Лекция 5
Коды Хэмминга

Семейство групповых систематических (n,k) кодов, предложенных Хэммингом (Hamming) в 1950 г., имеет следующие параметры:

$$n = 2^r - 1 = 2^{n-k} - 1 \qquad (1)$$

$$k = 2^r - 1 - r = 2^{n-k} - 1 - (n-k) \qquad (2)$$

Если задать количество проверочных символов $r = n - k = 3$, то по формулам (1), (2) получим код с параметрами:

$$n = 2^3 - 1 = 7; \quad k = 2^3 - 1 - 3 = 4,$$

то есть рассмотренный выше код (7,4) является также и кодом Хэмминга.

При $r = 4$ получим код Хэмминга с параметрами:

$$n = 2^4 - 1 = 15; \quad k = 2^4 - 1 - 4 = 11,$$

то есть код (15,11) и т. д.

Рассмотрим вопрос реализации кодера и декодера для кода Хэмминга (7,4). Функциональная схема кодера представлена на рис.1,а. На вход кодера поступает комбинация, состоящая из четырех двоичных информационных символов (a_1, a_2, a_3, a_4). На выходе необходимо получить семисимвольное кодовое слово в систематическом коде. Четыре информационных символа поступают со входа на выход кодера без изменений. Проверочные символы формируются с помощью трех сумматоров по модулю 2. На входы этих сумматоров поступают информационные символы в соответствии с уравнениями

$$\begin{cases} b_1 = a_1 + a_2 + a_3 \\ b_2 = a_2 + a_3 + a_4 \\ b_3 = a_1 + a_2 + a_4 \end{cases} \qquad (3)$$

На выходах сумматоров формируются проверочные символы b_1, b_2, b_3 , которые затем поступают на выход кодера.

Функциональная схема декодера для указанного кода представлена на рис. 1,б. На вход декодера поступают семисимвольные кодовые слова (a_1' , a_2' , a_3' , a_4' , b_1' , b_2' , b_3') , возможно содержащие ошибки.

Предположим, что перед данным декодером установлено устройство, позволяющее рассредоточивать пакеты ошибок (этот вопрос будет рассмотрен далее), и на вход декодера могут поступать кодовые слова только с одиночными ошибками. Исправление одиночных ошибок в декодере основано на проверке выполнения уравнений

$$\begin{cases} a_1 + a_2 + a_3 + b_1 = 0 \\ a_2 + a_3 + a_4 + b_2 = 0 \\ a_1 + a_2 + a_3 + b_3 = 0 \end{cases} \qquad (4)$$

Для этого в декодере установлены три четырехвходовых сумматора по модулю 2. Наборы символов, которые поступают на входы этих сумматоров, определяются в соответствии с уравнениями (4):

