5. Коды Боуза-Чоудхури-Хоквингема

Корректирующие свойства циклических кодов могут быть определены на основе двух теорем [1].

Теорема 1. Для любых m и t существует циклический код длиной $n = 2^m - 1$, с кратностью гарантированно исправляемой ошибки t и числом проверочных разрядов r не более mt [1].

Аналогично, если задана длина кода n, то можно определить необходимое для обеспечения требуемой корректирующей способности число проверочных разрядов.

Например, при заданных $n = 31$ и $m = 5$ для различных t можно получить

$$t = 1, n - k = mt = 5 \cdot 1 = 5 \longrightarrow \text{Код}(31,26);$$
$$t = 2, n - k = mt = 5 \cdot 2 = 10 \longrightarrow \text{Код}(31,21);$$
$$t = 3, n - k = mt = 5 \cdot 3 = 15 \longrightarrow \text{Код}(31,16).$$

Теорема 2. Эта теорема позволяет определить порождающий полином для кода с заданными параметрами. Она часто называется *теоремой БЧХ*, поскольку она была доказана Хоквингемом в 1959 и независимо от него Боузом и Чоудхури в 1960 [1].

Если среди корней порождающего полинома $g(x)$ циклического кода длиной $n = 2^m - 1$ содержится $d_0 - 1$ последовательных степеней ε^i, ε^{i+1}, ..., $\varepsilon^{i+d_{\min}-2}$, то кодовое расстояние $d_{\min} \geq d_0$ [1].

Согласно этому принципу строятся коды Боуза-Чоудхури-Хоквингема (коды БЧХ).

Перед рассмотрением кодов БЧХ примем ряд обозначений. Пусть ε — примитивный элемент поля $\text{GF}(2^m)$, а $m_i(x)$ — минимальный многочлен элемента ε^i [1].

Примитивным кодом БЧХ называется циклический код, порождающий полином $g(x)$ которого равен наименьшему общему кратному минимальных многочленов $m_1(x)$, ..., $m_{2t}(x)$

$$g(x) = \text{НОК}(m_1(x), \ldots, m_{2t}(x)).$$

В таком случае элементы

$$\varepsilon, \varepsilon^2, \ldots, \varepsilon^{2t} \tag{5.1}$$

являются корнями $g(x)$. То есть, $g(\varepsilon^j = 0)$ для $j = 1,2,\ldots,2t$. Тогда по теореме БЧХ кодовое расстояние этого кода $d_{\min} \geq (2t+1)$ [1].

Таким образом, для любого числа $n = 2^m - 1$ и любого $t < 2^{m-1}$ определен примитивный двоичный код БЧХ длиной n, с кодовым расстоянием $d_{\min} \geq (2t+1)$ и числом проверочных символов меньшим или равным mt [1].

Нижнюю границу для минимального кодового расстояния можно увеличить до $2t+2$. Для этого в число корней порождающего многочлена нужно включить единицу, минимальный многочлен для которой равен $x+1$. Таким образом, код БЧХ с порождающим многочленом $g_1(x) = (x-1)g(x)$ имеет кодовое расстояние $d_{\min} \geq (2t+2)$ [1].

Для примера построим порождающий многочлен для кода БЧХ, исправляющего две ошибки ($t=2$), с $n=31$. Из первой теоремы следует, что этот код построен над полем Галуа GF(2^5) ($m=5$) и имеет $r = mt = 5 \cdot 2 = 10$ проверочных символов, откуда следует, что это код (31,21). Будем считать, что образующий полином поля Галуа равен $p(x) = x^5 + x^2 + 1$.

Из теоремы БЧХ следует, что порождающий полином $g_{(31,21)}(x)$ этого кода должен иметь корни ε, ε^2, ε^3, ε^4. С другой стороны, из свойств циклического кода следует, что полином $g_{(31,21)}(x)$ должен быть делителем многочлена $x^n + 1 = x^{31} + 1$. Теперь необходимо определить минимальные многочлены, соответствующие корням полинома $g_{(31,21)}(x)$. Для этого проведем разложение полинома $(x^{31}+1)$ на множители, которые и будут являться минимальными многочленами. Вначале требуется определить корни каждого из этих минимальных многочленов. Механизм разложения представлен формулой (5.2).

