Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра радиотехнических систем

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ СВЕРТОЧНЫХ КОДОВ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторной работе

по курсам «Радиотехнические системы»,
«Радиотехнические системы передачи информации»
для студентов специальностей «Радиотехника»,
«Радиотехнические системы»

Минск 2007

УДК 621.391.2 (075.8)
ББК 32.841 я 73
И 85

С о с т а в и т е л и:
Э. М. Карпушкин, Г. Ф. Плугатарь

Исследование эффективности сверточных кодов : метод. указания
И 85 к лабораторной работе по курсам «Радиотехнические системы», «Радиотехнические системы передачи информации» для студ. спец. «Радиотехника», «Радиотехнические системы» / сост. Э. М. Карпушкин, Г. Ф. Плугатарь. – Минск : БГУИР, 2007. – 24 с. : ил.

В методических указаниях рассматривается ряд методов кодирования и декодирования сверочных кодов, исследуется их эффективность.

Настоящие методические указания предназначены для студентов, обучающихся по специальностям «Радиотехника», «Радиотехнические системы».

УДК 621.391.2(075.8)
ББК 32.841 я 73

1 ЦЕЛЬ РАБОТЫ

Цель работы – 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 [6]. 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 [1–7].

Двоичные равномерные коды делятся на блочные и сверточные (цепные, непрерывные). При использовании блочных кодов цифровая информация передаётся в виде 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, которые 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 [3] происходило в трёх направлениях в соответствии с тремя важнейшими *методами декодирования кодов*: методом декодирования по максимуму правдоподобия, методом последовательного декодирования и методом порогового декодирования. Метод декодирования по максимуму правдоподобия (алгоритм Витерби) теоретически наиболее эффективен, 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, полученные 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 сверточных 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 [3, 4].

Изучение сверточного кодирования и декодирования рассмотрим на примере систематического кода со скоростью передачи $R = \frac{k}{n} = \frac{1}{2}$.

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

На вход кодера поступает информационная последовательность $A(x)$, а с выхода снимаются две последовательности, одна из которых является повторением входной, а вторая – проверочной последовательностью $C(x)$. Далее с помощью ключа «К» формируется сверточный код путём чередования информационных и проверочных символов.

Проверочная последовательность $C(x)$ находится путём умножения входной последовательности

$$A(x) = a_0^{(1)} + a_1^{(1)} x + a_2^{(1)} x^2 + \dots$$

на генераторный (порождающий) полином

$$G(x) = g_0 + g_1 x + g_2 x^2 + \ldots + g_r x^r$$

следующим образом:

$$C(x) = A(x) \cdot G(x) = c_0^{(2)} + c_1^{(2)} x + c_2^{(2)} x^2 + \ldots .$$

Если степень полинома $G(x)$ равна r, то любой фиксированный информационный символ a_j может оказывать влияние на выходную последовательность $C(x)$ в течение не более чем $\mathrm{r}+1$ тактов. За это время с выхода кодера будет считано $m = n(r+1)$ символов. Величину m называют *кодовым ограничителем* кода, а n – количеством последовательностей на выходе кодера.

Кодирование, т.е. нахождение последовательности $C(x)$, осуществляется с помощью линейных фильтров без обратных связей (рисунок 2). При этом значения проверочных символов определяются выражением

$$C_j = g_0 \cdot A_j + g_1 \cdot A_{j-1} + \ldots + g_r \cdot A_{j-r} .$$

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

$$G(x) = 1 + x^3 + x^4 + x^5$$

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

$$\left[H^{\Delta}, I_e\right] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

где H^{Δ} – проверочная матрица;

I_e – единичная матрица.

