УДК 519.725, 681.3

ЦИКЛИЧЕСКОЕ КОДИРОВАНИЕ ЦИФРОВОЙ ИНФОРМАЦИИ НА ОСНОВЕ ДВОЙСТВЕННЫХ ПОЛИНОМОВ

Дяченко Валерий Олегович, Дяченко Олег Николаевич
Донецкий национальный технический университет
г. Донецк, Украина

Аннотация

Выполнен анализ способов построения и практической реализации циклических кодов. Рассмотрены особенности аппаратного декодирования кодов построенных на основе двойственных порождающих полиномов. Обоснован вывод о рациональности применения двойственных полиномов для кодов, исправляющих пакеты ошибок большой длины.

Ключевые слова: *циклические коды, синдромное декодирование, двойственные порождающие полиномы, поле Галуа.*

CYCLIC ENCODING DIGITAL INFORMATION ON THE BASIS OF DUAL POLYNOMIALS

Dyachenko Valery Olegovich, Dyachenko Oleg Nikolaevich
Donetsk national technical university
Donetsk, Ukraine

Abstract

The methods for constructing and implementation of the cyclic codes have been analyzed. Features of the hardware decoding of codes based on dual generator polynomials have been considered. The conclusion of rationality of applying the dual polynomials for burst-error-correcting codes of large length was substantiated.

Keywords: *Reed-Solomon codes, syndrome decoding, dual generator polynomials, Galois field.*

Введение

Расширение сфер внедрения современных инновационных технологий в настоящее время приводит к непрерывному увеличению объема информационных данных. Такая тенденция, несомненно, сохранится и в будущем. Поэтому, все большее значение приобретают способы помехоустойчивого кодирования, обеспечивающие требуемую достоверность при передаче, обработке и хранении информации. Одними из наиболее эффективных для исправления ошибок являются циклические коды. Эти коды нашли широкое применение благодаря простой аппаратной реализации и высоким корректирующим способностям. В связи с этим вопросы построения и аппаратной реализации циклических кодов являются актуальными, учитывая все большую их популярность и востребованность для различных сфер применения.

Естественно, что наиболее популярны в настоящее время коды, исправляющие пакеты ошибок: коды Файра, (255, 223, 33) код Рида-Соломона для космической связи NASA, расширенный (128, 122, 7), код Рида-Соломона над полем Галуа $GF(2^7)$ для кабельных модемов и многие другие [1-5]. Тем не менее, циклический код Хэмминга, исправляющий одиночные ошибки, заслуживает особого внимания, поскольку является фундаментом для понимания принципов построения более мощных кодов. Одним из примеров может служить декодеры двоичного кода Хэмминга и недвоичного кода Рида-Соломона, исправляющих одиночные (двоичные и недвоичные) ошибки[1-3]. Кроме того, схемные решения компактного тестирования, в частности, генераторы псевдослучайных тестов и сигнатурные

анализаторы, построенные на основе декодеров Хэмминга, используются во многих схемах со встроенным самотестированием. Например, диагностическое обеспечение микропроцессорной СБИС S/390 (IBM) содержит встроенные генераторы псевдослучайных тестовых последовательностей с управляемым весом и встроенный многоканальный сигнатурный анализатор.

При построении циклических кодов во многих случаях приходится их укорачивать (например, укороченные коды Рида-Соломона над полем Галуа $GF(2^8)$ для CD-ROM, DVD и цифрового телевидения высокого разрешения - формат HDTV) [1-3, 5]. В данной работе предлагается для кодирования и декодирования кодов применение двойственных полиномов, что дает преимущества при реализации укороченных кодов, исправляющих пакеты ошибок большой длины.

1. Укороченные коды

Из любого (n, k) циклического кода можно получить (n-i, k-i) укороченный код, где n - длина кода, k – количество информационных символов, $0<i<k$ – параметр укорачивания. Одним из способов декодирования укороченных кодов является использование декодеров, построенных для кодов максимальной длины. При этом принятому кодовому слову предпосылаются i нулей, которые кодером не передаются в канал связи. Недостатком такого способа декодирования является несогласованность скоростей передачи кодером кодового слова (длина такого слова n-i, поскольку нули не передаются) и обработки декодером принятого дополненного нулями кодового слова длины n. Кроме того, для формирования синдрома в этом случае необходимо n тактов работы декодера, в то время как при применении другого способа декодирования для этого достаточно n-i тактов.

В отличие от декодера кода максимальной длины, который для формирования синдрома выполняет операции умножения принятого слова на полином X^{n-k} и деления на порождающий полином, декодер укороченного кода умножает на полином, равный остатку от деления полинома X^{n-k+i} на порождающий полином, и полученное произведение делит на порождающий полином. Однако в случае очень большого параметра укорачивания довольно сложно получать остаток от деления полинома X^{n-k+i} на порождающий полином. Так, например, для кода Файра, исправляющего пакет длины 64, получаем такую длину кода, при которой он нереализуем в неукороченном виде. Вместе с тем, его нельзя реализовать при традиционном укорачивании кода. Существует несколько способов определения остатка X^{n-k+i} на порождающий полином. Один из них – аппаратное или программное деление с помощью кодера при подаче на его вход одной единицы и i нулей. Однако, если i мало, нельзя реализовать такой код, поскольку кодовое слово получаем невероятной величины. Таким образом, параметр укорачивания i в любом случае должен быть очень большим, сравнимым с длиной кода. Если i велико, получаем время деления аппаратным способом также невероятной величины. Второй способ – разложение X^{n-k+i} на сомножители для упрощения определения остатка от деления полинома X^{n-k+i} на порождающий полином также не представляется возможным, не говоря уже о третьем способе – деления “уголком”. Тем не менее, с помощью двойственных полиномов такой укороченный код все-таки можно реализовать.

