ДЕКОДИРОВАНИЕ КОДОВ РИДА-СОЛОМОНА С ИСПОЛЬЗОВАНИЕМ МОДИФИЦИРОВАННОЙ ТЕХНИКИ ВЫЛАВЛИВАНИЯ ОШИБОК

Егоров С.И.

Курский государственный технический университет

I. ВВЕДЕНИЕ

Простым способом декодирования циклических кодов является вылавливание ошибок в области проверочных символов кодового слова. Для этого символы принятого кодового слова переставляются по определенному закону до тех пор, пока все ошибки не попадут в проверочную часть слова [1].

На каждой перестановке виртуальные проверочные символы вычисляются из k информационных символов. Множество информационных символов и виртуальных проверочных символов образует новое кодовое слово длины n. Как только вес вектора разности виртуальных и принятых проверочных символов (он равен расстоянию между принятым и новым кодовым словом) становится меньше, чем число гарантированно исправляемых ошибок (t), то это значит, что все ошибки в принятом кодовом слове выловлены в области виртуальных проверочных символов. При этом вектор разности содержит все ненулевые компоненты вектора ошибок. Для исправления ошибок эти компоненты необходимо прибавить к принятому кодовому слову.

На практике чаще всего используют в качестве перестановок циклические сдвиги. При вылавливании ошибок в качестве синдрома обычно используют полином остатка $q(x)$, определяемый как остаток от деления полинома принятого слова $r(x)$ на порождающий полином кода $G(x)$. Меггит показал [2], что модифицированное значение $q'(x)$, соответствующее левому циклическому сдвигу кодового слова, может быть получено следующим образом:

$$q'(x) = Res_{G(x)}(x*q(x)),$$

где $Res_x(y)$ обозначает остаток от деления y на x. Попадание ошибок в проверочную часть слова определяется по весу полинома $q(x)$. Если $wt\ (q\ (x)) \leq\ t$, все ошибки – в области виртуальных проверочных символов.

Декодеры с вылавливанием ошибок, использующие циклические сдвиги, могут быть реализованы с небольшими аппаратными затратами. Однако их практическое применение ограничено случаями исправления одного пакета ошибок. Число гарантированно исправляемых независимых ошибок невелико, $t < n/k$.

В этой работе предлагается модификация техники вылавливания ошибок для декодирования РС-кодов с $t = 2$. Предлагаемая модификация предполагает неполное вылавливание ошибок, при котором некоторые ошибки могут находиться в области информационных символов.

II. ПРОЦЕДУРА ДЕКОДИРОВАНИЯ

Представленная процедура декодирования выполняется за n шагов. Принятое из канала кодовое слово после каждого шага подвергается перестановке. На каждом шаге исправляются ошибки в фиксированном проверочном символе r_i .

Неполное вылавливание ошибок предусматривает исправление ошибки в символе r_i и в том случае, когда часть ошибок находится в информационных символах кодового слова. При этом число этих ошибочных символов (t_m) не должно превышать $t-1$.

Пусть $e(x)$ – многочлен ошибок, и

$$e(x) = r(x) – t(x),$$

где $t(x)$ – многочлен кодового слова, переданного в канал. Представим $e(x)$ в виде суммы:

$$e(x) = e_m(x) + e_c(x),$$

где многочлен $e_m(x)$ представляет ошибки информационной части слова, а многочлен $e_c(x)$ – проверочной части, причем степень $e_c(x)$ меньше, чем $n-k$. Тогда многочлен остатка $q(x)$ можно представить в следующем виде:

$$q(x) = q_m(x) + e_c(x),$$

где

$$q_m(x) = Res_{G(x)}\ e_m(x).$$

Отсюда

$$q_i = q_{mi} + e_{ci}\ .$$

Следовательно, значение ошибки для символа r_i может быть представлено в следующем виде:

$$Y = e_{ci} = q_i – q_{mi}\ ,$$

где q_{mi} – компонента , обусловленная влиянием ошибок в информационной части слова.

Таким образом вычисление значения ошибки Y можно свести к вычислению значения q_{mi} .

Пусть РС-код имеет минимальное кодовое расстояние $d=2*t+1$. Тогда при наличии t_m ошибочных символов в информационной части слова и отсутствии ошибок в проверочных символах любые $2*t_m$ коэффициентов многочлена $q(x)$ однозначно определяют оставшиеся $d-1-2*t_m$ коэффициентов. Причем вид функциональной зависимости коэффициента от любых других $2*t_m$ коэффициентов определяется только их выбором и параметрами кода и не зависит от расположения и значения ошибочных символов в

информационной части слова. Параметры необходимого количества таких зависимостей должны быть определены на этапе проектирования декодера.

Использование зависимости q_{mi} от любых других $2*t_m$ неискаженных ошибками коэффициентов $q(x)$ позволяет вычислить значение ошибки Y при неполном вылавливании ошибок.

