УДК 621.391.01

КОРРЕКЦИЯ КОДОВЫХ СЛОВ НА ОСНОВАНИИ ПРОВЕРОК ЧЕТНОСТИ

А.С. ПОЛЯКОВ, В.Е. САМСОНОВ

Объединенный институт проблем информатики Национальной академии наук Беларуси,
Сурганова, 6, Минск, 220012, Беларусь

Поступила в редакцию 27 июля 2015

Для поиска ошибок в кодовых словах предлагается использовать проверку на четность по осям и диагоналям (главной и вспомогательной) кодовой плоскости. Алгоритм, использующий результаты таких проверок, позволяет находить кратные ошибки в строках и столбцах кодовой плоскости.

Ключевые слова: кодовые слова, кодовая плоскость, ошибки, четность по диагоналям.

Введение

В работе [1] предложен способ поиска адресов одиночных ошибок в кодовых словах двухмерных кодов, основанный на использовании результатов проверок четности по осям и главной диагонали кодовой плоскости (бинарной матрицы), представляющей кодовые слова. Для поиска ошибок в [1] использовались результаты проверок четности по осям X, Y [2–5] и по главной диагонали матрицы, предложенной в [6]. Под главной диагональю в [6] понимаются как основная главная диагональ матрицы, так и все параллельные ей диагонали.

Проверки четности по строкам, столбцам и главным диагоналям матрицы позволяют использовать для вычисления адресов элементов матрицы три координаты – номера строк, столбцов и главных диагоналей. Но, как было отмечено в [1], этих данных во многих случаях недостаточно для поиска двойных и более сложных типов ошибок в матрице.

В настоящей работе предлагается расширить возможности определения адресов ошибочных элементов путем введения в рассмотрение проверки на четность по вспомогательным диагоналям. Вспомогательными будем считать как основную вспомогательную диагональ матрицы, так и все параллельные ей диагонали. Значения проверочных символов четности по главным и вспомогательным диагоналям вычисляются согласно представленной на рис. 1 схеме, где в клетках матрицы вверху указаны номера главных, а внизу – вспомогательных диагоналей. Строка со значениями проверочных символов по главным диагоналям обозначена через $eTPC$, а для проверочных символов по вспомогательным диагоналям – через $\underline{eTPC}$.

Введем обозначения: m, n – количества строк и столбцов матрицы соответственно; a – номера строк; b, $\underline{b}$ – номера столбцов в прямом и обратном направлениях; d – номера главных диагоналей; $\underline{d}$ – номера вспомогательных диагоналей; (a,b) – элемент матрицы на пересечении строки a и столбца b; $d\ (a,b)$ – номер главной диагонали, соответствующий элементу (a,b); $\underline{d}\,(a,b)$ – номера вспомогательной диагонали, соответствующий элементу (a,b); $N\,(a,b)$ – порядковый номер элемента (a,b) в матрице; $N\ (1,1) = 1$; SX – список номеров ошибочных строк; SY – список номеров ошибочных столбцов; SE – список номеров ошибочных главных диагоналей; $\underline{SE}$ – список номеров ошибочных вспомогательных диагоналей.

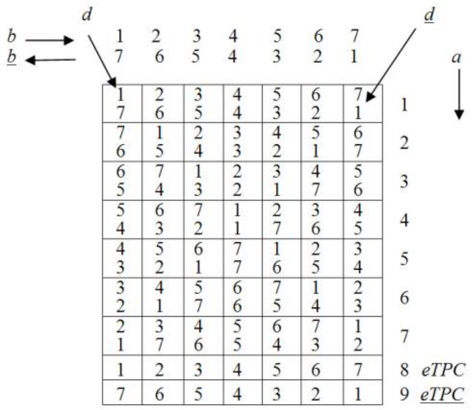

Рис. 1. Размещение номеров диагоналей и проверочных символов

С учетом указанных обозначений выведем достаточно очевидные формулы вычисления номеров строк, столбцов и диагоналей:

$d\,(a,b) = (n - a + b + 1)\ mod\ n,$ (1)