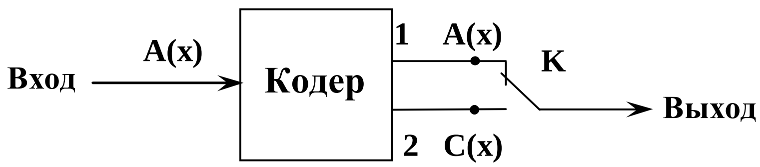

Рисунок 1

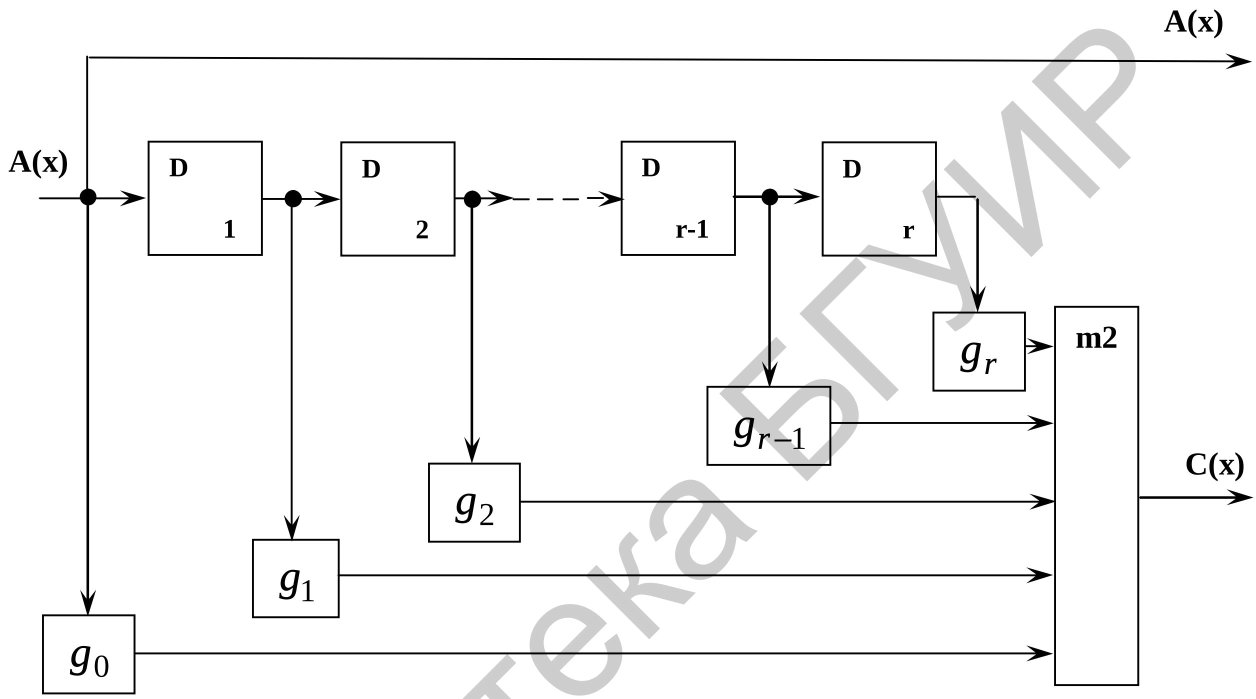

Рисунок 2

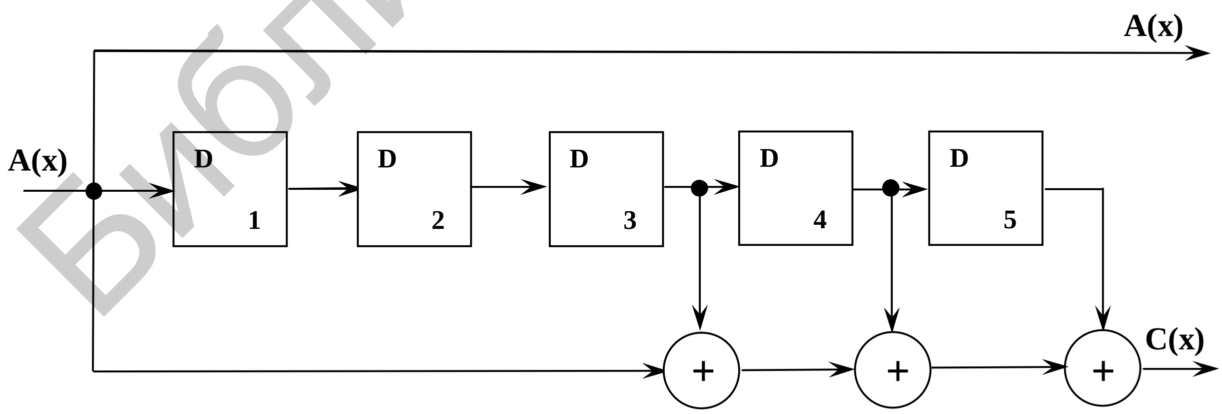

Рисунок 3

Проверочная матрица описывается коэффициентами порождающего полинома

$$H^{\Delta}=\begin{bmatrix} g_0 & 0 & 0 & 0 & 0 & 0 \\ g_1 & g_0 & 0 & 0 & 0 & 0 \\ g_2 & g_1 & g_0 & 0 & 0 & 0 \\ g_3 & g_2 & g_1 & g_0 & 0 & 0 \\ g_4 & g_3 & g_2 & g_1 & g_0 & 0 \\ g_5 & g_4 & g_3 & g_2 & g_1 & g_0 \end{bmatrix}.$$

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

$$A^*(x)=A(x)+E^1(x),$$
$$C^*(x)=C(x)+E^2(x),$$

где

$$E^1(x)=e_0^{(1)}+e_1^{(1)}x+e_2^{(1)}x^2+\ldots,$$
$$E^2(x)=e_0^{(2)}+e_1^{(2)}x+e_2^{(2)}x^2+\ldots$$

– шумовые последовательности, наложенные на информационную и проверочную последовательности.

Если принятую информационную последовательность $A^*(x)$ вновь умножить на неизменный генераторный полином $G(x)$ и сложить по модулю 2 с принятой проверочной последовательностью $C^*(x)$, то получим корректирующую последовательность – синдром:

$$S(x)=A^*(x)\cdot G(x)+C^*(x)=A(x)\cdot G(x)+$$
$$+E^{(1)}(x)\cdot G(x)+C(x)+E^{(2)}(x)=E^{(1)}(x)\cdot G(x)+E^{(2)}(x).$$

Отсюда видно, что значения синдрома

$$S(x)=s_0+s_1x+s_2x^2+\ldots$$

определяются исключительно шумами.

