УДК 621.391.037

Г.М. Тамразян, А.А. Гладких, Д.В. Ганин

АППАРАТНАЯ РЕАЛИЗАЦИЯ ОПТИМАЛЬНОГО ДЕКОДЕРА НИЗКОПЛОТНОСТНЫХ КОДОВ

Тамразян Георгий Михайлович, *аспирант кафедры «Телекоммуникации» Ульяновского государственного технического университета, окончил УлГТУ. Имеет публикации и патенты в области мягкого декодирования избыточных кодов. [e-mail: tamrazz@bk.ru].*

Гладких Анатолий Афанасьевич, *кандидат технических наук, окончил Военную академию связи им. С.М. Буденного, адъюнктуру ВАС. Профессор кафедры «Телекоммуникации» Ульяновского государственного технического университета. Имеет монографию, учебные пособия, статьи и патенты РФ в области помехоустойчивого кодирования и защиты информации. [e-mail: a.gladkikh@ulstu.ru].*

Ганин Дмитрий Владимирович, *кандидат экономических наук, доцент, окончил Нижегородскую государственную сельскохозяйственную академию. Заведующий кафедрой «Инфокоммуникационные технологии и системы связи» Нижегородского государственного инженерно-экономического университета. Имеет статьи в области инфокоммуникаций. [e-mail: ngiei135@mail.ru].*

Аннотация

В современных инфокоммуникационных системах все большее применение стали находить коды с низкой плотностью проверок (Low Density Parity Check – LDPC) за счет своей корректирующей способности. В настоящее время LDPC-коды максимально приблизились к порогу Шеннона. Кроме того, использование таких кодов, в отличие от турбо-кодов, не имеет ограничений, связанных с патентами. Эти факторы послужили причиной растущего интереса к низкоплотностным кодам. Несмотря на относительную простоту реализации такого кодека, мягкое декодирование LDPC-кода имеет значительную вычислительную сложность. В данной работе подробно рассмотрены основные проблемные области, связанные с аппаратной реализацией LDPC-кодека, а также пути их преодоления. В работе продемонстрированы результаты моделирования различных способов реализации декодера и их сравнение. Кроме того, представлен метод списочного декодирования LDPC-кода, который значительно снижает вычислительную нагрузку на декодер и ускоряет его работу. Использование тех или иных техник и алгоритмов, показанных в данной работе, позволит разработать оптимальный декодер низкоплотностного кода, спроектированного под определенную задачу.

Ключевые слова: LDPC-код, мягкий декодер, ПЛИС, кластер, граф Таннера, списочное декодирование.

HARDWARE IMPLEMENTATION OF THE OPTIMAL LDPC DECODER

Georgii Mikhailovich Tamrazian, *Post-Graduate Student at the Department of Telecommunications of Ulyanovsk State Technical University; graduated from Ulyanovsk State Technical University; an author of articles and patents in the field of redundant code soft decoding. e-mail: tamrazz@bk.ru.*

Anatolii Afanasievich Gladkikh, *Candidate of Engineering; graduated from S.M. Budyonny Military Communications Academy; finished his post-graduate studies at the same academy; Professor at the Department of Telecommunications at Ulyanovsk State Technical University; an author of a monograph, textbooks, articles, and patents in the field of noiseless coding and information security. e-mail: a.gladkikh@ulstu.ru.*

Dmitrii Vladimirovich Ganin, *Candidate of Economics, Associate Professor; graduated from Nizhny Novgorod State Agricultural Academy; Head of the Department of Infocommunication Technologies and Telecommunications at Nizhny Novgorod State University of Engineering and Economics; an author of articles in the field of infocommunications. e-mail: ngiei135@mail.ru.*

Abstract

Low Density Parity Check (LDPC) codes become more useful in modern infocommunication systems because of their error-correcting capability. At the present time, LDPC codes have reached Shannon's limit. Moreover, the application of such codes unlike the turbo codes don't have any license limitations. These factors have become the cause of growing interest to LDPC codes. Despite an easy way of the codec implementation, soft decoding of LDPC codes is a complex computational process.

This article deals in more detail with the main problems concerning the hardware implementation of LDPC decoder and the ways of their solving. It also demonstrates the simulation results of different ways to decoder implementation and the comparison of these ways. Furthermore, the article presents the method of LDPC codes list decoding that reduces materially computational load on decoder and speeds up its work operation. The use of various procedures and mechanisms described in the article will help to generate the optimal LDPC code decoder designed for a certain task.