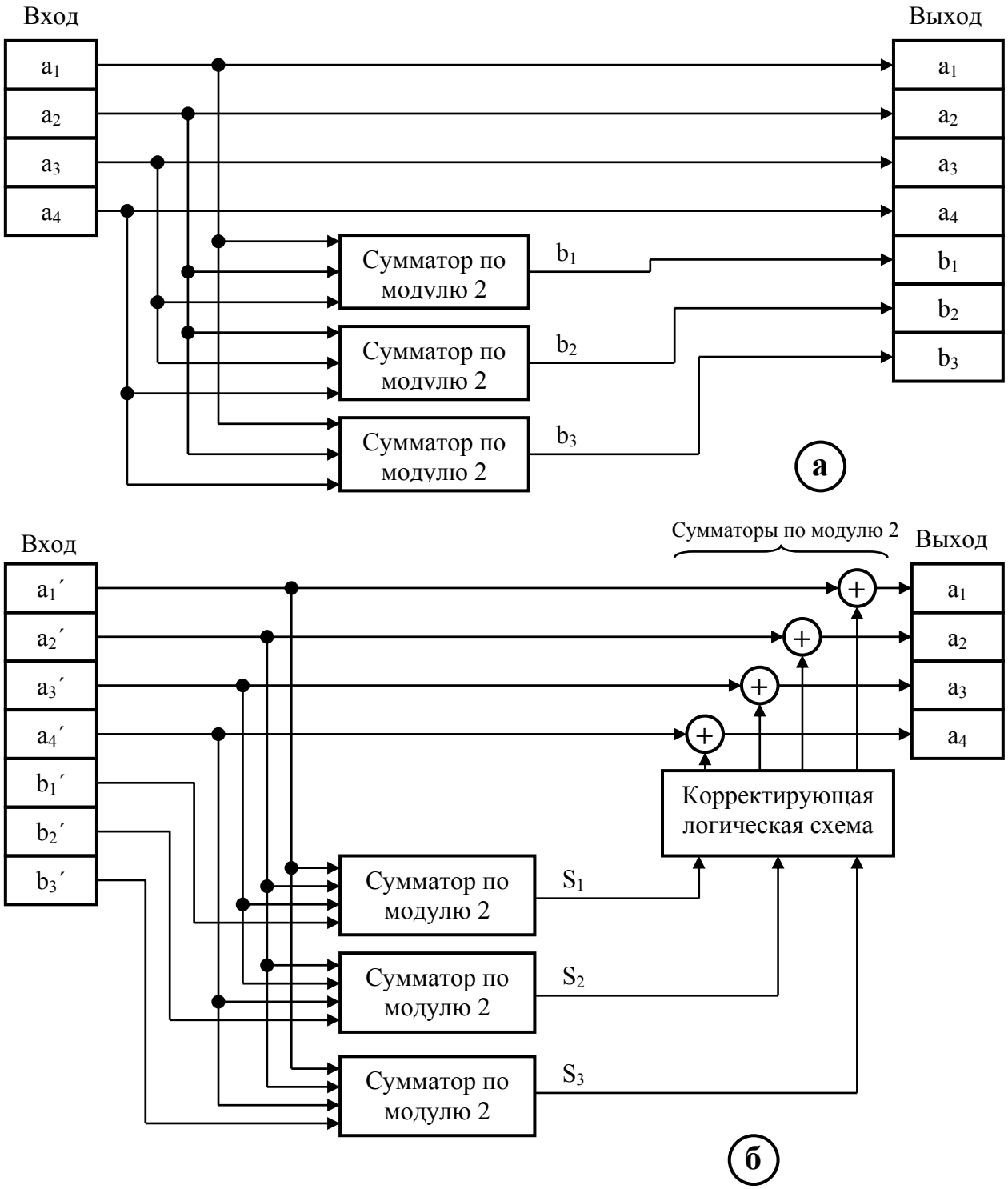

Рис.1. Функциональные схемы кодера и декодера кода Хэмминга (7,4): а – кодер; б – декодер.

на первый сумматор - a_1’ , a_2’ , a_3’ , b_1’ ;
на второй сумматор - a_2’ , a_3’ , a_4’ , b_2’ ;
на третий сумматор - a_1’ , a_2’ , a_4’ , b_3’ .

Символы, формируемые на выходах четырехвходовых сумматоров, которые получаются в результате сложения по модулю 2 перечисленных выше информационных и проверочных символов, обозначим как S_1, S_2, S_3. Совокупность символов S_1, S_2, S_3 называется синдромом. Этот термин позаимствован из медицины, где он обозначает совокупность признаков, характеризующих заболевание человека. В теории помехоустойчивого кодирования синдром характеризует конфигурацию ошибок в кодовом слове. Если синдром $S_1 = 0$, $S_2 = 0$, $S_3 = 0$, то, при указанном выше предположении, это означает, что выполняются все три уравнения системы (5.6), и кодовое слово не содержит ошибок. Если какие-либо составляющие синдрома не равны нулю, то это свидетельствует о наличии ошибок в кодовом слове. Например, если $S_1 = 1$, $S_2 = 0$, $S_3 = 1$, то это возможно при невыполнении первого и третьего уравнений системы (5.6), и ошибочным считается символ a_3'.