$b\,(a,d) = (a + d - n - 1)\ mod\ n,$ (2)

$a\,(b,d) = (b + n - d + 1)\ mod\ n,$ (3)

$\underline{b} = n - b + 1,$ (4)

$\underline{d}\,(a,\underline{b}) = (n - a + \underline{b} + 1)\ mod\ n,$ (5)

$N\,(a,b) = (a - 1)*\,n + b.$ (6)

Необходимо учитывать, что: если $b < 0$, то $b\ mod\ n = n - b\ mod\ n$, если $b = 0$, или $b = n$, то $b\ mod\ n = n$. Подставив значение $\underline{b}$ из (4) в формулу (5), найдем
$\underline{d}\,(a,b) = (n - a + (n - b + 1) + 1)mod\ n = (2\,n + 2 - a - b)mod\ n.$ (7)

Признаком наличия двойных ошибок в строках или столбцах является разница в количестве элементов в списках *SX*, *SY*, *SE* и <u>*SE*</u>, а именно: если есть одна строка с двойной ошибкой, то в *SY*, *SE* и <u>*SE*</u> будет на два элемента больше, чем в *SX*; если две строки с двойными ошибками, то в *SY*, *SE* и <u>*SE*</u> на четыре элемента больше, чем в *SX* и т.д.

Наличие двойных ошибок уменьшает число координат для вычисления адресов элементов (при двойной ошибке в строке отсутствует номер этой строки в *SX*, при двойной ошибке в столбце – номер столбца в *SY*), поэтому в качестве дополнительной информации будем использовать номера диагоналей из списков *SE* и <u>*SE*</u>.

Описание способа поиска ошибок

Способ предусматривает формирование множества строк S1–S2–S3–S4, где S1 и S2 – номера ошибочных строк и столбцов из списков *SX*, *SY*, а S3 и S4 – номера диагоналей из *SE* и <u>*SE*</u>, соответствующих элементам матрицы, адреса которых указаны в S1 и S2, с последующим анализом строк множества S1–S2–S3–S4 с целью исключения адресов ложных ошибок.

Для вычисления значений строк в S1–S2–S3–S4 используются формулы (1)–(7). Заметим, что при вычислении номеров диагоналей в столбце S4, принимаются во внимание только строки с номерами от 1 до (m–2), т.е. исключаются строки *eTPC* и <u>*eTPC*</u>.

Алгоритмы поиска при наличии только двойных ошибок в строках или столбцах отличаются формированием и анализом множества строк S1–S2–S3–S4. В качестве примера приведем алгоритм поиска двойных ошибок в строках матрицы **(**в списке *SX* нет элементов**).**

1. Формируем множество строк S1–S2–S3–S4, где S2 и S3 – номера столбцов и главных диагоналей из списков *SY* и *SE*, S1 и S4 – номера строк и вспомогательных диагоналей, соответствующих элементам матрицы на пересечении столбцов и диагоналей, указанных в S2 и S3.

2. Из S1–S2–S3–S4 удаляются строки, в которых: значение столбца S4 отсутствует в <u>*SE*</u>**;** значение столбца S1 превышает (m–2); имеются одинаковые значения в столбце S4.

3. Если в S1–S2–S3–S4 не осталось строк, то это означает невозможность определения адресов ошибочных элементов. Переход в п.6. Если в S1–S2–S3–S4 остались только две строки (при наличии в матрице одной строки с двойной ошибкой), или только четыре строки (при наличии двух строк с двойными ошибками), в которых значения в столбце S1 одинаковы, то эти строки представляют собой адреса двойных ошибок в строках, номера которых указаны в S1. Из *SY*, *SE* и <u>*SE*</u> удаляются элементы, значения которых присутствуют в оставшихся строках, а сами строки удаляются из S1–S2–S3–S4.

4. Если кроме строк, указанных в п.3 (или отсутствии таковых), в S1–S2–S3–S4 еще есть строки, то они тоже представляют адреса ошибочных элементов. Из *SX*, *SY*, *SE* и <u>*SE*</u> удаляются элементы, значения которых присутствуют в оставшихся строках, при этом, если в *SX* отсутствуют значения столбца S1, то такие значения записываются в *SX*, а сами строки удаляются из S1–S2–S3–S4.