Key words: LDPC code, soft decoder, FPGA, cluster, Tanner graph, list decoding.

Введение

В 1962 году Галлагер в своей работе ввел новый класс линейных кодов, известный теперь как коды с низкой плотностью проверок. Но реализовать такой код на элементной базе тех времен было невозможно, и он долгое время оставался красивой теорией. Только с развитием современной электроники, в частности систем на кристалле (System-on-a-Chip – SoC), эти коды стали находить свое практическое применение. В настоящее время LDPC-коды включены во многие известные стандарты передачи данных, где существенна высокая скорость, надежность, а также экономия энергии. Наиболее известны из них – WiMAX, 10G Ethernet, DVB-S2, DVB-T2 и многие другие. LDPC-коды активно вытесняют некогда популярные турбо-коды за счет своей высокой корректирующей способности при относительно низкой сложности реализации кодека. LDPC-коды, как и турбо-коды, используют итеративные методы декодирования, однако декодирование может выполняться параллельно, что существенно снижает сложность декодера и повышает его быстродействие. В настоящее время нерегулярный низкоплотностный код является лучшим из известных двоичных кодов, приблизившихся к порогу Шеннона на рекордные 0.004 дБ [1].

Способы представления LDPC-кодов

Матричное представление.

Как и любой линейный блочный код, LDPC-код можно описать с помощью порождающей матрицы G, размером $k \times n$, где k – длина информационной последовательности, а n – длина кодового блока. Тогда получение кодового вектора C будет осуществляться умножением информационной последовательности m на порождающую матрицу G:

$$C = m \cdot G. \quad (1)$$

Для систематического LDPC-кода порождающую матрицу можно представить в виде $G = [I, P]$, где I – единичная $k \times k$ матрица. Тогда код будет описываться проверочной матрицей $H = [P^T, I]$, причем

$$C \cdot H^T = 0. \quad (2)$$

Элементами такой матрицы являются коэффициенты проверочных уравнений, из которых вычисляются проверочные символы. Для эффективных LDPC-кодов проверочная матрица H должна быть разряженной, и плотность единиц в ней, как правило, составляет несколько десятков на сотни тысяч элементов. Именно благодаря матричному представлению низкоплотностные коды получили свое название.

Графическое представление.

Для любого линейного (N, K) кода существует двудольный граф, инцидентный матрице H. Этот граф известен как граф Таннера, названный так по имени его открывателя. Граф Таннера состоит из N кодовых вершин и $M = N - K$ проверочных вершин. Кодовые и проверочные вершины соединяются ребрами согласно проверочной матрице H. Пример построения такого графа для матрицы

$$H = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

приведен на рисунке 1.

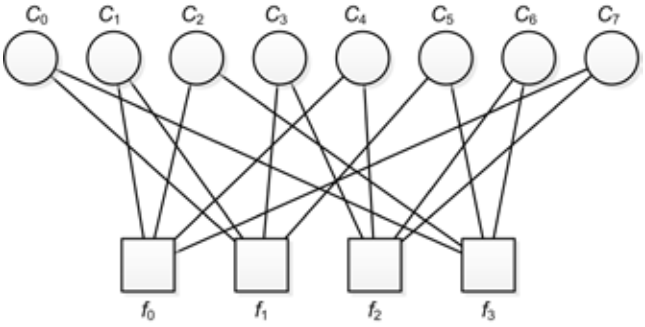

Рис. 1. Двудольный граф Таннера

Такое представление LDPC-кода позволяет хранить в памяти кодека только ребра и значения вершин графа, в отличие от матричного представления, в котором приходится выделять память под всю проверочную матрицу H. Кроме того, графическое представление позволяет лучше понять устройство и принципы работы аппаратно реализуемого LDPC-декодера, поэтому в дальнейшем описании будут использоваться термины графического представления кода.

Потенциальные проблемы при аппаратной реализации LDPC-декодера

При аппаратной реализации LDPC-кодека наибольшую сложность и интерес представляет реализация декодера. Наиболее эффективны алгоритмы, которые работают с мягкими решениями, что дополнительно усложняет декодер, так как обрабатывать в этом случае приходится не только двоичное значение сигнала, но и его метрику надежности.