Символы синдрома поступают на входы корректирующей логической схемы, в которой по этим символам определяется ошибочный информационный символ. На соответствующем выходе корректирующей логической схемы формируется символ 1, который поступает на один из входов двухвходового сумматора по модулю 2. На второй вход этого сумматора поступает ошибочный символ. В результате значение этого символа изменяется на противоположное, то есть происходит исправление ошибки.

Полиномиальные и циклические коды

Построение полиномиальных кодов основано на том, что кодовое слово вида

$$Z = (a_{n-1},\ a_{n-2}, \ldots, a_2, a_1, a_0)$$

представляется в виде степенного полинома (многочлена)

$$Y(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_2x^2 + a_1x^1 + a_0x^0 \qquad (5)$$

Подобным образом любое число в произвольной системе счисления можно записать в виде

$$C = c_{n-1}g^{n-1} + c_{n-2}g^{n-2} + \ldots + c_2g^2 + c_1g^1 + c_0g^0$$

где: g – основание системы счисления; c_{n-1} … c_0 - цифры этого числа.

Кодовое слово, состоящее из двоичных символов, например, Z = 10110010101 преобразуется в многочлен

$$Y(x) = 1\cdot x^{10} + 0\cdot x^9 + 1\cdot x^8 + 1\cdot x^7 + 0\cdot x^6 + 0\cdot x^5 + 1\cdot x^4 + 0\cdot x^3 +$$
$$+ 1\cdot x^2 + 0\cdot x^1 + 1\cdot x^0 = x^{10} + x^8 + x^7 + x^4 + x^2 + 1 \quad .$$

Можно произвести и обратное преобразование.

Указанное представление позволяет при кодировании и декодировании кодовых слов производить математические операции над степенными многочленами. Такие операции относительно несложно реализуются при помощи регистров сдвига с обратными связями.

К классу полиномиальных кодов относятся циклические избыточные коды (Cyclic Redundancy Codes – CRC), называемые также в отечественной литературе просто циклическими кодами. Кодовые слова CRC обладают следующим дополнительным свойством: при перестановке самого правого символа любого кодового слова на крайнюю левую позицию получается другое кодовое слово. Можно также переставлять самый левый символ на крайнюю правую позицию, снова получая кодовое слово, не содержащее ошибок. Это свойство обеспечивает дополнительные удобства при реализации кодирующих и декодирующих устройств для циклических кодов.

В теории помехоустойчивого кодирования показано, что среди всех степенных многочленов, соответствующих разрешенным кодовым словам данного полиномиального кода, имеется ненулевой многочлен наименьшей степени, которая равна $r=n-k$, который обладает следующим свойством: он делит без остатка, то есть нацело, все остальные многочлены, образованные из разрешенных кодовых слов этого кода. Такой многочлен (полином) $F(x)$ называется порождающим (образующим, генерирующим). Указанное свойство порождающего многочлена позволяет применять его для проверки кодовых слов на наличие ошибок. Порождающий многочлен полностью определяет данный полиномиальный код.

Формирование кодовых слов полиномиального кода состоит из трех операций.

1. Многочлен $G(x)$ степени k-1, образованный только из информационных символов, которые поступают на вход кодера, умножается на x^r . При этом происходит сдвиг кодового слова на r разрядов влево, а на крайних правых позициях образуется r нулей.

Например: кодовую комбинацию 101 можно преобразовать в многочлен x^2+1; если умножить этот многочлен на x^r и выбрать $r=3$, то получим

$$\left(x^2+1\right)x^3 = x^5 + x^3 = 1\cdot x^5 + 0\cdot x^4 + 1\cdot x^3 + 0\cdot x^2 + 0\cdot x^1 + 0\cdot x^0$$

что соответствует кодовой комбинации 101000.

2. Произведение $G(x)\cdot x^r$ делится на порождающий многочлен $F(x)$. При этом образуется остаток $q(x)$ степени, не превосходящей r-1.

