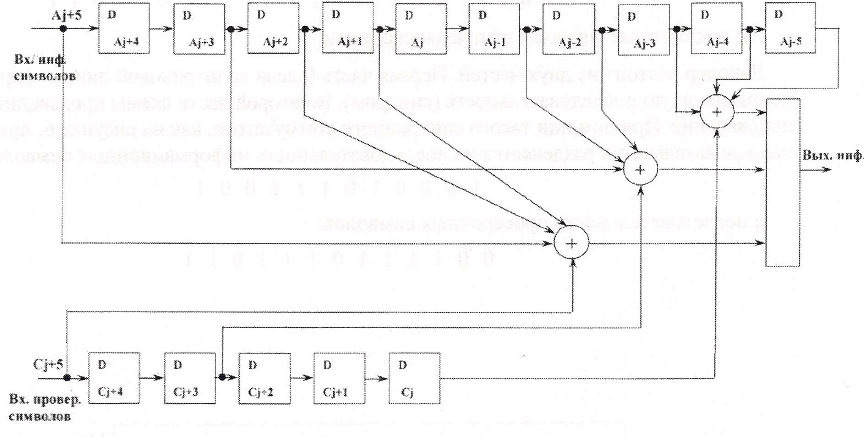


Рисунок 5 – Схема электрическая структурная бессиндромного декодирования

Этот код обладает способностью исправлять пакеты ошибок длины ≤4(b ≤ 4), имеющих защитный интервал ≥3b + l=13. Схема кодирующего устройства представлена на рисунке 6. С выхода коммутатора К поочередно следуют информационные и проверочные символы, формирующие выходную последовательность.

Пусть на вход кодирующего устройства подается последовательность информационных символов:

100100111001.

Последовательность проверочных символов на выходе сумматора по модулю 2 соответственно:

0010110111011.

Выходная последовательность, передаваемая в канал передачи информации, имеет вид:

10001100101101111010011.

Пусть в канале связи произошло искажение 7-го, 8-го и 9-го символов. Тогда символы последовательности на входе декодера, схема которого представлена на рисунке 7, примут значения:

100001(011)101101111010011.

где искаженные символы отмечены скобками.

Декодер состоит из двух частей. Первая часть (слева до штриховой линии) формирует исправляющую последовательность (синдром). Во второй части схемы производится само исправление. При помощи такого синхронного коммутатора, как на рисунке 6, принятая последовательность разделяется на последовательность информационных символов:

100010111001.

и последовательность проверочных символов:

0011110111011.

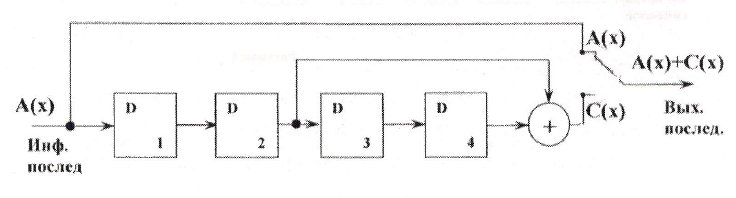


Рисунок 6 – Схема электрическая структурная формирования выходной последовательности

