

## Алгоритмы кодирования/декодирования для линейных, циклических и БЧХ кодов

В тексте все вычисления проводятся в двоичной арифметике.

### Задача помехоустойчивого кодирования

Рассмотрим задачу передачи потока битовой информации по каналу с шумом с возможностью автоматического исправления ошибок, допущенных при передаче. При *блоковом кодировании* входящий поток информации разбивается на непересекающиеся блоки фиксированной длины  $k$ , каждый из которых кодируется и декодируется независимо<sup>1</sup>. Обозначим один такой блок через  $\mathbf{u} \in \{0, 1\}^k$ . Здесь и далее жирным шрифтом обозначаются вектора-столбцы. Предполагается, что во входном потоке данных, вообще говоря, нет избыточности<sup>2</sup>. Поэтому для реализации схемы, способной исправлять ошибки, необходимо закодировать блок  $\mathbf{u}$  в некоторое кодовое слово большей длины путем добавления избыточности в передаваемые данные. Обозначим кодовое слово через  $\mathbf{v} \in \{0, 1\}^n$ ,  $n > k$ ,  $n$  – длина кода, а через  $m = n - k$  – количество требуемых дополнительных бит. Для кодирования всевозможных битовых блоков  $\mathbf{u}$  необходимо использовать  $2^k$  кодовых слов длины  $n$   $\{\mathbf{v}_1, \dots, \mathbf{v}_{2^k}\}$ . Назовём это множество  $(n, k)$ -*блоковым кодом*, а величину  $r = k/n$  – *скоростью кода*. Рассмотрим случай возникновения *случайных ошибок* в канале с шумом без добавлений/стираний битов. Здесь при передаче по каналу кодовое слово  $\mathbf{v}$  превращается в принятое слово  $\mathbf{w} = \mathbf{v} + \mathbf{e}$  той же длины  $n$ , где  $\mathbf{e} \in \{0, 1\}^n$  – вектор ошибок ( $e_i = 1$ , если в  $i$ -ом бите произошла ошибка). Таким образом, при передаче не происходит добавлений или стираний битов, а ошибки могут возникать в произвольных битах<sup>3</sup>. После получения  $\mathbf{w}$  алгоритм декодирования пытается восстановить переданное слово  $\mathbf{v}$  путем поиска среди всевозможных кодовых слов ближайшего к  $\mathbf{w}$  в метрике Хэмминга. Обозначим результат работы алгоритма декодирования через  $\hat{\mathbf{v}}$ . На последнем этапе декодированное слово  $\hat{\mathbf{v}}$  переводится в декодированное слово исходного сообщения  $\hat{\mathbf{u}}$ .

Определим минимальное расстояние блокового кода  $d$  как минимальное хэммингово расстояние для всех пар различных кодовых слов. Очевидно, что код с расстоянием  $d$  способен обнаруживать до  $d - 1$  ошибки<sup>4</sup> и исправлять до  $\lfloor (d - 1)/2 \rfloor$  ошибок.

В общем случае для задания  $(n, k)$ -блокового кода необходимо использовать таблицу всех кодовых слов размера  $2^k \times n$ , а на этапе декодирования необходимо перебирать все  $2^k$  кодовых слов для поиска ближайшего к принятому  $\mathbf{w}$ . В результате на практике использование произвольного  $(n, k)$ -блокового кода возможно только при небольших значениях  $n$  и  $k$ . Далее в тексте будет рассмотрен ряд дополнительных ограничений на множество кодовых слов блокового кода, которые позволят перейти от экспоненциальных требований по памяти для хранения кода и по сложности алгоритмов кодирования/декодирования к линейным по  $n$  и  $k$ .

### Линейные коды

Множество  $\{0, 1\}^n$  с операциями суммы и произведения по модулю 2 образует линейное пространство над конечным полем из двух элементов  $\{0, 1\}$ . Блоковый  $(n, k)$ -код  $C$  называется *линейным*, если множество его кодовых слов образует линейное подпространство размерности  $k$  общего линейного пространства  $\{0, 1\}^n$ . Таким образом, для линейного кода произвольная линейная комбинация кодовых слов является кодовым словом. Для линейного кода минимальное расстояние  $d$  может быть вычислено более эффективно, чем для произвольного блокового кода: это расстояние определяется как минимальный хэммингов вес (количество ненулевых бит) среди всех

<sup>1</sup>Альтернативой блоковому кодированию является свёрточное кодирование, при котором вычисляется свёртка входящего потока информации с некоторой фиксированной последовательностью, т.е. результат кодирования зависит не только от текущего блока длины  $k$ , но и от некоторого количества предыдущих блоков. Одним из наиболее успешных семейств свёрточных кодов являются т.н. турбо-коды.

<sup>2</sup>Например, по сети передаётся zip-архив.

<sup>3</sup>Альтернативой здесь является случай возникновения пакетов ошибок, связанных, например, с коммуникационными помехами.

<sup>4</sup>Здесь речь идёт о гарантированном обнаружении произвольных  $d - 1$  ошибок. В общем случае  $(n, k)$ -блоковый код способен обнаружить  $2^n - 2^k$  векторов ошибок, т.е. таких векторов ошибок, которые не переводят одно кодовое слово в другое.

ненулевых кодовых слов<sup>5</sup>. Код  $C$  является линейным пространством размерности  $k$ , следовательно, у него существует базис из  $k$  векторов  $\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_{k-1} \in \{0, 1\}^n$ . Тогда  $\mathbf{v} \in C$  может быть представлен как  $\mathbf{v} = \sum_{i=0}^{k-1} u_i \mathbf{g}_i$ ,  $u_i \in \{0, 1\}$  или, эквивалентно,  $\mathbf{v} = G\mathbf{u}$ , где  $G = [\mathbf{g}_0 | \mathbf{g}_1 | \dots | \mathbf{g}_{k-1}] \in \{0, 1\}^{n \times k}$  – порождающая матрица кода. Матрица  $G$  определена с точностью до эквивалентных преобразований столбцов.