3. Остаток $q(x)$ суммируется с произведением $G(x)\cdot x^r$. Поскольку степень остатка не более r-1 символы остатка располагаются вместо нулей на крайних правых позициях. Полученное в результате кодовое слово

$$Q(x) = G(x)\cdot x^r + q(x) \qquad (6)$$

должно делиться на порождающий полином $F(x)$ без остатка. Это слово представлено в систематическом коде, поскольку все информационные символы сохранили свои значения и расположены в начале блока, а проверочные символы, то есть символы остатка, размещены после информационных.

Укрупненную структурную схему кодера для полиномиального и, в частности, циклического кода можно представить в виде рис. 2,а. Кодер состоит из устройства умножения на x^r и деления на порождающий многочлен $F(x)$ (УУДПМ), ключа Кл, управляемого этим устройством.

При поступлении k информационных символов ключ Кл замкнут, и эти символы поступают одновременно на вход УУДПМ и на выход кодера. После прихода последнего информационного символа ключ Кл размыкается, и на выход кодера поочередно поступают символы остатка. В результате на выходе кодера образуется n-символьное кодовое слово.

Вариант структурной схемы декодера для указанных кодов представлен в укрупненном виде на рис. 2,б. Декодер состоит из регистра сдвига, содержащего n ячеек (РГС), устройства деления на порождающий многочлен (УДПМ), ключа Кл, детектора ошибки (ДО) и выходного сумматора по модулю 2. Декодер работает следующим образом. Кодовое слово $Q(x)$, состоящее из n символов, посимвольно записывается в регистр РГС. Одновременно через замкнутый ключ Кл символы поступают на вход УДПМ. После прихода всех n символов кодового слова ключ Кл размыкается. Символы остатка от деления принятого кодового слова на порождающий многочлен $F(x)$, которые в данном случае представляют собой синдром и зависят только от конфигурации ошибок, поступают на вход детектора ошибок. Если синдром равен нулю, то считается что кодовое слово не содержит ошибок, и его информационные символы поступают на выход декодера через сумматор по модулю 2 без изменений. Если синдром не равен нулю, то в

детекторе ошибок на основе анализа данного синдрома определяются ошибочные символы кодового слова. Затем при прохождении ошибочных символов через выходной сумматор по модулю 2 на его второй вход поступают логические единицы с выхода детектора ошибок. В результате значения ошибочных символов изменяются на противоположные. Таким образом происходит исправление ошибок.