$$x^{31}+1 = \underbrace{m_1(x)}_{\substack{\varepsilon \\ \varepsilon^2 \\ \varepsilon^4 \\ \varepsilon^8 \\ \varepsilon^{16}}} \cdot \underbrace{m_2(x)}_{\substack{\varepsilon^3 \\ \varepsilon^6 \\ \varepsilon^{12} \\ \varepsilon^{24} \\ \varepsilon^{48}=\varepsilon^{17}}} \cdot \underbrace{m_3(x)}_{\substack{\varepsilon^5 \\ \varepsilon^{10} \\ \varepsilon^{20} \\ \varepsilon^{40}=\varepsilon^{9} \\ \varepsilon^{18}}} \cdot \underbrace{m_4(x)}_{\substack{\varepsilon^7 \\ \varepsilon^{14} \\ \varepsilon^{28} \\ \varepsilon^{56}=\varepsilon^{25} \\ \varepsilon^{50}=\varepsilon^{19}}} \cdot \underbrace{m_5(x)}_{\substack{\varepsilon^{11} \\ \varepsilon^{22} \\ \varepsilon^{44}=\varepsilon^{13} \\ \varepsilon^{26} \\ \varepsilon^{52}=\varepsilon^{21}}} \cdot \underbrace{m_6(x)}_{\substack{\varepsilon^{15} \\ \varepsilon^{30} \\ \varepsilon^{60}=\varepsilon^{29} \\ \varepsilon^{58}=\varepsilon^{27} \\ \varepsilon^{54}=\varepsilon^{23}}} \cdot \underbrace{m_7(x)}_{\varepsilon^0} \qquad (5.2)$$

Теперь рассчитаем эти многочены:

$$\begin{aligned}
&m_1(x) = (x+\varepsilon)(x+\varepsilon^2)(x+\varepsilon^4)(x+\varepsilon^8)(x+\varepsilon^{16}) = 1+x^2+x^5;\\
&m_2(x) = (x+\varepsilon^3)(x+\varepsilon^6)(x+\varepsilon^{12})(x+\varepsilon^{24})(x+\varepsilon^{17}) = 1+x^2+x^3+x^4+x^5;\\
&m_3(x) = (x+\varepsilon^5)(x+\varepsilon^{10})(x+\varepsilon^{20})(x+\varepsilon^{9})(x+\varepsilon^{18}) = 1+x+x^2+x^4+x^5;\\
&m_4(x) = (x+\varepsilon^7)(x+\varepsilon^{14})(x+\varepsilon^{28})(x+\varepsilon^{25})(x+\varepsilon^{19}) = 1+x+x^2+x^3+x^5;\\
&m_5(x) = (x+\varepsilon^{11})(x+\varepsilon^{22})(x+\varepsilon^{13})(x+\varepsilon^{26})(x+\varepsilon^{21}) = 1+x+x^3+x^4+x^5;\\
&m_6(x) = (x+\varepsilon^{15})(x+\varepsilon^{30})(x+\varepsilon^{29})(x+\varepsilon^{27})(x+\varepsilon^{23}) = 1+x^3+x^5;\\
&m_7(x) = (x+\varepsilon^0) = 1+x.
\end{aligned} \qquad (5.3)$$

Из формул (5.2) и (5.3) видно, что корни ε, ε^2 и ε^4 принадлежат полиному $m_1(x) = 1+x^2+x^5$, а корень ε^3 — полиному $m_2(x) = 1+x^2+x^3+x^4+x^5$, следовательно, образующий полином рассматриваемого кода БЧХ (31,21) равен наименьшему общему кратному этих многочленов, а поскольку они являются различными неприводимыми полиномами, их наименьшее общее кратное

равно их произведению[1] [1].

$$g_{(31,21)}(x) = m_1(x)m_2(x) \begin{array}{l} = (1+x^2+x^5)(1+x^2+x^3+x^4+x^5) = \\ = 1+x^3+x^5+x^6+x^8+x^9+x^{10}. \end{array} \tag{5.4}$$

5.1. Кодирование кода БЧХ

Механизм кодирования кодом БЧХ полностью аналогичен кодированию обычным циклическим кодом. В случае несистематического кода используется умножение на образующий полином, а в случае систематического кода — поиск остатка от деления $x^r u(x)$, где $u(x)$ — информационный полином, на образующий полином $g(x)$.

