СЕВЕРО-КАВКАЗСКИЙ ФИЛИАЛ
ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
СВЯЗИ И ИНФОРМАТИКИ»

КАФЕДРА «ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА»

В.А. Данилов

Ю.В. Жабинский

А.С. Усик

ОБЩАЯ ТЕОРИЯ СВЯЗИ

Задание и методические указания

по выполнению курсовой работы № 1 по теме:

«Кодек помехоустойчивых кодов»

Ростов-на-Дону

2018 г.

УДК 621.396
ББК 32.88
Д 18
Данилов В.А., Жабинский Ю.В., Усик А.С. ОБЩАЯ ТЕОРИЯ СВЯЗИ. Переработанные задание и методические указания по выполнению курсовой работы № 1 по теме: «Кодек помехоустойчивых кодов». Ростов-на-Дону: Северо-Кавказский филиал МТУСИ, 2018. – 36 с.

Приводятся задание и методические указания для выполнения курсовой работы № 1 «Кодек помехоустойчивых кодов» по дисциплине «Общая теория связи» для студентов 3-х курсов направления подготовки 11.03.02 - «Инфокоммуникационные технологии и системы связи».

Составители:

профессор кафедры ИВТ СКФ МТУСИ д.т.н. Данилов В.А.,
доцент кафедры ИВТ СКФ МТУСИ к.т.н. Жабинский Ю.В.
зав. лабораторией ОТЦ и ЛС СКФ МТУСИ Усик А.С.

Рецензент: заведующий кафедрой «Инфокоммуникационные технологии и системы связи» Юхнов В.И., к.т.н., доцент

Утверждено на заседании кафедры ИВТ
(протокол № 1 - от 28.08.2017г.)

И з д а т е л ь с т в о С К Ф М Т У С И

Сдано в набор 15.02.18 Изд. № 256 Подписано в печать 12.03.18. Зак. 270.
Печ. Листов 2.3. Учетно-изд. л. 1,8. Печать оперативная. Тир. 41 экз.
Отпечатано в Полиграфическом центре СКФ МТУСИ, Серафимовича, 62.

СОДЕРЖАНИЕ

АННОТАЦИЯ

В описании приведены разработанные задание и методические указания по выполнению курсовой работы № 1 «Кодек помехоустойчивых кодов» по дисциплине «Общая теория связи» (ОТС).

Задание и методические указания предназначены для студентов 3-х курсов очной и заочной форм обучения направления подготовки 11.03.02 - «Инфокоммуникационные технологии и системы связи».

Основная цель курсовой работы - ознакомление с алгоритмами помехоустойчивого кодирования и декодирования информации, структурными схемами кодера и декодера, практическое применение различных видов кодирования. Выполнение данной курсовой работы способствует углублению, систематизации и закреплению теоретических знаний по дисциплине ОТС.

В методических указаниях подробно рассмотрены вопросы:

- кодирования информации при цифровой обработке сигналов;
- использования методов циклического, сверточного и блочного помехоустойчивого кодирования;
- построения алгоритмов и структурных схем кодера и декодера.

1. Основные задачи цифровой обработки сигналов

Цифровые системы передачи (ЦСП) информации широко применяются в отрасли связи. Важное место в них занимают системы цифровой обработки сигналов. Они включают в себя блок аналого-цифрового преобразования (БАЦП) и блок цифро-аналогового преобразования (БЦАП). В БАЦП осуществляется прием, дискретизация, квантование аналоговых сигналов поступающих на вход ЦСП, преобразование их в цифровые сигналы, кодирование их и модуляция. На выходе БАЦП формируется цифровой кодированный сигнал с фиксированным количеством разрядов. В БЦАП осуществляется прием кодированных цифровых сигналов, их декодирование, интерполяция и преобразование в аналоговый сигнал, идентичный аналоговому сигналу, поступившему на вход БАЦП.

Кодирование аналоговых сигналов осуществляет устройство, которое называется кодером. Декодирование цифровых сигналов осуществляет декодер. Поскольку в любой системе передачи сигналов нужно передавать и принимать информацию, то кодер и декодер объединены в одно устройство, которое называется кодеком.

При разработке структурных схем кодера и декодера используются методы помехоустойчивого кодирования и декодирования. В процессе разработки выбираются алгоритмы кодирования и декодирования, формируются совокупности разрешенных комбинаций помехоустойчивого кода для всех возможных информационных комбинаций.

2. Задание на курсовую работу

Задание на курсовую работу предполагает выполнение следующих этапов:

1. составить структурную схему цифровой системы связи и указать назначение каждого блока;

2. записать свою фамилию, имя, отчество и выбрать из записи первые 10 букв. Каждая буква - это импульс-отсчет начала очередного процесса. Амплитуда отсчета равна порядковому номеру буквы. Закодировать эти отсчеты двоичным кодом $(m=2, n=5)$, нарисовать отсчеты и соответствующий им сигнал импульсно-кодовой модуляции (ИКМ);

3. рассчитать дисперсию шума квантования, выбрав $U_{\max}$ в вольтах из графика квантованного сигнала;

4. определить вероятность дибитов (00,01,10,11) в двоичной последовательности сигнала ИКМ, полученной в пункте 2. Рассчитать энтропию источника с полученной вероятностью дибитов. Закодировать дибиты двоичным кодом с префиксными свойствами и определить его энтропию, избыточность и среднюю длину кодовой комбинации;

5. осуществить помехоустойчивое кодирование двоичных информационных комбинаций, используя для этого код, указанный в таблице 1. Согласно заданию на курсовую работу студены должны выбрать один из трех вариантов помехоустойчивых кодов :

- циклический;
- блочный;
- сверточный.

Все три варианта рассматриваются в данном пособии. Необходимо описать алгоритмы кодирования и декодирования, записать все разрешенные кодовые комбинации на выходе кодера для всех информационных комбинаций на входе, зарисовать структурные схемы кодера и декодера.

3. Состав обобщенной структурной схемы цифровой системы связи

Обобщенная структурная схема цифровой системы связи показана на рисунке 1:

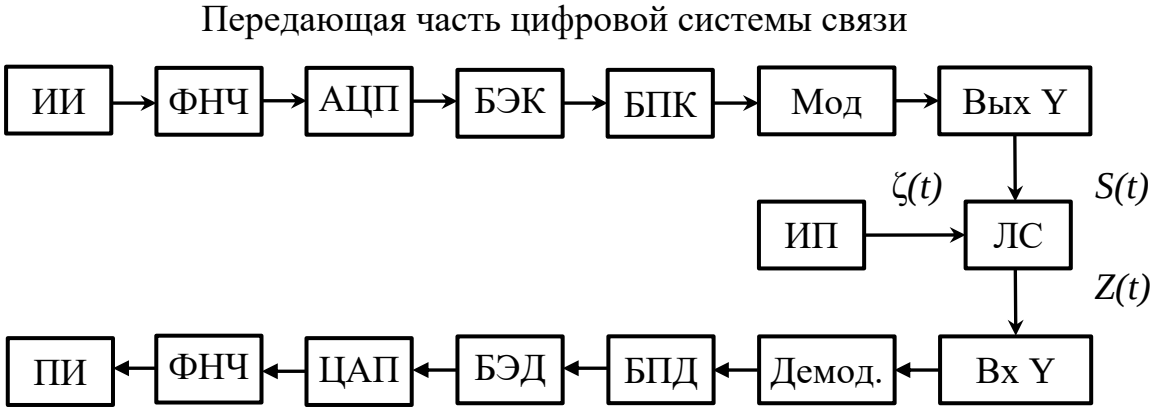

Рисунок 1. Обобщенная структурная схема системы передачи информации

Расшифровка обозначений на рисунке 1:

ИИ - источник информации;

ФНЧ - фильтр низких частот с верхней частотой пропускания F_B;

АЦП - аналогово-цифровой преобразователь;

БЭК - блок эффективного кодирования;

БПК - блок помехоустойчивого кодирования;

Мод. - модулятор несущей;

Вых. У - выходное устройство (выходной фильтр и усилитель передатчика);

ИП - источник помех $\zeta(t)$;

ЛС - линия (канал) связи;

Вх. У - входное устройство (входной фильтр и усилитель приемника);

Демод. - демодулятор входного сигнала;

БПД – блок помехоустойчивого декодирования;

БЭД - блок эффективного декодирования;

ЦАП - цифро-аналоговый преобразователь;

ПИ - приемник информации.