Реализовываться декодер будет на ПЛИС (программируемая логическая интегральная схема) фирмы ALTERA семейства Cyclone II. Такой выбор объясняется приемлемым соотношением цены и качества, однако он ограничивает возможности использования сложных математических функций. Так как в ПЛИС в отличие от микропроцессоров отсутствуют блоки аппаратного вычисления тригонометрических функций, извлечения корня, логарифма и так далее, в работе будут рассматриваться декодеры с приближенным вычислением метрик, пригодные для реализации на ПЛИС [2].

Числовое представление метрик.

Первая задача, которую необходимо решить при проектировании мягкого LDPC-декодера, это числовое представление метрик. От этого решения зависит то, как будет вести себя каждый элемент устройства [3].

Метрики представляют собой значения, полученные из принятого сообщения и прошедшие через SumProduct алгоритм. Этот алгоритм подразумевает получение вещественных значений метрик, что неоправданно усложняет декодер. Кроме того, эти значения могут быть как положительными, так и отрицательными. Поэтому компромисс между точностью метрик и сложностью проектируемого арифметико-логического устройства – использование комплементарного представления метрики с фиксированной точкой. Ссылаясь на результаты исследования, приведенные в [3], выберем значение разрядности метрики 8 бит, при этом принимаемые значения будут лежать в пределах от +7.9375 до -7.9375 включительно и изменяться с шагом 0.0625.

Распространение доверия.

Алгоритм декодирования LDPC-кода можно свести к итеративному обновлению вершин графа Таннера и получению итоговых метрик. Значения надежностей проверочных вершин пересчитываются за счет ассоциированных с ними кодовых вершин, а кодовые – за счет проверочных согласно графу. Причем эти операции независимы друг от друга и могут выполняться параллельно. Одним из способов осуществления данной процедуры является итеративный алгоритм распространения доверия (Iterative Belief Propagation – IBP), также известный как алгоритм сумма произведений (SumProduct). Алгоритм итеративно вычисляет два типа условных вероятностей:

- q_{ij}^{x} – вероятность того, что j-й бит принятого вектора имеет значение x по информации, полученной от всех проверочных вершин графа Таннера, кроме вершины i;
- r_{ij}^{x} – вероятность того, что уравнение, соответствующее проверочной вершине i, удовлетворяется, если значение j-го бита равно x, а остальные биты независимы с вероятностями q_{ij}.

Алгоритм IBP можно модифицировать так, чтобы использовать логарифм отношений правдоподобия (Log Likelihood Ratio – LLR) вместо вероятностей [1]. Такой подход позволяет снизить сложность проектируемого устройства за счет перехода от умножения вероятностей к сложению LLR. Таким образом, надежности кодовых вершин будут рассчитываться согласно выражению (3), а проверочных – согласно (4):

$$Q_{ij} = \lambda_j + \sum_{\alpha \in C[j] \alpha \neq i} R_{\alpha j}[k-1], \tag{3}$$

$$R_{ij} = 2\,tanh^{-1} \cdot \prod_{\alpha \in V[j] \alpha \neq i} tanh\left(\frac{Q_{\alpha j}}{2}\right), \tag{4}$$

где Q_{ij} – LLR надежность i-го кодового бита, участвующего в расчете j-го проверочного выражения, при условии, что i-й бит равен 1; λ_i – LLR априорной вероятности принятого кодового бита; C_j – набор проверочных вершин, ассоциированных с кодовой вершиной j; R_{ij} – LLR вероятность того, что j-е проверочное условие выполняется, если i-й бит равен 1; V_j – набор кодовых вершин, ассоциированных с проверочной вершиной j.

Нелинейная ψ-функция.

Как видно из выражения (4), вычисление проверочных вершин представляет наибольшую сложность при аппаратной реализации, так как содержит произведение и тригонометрические функции. Избавиться сразу от двух проблем можно введением новой ψ-функции.

$$\psi(x) = -ln\left(tanh\left|\frac{x}{2}\right|\right) = ln\frac{\left(1+e^{-|x|}\right)}{\left(1-e^{-|x|}\right)}. \tag{5}$$

Так как $tanh(x)$ – четная функция, получим

$$tanh\left|\frac{x}{2}\right| = tanh\left|\frac{x}{2}\right| \cdot \delta_{ij}, \tag{6}$$

где δ_{ij} – знак, полученный перемножением знаковых разрядов сообщений.