Схема декодера для сверточного кода, порождаемого полиномом $G(x)=1+x^3+x^4+x^5$, 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

$$S^T = \left[s_0^{(2)}, s_1^{(2)}, s_2^{(2)}, \ldots, s_5^{(2)} \right].$$

Символы этого синдрома подаются в регистр сдвига, показанный на рисунке 4 в нижней части декодера. Синдром S и шумовые символы e_u^i, ($u = 0, \ldots, 5$, $i = 1,2$) связаны между собой следующим матричным соотношением:

$$S = \begin{bmatrix} s_0^{(2)} \\ s_1^{(2)} \\ s_2^{(2)} \\ s_3^{(2)} \\ s_4^{(2)} \\ s_5^{(2)} \end{bmatrix} = \left[H^{\Lambda}, I_e \right] \cdot E(x) =$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} e_0^{(1)} \\ e_1^{(1)} \\ e_2^{(1)} \\ e_3^{(1)} \\ e_4^{(1)} \\ e_5^{(1)} \\ e_0^{(2)} \\ e_1^{(2)} \\ e_2^{(2)} \\ e_3^{(2)} \\ e_4^{(2)} \\ e_5^{(2)} \end{bmatrix}$$

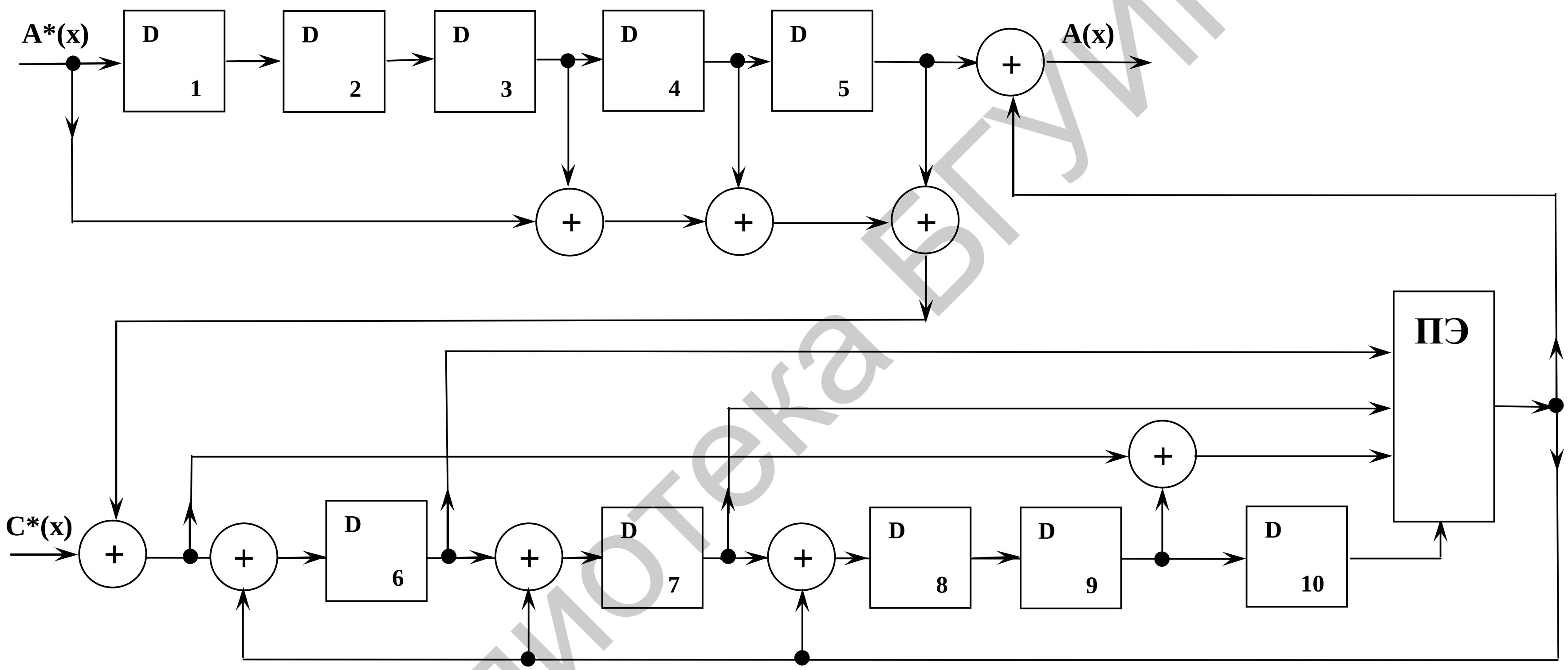

Рисунок 4

или, то же самое

$$
\begin{aligned}
s_0^{(2)} &= e_0^{(1)} + e_0^{(2)}\\
s_1^{(2)} &= + e_1^{(1)} + e_1^{(2)}\\
s_2^{(2)} &= + e_2^{(1)} + e_2^{(2)}\\
s_3^{(2)} &= e_0^{(1)} + + e_3^{(1)} + e_3^{(2)}\\
s_4^{(2)} &= e_0^{(1)} + e_1^{(1)} + + e_4^{(1)} + e_4^{(2)}\\
s_5^{(2)} &= e_0^{(1)} + e_1^{(1)} + e_2^{(1)} + + e_5^{(1)} + e_5^{(2)}.
\end{aligned}
$$

