

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СПОСОБОВ УКРАЧИВАНИЯ КОДОВ РИДА-СОЛОМОНА

*Зинченко Е.Ю., Дяченко О.Н.
Донецкий национальный технический университет*

Рассматриваются вопросы укорачивания кодов. В частности, выполнен анализ способов укорачивания циклических кодов, рассмотрены их достоинства и недостатки. Предлагается способ аппаратной реализации формирования синдрома для укороченных кодов Рида-Соломона.

Введение

Коды Рида-Соломона были предложены в 1960 году сотрудниками Массачусетского технологического института Ирвином Ридом и Густавом Соломоном [1]. Первое применение код Рида-Соломона получил только в 1982 году в серийном выпуске компакт-дисков, хотя эффективные алгоритмы декодирования были предложены уже в 1969 году Элвином Берлекэмпом и Джэймсом Месси (алгоритм Берлекэмпа-Мэсси) [2]. В настоящее время Коды Рида-Соломона используются в огромном числе приложений в сфере цифровых телекоммуникаций и при построении запоминающих устройств. Они применяются для исправления ошибок во многих системах: устройствах памяти (включая интегральные схемы памяти, CD, DVD, штриховые коды, и т.д.); беспроводных или мобильных коммуникациях (включая сотовые телефоны, микроволновые каналы и т.д.); спутниковых коммуникациях; цифровом телевидении DVB (digital video broadcast); скоростных модемах, таких как ADSL, xDSL и т.д. [3].

Цель данной работы – выполнить сравнительный анализ способов укорачивания кодов рида-Соломона и рассмотреть особенности построения их порождающих полиномов и аппаратной реализации - от функциональных до принципиальных схем.

Актуальность работы заключается в том, что, хотя коды Рида-Соломона широко описаны в литературе [4, 5, 6], вместе с тем аппаратная реализация кодирования и декодирования, в отличие от программной, освещена либо вкратце, либо вовсе отсутствует.

1 Определение порождающего полинома кода Рида-Соломона

Коды Рида-Соломона являются частным случаем кодов БЧХ. Таким образом, они являются блоковыми линейными циклическими кодами. Главное отличие кодов Рида-Соломона заключается в том, что они являются недвоичными, т.е. в качестве символа выступает не двоичный символ (один бит), а элемент поля Галуа (несколько битов). Вместе с тем кодирующие и декодирующие устройства реализуются с помощью обычных элементов двоичной логики.

Порождающий полином кода Рида-Соломона, исправляющего s ошибок, должен содержать $2s$ корней:

$$\{\alpha^j, \alpha^{j+1}, \alpha^{j+2}, \dots, \alpha^{j+2s-1}\},$$

где j_0 – конструктивный параметр.

Если j выбрать равным 1, тогда для кода Рида-Соломона, исправляющего s ошибок, порождающий полином имеет следующий вид:

$$RS(X) = (X - \alpha)(X - \alpha^2)(X - \alpha^3) \dots (X - \alpha^{2s}),$$

а множество корней полинома - $\{\alpha, \alpha^2, \alpha^3 \dots \alpha^{2^s}\}$.

Например, для кода Рида-Соломона, исправляющего одиночные ошибки, (KPC1) порождающий полином имеет общий вид $RS(X) = (X - \alpha)(X - \alpha^2)$.

2 Укороченные коды Рида-Соломона

Коды Рида-Соломона относятся к классу циклических кодов. Из любого (n, k) циклического кода можно получить $(n-i, k-i)$ укороченный код, где n - длина кода, k - количество информационных символов, $i < k$ - 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. Именно такой способ применения укороченных кодов упоминается при рассмотрении укороченных кодов Рида-Соломона. 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 альтернативного способа декодирования для этого достаточно $n-i$ тактов.

Идея такого декодирования заключается в том, что, в отличие от декодера кода максимальной длины, который для формирования синдрома выполняет операции умножения принятого слова на полином X^p и деления на порождающий полином, декодер укороченного кода умножает на полином, равный остатку от деления полинома X^{p+i} на порождающий полином, и полученное произведение делит на порождающий полином. Примеры такого декодирования широко освещены для двоичных кодов [5], чего нельзя сказать для укороченных кодов Рида-Соломона, в особенности их аппаратной реализации. Кодирующие устройства для укороченных кодов ничем не отличаются от кодеров кодов максимальной длины. Поэтому рассматривать будем только декодирующие схемы для формирования синдрома.

