УДК 681.325.3

Д. М. Романенко, кандидат технических наук, доцент (БГТУ);
Д. В. Шиман, кандидат технических наук, старший преподаватель (БГТУ);
М. Ф. Виткова, студент (БГТУ)

МНОГОПОРОГОВОЕ МАЖОРИТАРНОЕ ДЕКОДИРОВАНИЕ НИЗКОПЛОТНОСТНЫХ КОДОВ

В данной статье рассмотрены вопросы, касающиеся мажоритарного декодирования LDPC-кода – трехмерного линейного итеративного кода с двойными объединенными диагональными проверками. Проанализированы параметры рассматриваемого LDPC-кода, а также исправление различных типов одиночных, двойных и тройных ошибок. Обоснована целесообразность использования многопорогового мажоритарного декодирования. Рассмотренный алгоритм обеспечивает практически оптимальное декодирование длинных кодов при линейной сложности реализации с ростом длины кода даже при весьма высоком уровне шума канала.

The article is considered the results of the majority decoding LDPC-code: line three-dimensional iterative codes with double incorporated diagonal checks. Parameters of the LDPC-code and the correction of various types of single, double and triple errors are analyzed. The expediency of using multithreshold majority decoding is justified. The considered algorithm provides almost optimal decoding of long codes with linear complexity of the code length increases, even at a very high level of channel noise.

Введение. Наблюдаемое в последние годы резкое увеличение информационных потоков и ужесточение требований к целостности передаваемой и обрабатываемой информации обострили проблему надежного хранения и передачи двоичных данных. Увеличение плотности интеграции элементов в системах хранения и передачи информации приводит к возрастанию вероятности появления ошибок высокой кратности. Одним из способов решения описанных проблем является использование помехоустойчивого кодирования данных с помощью избыточных кодов. Известно достаточно много кодов с высокими корректирующими способностями (например, БЧХ-коды, Рида – Соломона, низкоплотностные и т. д.). Но при этом ключевую роль в процессе исправления ошибок играет именно декодер, и его выбор является важнейшей задачей, решаемой при разработке систем хранения и передачи информации.

Таким образом, целью данной работы является изучение многопорогового мажоритарного декодирования низкоплотностных кодов.

Основная часть. Двумерные итеративные коды, широко применяемые на практике и более известные как HV-коды, являются простейшим примером использования методов комбинирования известных кодов для построения новых и представляют собой прямое произведение кодов простой проверки на четность. Поступательное развитие в данном направление избыточного кодирования привело к появлению двумерных линейных итеративных кодов с объединенными диагональными проверками [1, 2], а также их трехмерных вариантов (рис. 1). Трехмерные линейные итеративные коды [3, 4] благодаря низкой плотности единиц в порождающей матрице (в строках не более чем $\sqrt[3]{k}$ единиц) могут быть отнесены к LDPC-кодам (Low Density Paritet Check Codes).

Принцип формирования проверочных символов для такого кода при $k = 64$ бит представлен на рис. 2 (*1* – информационные биты; *2* – горизонтальные паритеты; *3* – вертикальные паритеты; *4*, *5* – соответственно первые и вторые объединенные диагональные паритеты; *6* – *z*-паритеты; *7* – контрольная сумма) [2–4].

Отметим, что предполагается использовать для исправления ошибок различные мажоритарные декодеры, следовательно, необходимо исключить контрольные суммы в четырех плоскостях, так как они являются линейно зависимыми проверками и могут быть получены суммой всех горизонтальных или вертикальных паритетов в соответствующей плоскости.

Трехмерный линейный итеративный код с объединенными диагональными проверками, являющийся низкоплотностным кодом и представленный на рис. 2, характеризуется минимальным кодовым расстоянием, равным 6 (без учета контрольных сумм в плоскостях), так как любые два кодовых слова будут отличаться минимум в шести позициях. Следовательно, при использовании такого кода могут быть исправлены все одиночные и двойные ошибки, а также обнаружены все тройные ошибки.