4. Примитивное кодирование исходной текстовой информации

Записываем свою фамилию, имя и отчество (например, Данилов Виктор Александрович). По этим данным формируем исходное текстовое сообщение, состоящее из первых 10 букв. Выбранные буквы, импульс-отсчет каждой буквы и двоичный кодовый номер для всех букв приведены в таблице 1. Также в этой таблице приведены все необходимые данные для помехоустойчивого кодирования циклическим кодом.

К ним относятся:

а) $m(x)$ – полином сообщения;

б) $m(x) \cdot x^4$ – произведение двух полиномов (промежуточный полином);

в) $R(x)$ – остаток от деления полинома $m(x) \cdot x^4$ на порождающий полином ($P(x) = x^4 + x + 1$).

Выбор данного полинома обоснован в соответствующем разделе пособия.

Таблица 1. Исходные данные для помехоустойчивого кодирования

№ п/п	Буква	Амплитуда отсчета	Двоичный код буквы	Полином сообщения - $m(x)$	Произведение $m(x) \cdot x^4$	Остаток $R(x)$
1	Д	4	00100	x^2	x^6	$x^3 + x^2$
2	А	0	00000	0	0	0
3	Н	13	01101	$x^3 + x^2 + 1$	$x^7 + x^6 + x^4$	x^2
4	И	8	01000	x^3	x^7	$x^3 + x + 1$
5	Л	11	01011	$x^3 + x + 1$	$x^7 + x^5 + x^4$	$x^3 + x^2 + x$
6	О	14	01110	$x^3 + x^2 + x$	$x^7 + x^6 + x^5$	1
7	В	2	00010	x	x^5	$x^2 + x$
8	В	2	00010	x	x^5	$x^2 + x$
9	И	8	01000	x^3	x^7	$x^3 + x + 1$
10	К	10	01010	$x^3 + x$	$x^7 + x^5$	$x^3 + x^2 + 1$

По данным таблицы 1 нарисуем квантованный сигнал $x_{кв}(t)$ (рисунок 2) и соответствующий сигнал импульсно-кодовой модуляции $x_{икм}(t)$ (рисунок 3):

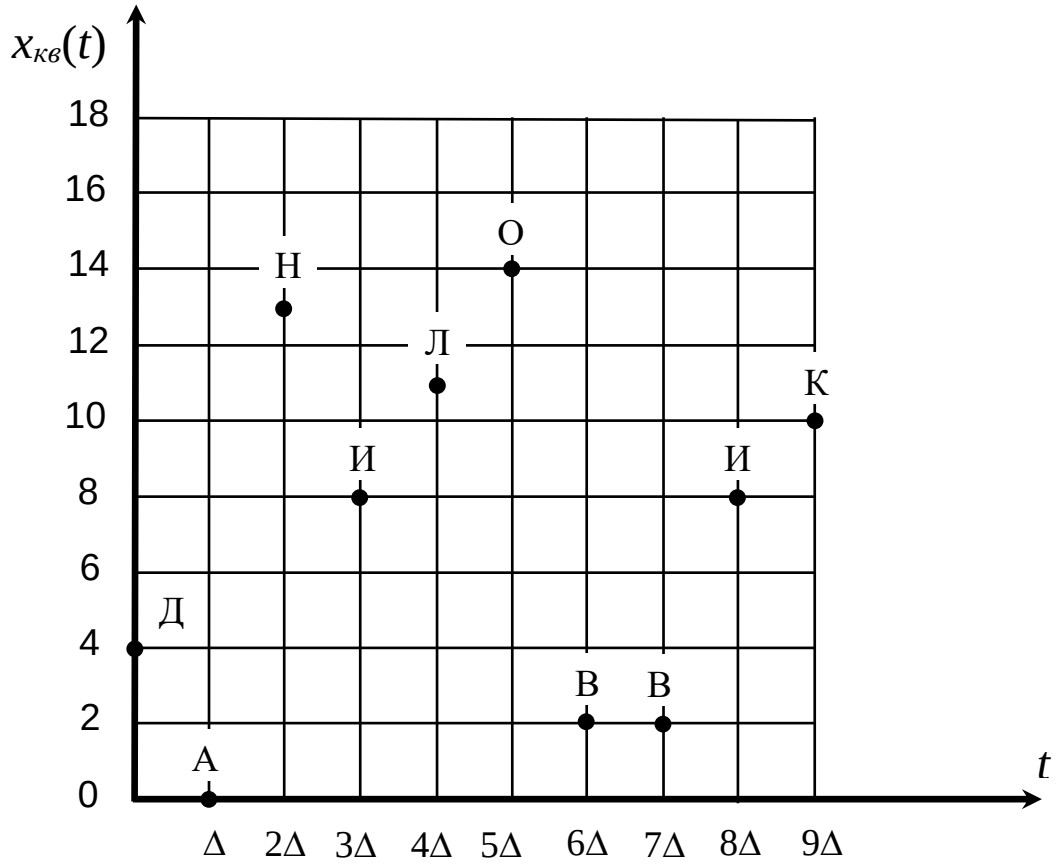

Рисунок 2. График квантованного сигнала

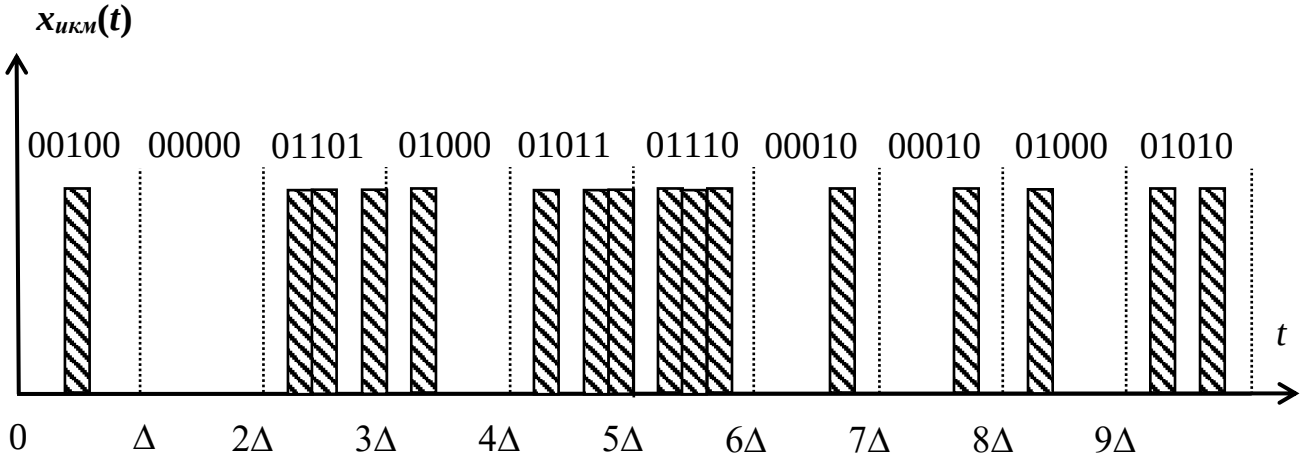

Рисунок 3. График сигнала импульсно-кодовой модуляции

5. Расчет дисперсии шума квантования и энтропии исходной текстовой информации

Дисперсия шума квантования рассчитывается по формуле:

$$\sigma^2 = \frac{\Delta^2}{12}, \tag{1}$$

где Δ (шаг квантования) находится по формуле :

$$\Delta = \frac{U_{\max} - U_{\min}}{N-1}, \tag{2}$$

где $N = 14$ - количество уровней квантования исходного сигнала;