5.2. Декодирование кода БЧХ

Основным алгоритмом декодирования кодов БЧХ является так называетмый *алгебраический* или *синдромный* метод декодирования. Принципом, лежащим в основе этого метода, является использование элементов поля Галуа для нумерации позиций элементов принятого кодового слова $r(x)$ (см. формулу (5.5)) [6].

$$\begin{array}{rcccc} \text{Значения:} & [r_0 & r_1 & \ldots & r_{n-1}] \\ \text{Локаторы позиций:} & 1 & \varepsilon & \ldots & \varepsilon^{n-1} \end{array} \tag{5.5}$$

Для определения позиций ошибок необходимо решить систему уравнений над полем Галуа $\mathrm{GF}(2^m)$, использованным для построения кода [6]. Далее рассмотрим, как получается эта система уравнений.

Для этого рассмотрим полином ошибок $e(x)$, представленный как

$$e(x) = e_{j_1}x^{j_1} + e_{j_2}x^{j_2} + \ldots + e_{j_q}x^{j_q}, \tag{5.6}$$

где $q \leq t_{\text{испр.}}$ — число ошибок в принятом слове $r(x)$ [6].

Множество

$$\{e_{j_1}, e_{j_2}, \ldots, e_{j_q}\}, \quad e_j \in \{0,1\} \tag{5.7}$$

называется множеством *значений ошибок*, а соответствующее ему множество

$$\{\varepsilon^{j_1}, \varepsilon^{j_2}, \ldots, \varepsilon^{j_q}\}, \quad \varepsilon^j \in \mathrm{GF}(2^m) \tag{5.8}$$

называется множеством *локаторов ошибок* [6].

Синдромы при декодировании кода БЧХ определяются как значения принятого полинома $r(x)$ в *нулях кода*, под которыми понимаются $2t$ корней

[1] Аналогично тому, как НОК двух простых чисел равно их произведению [1].

с последовательными степенями (5.1) [6].

$$\begin{aligned} S_1 &= r(\varepsilon) \equiv e_{j_1}(\varepsilon)^{j_1} + e_{j_2}(\varepsilon)^{j_2} + \ldots + e_{j_q}(\varepsilon)^{j_q}, \\ S_2 &= r(\varepsilon^2) \equiv e_{j_1}(\varepsilon^2)^{j_1} + e_{j_2}(\varepsilon^2)^{j_2} + \ldots + e_{j_q}(\varepsilon^2)^{j_q}, \\ &\ldots \\ S_{2t} &= r(\varepsilon^{2t}) \equiv e_{j_1}(\varepsilon^{2t})^{j_1} + e_{j_2}(\varepsilon^{2t})^{j_2} + \ldots + e_{j_q}(\varepsilon^{2t})^{j_q}. \end{aligned} \tag{5.9}$$

Необходимо отметить, что для двоичных кодов верно следующее равенство.

$$S_{2i} = S_i^2. \tag{5.10}$$

Далее вводится *полином локаторов ошибок* $\sigma(x)$

$$\sigma(x) = \prod_{l=1}^{q}(1 + \varepsilon^{j_l}x) = 1 + \sigma_1 x + \sigma_2 x^2 + \ldots + \sigma_q x^q, \tag{5.11}$$

корни которого равны обратным величинам локаторов ошибок (5.8) [6].

Из приведенных формул и следует *ключевое уравнение*, в матричном виде показанное в формуле (5.12) [6].

$$\begin{bmatrix} S_{q+1} \\ S_{q+2} \\ \vdots \\ S_{2q} \end{bmatrix} = \begin{bmatrix} S_1 & S_2 & \cdots & S_q \\ S_2 & S_3 & \cdots & S_{q+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_q & S_{q+1} & \cdots & S_{2q-1} \end{bmatrix} \begin{bmatrix} \sigma_q \\ \sigma_{q-1} \\ \vdots \\ \sigma_1 \end{bmatrix}. \tag{5.12}$$

Известны три основных метода решения ключевого уравнения [6].

1. Алгоритм Берлекэмпа-Мэсси (BMA).
2. Алгоритм Евклида (EA).
3. Прямое решение или алгоритм Питерсона-Горенштейна-Цирлера (PGZ).