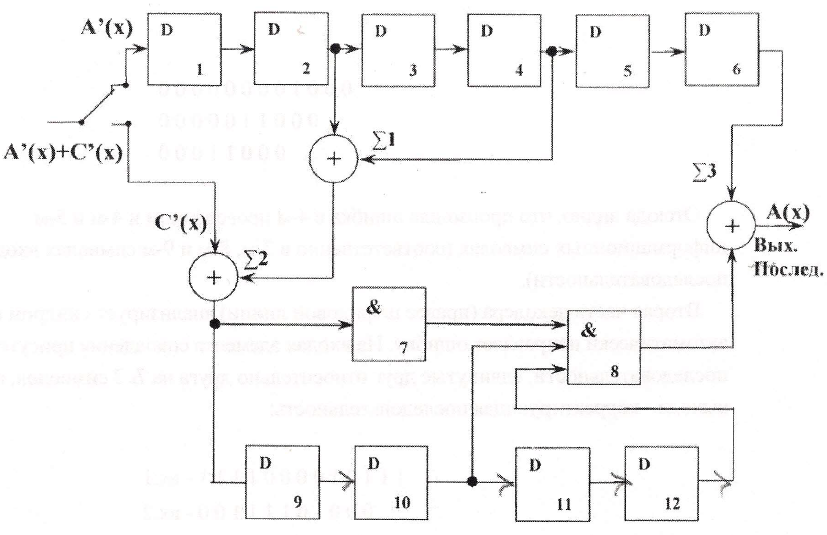


Рисунок 7 – Схема электрическая структурная декодирования сверточного кода

Первая из последовательностей поступает на регистр, в точности, повторяющей кодер (см. рисунок 6). Если ошибок нет, то сформированная этим регистром и сумматором по модулю 2 последовательность в точности совпадает со второй проверочной последовательностью и их суммирование дает нулевую последовательность. Если же имеются ошибки, как в нашем случае, то вычисляется исправляющая последовательность **(Синдром):**

0001011110000.

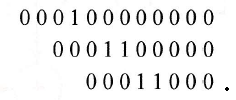
Известно, что каждому пакету ошибок соответствует свой синдром. Определим его структуру. Будем считать, что произошел наихудший случай - исказились b символов. Следовательно, будет поражено b/2 информационных и b/2 проверочных символов. До поступления первого ошибочного символа на входе декодера регистр содержит безошибочные символы, поэтому в первых b/2 тактах в синдроме возникают единицы за счет ошибок в проверочных символах. На этом пакет ошибок заканчивается, и в дальнейшем на сумматор формирователя синдрома будут поступать лишь безошибочные символы. За следующие b тактов единицы формируются в синдроме сначала за счет поступления ошибочных информационных символов из первого полурегистра, а затем - из второго. Таким образом, синдром содержит:

1.Единицы на местах ошибок в проверочных символах.

2.Со сдвигом на b/2 символов - единицы на местах ошибок в информационных символах.

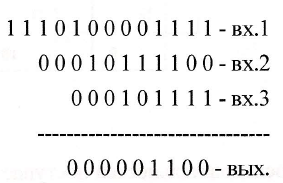
3.Еще со сдвигом на b/2 символов повторяется комбинация, полученная в предыдущем случае.

Следовательно, для рассматриваемого примера, синдром может быть представлен тремя составляющими:



Отсюда видно, что произошла ошибка в 4-м проверочном и 4-м и 5-м информационных символах (соответственно в 7-м, 8-м и 9-м символах входной последовательности).

Вторая часть декодера (правее штриховой линии) анализирует синдром и автоматически исправляет ошибку. На входах элемента совпадения присутствуют три последовательности, сдвинутые друг относительно друга на b/2 символов, а на его выходе - корректирующая последовательность:



Поскольку корректирующая последовательность формируется через Зb/2 тактов, а информационные символы задерживаются только на b тактов, то возникает необходимость в дополнительной задержке информаци­онных символов на b/2 тактов (b/2 = 2).

Исправленная информационная последовательность имеет вид:



На пути информационных символов в декодере имеются Зb/2 запоминающих ячеек. Это соответствует 36 символам во входной последовательности. Очевидно, чтобы вывести все ошибочные символы, требуется защитный промежуток из Зb + 1 безошибочных символов. Чтобы не проводилось исправление в случае появления ошибочных символов, в этот период предусмотрен элемент И-НЕ.

**3 ПРЕДВАРИТЕЛЬНОЕ ЗАДАНИЕ**

Предварительное задание выполняется при подготовке к работе и состоит в следующем:

1.Изучить принципы построения и методы реализации свёрточных кодов.

2.Построить кодер, синдромный и бессиндромный декодеры кода, задаваемого полиномом G{x} =1 + х6 + х7 + х9 + х10 + х11

3.Найти проверочную матрицу, систему составных проверок и систему уравнений при бессиндромном декодировании. Сколько ошибок может исправить этот код? Чему равно кодовое ограничение кода и эффективное кодовое ограничение?