значения U_{max}, U_{min} находятся по графику квантования сигнала :

$$\left.\begin{array}{l} U_{\max} = 14 \ [B] \\ U_{\min} = 0 \ [B] \end{array}\right\} \tag{3}$$

В результате расчетов, находим :

$$\Delta = \frac{14-0}{14-1} = 1{,}0769 \tag{4}$$

$$\sigma^2 = \frac{\Delta^2}{12} = \frac{1.0796^2}{12} 0{,}0966 \tag{5}$$

Для расчета энтропии исходного текстового сообщения, определим вероятность нулей и вероятность единиц в битовой последовательности. С этой целью определим число нулей (N_0 = 34) и число единиц (N_1 = 16) в данной текстовой информации. Далее находим соответствующие вероятности:

$$p_0 = \frac{N_0}{50} = \frac{34}{50} = 0{,}68 \tag{6}$$

$$p_1 = \frac{N_1}{50} = \frac{16}{50} = 0{,}32 \tag{7}$$

Величина энтропии находится по формуле [1]:

$$H(A) = -p(0)\, log_2\, p(0) - p(1)\, log_2\, p(1) =$$

$$= 0{,}3781 + 0{,}5261 = 0{,}9042 \left[\frac{\text{Бит}}{\text{символ}}\right] \quad (8)$$

Избыточность данного сообщения составляет величину:

$$r(u) = H_{max} - H(A) = 1 - 0{,}9042 = 0{,}0958\,. \quad (9)$$

6. Эффективное кодирование информационной последовательности с помощью разбивки битовой последовательности на совокупность дибитов

Эффективное кодирование – это такое кодирование, при котором устраняется избыточность. Избыточность возникает в том случае, когда энтропия $H(A)$ исходного сообщения меньше максимально возможной величины ($H(A) < H_{max}$). В результате такого кодирования скорость передачи информации увеличивается до максимально-возможной величины.

Существуют два метода эффективного кодирования:

1.Метод вспомогательных группировок (метод Хафмена);

2.Метод равновероятных группировок (метод Шеннона – Фано).

В обоих случаях при кодировании учитывается статистика входных символов. А именно – наиболее вероятные символы кодируются более короткими кодовыми комбинациями, а символы с маленькой вероятностью – кодируются более длинными кодовыми комбинациями. В результате такого кодирования формируется код с префиксным свойством. Префиксный код – это такой код при котором наиболее короткие кодовые комбинации не входят в начало более длинных кодовых комбинаций. Поэтому при кодировании и декодировании можно обойтись без дополнительного символа для разделения отдельных элементов.

Рассмотрим оба метода кодирования на примере разбивки исходной информационной последовательности на совокупность дибитов (00,01,10,11).

С этой целью исходный поток битов разбиваем на дибиты и находим количество дибитов:

Д	А	Н	И	Л	О	В	В	И	К	(10)
00100	00000	01101	01000	01011	01110	00010	00010	01000	01010	

По данным (10) находим количество дибитов, вероятность каждого величину функции $(-p\,log_2\,p)$ соответствующего дибита. Найденные величины заносим в таблицу 2.

Таблица 2. – Вероятности и величины функции каждого дибита

№ П/п	Дибит	Количество (шт.)	Вероятность дибита	$(-p\,log_2\,p)$
1	$S_0(00)$	10	$p_0 = \frac{10}{25} = 0{,}40$	0,5288
2	$S_1(01)$	5	$p_1 = \frac{5}{25} = 0{,}20$	0,4644
3	$S_2(10)$	9	$p_2 = \frac{9}{25} = 0{,}36$	0,5306
4	$S_3(11)$	1	$P_3 = \frac{1}{25} = 0{,}04$	0,1858
Итого		25	1,0	1,7096

По данным таблицы 2 находим энтропию источника с полученной вероятностью дибитов:

$$H_1(S) = -\sum_{k=1}^{4} p_k\,log_2\,p_k = 1{,}7096 \qquad \text{бит/дибит} \qquad (11)$$

6.1. Кодирование дибитов кодом Хафмена

В данном случае порядок кодирования состоит в следующем. Выписываем все дибиты в порядке убывания их вероятностей. Затем два последних дибита объединяем и присваиваем вероятность объединенному элементу в виде суммы вероятностей исходных элементов. Такую операцию производим до тех пор, пока сумма вероятностей оставшихся элементов не будет равна единице. Таким образом, формируется граф кода Хафмена (рисунок 4):

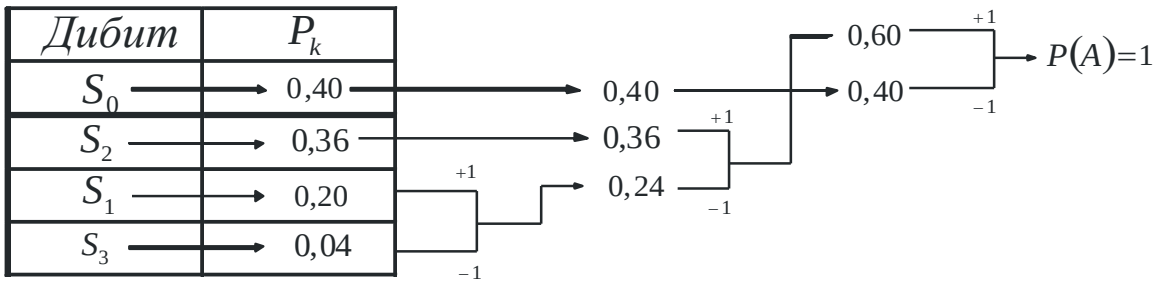

Рисунок 4. Граф кода Хафмена.

По данным графа на рисунок 4 определяем код промежуточный, код префиксный и количество дибитов, приходящихся на сто штук среднестатистических сообщений. Эти данные заносим в таблицу 3:

Таблица 3. Коды дибитов

Дибит	Код промежуточный	Код префиксный	Количество дибитов (шт.)
S_3	+1-1-1	100	4
S_1	+1-1+1	101	20
S_2	+1+1	11	36
S_0	-1	0	40

Находим энтропию кода с префиксными свойствами. С этой целью по данным таблицы 3 определяем количество единиц и количество нулей приходящихся на сто штук среднестатистических сообщений:

количество единиц:

$$N_1 = 1 \cdot 4 + 2 \cdot 20 + 2 \cdot 36 = 116 \quad (12)$$

количество нулей:

$$N_0 = 2 \cdot 4 + 1 \cdot 20 + 1 \cdot 40 = 68 \quad (13)$$

общее число символов:

$$N_{общ} = N_1 + N_0 = 184 \quad (14)$$

Далее находим вероятности:

$$\left.\begin{aligned} p_1 &= \frac{N_1}{N_{\text{общ}}} = \frac{116}{184} = 0{,}63 \\ p_0 &= \frac{N_0}{N_{\text{общ}}} = \frac{68}{184} = 0{,}37 \end{aligned}\right\} \quad (15)$$

Энтропия источника с префиксным кодом Хафмена:

$$H_2(S) = -p_1\, log_2\, p_1\ -\ p_0\, log_2\, p_0 = 0.4202 + 0.5306 = 0.9508 \quad (16)$$