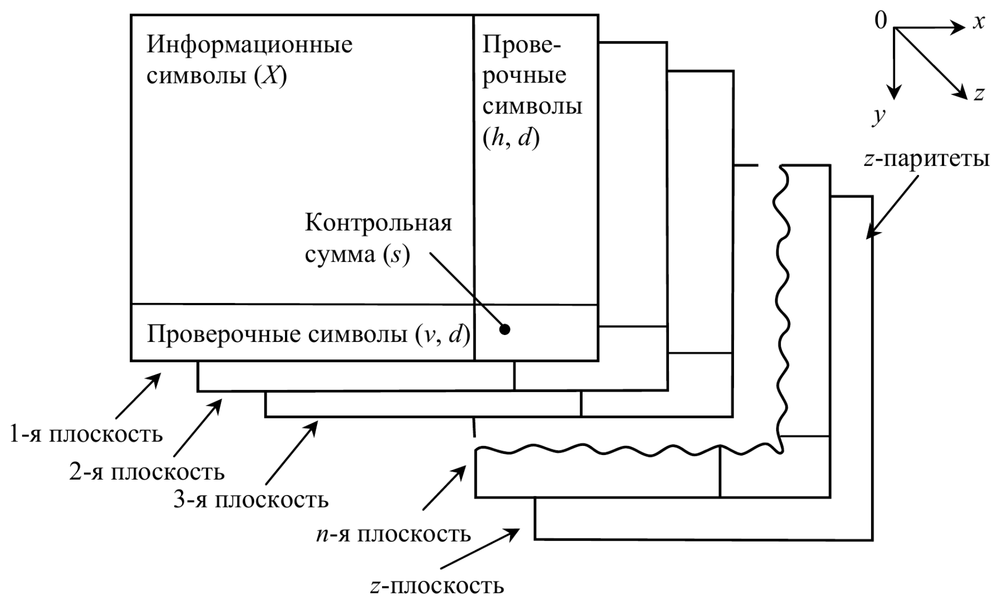

Рис. 1. Общий вид трехмерных итеративных кодов

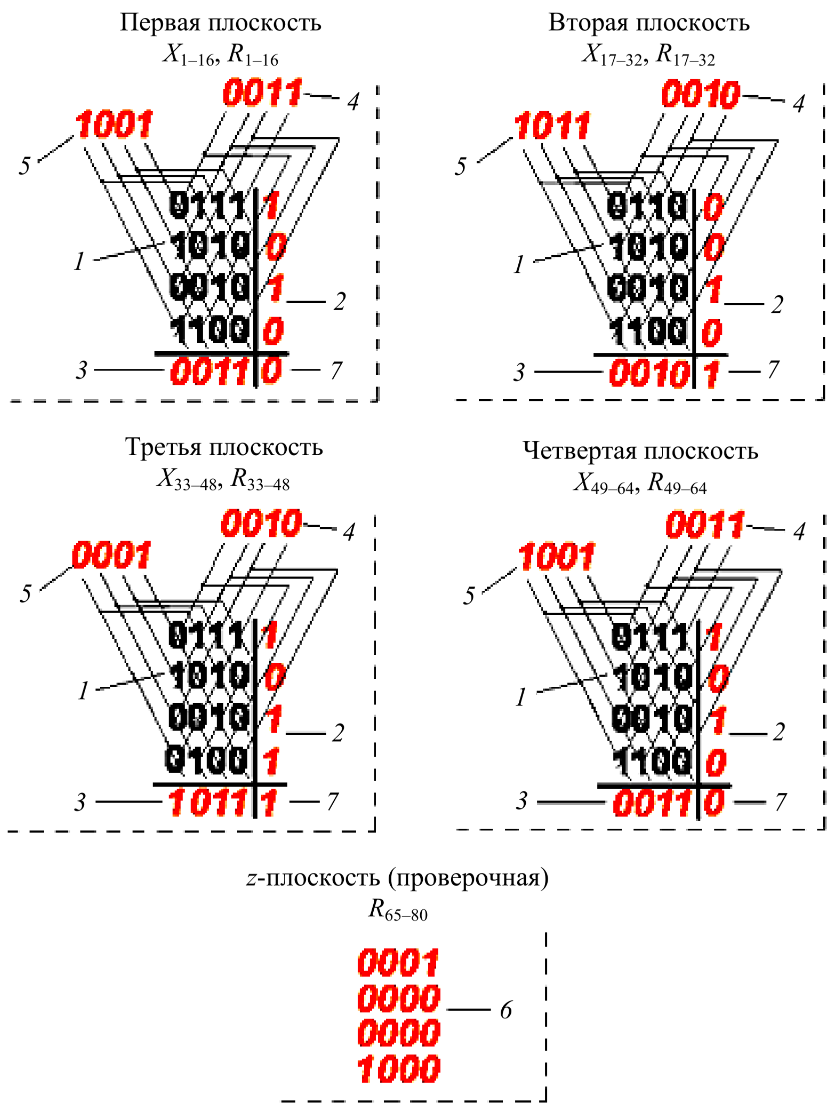

Рис. 2. Принцип формирования проверочных символов трехмерным линейным итеративным кодом с объединенными диагональными проверками

