

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ DIGITAL PROCESSING OF SIGNALS AND IMAGES

УДК 004.67

П. С. Поперечный, аспирант ИППМ РАН, e-mail: ppoperechny@elvees.com
Институт проблем проектирования в микроэлектронике РАН

Применение цифровой фильтрации для реализации кодера Рида — Соломона

Предложен способ построения схем кодирования с помощью цифровых фильтров. Данный подход позволяет использовать способы цифровой фильтрации (разбиение на фильтры меньшего порядка, конвейеризация) в целях изменения корректирующей способности кода. Приведено описание данного способа для применения в кодировании РС (Рида — Соломона), аппаратно реализовано устройство кодера, приведены сравнительные характеристики.

Ключевые слова: РС, корректирующая способность, регистр с линейной обратной связью (РЛОС), КИХ-, БИХ-фильтры (фильтры с конечной/бесконечной импульсной характеристикой), поле Галуа

Введение

Коды РС (Рида — Соломона) относятся к блочному кодированию и широко используются в системах хранения и передачи информации. Данные коды позволяют исправлять множественные ошибки в блоках данных от нескольких битов до нескольких килобайтов (увеличение блока данных приводит к аппаратным сложностям) [1].

В данной статье предложен метод построения кодера с использованием цифровых фильтров. Хотя арифметика в данном кодировании построена на полях Галуа, основные методы цифровой фильтрации применимы и в конечных полях. Данный подход позволяет разбивать схему на последовательно соединенные фильтры первого порядка с возможностью конвейеризации в целях увеличения быстродействия. Также показано, что при таком подходе кодер легко становится реконфигурируемым, т. е. включением, отключением определенных фильтров в цепи можно изменять корректирующую способность кода.

В настоящее время данные коды массово используются в таких системах хранения информации, как твердотельные накопители, флэш-память и др. Таким образом, ввиду использования разных накопителей для работы с одним устройством необходимо применение кодов с разной корректирующей способностью соответственно. Например, для того чтобы в блоке данных используемый код позволял исправлять до 16 ошибок, необходимо применение определенного порождающего полинома определенной длины. Однако для того чтобы в этом же блоке данных код позволял исправлять например 12 ошибок, необходимо применение другого

порождающего полинома с меньшей длиной. То есть для использования одного и того же устройства с разными накопителями необходимо применение разных порождающих полиномов и, как следствие, разных кодеров, что приводит к увеличению аппаратных ресурсов. Однако использование кодера с регулируемой корректирующей способностью (переменный порождающий полином) может также удовлетворить различным требованиям к корректирующей способности [2].

Традиционный способ кодирования РС

Из определения кодов РС следует, что систематическое кодирование осуществляется следующим образом:

$$\frac{m(x)x^{2t}}{g(x)} = q(x) + \frac{r(x)}{g(x)}, \quad (1)$$

где $m(x)$ — входные незакодированные данные; $g(x)$ — порождающий полином; t — число исправляемых ошибок; $q(x)$ — частное от деления; $r(x)$ — остаток от деления на $g(x)$.

При этом результирующее кодовое слово (закодированные данные) в систематическом виде представляются как

$$c(x) = m(x)x^{2t} + r(x), \quad (2)$$

где $c(x)$ — кодовое слово.

Таким образом, данные на выходе кодера остаются неизменными, однако к ним добавляются контрольные данные $r(x)$.

Аппаратная реализация выражения (2) выполняется с помощью регистра с линейной обратной связью (РЛОС), представленного на рис. 1. В пер-