С помощью линейного преобразования синдрома S находятся величины K_1, K_2, K_3, K_4:

$$
\begin{aligned}
K_1 &= s_0^{(2)} = e_0^{(1)} + e_0^{(2)},\\
K_2 &= s_3^{(2)} = e_0^{(1)} + e_3^{(1)} + e_3^{(2)},\\
K_3 &= s_4^{(2)} = e_0^{(1)} + e_1^{(1)} + e_4^{(1)} + e_4^{(2)},\\
K_4 &= s_5^{(2)} + s_1^{(2)} = e_0^{(1)} + e_1^{(1)} + e_5^{(1)} + e_1^{(2)} + e_5^{(2)}.
\end{aligned}
\tag{1}
$$

Линейное преобразование, с помощью которого по синдрому S находится величина K_i, называется правилом о р т о г о н о л и з а ц и и, а получаемые с его помощью величины K_i называются с о с т а в н ы м и   п р о в е р к а м и. Составные проверки обладают следующим свойством: шумовой символ $e_0^{(1)}$, воздействующий на информационный символ в момент 0, входит во все составные проверки K_i $(\mathrm{i} = 1, 2, 3, 4)$, и в то же время любой другой шумовой символ входит не более чем в одну из этих проверок. Такая совокупность составных проверок $\{K_i\}$ называется о р т о г о н а л ь н о й относительно $e_0^{(1)}$.

Если $e_0^{(1)} = 1$ и остальные шумовые символы равны нулю, то составные проверки, задаваемые формулами (1), равны 1, а именно $K_1 = K_2 = K_3 = K_4 = 1$. В случае когда один из шумовых символов, отличных от $e_0^{(1)}$, равен 1, среди проверок K_i по крайней мере три будут равны 1. Если $e_0^{(1)} = 0$ и среди остальных шумовых символов имеется не более чем две единицы, то среди проверок равными единице будут самое большее две проверки. Если эти составные проверки подать на вход порогового элемента с порогом 2,5, то символ на выходе последнего будет равен $e_0^{(1)}$ всегда, когда число случайных ошибок в пределах кодового ограничения не больше двух. Складывая символ с выхода порогового элемента с принятым двоичным символом $\left(a_0^{(1)} + e_0^{(1)}\right)$, который хранится в ячейке регистра в верхней части декодера, шумовой символ $e_0^{(1)}$ всегда можно исправить. При этом выходной сигнал с порогового элемента по цепи обратной связи подаётся на схему вычисления синдрома и корректирует последний, устраняя влияние $e_0^{(1)}$. Аналогичным образом устраняют влияние $e_1^{(1)}$, $e_2^{(1)}$ и т. д. Такая процедура называется *декодированием с обратной связью*.

Рассмотренная схема декодирования имеет два существенных недостатка:
- в общем случае реализация порогового элемента требует больших затрат оборудования;
- если при декодировании произошла ошибка (значение $e_j^{(1)}$ 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 $e_0^{(1)}$ проверочных соотношений, аналогичную вышеприведённому примеру с $G(x) = 1 + x^3 + x^4 + x^5$.

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 $m^* = i/2 \cdot (i+1) + 1$. Смысл понятия эффективного кодового ограничения ясен из утверждения: если количество возникших ошибок на длине m^* разрядов не превосходит $i/2$, то символ $e_0^{(1)}$ может быть правильно декодирован с помощью мажоритарного элемента. Здесь значение i эквивалентно понятию реализуемого кодового расстояния d ($d \le d$, 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 n_0 информационных символов в течение следующих r тактов кодируются нули. При этом после декодирования $n(n_0+r)$ символов декодер возвращается в исходное нулевое состояние и начинается процесс декодирования для следующей группы символов. Таким образом, размножение ошибок возможно только на длине $n(n_0+r)$ символов. Скорость передачи в этом случае уменьшается и равна

$$R^{'} = n_0 \cdot k / n(n_0 + r).$$

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 канале 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 определяются 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 a_j по принятой совокупности информационных и проверочных символов. Эти уравнения находятся из проверочной матрицы H^*, несколько отличающейся от вышеприведённой отсутствием коррекции содержимого декодера, т. е.

$$H^* = \begin{bmatrix} g_r & g_{r-1} & \dots & g_1 & g_0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & g_r & \dots & g_2 & g_1 & g_0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ - & - & - & - & - & - & - & - & - & - & - & - & - \\ 0 & 0 & \dots & 0 & g_r & g_{r-1} & \dots & g_0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}.$$

Для рассматриваемого примера эта матрица будет иметь следующий вид:

$$
H^* = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
a_{j-5} & & a_{j-3} & & a_{j-1} & & a_{j+1} & & a_{j+3} & & a_{j+5} & & c_{j+1} & & c_{j+3} & & c_{j+5} \\
 & a_{j-4} & & a_{j-2} & & a_j & & a_{j+2} & & a_{j+4} & & c_j & & c_{j+2} & & c_{j+4} &
\end{bmatrix}.
$$

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

$$a_j = a_{j-5} + a_{j-4} + a_{j-3} + c_j;$$

$$a_j = a_{j-2} + a_{j-1} + a_{j+3} + c_{j+3};$$

$$a_j = a_{j+6} + a_{j+2} + a_{j+5} + c_{j+5}.$$

Добавляя к этим уравнениям тривиальное $a_j = a_j$, получим 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 (a_0, c_0). При бессиндромном декодировании без размножения ошибок декодер содержит r дополнительных ячеек регистра и, кроме того, длина последовательности, на которой реализуется расстояние $d = i$, равна

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

вместо $2(r+1)$ символов.

