

УДК 681.391

М. Ф. Виткова, магистрант (БГТУ);**Д. М. Романенко**, кандидат технических наук, доцент (БГТУ)**АДАПТИВНОЕ МНОГОПороГОВОЕ ДЕКОДИРОВАНИЕ
МНОГОМЕРНЫХ ИТЕРАТИВНЫХ КОДОВ**

В статье рассмотрены особенности исправления ошибок различной кратности трехмерными итеративными кодами методом многопорогового мажоритарного декодирования. Проанализировано исправление независимых ошибок как низкой, так и высокой кратности, а также возможность оптимизации процесса декодирования за счет сокращения числа стадий декодирования. Особенности использования многопорогового декодера с трехмерными итеративными кодами позволяют уже на первой стадии определить, какие стадии могут быть пропущены, что приводит к сокращению общего времени декодирования.

The article are considered a features of different multiplicity error correction by tree-dimensional iterative codes with multithreshold majority decoding method. A correction of independent errors both low and high frequency rate, and also possibility of optimization decoding process by reducing the number of stages are analyzed. Some features of using multithreshold decoder with tree-dimensional iterative codes allow to define at the first stage what stages can to be passed, which leads to reducing total decoding time.

Введение. В настоящее время весьма актуальна проблема надежного хранения и передачи информации, связанная с необходимостью коррекции появляющихся ошибок. При этом ключевую роль в процессе исправления ошибок играет именно декодер. Его выбор, а также адаптация к особенностям избыточного кода является важнейшей задачей, решаемой при разработке систем хранения и передачи информации. На сегодняшний день широко используются различные декодеры, но при этом активно развивается идея многопорогового мажоритарного декодирования.

Целью данной работы является изучение свойств многопорогового декодера (МПД) и анализ возможности оптимизации процесса исправления ошибок.

Основная часть. Многопороговые мажоритарные декодеры считаются одними из самых быстродействующих. В методе многопорогового декодирования каждый информационный бит проверяется отдельно на каждой стадии [1, 2].

Под стадией многопорогового декодирования следует понимать декодирование кодового слова с заданным пороговым значением. Стадии декодирования выполняются последовательно друг за другом. Таким образом, можно сказать, что идентификация и исправление ошибок в информационной последовательности выполняется несколько раз при различных пороговых значениях.

При декодировании итеративных кодов число стадий и пороговое значение зависит от числа паритетов.

Пороговое значение – это недопустимое число проверочных символов, указывающих на наличие ошибки в проверяемом бите. При достижении порогового значения проверяемый бит принимается за ошибочный и инвертируется.

Исправление ошибочных битов зависит от значения порога на каждой стадии декодирования.

Следует отметить, что многопороговый декодер относится к декодерам мажоритарного типа, но для каждой стадии выбирается свое пороговое значение. Максимальное число стадий (S_{\max}) для декодирования избыточного кода определяется исходя из числа линейно-независимых паритетов (p): если p нечетное, то

$$S_{\max} = \text{round}\left(\frac{p}{2}\right); \quad (1)$$

если p четное, то

$$S_{\max} = \frac{p}{2}. \quad (2)$$

Выполнение данных зависимостей необходимо для реализации принципа «большинства», используемого при мажоритарном декодировании. Таким образом, пороговое значение фактически показывает то необходимое число несовпадающих паритетов в процессе декодирования, при достижении которого «принимается решение» об инвертировании соответствующего декодируемого бита.

Максимальное пороговое значение (T_{\max}), используемое на первой стадии, принимается равным числу линейно-независимых паритетов (p). Минимальное же пороговое значение, используемое на последней стадии, будет равно $T_{\min} = T_{\max} / 2$ с учетом правила округления.

Рассмотрим процесс исправления однократной ошибки в информационной последовательности длиной 64 бит итеративным кодом с пятью паритетами: горизонтальным паритетом, вертикальным, первым объединенным диагональным, вторым объединенным диагональным

и z -паритетом. Общий вид данного кода, схема формирования проверочных символов, основные параметры (скорость, избыточность, минимальное кодовое расстояние) и проверочная матрица приведены в [1].