Проверочные символы R_{1-80} при $k = 64$ бит в соответствии с рис. 2 могут быть рассчитаны по следующим зависимостям:

$$\begin{gathered}R_1 = X_1 \oplus X_2 \oplus X_3 \oplus X_4, \\ R_2 = X_5 \oplus X_6 \oplus X_7 \oplus X_8, \\ \vdots \\ R_5 = X_1 \oplus X_5 \oplus X_9 \oplus X_{13}, \\ \vdots \\ R_{14} = X_3 \oplus X_8 \oplus X_9 \oplus X_{14}, \\ \vdots \\ R_{16} = X_1 \oplus X_6 \oplus X_{11} \oplus X_{16}, \\ R_{17} = X_{17} \oplus X_{18} \oplus X_{19} \oplus X_{20}, \\ \vdots \\ R_{37} = X_{33} \oplus X_{37} \oplus X_{41} \oplus X_{45}, \\ \vdots \\ R_{60} = X_{52} \oplus X_{55} \oplus X_{58} \oplus X_{61}, \\ \vdots \\ R_{65} = X_1 \oplus X_{17} \oplus X_{33} \oplus X_{49}, \\ \vdots \\ R_{80} = X_{16} \oplus X_{32} \oplus X_{48} \oplus X_{64}.\end{gathered} \tag{1}$$

Однако на практике, например в спутниковых каналах связи, низкоплотностные коды используются для информационных последовательностей длиной сотни или тысячи бит. При этом, согласно теории кодирования, основными параметрами кодов являются избыточность (r) и скорость ($R = k / (k + r)$) [1]. Данные параметры для трехмерного линейного итеративного кода с объединенными диагональными проверками при различных длинах k (от 64 до 4096 бит) представлены в табл. 1.

Таблица 1

Параметры низкоплотностных трехмерных итеративных кодов

Длина информационной последовательности k	Избыточность r	Скорость $k / (k + r)$
64	80	0,44
128	144	0,47
256	192	0,57
512	320	0,62
1024	576	0,64
2048	768	0,72
4096	1280	0,76

Отметим, что с увеличением длины информационной последовательности наблюдается рост скорости кода.

Ключевую роль в использовании избыточных кодов играет процесс декодирования. Одним из самых быстродействующих является мажоритарное декодирование – декодирование по принципу «большинства». При этом в последнее время активно развивается идея многопорогового мажоритарного декодирования. Проанализируем целесообразность использования таких декодеров для приведенного выше кода с точки зрения возможности исправления ошибок различной кратности.

В трехмерных итеративных кодах с двойными диагональными проверками правильность каждого бита информационного слова *1* контролируется пятью проверочными символами (рис. 2) (контрольная сумма в плоскости не учитывается по описанным выше причинам): горизонтальным паритетом *2*, вертикальным *3*, первым объединенным диагональным *4*, вторым объединенным диагональным *5* и *z*-паритетом *6*.

Алгоритм мажоритарного декодирования, представленный в [5] на примере двумерного линейного итеративного кода с объединенными диагональными проверками, аналогично может применяться и для рассматриваемого в данной статье низкоплотностного кода. Поэтому далее более подробно рассмотрим только метод многопорогового декодирования.

Пусть закодированы 64 бит ($k = 64$) информации $X = 0111\ 1010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100$, т. е. согласно порождающей матрице (рис. 2), сформированы следующие проверочные символы:

$$\begin{gathered}R_{1-4} = 1010, \\ R_{5-8} = 0011, \\ R_{9-12} = 0011, \\ R_{13-16} = 1001, \\ \vdots \\ R_{65-80} = 0001000000001000.\end{gathered} \tag{2}$$

Пусть при хранении (или передаче) в 1, 3 и 5-м бите информационной последовательности X возникли ошибки (инверсия бита). Тогда $X' = 1101\ 0010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100$, а новые проверочные символы станут следующими:

$$\begin{gathered}R'_{1-4} = 1110, \\ R'_{5-8} = 0001, \\ R'_{9-12} = 1101, \\ R'_{13-16} = 1110, \\ \vdots \\ R'_{65-80} = 1011100000001000.\end{gathered} \tag{3}$$

При многопороговом мажоритарном декодировании каждый информационный бит декодируется (проверяется) отдельно на каждой стадии. При пяти линейно независимых паритетах целесообразно использовать трехстадийное декодирование со следующими пороговыми значениями: T1 = 5, T2 ≥ 4, T3 ≥ 3.