**Кодирование.** Под процедурой кодирования будем понимать произвольное взаимно-однозначное преобразование из множества исходных блоков длины  $k$  во множество кодовых слов. Для линейного кода, заданного своей порождающей матрицей  $G$ , процедура кодирования может быть организована как  $\mathbf{v} = G\mathbf{u}$ , где  $\mathbf{u} \in \{0, 1\}^k$  – блок исходного сообщения. Однако, на практике оказывается удобным т.н. *систематическое кодирование*, при котором  $k$  бит исходного сообщения копируются в фиксированные  $k$  бит кодового слова, а затем вычисляются остальные  $m = n - k$  проверочных бит. При использовании систематического кодирования этап преобразования из декодированного кодового слова  $\hat{\mathbf{v}}$  в декодированное сообщение  $\hat{\mathbf{u}}$  становится тривиальным. Рассмотрим способ систематического кодирования для линейного кода, заданного порождающей матрицей  $G$ . Очевидно, что с помощью эквивалентных преобразований столбцов матрица  $G$  может быть приведена к виду, в котором некоторые  $k$  строк образуют единичную подматрицу. Пусть, не ограничивая общности, единичная подматрица образуется в первых  $k$  строках преобразованной матрицы  $\tilde{G}$ , т.е.  $\tilde{G} = \begin{bmatrix} I_k \\ P \end{bmatrix}$ , где  $I_k$  – единичная матрица размера  $k \times k$ , а  $P \in \{0, 1\}^{m \times k}$ . Тогда кодирование по правилу  $\mathbf{v} = \tilde{G}\mathbf{u}$  будет систематическим, при котором первые  $k$  бит кодового слова  $\mathbf{v}$  являются битами исходного сообщения  $\mathbf{u}$ .

Линейное пространство  $\{0, 1\}^n$  может быть разложено в прямую сумму линейных подпространств  $C$  и  $C^\perp$ , где  $C^\perp$  – ортогональное подпространство для  $C$ ,  $\dim C^\perp = m$ . Пусть  $\mathbf{h}_0, \dots, \mathbf{h}_{m-1} \in \{0, 1\}^n$  – базис  $C^\perp$ . Составим матрицу

$$H = \begin{bmatrix} \mathbf{h}_0^T \\ \mathbf{h}_1^T \\ \vdots \\ \mathbf{h}_{m-1}^T \end{bmatrix} \in \{0, 1\}^{m \times n}.$$

Очевидно, что  $\mathbf{h}_j^T \mathbf{g}_i = 0 \forall i, j$ . Следовательно, для любого кодового слова  $\mathbf{v} \in C$  выполнено  $H\mathbf{v} = \mathbf{0}$ . Матрица  $H$  называется *проверочной матрицей* кода. Проверочная матрица определена с точностью до эквивалентных преобразований строк. Рассмотрим процедуру построения систематической проверочной матрицы  $H$  по заданной порождающей матрице кода  $G$ . С помощью эквивалентных преобразований столбцов приведем порождающую матрицу  $G$  к виду  $\tilde{G} = \begin{bmatrix} I_k \\ P \end{bmatrix}$ . Тогда проверочная матрица  $H$  может быть построена как  $H = [P \quad I_m] \in \{0, 1\}^{m \times n}$ , где  $I_m$  – единичная матрица размера  $m \times m$ . Действительно, в этом случае  $H\mathbf{v} = H\tilde{G}\mathbf{u} = (P + P)\mathbf{u} = \mathbf{0}$ .

Таким образом, задание линейного кода требует рассмотрения порождающей матрицы размера  $n \times k$  или проверочной матрицы размера  $m \times n$  (линейные требования по  $k$  и  $n$ ). Каждая из этих матриц определена с точностью до эквивалентных преобразований столбцов или строк (соответствует произволу в выборе базиса в пространствах  $C$  и  $C^\perp$ ). Однако, фиксирование позиций бит при систематическом кодировании задаёт порождающую и проверочную матрицу однозначно.

**Пример 1.** Линейный блочный (6,3)-код задан порождающей матрицей

$$G = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

Требуется для данного кода осуществить несистематическое и систематическое кодирование векторов  $\mathbf{u}_1 = [0 \ 1 \ 1]^T$  и  $\mathbf{u}_2 = [1 \ 0 \ 1]^T$ , построить проверочную матрицу кода  $H$ , а также определить минимальное кодовое расстояние  $d$ .

<sup>5</sup>Однако, такой поиск  $d$  по-прежнему является экспоненциальным по  $k$ .

Несистематическое кодирование находим непосредственно:

$$[\mathbf{v}_1, \mathbf{v}_2] = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} [\mathbf{u}_1, \mathbf{u}_2] = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}. \quad (1)$$

Для построения систематического кодирования с помощью эквивалентных преобразований столбцов выделим в матрице  $G$  единичную подматрицу размера  $3 \times 3$  (над стрелками указано проводимое преобразование над столбцами):

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \xrightarrow{1 \leftarrow 1+2} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

В последней матрице в строках (3, 5, 1) стоит единичная подматрица. В результате систематического кодирования  $\mathbf{u}_1, \mathbf{u}_2$ , при котором биты 1, 2, 3 исходного сообщения переходят, соответственно, в биты 3, 5, 1 кодового слова, вычисляется следующим образом

$$[\mathbf{v}_1^{systematic}, \mathbf{v}_2^{systematic}] = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} [\mathbf{u}_1, \mathbf{u}_2] = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}. \quad (2)$$

С учетом выделенной единичной подматрицы в порождающей матрице  $G$ , нетрудно найти проверочную матрицу кода  $H$ . Для этого формируем матрицу  $P$  из строк  $G$ , отличных от строк с единичной подматрицей, т.е.  $P = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ . Далее нужно разместить столбцы  $P$ , соответственно, в 3-ом, 5-ом и 1-ом столбце  $H$ , а во 2-й, 4-ый и 6-ой столбец  $H$  поставить единичную подматрицу:

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Проверим, что в результате как систематического кодирования (2), так и несистематического (1) были действительно найдены кодовые слова:

$$H[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_1^{systematic}, \mathbf{v}_2^{systematic}] = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Найдем теперь минимальное кодовое расстояние  $d$ . Для этого построим матрицу всех  $2^3 = 8$  кодовых слов и найдем для них минимальный ненулевой хэммингов вес:

$$[\mathbf{v}_1, \dots, \mathbf{v}_8] = G[\mathbf{u}_1, \dots, \mathbf{u}_8] = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Отсюда получаем, что  $d = 3$ . Следовательно, рассмотренный код способен исправить одну ошибку и обнаружить две ошибки.

