ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ КОДЕКА РАСШИРЕННОГО КОДА РИДА-СОЛОМОНА

Бодян Д. Г., Бодян Г. К., Шестакова Т. В.
Технический университет Молдовы
ул. Шт. чел Маре 168, Кишинев, MD2012, Р. Молдова
тел.: +37322-237505, e-mail: gbodean@mail.utm.md

Аннотация – Представлены проектные решения и VHDL-модули последовательного кодера и декодера (кодека) расширенного кода Рида-Соломона. Декодер реализует модифицированный алгоритм Евклида деления полиномов. В работе уделяется особое внимание алгоритму управления блоками решения ключевого уравнения. Представлены результаты проектирования кодека для скорости передачи информации 3/4 и корректирующей способности равной 32.

I. Введение

С появлением программируемых интегральных схем (PLD) высокой степени интеграции, стала возможной компактная реализация в “домашних условиях” такого сложного устройства, как кодер-декодер (кодек) кода Рида-Соломона (РС). Наиболее сложным (с вычислительной точки зрения) компонентом кодека является блок декодирования РС-кода. Из трех известных алгоритмов декодирования РС-кодов - Берлекампа-Месси, Питерсона-Горенштейна-Цирлера и Евклида [1], для аппаратной реализации чаще всего выбирают алгоритм Евклида [2-4].

Фирмы, в том числе производители PLD, предлагают программные HDL-прототипы (модули интеллектуальной собственности), которые позволяют генерировать проекты РС-кодеков с требуемыми параметрами. Но все предлагаемые разработки ограничиваются длиной слова $n \le 2^m-1$. На сегодняшний день не известны HDL-проекты и аппаратная реализация кодеков т.н. расширенного РС-кода (ХРС), т.е. кода с длиной слова $n=2^m$ и $n=2^m+1$. А эти случаи являются важными с практической точки зрения!

В работе предлагается и анализируется VHDL-проект ХРС-кодека с евклидовым декодированием. Рассмотрены алгоритмические особенности блока решения ключевого уравнения, а также представлены результаты проектной реализации ХРС-кодеков.

II. Основная часть

При анализе РС-кодов, как правило, ограничиваются длиной слова $n \le 2^m-1$, где m – разрядность символов. Но в коммуникационных системах, особенно где используется пакетная обработка данных, требуются значения n кратные степени двойки, т.е. $n=2^m$. В известном стандарте мультимедийного и телевизионного вещания DBV-H данные обрабатываются по-байтно, т.е. $m=8$, а корректирующий код должен обеспечить исправление до 32 ошибочных символов, т.е. $t=32$. При таких ограничениях, примем параметры ХРС-кода равными $(n, k, t)= (256, 192, 32)$.

Проектирование РС-кодека начинается с определения контрольной матрицы **H**, которая для ХРС-кодов может быть представлена в виде:

$$\mathbf{H}=\begin{bmatrix} 0 & 1 & \alpha^{1} & \dots & \alpha^{n-2} \\ 0 & 1 & \alpha^{2} & \dots & \alpha^{2(n-2)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 1 & \alpha^{2t-1} & \dots & \alpha^{(2t-1)(n-2)} \\ 1 & 1 & \alpha^{2t} & \dots & \alpha^{2t(n-2)} \end{bmatrix}, \text{ где } \alpha^{i} \in [2..2^m-1]. \qquad (1)$$

ХРС-кодер формирует $2t$ контрольных символа, которые присоединяются к информационной части слова, и содержит классический РС-кодер, на выходе которого добавлен т.н. базовый программный элемент (BPE), выполняющий операцию аддитивно-мультипликативного аккумулирования порождаемых символов (рис. 1).

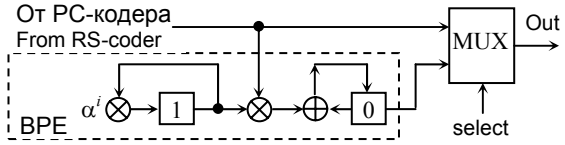

Рис. 1. Схема вспомогательного блока ХРС-кодера.

Fig. 1. Diagram of the additional XRS-coder module

На рис. 1 символ □ означает элемент памяти, символ ⊕ - побитовый XOR-сумматор, символ ⊗ - универсальный мультипликатор, а сочетание $\alpha^i\otimes$ - мультипликатор на константу.

Блок вычисления синдрома ХРС-декодера построен по классической итеративной схеме, на основе сдвигового регистра типа FIFO [1-4]. Но к выходу младшей позиции FIFO подключен мультиплексор MUX, который коммутируется на ‘0’ (GND) при вычислении последней компоненты синдрома $\mathbf{S}(x)$. Если $\mathbf{S}(x)\neq 0$, то переходят к составлению и решению ключевого уравнения.