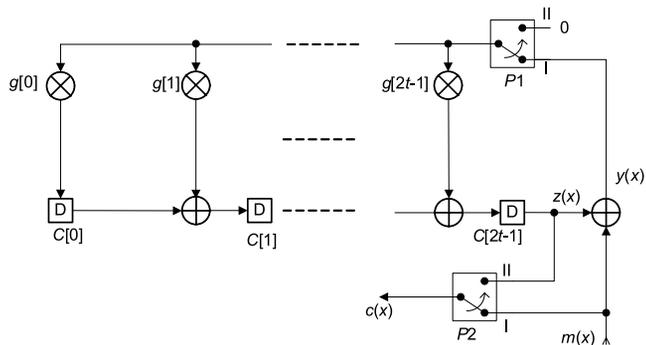


Рис. 1. Традиционная схема кодера РС

вые k тактов (k — число информационных символов) работы схемы данные проходят на выход схемы неизменными, при этом одновременно поступают на вход РЛОС, где с учетом обратной связи происходит вычисление остатка $r(x)$. После k тактов в схеме РЛОС отключается обратная связь, и значение остатка $r(x)$ фиксируется в сдвиговом регистре. В течение последующих $2t$ тактов из схемы выгружается значение остатка $r(x)$ и поступает на выход схемы.

При необходимости изменения требований к корректирующей способности кода РС необходимо изменить порождающий полином $g(x)$, что ведет к изменению схемы РЛОС. В статье предлагается способ построения схемы деления на порождающий полином с возможностью минимальными затратами изменять порождающий полином в процессе работы.

Способ реализации схемы РЛОС с помощью цифровых фильтров

Для реализации выражения (2) применяется схема РЛОС, представленная на рис. 1. Когда переключатели $P1, P2$ находятся в положении I, схема РЛОС становится похожей на схему БИХ-фильтра (фильтр с бесконечной импульсной характеристикой), с отличием в том, что выход кодера не совпадает с выходом БИХ-фильтра. А именно, выход РЛОС $z(x)$ выведен с регистра, однако в БИХ-фильтре с подобной реализацией выход данных $y(x)$ выведен с сумматора. Опишем математически работу схемы РЛОС при положении I переключателей $P1, P2$. Выход $y(x)$ БИХ-фильтра можно записать следующим образом:

$$y(x) = \frac{m(x)x^{2t}}{g(x)}. \quad (3)$$

Однако выход $z(x)$ РЛОС в случае кодирования отличается, так как выведен до сумматора, а именно:

$$z(x) = \frac{m(x)x^{2t}}{g(x)} + m(x). \quad (4)$$

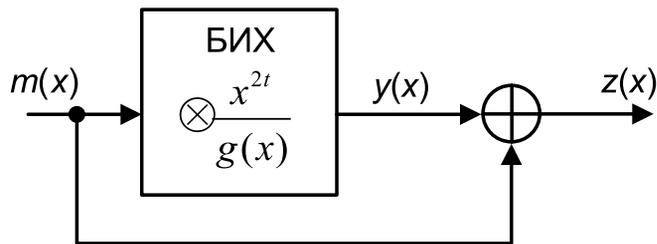


Рис. 2. Эквивалентная схема РЛОС с применением БИХ-фильтра

Эквивалентная схема, описываемая выражением (4), представлена на рис. 2.

Умножив и разделив последнее слагаемое $m(x)$ в выражении (4) на $\frac{x^{2t}}{g(x)}$, получим:

$$z(x) = \frac{m(x)x^{2t}}{g(x)} + \frac{m(x)x^{2t}}{g(x)} \frac{g(x)}{x^{2t}}. \quad (5)$$

Эквивалентная схема выражения (5) представлена на рис. 3.

Реализация конвейерного кодера РС с помощью цифровых фильтров

После включения переключателей $P1, P2$ в положение II в схеме на рис. 1 отключается цепь обратной связи, а на вход схемы подается ноль. В эквивалентной схеме на рис. 3 на вход КИХ-фильтра (фильтра с конечной импульсной характеристикой) вместо выхода БИХ-фильтра подается ноль. Так как в первые k тактов схемы на выход кодера должны выдаваться данные без изменения, а в последующие $2t$ тактов выдаются данные, вычисленные кодером, можно пренебречь сумматором в схеме 3, добавить переключатель $P3$, в результате получится эквивалентная схема всего кодера, представленная на рис. 4.