Избыточность источника с префиксным кодом Хафмена составляет:

$$r_n = 1 - H_2(S) = 0{,}0492 \quad (17)$$

Средняя длина кодовой комбинации для источника с кодом Хафмена составляет:

$$\bar{n}_{\text{ср}} = \frac{N_{\text{общ}}}{100} = \frac{184}{100} = 1{,}84 \quad (18)$$

По теореме Шеннона для канала без помех должно выполняться неравенство:

$$\bar{n}_{\text{ср}} \geq H_1(S) = 1{,}7096 \quad (19)$$

6.2. Кодирование дибитов кодом Шеннона – Фано

В данном случае процесс кодирования заключается в следующем. Всю совокупность дибитов следует выписать в порядке убывания их вероятностей. Далее эти дибиты делим на две части таким образом, чтобы сумма вероятностей в обеих частях была бы примерно одинаковой. Первой части присваиваем кодовый символ “0”, второй части – символ “1”. Процесс деления на части повторяем до тех пор, пока в каждой из частей останется по одному элементу. Данные по кодированию сводим в таблицу 4:

Таблица 4. Кодирование дибитов кодом Шеннона - Фано

Дибит	Вероятность дибита	Этапы деления на подгруппы			Код дибита	Количество дибитов (шт.)
		1 этап	2 этап	3 этап		
S_0	0,40	} 0			0	40
S_2	0,36	} 1	} 0		10	36
S_1	0,20		} 1	0	110	20
S_3	0,04			1	111	4

В результате кодирования сформирован код с префиксными свойствами. Определим энтропию этого кода. С этой целью по данным таблицы 4 определяем количество нулей и количество единиц, приходящихся на сто штук среднестатистических сообщений:

количество единиц:

$$N_1 = 1 \cdot 36 + 2 \cdot 20 + 3 \cdot 4 = 88 \quad (20)$$

количество нулей:

$$N_0 = 1 \cdot 40 + 1 \cdot 36 + 1 \cdot 20 = 96 \quad (21)$$

общее количество символов:

$$N_{\text{общ}} = 88 + 96 = 184 \quad (22)$$

Далее находим вероятности:

$$\left.\begin{aligned} p_1 &= \frac{N_1}{N_{\text{общ}}} = \frac{88}{184} = 0{,}48 \\ p_0 &= \frac{N_0}{N_{\text{общ}}} = \frac{96}{184} = 0{,}52 \end{aligned}\right\} \quad (23)$$

энтропию источника с префиксным кодом Шеннона – Фано :

$$H_2(S) = -p_1 \log_2 p_1 - p_0 \log_2 p_0 = 0{,}5083 + 0{,}4904 = 0{,}9987 \quad (24)$$

Избыточность источника с кодом Шеннона – Фано составляет:

$$r_n = 1 - H_2(S) = 0{,}0013 \quad (25)$$

Сравнивая данную величину со значением (17) заключаем, что кодирование по методу Шеннона-Фано обеспечивает существенно меньшую

избыточность. Следовательно, код Шеннона-Фано более эффективен по сравнению с кодом Хафмена.

7. Помехоустойчивое кодирование двоичных информационных комбинаций

Помехоустойчивое кодирование – это такое преобразование исходной информации, при котором обеспечивается заданная верность передачи в условиях действия помех. В общем случае помехоустойчивое кодирование заключается во введении в исходную последовательность дополнительной избыточности в виде проверочных (контрольных) разрядов. Число контрольных разрядов определяется заданной эффективностью кода. Эффективность кода – это способность кода обнаруживать и исправлять ошибки в принятой информационной комбинации.

В курсовой работе разрабатывается помехоустойчивый код, способный обнаруживать и исправлять одиночные ошибки (Δ=1). Для кода, гарантированно исправляющего одну ошибку, величина минимального кодового расстояния определяется как [1]:

$$d_{min} = 2\Delta + 1 = 3 \tag{26}$$

В этом случае при заданном числе информационных разрядов (k = 5) находим число контрольных символов кода по формуле:

$$r = \left[log_2\left((k+1) + [log_2(k+1)]_1\right)\right]_1, \tag{27}$$

где знаки $[\dots]_1$ – округление до ближайшего целого числа в большую сторону. Итого , при $k = 5$ получаем :

$$\left.\begin{aligned} log_2(k+1) = log_2 6 = 2{,}58 \Rightarrow 3, \\ log_2((k+1)+3) = log_2 9 = 3{,}17 \Rightarrow 4. \end{aligned}\right\} \tag{28}$$

Следовательно, величина $r = 4$ и таким образом необходимо разработать систематический линейный код с параметрами

$$(n, k) = (9{,}5), \tag{29}$$

где $n -$ общее число разрядов кода.

7.1. Помехоустойчивое кодирование циклическим кодом

Помехоустойчивый код называется циклическим, если циклическая перестановка символов в разрешенной кодовой комбинации снова дает разрешенную кодовую комбинацию. Процедура формирования циклического кода основана на отображении кодовых комбинаций в форме соответствующих многочленов.

Основой такого описания является представление кодовой комбинации длины n в виде многочлена $(n-1)-$ степени, содержащего фиктивную переменную x . Запишем, для примера, полином сообщения $m(x)$ для буквы “Н”:

$$\text{"Н"} \Rightarrow 01101 \Rightarrow m(x) = 0 \cdot x^4 + 1 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0. \quad (30)$$

Опуская нулевые члены получаем:

$$m(x) = x^3 + x^2 + 1. \quad (31)$$

Процедура циклического кодирования заключается в следующем. Исходную информационную комбинацию в форме $m(x)$ умножаем на так называемый порождающий полином $P(x)$, в результате чего формируется код с некоторым числом дополнительных элементов. Для того , чтобы число дополнительных элементов составляло величину $r = 4$ максимальная степень порождающего полинома $P(x)$ должна равняться 4. В приемном тракте канала связи принятую комбинацию следует разделить на полином $P(x)$. Если деление выполнено без остатка, то в принятой последовательности ошибок нет. Наличие остатка говорит о том, что в принятой кодовой комбинации содержится ошибка, следовательно, по виду остатка можно определить номер искаженного разряда.

Описанная процедура циклического кодирования обладает существенным недостатком. Этот недостаток заключается в том , что результирующая последовательность циклического кода является не разделимой. Кодовая комбинация считается не разделимой , если в этой

последовательности информационные и контрольные разряды идут вперемешку. Декодировать такой код – крайне затруднительно.

Для того чтобы получить разделимый код процедура кодирования должна быть иной. В этом случае информационный полином $m(x)$ в форме (31) следует сначала умножить на полином $x^r = x^4$, в результате получаем промежуточный полином в виде:

$$m(x) \cdot x^4 = x^7 + x^6 + x^4 \ . \qquad (32)$$

Отметим здесь, что умножение полинома $m(x)$ на x^4 для двоичной комбинации означает простое приписывание справа четырех нулей: (01101|0000). Далее следует полученное произведение (32) разделить на порождающий полином $P(x)$. В результате деления формируется остаток $R(x)$, который используется затем для формирования помехоустойчивой комбинации циклического кода $F(x)$ в форме суммы:

$$F(x) = m(x) \cdot x^4 + R(x) \qquad (33)$$

В результате такого кодирования получаем систематический циклический код вида (9,5) с разделимыми элементами. Данный полином (33) делится без остатка на порождающий полином.

Из вышеописанной процедуры кодирования можно заключить, что фундаментальную роль здесь играет выбор порождающего полинома.

Сформулируем основные **требования к порождающему полиному:**

Как уже отмечалось, максимальная степень полинома $P(x)$ должна быть равна $r = 4$. Это одно из основных требований. Кроме того порождающий полином $P(x)$ должен быть не приводимым (или простым). Как и простые числа, простой многочлен $P(x)$ – это такой многочлен, который делится сам на себя и на единицу, и не делится ни на какой другой многочлен меньшей степени. Именно поэтому простой многочлен может быть сформирован с помощью простого числа, записанного в двоичной форме.

