ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное государственное образовательное бюджетное учреждение высшего профессионального образования «САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ им. проф. М. А. БОНЧ-БРУЕВИЧА»

С. С. Владимиров

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ТЕОРИИ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ

Курс лекций

СПб ГУТ)))

Санкт-Петербург
2014

6. Циклические коды

Циклические коды являются подмножеством линейных кодов. Как и ранее, мы остановимся на рассмотрении двоичных циклических кодов.

Линейный (n,k)-код C называется *циклическим*, если циклический сдвиг любого кодового слова v из C также принадлежит коду C [33].

Циклический сдвиг кодового слова $v = (v_0, v_1, \ldots, v_{n-1})$ соответствует сдвигу всех элементов слова на одну позицию вправо, в результате чего будет получено кодовое слово $v^{(1)} = (v_{n-1}, v_0, v_1, \ldots, v_{n-2})$. В результате же i-кратного сдвига получается кодовое слово $v^{(i)} = (v_{n-i}, \ldots, v_{n-1}v_0, v_1, \ldots, v_{n-i-1})$ [33].

Аппаратно циклический сдвиг реализуется при помощи n-разрядного регистра сдвига с обратной связью (см. рис. 6.1) [33].

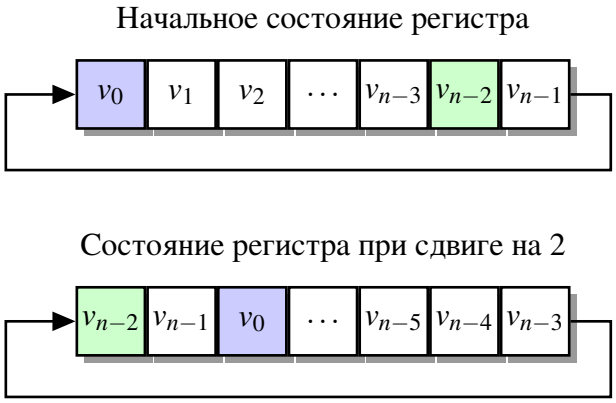

Рис. 6.1. Регистр сдвига с обратной связью

Кодовые слова циклического кода удобно представлять в виде многочленов над полем GF(2) [33]. Так кодовое слово $v = (v_0, v_1, \ldots, v_{n-1})$ можно представить полиномом

$$v(x) = v_0x^0 + v_1x^1 + \ldots + v_{n-1}x^{n-1} = v_0 + v_1x + \ldots + v_{n-1}x^{n-1}. \qquad (48)$$

Циклический сдвиг кодового слова v на i позиций также можно представить в виде полинома, начальную часть которого выделим и обозначим как $q(x)$ [33].

$$v^{(i)}(x) = \underbrace{v_{n-i} + v_{n-i+1}x + \ldots + v_{n-1}x^{i-1}}_{q(x)} + v_0x^i + v_1x^{i+1} + \ldots + v_{n-i-1}x^{n-1}. \qquad (49)$$

Если сравнить (49) с произведением $v(x)$ на x^i, можно увидеть, что в нем также присутствует $q(x)$ [33].

$$\begin{aligned} x^i \cdot v(x) \quad &= v_0x^i + v_1x^{i+1} + \ldots + v_{n-i-1}x^{n-1} + \\ &+ \underbrace{v_{n-i}x^n + \ldots + v_{n-1}x^{n+i-1}}_{q(x)\cdot x^n}. \end{aligned} \qquad (50)$$

Сравнивая эти выражения, можно составить равенство (51)

$$x^i \cdot v(x) = q(x) \cdot (x^n + 1) + v^{(i)}(x), \tag{51}$$

из которого следует, что *многочлен, соответствующий циклическому сдвигу вектора v на i позиций, можно вычислить как остаток от деления многочлена* $x^i v(x)$ *на* $(x^n + 1)$. Это свойство используется в процедуре эффективного обнаружения ошибок [33].

6.1. Порождающий многочлен циклического кода

Для построения циклического (n,k)-кода C используется *порождающий* (или *образующий*) многочлен $g(x)$ степени $r = n - k$

$$g(x) = g_0 + g_1 x + \ldots + g_r x^r. \tag{52}$$

Для порождающего многочлена кода верен ряд утверждений [33].