Учитывая, что $tanh^{-1}(x) = \frac{1}{2} ln\left(\frac{1+x}{1-x}\right)$, и согласно (5) и (6) получим следующее выражение для расчета метрики проверочной вершины графа:

$$R_{ij}[k] = 2\,tanh^{-1} \cdot \left(exp\left(-\sum\nolimits_{\substack{\alpha \in V[j], \\ \alpha \neq i}} \psi\left(Q_{\alpha j}[k-1]\right)\right) \cdot \delta_{ij}\right) =$$

$$= ln\frac{1+exp\left(-\sum\nolimits_{\substack{\alpha \in V[j], \\ \alpha \neq i}} \psi\left(Q_{\alpha j}[k-1]\right)\right)}{1-exp\left(-\sum\nolimits_{\substack{\alpha \in V[j], \\ \alpha \neq i}} \psi\left(Q_{\alpha j}[k-1]\right)\right)} \cdot \delta_{ij},$$

$$R_{ij}[k] = \psi\left(\sum\nolimits_{\substack{\alpha \in V[j], \\ \alpha \neq i}} \psi\left(Q_{\alpha j}[k-1]\right)\right). \tag{7}$$

Введенная в (5) ψ-функция нелинейна, что представляет существенные трудности при нахождении ее значения. Существуют три подхода к аппаратной реализации данной функции, описанные в [4, 5]:

- кусочно-линейная аппроксимация (Piece-wise Linear – PWL),
- аппроксимация на основе двоичной арифметики,
- использование таблиц преобразований (Lookup Table – LUT).

Таблица 1

LUT-значения $\psi(x)$-функции

x	$\psi(x)$	x	$\psi(x)$	x	$\psi(x)$	x	$\psi(x)$
00000000	1111111	00000100	0100000	00001000	0011000	00001100	0010000
00000001	1001111	00000101	0011110	00001001	0010110	00001101	0001111
00000010	0101111	00000110	0011100	00001010	0010100	⋮	⋮
00000011	0101000	00000111	0011010	00001011	0010010	11111111	0000000

На этапе моделирования аппроксимация на основе двоичной арифметики показала неудовлетворительные результаты, затратив неоправданно большие вычислительные ресурсы, поэтому данный способ в дальнейшем рассмотрении участвовать не будет.

Таблицы преобразований (LUT) представляют собой набор заранее заготовленных значений ψ-функции, хранящихся в памяти декодера. Такой метод аппроксимации является наименее требовательным к вычислительным ресурсам кодека и имеет наилучшие скоростные характеристики. К его недостаткам можно отнести ограниченную точность получаемых значений, из-за чего функция чувствительна к выбранному представлению входных данных. При увеличении точности увеличиваются размер таблицы и, соответственно, объем необходимой памяти под ее хранение. Для выбранной точности метрик LUT-таблица будет состоять из 256 7-разрядных элементов. В таблице 1 приведена часть LUT-таблицы в двоичном виде.

PWL-аппроксимация представляет собой ломаную линию, заменяющую все нелинейные участки ψ-функции прямолинейными отрезками. Повышение точности аппроксимации увеличивает количество отрезков и, соответственно, сложность декодера. Коэффициенты уравнения данной линии будут выбираться с учетом простоты их представления в формате числа с фиксированной точкой, а также во избежание сложных операций умножения. Для выбранных метрик PWL-аппроксимация примет вид (8).

$$\psi(x)=\begin{cases}-48x+7.9375 & при \quad |x|<0.125,\\ -7.5x+3.875 & при \quad |x|\le 0.25,\\ -2x+2.5625 & при \quad |x|<0.75,\\ -x+1.75 & при \quad |x|\le 1,\\ -0.5x+1.25 & при \quad |x|\le 2,\\ -0.125x+0.5 & при \quad |x|\le 2.8125,\\ -0.0625x+0.3125 & при \quad |x|\le 3.75,\\ -0.0625 & при \quad |x|\le 6.0625,\\ 0 & для\ всех\ остальных\ x.\end{cases} \quad (8)$$

На рисунке 2 представлена схема аппаратной реализации PWL-функции. Согласно этой схеме сначала вычисляются обратные величины входных значений, которые затем поступают в умножитель. Коэффициенты в (8) подобраны таким образом, что умножение заменяется на операции смещения или вычисляется из суммы смещенных данных. Умножитель в параллельном режиме вычисляет все произведения, после чего компаратор выбирает нужные данные и коммутирует их с нужными значениями констант свободного члена согласно (8), после чего полученные значения складываются в специальном сумматоре.