Последнее требование к порождающему полиному состоит в том , что число не нулевых элементов $P(x)$ должно быть d_{min}= 3. Существуют таблицы порождающих полиномов [2]. Из этой таблицы выбираем порождающий полином вида:

$$P(x) = x^4 + x + 1\,. \qquad (34)$$

Двоичная форма данного полинома записывается как (10011), что соответствует простому числу 19.

Рассмотрим более подробно процедуру формирования систематического циклического кода для буквы "Н" заданного сообщения. С этой целью разделим произведение вида (32) на порождающий полином $P(x)$:

$$\begin{array}{r|l} x^7 + x^6 + x^4 & x^4 + x + 1 \\ \hline x^7 + x^4 + x^3 & x^3 + x^2 \\ \hline x^6 + x^3 & \\ x^6 + x^3 + x^2 & \\ \hline \end{array}$$

$$x^2 = R(x) \Rightarrow (0100) \qquad (35)$$

В результате деления вида (35) получился остаток $R(x)$, представимый в виде полинома и в двоичном виде следующим образом:

$$R(x) = x^2 \Rightarrow (0100). \qquad (36)$$

Полученный остаток $R(x)$ используем для формирования разрешенной помехоустойчивой комбинации циклического кода. Для рассматриваемого примера разрешенная комбинация для буквы "Н" записывается следующим образом:

$$F(x) = m(x) \cdot x^4 + R(x) = x^7 + x^6 + x^4 + x^2 \qquad (37)$$

Полиноминальное представление (37) преобразуем в двоичную форму:

$$F(x) \Rightarrow (01101 \mid 0100) \qquad (38)$$

Информ Контр

В результате всех операций, получили циклический код вида (9,5), задаваемый полиномом $F(x)$ в форме (37). Данный полином делится без остатка на порождающий полином $P(x)$.

Кодирование остальных букв исходного текстового сообщения производится в той же самой последовательности. Результаты кодирования всех букв сообщения приведены в таблице 4 в форме соответствующих полиномов:

Таблица 4. Результата кодирования всех букв сообщения

N п/п	Буква	Полином сообщения $m(x)$	Остаток деления $R(x)$	Полином циклического кода (разделимый) $F(x)$
1	Д	x^2	$x^3 + x^2$	$x^6 + x^3 + x^2$
2	А	0	0	0+0
3	Н	$x^3 + x^2 + 1$	x^2	$x^7 + x^6 + x^4 + x^2$
4	И	x^3	$x^3 + x + 1$	$x^7 + x^3 + x + 1$
5	Л	$x^3 + x + 1$	$x^3 + x^2 + x$	$x^7 + x^5 + x^4 + x^3 + x^2 + x$
6	О	$x^3 + x^2 + x$	1	$x^7 + x^6 + x^5 + 1$
7	В	x	$x^2 + x$	$x^5 + x^2 + x$
8	В	x	$x^2 + x$	$x^5 + x^2 + x$
9	И	x^3	$x^3 + x + 1$	$x^7 + x^3 + x + 1$
10	К	$x^3 + x$	$x^3 + x^2 + 1$	$x^7 + x^5 + x^3 + x^2 + 1$

Рассмотренная методика кодирования позволяет сформировать циклический код с разделимыми элементами. Это означает, что любая последовательность двоичных элементов кода, записанная на основе полинома $F(x)$, приводит к кодовой комбинации, представляемой в форме (38). Первые пять элементов этого кода представляют соответствующую информационную комбинацию, а последние четыре элемента определяют проверочную или контрольную комбинацию. Разделимый циклический код существенно упрощает аппаратурную реализацию кодирования и поэтому данный метод получил наибольшее практическое применение.

Формирование циклического кода с неразделимыми элементами заключается в следующем. Для каждой информационной последовательности записываем соответствующий полином $m(x)$ сообщения. Этот полином умножается на порождающий полином $P(x)$ с приведением подобных членов по правилу сложения чисел по модулю два. Таким образом, формируется полином циклического кода $D(x)$, в котором информационные и контрольные символы не разделены, что приводит к усложнению процесса обработки на приемной стороне.

Приведем пример формирования неразделимого полинома $D(x)$ для буквы "H" $\Rightarrow$ 01101 сообщения с информационным полиномом вида:

$$m(x) = x^3 + x^2 + 1\,. \tag{39}$$

Умножим функцию (39) на порождающий полином $P(x)$ в форме (34), в результате умножения двух полиномов получим:

$$D(x) = m(x) \cdot P(x) = (x^3 + x^2 + 1)[x^4 + x^2 + 1] = \\ = x^7 + x^6 + x^2 + x + 1\,. \tag{40}$$

Полиномиальному вектору (40) соответствует двоичная комбинация циклического кода $(9{,}5)$:

$$D(x) \Rightarrow (011000111), \tag{41}$$

в которой информационные и контрольные разряды не разделены. Результаты циклического кодирования всех букв исходного текста методом умножения двух полиномов приведены в таблице 5 в полиномиальной форме и в соответствующей двоичной записи в виде (41):

Таблица 5. Результаты циклического кодирования

N п/п	Буква	Полином сообщения $m(x)$	Циклический код	
			Полином - $D(x)$	Битовое слово
1	Д	x^2	$x^6 + x^3 + x^2$	001001100
2	А	0	0	000000000
3	Н	$x^3 + x^2 + 1$	$x^7 + x^6 + x^2 + x + 1$	011000111
4	И	x^3	$x^7 + x^4 + x^3$	010011000
5	Л	$x^3 + x + 1$	$x^7 + x^5 + x^3 + x^2 + x + 1$	010101101
6	О	$x^3 + x^2 + x$	$x^7 + x^6 + x^5 + x^4 + x$	011110010
7	В	x	$x^5 + x^2 + x$	000100110
8	В	x	$x^5 + x^2 + x$	000100110
9	И	x^3	$x^7 + x^4 + x^3$	010011000
10	К	$x^3 + x^2 + x$	$x^7 + x^5 + x^4 + x^3 + x^2 + x$	010111110

7.2. Структурная схема кодера циклического кода

В основе структурной схемы кодера циклического кода лежит использование $(n-k)$- разрядного регистра сдвига с определенным числом обратных связей. Конкретная структура кодера определяется типом порождающего полинома $P(x)$. Рассмотрим структурную схему кодера для полинома вида:

$$P(x) = x^4 + x + 1. \quad (42)$$

Разрядность регистра равна степени полинома $(m = 4)$, а число обратных связей равно числу ненулевых членов полинома $P(x)$. Таким образом, структурная схема кодера, имеющего полную длину кода равную 9 разрядам, а количество информационных разрядов равное 5, имеет следующий вид (рисунок 5):