Формирование проверочных символов данным многомерным линейным итеративным кодом с пятью линейно-независимыми паритетами осуществляется по зависимостям:

$$\begin{aligned}
 R_1 &= X_1 \oplus X_2 \oplus X_3 \oplus X_4, \\
 R_2 &= X_5 \oplus X_6 \oplus X_7 \oplus X_8, \\
 &\vdots \\
 R_{17} &= X_1 \oplus X_5 \oplus X_9 \oplus X_{13}, \\
 R_{18} &= X_2 \oplus X_6 \oplus X_{10} \oplus X_{14}, \\
 &\vdots \\
 R_{33} &= X_1 \oplus X_{18} \oplus X_{35} \oplus X_{52}, \\
 R_{34} &= X_2 \oplus X_{19} \oplus X_{36} \oplus X_{49}, \\
 &\vdots \\
 R_{49} &= X_1 \oplus X_{21} \oplus X_{41} \oplus X_{61}, \\
 R_{50} &= X_5 \oplus X_{25} \oplus X_{45} \oplus X_{49}, \\
 &\vdots \\
 R_{65} &= X_1 \oplus X_{17} \oplus X_{33} \oplus X_{49}, \\
 R_{66} &= X_2 \oplus X_{18} \oplus X_{34} \oplus X_{50}, \\
 &\vdots \\
 R_{80} &= X_{16} \oplus X_{32} \oplus X_{48} \oplus X_{64}.
 \end{aligned} \tag{3}$$

При пяти линейно-независимых паритетах в соответствии с (1) и (2) целесообразно использовать трехстадийное декодирование с пороговыми значениями $T_1 = 5$, $T_2 \geq 4$, $T_3 \geq 3$ для первой, второй и третьей стадий соответственно [1].

Пусть необходимо закодировать информационную последовательность $X = 0111\ 1010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100$. Следовательно, согласно [1], сформированы следующие проверочные символы R_{1-80} :

$$\begin{aligned}
 R_{1-4} &= 1010, \\
 &\vdots \\
 R_{17-20} &= 0011, \\
 &\vdots \\
 R_{33-36} &= 1111, \\
 &\vdots \\
 R_{49-52} &= 0000, \\
 &\vdots \\
 R_{65-68} &= 0000, \\
 &\vdots \\
 R_{77-80} &= 0000.
 \end{aligned} \tag{4}$$

Получим кодовое слово следующего вида:

$$Y_{1-144} = X_{1-64} R_{1-80}. \tag{5}$$

Допустим, ошибка произошла в первом информационном бите (инвертированный бит в ошибочной информационной последовательности X' подчеркнут): $X' = \underline{1}111\ 1010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100$. Тогда получим следующие новые проверочные символы R'_{1-80} :

$$\begin{aligned}
 R'_{1-4} &= 0010, \\
 &\vdots \\
 R'_{17-20} &= 1011, \\
 &\vdots \\
 R'_{33-36} &= 0111, \\
 &\vdots \\
 R'_{49-52} &= 1000, \\
 &\vdots \\
 R'_{65-68} &= 1000, \\
 &\vdots \\
 R'_{77-80} &= 0000.
 \end{aligned} \tag{6}$$

Правильность каждого информационного бита контролируется пятью паритетами.

При проверке первого бита на наличие ошибки на первой стадии декодирования с пороговым значением $T_1 = 5$ получается:

$$\begin{aligned}
 R_1 \oplus R'_1 &= 1 \oplus 0 = 1, \\
 R_{17} \oplus R'_{17} &= 0 \oplus 1 = 1, \\
 R_{33} \oplus R'_{33} &= 1 \oplus 0 = 1, \\
 R_{49} \oplus R'_{49} &= 0 \oplus 1 = 1, \\
 R_{65} \oplus R'_{65} &= 0 \oplus 1 = 1.
 \end{aligned} \tag{7}$$

Результат суммирования, равный «1» (несовпадение полученного и рассчитанного проверочного символа), свидетельствует о том, что ошибка произошла в данном бите, «0» – ошибки нет [1]. Значит, на первой стадии декодирования на наличие ошибки в первом бите будут указывать все пять паритетов. Следовательно, данный информационный бит будет инвертирован, так как достигнуто пороговое значение. По завершении первой стадии декодирования вновь рассчитываются проверочные символы R'_{1-80} .

При проверке всех 64 информационных бит на второй стадии не было получено ни одной «1» при декодировании ни одного из них, т. е. выполняется следующее равенство:

$$\sum_{i=1}^r (R_i \oplus R'_i) = 0, \tag{8}$$

где r – величина избыточности (для описываемого многомерного итеративного кода с пятью линейно-независимыми паритетами равна 80) [1]. Выполнение операций суммирования в (8) осуществляется по модулю 2.

Значит, в декодируемой информационной последовательности ошибки отсутствуют (или

все исправлены). Следовательно, нет необходимости выполнять третью стадию декодирования, а значит, ее можно опустить.