5.2.1. Общий алгоритм декодирования двоичных кодов БЧХ

Блок-схема декодера двоичных кодов БЧХ в общем виде показана на рис. 5.1. Каждый из блоков декодера реализует определенный шаг алгоритма декодирования [6].

1. Вычисление синдромов.
2. Определение коэффициентов полинома локаторов ошибок $\sigma(x)$, т. е. решение ключевого уравнения.
3. Нахождение позиций ошибок, как обратных значений корней $\sigma(x)$.
4. Исправление кодового слова.

Далее рассмотрим работу декодера на каждом из этапов декодирования.

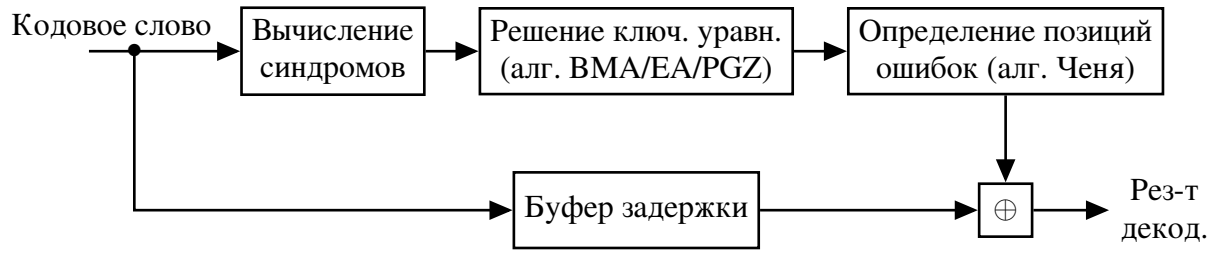

Рис. 5.1. Блок-схема декодера двоичных кодов БЧХ в общем виде

5.2.2. Вычисление синдромов

Поскольку формулы для вычисления синдромов (5.9) и (5.10) были приведены ранее, рассмотрим их вычисление на примере.

Для примера используем код БЧХ (15,7,5) над полем GF(2^4), исправляющий две ошибки с образующим полиномом $g_{(15,7)}(x)$, показанным на формуле (5.13).

$$g_{(15,7)}(x) = 1 + x^4 + x^6 + x^7 + x^8 \longrightarrow \mathbf{g}_{(15,7)} = \{1\,0\,0\,0\,1\,0\,1\,1\,1\}. \tag{5.13}$$

Согласно теореме БЧХ этот образующий полином будет иметь корни

$$\{\varepsilon, \varepsilon^2, \varepsilon^3, \varepsilon^4\}. \tag{5.14}$$

Выбранный для примера информационный полином $v_{(15,7)}(x)$ показан на формуле (5.15).

$$v_{(15,7)}(x) = x^2 + x^5 \longrightarrow \mathbf{v}_{(15,7)} = \{0\,0\,1\,0\,0\,1\,0\}. \tag{5.15}$$

После кодирования по систематическому алгоритму будет получено кодовое слово $u_{(15,7)}(x)$.

$$\begin{aligned} u_{(15,7)}(x) &= x + x^4 + x^7 + x^{10} + x^{13}, \\ \mathbf{u}_{(15,7)} &= \{0\,1\,0\,0\,1\,0\,0\,1\,0\,0\,1\,0\,0\,1\,0\}. \end{aligned} \tag{5.16}$$

Зададим полином ошибок $e_{(15,7)}(x)$.

$$\begin{aligned} e_{(15,7)}(x) &= x^3 + x^{10}, \\ \mathbf{e}_{(15,7)} &= \{0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\}. \end{aligned} \tag{5.17}$$

В таком случае на вход декодера будет принята комбинация $r_{(15,7)}(x)$.

$$\begin{aligned} r_{(15,7)}(x) &= x + x^3 + x^4 + x^7 + x^{13}, \\ \mathbf{r}_{(15,7)} &= \{0\,1\,0\,1\,1\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0\}. \end{aligned} \tag{5.18}$$

Теперь можно приступить к расчету синдромов по формулам (5.9) и (5.10).