2.2 Сверточные 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 длины b, то на каждую из последовательностей (имеется в виду k информационных и $(n-k)$ проверочных последовательностей) накладывается вспышка ошибок длиной в среднем b/n. Основная идея, использованная при построении всех свёрточных кодов, исправляющих пакеты ошибок [1, 4], 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 неискаженных 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 сверточный код не обеспечит исправления ошибок.

В настоящее время известно достаточно много разнообразных сверточных 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 $\leq b$ при условии, что две соседние пачки разделены между собой защитным промежутком $\geq 3b+1$. 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

$$G(x) = x^2 + x^4.$$

Этот код обладает способностью исправлять пакеты ошибок длины $\leq 4 (b \leq 4)$, имеющих защитный интервал $\geq 3b+1 = 13$. Схема кодирующего устройства представлена на рисунке 6. С выхода коммутатора K поочередно следуют информационные и проверочные символы, формирующие выходную последовательность.

Пусть на вход кодирующего устройства подается последовательность информационных символов

$$\mathbf{1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1}.$$

Последовательность проверочных символов на выходе сумматора по модулю 2 соответственно имеет следующий вид

$$0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 1\ \ 1\ \ 0\ \ 1\ \ 1.$$

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

$$\mathbf{1}\ 0\ \mathbf{0}\ 0\ \mathbf{0}\ 1\ \boxed{\mathbf{1}\ 0\ \mathbf{0}}\ 1\ \mathbf{0}\ 1\ \mathbf{1}\ 0\ \mathbf{1}\ 1\ \mathbf{1}\ 1\ \mathbf{0}\ 1\ \mathbf{0}\ 0\ \mathbf{1}\ 1.$$

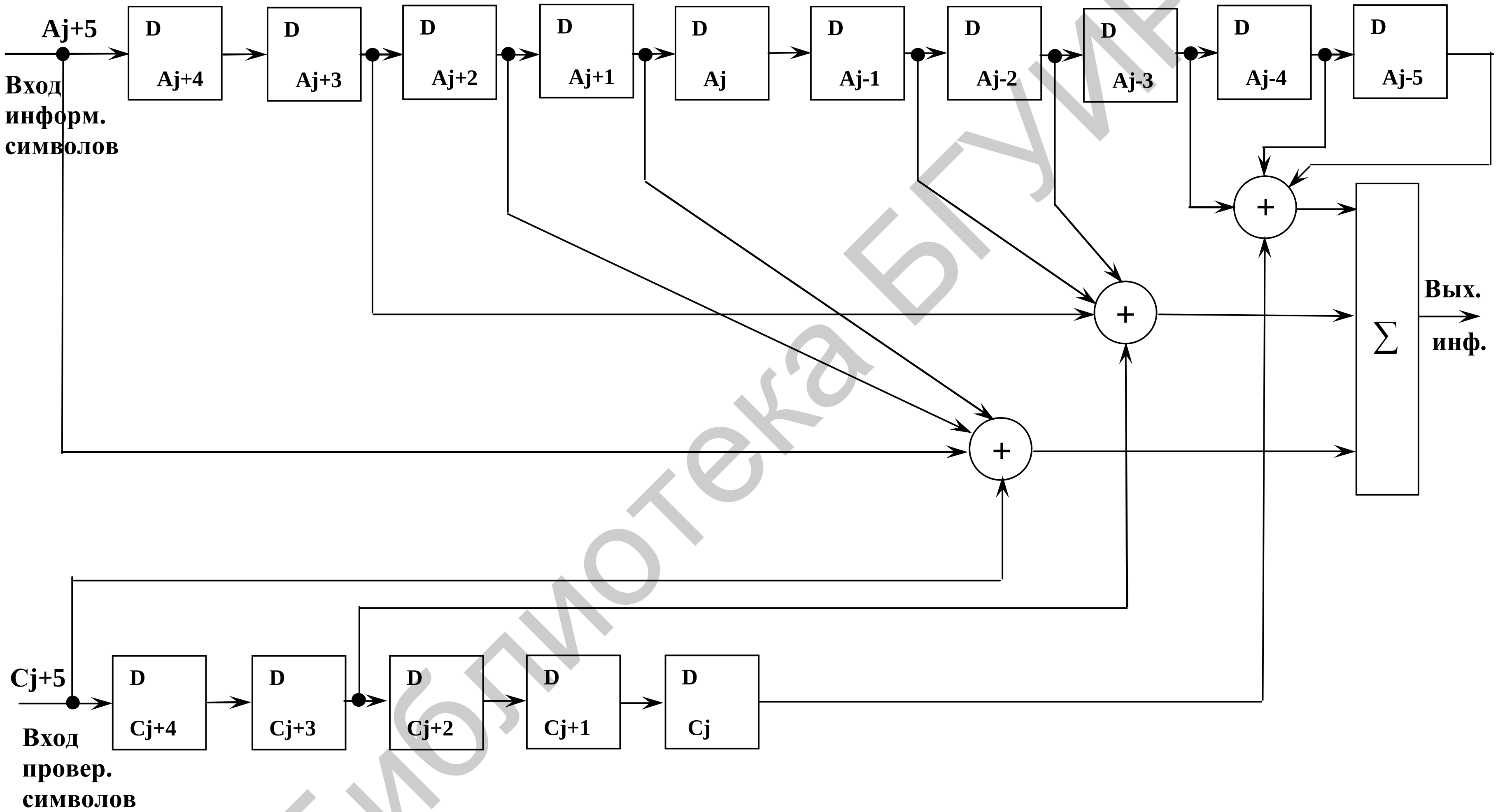
Aj+5
Вход
информ.
символов
D
Aj+4
D
Aj+3
D
Aj+2
D
Aj+1
D
Aj
D
Aj-1
D
Aj-2
D
Aj-3
D
Aj-4
D
Aj-5
+
+
+
∑
Вых.
инф.
Cj+5
Вход
провер.
символов
D
Cj+4
D
Cj+3
D
Cj+2
D
Cj+1
D
Cj