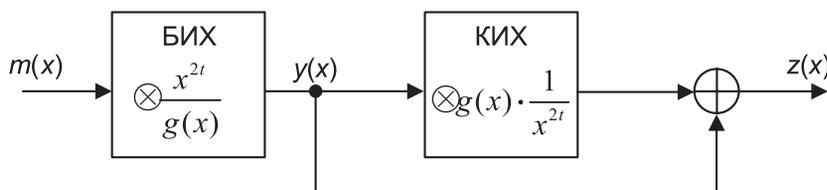


Рис. 3. Эквивалентная схема РЛОС с применением БИХ- и КИХ-фильтров

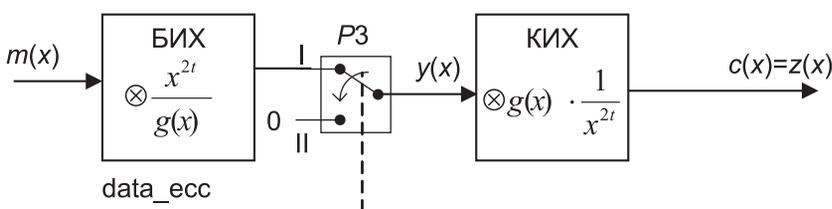


Рис. 4. Эквивалентная схема кодера РС с применением БИХ- и КИХ-фильтров

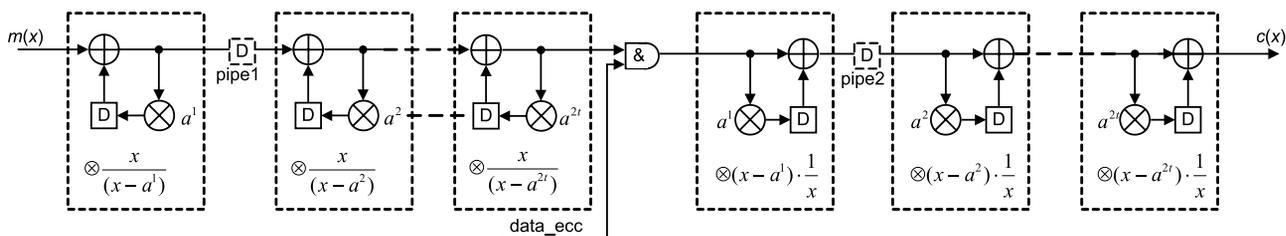


Рис. 5. Функциональная схема кодера РС с применением БИХ- и КИХ-фильтров с конвейеризацией

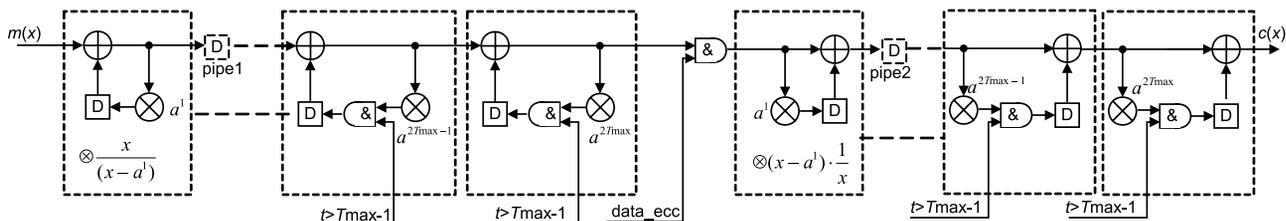


Рис. 6. Функциональная схема реконфигурируемого кодера РС с применением БИХ- и КИХ-фильтров с конвейеризацией

То есть в первые k тактов схемы на выход кодера поступают данные после БИХ- и КИХ-фильтров, соединенных последовательно, а так как оба фильтра имеют один и тот же полином, данные на выходе будут поступать без изменений, что и требуется по определению кодера систематического кода РС. Однако в эту фазу происходит вычисление, и после k -го такта БИХ-фильтр из цепи отключается переключателем $P3$ и продолжает работать лишь КИХ-фильтр, подавая на выход кодера значения контрольных символов.