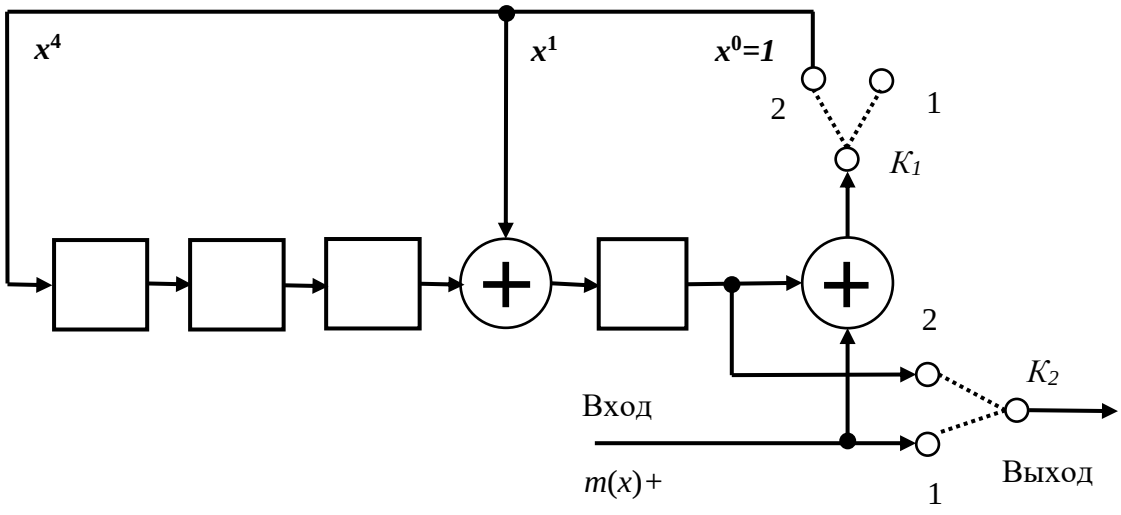

Рисунок 5. Структурная схема кодера циклического кода

Циклический кодер содержит два сумматора по модулю два $\oplus$, четыре элемента памяти и два переключателя (K_1, K_2). Схема содержит три ветви обратной связи, обозначенные: x^4, x^1 и $x^0 = 1$. Количество ветвей соответствует числу ненулевых членов порождающего полинома $P(x)$.

Работа схемы кодирования происходит в такой последовательности:

1) При первых $K = 5$ сдвигах ключ K_1 закрыт для передачи битов сообщения в $(n-k)$разрядный регистр сдвига;
2) Ключ K_2 установлен в положение 1 для передачи битов сообщения на выходной регистр в течение первых $K = 5$ сдвигов;
3) После передачи $K-го$ бита сообщения оба ключа (K_1, K_2)переводятся в положение 2. При этом ключ K_1 открывается , а ключ K_2 переходит в верхнее положение;
4) При остальных $(m = n-k)$ сдвигах происходит очищение кодирующих регистров, биты четности перемещаются на выходной регистр;
5) Общее число сдвигов равно $n = 9$, и содержимое выходного регистра представляет собой полином кодового слова (27):

$$F(x) = m(x) \cdot x^4 + R(x); \qquad (43)$$

7.3. Структурная схема декодера циклического кода

Структурная схема декодера, имеющего полную длину кода, равную 9 разрядам, а количество информационных разрядов, равное 5, приведена на рисунке 6:

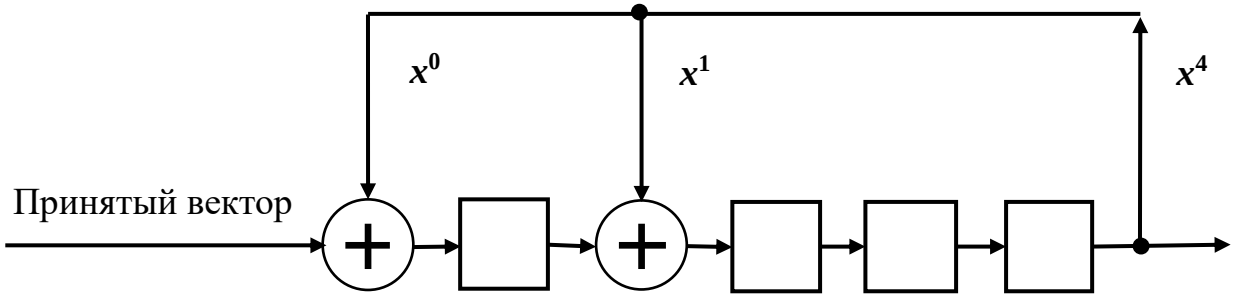

Рисунок 6. Структурная схема декодера циклического кода

Декодер циклического кода выполняет операцию деления принятой кодовой комбинации на порождающий полином $P(x)$. Если в результате деления не образуется остатка, то принятая комбинация считается разрешенной и содержащееся в ней сообщение декодируется. Если при делении образуется остаток , то принятая комбинация считается запрещенной. Наличие остатка позволяет обнаружить ошибку, а вид остатка (синдром) - определить номер искаженного разряда, т.е. исправить ошибку.

Структурная схема декодера (рисунок 6) почти аналогична схеме кодирования. Декодер также содержит два сумматора по модулю два и четыре элемента памяти. Схема декодера является зеркальной по отношению к схеме кодирования. Декодер также содержит три ветви обратной связи, обозначенные на рисунке 6 как x^4, x^1 и $x^0 = 1$.

Результатом работы декодера является формирование вектора синдрома. Если синдром нулевой, то считается, что принятый вектор является правильным кодовым словом. Если синдром отличен от нуля, значит обнаружена ошибка и принятый вектор - это искаженное кодовое

слово; данная ошибка исправляется путем прибавления к принятому вектору специального вектора ошибки. Вектор ошибки формируется специальной логической схемой анализирующей конкретную структуру синдрома.

7.4. Сверточное кодирование

Сверточный код относится к классу непрерывных кодов, у которых выходные символы - это функции входных символов и конструкции кодера. В простейшем случае сверточный кодер представляет собой линейный сдвиговый регистр с конечным числом состояний, а также сумматоры по модулю два. Будем рассматривать сверточный кодер по схеме на рисунке 7:

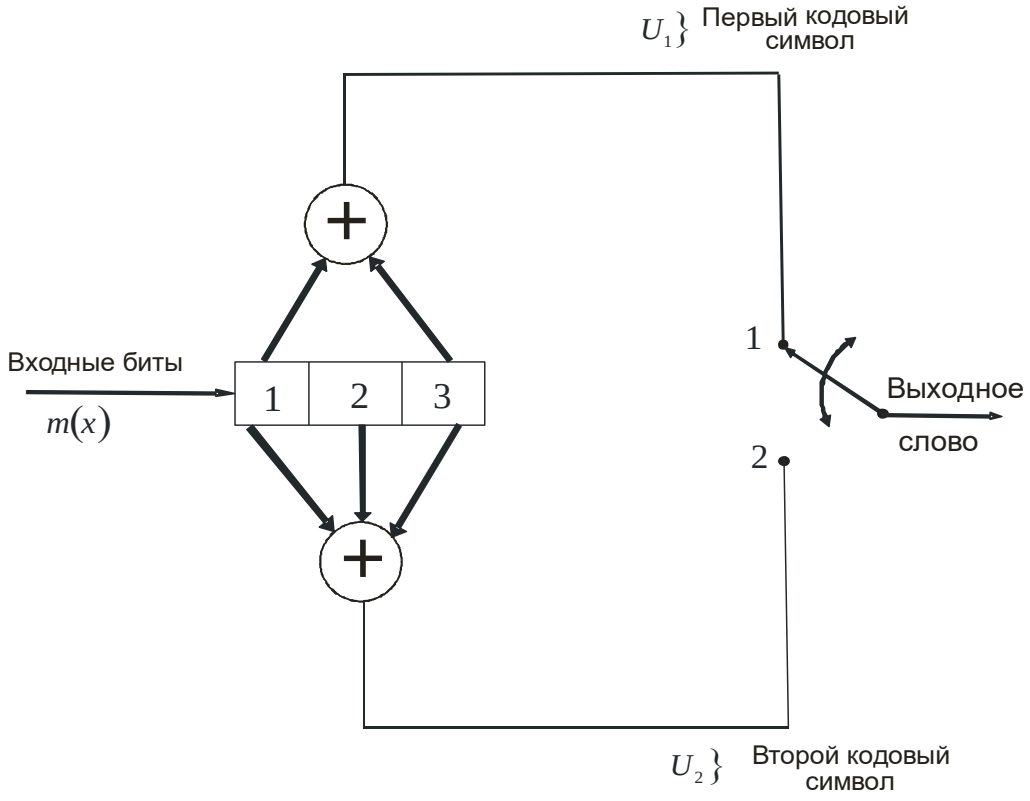

Рисунок 7. Сверточный кодер

Сверточный кодер состоит из трехразрядного регистра сдвига и двух сумматоров по модулю два. Рассмотрим последовательность