$$\begin{aligned} S_1 &= r(\varepsilon) = \varepsilon + (\varepsilon)^3 + (\varepsilon)^4 + (\varepsilon)^7 + (\varepsilon)^{13} = \varepsilon^{12}, \\ S_2 &= S_1^2 = \varepsilon^9, \\ S_3 &= r(\varepsilon^3) = \varepsilon^3 + (\varepsilon^3)^3 + (\varepsilon^3)^4 + (\varepsilon^3)^7 + (\varepsilon^3)^{13} = \varepsilon^7, \\ S_4 &= S_2^2 = \varepsilon^3. \end{aligned} \tag{5.19}$$

5.2.3. Решение ключевого уравнения по алгоритму Берлекэмпа-Мэсси

Алгоритм Берлекэмпа-Мэсси часто рассматривается как итеративный процесс построения минимального линейного регистра сдвига с обратной связью (ЛРОС), который генерирует известную последовательность синдромов (5.9) [6].

Основной целью алгоритма БМ является построение полинома $\sigma^{(i+1)}(x)$, удовлетворяющего уравнению (5.20), выводимому из (5.12).

$$\sum_{j=0}^{l_i+1} S_{k-j}\sigma_j^{(i+1)} = 0, \quad l_i < k < i+1. \tag{5.20}$$

Решение (5.20) эквивалентно тому, что полином

$$\sigma^{(i+1)}(x) = 1 + \sigma_1^{(i+1)}x + \ldots + \sigma_{l_{i+1}}^{(i+1)}x^{l_{i+1}} \tag{5.21}$$

является многочленом обратной связи ЛРОС, генерирующим последовательность синдромов (5.9) [6].

Для определения соответствия между синдромной последовательностью и генерируемой ЛРОС водится так называемая *несовместность* (*невязка*), определяемая на i-й итерации как

$$d_i = S_{i+1} + S_i\sigma_1^{(i)} + \ldots + S_{i-l_i+1}\sigma_{l_i}^{(i)} \tag{5.22}$$

и содержащая корректирующий множитель для вычисления $\sigma^{(i+1)}$ на следующей итерации [6].

При вычислении значения невязки возможны два варианта.

- $d_i = 0$. В этом случае уравнение (5.20) удовлетворяется с результатом

$$\sigma^{(i+1)}(x) = \sigma^{(i)}(x), \quad l_{i+1} = l_i. \tag{5.23}$$

- $d_i \neq 0$. В таком случае решение на следующей итерации имеет вид

$$\begin{aligned} &\sigma^{(i+1)}(x) = \sigma^{(i)}(x) + d_i d_m^{-1} x^{i-m}\sigma^{(m)}(x), \\ &l_{i+1} = \max\{l_i, l_m + i - m\}, \end{aligned} \tag{5.24}$$

  где $\sigma^{(m)}(x)$ — решение на m-й итерации такое, что $-1 \leq m < i$, $d \neq 0$ и $(m - l_m)$ максимально [6].

Расчет $\sigma^{(i+1)}(x)$ начинается с $i=0$ и продолжается пока не будут выполнены одно или оба условия (5.25) [6].

$$i \geq l_{i+1} + t - 1 \quad \text{или} \quad i = 2t - 1. \tag{5.25}$$

Начальные условия алгоритма показаны в формуле (5.26) [6].

$$\begin{aligned} &\sigma^{(-1)}(x) = 1, \quad l_{-1} = 0, \quad d_{-1} = 1, \\ &\sigma^{(0)}(x) = 1, \quad l_0 = 0, \quad d_0 = S_1. \end{aligned} \tag{5.26}$$

Далее рассмотрим процесс решения ключевого уравнения на примере.

- Итерация 0. Начальные условия.

$$\begin{aligned} &\sigma^{(-1)}(x) = 1, \quad l_{-1} = 0, \quad d_{-1} = 1, \\ &\sigma^{(0)}(x) = 1, \quad l_0 = 0, \quad d_0 = S_1 = \varepsilon^{12}. \end{aligned}$$

- Итерация 1.