1. $g(x)$ является кодовым многочленом наименьшей степени r.
2. Коэффициент g_0 многочлена $g(x)$ всегда равен 1.
3. Полином $v(x)$ является кодовым тогда и только тогда, когда он кратен полиному $g(x)$.
4. Полином $g(x)$ делит $(x^n + 1)$ без остатка. Для этого утверждения верно обратное ему.
5. Если полином некоторый полином $g(x)$ степени $n - k$ делит $(x^n + 1)$ без остатка, то он порождает некоторый циклический (n,k)-код.

В качестве примера определим порождающий многочлен циклического кода $(7,4)$. Как следует из определения, он строится посредством порождающего многочлена $g(x)$ степени $r = n - k = 7 - 3 = 4$, являющегося делителем полинома $x^7 + 1$. В формуле показано разложение полинома $x^7 + 1$ на множители.

$$x^7 + 1 = (1 + x)(1 + x + x^3)(1 + x^2 + x^3). \tag{53}$$

Видно, что в разложении присутствуют два полинома степени 3. Каждый из них может быть использован для построения циклического кода.

6.2. Несистематическое кодирование циклического кода

Несистематическое кодирование циклического (n,k)-кода заключается в умножении информационного вектора $u(x)$ на образующий полином кода. Процесс кодирования можно представить формулой

$$v(x) = u(x) \cdot g(x). \tag{54}$$

Для примера рассмотрим кодирование вектора $u = (1\ 0\ 1\ 0)$ для циклического кода $(7,4)$, образованного полиномом $g(x) = 1 + x + x^3$. Вектор u

представим в виде полинома

$$u(x) = 1 + x^2.$$

Тогда кодовый полином $v(x)$ будет равен

$$v(x) = (1+x^2)(1+x+x^3) = 1+x+x^2+x^5 \longrightarrow v = (1\ 1\ 1\ 0\ 0\ 1\ 0).$$

Можно видеть, что в случае несистематического кодирования кодовое слово v не содержит символов информационного вектора f.

6.3. Систематические циклические коды

Информационный полином $u(x)$ для циклического (n,k)-кода имеет степень $k-1$

$$u(x) = u_0 + u_1 x + \ldots + u_{k-1}x^{k-1}, \tag{55}$$

из чего следует, что его $(r = n-k)$-кратный сдвиг

$$x^r \cdot u(x) = u_0 x^r + u_1 x^{r+1} + \ldots + u_{k-1}x^{n-1} \tag{56}$$

не приводит к переполнению n-разрядного регистра сдвига и соответствует заполнению k правых двоичных разрядов регистра информационным словом. Теперь, необходимо заполнить r левых разрядов так, чтобы полученный n-разрядный вектор принадлежал коду. Для этого необходимо представить полином $x^r u(x)$ как

$$x^r \cdot u(x) = a(x) \cdot g(x) + b(x), \tag{57}$$

где $b(x)$ — остаток от деления $x^r u(x)$ на $g(x)$ [33].

Из (57) следует, что

$$x^r \cdot u(x) + b(x) = a(x) \cdot g(x). \tag{58}$$

Таким образом можно составить алгоритм кодирования систематического циклического (n,k)-кода.

1. Умножить информационный полином $u(x)$ (степень $\leq (k-1)$) на x^r.
2. Найти остаток $b(x)$ (степень $\leq (r-1)$) от деления $x^r u(x)$ на $g(x)$.
3. Сложить $b(x)$ и $x^r u(x)$, в результате получив кодовый полином $v(x)$ (степень $\leq (n-1)$).

Из (58) следует, что полученный многочлен $v(x)$

$$\begin{aligned} v(x) \quad &= b(x) + x^r \cdot u(x) = \underbrace{b_0 + b_1 x + \ldots + b_{r-1}x^{r-1}}_{r \text{ проверочных символов}} + \\ &\underbrace{u_0 x^r + u_1 x^{r+1} + \ldots + u_{k-1}x^{n-1}}_{k \text{ информационных символов}} \end{aligned} \tag{59}$$

вляется кодовым поскольку делится на $g(x)$ без остатка, а сам код является систематическим, поскольку из (59) видно, что старшие k элементов кодового вектора являются информационным вектором (см. рис. 6.2) [33].