В таблице 2 сведены результаты синтеза обоих видов аппроксимации ψ-функции. Синтез проводился на выбранной ПЛИС фирмы ALTERA семейства Cyclone II, выполненной по технологии 90 нм. Из таблицы 2 видно, что PWL-аппроксимация занимает большую площадь на кристалле и работает несколько медленнее, нежели LUT-аппроксимация, но такой подход снижает зависимость декодера от формата представления метрик и является компромиссом между точностью метрик, сложностью декодера и скоростью его работы. Оба варианта годятся для реализации оптимального декодера LDPC-кода. Выбор того или иного варианта зависит от конкретной задачи.

Таблица 2

Сравнение аппаратных реализаций LUT- и PWL-архитектур

Архитектура	F_{max}, [МГц]	Площадь, [мкм^2]	Логические ячейки
PWL	1055	2295.97	217
LUT	1450	1299.07	135

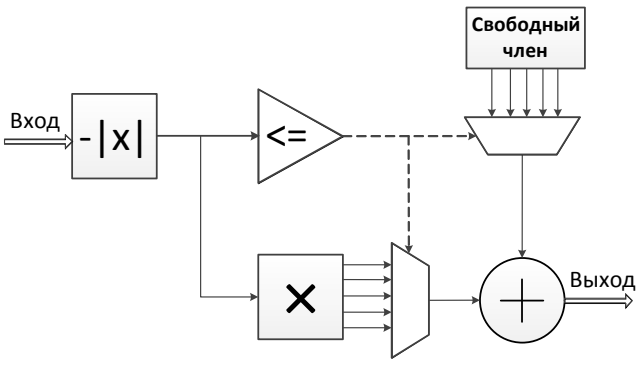

Рис. 2. Схема аппаратной реализации PWL-функции

Декодирование LDPC-кодов методом разбиения на кластеры

Дальнейшего снижения вычислительной нагрузки на декодер можно добиться за счет использования кластерного подхода к декодированию. Для лексикографического разбиения пространства комбинаций LDPC-кода проверочную матрицу $H_{(n,k,d)}$ необходимо привести к систематическому виду:

$$H'_{(n,k,d)} = \left[I_{(n-k)\times(n-k)} \ \vdots \ P_{(n-k)\times k} \right]. \quad (9)$$

Отсюда получают порождающую матрицу кода $G_{(n,k,d)}$ в систематической форме:

$$G_{(n,k,d)} = \left[P^{T}{}_{k\times(n-k)} \ \vdots \ I_{k\times k} \right], \quad (10)$$

после чего множество кодовых комбинаций $\{c_i\}$ считается заданным. Для LDPC-кода с параметрами (6,3,3) матрица H имеет вид

$$H_{(6,3,3)} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \quad (11)$$

Обозначая номера строк матрицы (11) через b_i, где $i = \overline{1,4}$, путем линейных преобразований строк этой матрицы вида $b'_1 = b_3;\ b'_2 = b_2;\ b'_3 = b_1 \oplus b_2 \oplus b_3$ и $b'_4 = b_1 \oplus b_2 \oplus b_3 \oplus b_4$ избавляются от вертикальных проверок четности LDPC-кода и получают матрицу $H'_{(6,3,3)}$, которую далее представляют в формате (9).

$$H'_{(6,3,3)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \rightarrow$$

$$\rightarrow H''_{(6,3,3)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \quad (12)$$

В соответствии с (10) порождающая матрица LDPC-кода принимает вид:

$$G_{(6,3,3)} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}. \quad (13)$$

Полное множество разрешенных кодовых комбинаций LDPC-кода с учетом лексикографического разбиения на кластеры принимает вид:

$$\alpha^i \in GF\left(2^3\right) \times G_{(6,3,3)} = \left\{ \begin{matrix} \langle \varphi \rangle & \langle k-\varphi \rangle & \langle n-k \rangle \\ 00 & 0 & 000 \\ 01 & 1 & 100 \\ 11 & 0 & 010 \\ 11 & 1 & 001 \\ 00 & 1 & 011 \\ 10 & 1 & 110 \\ 10 & 0 & 101 \\ 01 & 0 & 111 \end{matrix} \right\} \in C_{6,3}, \quad (14)$$