Далее рассмотрим непосредственно алгоритм декодирования.

Первая стадия. На данной стадии ни один бит не был инвертирован, поскольку пороговое значение, равное пяти, не было достигнуто. Так, например, при декодировании 11-го бита, который мог бы быть принят за ошибочный при стандартном мажоритарном декодировании, получим следующее (результат суммирования, равный «1», свидетельствует о том, что ошибка произошла в данном бите, «0» – ошибки нет):

$$\begin{aligned} R_3 \oplus R_3' &= 1 \oplus 1 = 0, \\ R_7 \oplus R_7' &= 1 \oplus 0 = 1, \\ R_9 \oplus R_9' &= 0 \oplus 1 = 1, \\ R_{16} \oplus R_{16}' &= 1 \oplus 0 = 1, \\ R_{75} \oplus R_{75}' &= 0 \oplus 0 = 0. \end{aligned} \tag{4}$$

Вторая стадия. При проверке 1-го информационного бита выполнялось сравнение R_1 и R'_1, R_5 и R'_5, R_9 и R'_9, R_{16} и R'_{16}, R_{65} и R'_{65}. Иными словами, были выполнены следующие операции суммирования по модулю 2:

$$\begin{aligned} R_1 \oplus R_1' &= 1 \oplus 1 = 0, \\ R_5 \oplus R_5' &= 0 \oplus 0 = 0, \\ R_9 \oplus R_9' &= 0 \oplus 1 = 1, \\ R_{16} \oplus R_{16}' &= 1 \oplus 0 = 1, \\ R_{65} \oplus R_{65}' &= 0 \oplus 1 = 1. \end{aligned} \tag{5}$$

По результатам суммирования, согласно выбранному пороговому значению (четыре и более), был сделан вывод, что ошибки в данном информационном бите нет. При проверке 3-го и 5-го бита принималось решение о необходимости инвертирования, так как пороговое значение было достигнуто:

– для 3-го бита

$$\begin{aligned} R_1 \oplus R_1' &= 1 \oplus 1 = 0, \\ R_7 \oplus R_7' &= 1 \oplus 0 = 1, \\ R_{11} \oplus R_{11}' &= 1 \oplus 0 = 1, \\ R_{14} \oplus R_{14}' &= 0 \oplus 1 = 1, \\ R_{67} \oplus R_{67}' &= 0 \oplus 1 = 1, \end{aligned} \tag{6}$$

– для 5-го бита

$$\begin{aligned} R_2 \oplus R_2' &= 0 \oplus 1 = 1, \\ R_5 \oplus R_5' &= 0 \oplus 0 = 0, \\ R_{10} \oplus R_{10}' &= 0 \oplus 1 = 1, \\ R_{15} \oplus R_{15}' &= 0 \oplus 1 = 1, \\ R_{69} \oplus R_{69}' &= 0 \oplus 1 = 1. \end{aligned}$$

Поскольку на данной стадии были исправлены ошибки, то после ее окончания были пересчитаны проверочные символы, в формировании которых участвовали инвертированные биты. Следовательно, были получены следующие проверочные символы:

$$\begin{aligned} R'_{1-4} &= 0010, \\ R'_{5-8} &= 1011, \\ R'_{9-12} &= 1011, \\ R'_{13-16} &= 1000, \\ &\vdots \\ R'_{65-80} &= 1001000000001000. \end{aligned} \tag{7}$$

Третья стадия. Рассмотрим декодирование 1-го информационного бита аналогично тому, как это было сделано на 2-й стадии:

$$\begin{aligned} R_1 \oplus R_1' &= 1 \oplus 0 = 1, \\ R_5 \oplus R_5' &= 0 \oplus 1 = 1, \\ R_9 \oplus R_9' &= 0 \oplus 1 = 1, \\ R_{16} \oplus R_{16}' &= 1 \oplus 0 = 1, \\ R_{65} \oplus R_{65}' &= 0 \oplus 1 = 1. \end{aligned} \tag{8}$$

Ввиду того, что пороговое значение (три и более) было достигнуто, то было принято решение об инвертировании данного бита. При декодировании всех остальных символов пороговое значение не достигалось.