Рассмотрим принципы вычисления Y. Примем следующие обозначения: I – множество всех информационных символов кодового слова $r(x)$, R^i – множество всех символов $r(x)$ за исключением r_i, C^i – множество всех проверочных символов за исключением r_i , Q^i – множество коэффициентов многочлена $q(x)$ за исключением q_i.

Значение Y должно вычисляться в случаях, когда число ошибочных символов в R^i меньше t. Если в R^i число ошибочных символов равняется t, исправление ошибки в символе r_i не должно осуществляться.

Случай, когда все ошибки произошли в проверочной части, может быть определен по весу многочлена $q(x)$, он должен быть в этом случае меньше или равен t. В этом случае значение q_{mi} принимается равным нулю.

Если I содержит t-1 ошибочных символов ($t_m = t$-1) и в C^i ошибок нет, то любой коэффициент $q(x)$ из Q^i будет однозначно определяться оставшимися $2*(t$-1$)$ коэффициентами из Q^i. Ситуация наличия t-1 ошибочных символов в I (при этом возможна ошибка в r_i) определяется по выполнению вышеупомянутой функциональной зависимости для коэффициентов из Q^i. В этом случае q_{mi} можно вычислить как функцию любых $2*t$-2 коэффициентов $q(x)$ из Q^i.

В случаях, когда t-1 ошибок распределились между I и C^i , значение t_m и расположение $2*t_m$ коэффициентов $q(x)$ из Q^i , соответствующих не пораженным ошибками проверочным символам, могут быть найдены путем анализа выполнения множества функциональных зависимостей между различными коэффициентами $q(x)$ из Q^i. После чего вычисляется q_{mi} как функция от $2*t_m$ коэффициентов многочлена $q(x)$.

Рассмотрим более подробно процедуру декодирования в случае исправления 2-х ошибочных символов ($t=2$). Этот случай интересен с практической точки зрения, поскольку реализация декодера получается простой. Сложность декодирования сильно возрастает при $t>2$.

III. ИСПРАВЛЕНИЕ ДВУХ ОШИБОЧНЫХ СИМВОЛОВ

Для получения функции $Y=f(q(x))$ в случае исправления двух ошибочных символов нам потребуется Следствие 1 из следующей теоремы.

Теорема.

Пусть два произвольно выбранных ненулевых коэффициента q_i и q_j полинома остатка $q(x)$ имеют свойства:

$$q_i/g_i \neq q_j/g_j, \quad q_i/(g_i a^i) \neq q_j/(g_j a^j). \tag{1}$$

Смежный класс РС-кода имеет лидер веса 1 с ненулевой компонентой, расположенной в информационной части слова, тогда и только тогда, когда другие d-3 коэффициента полинома остатка $q(x)$ связаны с выбранными коэффициентами следующим образом:

$$q_l = g_l a^l (a^i + a^j)/[q_i^{-1} g_i a^i (a^l + a^j) + q_j^{-1} g_j a^i (a^j + a^l)], \quad l=0,..,d-2;\ l \neq i,j\ , \tag{2}$$

где

$$g(x) = \prod_{j=b+2}^{b+d-1} (x - \alpha^j) = \frac{G(x)}{(x - \alpha^{b+1})} = \sum_{j=0}^{d-2} g_j x^j \quad ,$$

α – примитивный элемент конечного поля и b – целочисленная константа.

При доказательстве теоремы использовался материал из [3].

Следствие 1. Пусть в информационной части кодового слова РС-кода произошла одна ошибка. Тогда справедливо следующее:

$$q_i^{-1} g_i a^i (a^l + a^j) + q_j^{-1} g_j a^j (a^i + a^l) + q_l^{-1} g_l a^l (a^i + a^j) = 0. \tag{3}$$

где q_i , q_j , q_l – произвольно выбранные коэффициенты полинома остатка $q(x)$.

Уравнение (3) не может выполняться при наличии двух ошибок в информационной части кодового слова.

Рассмотрим вычисление функции $f(q(x))$ при различном числе и расположении исправимых ошибок в кодовом слове. Без ограничения общности будем рассматривать исправление ошибок применительно к проверочному символу r_0.

1. В кодовом слове нет ошибок. Тогда $q(x)=0$, $wt(q(x))=0$ и $Y=0$ ($wt((q(x))$ – вес многочлена $q(x)$).

2. В кодовом слове один ошибочный символ. Тогда $q(x)\neq 0$.

Если ошибочный символ находится в I, то в соответствии со Следствием 1 должно выполняться соотношение (3) для коэффициентов из Q^0. В этом случае Y должно равняться нулю.

Если ошибочный символ находится среди проверочных символов кодового слова, то $wt(q(x))=1$, и единственный ненулевой символ многочлена остатка дает значение ошибки для соответствующего проверочного символа. Тогда мы можем записать $Y=q_0$.