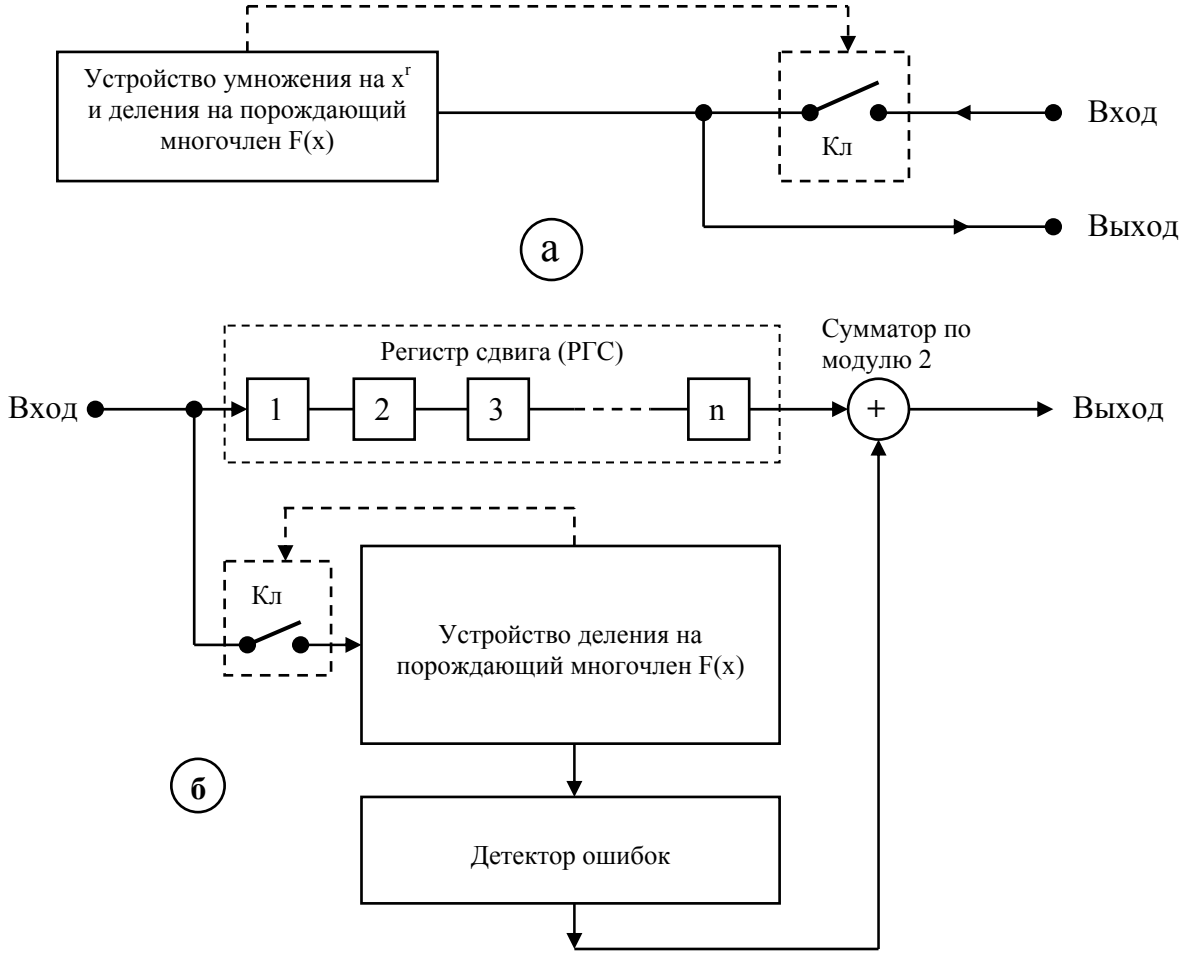

Рис.2. Функциональные схемы кодера и декодера для полиномиального кода: а – кодер; б – декодер.

Рассмотрим более подробно схему кодера для указанного выше кода (7,4). Данный код можно рассматривать и как полиномиальный . Как отмечалось выше, порождающая матрица для этого кода имеет вид (5.9)

$$G = \begin{bmatrix} 1000101 \\ 0100111 \\ 0010110 \\ 0001011 \end{bmatrix} . \quad (6)$$

Кодовое слово, стоящее в нижней строке порождающей матрицы, - 0001011 соответствует порождающему многочлену

$$F(x) = x^3 + x + 1 \quad (7)$$

Степень *F(x)* равна количеству проверочных символов $r = 3$.

Схема кодера для кода (7,4) представлена на рис. 3,а.

Схема содержит регистр сдвига, состоящий из трех ячеек (пронумерованных цифрами 1,2,3), двух сумматоров по модулю 2 и двух ключей - Кл1 и Кл2. Данная схема

осуществляет умножение многочлена, соответствующего информационным символам, на $x^r=x^3$ и деление получившегося произведения на порождающий многочлен $F(x) = x^3 + x + 1$.