Рисунок 5

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

1 0 **0** 0 **0** 1 $\boxed{\mathbf{0}\ 1\ \mathbf{1}}$ 1 **0** 1 **1** 0 **1** 1 **1** 1 **0** 1 **0** 0 **1** 1,

где искаженные символы отмечены прямоугольником.

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 7, принятая последовательность разделяется на последовательность информационных символов A’(x):

1 0 0 0 1 0 1 1 1 0 0 1

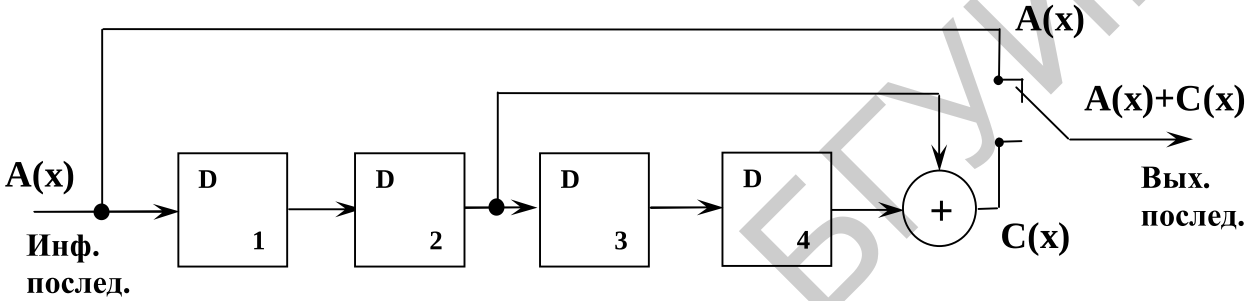

Рисунок 6

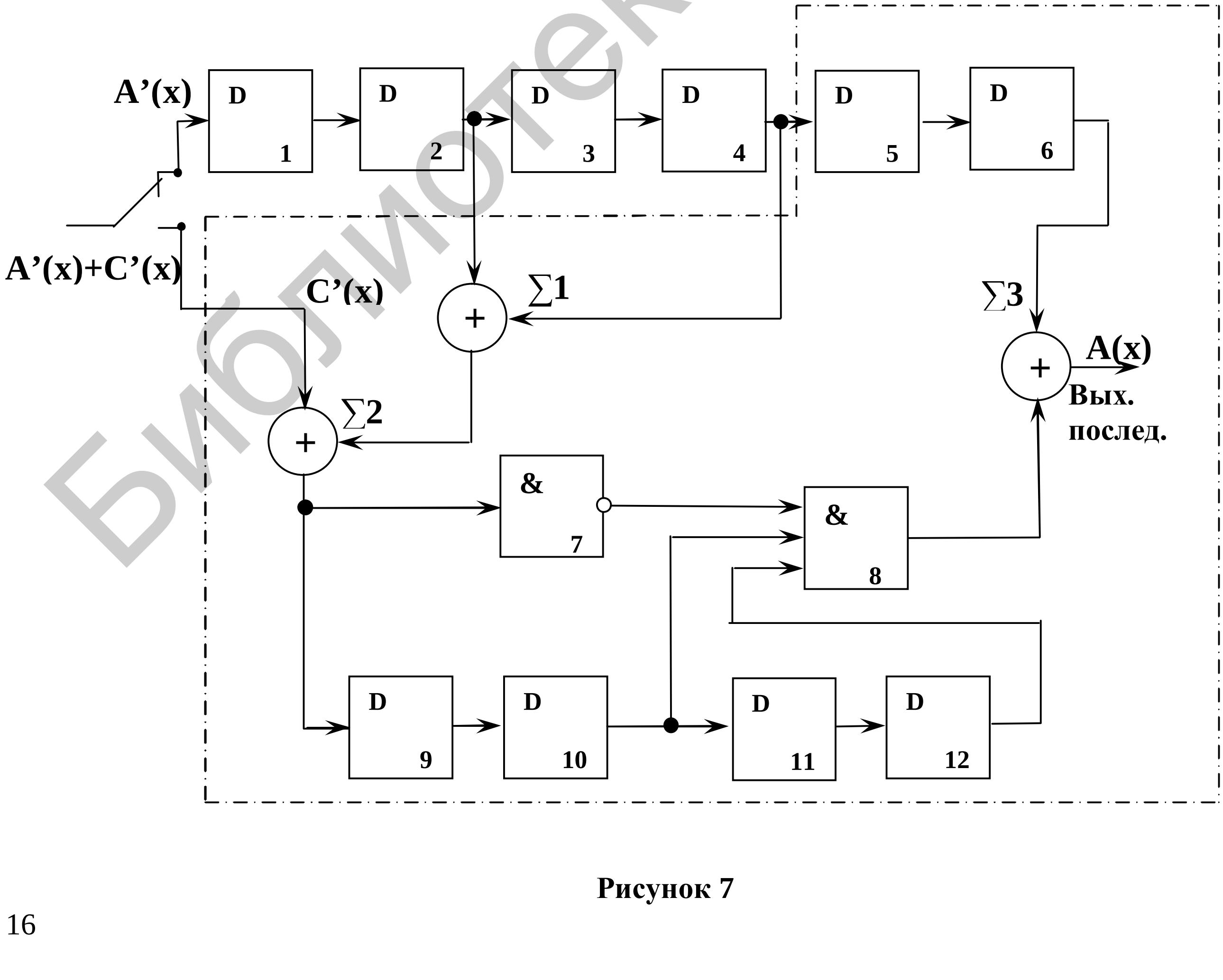

Рисунок 7

и последовательность проверочных символов C'(x):

0 0 1 1 1 1 0 1 1 1 0 1 1.

Первая из последовательностей поступает на регистр, в точности повторяющей кодер (см. рисунок 6). Если ошибок нет, то сформированная этим регистром и сумматором ∑2 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 и с п р а в л я ю щ а я   п о с л е д о в а т е л ь н о с т ь (**синдром**)

0 0 0 1 0 1 1 1 1 0 0 0 0.

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 следующими тремя составляющими:

0 0 0 1 0 0 0 0 0 0 0 0

0 0 0 1 1 0 0 0 0 0.

0 0 0 1 1 0 0 0

Отсюда видно, что произошла ошибка в 4-м проверочном и 4 и 5-м информационных символах (соответственно в 7, 8 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

1 1 1 0 1 0 0 0 0 1 1 1 1 - **вх. 1**
0 0 0 1 0 1 1 1 1 0 0 - **вх. 2**
0 0 0 1 0 1 1 1 1 - **вх. 3**

0 0 0 0 0 1 1 0 0 - **вых.**

Поскольку корректирующая последовательность формируется через $3b/2$ 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 следующий вид:

1 0 0 0 1 0 1 1 1 0 0 1
0 0 0 0 0 1 1 0 0 0 0 0 0 0

1 0 0 1 0 0 1 1 1 0 0 1 .

На пути информационных символов в декодере имеются $3b/2$ запоминающих ячеек. Это соответствует $3b$ символам во входной последовательности. Очевидно, чтобы вывести все ошибочные символы, требуется защитный промежуток из $3b+1$ 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 сверточных кодов, описанных разделе 2.

2 Построить кодер, синдромный и бессиндромный декодеры кода, задаваемого полиномом $G(x)=1+x^6+x^7+x^9+x^{10}+x^{11}$.

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 сверточного кода, задаваемого полиномом $G(x)=x^3+x^6$. Какой пакет ошибок код может исправить? Чему равен защитный промежуток?

4 ОПИСАНИЕ ЛАБОРАТОРНОЙ УСТАНОВКИ

Установка включает два кодирующих устройства, задаваемых полиномами

$G(x)=1+x^3+x^4+x^5$ и $G(x)=x^2+x^4$,