Такое построение эквивалентной схемы (рис. 4) позволяет избежать общей цепи обратной связи, охватывающей всю схему РЛОС, как в схеме на рис. 1. Порождающий полином $g(x)$ для кодов РС представим в виде

$$g(x) = (x - a)(x - a^1) \dots (x - a^{2^t}), \quad (6)$$

где $a \in GF(2^m)$ — примитивный элемент поля Галуа размерности m .

Таким образом, схемы БИХ- и КИХ-фильтров можно разбить на последовательно соединенные фильтры первого порядка, как показано на рис. 5. А так как при таком построении схемы отсутствует общая цепь обратной связи с выхода на вход кодера, для уменьшения критического пути через цепь последовательно соединенных сумматоров можно конвейеризовать схему дополнительными регистрами, например $pipe1$ и $pipe2$ и т. д. при необходимости. При этом латентность схемы увеличивается на число тактов, равное числу дополнительных конвейерных регистров, и быстродействие становится крайне высоким.

Наилучшим применением схемы на рис. 5 получается при использовании дополнительного сигнала разрешения определенных фильтров. На рис. 6 представлена функциональная схема реконфигурируемого кодера РС с возможной конвейеризацией.

В представлении кодера в виде БИХ-фильтров, рассмотренном в работе [3], отсутствует основное преимущество подхода цифровой фильтрации. А именно, цифровые фильтры, передаточная функция которых описывается в виде множителей, можно представить в виде последовательной цепи фильтров меньшего порядка, как показано на рис. 5, 6.

Схема на рис. 6 наделяет код корректирующей способностью от максимально возможного числа исправляемых ошибок T_{max} до 1. Из выражения (6) следует, что для того чтобы код позволял исправлять t ошибок, необходимо порождать полином степени $2t$. Таким образом, схема реконфигурируемого кодера рассчитана на полином максимальной степени $2T_{max}$. При необходимости кодировать данные кодом с числом исправляемых ошибок меньше максимального T_{max} необходимо с помощью сигналов разрешения отключить определенные фильтры (т. е. часть множителей в полиноме (6) заменяются на 1), и результирующий полином получится необходимой степени, а значит, с заданной корректирующей способностью.

Процедура декодирования кодов РС

Декодирование кодов РС структурно показано на рис. 7. Принятые данные (с возможными ошибками) $v(x)$ поступают в схему декодера, одновременно

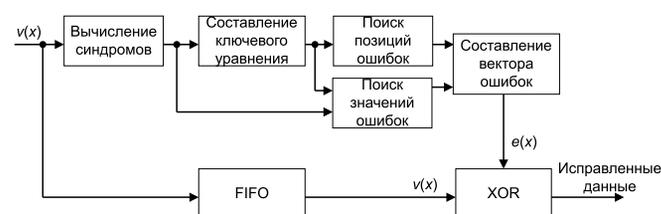


Рис. 7. Структурная схема декодирования кодов РС

менно происходит запись этих данных в буфер FIFO. Декодирование поделено на четыре основных этапа [4]. Сначала данные поступают в схему вычисления синдромов (признаки ошибок) в первые n тактов, потому как кодовое слово стало длины n . Дальнейшие вычисления декодер проводит с вычисленными синдромами. Следующий этап — вычисление полинома локаторов ошибок. Например, в алгоритме Берлекэмп — Мэсси (ВМА) без инверсии для данного этапа требуется $2t$ тактов (где t — число исправляемых ошибок, с которым было закодировано переданное кодовое слово). Далее вычисленные коэффициенты уравнения поступают в схемы поиска позиций ошибок и в схему поиска значений ошибок. После вычисления полинома значений ошибок этот полином вместе с указателями позиций ошибок из схемы поиска позиций ошибок подаются на схему составления вектора ошибок. Данная схема формирует маску $e(x)$ для исправления искаженных данных $v(x)$ при считывании из буфера FIFO. При фиксированном параметре m поля Галуа декодер, реконфигурируемый по числу исправляемых ошибок, реализуется изменением числа тактов, необходимых для работы алгоритма ВМА и схемы поиска значений ошибок. Таким образом, добавив к схеме декодера управление в зависимости от параметра t (максимальное число исправляемых ошибок), декодер становится реконфигурируемым в зависимости от корректирующей способности, с которой было закодировано кодовое слово.