$$\begin{aligned} &i = 0, d_0 = \varepsilon^{12} \neq 0, m = -1 = \arg(\max(-1+0) = -1) \text{ для } d_{-1} \neq 0. \\ &\sigma^{(1)}(x) = \sigma^{(0)}(x) + d_0 d_{-1}^{-1} x^{(0-(-1))} \sigma^{(-1)}(x) = 1 + \varepsilon^{12} x, \\ &l_1 = \max\{l_0, l_{-1} + 0 - (-1)\} = 1, \\ &l_1 + 2 - 1 \leq 0 \text{ ? Нет:} \\ &d_1 = S_2 + S_1 \sigma_1^{(1)} = \varepsilon^9 + \varepsilon^{12}\varepsilon^{12} = 0. \end{aligned}$$

- Итерация 2.

$$\begin{aligned} &i = 1, d_1 = 0, \\ &\sigma^{(2)}(x) = \sigma^{(1)}(x) = 1 + \varepsilon^{12} x, \\ &l_2 = l_1 = 1, \\ &l_2 + 2 - 1 \leq 1 \text{ ? Нет:} \\ &d_2 = S_3 + S_2 \sigma_1^{(1)} = \varepsilon^7 + \varepsilon^9 \varepsilon^{12} = \varepsilon^{10}. \end{aligned}$$

- Итерация 3.

$$\begin{aligned} &i = 2, d_2 = \varepsilon^{10} \neq 0, m = 0 = \arg(\max(0-0) = 0) \text{ для } d_0 \neq 0. \\ &\sigma^{(3)}(x) = \sigma^{(2)}(x) + d_2 d_0^{-1} x^{(2-0)} \sigma^{(0)}(x) = \\ &= (1 + \varepsilon^{12} x) + \varepsilon^{10}\varepsilon^{-12} x^2 = 1 + \varepsilon^{12} x + \varepsilon^{13} x^2, \\ &l_3 = \max\{l_2, l_0 + 2 - (0)\} = 2, \\ &l_3 + 2 - 1 \leq 2 \text{ ? Нет:} \\ &d_3 = S_4 + S_3 \sigma_1^{(3)} + S_2 \sigma_2^{(3)} = \varepsilon^3 + \varepsilon^7 \varepsilon^{12} + \varepsilon^9 \varepsilon^{13} = 0. \end{aligned}$$

- Итерация 4.

$$\begin{aligned} &i = 3, d_3 = 0, \\ &\sigma^{(4)}(x) = \sigma^{(3)}(x) = 1 + \varepsilon^{12} x + \varepsilon^{13} x^2, \\ &l_4 = l_3 = 2, \\ &l_4 + 2 - 1 \leq 3 \text{ ? Да: Конец.} \end{aligned}$$

Таким образом

$$\sigma^{bma}_{(15,7)}(x) = 1 + \varepsilon^{12}x + \varepsilon^{13}x^2. \tag{5.27}$$

Необходимо отметить, что тот факт, что на нечетных итерациях $d_i = 0$, является закономерностью для двоичных кодов БЧХ, что позволяет сократить алгоритм, выполняя только четные итерации и исправив правило остановки на

$$i \geq l_{i+1} + t - 2 \quad \text{или} \quad i = 2t - 1. \tag{5.28}$$

5.2.4. Решение ключевого уравнения по алгоритму Питерсона-Горенштейна-Цирлера

Согласно этому алгоритму для решения ключевого уравнения используется стандартный алгоритм решения системы линейных уравнений. При этом, поскольку неизвестно действительное число ошибок в принятом слове, приходится предварительно проверять гипотезу о том, действительное количество ошибок равно q, начиная с максимально возможного числа ошибок [6].

Декодер начинает работу с предположения о том, что возникло максимальное число ошибок $q = t$. Вычисляется определитель Δ_q для матрицы синдромов

$$\Delta_q = \det \begin{bmatrix} S_1 & S_2 & \cdots & S_q \\ S_2 & S_3 & \cdots & S_{q+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_q & S_{q+1} & \cdots & S_{2q-1} \end{bmatrix} \tag{5.29}$$

и сравнивается с нулем. Если $\Delta_q = 0$, то действительное число ошибок меньше, чем предполагалось. Тогда q уменьшается на 1 и снова проверяется определитель. Процедура при необходимости повторяется, пока $q > 1$. Как только оказывается, что$\Delta_q \neq 0$, вычисляется обратная матрица для матрицы синдромов и вычисляются значения $\sigma_1, \ldots, \sigma_q$. Если $\Delta_q = 0$ для всех q от 1 до t, декодирование считается безуспешным и регистрируется обнаружение неисправляемой комбинации ошибок [6].