Однако возможны и другие ситуации. Допустим, в информационной последовательности произошли ошибки в первых четырех информационных битах: $X' = \underline{1000}1010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100$. Тогда проверочные символы R'_{1-80} будут следующими:

$$\begin{aligned} R'_{1-4} &= 1010, \\ &\vdots \\ R'_{17-20} &= 1100, \\ &\vdots \\ R'_{33-36} &= 0000, \\ &\vdots \\ R'_{49-52} &= 1000, \\ &\vdots \\ R'_{65-68} &= 1111, \\ &\vdots \\ R'_{77-80} &= 0000. \end{aligned} \quad (9)$$

На первой стадии декодирования с пороговым значением $T_1 = 5$ проверка на наличие ошибок в первых четырех битах привела к следующим результатам.

Проверка 1-го бита:

$$\begin{aligned} R_1 \oplus R'_1 &= 1 \oplus 1 = 0, \\ R_{17} \oplus R'_{17} &= 0 \oplus 1 = 1, \\ R_{33} \oplus R'_{33} &= 1 \oplus 0 = 1, \\ R_{49} \oplus R'_{49} &= 0 \oplus 1 = 1, \\ R_{65} \oplus R'_{65} &= 0 \oplus 1 = 1. \end{aligned} \quad (10)$$

Проверка 2-го бита:

$$\begin{aligned} R_1 \oplus R'_1 &= 1 \oplus 1 = 0, \\ R_{18} \oplus R'_{18} &= 0 \oplus 1 = 1, \\ R_{34} \oplus R'_{34} &= 1 \oplus 0 = 1, \\ R_{53} \oplus R'_{53} &= 0 \oplus 1 = 1, \\ R_{66} \oplus R'_{66} &= 0 \oplus 1 = 1. \end{aligned} \quad (11)$$

Проверка 3-го бита:

$$\begin{aligned} R_1 \oplus R'_1 &= 1 \oplus 1 = 0, \\ R_{19} \oplus R'_{19} &= 1 \oplus 0 = 1, \\ R_{35} \oplus R'_{35} &= 1 \oplus 0 = 1, \\ R_{57} \oplus R'_{57} &= 1 \oplus 0 = 1, \\ R_{67} \oplus R'_{67} &= 0 \oplus 1 = 1. \end{aligned} \quad (12)$$

Проверка 4-го бита:

$$\begin{aligned} R_1 \oplus R'_1 &= 1 \oplus 1 = 0, \\ R_{20} \oplus R'_{20} &= 1 \oplus 0 = 1, \\ R_{36} \oplus R'_{36} &= 1 \oplus 0 = 1, \\ R_{61} \oplus R'_{61} &= 1 \oplus 0 = 1, \\ R_{68} \oplus R'_{68} &= 0 \oplus 1 = 1. \end{aligned} \quad (13)$$

Очевидно, что на первой стадии ни одной ошибки в первых четырех битах не будет исправлено, так как пороговое значение, равное 5, не достигнуто. При декодировании остальных 60 бит (тех бит информационной последовательности, в которых ошибок не было) также не будет достигнуто пороговое значение $T_1 = 5$, например:

проверка 6-го бита:

$$\begin{aligned} R_2 \oplus R'_2 &= 0 \oplus 0 = 0, \\ R_{18} \oplus R'_{18} &= 0 \oplus 1 = 1, \\ R_{35} \oplus R'_{35} &= 1 \oplus 0 = 1, \\ R_{49} \oplus R'_{49} &= 0 \oplus 1 = 1, \\ R_{70} \oplus R'_{70} &= 0 \oplus 0 = 1, \end{aligned} \quad (14)$$

проверка 8-го бита:

$$\begin{aligned} R_2 \oplus R'_2 &= 0 \oplus 0 = 0, \\ R_{20} \oplus R'_{20} &= 1 \oplus 0 = 1, \\ R_{33} \oplus R'_{33} &= 1 \oplus 0 = 1, \\ R_{57} \oplus R'_{57} &= 1 \oplus 0 = 1, \\ R_{72} \oplus R'_{72} &= 0 \oplus 0 = 0. \end{aligned} \quad (15)$$

Следовательно, на первой стадии не будет инвертирован ни один бит, поскольку не было достигнуто пороговое значение – максимальное число несовпадающих паритетов при проверке четырех первых бит было 4. При этом существуют проверочные символы, указывающие на наличие ошибок. Значит, необходимо осуществить проверку на второй стадии декодирования с пороговым значением $T_2 \geq 4$. Отметим, что пересчет проверочных символов для второй стадии декодирования не требуется, так как на первой стадии ни один из информационных бит не был инвертирован.