где $i = 0, ..., 2(2^2-1)$ – степень примитивного элемента поля $GF(2^3)$, совокупность которых представляет набор информационных символов источника информации, при этом подразумевается наличие единичного элемента поля по операции сложения. В качестве элементов номеров кластеров $\langle \varphi \rangle$ в выражении (14) выбраны первые два разряда каждой кодовой комбинации. Очевидно, для этого могли быть использованы любые пары элементов. Номера разрядов других элементов для блоков $\langle k-\varphi \rangle$ и $\langle n-k \rangle$ очевидны [6]. Упорядочение комбинаций по возрастанию значений $\langle \varphi \rangle$ из набора (14) приводит структуру LDPC-кода $C_{6,3}$ к виду:

$$\alpha^i \in GF\left(2^3\right) \times G_{(6,3,3)} = \begin{matrix} \langle \varphi \rangle & \langle k-\varphi \rangle & \langle n-k \rangle & i_{10} \\ 00 & 0 & 000 & i=0 \\ 00 & 1 & 011 & i=0 \\ 01 & 1 & 100 & i=1 \\ 01 & 0 & 111 & i=1. \\ 10 & 0 & 101 & i=2 \\ 10 & 1 & 110 & i=2 \\ 11 & 0 & 010 & i=3 \\ 11 & 1 & 001 & i=3 \end{matrix} \quad (15)$$

Все кластеры с $i \neq 0$ подобны набору $\{c_{i=0}\}$, при этом в роли транслятора (специфического коэффициента подобия) выступает ключевая комбинация i-го кластера, имеющая структуру вида $V_{кл} = \langle \varphi_{i\neq 0} \rangle \langle k-\varphi = 0 \rangle ...$. Сложение любой комбинации кластера с $V_{кл}$ переводит эту комбинацию в кластер $\{c_{i=0}\}$. Если номер кластера вычислен верно, то по известной структуре $V_{кл}$ приемник способен обрабатывать все кодовые векторы методом списочного декодирования с использованием единственного списка в формате кластера $\{c_{i=0}\}$. Значения $V_{кл}$ для множества кластеров кода $C_{n,k}$ могут быть достаточно просто вычислены или содержаться в списке подобных комбинаций. Например, для $i=1$ можно получить: $010111 \oplus 011100 = 001011$, что соответствует комбинации из $\{c_{i=0}\}$. Легко проверить справедливость этого свойства для других кластеров LDPC-кода $C_{6,3}$.

Пример.

Передатчик отправляет вектор $V_{пер} = 101110$. В канале связи действует помеха вида $V_{ош} = 000111$. Приемник принимает комбинацию кода $V_{пр} = V_{пер} \oplus V_{ош} = 101001$, которая не принадлежит списку (15). Для $V_{пр}$ декодер определяет номер кластера $\langle \varphi_2 \rangle \rightarrow \langle 10_2 \rangle$ и на этой основе вычисляет (извлекает из списка) значение ключевой комбинации $V_{кл} = 100101$, после чего $V_{пр}$ переводится в список базового кластера $V_{пр} \oplus V_{кл} = V'_{бк} = 001100$. Но истинный вектор базового кластера по признаку $\langle k-\varphi \rangle$ равен $V_0 = 001011$, следовательно, $V_{ош} = V'_{бк} \oplus V_0 = 000111$. Выполняя $V_{пр} \oplus V_{ош}$, получаем $V_{пер}$. Представленная схема декодирования не носит общего характера, поскольку не отвечает требованию исправления любого набора

ошибок кратности $t \le n-k$ на длине кодовой комбинации n.

В случае поражения ошибками части разрядов $\langle \varphi \rangle$ и $\langle k-\varphi \rangle$ целесообразно использовать алгоритм, основанный на применении метода упорядоченной статистики мягких решений символа (МРС), когда оценки МРС ранжируются по убыванию [7, 8]. В подобном декодере жесткие решения символов кодовых комбинаций 1 или 0, например, регистрируются в виде знака «+» или «–» соответственно. Кодовый вектор длины n в процессоре приемника фиксируется в виде некоторой последовательности $+\lambda_1, +\lambda_2, -\lambda_3, \ldots, -\lambda_{n-1}, +\lambda_n$. Для реализации процедуры упорядочения символов по значениям их МРС знаки жестких решений значения не имеют. Сортировка завершается, если, например, установлено, что

$$|\lambda_3| \ge |\lambda_n| \ge |\lambda_5| \ge \ldots \ge |\lambda_1| \ge |\lambda_2|, \qquad (16)$$

и на этой основе сформирована перестановочная матрица R [9].