Экспериментальные результаты

На основе предложенной методики построения схемы кодирования с помощью цепи БИХ- и КИХ-фильтров разработано аппаратное описание кодера РС. Проведены сравнения по занимаемой площади (число логических вентилях), а также по задержкам на критических путях для ПЛИС семейства Altera (Arria V). Представлена таблица полученных характеристик для кодеров в поле Галуа $GF(2^m)$ при $m = 8$ (длина закодированных данных не будет превышать $n < 2^8$) в зависимости от числа исправляемых ошибок. Анализ проведен в сравнении с каноническим представлением кодера.

Например, для $T = 16$ предложенный кодер будет кодировать данные наиболее типовыми кодами для хранения информации во flash-памяти с числом исправляемых ошибок 16, 15, 14 и т. д. согласно используемой flash-памяти, что делает его гораздо более универсальным в сравнении с каноническим кодером.

Результаты синтеза представлены на графике рис. 8.

Результаты экспериментов по синтезу показали следующее:

Ресурсы ПЛИС (число ячеек), задержка на критическом пути в зависимости от числа исправляемых ошибок канонического кодера (const) и кодера на базе КИХ-БИХ фильтров (var)

Число ошибок t	Число ячеек		Задержка, нс	
	const	var	const	var
2	43	85	2,19	2,75
4	79	204	2,24	3,23
8	145	464	2,49	4,11
16	274	996	2,76	5,23

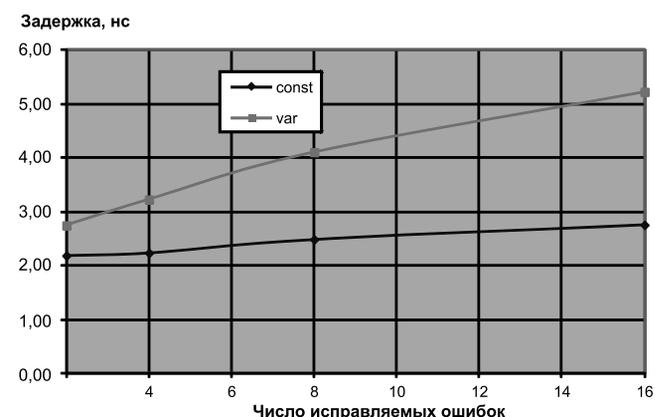
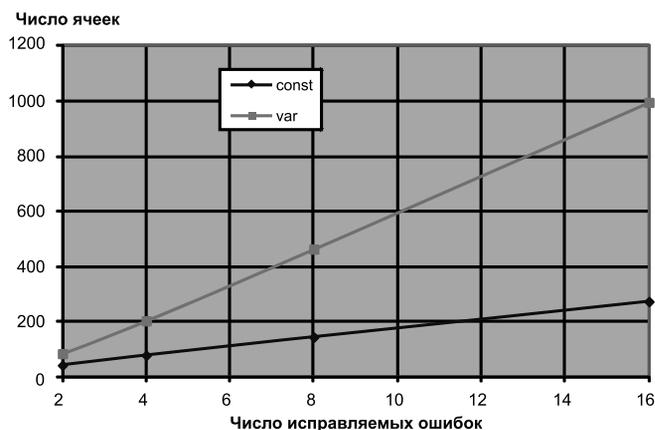


Рис. 8. Ресурсы ПЛИС (число ячеек), задержка на критическом пути в зависимости от числа исправляемых ошибок канонического кодера (const) и кодера на базе КИХ-БИХ фильтров (var)

- задержка на критическом пути предложенного кодера растет быстрее канонического в зависимости от числа исправляемых ошибок. Однако эксперимент проведен без использования конвейерных регистров. Очевидно, что с добавлением хотя бы одного конвейерного регистра (см. рис. 6) задержка уменьшится в 2 раза и быстродействие увеличится в сравнении с каноническим видом кодера;
- для конкретного семейства ПЛИС (Arria V) в среднем занимаемая площадь (в единицах логических ячеек) реконфигурируемого кодера в 2,5...3 раза больше кодера с постоянным множителем.