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); пятиразрядный двоичный счётчик числа тактируемых импульсов; сумматор по модулю 2, имитирующий канал передачи информации; устройство вывода информации, представляющее собой 12-разрядный сдвиговый регистр; индикаторы (индикатор светится при наличии логической «1» в последовательности символов); галетный переключатель.

Положение 1 галетного переключателя соответствует кодированию и декодированию по синдрому свёрточного кода с $G(x)=1+x^3+x^4+x^5$, исправляющего одиночные ошибки.

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

Положение 3 соответствует кодированию и декодированию свёрточного кода с $G(x)=x^2+x^4$, исправляющего пакеты ошибок длиной $b \leq 4$.

Ввод информации в кодирующее устройство и наложение помех в канале передачи информации осуществляется путём нажатия на кнопку «Счёт».

5 ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОГО ЗАДАНИЯ

Подключать установку только к сети +5 В !

1 Ознакомиться со схемой лабораторной установки, представленной на передней панели.

2 Тумблер «Сеть» установить в положение «Вкл». При этом загорятся некоторые световые индикаторы.

3 Нажать кнопку «Сброс», обнуляющую все регистры. Большинство лампочек должны погаснуть.

4 Переключатель П1 поставить в положение «1».

5 Набрать заданную преподавателем последовательность информационных символов на наборном поле «Входная последовательность». Переключатели наборного поля «Помехи» установить в положение «0».

6 Подать набранную последовательность на устройство кодирования и декодирования 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 заполнит выходной регистр «Выходная информация».

7 Нажать на кнопку «Сброс».

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

Устанавливать поочередно переключатель П1 в соответствующее положение при проверке декодеров 1, 2, 3. Составить таблицу 1 прохождения информации (в указанных контрольных точках).

Таблица 1 – Прохождение информации

Такты	1	2	3	4	…	20	21	22
П1а $I_{Вх}$								
П1б a_i								
П1в c_i								
П1г a_i								
П1 c_i								
М2 a_i								
d_i								
$в_i$								
ДШ 1								
2								
3								
4								
П1д = $I_{вых}$								

9 Проверить прохождение и исправление ошибок декодерами в информационных (нечетных) символах. Количество ошибок устанавливается ***нечетными*** тумблерами регистра помехи ($a_i, 2a_i, 3a_i, 4a_i, 5a_i$ и т.д.). При каждой новой проверке увеличивается количество ошибочных символов в канале связи. Определить количество информационных символов, исправляемых данным декодером.