Последовательность (16) является упорядоченной, в ней старшие разряды являются наиболее надежными. Используя эти разряды в качестве информационных, декодер формирует комбинацию эквивалентного кода, которую в последующем сравнивает с принятым и упорядоченным вектором (16). Результат сравнения с высокой вероятностью представляет вектор ошибок, действовавший в канале связи при передаче кодовой комбинации.

Сравнительные характеристики по общему числу выполняемых арифметических операций для названных методов приведены на рисунке 3.

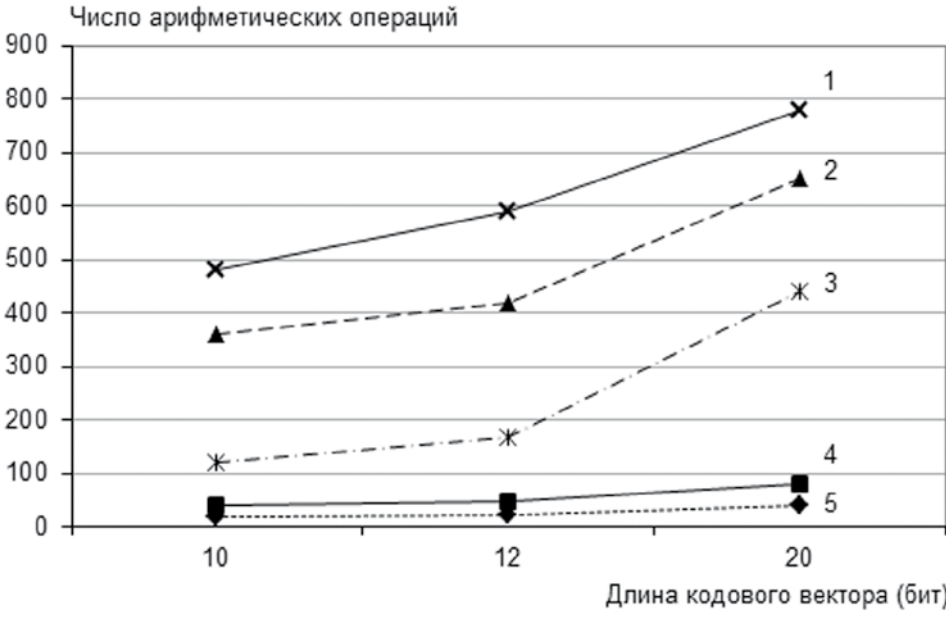

Рис. 3. Сравнительные данные сложности реализации декодера LDPC-кода:
1 – декодирование методом распространения доверия; 2 – декодирование методом быстрого преобразования Фурье; 3 – перестановочное декодирование; 4 – лексикографическое декодирование (верхняя оценка); 5 – лексикографическое декодирование (нижняя оценка)

Заключение

С появлением новых стандартов связи, внедрением LDPC-кодов в существующие стандарты растет спрос на данный вид кодирования, а значит, возрастают и требования к скорости работы кодека, занимаемой им на кристалле площади, потреблению электроэнергии. Все это приводит к постоянному совершенствованию техники декодирования низкоплотностного кода, поиску оптимального алгоритма работы.

В рамках этой работы были подробно рассмотрены сложности, с которыми может столкнуться разработчик при реализации LDPC-кодека. Предложено несколько путей их преодоления. Также проведен анализ эффективности каждого метода по нескольким критериям: скорости работы, занимаемой на кристалле площади, количеству затрачиваемых логических элементов. Каждый метод имеет свои преимущества и недостатки, а выбор того или иного алгоритма зависит от конкретной задачи. Стоит также отметить, что описанные в работе проблемы аппаратной реализации LDPC-декодера характерны для любых линейных кодов, использующих итеративный алгоритм распространения доверия для получения оценок надежности принятых символов. Поэтому предложенные методы по сокращению вычислительных мощностей, затрачиваемых на реализацию устройства, универсальны и могут быть распространены на другие виды кодирования.