Таким образом, приведенная тройная ошибка при использовании многопорогового декодера была исправлена, однако необходимо отметить, что данная ошибка при стандартном мажоритарном декодировании также была бы исправлена, но при этом был бы ошибочно инвертирован еще и 6-й информационный бит.

Очевидно, возможны ситуации, когда тройная ошибка не будет исправлена, либо другие биты при декодировании будут приняты за ошибочные и при использовании метода многопорогового декодирования. Например, попытка декодирования информационной последовательности с тройной ошибкой в 1, 2 и 5-м символах приведет к ошибочному декодированию 6-го бита. Иными словами, после всех стадий декодирования в информационной последовательности окажется один ошибочный бит.

Для изучения данных ситуаций и сравнения двух приведенных в данной статье методов декодирования была разработана программная модель, имитирующая процессы кодирования, появления и исправления ошибок.

Результаты моделирования процесса декодирования различных типов трехкратных ошибок для информационной последовательности длиной k = 64 бит представлены в табл. 2.

Таблица 2

Результаты моделирования процесса декодирования

Количество исправленных тройных ошибок, %	Количество ошибок после декодирования, %			
	одиночных	двойных	тройных	четырехкратных
LDPC-код (k = 64) + мажоритарный декодер				
51,8	12,1	22,7	11,5	1,8
LDPC-код (k = 64) + многопороговый мажоритарный декодер (пороговые значения T1 = 5, T2 ≥> 4, T3 ≥ 3)				
98,6	0,6	0,6	0	0,2

Анализ методов мажоритарного декодирования низкоплотностных кодов на примере трехмерного линейного итеративного кода с объединенными диагональными проверками показал, что все одиночные и двойные, а также часть тройных ошибок могут быть исправлены. Так, при длине информационной последовательности k = 64 и использовании стандартного мажоритарного декодера 51,8% ошибок кратностью три будут исправлены. Аналогичный же показатель при использовании метода многопорогового декодирования достигает значения 98,6%.

Заключение. Таким образом, многопороговые мажоритарные декодеры могут эффективно применяться для исправления многократных ошибок низкоплотностным кодом, в виде которого может выступать многомерный линейный итеративный код с двойными диагональными проверками. При этом все одиночные, двойные, а также 98,6% тройных ошибок будут нейтрализованы. Также необходимо отметить, что с увеличением длины информационной последовательности для рассмотренного низкоплотностного кода параметр скорости возрастает и при k = 4096 бит достигает значения 0,76.

Литература

1. Шиман, Д. В. Свойства и параметры линейных итеративных кодов с двойными диагональными проверками / Д. В. Шиман, Д. М. Романенко // Труды БГТУ. Сер. VI, Физ.-мат. науки и информатика. – 2007. – Вып. XV. – С. 151–154.

2. Multilevel turbocoding schemes on the basis of twodimensional linear iterative codes with diagonal checks / P. P. Urbanovich [et al.] // Przeglad elektrotechniczny. – 2008. – № 3. – P. 152–154.

3. Урбанович, П. П. Особенности реализации трехмерного итеративного кода для систем полупроводниковой памяти с интеграцией на пластине / П. П. Урбанович, Д. М. Романенко, А. В. Орлов // Труды БГТУ. Сер. VI, Физ.-мат. науки и информатика. – 2004. – Вып. XII. – С. 173–176.

4. Романенко, Д. М. Использование трехмерных итеративных кодов в каналах передачи данных / Д. М. Романенко // Труды БГТУ. Сер. VI, Физ.-мат. науки и информатика. – 2006. – Вып. XIV. – С. 133–135.

5. Романенко, Д. М. Мажоритарное декодирование двухмерных линейных итеративных кодов с объединенными диагональными проверками / Д. М. Романенко, Д. В. Шиман // Труды БГТУ. Сер. VI, Физ.-мат. науки и информатика. – 2009. – Вып. XVII. – С. 119–121.

Поступила 01.03.2011