3 Декодирующие устройства кодов Рида-Соломона

Для определения параметров кода (таких как длина кода, количество информационных символов, корректирующие возможности кода) и построения кодирующего и декодирующих устройств на уровне принципиальной схемы недостаточно знать только порождающий полином в общем виде. Для этого необходимо определить поле Галуа, над которым строится код.

Например, построим код KPC1 над полем Галуа $GF(2^4)$. В этом случае длина кода $n=15$ символов поля Галуа $GF(2^4)$ ($n=2^4-1$), количество информационных символов $k=n-r$, и, поскольку количество проверочных символов r равно степени порождающего полинома (для KPC1 $r=2$), $k=13$. Вместе с тем, поле Галуа $GF(2^4)$ может быть построено как расширение поля $GF(2)$ над разными полиномами $p(z)$.

Возможно несколько форм записи порождающего полинома для кодов Рида-Соломона. Наиболее используемая форма зависит от поля Галуа, над которым строится код Рида-Соломона.

Рассмотрим поле Галуа $GF(2^4)$. Выберем в качестве полинома $p(z)$ неприводимый примитивный z^4+z^3+1 . Элементы поля могут быть представлены в различном обозначении (табл.1). Используя эту таблицу, можно получить следующую форму порождающего полинома кода Рида-Соломона, исправляющего одиночную ошибку:

$$RS(X) = (X - \alpha)(X - \alpha^2) = X^2 + (\alpha + \alpha^2)*X + \alpha^3 = X^2 + \alpha^{13}*X + \alpha^3$$

Таблица 1. Элементы поля Галуа GF(2⁴)

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

Однако для построения устройств лучше использовать другую форму, для которой нет необходимости в построении поля Галуа.

Как правило, для построения кодов Рида–Соломона используют расширения поля GF(2) над примитивным полиномом $p(z)$. В этом случае, в соответствии с определением примитивного полинома, элемент поля z является примитивным. Поэтому вместо обозначения примитивного элемента α можно использовать z :

$$RS(X) = X^2 + (\alpha + \alpha^2) * X + \alpha^3 = X^2 + (z + z^2) * X + z^3.$$

Кодер КРС1 аналогичен кодеру циклического кода Хэмминга. Как и для кодов Хэмминга, кодер систематического кода Рида-Соломона представляет собой схему умножения на полином X^p и деления на порождающий полином; кодер несистематического кода Рида-Соломона представляет собой схему умножения на порождающий полином. Отличие в этих кодах - разная интерпретация символа кодового слова и, соответственно, элементов памяти и умножителей на константу.

Схема декодера аналогична декодеру Меггитта для циклического кода Хэмминга, за исключением того, что каждый элемент задержки в данном случае не один элемент памяти, а четыре. Кроме того, у этой схемы специфические умножители на константу. Именно эти умножители и представляют затруднение при преобразовании функциональной схемы в принципиальную. Особенности реализации для кодеров и декодеров одинаковы, поэтому рассматривать их будем на примере декодеров.

На рисунке 1 представлена функциональная схема декодера (15, 13) КРС1. Для аппаратной реализации такой схемы на элементах двоичной логики необходимо преобразовать способ представления умножителей на константу z^3 и (z^2+z) .

Элемент поля Галуа GF(2⁴) в общем виде можно представить следующим образом:

$$a_3z^3 + a_2z^2 + a_1z + a_0.$$

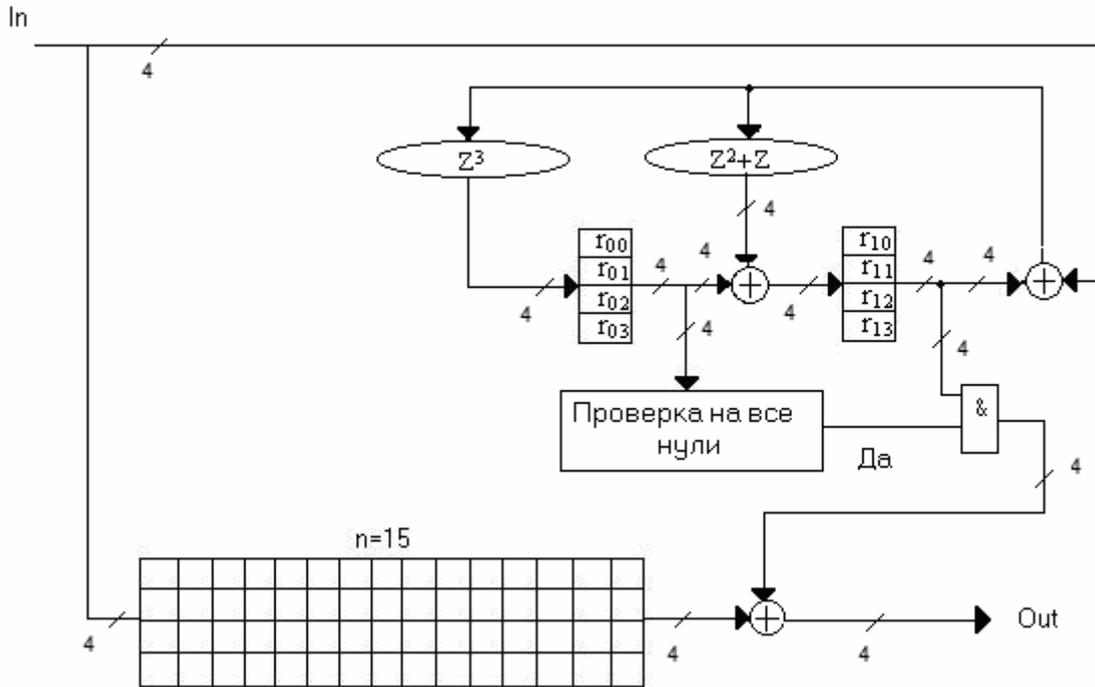


Рисунок 1 – Декодер кода Рида-Соломона для поля Галуа $GF(2^4)$ при $s=1$

Все операции над таким элементом выполняются по модулю 2 и по модулю $p(z)$. Для преобразования умножителей на константу z^3 найдем остаток от деления произведения $a_3z^3 + a_2z^2 + a_1z + a_0$ и z^3 на полином $p(z) = z^4 + z^3 + 1$.

$$\begin{array}{r}
 a_3z^6 + a_2z^5 + a_1z^4 + a_0z^3 \\
 - a_3z^6 + a_3z^5 + a_3z^2 \\
 \hline
 (a_3 + a_2)z^5 + a_1z^4 + a_0z^3 + a_3z^2 \\
 - (a_3 + a_2)z^5 - (a_3 + a_2)z^4 + (a_3 + a_2)z \\
 \hline
 (a_3 + a_2 + a_1)z^4 + a_0z^3 + a_3z^2 + (a_3 + a_2)z \\
 - (a_3 + a_2 + a_1)z^4 + (a_3 + a_2 + a_1)z^3 + (a_3 + a_2 + a_1) \\
 \hline
 (a_3 + a_2 + a_1 + a_0)z^3 + a_3z^2 + (a_3 + a_2)z + (a_3 + a_2 + a_1)
 \end{array}
 \left| \begin{array}{l}
 z^4 + z^3 + 1 \\
 \hline
 a_3z^2 + (a_3 + a_2)z + (a_3 + a_2 + a_1)
 \end{array} \right.$$

Каждый шаг выполнения операции деления позволяет определить остатки от деления произведения $1, z, z^2, z^3$ и $p(z)$ на полином $p(z)$:

$$\begin{aligned}
 1 \quad R_0 &= R_{p(z)}[(a_3z^3 + a_2z^2 + a_1z + a_0)1] = a_3z^3 + a_2z^2 + a_1z + a_0 \\
 z \quad R_1 &= R_{p(z)}[(a_3z^3 + a_2z^2 + a_1z + a_0)z] = (a_3 + a_2)z^3 + a_1z^2 + a_0z + a_3 \\
 z^2 \quad R_2 &= R_{p(z)}[(a_3z^3 + a_2z^2 + a_1z + a_0)z^2] = (a_3 + a_2 + a_1)z^3 + a_0z^2 + a_3z + (a_3 + a_2) \\
 z^3 \quad R_3 &= R_{p(z)}[(a_3z^3 + a_2z^2 + a_1z + a_0)z^3] = (a_3 + a_2 + a_1 + a_0)z^3 + a_3z^2 + (a_3 + a_2)z + (a_3 + a_2 + a_1)
 \end{aligned}$$