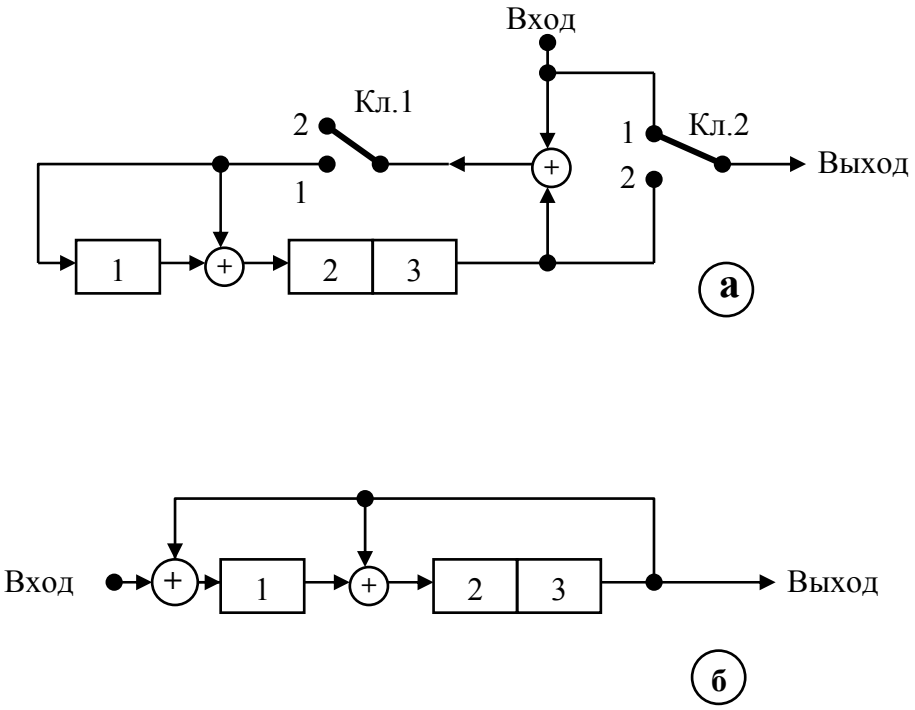

Рис.3. а – схема кодера для полиномиального кода (7,4); б – схема устройства деления на порождающий многочлен $F(x) = x^3+x+1$.

Перед поступлением очередной комбинации информационных символов на вход кодера в ячейках регистра сдвига должны быть записаны нули. Допустим, что на вход кодера поступила комбинация информационных символов 1000 (первая строка порождающей матрицы (6): $a_1=1$, $a_2=a_3=a_4=0$). Кодер должен сформировать проверочные символы 101 ($b_1=1$, $b_2=0$, $b_3=1$). Для облегчения понимания работы кодера составим таблицу (табл. 1). В первой колонке указаны номера сдвигов в регистре, во второй - значения информационных символов, в следующих трех колонках указаны значения символов, записанных в ячейках регистра при каждом сдвиге. В исходном состоянии ключи Кл1 и Кл2 находятся в положении 1. Информационные символы поступают сразу на выход кодера и одновременно записываются в регистр сдвига.

Таблица 1

№ сдвига	Входные символы	Содержимое ячеек регистра		
		1	2	3
1	1	1	1	0
2	0	0	1	1
3	0	1	1	1
4	0	**1**	**0**	**1**

После прихода последнего четвертого информационного символа ключи Кл1 и Кл2 переводятся в состояние 2 и символы остатка 101, которые остаются записанными в ячейках регистра после четвертого сдвига, поступают на выход, начиная с третьей ячейки.

Аналогичным образом формируются 15 остальных кодовых слов данного кода.

В рассмотренном примере выполняются следующие математические операции. Исходная последовательность информационных символов 1000 соответствуют многочлену $G(x)=x^3$. В кодере производится умножение $G(x)$ на $x^r = x^3$ то есть

$$G(x)\cdot x^r = x^3 \cdot x^3 = x^6$$

Полученное произведение делится на порождающий многочлен

$$\frac{G(x)\cdot x^r}{F(x)} = \frac{x^6}{x^3+x+1}$$

По правилам деления многочленов

$$\begin{array}{rl} x^6 & \left|\underline{x^3+x+1}\right. \\ \oplus\ \underline{x^6+x^4+x^3} & x^3+x+1 \\ x^4+x^3 & \\ \oplus\ \underline{x^4+x^2+x} & \\ x^3+x^2+x & \\ \oplus\ \underline{x^3+x+1} & \\ \textit{остаток}\quad x^2+1 & \end{array}$$

Получили остаток $x^2 + 1$, который преобразуется в требуемую кодовую комбинацию 101. Таким же образом можно показать, что при поступлении на вход кодера остальных 15 возможных наборов информационных символов формируются проверочные символы (символы остатка), соответствующие разрешенным кодовым словам данного кода (табл. 5.2).