Разберем процедуру на примере.

Предположим, что имеется $q = t = 2$ ошибки. В этом случае матрица синдромов будет равна

$$\mathbf{S}_2 = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_3 \end{bmatrix} = \begin{bmatrix} \varepsilon^{12} & \varepsilon^{9} \\ \varepsilon^{9} & \varepsilon^{7} \end{bmatrix}, \tag{5.30}$$

И определитель Δ_2 будет равен

$$\Delta_2 = \det(\mathbf{S}_2) = \begin{vmatrix} \varepsilon^{12} & \varepsilon^{9} \\ \varepsilon^{9} & \varepsilon^{7} \end{vmatrix} = \varepsilon^{12}\varepsilon^{7} + \varepsilon^{9}\varepsilon^{9} = \varepsilon^{7}. \tag{5.31}$$

Так как $\Delta_2 \neq 0$ подтверждается предположение о том, что в принятой комбинации две ошибки. Подставим полученные синдромы в ключевое уравнение (5.12) получим

$$\begin{bmatrix} S_1 & S_2 \\ S_2 & S_3 \end{bmatrix} \begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} S_3 \\ S_4 \end{bmatrix} \Leftrightarrow \begin{bmatrix} \varepsilon^{12} & \varepsilon^{9} \\ \varepsilon^{9} & \varepsilon^{7} \end{bmatrix} \begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \varepsilon^{7} \\ \varepsilon^{3} \end{bmatrix}. \tag{5.32}$$

Решая систему (5.32), получим

$$\begin{aligned} \sigma_2 &= \Delta_2^{-1} \cdot \begin{vmatrix} \varepsilon^7 & \varepsilon^9 \\ \varepsilon^3 & \varepsilon^7 \end{vmatrix} = (\varepsilon^7)^{-1} \cdot (\varepsilon^7 \varepsilon^7 + \varepsilon^9 \varepsilon^3) = \varepsilon^{13} \\ \sigma_1 &= \Delta_2^{-1} \cdot \begin{vmatrix} \varepsilon^{12} & \varepsilon^7 \\ \varepsilon^9 & \varepsilon^3 \end{vmatrix} = (\varepsilon^7)^{-1} \cdot (\varepsilon^{12} \varepsilon^3 + \varepsilon^7 \varepsilon^9) = \varepsilon^{12} \end{aligned} \tag{5.33}$$

Таким образом

$$\sigma_{(15,7)}^{pgz}(x) = 1 + \varepsilon^{12} x + \varepsilon^{13} x^2, \tag{5.34}$$

что совпадает с результатом (5.27), полученным по алгоритму Берлекэмпа-Мэсси.

5.2.5. *Решение ключевого уравнения по алгоритму Евклида*

В основе алгоритма Евклида лежит процедура нахождения наибольшего общего делителя двух полиномов [6].

Вводят *полином значений ошибок* как

$$\Lambda(x) = \sigma(x) S(x), \tag{5.35}$$

где *синдромный полином* $S(x)$ имеет вид

$$S(x) = 1 + S_1 x + \ldots + S_{2t} x^{2t}. \tag{5.36}$$

Из ключевого уравнения (5.12) следует, что

$$\Lambda(x) = \sigma(x) S(x) \bmod (x^{2t+1}). \tag{5.37}$$

Задача декодирования может быть переформулирована как задача определения полинома $\Lambda(x)$, удовлетворяющего уравнению (5.37). Для это применяют расширенный алгоритм Евклида к полиномам $q_0(x) = x^{2t+1}$ и $q_1(x) = S(x)$. Если на j-м шаге алгоритма получено такое решение

$$q_j(x) = a_j(x) x^{2t+1} + b_j(x) S(x),$$

что степень $q_j(x)$ меньше либо равна t ($\deg[q_j(x)] \leq t$, то $\Lambda(x) = q_j(x)$ и $\sigma(x) = b_j(x)$ [6].

Алгоритм Евклида для решения ключевого уравнения [6].