Основная идея отличия применения двойственных полиномов в качестве порождающих полиномов для кодирования и декодирования циклических кодов заключается в том, что декодер выполняет исправление принятого слова по принципу LIFO, а не FIFO, то есть, в обратном порядке следования кодового слова.

2. Двойственные полиномы

Полином $K^*(X) = X^{\deg K(X)} * K(X^{-1})$ называется двойственным полиномом по отношению к полиному K(X).

Коды двойственных полиномов имеют одинаковые характеристики, в частности, одинаковые корректирующие способности, избыточность, аппаратные затраты схемной реализации кодеров и декодеров, быстродействие. В случае разных полиномов получаем декодер, который обрабатывает биты кодового слова по принципу "последний пришел – первый вышел".

Пример 1. Построим два поля Галуа $GF(2^4)$ как расширения поля GF(2) над примитивными полиномами $p(z)=z^4+z+1$ и $p^*(z)=z^4+z^3+1$ (табл. 1). Элементы поля могут быть представлены в различном обозначении и для ненулевых элементов со степенью большей степени порождающего полинома следуют в обратном порядке (в декодере Меггитта кода максимальной длины выполняется умножение на полином X^{n-k}).

Таблица 1 – Элементы поля Галуа $GF(2^4)$ с двойственными порождающими полиномами

$p(z)=z^4+z+1$			$p^*(z)=z^4+z^3+1$		
В виде степени	В виде полинома	В двоичном виде	В виде степени	В виде полинома	В двоичном виде
0	0	0000	0	0	0000
α^0	1	0001	$\alpha^0=\alpha^{-15}$	1	0001
α^1	z	0010	$\alpha^1=\alpha^{-14}$	z	0010
α^2	z^2	0100	$\alpha^2=\alpha^{-13}$	z^2	0100
α^3	z^3	1000	$\alpha^3=\alpha^{-12}$	z^3	1000
α^4	z +1	0011	$\alpha^4=\alpha^{-11}$	z^3 +1	1001
α^5	z^2 + z	0110	$\alpha^5=\alpha^{-10}$	z^3 + z + 1	1011
α^6	z^3 + z^2	1100	$\alpha^6=\alpha^{-9}$	z^3 + z^2 + z + 1	1111
α^7	z^3 + z + 1	1011	$\alpha^7=\alpha^{-8}$	z^2 + z + 1	0111
α^8	z^2 + 1	0101	$\alpha^8=\alpha^{-7}$	z^3 + z^2 + z	1110
α^9	z^3 + z	1010	$\alpha^9=\alpha^{-6}$	z^2 + 1	0101
α^{10}	z^2 + z + 1	0111	$\alpha^{10}=\alpha^{-5}$	z^3 + z	1010
α^{11}	z^3 + z^2 + z	1110	$\alpha^{11}=\alpha^{-4}$	z^3 + z^2 + 1	1101
α^{12}	z^3 + z^2 +z + 1	1111	$\alpha^{12}=\alpha^{-3}$	z + 1	0011
α^{13}	z^3 + z^2 + 1	1101	$\alpha^{13}=\alpha^{-2}$	z^2 +z	0110
α^{14}	z^3 + z + 1	1001	$\alpha^{14}=\alpha^{-1}$	z^3 + z^2	1100
α^0	1	0001	$\alpha^{15}=\alpha^0$	1	0001
α^1	z	0010	α^1	z	0010
α^2	z^2	0100	α^2	z^2	0100
α^3	z^3	1000	α^3	z^3	1000

Анализ таблицы показывает, что, если остаток от деления кодового слова без ошибки на порождающий полином K(X) $R_{K(X)}(A(X)*X^{n-k})=0$, тогда остаток от деления кодового слова в обратном порядке следования без ошибки на двойственный порождающий полином $K^*(X)$ $R_{K^*(X)}(A(X^{-1})*X^{n-k})=0$.

Пример 2. Рассмотрим более подробно работу кодера и декодера для кода Хэмминга (15, 11) с порождающим полиномом кодера $K(X) = X^4 + X + 1$ и порождающим полиномом декодера $K^*(X) = X^4 + X^3 + 1$.

Пусть информационные символы A = 11010001001 в двоичном виде, или $A(X) = X^{10}+X^9+X^7+X^3 +1$ в полиномиальном виде. Для систематического кода кодер циклического

кода Хэмминга выполняет операцию умножения информационной последовательности на полином X^{n-k} и деление на порождающий полином. Полученный остаток от деления представляет собой проверочную часть кодового слова. После выполнения этих операций получаем систематический код $X^{14}+X^{13}+X^{11}+X^7+X^4+X$ или 11010001001 0010, где первая часть – информационная, а вторая – проверочная.