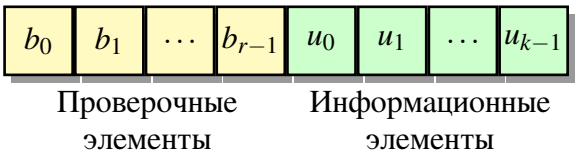

Рис. 6.2. Кодовое слово систематического циклического кода в регистре сдвига

Для примера, как и в случае несистематического кода, рассмотрим кодирование вектора $u = (1\ 0\ 1\ 0)$ для систематического циклического кода $(7,4)$, образованного полиномом $g(x) = 1 + x + x^3$. Вектор u представим в виде полинома

$$u(x) = 1 + x^2.$$

Умножаем информационный полином на x^3

$$x^3 u(x) = x^3 + x^5.$$

Определяем остаток от деления $x^3 u(x)$ на $g(x)$

$$x^3 u(x) = (x^2) g(x) + \underbrace{x^2}_{b(x)}.$$

Получаем кодовый полином

$$v(x) = x^3 u(x) + b(x) = x^2 + x^3 + x^5 \longrightarrow v = (0\ 0\ 1\ 1\ 0\ 1\ 0).$$

По аналогии с рис. 6.2 его можно представить в виде регистра (см. рис. 6.3).

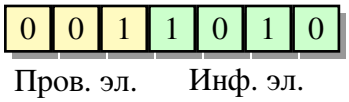

Рис. 6.3. Пример кодового слова систематического циклического кода $(7,4)$ в регистре сдвига

6.4. Порождающая матрица

Из свойств порождающего полинома циклического кода следует, что каждый кодовый многочлен может быть представлен произведением

$$v(x) = u(x) g(x) = u_0 g(x) + u_1 x g(x) + \ldots + u_{k-1} x^{k-1} g(x). \qquad (60)$$

Каждое слагаемое в (60) содержит сдвиг порождающего полинома $g(x)$, следовательно, кодовый вектор v, соответствующий полиному $v(x)$ можно представить как произведение информационного вектора u на порождающую матрицу G

$$v_{1\times n} = u_{1\times k} \odot G_{k\times n}. \tag{61}$$

Порождающая матрица при этом имеет вид (для несистематического кода)

$$G_{k\times n} = \begin{bmatrix} 1 & g_1 & \dots & g_{r-1} & g_r & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & g_{r-2} & g_{r-1} & g_r & 0 & \dots & 0 \\ 0 & 0 & \dots & g_{r-3} & g_{r-2} & g_{r-1} & g_r & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & g_1 & \dots & g_{r-1} & g_r \end{bmatrix}. \tag{62}$$

Для примера приведем порождающую матрицу несистематического циклического кода $(7,4)$ с порождающим многочленом $g(x) = 1 + x + x^3$.

$$G_{4\times 7} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}. \tag{63}$$

Путем элементарных матричных преобразований матрицу (63) можно привести к систематическому виду.

$$G^{syst}_{4\times 7} = (P_{4\times 3}\ I_4) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{64}$$

Кодовые векторы (полиномы), образующие порождающую матрицу систематического циклического кода называются *базисными*.

6.5. Проверочный полином и проверочная матрица

Для порождающего многочлена $g(x)$ степени $r = n - k$ циклического (n,k)-кода C существует *проверочный многочлен* $h(x)$ степени k такой, что $x^n + 1 = g(x)h(x)$ [33].

$$h(x) = h_0 + h_1 x + \ldots + h_k x^k. \tag{65}$$

Многочлен взаимообратный $h(x)$

$$x^k(h(x^{-1}) = h_k + h_{k-1}x + \ldots + h_0 x^k \tag{66}$$

является порождающим многочленом $(n, n-k)$-кода C_d, дуального коду C.

На основе проверочного полинома $h(x)$ строится проверочная матрица $H_{r\times n}$ вида

$$H_{r\times n} = \begin{bmatrix} h_k & h_{k-1} & \dots & h_1 & h_0 & 0 & 0 & \dots & 0 \\ 0 & h_k & \dots & h_2 & h_1 & h_0 & 0 & \dots & 0 \\ 0 & 0 & \dots & h_3 & h_2 & h_1 & h_0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & 0 & h_k & h_{k-1} & \dots & h_1 & h_0 \end{bmatrix}, \tag{67}$$

На основе которой строится уравнение для синдромного декодирования

$$v \odot H^T = 0. \tag{68}$$

Проверочную матрицу $H^{syst}_{(n-k)\times n}$ систематического циклического кода можно построить из порождающей матрицы $G^{syst}_{k\times n}$ по рассмотренным ранее правилам.

$$H_{r\times n} = (I_r \; P^T_{k\times r}) \tag{69}$$

6.6. Схемная реализация кодирования циклического кода

По МУ «Код (9, 5)».