**Декодирование.** Под процедурой декодирования будем понимать разбиение множества из всех  $2^n$  принятых сообщений  $\mathbf{w}$  на  $2^k$  подмножеств, каждое из которых соответствует декодированию в своё кодовое слово. Назовём *синдромом* принятого сообщения  $\mathbf{w}$  вектор  $\mathbf{s} = H\mathbf{w}$ ,  $\mathbf{s} \in \{0, 1\}^m$ , где  $H$  – проверочная матрица кода. С учётом представления  $\mathbf{w} = \mathbf{v} + \mathbf{e}$ , получаем  $\mathbf{s} = H\mathbf{w} = H(\mathbf{v} + \mathbf{e}) = H\mathbf{v} + H\mathbf{e} = H\mathbf{e}$ . Очевидно, что синдром является нулевым вектором тогда и только тогда, когда принятое сообщение является кодовым словом. Вектор ошибок  $\mathbf{e}$  удовлетворяет системе линейных уравнений  $H\mathbf{e} = \mathbf{s}$ . Матрица данной системы имеет размер  $m \times n$ . Следовательно, все решения данной СЛАУ описываются как  $\mathbf{e} = \hat{\mathbf{e}} + G\mathbf{u}$ , где  $\hat{\mathbf{e}}$  – произвольное частное решение системы, а  $G\mathbf{u}$  – решение однородной системы  $H\mathbf{e} = \mathbf{0}$ , где  $\mathbf{u}$  – произвольный вектор длины  $k$ . Таким образом, получаем  $2^k$  различных решений для вектора ошибок  $\mathbf{e}$ . Среди них решение с наименьшим хэмминговым весом даёт искомый вектор ошибок. После получения вектора ошибок  $\mathbf{e}$  декодирование осуществляется по правилу  $\hat{\mathbf{v}} = \mathbf{w} + \mathbf{e}$ . Заметим, что описанная процедура декодирования в синдромной формулировке полностью не зависит от конкретной схемы кодирования.

Общая процедура декодирования линейного кода предполагает построение таблицы размера  $2^m \times n$ , в которой для каждого синдрома принятого сообщения  $\mathbf{s}$  записывается решение СЛАУ  $H\mathbf{e} = \mathbf{s}$  с наименьшим хэмминговым весом. Сложность построения такой таблицы равна  $\tilde{O}(2^n)^6$ , т.к. для каждого из  $2^m$  синдромов необходимо перебирать  $2^k$  решений очередной СЛАУ. В результате процедура декодирования произвольного линейного кода требует экспоненциальных затрат как по памяти, так и по сложности алгоритма декодирования.

**Пример 2.** Рассмотрим линейный код из примера 1. Пусть исходный вектор  $\mathbf{u} = [0 \ 1 \ 1]^T$ . Систематическое кодирование для него было получено раньше:  $\mathbf{v} = [1 \ 1 \ 0 \ 0 \ 1 \ 0]^T$ . Пусть при передаче происходит ошибка во втором бите, т.е. принятый вектор  $\mathbf{w} = [1 \ 0 \ 0 \ 0 \ 1 \ 0]^T$ . Для декодирования  $\mathbf{w}$  найдём его синдром

$$\mathbf{s} = H\mathbf{w} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Далее необходимо найти все решения системы  $H\mathbf{e} = \mathbf{s}$ . Частное решение этой системы легко найти с учётом того, что в столбцах 2, 4, 6 проверочной матрицы  $H$  стоит единичная подматрица. Пусть  $\hat{e}_1 = \hat{e}_3 = \hat{e}_5 = 0$ . Тогда  $\hat{e}_2 = s_1, \hat{e}_4 = s_2, \hat{e}_6 = s_3$ , т.е.  $\hat{\mathbf{e}} = [0 \ 1 \ 0 \ 0 \ 0 \ 0]^T$ . Решение однородной системы уже было найдено раньше на этапе вычисления кодового расстояния  $d$ :

$$G[\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_8] = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Таким образом, все 8 решений системы  $H\mathbf{e} = \mathbf{s}$  могут быть записаны как

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Выбирая среди них решение с наименьшим весом, находим  $\mathbf{e} = [0 \ 1 \ 0 \ 0 \ 0 \ 0]^T$ , т.е.  $\hat{\mathbf{v}} = \mathbf{w} + \mathbf{e} = [1 \ 1 \ 0 \ 0 \ 1 \ 0]^T$ .

## Циклические коды

Ранее в курсе были рассмотрены т.н. циклические подпространства  $\mathbb{F}_p^n$ , рассматриваемого как линейное пространство размерности  $n$  с элементами векторов из  $\mathbb{F}_p$ . Напомним, что подпространство  $I$  называется циклическим, если вместе с любым вектором  $\mathbf{v} = \{v_0, v_1, \dots, v_{n-1}\} \in I$  оно содержит и его циклический сдвиг

<sup>6</sup>Символом  $\tilde{O}(x)$  обозначена сложность с точностью до логарифмических факторов, т.е.  $O(x \log^\gamma x)$ .

$\{v_1, \dots, v_{n-1}, v_0\}$ . При дополнительном требовании линейности циклического подпространства  $I$  можно показать, что  $I$  является главным идеалом в фактор-кольце  $\mathbb{F}_p[x]/(x^n - 1)$ , порождённым некоторым делителем  $g(x)$  многочлена  $x^n - 1$ . Переведём этот алгебраический результат на язык кодов с учётом  $p = 2$ .