Используя полученные выражения и принцип суперпозиции, получаем остаток от деления:

$$\begin{aligned}
 &\text{произведения } a_3z^3 + a_2z^2 + a_1z + a_0 \text{ и } z^2 + z \text{ на полином } p(z) \\
 &R_{p(z)}[(a_3z^3 + a_2z^2 + a_1z + a_0)(z^2 + z)] = R_2 + R_1 = a_1z^3 + (a_1 + a_0)z^2 + (a_3 + a_0)z + a_2; \\
 &\text{произведения } a_3z^3 + a_2z^2 + a_1z + a_0 \text{ и } z^2 + 1 \text{ на полином } p(z) \\
 &R_{p(z)}[(a_3z^3 + a_2z^2 + a_1z + a_0)(z^2 + 1)] = R_2 + R_0 = (a_2 + a_1)z^3 + (a_2 + a_0)z^2 + (a_3 + a_1)z + (a_3 + a_2 + a_0).
 \end{aligned}$$

Обозначим разряды генератора синдрома r_0, r_1 , входные разряды in , разряды линии обратной связи fb . Получаем следующие равенства для реализации декодера на элементах двоичной логики:

$$\begin{aligned} fb_0(t) &= r_{10}(t) + in_0(t), fb_1(t) = r_{11}(t) + in_1(t), fb_2(t) = r_{12}(t) + in_2(t), fb_3(t) = r_{13}(t) + in_3(t) \\ r_{00}(t+1) &= fb_3(t) + fb_2(t) + fb_1(t) \\ r_{01}(t+1) &= fb_3(t) + fb_2(t) \\ r_{02}(t+1) &= fb_3(t) \\ r_{03}(t+1) &= fb_3(t) + fb_2(t) + fb_1(t) + fb_0(t) \\ r_{10}(t+1) &= fb_2(t) + r_{00}(t) \\ r_{11}(t+1) &= fb_3(t) + fb_0(t) + r_{01}(t) \\ r_{12}(t+1) &= fb_1(t) + fb_0(t) + r_{02}(t) \\ r_{13}(t+1) &= fb_1(t) + r_{03}(t) \end{aligned}$$

2 Декодирующие устройства укороченных кодов Рида-Соломона

Для построения декодера укороченного КРС1 необходимо сначала найти остаток от деления полинома X^{p+i} , где i – параметр укорачивания кода, на порождающий полином. Рассмотрим особенности построения генератора синдрома на примере $i = 1$. $X^{p+i} = X^3$.

$$\begin{array}{r} X^3 \\ - X^3 + (z^2 + z)X^2 + z^3X \\ \hline (z^2 + z)X^2 + z^3X \\ - (z^2 + z)X^2 + (z^2 + z)X + (z^2 + z)z^3 \\ \hline (z^4 + z^3 + z^2)X + (z^5 + z^4) \end{array} \quad \begin{array}{l} X^2 + (z^2 + z)X + z^3 \\ \hline X + (z^2 + z) \end{array}$$

Приведем полученные коэффициенты остатка по модулю $p(z)$.

$$\begin{array}{r} z^4 + z^3 + z^2 \\ - z^4 + z^3 + 1 \\ \hline z^2 + 1 \end{array} \quad \begin{array}{l} z^4 + z^3 + 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} z^5 + z^4 \\ - z^5 + z^4 + z \\ \hline z \end{array} \quad \begin{array}{l} z^4 + z^3 + 1 \\ \hline z \end{array}$$

Таким образом, $R_{RS(X)}[X^3] = (z^2 + 1)X + z$. На рисунке 2 представлен декодер укороченного (14, 12) кода Рида-Соломона для поля Галуа $GF(2^4)$. Для этого декодера получаем следующие равенства для его реализации на элементах двоичной логики:

$$\begin{aligned} r_{00}(t+1) &= in_3(t) + r_{13}(t) + r_{12}(t) + r_{11}(t) \\ r_{01}(t+1) &= in_0(t) + r_{13}(t) + r_{12}(t) \\ r_{02}(t+1) &= in_1(t) + r_{13}(t) \\ r_{03}(t+1) &= in_3(t) + in_2(t) + r_{13}(t) + r_{12}(t) + r_{11}(t) + r_{10}(t) \\ r_{10}(t+1) &= in_3(t) + in_2(t) + in_0(t) + r_{12}(t) + r_{00}(t) \\ r_{11}(t+1) &= in_3(t) + in_1(t) + r_{13}(t) + r_{10}(t) + r_{01}(t) \\ r_{12}(t+1) &= in_2(t) + in_0(t) + r_{11}(t) + r_{10}(t) + r_{02}(t) \\ r_{13}(t+1) &= in_2(t) + in_1(t) + r_{11}(t) + r_{03}(t) \end{aligned}$$

Выводы

Рассмотрены способы схемной реализации укороченных кодов Рида-Соломона – от построения порождающих полиномов и выбора их формы для последующего синтеза схем кодеров и декодеров на элементах двоичной логики. С точки зрения согласования скоростей передачи кодером кодового слова и его декодирования, а также времени

формирования синдрома, первый способ уступает второму. И хотя второй способ требует предварительных расчетов, он является более предпочтительным.