1. Начальные значения.

$$q_0(x) = x^{2t+1}, \quad q_1(x) = S(x)$$
$$b_0(x) = 0, \qquad b_1(x) = 1.$$

2. На шаге $j = 2$ поделить $q_{j-2}(x)$ на $q_{j-1}(x)$.

$$q_{j-2}(x) = f_j(x)q_{j-1}(x) + q_j(x), \quad 0 \le \deg[q_j(x)] < \deg[q_{j-1}(x)].$$

3. Вычислить

$$b_j(x) = b_{j-2}(x) + f_j(x)b_{j-1}(x).$$

4. Остановить вычисления при

$$\deg[q_j(x)] \le t.$$

Полином локаторов ошибок при этом равен

$$\sigma(x) = b_j(x).$$

Далее разберем процедуру на примере.

- Начальные условия.

$$q_0(x) = x^5,$$
$$q_1(x) = S(x) = 1 + \varepsilon^{12}x + \varepsilon^9x^2 + \varepsilon^7x^3 + \varepsilon^3x^4,$$
$$b_0(x) = 0, \qquad b_1(x) = 1.$$

- Шаг $j = 2$.

$$x^5 = (1 + \varepsilon^{12}x + \varepsilon^9x^2 + \varepsilon^7x^3 + \varepsilon^3x^4)(\varepsilon + \varepsilon^{12}x) + (1 + \varepsilon x + \varepsilon^{13}x^2 + \varepsilon^{14}x^3),$$
$$q_2(x) = 1 + \varepsilon x + \varepsilon^{13}x^2 + \varepsilon^{14}x^3,$$
$$f_2(x) = \varepsilon + \varepsilon^{12}x,$$
$$b_2(x) = b_0(x) + f_2(x)b_1(x) = 0 + (\varepsilon + \varepsilon^{12}x) \cdot 1 = \varepsilon + \varepsilon^{12}x,$$
$$\deg[q_2(x)] > t = 2.$$

- Шаг $j = 3$.

$$1 + \varepsilon^{12}x + \varepsilon^9x^2 + \varepsilon^7x^3 + \varepsilon^3x^4 = (1 + \varepsilon x + \varepsilon^{13}x^2 + \varepsilon^{14}x^3)(\varepsilon^{13} + \varepsilon^4x) +$$
$$+(\varepsilon^6 + \varepsilon^8x + \varepsilon x^2),$$
$$q_3(x) = \varepsilon^6 + \varepsilon^8x + \varepsilon x^2,$$
$$f_3(x) = \varepsilon^{13} + \varepsilon^4x,$$
$$b_3(x) = b_1(x) + f_3(x)b_2(x) = 1 + (\varepsilon^{13} + \varepsilon^4x)(\varepsilon + \varepsilon^{12}x) = \varepsilon^3 + x + \varepsilon x^2,$$
$$\deg[q_3(x)] \le t = 2 \longrightarrow \text{Завершение}.$$

Получаем, что полином локаторов ошибок равен

$$\sigma^{ea}_{(15,7)}(x) = \varepsilon^3 + x + \varepsilon x^2 = \varepsilon^3(1 + \varepsilon^{12}x + \varepsilon^{13}x^2), \tag{5.38}$$

что с точностью до постоянного множителя, не влияющего на значения корней $\sigma_{(15,7)}(x)$, совпадает с результатом (5.27), полученным согласно алгоритмам Берлекэмпа-Мэсси и Питерсона-Горенштейна-Цирлера.

5.2.6. Определение позиций ошибок по алгоритму Ченя

Для поиска корней $\sigma(x)$ на множестве локаторов позиций кодовых символов используется метод проб и ошибок, получивший название *метод Ченя*. Согласно этому методу, для всех ненулевых элементов поля $\beta \in \mathrm{GF}(2^m)$, проверяется условие $\sigma(\beta^{-1}) = 0$. Далее производится исправление ошибок, что для двоичных кодов БЧХ сводится к сложению позиции с ошибкой с единицей по модулю 2 [6].

Продолжая пример, подставим обратные значения всех возможных элементов поля $\mathrm{GF}(2^4)$ в $\sigma(x)$ и получим, что корни его равны $\varepsilon^5 = \varepsilon^{-10}$ и $\varepsilon^{12} = \varepsilon^{-3}$, что показывает наличие ошибок в третьем и десятом разрядах.

Таким образом, можно восстановить полином ошибок

$$e(x) = x^3 + x^{10}$$

и исправить неверно принятую комбинацию.