Назовём линейный блочный  $(n, k)$ -код *циклическим*, если в нём любой циклический сдвиг кодового слова является кодовым словом. Поставим в соответствие произвольному вектору  $\mathbf{v} = \{v_0, v_1, \dots, v_{n-2}, v_{n-1}\} \in \{0, 1\}^n$  полином вида  $v(x) = v_0 + v_1x + \dots + v_{n-2}x^{n-2} + v_{n-1}x^{n-1}$ . Тогда рассмотренное выше свойство главного идеала для циклического кода можно переформулировать следующим образом: для  $(n, k)$ -линейного циклического блочного кода найдётся полином  $g(x)$  степени  $m = n - k$  такой, что

1. Все кодовые слова  $v(x)$  могут быть представлены как  $g(x)q(x)$ , где  $q(x)$  – некоторый полином степени не выше, чем  $k - 1$ ;
2. Полином  $g(x)$  является делителем полинома  $x^n + 1$ .

Такой полином  $g(x)$  называется *порождающим полиномом* циклического кода. Любой полином, являющийся делителем  $x^n + 1$ , является порождающим для некоторого циклического кода длины  $n$ .

**Кодирование.** С учётом того, что для циклического кода любой кодовый полином  $v(x)$  делится на порождающий полином  $g(x)$ , процедура кодирования может быть организована как  $v(x) = g(x)u(x)$ , где  $u(x)$  – входящий полином степени не выше  $k - 1$ . Такое кодирование будет несистематическим. Оно соответствует рассмотрению порождающей матрицы кода  $G = [g_0 | g_1 | \dots | g_{k-1}]$ , где базисные вектора  $g_i$  определяются полиномами  $g(x)x^i$ ,  $i = 0, \dots, k - 1$ .

Для организации систематического кодирования полинома  $u(x)$  рассмотрим полином  $x^m u(x)$ , где  $m$  – число проверочных бит кода и степень  $g(x)$ . Поделим  $x^m u(x)$  на  $g(x)$  с остатком:

$$x^m u(x) = g(x)q(x) + r(x),$$

где  $\deg r(x) < m$ . Тогда

$$x^m u(x) + r(x) = g(x)q(x) \in C.$$

Заметим, что  $\deg x^m u(x) \geq m$ . Таким образом, в векторном представлении полином  $x^m u(x) + r(x)$  имеет в крайних правых позициях элементы  $u(x)$ . В результате процесс систематического кодирования  $u(x)$  может быть реализован как

$$v(x) = x^m u(x) + \text{mod}(x^m u(x), g(x)).$$

Такая схема кодирования соответствует выбору порождающей матрицы кода с базисными векторами  $g_i$ , определяемыми полиномами  $x^{m+i} + \text{mod}(x^{m+i}, g(x))$ .

По аналогии с понятием проверочной матрицы для циклического кода можно ввести понятие *проверочного полинома*. Полином  $h(x)$  степени  $k$  называется проверочным, если выполнено свойство

$$g(x)h(x) = x^n + 1.$$

Таким образом, проверочный полином является обратным для порождающего полинома  $g(x)$  в фактор-кольце  $\mathbb{F}_2[x]/(x^n + 1)$  и может быть найден с помощью расширенного алгоритма Евклида. Пусть  $v(x)$  – произвольный кодовый полином, т.е. он представим в виде  $v(x) = g(x)q(x)$ . Тогда

$$v(x)h(x) = q(x) \underbrace{g(x)h(x)}_0 = 0.$$

Пусть проверочный полином имеет вид  $h(x) = \sum_{i=0}^k h_i x^i$ . Тогда нетрудно построить проверочную матрицу кода как

$$H = \begin{bmatrix} h_0 & h_1 & h_2 & \dots & h_k & 0 & \dots & 0 & 0 \\ 0 & h_0 & h_1 & \dots & h_{k-1} & h_k & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & h_0 & h_1 & \dots & h_{k-1} & h_k \end{bmatrix} \in \{0, 1\}^{m \times n}.$$

Такой выбор проверочной матрицы соответствует рассмотрению базисных векторов  $h_j$ , определяемых полиномами  $h(x)x^j$ ,  $j = 0, \dots, m - 1$ .

**Декодирование.** Назовём *синдромом* принятого полинома  $w(x)$  полином  $s(x) = \text{mod}(w(x), g(x)) = \text{mod}(v(x) + e(x), g(x)) = \text{mod}(e(x), g(x))$ <sup>7</sup>. Как и раньше для случая линейных кодов синдром является нулевым многочленом тогда и только тогда, когда  $w(x)$  является кодовым словом. Общий алгоритм декодирования

<sup>7</sup>Очевидно, что синдром можно определить и с помощью проверочного полинома как  $s(x) = \text{mod}(w(x)h(x), x^n + 1)$ .

циклического кода принципиально не отличается от общего алгоритма декодирования линейного кода. В нём рассматривается общее решение системы  $e(x) = s(x) + g(x)u(x)$  для всех возможных полиномов  $u(x)$  степени  $k - 1$ , а искомый полином ошибок определяется как решение с минимальным числом ненулевых слагаемых.

Таким образом, для задания циклического кода достаточно определить порождающий или проверочный полином. По сравнению с линейными кодами в циклических кодах операции умножения матриц на вектора и решение СЛАУ заменяются на более простые операции умножения полиномов и деление полиномов с остатком. Однако, общий алгоритм декодирования по-прежнему имеет экспоненциальную сложность по  $k$ .

**Пример 3.** Пусть длина циклического кода  $n = 7$ . Для построения циклического кода необходимо сначала найти разложение полинома  $x^7 + 1$  на неприводимые множители. Так как  $7 = 2^3 - 1$ , то все ненулевые элементы поля  $\mathbb{F}_2^3$  являются корнями  $x^7 + 1$ . Известно, что каждый многочлен над  $\mathbb{F}_2$  содержит в расширении поля  $\mathbb{F}_2^3$  вместе с любым своим корнем  $\beta$  также смежные корни вида  $\beta^2, \beta^{2^2}, \beta^{2^3}, \dots$ . Пусть  $\alpha$  – произвольный примитивный элемент поля  $\mathbb{F}_2^3$ . Тогда с учетом  $\alpha^7 = 1$  легко найти разбиение всех элементов поля на смежные классы:

$$(\alpha, \alpha^2, \alpha^4); (\alpha^3, \alpha^6, \alpha^5); \alpha^7.$$

Таким образом, многочлен  $x^7 + 1$  имеет один неприводимый делитель степени 1, а также два неприводимых делителя степени 3. В результате получаем разложение

$$x^7 + 1 = (x + 1)(x^3 + x + 1)(x^3 + x^2 + 1).$$

В качестве  $g(x)$  может выступать произвольный делитель  $x^7 + 1$ . Выберем в качестве  $g(x)$  многочлен  $x^3 + x + 1$ . В результате получаем циклический  $(7, 4)$ -код.