Исправление одного ошибочного символа по предлагаемому алгоритму осуществляется путем, традиционным для декодеров с вылавливанием ошибок.

3. В кодовом слове два ошибочных символа.

Если оба этих символа находятся в проверочной части, то $wt(q(x))=2$ и $Y=q_0$.

Если оба ошибочных символа находятся в I, то в соответствии со Следствием 1 соотношение (3) не может выполняться для коэффициентов из Q^i и $Y=0$.

Если один ошибочный символ находится в I, и один - в C^i , то в соответствии со Следствием 1 соотношение (3) для коэффициентов из Q^i не может выполняться и $Y=0$.

Если один ошибочный символ находится в I, и один - в r_0, то в соответствии со Следствием 1 соотношение (3) для коэффициентов из Q^i будет выполняться. Ошибка в младшем символе не влияет на Q^i и q_{m0} можно найти из соотношения (3):

$$q_{m0}=g_0a^0(a^i+a^j)/[q_i^{-1}g_ia^i(a^0+a^j) + q_j^{-1}g_ja^i(a^j+a^0)],$$

и

$$Y=q_0-q_{m0}\ .$$

Тогда алгоритм исправления ошибки в младшем разряде кодового слова РС-кода можно записать в следующем виде.

1. Если $wt(q(x))=0$ ошибок в принятом слове нет и $Y=0$.
2. Если $wt(q(x))$ меньше или равен 2, то все ошибки в проверочной части, и $Y=q_0$.
3. Если не выполняется соотношение (3) , то R^i содержит не менее двух ошибочных символов и $Y=0$.
4. Если соотношение (3) выполняется, то $Y=q_0-q_{m0}$.

Теперь функция $f(q(x))$ может быть записана в следующем виде:

$$Y = f(q(x)) = \begin{cases} 0, \text{если } wt(q(x)) = 0 \\ q_0, \text{если } wt(q(x)) \le 2 \\ 0, \text{если } wt(q(x)) > 2 \text{ и (3) не выполняется} \\ 0, \text{если } wt(q(x)) > 2 \text{ и (3) выполняется} \end{cases}$$

где i, j, l могут принимать значения 3, 2, 1 в любом порядке.

Структурная схема потокового декодера, предназначенного для работы в оборудовании коммутации данных телекоммуникационных систем, приведена на рисунке 1. Этот декодер позволяет исправлять ошибки в данных в темпе их вывода в выходной порт сразу же после завершения приема данных в буферное ОЗУ.

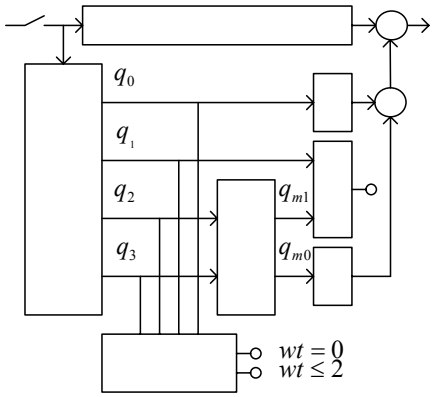

Рис.1. Структурная схема декодера

Для использования потокового декодера в аппаратуре коммутации данных желательно не допускать ложной коррекции не буферируя дополнительно при этом данные. Ложную коррекцию можно предотвратить используя РС-коды с кодовым расстоянием $d>2*t+1$. Эти коды позволяют наряду с исправлением t ошибок гарантированно обнаруживать до $t_e=d-2*t-1$ дополнительных ошибок. В рассмотренной в разделе II процедуре декодирования ложную коррекцию можно предотвратить, используя такую функцию $f(q(x))$, которая будет равняться нулю, если число ошибок в слове будет больше t. Такого рода функцию для рассматриваемых потоковых декодеров с t=2 можно получить используя вышеприведенную теорему.

IV. ВЫВОДЫ

Предлагаемая процедура декодирования расширяет возможности техники вылавливания ошибок применительно к РС-кодам и позволяет в случае исправления 2 ошибок реализовать потоковые декодеры с небольшими аппаратными затратами и минимальной задержкой декодирования (в одно кодовое слово).

Литература

1. R.E. Blahut, *Theory and Practice of Error Control Codes.* New York: Addison-Wesley, 1983.
2. J.R. Meggitt, "Error-correcting codes and their implementation for data transmission systems," *IRE Trans. on Information Theory*, vol. IT-7, pp. 234-244, October 1961.
3. Пат.WO 85/01625 PCT, МКИ 4 H03M13/00,G06F11/10. Error correction for algebraic block codes / E.R.Berlekamp, L.R.Welch (США). -Заявлено 26.09.84; PCT/US84/01557; Опубликовано 11.04.85.