Рассмотрим различные варианты применения порождающих полиномов. В таблице 2: А – кодовое слово, ФС1(А) – формирователь синдрома для К(Х), А* – кодовое слово в обратном порядке следования, ФС2(А*) – формирователь синдрома для К*(Х), A^*_{15} – кодовое слово в обратном порядке следования с ошибкой в 15-м символе кодового слова А, E^*_{15} – полином ошибки, ФС2(E^*_{15}) – формирователь синдрома для К*(Х).

ФС представляют собой регистры сдвига с линейной обратной связью (РСЛОС), выполняющие функцию деления кодового слова на порождающий полином. В 6 и 8 столбцах в последних строках получаем комбинацию все нули и последняя единица (ФС для декодера Меггитта). Анализ таблицы 2 показывает, что, во-первых, результат деления А на К(Х) и А* на К*(Х) равны нулю (остаток от деления кодового слова, записанного в разном порядке и деленного на двойственные полиномы равен нулю, что означает отсутствие ошибки). Во-вторых, результаты деления реальной последовательности A^*_{15} с ошибкой и последовательности ошибки $E^*_{15} = A + A^*_{15}$ совпадают, что позволяет рассматривать только последовательности ошибок вне зависимости от реальных значений кодовых слов (такой же результат будет для любых одиночных ошибок).

Таблица 2 – Различные варианты применения порождающих полиномов

№	А	ФС1(А)	A^*	ФС2(A^*)	A^*_{15}	ФС2(A^*_{15})	E^*_{15}	ФС2(E^*_{15})
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	1	1100	0	0000	1	1001	1	1001
2	1	1010	1	1001	1	0100	0	1101
3	0	0101	0	1101	0	0010	0	1111
4	1	0010	0	1111	0	0001	0	1110
5	0	0001	1	0111	1	0000	0	0111
6	0	1100	0	1010	0	0000	0	1010
7	0	0110	0	0101	0	0000	0	0101
8	1	1111	1	0010	1	1001	0	1011
9	0	1011	0	0001	0	1101	0	1100
10	0	1001	0	1001	0	1111	0	0110
11	1	0100	0	1101	0	1110	0	0011
12	0	0010	1	0110	1	1110	0	1000
13	0	0001	0	0011	0	0111	0	0100
14	1	0000	1	0001	1	0011	0	0010
15	0	0000	1	0000	1	0001	0	0001

В-третьих, при реализации ФС в виде РСЛОС, выполняющего функцию умножения на полином X^{n-k} и деления на двойственный полином последовательности кодового слова в порядке обратного следования дает остаток из всех нулей и последней единицы (как и в обычном декодере Меггитта), что дает преимущества при построении схемы исправления в декодере.

Заключение

Таким образом, при предлагаемой схемной реализации циклических кодов кодер остается прежним. Декодер имеет следующие отличия. В своем составе он содержит реверсивный циклический n-разрядный буферный регистр, формирователь синдрома и схему исправления ошибки. За первые n тактов принятое кодовое слово записывается в буферный регистр и одновременно формируется синдром. Затем направление сдвига в буферном регистре изменяется и за следующие n тактов с помощью формирователя синдрома и схемы исправления декодер устраняет ошибку, и кодовое слово уже без ошибки перезаписывается в буферный регистр. За третьи n тактов направление сдвига в буферном регистре снова меняется и на выходе появляется исправленное кодовое слово.

Литература

1. Richard E.Blahut. Algebraic Codes for Data Transmission/ Cambridge University Press, 2012. – 498 p.

2. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. – М.: Мир, 1976. – 595 с.: ил.

3. Дяченко В.О., Зинченко Ю.Е., Дяченко О.Н. Исследование способов проектирования кодов Рида-Соломона// Інформаційні управляючі системи та комп'ютерний моніторинг (ІУС КМ-2014) : V Всеукраїнська науково-технічна конференція студентів, аспірантів та молодих вчених, 22-23 квітня 2014 р., м. Донецьк : зб. доп./ Донец. національ. техн. ун-т; редкол. В.А.Світлична. – Донецьк: ДонНТУ, 2014. – в 2 тт. – т.2. – С. 72-78.

4. Дяченко В.О., Дяченко О.Н. Анализ способов реализации кодов Рида-Соломона, исправляющих двойные ошибки// Современные тенденции развития и перспективы внедрения инновационных технологий в машиностроении, образовании и экономике: материалы Международной научно-практической конференции (Азов, 19 мая 2014 г.). – Ростов н/Д, ДГТУ, 2014. – С. 18-22.

5. Зинченко Е.Ю., Дяченко О.Н. Сравнительный анализ способов укорачивания кодов Рида-Соломона// Збірка праць VII міжнародної науково-технічної конференції студентів, аспірантів та молодих науковців – 22-23 листопада 2011 р., Донецьк, ДонНТУ. – 2011. У 2-х томах, Т. 1 – С. 48-52.