Исходя из результатов расчетов по формулам (10)–(13) на второй стадии будут исправлены 1-й и 2-й биты, поскольку на наличие ошибки в данных битах указывают четыре паритета, что соответствует пороговому значению. Следовательно, по завершении второй стадии получим $X' = \underline{0100}1010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100\ 0111\ 1010\ 0010\ 1100$ (ошибки остались в 3-м и 4-м битах), а также в результате пересчета новых проверочных символов имеем:

$$\begin{aligned}
R'_{1-4} &= 1010, \\
&\vdots \\
R'_{17-20} &= 0000, \\
&\vdots \\
R'_{33-36} &= 1100, \\
&\vdots \\
R'_{49-52} &= 0001, \\
&\vdots \\
R'_{65-68} &= 0011, \\
&\vdots \\
R'_{77-80} &= 0000.
\end{aligned} \tag{16}$$

При проверке первых четырех бит на третьей стадии декодирования с пороговым значением $T_3 \geq 3$ получены следующие результаты.

Проверка 1-го бита:

$$\begin{aligned}
R_1 \oplus R'_1 &= 1 \oplus 1 = 0, \\
R_{17} \oplus R'_{17} &= 0 \oplus 0 = 0, \\
R_{33} \oplus R'_{33} &= 1 \oplus 1 = 0, \\
R_{49} \oplus R'_{49} &= 0 \oplus 0 = 0, \\
R_{65} \oplus R'_{65} &= 0 \oplus 0 = 0.
\end{aligned} \tag{17}$$

Проверка 2-го бита:

$$\begin{aligned}
R_1 \oplus R'_1 &= 1 \oplus 1 = 0, \\
R_{18} \oplus R'_{18} &= 0 \oplus 0 = 0, \\
R_{34} \oplus R'_{34} &= 1 \oplus 1 = 0, \\
R_{53} \oplus R'_{53} &= 0 \oplus 0 = 0, \\
R_{66} \oplus R'_{66} &= 0 \oplus 0 = 0.
\end{aligned} \tag{18}$$

Проверка 3-го бита:

$$\begin{aligned}
R_1 \oplus R'_1 &= 1 \oplus 1 = 0, \\
R_{19} \oplus R'_{19} &= 0 \oplus 0 = 0, \\
R_{35} \oplus R'_{35} &= 1 \oplus 0 = 1, \\
R_{57} \oplus R'_{57} &= 1 \oplus 0 = 1, \\
R_{67} \oplus R'_{67} &= 0 \oplus 1 = 1.
\end{aligned} \tag{19}$$

Проверка 4-го бита:

$$\begin{aligned}
R_1 \oplus R'_1 &= 1 \oplus 1 = 0, \\
R_{20} \oplus R'_{20} &= 0 \oplus 0 = 0, \\
R_{36} \oplus R'_{36} &= 1 \oplus 0 = 1, \\
R_{61} \oplus R'_{61} &= 1 \oplus 0 = 1, \\
R_{68} \oplus R'_{68} &= 0 \oplus 1 = 1.
\end{aligned} \tag{20}$$

Следовательно, на третьей стадии будут исправлены 3-й и 4-й биты, так как было достигнуто пороговое значение, равное 3. В итоге получим декодированную информационную последовательность $X' = 0111 1010 0010 1100 0111 1010 0010 1100 0111 1010 0010 1100$. Таким образом, по

результатам трех стадий все ошибки были исправлены.

В последнем примере были задействованы все три стадии декодирования и ни одну из стадий пропускать было нельзя.

Приведенные ситуации показывают, что в зависимости от кратности ошибки могут быть пропущены одна и более стадий. Однако также возможны ситуации, когда стадии не будут пропущены вообще. Число пропущенных стадий также зависит от числа линейно-независимых паритетов. Пропуская стадии, на которых ошибки не будут исправлены, можно существенно сократить время декодирования.

Для изучения данной особенности исправления ошибок многомерными итеративными кодами было использовано программное средство, описанное в [3], в которое добавлен модифицированный алгоритм многопорогового декодирования, адаптированный под возможность пропуска отдельных стадий.

В ходе исследования использовались трехмерные линейные итеративные коды с числом линейно-независимых паритетов 5, 7 и 9, поскольку данные коды формируют достаточное число линейно-независимых паритетов для анализа возможности оптимизации декодера. Основные параметры используемых кодов представлены в табл. 1. К ним относятся избыточность (r) и скорость кода ($R = k / (k + r)$).