6.7. Синдромное декодирование циклического кода

При передаче по каналу связи к кодовому полиному $v(x)$ добавляется полином ошибок $e(x)$, в результате чего полином полученного на приеме слова имеет вид

$$r(x) = v(x) + e(x) \tag{70}$$

или

$$r(x) = a(x)g(x) + s(x), \tag{71}$$

где $s(x)$ — *синдром* ошибки. Синдром кодового слова равен нулю. Обнаружение и исправление ошибок в систематических кодах может производиться только на основе анализа синдрома.

Как видно из формулы (71), синдром $s(x)$ равен остатку от деления принятого слова $r(x)$ на образующий полином $g(x)$.

Для обнаружения ошибки достаточно посчитать синдром принятого слова. Если он является ненулевым, то принятое слово содержит ошибки.

Схема вычисления синдрома представлена на рис. 6.4.

Для исправления ошибок используется тот факт, что каждый вектор ошибки (исправляемой кодом) имеет свой синдром, который может быть получен путем деления полинома ошибки $e(x)$ на порождающий полином кода $g(x)$. При этом синдром не зависит от переданной кодовой комбинации.

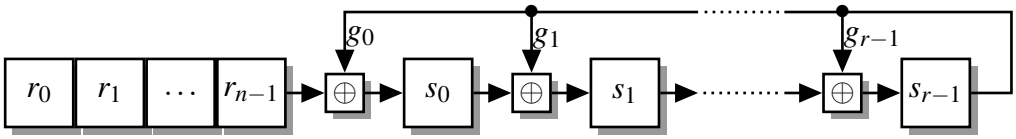

Рис. 6.4. Схема вычисления синдрома для циклического (n,k)-кода

На основании приведенного свойства существует метод определения места ошибки по синдрому s_{n-1}, соответствующий наличию ошибки в старшем разряде. Если ошибка произошла в следующем разряде (более низком), то такой же остаток получится в произведении принятого многочлена $r(x)$ и x.

Рассмотрим принцип декодирования на примере систематического кода $(7,4)$, образующий полином $g(x) = 1 + x + x^3 = (1\ 1\ 0\ 1)$. Пусть передавалась кодовая комбинация $v = (0\,0\,1\,1\,0\,1\,0)$. Первые три разряда проверочные. Вектор ошибки $e = (0\,0\,0\,1\,0\,0\,0)$, т. е. ошибка в четвертом слева разряде, следовательно на вход декодера принята кодовая комбинация $r = (0\,0\,1\,0\,0\,1\,0)$.

Определяем синдром ошибки в крайнем правом разряде как остаток от деления $e_6 = (0\,0\,0\,0\,0\,0\,1)$ на вектор образующего полинома. В результате получается синдром

$$s_6 = 1. \tag{72}$$

Теперь делим принятый вектор r на вектор образующего полинома до получения остатка $s_6 = 1$. По необходимости приписываем справа нули. Поскольку результат деления нас не интересует, не указываем его.

$$\begin{array}{ccccccc|cccc}
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
\cline{8-11}
 & & 1 & 1 & 0 & 1 & & ? & & & \\
\cline{3-6}
 & & & 1 & 0 & 0 & 0 & & & & \\
 & & & 1 & 1 & 0 & 1 & & & & \\
\cline{4-7}
 & & & & 1 & 0 & 1 & \boxed{0} & & & \\
 & & & & 1 & 1 & 0 & 1 & & & \\
\cline{5-8}
 & & & & & 1 & 1 & 1 & \boxed{0} & & \\
 & & & & & 1 & 1 & 0 & 1 & & \\
\cline{6-9}
 & & & & & & & 1 & 1 & \boxed{0} & \boxed{0} \\
 & & & & & & & 1 & 1 & 0 & 1 \\
\cline{8-11}
 & & & & & & & & & & \boxed{1}
\end{array}$$

Из процедуры деления видно, что для того, чтобы получить остаток $s_6 = 1$, потребовалось дописать четыре нуля. Следовательно, ошибка находится на четвертой позиции слева.

При такой процедуре декодирования возможны ещё два варианта.

Если при нахождении остатка он сразу получается равным s_{n-1} (s_6 для нашего примера), то ошибка в крайнем справа разряде и мы можем сразу ее исправить. Если же при нахождении остатка нам пришлось добавить n нулей и остаток s_{n-1} так и не был обнаружен, то такая ошибка не может быть исправлена.