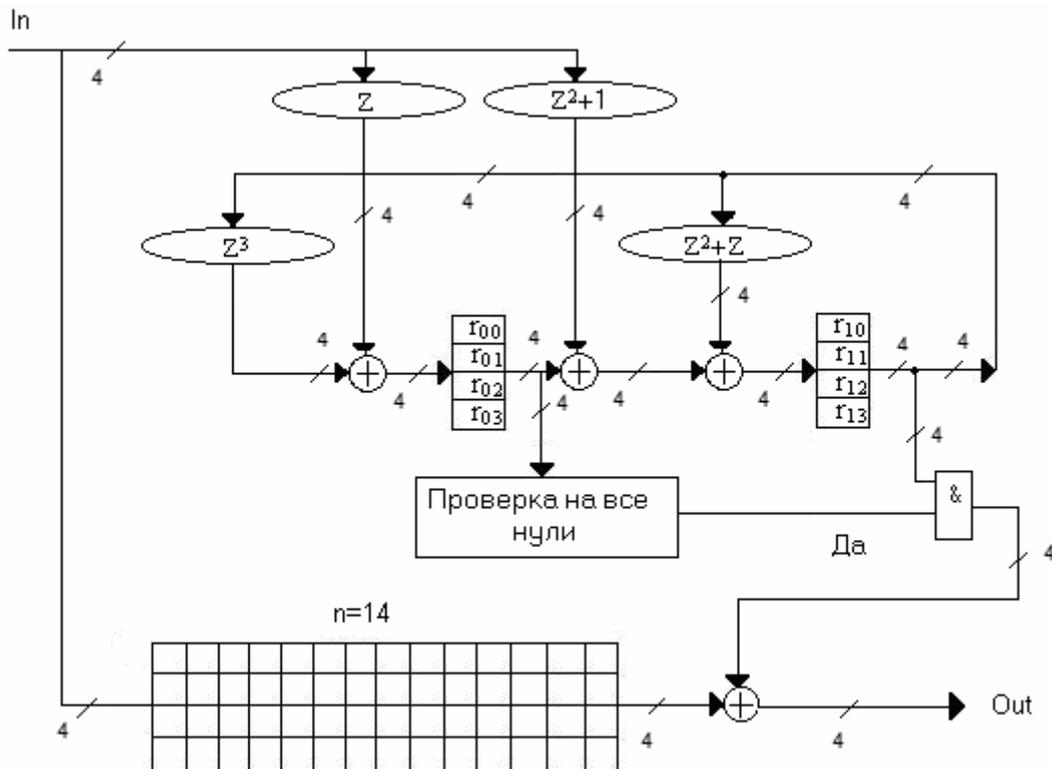


Рисунок 2 – Декодер укороченного кода Рида-Соломона для поля Галуа $GF(2^4)$

Полученные результаты могут оказаться полезными при разработке устройств, исправляющих пакетные ошибки. Кроме того, многие вопросы, связанные с построением кодов Рида-Соломона, которое использует математический аппарат алгебры полей Галуа, становятся более понятными при иллюстрации их конкретными примерами аппаратной реализации.

Литература

- [1] Reed I.S., Solomon G. Polynomial codes over certain finite fields // J. Soc. Industrial Appl. Math., 1960, vol.8, PP.300–304
- [2] Код Рида-Соломона. Материал из Википедии – свободной энциклопедии. Электронный ресурс. Режим доступа: http://ru.wikipedia.org/wiki/Код_Рида_-_Соломона
- [3] Алгоритмы и протоколы каналов и сетей передачи данных. Материал из Интернет-Университета Информационных Технологий. Электронный ресурс. Режим доступа: http://www.intuit.ru/department/network/algoprotnet/4/algoprotnet_4.html
- [4] Robert H. Morelos-Zaragoza. The Art of Error Correcting Coding. First Edition, John Wiley & Sons, 2002. – 221 p.
- [5] R.E.Blahut. Theory and Practice of Error Control Codes. Addison-Wesley Publishing Company, Massachusetts, 1984. – 576 p.
- [6] Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. - М.: Мир, 1976. – 595 с.