10 Проверить прохождение и исправление ошибок декодерами в проверочных (четных) символах. Количество ошибок устанавливается ***четными*** тумблерами регистра помехи ($c_i, 2c_i, 3c_i, 4c_i, 5c_i$ и т.д.). При каждой новой проверке увеличивается количество проверочных символов. Определить количество проверочных символов, исправляемых каждым декодером.

11 Проверить прохождение и исправление декодерами пакета ошибок длиной «b», состоящего из определенного количества символов ($a_i c_i, 2a_i c_i, 2a_i 2c_i, 3a_i 2c_i, 3a_i 3c_i$ и т.д.). Определить количество исправляемых символов каждым декодером в пакете ошибок длиной «b».
12 Проверить прохождение и исправление информации декодерами при постоянном защитном промежутке $(3b+1)=13$ и переменной длине «b» пакета ошибок, равного $2, 3, 4, 5$ и т.д. Определить количество исправляемых символов каждым декодером в пакете ошибок длиной «b».
13 Проверить прохождение и исправление информации при постоянной длине пакета ошибок «$b=4$» и переменном защитном промежутке $3b+1$, равном соответственно $11, 12, 13, 14, 15$ и т.д.

Примечание:

1) по пунктам $6–12$ проверить 3 декодера $(1, 2, 3)$ и составить не менее одной таблицы прохождения информации для указанного преподавателем типа декодера;

2) после каждого выполненного пункта сделать и записать выводы о результатах эксперимента, указав количество исправляемых символов *каждым* декодером.

14 Установить тумблер «Сеть» в положение «Выкл».

6 СОДЕРЖАНИЕ ОТЧЁТА

1 Решение предварительного задания.
2 Результаты экспериментальной проверки.
3 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? Каковы его достоинства и недостатки?
4 В чем сущность бессиндромного декодирования? Каковы его 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 Объяснить как составлен алгоритм кодирования для сверточного кода, исправляющего пакет ошибок?
11 Что такое защитный промежуток?

8 ЗАЩИТА

1 Какова структура синдрома сверточного кода, исправляющего пакеты ошибок? Показать сигнал на выходе каждого каскада (см. рисунки 6,7), если входная информация соответствует указанной 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.

Таблица 2 – Расположение информации на выходе каскадов в тактах

Такты	0	1	2	3	4	…	18	19	20
Iвх									
D1									
D2									
D3									
D4									
Cx									
Ax+Cx									
A’x+C’x									
A’x									
C’x									
D1									
D2									
D3									
D4									
D5									
D6									
Σ1									
Σ2									
D9									
D10									
D11									
D12									
&7									
&8									
Σ3									

2 Какие изменения необходимо ввести в кодер и декодер, изучаемые в лабораторной работе (см. рисунки 6,7), для обеспечения коррекции пакета ошибок, состоящего из следующего числа символов:
а) 6; б) 8; в) 10; г) 12 .

9 ЛИТЕРАТУРА

1 Информационные технологии в радиотехнических системах : учеб. пособие / под ред. И. Б. Федорова. – М. : МГТУ им. Н. Э. Баумана, 2003.
2 Скляр, Б. Цифровая связь. Теоретические основы и практическое применение. Пер. с англ. – 2-е изд. / Б. Скляр. – М. : Изд. дом «Вильямс», 2003.
3 Кассами, Т. Теория кодирования / Т. Кассами [и др.]. – М. : Мир, 1978.
4 Питерсон, У. Коды, исправляющие ошибки / У. Питерсон, Э. Уэлдон. – М. : Мир, 1976.
5 Хетагуров, Я. А. Повышение надежности цифровых устройств методами избыточного кодирования / Я. А. Хетагуров, Ю. П. Руднев. – М. : Энергия, 1974.
6 Шляпоберский, В. И. Основы техники передачи дискретных сообщений / В. И. Шляпоберский. – М. : Связь, 1973.
7 Харкевич, А. А. Борьба с помехами / А. А. Харкевич. – М. : Наука, 1965.

Св. план 2006, поз. 80

Учебное издание

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ СВЕРТОЧНЫХ КОДОВ

Методические указания
к лабораторной работе
по курсам «Радиотехнические системы»,
«Радиотехнические системы передачи информации»
для студентов специальностей «Радиотехника»,
«Радиоэлектронные системы»

С о с т а в и т е л и:
Карпушкин Эдуард Михайлович
Плугатарь Геннадий Федорович

Редактор Т. П. Андрейченко
Корректор М. В. Тезина

Подписано в печать 29.12.2006.	Формат 60x84 1/16.	Бумага офсетная.
Гарнитура «Таймс».	Печать ризографическая.	Усл. печ. л. 1,51.
Уч.-изд. л. 1,2.	Тираж 200 экз.	Заказ 566.

Издатель и полиграфическое исполнение: Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
ЛИ №02330/0056964 от 01.04.2004. ЛП №02330/0131666 от 30.04.2004.
220013, Минск, П. Бровки, 6