Для случая обратной записи кодовых слов и образующего многочлена процедура декодирования остается той же, но меняется положение ошибки в зависимости отчисла дописанных нулей.

6.8. Оптимальное декодирование на основе анализа веса

Для нахождения ошибочных элементов в циклических кодах с $d_{\min} > 5$ получили распространение методы, основанные на анализе веса остатка. При этом осуществляются следующие процедуры:

1. Принятая кодовая комбинация делится на $g(x)$.
2. Подсчитывается вес остатка от деления ω (т. е. количество единиц в остатке).
3. Если $\omega \leq t_{\text{испр.}}$, где $t_{\text{испр.}}$ — кратность гарантированно исправляемой ошибки, то исправление сводится к сложению принятой кодовой комбинации с остатком от деления.
4. Если $\omega > t_{\text{испр.}}$, то производят циклический сдвиг принятой кодовой комбинации вправо на один разряд, а затем делят на ее и определяют вес остатка. Если то делимое суммируют с остатком, а затем производят циклический сдвиг на один элемент влево. Это и будет исправленная кодовая комбинация.
5. Если после первого сдвига остаток дает $\omega > t_{\text{испр.}}$, то процедуру повторяют до тех пор, пока не будет удовлетворяться условие $\omega \leq t_{\text{испр.}}$. Исправленная комбинация получается в результате сдвига влево суммы последней кодовой комбинации и остатка на столько разрядов, на сколько была сдвинута исходная кодовая комбинация вправо.

Для случая обратной записи кодовых слов и образующего многочлена процедура декодирования остается той же, но сдвиг производится вначале влево, а после исправления ошибки вправо.

Рассмотрим принцип оптимального декодирования также на примере систематического кода $(7,4)$ с $d_{\min} = 3$ ($t_{\text{испр.}} = 1$), образующий полином $g(x) = 1 + x + x^3 = (1\ 1\ 0\ 1)$. Пусть передавалась кодовая комбинация $v = (0\ 0\ 1\ 1\ 0\ 1\ 0)$. Первые три разряда проверочные. Вектор ошибки $e = (0\ 1\ 0\ 0\ 0\ 0\ 0)$, т. е. ошибка во втором слева разряде, следовательно на вход декодера принята кодовая комбинация $r = (0\ 1\ 1\ 1\ 0\ 1\ 0)$.

1. Ищем остаток от деления r на g.

$$\begin{array}{cccccc|cccc} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ \hline & & & & & 1 & 1 & ? & & & \end{array}$$

Вес остатка $\omega = 2 > t_{\text{испр.}}$.

2. Сдвигаем r на разряд вправо $(0\,1\,1\,1\,0\,1\,0) \to (0\,0\,1\,1\,1\,0\,1)$ и находим остаток.

$$\begin{array}{cccccc|cccc} 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ \hline & & & & 1 & 1 & 1 & ? & & & \end{array}$$

Вес остатка $\omega = 3 > t_{\text{испр.}}$.

3. Сдвигаем на разряд вправо $(0\,0\,1\,1\,1\,0\,1) \to (1\,0\,0\,1\,1\,1\,0)$ и находим остаток.

$$\begin{array}{cccccc|cccc} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ \hline & & & & 1 & 0 & 1 & ? & & & \end{array}$$

Вес остатка $\omega = 2 > t_{\text{испр.}}$.

4. Сдвигаем на разряд вправо $(1\,0\,0\,1\,1\,1\,0) \to (0\,1\,0\,0\,1\,1\,1)$ и находим остаток.

$$\begin{array}{cccccc|cccc} 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ \hline & & & & 1 & 0 & 0 & ? & & & \end{array}$$

Вес остатка $\omega = 1 \leq t_{\text{испр.}}$, следовательно, суммируем делимое с остатком.

$$\begin{array}{cc} & 0\ 1\ 0\ 0\ 1\ 1\ 1 \\ \oplus & \quad\quad\quad\ 1\ 0\ 0 \\ \hline & 0\ 1\ 0\ 0\ 0\ 1\ 1 \end{array}$$

5. Поскольку мы трижды сдвигали исходный вектор вправо, сдвигаем его на три разряда влево.

$$(0\,1\,0\,0\,0\,1\,1) \to (1\,0\,0\,0\,1\,1\,0) \to (0\,0\,0\,1\,1\,0\,1) \to (0\,0\,1\,1\,0\,1\,0).$$

В результате получен исправленный кодовый вектор.

6.9. Схемная реализация декодирования циклического кода

По МУ «Код (9, 5)».