Пусть входной полином  $u(x)$  равен  $x^3 + x^2$ . Его несистематическое кодирование находим как

$$v(x) = u(x)g(x) = (x^3 + x^2)(x^3 + x + 1) = x^6 + x^5 + x^4 + x^2.$$

В бинарном представлении это соответствует

$$\mathbf{u} = [0011], \mathbf{v} = [0010111].$$

Для осуществления систематического кодирования необходимо найти остаток от деления многочлена  $x^3u(x)$  на  $g(x)$ . В результате находим

$$v(x) = x^3u(x) + \text{mod}(x^3u(x), g(x)) = x^6 + x^5 + \text{mod}(x^6 + x^5, x^3 + x + 1) = x^6 + x^5 + x.$$

В бинарном представлении этот соответствует

$$\mathbf{u} = [0011], \mathbf{v} = [0100011].$$

Таким образом, действительно биты входного сообщения  $\mathbf{u}$  воспроизводятся в крайних правых битах кодового слова  $\mathbf{v}$ .

## Коды БЧХ

Полином  $m_\alpha(x) \in \mathbb{F}_2[x]$  называется минимальным полиномом для элемента  $\alpha \in \mathbb{F}_2^l$ , если он является полиномом минимальной степени, для которого  $\alpha$  является корнем. В частности, минимальный полином для примитивного элемента  $\alpha$  называется примитивным полиномом. Можно показать, что корнями минимального полинома  $m_\alpha(x)$  являются  $\{\alpha, \alpha^2, \alpha^4, \dots, \alpha^{2^q}\}$ . Данный набор элементов из поля  $\mathbb{F}_2^l$  называется классом смежности для элемента  $\alpha$ . Смежные классы, порожденные различными элементами поля, либо совпадают, либо не пересекаются. Можно показать, что полином

$$\prod_{i=1}^q (x + \alpha^{2^i}) = x^q + \lambda_{q-1}x^{q-1} + \dots + \lambda_1x + \lambda_0$$

имеет коэффициенты из  $\mathbb{F}_2$  и является минимальным полиномом для  $\alpha$ , а также для всех элементов поля, входящих вместе с  $\alpha$  в один смежный класс. Отсюда выводится метод построения минимального полинома для заданного элемента поля  $\alpha$ :

1. Построить смежный класс, порождённый элементом  $\alpha$ ;
2. Найти коэффициенты полинома  $m_\alpha(x)$  путем перемножения многочленов  $x + \alpha^{2^i}$  для всех  $i = 1, \dots, q$ .

В примере 3 для циклического кода был рассмотрен случай, когда длина кода  $n$  представима в виде  $2^l - 1$ . В этом случае корнями многочлена  $x^n + 1$  являются все ненулевые элементы поля  $\mathbb{F}_2^l$ , а порождающими многочленами циклического кода могут быть только минимальные многочлены для некоторой совокупности элементов  $\mathbb{F}_2^l$ . Коды БЧХ являются подмножеством циклических кодов, которые существенно используют указанную связь между порождающим полиномом и элементами из  $\mathbb{F}_2^l$ . Прямым следствием такого подхода является то, что длины кодов БЧХ перестают быть произвольными<sup>8</sup>.

Итак, пусть  $n = 2^l - 1$ ,  $d = 2t + 1 \leq n$ . Тогда *кодом БЧХ* называется  $(n, k, d)$ -линейный циклический код, в котором порождающий многочлен  $g(x)$  определяется как минимальный многочлен для элементов  $\alpha, \alpha^2, \dots, \alpha^{d-1}$  из поля  $\mathbb{F}_2^l$ , где  $\alpha$  – произвольный примитивный элемент поля  $\mathbb{F}_2^l$ , т.е.

$$g(x) = \text{НОК}(m_\alpha(x), m_{\alpha^2}(x), \dots, m_{\alpha^{d-1}}(x)).$$

Набор элементов  $\alpha, \alpha^2, \dots, \alpha^{d-1}$  называется *нулями БЧХ-кода*. Можно показать, что минимальное кодовое расстояние кода БЧХ не меньше, чем величина  $d$ . В результате БЧХ-коды по построению способны исправлять не менее  $t$  ошибок, где  $t = \lfloor (d-1)/2 \rfloor$ .

С учетом того, что все кодовые слова циклического кода делятся на  $g(x)$ , а  $g(x)$  является минимальным полиномом для выбранных нулей кода, то

$$v(x) \in C \Leftrightarrow v(\alpha^i) = 0 \quad \forall i = 1, \dots, 2t.$$

Назовём синдромами принятого сообщения  $w(x)$  значения  $w(x)$  в нулях кода:  $s_i = w(\alpha^i)$ ,  $i = 1, \dots, 2t$ . Все синдромы равны нулю тогда и только тогда, когда  $w(x)$  является кодовым словом.

**Пример 4.** Рассмотрим БЧХ-коды для случая поля  $\mathbb{F}_2^3 = \mathbb{F}_2[x]/(x^3 + x + 1)$ . Здесь  $l = 3$ ,  $n = 7$ . Полином  $x^3 + x + 1$  является примитивным полиномом над  $\mathbb{F}_2$ . Обозначим через  $\alpha \in \mathbb{F}_2^3$  его произвольный корень.

Построим код БЧХ, исправляющий не менее, чем  $t = 1$  ошибку. Его порождающий полином  $g(x)$  строится как минимальный многочлен для элементов  $\alpha, \alpha^2$ . Эти элементы входят в один смежный класс  $(\alpha, \alpha^2, \alpha^4)$ . Поэтому  $g(x) = m_\alpha(x) = m_{\alpha^2}(x) = x^3 + x + 1$ . В результате получаем  $(7, 4, 3)$ -код, являющийся кодом Хэмминга.