Эта работа создает задел для дальнейших работ по оптимизации аппаратно реализуемого LDPC-кодека. Следующим этапом развития данной идеи станет разработка оптимального алгоритма для динамически перестраиваемого LDPC-декодера.

СПИСОК ЛИТЕРАТУРЫ

1. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. – М. : Техносфера, 2005.– 320 с.

2. Хлынов А.А. Исследование принципов реализации LDPC-кодека на ПЛИС//Фундаментальные проблемы радиоэлектронного приборостроения: материалы XI Международной научно-технической конференции «INTERMATIC–2012». – Москва, 2012. – Часть 6. – С. 150–156.

3. Hayes, D. FPGA implementation of a flexible LDPC decoder / The University of Newcastle. – 2008. – October.

4. Masera, G. Implementation of a flexible LDPC decoder / G. Masera, F. Quaglio, F. Vacca // Circuits and

Systems II: Express Briefs, IEEE Transactions. – 2007. – June, vol. 54, no. 6, pp. 542–546.

5. Quaglio, F. Low complexity, flexible LDPC decoders / F. Quaglio, F. Vacca, G. Masera // 14th IST Mobile & Wireless Communications Summit. – 2005.

6. Гладких А.А. Обобщенный метод декодирования по списку на базе кластеризации пространства кодовых векторов // Радиотехника. – 2015. – № 6.– С.37–41.

7. Alon, N. Linear Time Erasure Codes with Nearly Optimal Recovery / N. Alon, J. Edmonds, and M. Luby // IEEE Symposium on Foundations of Computer Science, Proc. IEEE Vol. 3, 1995, pp. 512–519.

8. Гладких А.А. Основы теории мягкого декодирования избыточных кодов в стирающем канале связи. – Ульяновск : УлГТУ, 2010. – 379 с.

9. Мартюшев, Ю.Ю. Практика функционального цифрового моделирования в радиотехнике. – М. : Горячая линия – Телеком, 2012. – 188 с. : ил.

REFERENCES

1. Morelos-Saragosa R. *Iskusstvo pomekhoustoichivogo kodirovaniia. Metody, algoritmy, primenenie* [The Art of Error Correcting Coding. Methods, Algorithms, Application]. Moscow, Tekhnosfera Publ., 2005. 320 p.

2. Khlynov A.A. Issledovanie printsipov realizatsii LDPC kodeka na PLIS [Research on Realization Principles of LDPC Decoder FPGA-based]. *Materialy Mezhdunarodnoi nauchno-tekhnicheskoi konferentsii «INTERMATIC – 2012»* [Fundamental Problems of Radioengineering and Device Construction. Proceedings of the International Scientific Conference "INTERMATIC – 2012"], Moscow, 2012, part 6, pp. 150–156.

3. Hayes D. FPGA *Implementation of a Flexible LDPC Decoder*. The University of Newcastle, 2008, October.

4. Masera G., F. Quaglio, and F. Vacca. Implementation of a Flexible LDPC Decoder. *Circuits and Systems II: Express Briefs, IEEE Transactions,* 2007, June, vol. 54, no. 6, pp. 542–546.

5. Quaglio, F. F. Vacca, and G. Masera. Low complexity, flexible LDPC decoders. *The 14th IST Mobile & Wireless Communications Summit,* 2005.

6. Gladkikh A.A. Obobshchennyi metod dekodirovaniia po spisku na baze klasterizatsii prostranstva kodovykh vektorov [Generalized Method of List Decoding on the Basis of Code Vectors Space Clustering]. *Radiotekhnika* [Radioengineering], 2015, no. 6, pp. 37–41.

7. Alon N., Edmonds J., and Luby M. Linear Time Erasure Codes with Nearly Optimal Recovery. *IEEE Symposium on Foundations of Computer Science, Proc. IEEE,* 1995, vol. 3, pp. 512–519.

8. Gladkikh A.A. *Osnovy teorii miagkogo dekodirovaniia izbytochnykh kodov v stiraiushchem kanale sviazi* [Foundations of the Theory of Soft-Decision Decoding of Redundant Codes in Erasure Communication Channels]. Ulyanovsk, UlSTU Publ., 2010. 379 p.

9. Martiushev Iu.Iu. *Praktika funktsionalnogo tsifrovogo modelirovaniia v radiotekhnike* [Functional Digital Modeling Practice in Radioengineering]. Moscow, Goriachaia liniia – Telekom Publ., 2012. 188 p.