Рассмотрим схему деления на порождающий многочлен, установленную в декодере рис. 3,б. Кодовое слово

$(a_1', a_2', a_3', a_4', b_1', b_2', b_3')$, возможно содержащее ошибки, представляется в виде многочлена

$$Y(x) = a_1^{'}x^6 + a_2^{'}x^5 + a_3^{'}x^4 + a_4^{'}x^3 + b_1^{'}x^2 + b_2^{'}x^1 + b_3^{'}x^0$$

Если кодовое слово не содержит ошибок, то этот многочлен должен делиться на порождающий многочлен $F(x)=x^3+x+1$ без остатка.

Пусть на вход декодера поступило кодовое слово 1000101, соответствующее первой строке порождающей матрицы (6). Это кодовое слово преобразуется в многочлен x^6+x^2+1. Убедимся, что после прихода всех семи символов на вход схемы рис. 3,б в ячейках регистра сдвига будут записаны символы 0 (табл. 2).

Таблица 2

№ сдвига	Входные символы	Содержимое ячеек регистра		
		1	2	3
1	1	1	0	0
2	0	0	1	0
3	0	0	0	1
4	0	1	1	0
5	1	1	1	1
6	0	1	0	1
7	1	**0**	**0**	**0**

Таким образом в результате получен нулевой остаток, то есть кодовое слово не содержит ошибок.

Математически выполнено деление многочлена x^6+x^2+1 на порождающий многочлен

$$\begin{array}{ll} x^6 + x^2 + 1 & \left|\underline{x^3 + x + 1}\right. \\ \oplus \quad \underline{x^6 + x^4 + x^3} & x^3 + x + 1 \\ \quad x^4 + x^3 + x^2 + 1 & \\ \oplus \quad \underline{x^4 + x^2 + x} & \\ \quad x^3 + x + 1 & \\ \oplus \quad \underline{x^3 + x + 1} & \\ \quad \textit{остаток} \quad 0 & \end{array}$$

Можно показать, что тот же результат будет получен и для остальных 15 разрешенных кодовых слов.

Теперь допустим, что на вход декодера поступило кодовое слово 1100101, то есть возникла ошибка во втором символе от начала (a_2). Этому кодовому слову соответствует многочлен $x^6+x^5+x^2+1$. Разделим его на порождающий многочлен

$$
\begin{array}{rll}
 & x^6 + x^5 + x^2 + 1 & \left| \underline{x^3 + x + 1} \right. \\
\oplus & \underline{x^6 + x^4 + x^3} & x^3 + x^2 + x \\
 & x^5 + x^4 + x^3 + x^2 + 1 & \\
\oplus & \underline{x^5 + x^3 + x^2} & \\
 & x^4 + 1 & \\
\oplus & \underline{x^4 + x^2 + x} & \\
 & \textit{остаток } x^2 + x + 1 &
\end{array}
$$

Остаток x^2+x+1, который является синдромом, преобразуется в комбинацию двоичных символов 111. Такой же остаток должен образоваться в ячейках регистра сдвига, изображенного на схеме рис. 3,б, после семи сдвигов. Проверим это с помощью табл. 3

Таблица 3

№ сдвига	Входные символы	Содержимое ячеек регистра		
		1	2	3
1	1	1	0	0
2	1	1	1	0
3	0	0	1	1
4	0	1	1	1
5	1	0	0	1
6	0	1	1	0
7	1	**1**	**1**	**1**

Аналогичным образом можно показать, что все остатки (синдромы) при наличии одиночных и двойных ошибок в кодовых словах те же, что указаны в табл. 3 (читать символы в колонках табл. 3 следует сверху вниз).