Построим код БЧХ, исправляющий не менее, чем  $t = 2$  ошибок. Его порождающий полином  $g(x)$  строится как минимальный многочлен для элементов  $\alpha, \alpha^2, \alpha^3, \alpha^4$ . Эти элементы входят в два смежных класса  $(\alpha, \alpha^2, \alpha^4)$  и  $(\alpha^3, \alpha^6, \alpha^5)$ . Поэтому  $g(x) = m_\alpha(x)m_{\alpha^3}(x)$ . Найдем минимальный многочлен  $m_{\alpha^3}(x)$  по его корням

$$m_{\alpha^3}(x) = (x + \alpha^3)(x + \alpha^6)(x + \alpha^5) = x^3 + (\alpha^3 + \alpha^6 + \alpha^5)x^2 + (\alpha^9 + \alpha^8 + \alpha^{11})x + \alpha^{14}.$$

Все коэффициенты многочлена  $m_{\alpha^3}(x)$  принадлежат  $\mathbb{F}_2$ . Вычислим эти коэффициенты, используя свойства  $\alpha^7 = 1$  (т.к.  $\alpha$  является примитивным элементом) и  $\alpha^3 = \alpha + 1$  (т.к.  $\alpha$  является корнем  $x^3 + x + 1$ ):

$$\alpha^3 + \alpha^6 + \alpha^5 = \alpha + 1 + (\alpha + 1)^2 + \alpha^2(\alpha + 1) = \alpha + 1 + \alpha^2 + 1 + \alpha^3 + \alpha^2 = 1,$$

$$\alpha^9 + \alpha^8 + \alpha^{11} = \alpha^2 + \alpha + \alpha^4 = \alpha^2 + \alpha + \alpha(\alpha + 1) = 0,$$

$$\alpha^{14} = 1.$$

Таким образом,  $m_{\alpha^3}(x) = x^3 + x^2 + 1$ ,  $g(x) = m_\alpha(x)m_{\alpha^3}(x) = (x^3 + x + 1)(x^3 + x^2 + 1) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$ .

Заметим, что многочлен  $g(x)$  имеет корни  $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6$ . Следовательно, построен код БЧХ  $(7, 1, 7)$ , который исправляет не менее, чем  $t = 3$  ошибки. Нетрудно видеть, что данный код содержит всего два кодовых слова  $\mathbf{v}_1 = [0000000]$ ,  $\mathbf{v}_2 = [1111111]$ .

**Декодирование кода Хэмминга.** Код Хэмминга является частным случаем кода БЧХ для  $t = 1$ . Поэтому нулями кода Хэмминга являются  $\alpha$  и  $\alpha^2$ , где  $\alpha$  – примитивный элемент поля  $\mathbb{F}_2^l$ . Для декодирования принятого слова  $w(x)$  вычислим синдром  $s_1 = w(\alpha)$ . Синдром  $s_2$  автоматически определяется по  $s_1$  как  $s_2 = w(\alpha^2) = (w(\alpha))^2 = s_1^2$ . Если  $s_1 = 0$ , то  $w(x)$  уже является кодовым словом и  $\hat{v}(x) = w(x)$ . Предположим, что при передаче по каналу была совершена одна ошибка. Тогда  $w(x) = v(x) + e(x)$ , где полином ошибок  $e(x) = x^j$  для некоторого  $j$ . Для нахождения позиции ошибки вычислим значения полинома ошибок в точке  $\alpha$  для всех  $j = 1, \dots, n$  (линейная сложность по  $n$ ). Если  $e(\alpha) = \alpha^j = s_1$ , то  $j$  – искомая позиция ошибки и  $\hat{v}(x) = w(x) + x^j$ . Если  $\alpha^j \neq s_1 \quad \forall j$ , то при передаче было совершено более одной ошибки, и процедура декодирования выдает отказ.

**Пример 5.** Рассмотрим код Хэмминга из примера 4. Он имеет порождающий полином  $g(x) = x^3 + x + 1$ , а все вычисления проводятся в поле  $\mathbb{F}_2^3 = \mathbb{F}_2[x]/(x^3 + x + 1)$ . Пусть входной полином  $u(x) = x^3 + x^2$ . Его систематическое кодирование было найдено в примере 3:  $v(x) = x^6 + x^5 + x$ . Пусть при передаче произошла

<sup>8</sup>Простейшим способом обойти указанное ограничение на длину кода является рассмотрение т.н. укороченных кодов БЧХ. Пусть  $n, k$  – необходимые параметры кода,  $n_1$  – некоторое число такое, что  $n_1 \geq n$ ,  $n_1 = 2^l - 1$ ,  $k_1 = n_1 - n + k$ . Тогда можно рассмотреть  $(n_1, k_1)$ -код БЧХ, в котором на вход подавать блоки длины  $k$ , дополненные нулями до длины  $k_1$ .

ошибка в позиции 5, т.е.  $w(x) = x^6 + x$ , а истинный  $e(x) = x^5$ . Для декодирования сначала найдем синдром  $s_1 = w(\alpha) = \alpha^6 + \alpha = (\alpha^3)^2 + \alpha = (\alpha + 1)^2 + \alpha = \alpha^2 + \alpha + 1$ . Теперь вычислим  $\alpha^j$  для всех  $j = 0, \dots, 6$ :

$$\begin{aligned} \alpha^0 &= 1, \\ \alpha^1 &= \alpha, \\ \alpha^2 &= \alpha^2, \\ \alpha^3 &= \alpha + 1, \\ \alpha^4 &= \alpha(\alpha + 1) = \alpha^2 + \alpha, \\ \alpha^5 &= \alpha^2(\alpha + 1) = \alpha^3 + \alpha^2 = \alpha^2 + \alpha + 1, \\ \alpha^6 &= (\alpha + 1)^2 = \alpha^2 + 1. \end{aligned}$$

Отсюда находим, что  $\alpha^5 = s_1$ , т.е. ошибка произошла в позиции 5. Таким образом,  $\hat{v}(x) = w(x) + x^5 = x^6 + x^5 + 1$ .