формирования сверточного кода для буквы "$H \Rightarrow 01101$" сообщения. При этом будем считать, что регистр содержит нулевые элементы. Последовательность формирования выходных кодовых символов (U_1, U_2) для соответствующего символа на входе кодера показана на рисунке 8:

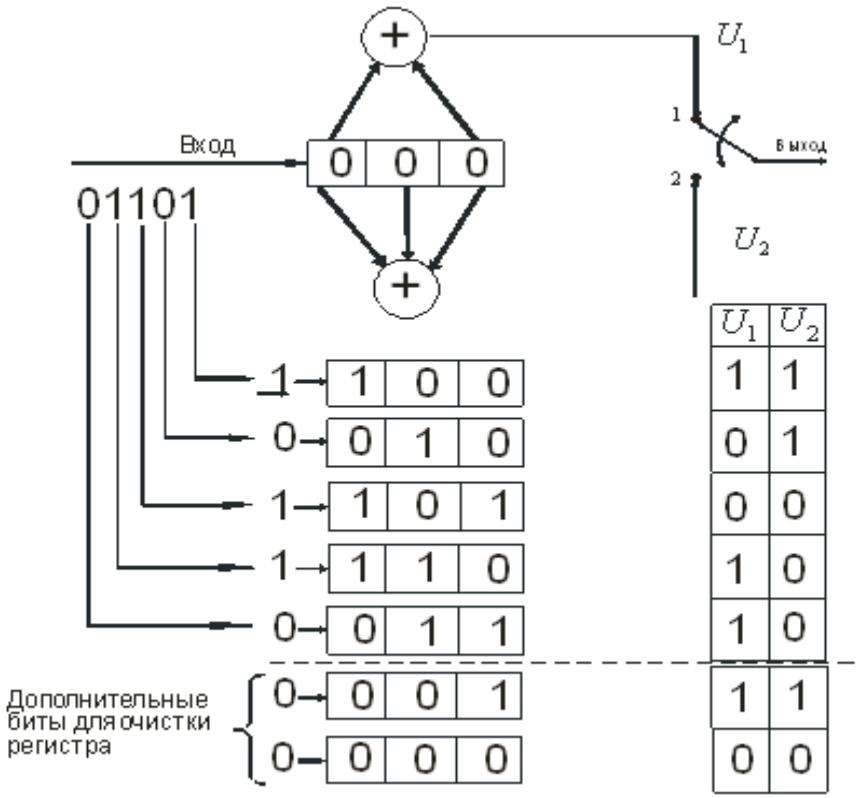

Рисунок 8. Последовательность формирования выходных кодовых символов

Таким образом, для буквы "*H*" сообщения последовательность на выходе кодера выглядит следующим образом:

$$U = \{11\ 01\ 00\ 10\ 10\ 11\ 00\} \qquad (44)$$

Рассмотрим процедуру формирования разрешенной комбинации сверточного кода с помощью порождающих полиномов. Для кодера на рисунке 7 запишем полиномиальный генератор $g_1(x)$ для верхних связей и генератор $g_2(x)$ - для нижних связей:

$$\left.\begin{array}{l} \mathrm{g}_1(x) = 1 + x^2 \\ \mathrm{g}_2(x) = 1 + x + x^2 \end{array}\right\} \quad (45)$$

Выразим вектор сообщения "*H*" в виде полинома:

$$m(x) = 1 + x + x^2 \quad (46)$$

Тогда выходящий полином $U(x)$ или выходящая последовательность U сверточного кодера для входящего сообщения $m(x)$ может быть получена следующим образом:

$$m(x) \cdot \mathrm{g}_1(x) = (1 + x^2 + x^3)(1 + x^2) = 1 + x^3 + x^4 + x^5 \quad (47)$$

$$m(x) \cdot \mathrm{g}_2(x) = (1 + x^2 + x^3)(1 + x + x^2) = 1 + x + x^5 \quad (48)$$

По данным (47) и (48) составляем таблицу соответствия для произведения полиномов:

$$\left.\begin{array}{l} m(x) \cdot \mathrm{g}_1(x) = 1 + 0 \cdot x + 0 \cdot x^2 + 1 \cdot x^3 + 1 \cdot x^4 + 1 \cdot x^5 + 0 \cdot x^6 \\ m(x) \cdot \mathrm{g}_2(x) = 1 + 1 \cdot x + 0 \cdot x^2 + 0 \cdot x^3 + 0 \cdot x^4 + 1 \cdot x^5 + 0 \cdot x^6 \end{array}\right\} \quad (49)$$

По данным (47), (48) записываем полином соответствия $U(x)$:

$$U(x) = (11) + (01)x + (00)x^2 + (10)x^3 + (10)x^4 + (11)x^5 + (00)\,x^6$$

Следовательно, сверточный код примет вид:

$$U = \{11\ 01\ 00\ 10\ 10\ 11\ 10\} \quad (50)$$

Полученная последовательность (50) совпадает с последовательностью (44). Найденная последовательность U соответствует разрешенной кодовой комбинации для буквы "H – 01101" сообщения.

Применим рассмотренную полиномиальную процедуру для определения набора битовых последовательностей сверточного кода для всех букв исходного текстового сообщения. В таблице 6 приведены исходные полиномы $m(x)$ для всех букв сообщения, произведения полиномов $m(x) \cdot$

$g_1(x)$ и $m(x) \cdot g_2(x)$, а также битовая последовательность сверточного кода для всех букв текста.

Таблица 6. Битовая последовательность сверточного кода всех букв

N п/п	Буква	Полином сообщения $m(x)$	Произведение полиномов		Битовая последовательность сверточного кода
			$m(x) \cdot g_1(x)$	$m(x) \cdot g_2(x)$	
1	Д	x^2	$x^2 + x^4$	$x^2 + x^3 + x^4$	00 00 11 01 11 00 00
2	А	0	0	0	00 00 00 00 00 00 00
3	Н	$1 + x^2 + x^3$	$1 + x^3 + x^4 + x^5$	$1 + x + x^5$	11 01 00 10 10 11 00
4	И	x^3	$x^3 + x^5$	$x^3 + x^4 + x^5$	00 00 00 11 01 11 00
5	Л	$1 + x + x^3$	$1 + x + x^2 + x^5$	$x^3 + x^4 + x^5$	11 10 10 00 01 11 00
6	О	$x + x^2 + x^3$	$x + x^2 + x^4 + x^5$	$x^3 + x^4 + x^5$	00 11 10 01 10 11 00
7	В	x	$x + x^3$	$x^3 + x^4 + x^5$	00 11 01 11 00 00 00
8	В	x	$x + x^3$	$x^3 + x^4 + x^5$	00 11 01 11 00 00 00
9	И	x^3	$x^3 + x^5$	$x^3 + x^4 + x^5$	00 00 00 11 01 11 00
10	К	$x + x^3$	$x + x^5$	$x + x^2 + x^4 + x^5$	00 11 01 00 01 11 00

7.5. Помехоустойчивое кодирование блочным кодом

Помехоустойчивый код называется блочным, если из всей совокупности разрешенных кодовых комбинаций можно выделить некоторую совокупность линейно-независимых элементов. Такую совокупность элементов можно представить в виде единичной матрицы размером $(k \times k) = (5 \times 5)$:

$$E(5 \times 5) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad . \tag{51}$$

Совокупность элементов (51) носит название информационной матрицы [1]. Для записи всей совокупности разрешенных комбинаций блочного кода очень часто используют матричный метод кодирования. В основе данного метода лежит формирование порождающей матрицы $[G]$ и матрицы $[H]$ используемой в качестве проверочной. Матрица $[G]$ используется для формирования всей совокупности кодовых комбинаций при передачи текста, а матрица $[H]$ используется при декодировании текста на приеме.