Таблица 1

Параметры итеративных кодов

Длина информационной последовательности k	Избыточность r	Скорость $k / (k + r)$
Пять линейно-независимых паритетов		
64	80	0,44
128	128	0,50
256	192	0,57
512	320	0,62
Семь линейно-независимых паритетов		
64	112	0,36
128	160	0,44
256	256	0,50
512	448	0,53
Девять линейно-независимых паритетов		
64	144	0,31
128	208	0,38
256	320	0,44
512	576	0,47

Для анализа эффективности сокращения стадий декодирования использовалась информационная последовательность длиной 64 бит.

Разработанное приложение позволило получить данные (табл. 2), на основании которых построена диаграмма, изображенная на рисунке.

Таблица 2
Среднее количество пропущенных стадий, %

Кратность независимых ошибок	Число паритетов		
	5	7	9
1	52	66	74
2	41	58	68
3	32	51	63
4	24	44	56

Представленная диаграмма отображает зависимость количества пропущенных стадий от количества паритетов и от кратности независимых ошибок.

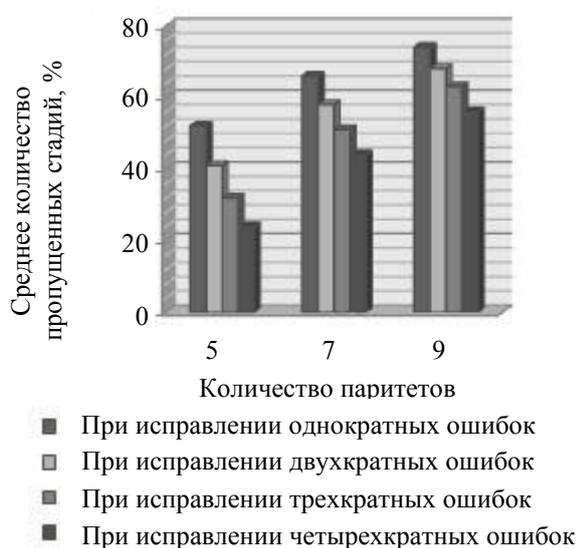


Диаграмма зависимости количества пропущенных стадий от количества паритетов и кратности независимых ошибок

Были выявлены следующие зависимости:

1) с ростом числа паритетов увеличивается среднее количество стадий, на которых ошибки не будут исправлены. Так, в случае исправления однократной ошибки при использовании матрицы с пятью паритетами было в среднем пропущено 52% стадий, при семи па-

ритетах – в среднем 66%, а при девяти – в среднем 74%;

2) с ростом кратности независимых ошибок среднее количество пропущенных стадий будет уменьшаться. Например, при использовании проверочной матрицы с пятью паритетами для исправления однократных ошибок число стадий сокращается в среднем на 52%, при исправлении двухкратных – в среднем на 41%, при трехкратных – в среднем на 32%, а при четырехкратных – в среднем на 24%.

Стоит отметить, что длина декодируемой информационной последовательности на количество пропущенных стадий не влияет.

Как видно из табл. 2, пропуская стадии, можно значительно сократить время декодирования.

Заключение. По результатам проведенного исследования можно сделать вывод, что на первой стадии декодирования можно определить, какие из последующих стадий целесообразно пропустить, так как на них ни один из информационных бит не будет инвертирован. Таким образом, можно автоматически в процессе исправления ошибок уменьшить число стадий декодирования, сократив тем самым время декодирования в целом.

Литература

1. Романенко, Д. М. Многопороговое мажоритарное декодирование низкоплотных кодов / Д. М. Романенко, Д. В. Шиман, М. Ф. Виткова // Труды БГТУ. – 2011. – № 6: Физ.-мат. науки и информатика. – С. 128–132.
2. Urbanovich, P. P. Multithreshold majority decoding of LDP-codes / P. P. Urbanovich, D. M. Romanenko, D. V. Shiman // News electrical and electronic technologies and their industrial implementation: proc. VII International Conference, Zakopane, Poland, June 28 – July 1, 2011. – Zakopane, 2011. – P. 121.
3. Виткова, М. Ф. Программное средство для анализа свойств многопорогового декодера в системах передачи информации / М. Ф. Виткова // Сборник научных работ 62-й НТК студентов и магистрантов БГТУ, Минск, 18–23 апр. 2011 г. / Белорус. гос. технол. ун-т. – Минск, 2011. – Ч. 3. – С. 67–70.

Поступила 25.02.2012