4.Построить кодер и декодер свёрточного кода, задаваемого полиномом

G(x) = х3 + х6. Какой пакет ошибок код может исправить? Чему равен ошибок код может исправить? Чему равен защитный промежуток?

**4 ОПИСАНИЕ ЛАБОРАТОРНОЙ УСТАНОВКИ**

Установка включает два кодирующих устройства, задаваемых полиномами С(х) = 1 + х3 + х4 + х5 и G(x) = х2 + х4 , три декодирующих устройства (синдромный декодер с мажоритарным элементом, бессиндромный декодер без обратной связи и декодер для исправления пакетов ошибок), наборное поле из 12 переключателей для задания информационной последовательности и наборного поля из 24 переключателей (12 переключателей используются для создания последовательности помех, налагаемых на информационные символы, остальные 12 - на проверочные символы), пятиразрядного двоичного счётчика числа тактируемых импульсов; сумматора по модулю 2, имитирующего канал передачи информации; устройства вывода информации, представляющего собой 12-разрядный сдвиговый регистр; индикаторов (индикатор светится при наличии логической "1" в последовательности символов); галетного переключателя.

Положение 1 галетного переключателя соответствует кодированию и декодированию по синдрому свёрточного кода G(x) = 1 +х3 +х4 + х5, исправляющего одиночные ошибки.

Положение 2 соответствует кодированию и мажоритарному бессиндромному декодированию того же кода.

Положение 3 соответствует кодированию и декодированию свёрточного кода G(x) = х2 + х4, исправляющего пакеты ошибок длиной b≤4.

Ввод информации в кодирующее и наложение помех в канале передачи информации осуществляется путём нажатия на кнопку "счёт".

**5 ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОГО ЗАДАНИЯ**

5.1.Подключать установку только к сети. Ознакомиться со схемой лабораторной установки, представленной на передней панели.

5.2.Включить тумблер «Сеть». При этом загорятся некоторые световые индикаторы.

5.3.Нажать кнопку «сброс», обнуляющую все регистры. Большинство лампочек должны погаснуть.

5.4.Переключатель П1; поставить в положение «1».

5.5.Набрать заданную преподавателем последовательность информационных символов на наборном поле «Входная последовательность». Переключатели наборного поля «Помехи» установить в положение «О».

5.6.Подать набранную последовательность на устройство кодирования идекодирования путем нажатия кнопки тактовых импульсов «Счётчик тактов». При этом в каждом такте необходимо записать состояние контрольных точек (в местах установки индикаторов). Кодовая последовательность, передаваемая в канал передачи информации, формируется в каждом такте путем записи состояния индикатора.

Сравнить полученные результаты с информацией на входе кодера. Определить количество тактов, когда последовательность информационных символов заполнят выходной регистр «Выходная информация».

5.7.Нажать кнопку «Сброс».

5.8.Проверка исправности всех кодеров и декодеров. Набрать код передаваемой информации на регистре «Входная информация». Нажать на кнопку «Сброс». Нажимая на кнопку «Счетчик тактов»; ввести эту информацию в регистр «Выходная информация».

Устанавливать поочередно переключатель П1 в соответствующее

5.9.Проверка прохождения и исправления ошибок декодерами в информационных (нечетных) символах. Количество ошибок устанавливается нечетными тумблерами регистра помехи (а1,2а1,3а1,4а1, 5а1, и т.д.). При каждой новой проверке увеличивается количество ошибочных символов в канале связи. Определить количество информационных символов, исправляемых данным декодером.

5.10.Проверка прохождения и исправления ошибок декодерами в проверочных символах. Количество ошибок устанавливается четными тумблерами регистра помехи (с1,2с1,3с1,4с1, 5с1, и т.д.). При каждой новой проверке увеличивается количество проверочных символов. Определить количество проверочных символов, исправляемых каждым декодером.

5.11.Проверка прохождения и исправления декодерами пакета ошибок «в»,состоящего из определенного количества символов (а1с1,2а1с1,2а12с1,3а12с1,3а13с1 и т.д.).

Определить количество исправляемых символов в пакете ошибок "b"каждым декодером.