**Декодирование БЧХ кода.** Пусть при передаче по каналу произошло  $\nu$  ошибок, т.е.  $e(x) = x^{j_1} + x^{j_2} + \dots + x^{j_\nu}$ , где  $j_1, \dots, j_\nu$  – позиции ошибок. Запишем свойство  $s_i = w(\alpha^i) = e(\alpha^i)$ ,  $i = 1, \dots, 2t$  в виде системы:

$$\begin{aligned} s_1 &= \alpha^{j_1} + \alpha^{j_2} + \dots + \alpha^{j_\nu}, \\ s_2 &= (\alpha^{j_1})^2 + (\alpha^{j_2})^2 + \dots + (\alpha^{j_\nu})^2, \\ &\dots\dots\dots \\ s_{2t} &= (\alpha^{j_1})^{2t} + \dots + (\alpha^{j_\nu})^{2t}. \end{aligned} \tag{3}$$

Любое решение данной системы  $j_1, \dots, j_\nu$  для некоторого  $\nu \leq t$  будет искомым (такое решение всегда существует и единственно, т.к. БЧХ код по построению способен исправлять до  $t$  ошибок).

Введём обозначение  $\beta_i = \alpha^{j_i}$ ,  $i = 1, \dots, \nu$ . Тогда систему (3) можно записать как

$$\begin{aligned} s_1 &= \beta_1 + \beta_2 + \dots + \beta_\nu, \\ s_2 &= \beta_1^2 + \beta_2^2 + \dots + \beta_\nu^2, \\ &\dots\dots\dots \\ s_{2t} &= \beta_1^{2t} + \beta_2^{2t} + \dots + \beta_\nu^{2t}. \end{aligned} \tag{4}$$

Рассмотрим *полином локаторов ошибок*

$$\sigma(x) = \prod_{i=1}^{\nu} (1 + \beta_i x) = 1 + \sigma_1 x + \sigma_2 x^2 + \dots + \sigma_\nu x^\nu.$$

Корнями данного полинома являются  $\beta_i^{-1} = \alpha^{-j_i}$ . Связь между коэффициентами полинома  $\sigma_k$  и элементами  $\beta_i$  можно определить по теореме Виета:

$$\begin{aligned} \sigma_1 &= \beta_1 + \beta_2 + \dots + \beta_\nu, \\ \sigma_2 &= \beta_1\beta_2 + \beta_2\beta_3 + \beta_1\beta_3 + \dots + \beta_{\nu-1}\beta_\nu, \\ \sigma_3 &= \sum_{i_1 < i_2 < i_3} \beta_{i_1}\beta_{i_2}\beta_{i_3}, \\ &\dots\dots\dots \\ \sigma_\nu &= \beta_1\beta_2 \dots \beta_\nu. \end{aligned} \tag{5}$$

Система (4) определяет значения симметрического многочлена суммы  $k$ -ых степеней  $\beta_i$  для всех  $k = 1, \dots, 2t$ . Система (5) определяет значения элементарного симметрического многочлена. Соотношения между этими двумя типами симметрических многочленов задаются тождествами Ньютона-Жирара, которые с учётом двоичной арифметики могут быть записаны как





**Пример 6.** Рассмотрим (15,5,7) БЧХ код, исправляющий три ошибки. Он имеет порождающий полином  $g(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$ , а все вычисления проводятся в поле  $\mathbb{F}_2^4 = \mathbb{F}_2[x]/(x^4 + x + 1)$ . Пусть входной полином имеет вид  $u(x) = x^4 + x^2 + x$ . При систематическом кодировании ему соответствует кодový полином  $v(x) = x^{14} + x^{12} + x^{11} + x^8 + x^4 + x^3 + x^2 + x$ . Пусть полином ошибок равен  $e(x) = x^{12} + x^6 + 1$ . Следовательно, принятое слово  $w(x) = x^{14} + x^{11} + x^8 + x^6 + x^4 + x^3 + x^2 + x + 1$ .

Для удобства последующих вычислений построим для всех элементов поля  $\mathbb{F}_2^4$  таблицу соответствий между степенным представлением и полиномиальным (см. таблицу 1).

Таблица 1: Таблица вычислений для поля  $\mathbb{F}_2^4 = \mathbb{F}_2[x]/(x^4 + x + 1)$ .

$\alpha$	$\alpha$
$\alpha^2$	$\alpha^2$
$\alpha^3$	$\alpha^3$
$\alpha^4$	$\alpha + 1$
$\alpha^5$	$\alpha^2 + \alpha$
$\alpha^6$	$\alpha^3 + \alpha^2$
$\alpha^7$	$\alpha^3 + \alpha + 1$
$\alpha^8$	$\alpha^2 + 1$
$\alpha^9$	$\alpha^3 + \alpha$
$\alpha^{10}$	$\alpha^2 + \alpha + 1$
$\alpha^{11}$	$\alpha^3 + \alpha^2 + \alpha$
$\alpha^{12}$	$\alpha^3 + \alpha^2 + \alpha + 1$
$\alpha^{13}$	$\alpha^3 + \alpha^2 + 1$
$\alpha^{14}$	$\alpha^3 + 1$
$\alpha^{15}$	1

С использованием таблицы 1 найдём синдромы для принятого слова:

$$s_1 = w(\alpha) = \alpha^3 + 1 + \alpha^3 + \alpha^2 + \alpha + \alpha^2 + 1 + \alpha^3 + \alpha^2 + \alpha + 1 + \alpha^3 + \alpha^2 + \alpha + 1 = \alpha,$$

$$s_2 = w(\alpha^2) = (w(\alpha))^2 = \alpha^2,$$

$$\begin{aligned} s_3 = w(\alpha^3) &= \alpha^{42} + \alpha^{33} + \alpha^{24} + \alpha^{18} + \alpha^{12} + \alpha^9 + \alpha^6 + \alpha^3 + 1 = \alpha^{12} + \alpha^3 + \alpha^9 + \alpha^3 + \alpha^{12} + \alpha^9 + \alpha^6 + \alpha^3 + 1 = \\ &= \alpha^6 + \alpha^3 + 1 = \alpha^3 + \alpha^2 + \alpha^3 + 1 = \alpha^2 + 1 = \alpha^8, \end{aligned}$$

$$s_4 = w(\alpha^4) = (w(\alpha^2))^2 = \alpha^4,$$

$$s_5 = w(\alpha^5) = \alpha^{70} + \alpha^{55} + \alpha^{40} + \alpha^{30} + \alpha^{20} + \alpha^{15} + \alpha^{10} + \alpha^5 + 1 = \alpha^{10} + \alpha^{10} + \alpha^{10} + 1 + \alpha^5 + 1 + \alpha^{10} + \alpha^5 + 1 = 1,$$

$$s_6 = w(\alpha^6) = (w(\alpha^3))^2 = (\alpha^2 + 1)^2 = \alpha^4 + 1 = \alpha.$$