Процедура формирования порождающей матрицы заключается в следующем. Сначала формируется матрица промежуточных комбинаций. Эта матрица имеет такую структуру:

$$[E|P_0] = [E(k \times k)|P_0(k \times m)] \ , \tag{52}$$

где - $E(k \times k)$ – единичная матрица размером $(k \times k)$; $P_0(k \times m)$ – матрица размером $(k \times m)$ полностью состоящая из нулевых элементов.

Матрицу вида (52) запишем в явном виде с применением полиноминальных элементов типа $(x^8, x^7, x^6, x^5, x^4)$ на главной диагонали единичной матрицы:

$$[E|P_0] = \left(\begin{array}{ccccc|cccc} x^8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x^7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x^6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x^5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x^4 & 0 & 0 & 0 & 0 \end{array}\right) \tag{53}$$

Каждую строку данной матрицы (53) делим на порождающий полином $P(x)$ и в результате деления определяем набор соответствующих остатков. Покажем к примеру процесс деления первой строки матрицы:

$$
\begin{array}{l|l}
x^8 & x^4+x+1 \\
\hline
x^8+x^5+x^4 & \\
\hline
x^5+x^4 & x^4+x+1 \\
x^5+x^2+x & \\
\hline
x^4+x^2+x & \\
x^4+x+1 & \\
\hline
x^2+1=R_1(x)\Rightarrow(0101) &
\end{array}
\tag{54}
$$

В результате деления первой строки получился остаток вида

$$R_1 = x^2 + 1 \Rightarrow (0101). \tag{55}$$

Выполняя аналогичные операции для других строк матрицы (53), находим соответствующие им остатки:

$$
\begin{aligned}
R_2 &= x^3 + x + 1 \Rightarrow (1011),\\
R_3 &= x^3 + x^2 \Rightarrow (1100),\\
R_4 &= x^2 + x \Rightarrow (0110),\\
R_5 &= x + 1 \Rightarrow (0011);
\end{aligned}
\tag{56}
$$

С помощью найденных остатков можно сформировать порождающую матрицу, которая имеет вид:

$$
[G] = \left[\begin{array}{ccccc|cccc}
x^8 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1\\
0 & x^7 & 0 & 0 & 0 & 1 & 0 & 1 & 1\\
0 & 0 & x^6 & 0 & 0 & 1 & 1 & 0 & 0\\
0 & 0 & 0 & x^5 & 0 & 0 & 1 & 1 & 0\\
0 & 0 & 0 & 0 & x^4 & 0 & 0 & 1 & 1
\end{array}\right] = [E|\mathrm{P}]
\tag{57}
$$

Матрица – дополнение $[P]$ в выражении (57) имеет вид:

$$
[P] = \begin{bmatrix} R_1(x)\\ R_2(x)\\ R_3(x)\\ R_4(x)\\ R_5(x) \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 1 & 0 & 1\\ 1 & 0 & 1 & 1\\ 1 & 1 & 0 & 0\\ 0 & 1 & 1 & 0\\ 0 & 0 & 1 & 1 \end{bmatrix}
\tag{58}
$$

Порождающая матрица (57) определяет первые пять разрешенных комбинаций блочного кода. Остальные комбинации можно получить путем различных сочетаний строк порождающей матрицы. Общее число сочетаний определяется как:

$$C_5^0 + C_5^1 + C_5^2 + C_5^3 + C_5^4 + C_5^5 = 1 + 5 + 10 + 10 + 5 + 1 = 2^5 = 32. \qquad (59)$$

Этого количества вполне достаточно для кодирования всех букв русского алфавита.

С помощью порождающей матрицы (57) можно сформировать блочный код для всех букв исходного сообщения. С этой целью исходную пятиэлементную кодовую комбинацию следует умножить на матрицу $[G]$ по правилам умножения двух матриц.

Покажем использование матричного метода для метода для буквы "*Н*" исходного текста. Имеем в данном случае:

$$(01101) \times [G] = (01101) \times [E|P] = (01101|0100) \qquad (60)$$

Полученной битовой комбинации блочного кода (9,5) соответствует разделимый полином вида:

$$F(x) = x^7 + x^6 + x^4 + x^2 \qquad (61)$$

Данный полином соответствует ранее найденной функции (37). Совершенно аналогично можно получить всю совокупность разрешенных кодовых комбинаций для всех букв исходного сообщения.

Матричный метод удобно использовать для определения ошибок в принятой кодовой комбинации. С этой целью формируется так называемая проверочная матрица [*H*], которая имеет такую структуру:

$$[H] = [P^T|E], \qquad (62)$$

где - $[P^T]$ - транспонированная матрица - дополнение вида (58);

$[E]$ – единичная матрица размером $m-$ строк и $m-$ столбцов (m$\times m$).

С помощью (58) находим:

$$[P^T] = \begin{bmatrix} 01100 \\ 10110 \\ 01011 \\ 11001 \end{bmatrix}, \tag{63}$$

Подставляя (63) в выражение (62), получим

$$[H] = \left[\begin{array}{c|c} 01100 & 1000 \\ 10110 & 0100 \\ 01011 & 0010 \\ 11001 & 0001 \end{array}\right]. \tag{64}$$

С помощью проверочной матрицы $[H]$ определяем синдром принятой кодовой комбинации. С этой целью принятую кодовую комбинацию следует умножить на транспонированную проверочную матрицу. Транспонированная проверочная матрица $[H^T]$ имеет такую структуру:

$$[H^T] = \left[\frac{P}{E}\right] = \left[\begin{array}{c} 0\ 1\ 0\ 1 \\ 1\ 0\ 1\ 1 \\ 1\ 1\ 0\ 0 \\ 0\ 1\ 1\ 0 \\ \underline{0\ 0\ 1\ 1} \\ 1\ 0\ 0\ 0 \\ 0\ 1\ 0\ 0 \\ 0\ 0\ 1\ 0 \\ 0\ 0\ 0\ 1 \end{array}\right] \tag{65}$$

Для всех кодовых комбинаций, не содержащих ошибки, и сформированных с помощью порождающей матрицы $[G]$ из (57), произведение матрицы $[G] = [E|P]$ на матрицу $[H^T]$ имеет вид :

$$[G] \cdot [H^T] = [E|P] \cdot \left[\frac{P}{E}\right] = [P + P] = [0] \tag{66}$$

Определив номер искаженного разряда необходимо его исправить путем суммирования содержимого разряда с единицей.

Нулевая матрица (66) означает, что ошибок в принятой кодовой комбинации нет. Наличие ошибки в принятой кодовой комбинации приводит к произведению матриц (66) отличному от нуля. Таким образом, формируется синдром (или опознаватель) ошибок., Совокупность синдромов определяется матрицей (65). Можно показать, что одиночная

ошибка в первом разряде блочного кода дает синдром (0 001) матрицы (65), а ошибке в последнем разряде – соответствует синдром (0 001).

Определив номер искаженного разряда необходимо его исправить путем суммирования содержимого разряда с единицей.

8. Список использованных источников

1. Данилов В.А. Общая теория связи. Конспект лекций, часть 1. СКФ МТУСИ, 2012.

2. Акулиничев Ю.П. Теория электрической связи. СПб, «Лань», 2010

3. Родин А.Р., Тюнин Н.А. Цифровая обработка сигналов. – М.: Солон - Пресс, 2013

4. Передача дискретных сообщений. Под ред. В.П. Шувалова. – М.: Радио и связь, 1992

5. Гольденберг Л.М., Матюшкин Б.Д., Поляк М.Н. Цифровая обработка сигналов. – М.: Радио и связь, 1990