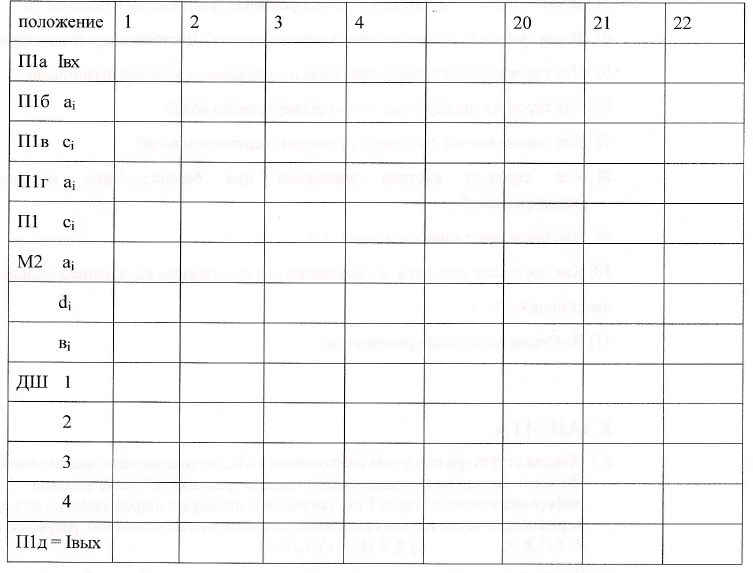
5.12.Прохождение и исправление информации декодерами при постоянном защитном промежутке (Зb + 1) = 13 и переменной длине "b" пакета ошибок, равного 3, 4, 5 и т.д. Определить количество исправляемых символов в пакете ошибок "Ъ" каждым декодером.

5.13.Проверить прохождение и исправление информации при постоянной длине пакета ошибок "b = 4" и переменном защитном промежутке Зb + 1, равном соответственно 12, 13, 14, 15 и т.д.

**5.14.Примечание:**

1.По пунктам 5.6-5.12 проверить 3 декодера (1, 2, 3) и составить не менее одной таблицы прохождения информации для указанного преподавателем типа декодера.

Таблица 1 – Прохождение информации



2.После каждого выполненного пункта сделать и записать выводы о результатах эксперимента, указав количество исправляемых символов каждым декодером. 5.14. Поставить тумблер «Сеть» в положение «Выкл.».

**6 СОДЕРЖАНИЕ ОТЧЁТА**

1.Цель работы.

2.Решение предварительного задания.

3.Результаты экспериментальной проверки,

4.Анализ результатов и выводы.

**7 КОНТРОЛЬНЫЕ ВОПРОСЫ**

1.Какова структура сверточных кодов?

2.Как задать и построить сверточный код?

3.В чем сущность синдромного декодирования? Достоинства и недостатки.

4.В чем сущность бессиндромного декодирования? Достоинства и недостатки.

5.Что такое кодовое ограничение кода и эффективное кодовое ограничение?

6.Как строится проверочная матрица сверточного кода?

7.Как определяются составные проверки сверточного кода?

8.Как строится система уравнений при бессиндромном мажоритарном декодировании?

9.Что такое пакет ошибок длиной b?

10.Как составлен алгоритм кодирований для сверточного кода, исправляющего пакет ошибок?

11.Что такое защитный промежуток?

**8 ЗАЩИТА**

8.1.Какова структура синдрома сверточного кода, исправляющего пакеты ошибок? Показать сигнал на выходе каждого каскада (рисунки 6,7) если входная информация соответствует 1 вх, указанная в примере а информация на выходе передающей части канала связи имеет ошибки в нижеуказанных разрядах:

а) 6,7, 8;

б)7,8,9;

в) 8,9,10;

г) 9,10,11

Результаты для вариантов а, б, в, г представить в виде таблицы 2.

8.2. Какие изменения необходимо внести в кодер и декодер, изучаемые в лабораторной работе ( рисунок 6,7 ) для обеспечения коррекции пакета ошибок, состоящего из следующего числа символов:

а) 6, б) 8, в) 10, г) 12 .

Таблица 2