В анализируемом проекте применяется евклидов алгоритм вычисления наибольшего общего делителя двух полиномов. На рис.2 представлена схема блока деления полиномов, который содержит два регистра - верхний и нижний. Вначале процедуры деления в верхний регистр загружаются компоненты синдрома $\mathbf{S}(x)$, а разряды нижнего регистра, кроме самого старшего разряда, устанавливаются в ‘0’. Старший разряд устанавливается в ‘1’.

Один такт сдвига – это выполнение одного шага итеративного деления полиномов, в результате которого в нижнем регистре получается остаток $\mathbf{r}(x)$, а в верхнем регистре – частное $\mathbf{q}(x)$. Отметим, что старший разряд верхнего регистра должен быть всегда значим, чтобы операция деления имела смысл! После каждого сдвига, значения коэффициентов $\mathbf{r}(x)$ должны поменяться местами со значениями коэффициентов $\mathbf{q}(x)$. Но перед тем, как выполнить операцию обмена, необходимо проверить степень (Deg) остатка $\mathbf{r}(x)$.

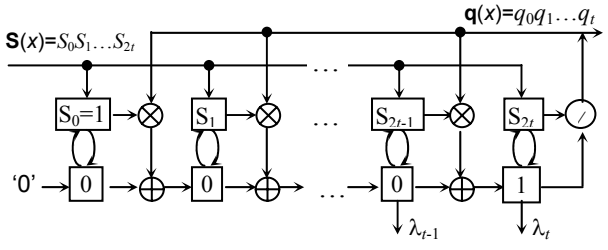

Рис. 2. Схема блока деления полиномов.

Fig. 2. Diagram of the polynomial division module

Операции “сдвиг-обмен” повторяются до тех пор, пока не будет достигнуто условия Deg $\mathbf{r}(x)\le t$. При

 ISBN: 978-966-335-014-1. IEEE Catalog Number: 07EX1742

этом коэффициенты частного **q**(*x*) поступают *последовательно* на вход блока вычисления свертки, а коэффициенты остатка **r**(*x*) подаются *параллельно* на вход блока вычисления полинома ошибок $\Lambda(x)$. Здесь, также, необходимо обеспечить выравнивание коэффициентов остатка **r**(*x*) по соответствующим позициям нижнего регистра!

Вышеперечисленные особенности стали основанием для разработки автоматной (VHDL-) модели блока управления и контроля ходом выполнения евклидового алгоритма. Отличительная особенность разработанной модели – ее универсальность.

Все блоки XPC-кодека разработаны в виде настраиваемых VHDL-модулей. Чтобы проследить динамику имплементирования, были сгенерированы XPC-кодеки скорости $\boldsymbol{R}$=3/4 и длины слова $n=2^m$ для различных значений *m*. В таблице 1 представлены результаты имплементирования проектов на PLD EP3SL70F484C2 семейства Stratix III фирмы Altera.

Таблица 1.

Ресурсы, затраченные на реализацию XPC-кодеков.

The resources required for implementation of XRS-codecs

m	Coder Unit	Syndrome Unit, ALUT	Euclid Unit	Convolution Unit	F_{max}, MHz	v_{max}, Mbps
4	37	182	79	87	220,46	881.8
5	69	578	168	302	179,4	897
6	139	1649	417	1151	146,2	877.2
7	287	4429	1064	4938	134,12	938.8
8	595	11598	2946	22136	95,2	761.6

F_{max} - максимальная частота работы XPC-кодека;
v_{max} - результирующая скорость передачи данных.

III. Заключение

Выполненные проектные работы являются заключительным этапом по исследованию и разработке средств кодирования и декодирования матроидных кодов [5]. Как было показано ранее [6], матроидные (M-) коды являются обобщением кодов Рида-Соломона. Также, в работе [6] был представлен M-кодек с *параллельным* (однотактным) декодированием, а в данной работе представлены результаты проектирования *последовательного* M-кодека как следствие расширения средств кодирования-декодирования PC-кодов.

IV. Список литературы

[1] *Morelos-Zaragoza R. H..* The Art of Error Correcting Coding. John Wiley & Sons, 2006, 264 P.

[2] *Chang H.-C., Chung C.-C., Lin C.-C., and Lee C.-Y.* A High Speed Reed-Solomon Decoder Chip Using Inversionless Decomposed Architecture for Euclidean Algorithm. Proc. ESSCIRC 2002, pp. 519-522.