Таким образом, синдромный полином  $s(x) = \alpha x^6 + x^5 + \alpha^4 x^4 + \alpha^8 x^3 + \alpha^2 x^2 + \alpha x + 1$ .

Теперь решим уравнение  $x^7 a(x) + s(x)\sigma(x) = \lambda(x)$  расширенным алгоритмом Евклида.

**Шаг 0.**  $r_{-2}(x) = x^7,$   
 $r_{-1}(x) = s(x) = \alpha x^6 + x^5 + \alpha^4 x^4 + \alpha^8 x^3 + \alpha^2 x^2 + \alpha x + 1,$   
 $y_{-2}(x) = 0,$   
 $y_{-1}(x) = 1.$

**Шаг 1.**  $r_{-2}(x) = r_{-1}(x)q_0(x) + r_0(x),$   
 $q_0(x) = \alpha^{14}x + \alpha^{13},$   
 $r_0(x) = \alpha^8 x^5 + \alpha^{12} x^4 + \alpha^{11} x^3 + \alpha^{13},$   
 $y_0(x) = y_{-2}(x) + y_{-1}(x)q_0(x) = q_0(x) = \alpha^{14}x + \alpha^{13}.$

**Шаг 2.**  $r_{-1}(x) = r_0(x)q_1(x) + r_1(x),$   
 $q_1(x) = \alpha^8 x + \alpha^2,$   
 $r_1(x) = \alpha^{14} x^4 + \alpha^3 x^3 + \alpha^2 x^2 + \alpha^{11} x,$   
 $y_1(x) = y_{-1}(x) + y_0(x)q_1(x) = \alpha^7 x^2 + \alpha^{11} x.$

**Шаг 3.**  $r_0(x) = r_1(x)q_2(x) + r_2(x),$   
 $q_2(x) = \alpha^9 x,$   
 $r_2(x) = \alpha^5 x + \alpha^{13},$   
 $y_2(x) = y_0(x) + y_1(x)q_2(x) = \alpha x^3 + \alpha^5 x^2 + \alpha^{14} x + \alpha^{13}.$

Это последний шаг алгоритма Евклида, т.к. текущий остаток  $r_2(x)$  имеет степень меньше, чем  $t = 3$ . Таким образом,  $\sigma(x) = y_2(x) = \alpha x^3 + \alpha^5 x^2 + \alpha^{14} x + \alpha^{13}$  и  $\nu = \deg \sigma(x) = 3$ .

Найдём корни  $\sigma(x)$  полным перебором:

$$\begin{aligned}
 \sigma(\alpha) &= \alpha^4 + \alpha^7 + 1 + \alpha^{13} = \alpha^2, \\
 \sigma(\alpha^2) &= \alpha^7 + \alpha^9 + \alpha + \alpha^{13} = \alpha^3 + \alpha^2 + \alpha, \\
 \sigma(\alpha^3) &= \alpha^{10} + \alpha^{11} + \alpha^2 + \alpha^{13} = 0, \\
 \sigma(\alpha^4) &= \alpha^{13} + \alpha^{13} + \alpha^3 + \alpha^{13} = \alpha^2 + 1, \\
 \sigma(\alpha^5) &= \alpha + 1 + \alpha^4 + \alpha^{13} = \alpha^{13}, \\
 \sigma(\alpha^6) &= \alpha^4 + \alpha^2 + \alpha^5 + \alpha^{13} = \alpha^3 + \alpha^2, \\
 \sigma(\alpha^7) &= \alpha^7 + \alpha^4 + \alpha^6 + \alpha^{13} = \alpha^3 + 1, \\
 \sigma(\alpha^8) &= \alpha^{10} + \alpha^6 + \alpha^7 + \alpha^{13} = \alpha^3 + \alpha^2 + 1, \\
 \sigma(\alpha^9) &= \alpha^{13} + \alpha^8 + \alpha^8 + \alpha^{13} = 0, \\
 \sigma(\alpha^{10}) &= \alpha + \alpha^{10} + \alpha^9 + \alpha^{13} = \alpha, \\
 \sigma(\alpha^{11}) &= \alpha^4 + \alpha^{12} + \alpha^{10} + \alpha^{13} = \alpha^2 + \alpha, \\
 \sigma(\alpha^{12}) &= \alpha^7 + \alpha^{14} + \alpha^{11} + \alpha^{13} = 1, \\
 \sigma(\alpha^{13}) &= \alpha^{10} + \alpha + \alpha^{12} + \alpha^{13} = \alpha^2 + \alpha + 1, \\
 \sigma(\alpha^{14}) &= \alpha^{13} + \alpha^3 + \alpha^{13} + \alpha^{13} = \alpha^2 + 1, \\
 \sigma(\alpha^{15}) &= \alpha + \alpha^5 + \alpha^{14} + \alpha^{13} = 0.
 \end{aligned}$$

По найденным корням  $\alpha^3, \alpha^9, \alpha^{15}$  легко вычислить позиции ошибок:  $j_1 = -3 \bmod 15 = 12$ ,  $j_2 = -9 \bmod 15 = 6$ ,  $j_3 = -15 \bmod 15 = 0$ . Значит,  $e(x) = x^{12} + x^6 + 1$  и  $\hat{v}(x) = w(x) + e(x) = x^{14} + x^{12} + x^{11} + x^8 + x^4 + x^3 + x^2 + x$ .