Заключение

В данной статье описан способ применения цифровой фильтрации для устройств блочного кодирования. Алгоритмы и способы цифровой фильтрации позволили преобразовать схему кодера РС в удобную последовательность цифровых фильтров. Такое построение схемы делает кодер реконфигурируемым под разную корректирующую способность.

На основе предложенного способа разработан кодер, обладающий преимуществами при определенных условиях, например, при заведомо неизвестном числе исправляемых ошибок. Показано, что разработанное устройство обладает большим быстродействием в сравнении с кодером традиционного вида.

При применении устройства, разработанного предлагаемым способом, для работы с flash-памя-

тью различных типов более 3 делает его выгодным по занимаемой площади. По быстродействию при добавлении конвейерных регистров есть выигрыш в сравнении с устройствами канонического вида, где невозможно сделать конвейеризацию ввиду общей обратной связи [5].

Список литературы

1. Блейхут Р. Теория и практика кодов, контролирующих ошибки. М.: Мир, 1986.
2. Yoo H., Lee Y., 7.3 Gb/s Universal BCH Encoder and Decoder for SSD Controllers. IEEE 978-1-4799-2816-3. 2014, pp. 37–38.
3. Кириллов С. Н., Семин Д. С. Модифицированный помехозащищенный кодер на основе БИХ-фильтра // Вестник РГРТУ. 2009. № 2.
4. Вернер М. Основы кодирования. М.: Техносфера, 2004. 288 с.
5. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. М.: Техносфера, 2005.

P. S. Poperechny, Postgraduate,
Institute for Design Problems in Microelectronics (IPPM RAS)

Digital Filtering Application for Reed-Solomon Coder Implementation

This article proposes a method for coding circuit development by means of digital filtering. This approach allows to use digital filtering methods (cutting to small-order filters, pipe-lining) to change correcting level capability. The well-known coder scheme for RS includes linear feedback shift register (LFSR), which is very similar to TIR-filter (infinite impulse response). So, it is possible to use digital filtering for coder. Here are algebraic and scheme presentations of this approach. It is shown that we can not just change LFSR by IIR — filter, because both of them have different outputs. As well the coder works at different modes for input data and for output. But if we will use FIR-filter (finite-impulse response) serial-connected to IIR we can overcome these issues. This approach allows using all digital filtering advantages, such as to split one big order filter to many first-order filters.

Exactly this property is very useful for reconfiguration of the RS-coder for differ applications. It means that we can use only one proposed coder for any purposes, for example for different flash-memory types rather than many different coders. Here are descriptions of this method for using in RS, coder hardware description, comparing of synthesis results on FPGA with traditional coder.

Keywords: RS, correcting level capability, linear feedback shift register (LFSR), FIR-, IIR-filters (finite/infinite impulse response filters), Galois field

References

1. Blahut R. *Teorija i praktika kodov, kontrolirujushih oshibki*. Moscow: Mir, 1986.
2. Yoo H., Lee Y. 7.3 Gb/s Universal BCH Encoder and Decoder for SSD Controllers, IEEE 978-1-4799-2816-3, 2014, pp. 37–38.
3. Kirillov S. N., Semín D. S. Modificirovannij pomehozashishennij koder na osnove BГH-filtra, *Vestnik RGRTU*, Rjazan, 2009, no. 2.
4. Verner M. *Osnovi kodirovanija*. Moscow: Tehnosfera, 2004. 288 p.
5. Morelos-Saragosa R. *Iskusstvo pomehoustojchivogo kodirovanija. Metodi algoritmi primenenije*. Moscow: Tehnosfera, 2005.