[3] *Lee H..* An Ultra High-Speed Reed-Solomon Decoder. IEEE ISCAS, 2005, Vol. 2, pp. 1036-1039.

[4] *Hsu H.-Y., Yeo J.-C., and Wu A.-Y.* Ultra Low-Cost 3,2 Gb/s Optical Rate Reed-Solomon Decoder IC Design. Asian Solid-State Circuits Conference, 2005, pp. 533-536.

[5] *Бодян Г. К.* Об одном методе помехозащищенного кодирования. КрыМиКо'2003: Материалы конференции. — Севастополь: Вебер, 2003, с. 357-358.

[6] *Бодян Г. К., Бодян Д. Г., Дунай Л. Ф.* Альтернативное помехозащищенное кодирование в мобильных системах связи. КрыМиКо'2006: Материалы конференции. — Севастополь: Вебер, 2006, с. 369-370.

PRACTICAL IMPLEMENTATION OF THE EXTENDED REED-SOLOMON CODEC

Bodyan D. G., Bodyan G. C., Shestakov T. V.
Technical University of Moldova
168, St. cel Mare str., Kishinau, MD2012, R. Moldova
Ph.: +37322-237505. E-mail: gbodean@mail.md

Abstract – Architecture and VHDL-entities of the sequential coder and the decoder (codec) of the extended Read-Solomon code are presented. Decoder implements the modified Euclidean algorithm of polynomials division. In the present paper the special attention is given to control algorithm of the key-equation solution blocks. Results of codec designing for rate of 3/4 and correcting ability of 32 are presented.

I. Introduction

Programmable Logic Devices (PLD) allow implementing in the "home conditions" such complex device as the coder-decoder (codec) of the Read-Solomon (RS) code. The most complex component of the RS-codec is a decoding block. Three algorithms of RS-codes decoding are known: Berlekamp-Massey, Peterson-Gorenstein-Zierler and Euclidean [1]. The Euclidean algorithm is chosen to implement in hardware [2-4] more often.

The all-known RS-codec developments are limited to codeword length $n \le 2^m-1$. For today, HDL-designs and hardware implementations of codecs of the so-called extended RS-code (XRS), i.e. a code with code length $n=2^m$ and $n=2^m+1$ are not known. But these cases are very important from the practical point of view!

In this paper the VHDL-design of the XRS-codec with Euclidean decoding is analyzed. Algorithmic features of the Key-Equation Solver block are considered. Also results of design implementation of XRS-codecs are presented.

II. Main Part

In the known standard of multimedia and television broadcasting DBV-H, data are processed byte-by-byte, i.e. *m*=8, and the error-correcting code should provide correction up to 32 erroneous symbols, i.e. *t*=32. So, further we shall accept parameters of the XRS-code equal (*n*, *k*, *t*) = (256, 192, 32). It is easy to calculate that the data rate $\boldsymbol{R}$ is equal to 3/4.

Designing of the RS-codec begins with definition of control matrix **H** that for XRS-codes can be presented as in (1).

XRS-coder performs systematic encoding, i.e. generates 2*t* control symbols, and contains the classical RS-coder with the so-called base program element BPE (fig. 1) being attached. In Fig. 1 symbol □ designates a memory, symbol ⊕ means the bitwise XOR, symbol ⊗ specifies the universal multiplier, and combination $\alpha^i\otimes$ means the constant multiplier.

Euclidean algorithm is applied for computation the greatest common divisor of two polynomials in the analyzed design. Fig. 2 presents the scheme of the division block, which contains two registers called a *top* and a *bottom*. Procedure of division begins by loading components of syndrome S(*x*) into the top register and the stages of bottom register, excepting the most significant stage (MSS), are set in '0'. MSS is set up to '1'.

One shift is a one step of iterative division of polynomials. As a result in the bottom register the rest **r**(*x*) is obtained, and in the top register – the quotient **q**(*x*). Operations "shift-exchange" will be repeated until the conditions Deg **r**(*x*) ≤ *t* reached. All this features became the basis for designing the universal Euclidean control unit.

Table 1 shows the results of XRS-codecs implementation on PLD EP3SL70F484C2 of family Stratix III from Altera

III. Conclusion

Carried out design work is the final stage of the research and development of means of coding and decoding matriod codes [5]. As it has been shown earlier [6], matroid (M-) codes are generalization of RS-codes. In paper [6] the M-codec with *parallel* (on-shot) decoding has been presented, and in this paper results of designing *of sequential* M-codec as extension consequences of RS-codes coding-decoding means are presented.

2007 17th Int. Crimean Conference "Microwave & Telecommunication Technology" (CriMiCo'2007). 10-14 September, Sevastopol, Crimea, Ukraine