5. Если в *SX* и *SY* больше нет элементов, то переход в п. 6, иначе переход в п. 1.

6. Останов алгоритма.

Рассмотрим работу приведенного алгоритма на примере матрицы M1, рис. 2. Строки 2 и 7 содержат двойные ошибки (отмечены ■). Номера ошибочных диагоналей отмечены символами •• в строках *eTPC* и <u>*eTPC*</u>.

1. Запишем в столбцы S2 и S3 все пары элементов из списков *SY* и *SE*, в столбцы S1 и S4 – номера строк и вспомогательных диагоналей, которые соответствуют элементам матрицы, значения которых указаны в столбцах S2 и S3. Для вычисления значений столбцов S1 и S4 используем приведенные выше формулы (3) и (7).

1	2	3	4	5	6	7	8	9		
1 9	2 8	3 7	4 6	5 5	6 4	7 3	8 2	9 1	1	
9 8	1■ 7	2 6	3 5	4 4	5■ 3	6 2	7 1	8 9	2	
8 7	9 6	1 5	2 4	3 3	4 2	5 1	6 9	7 8	3	
7 6	8 5	9 4	1 3	2 2	3 1	4 9	5 8	6 7	4	
6 5	7 4	8 3	9 2	1 1	2 9	3 8	4 7	5 6	5	
5 4	6 3	7 2	8 1	9 9	1 8	2 7	3 6	4 5	6	
4 3	5 2	6 1	7■ 9	8 8	9 7	1 6	2■ 5	3 4	7	
3	4	5••	6	7••	8	9	1••	2••	8	*eTPC*
2	1	9••	8	7••	6	5••	4	3••	9	<u>*eTPC*</u>

SX	*SY*	*SE*	<u>*SE*</u>		S1	S2	S3	S4	
-	2	1	7	1)	**2**	**2**	**1**	**7**	+
-	4	5	3	2)	7	2	5	2	--
-	6	7	9	3)	5	2	7	4	--
-	8	2	5	4)	1	2	2	8	--
				5)	4	4	1	3	---
				6)	9	4	5	7	---
				7)	7	4	7	9	---
				8)	3	4	2	4	--
				9)	6	6	1	8	--
				10)	2	6	5	3	---
				11)	9	6	7	5	---
				12)	5	6	2	9	---
				13)	8	8	1	4	--
				14)	4	8	5	8	--
				15)	2	8	7	1	--
				16)	**7**	**8**	**2**	**5**	+
2	4	5	3	17)	9	4	5	8	--
7	6	7	9	18)	**7**	**4**	**7**	**9**	+
-	-	-	-	19)	2	6	5	3	+
				20)	**9**	**6**	**7**	**6**	--

Рис. 2. Матрица M1

2. Из множества S1–S2–S3–S4 удалим строки, в которых значения столбца S4 отсутствуют в списке <u>*SE*</u> – это строки 2, 3, 4, 8, 9, 14, 15; отмеченные символами --; строки, в которых значения столбца S1 превышают (*m*–2) : строки 6, 11, 13; строки с одинаковыми значениями в столбце S4: 5, 10 и 7, 12.

3. Остались строки 1 и 16, отмеченные знаком +. Среди оставшихся нет ни одной пары строк с одинаковыми значениями в столбце S1.

4. Оставшиеся строки представляют адреса ошибочных элементов (2,2) в строке 2 и (7,8) в строке 7. Удаляем из *SY* значения 2 и 8, из *SE* – 1 и 2, из <u>*SE*</u> – 7 и 5, содержащиеся в строках 1 и 16. Поскольку значения 2 и 7 столбца S1 в строках 1 и 16 отсутствуют в *SX*, записываем эти значения в *SX*. Из S1–S2–S3–S4 удаляем строки 1 и 16.

5. В списках *SX* и *SY*, представленных под сплошной чертой, есть элементы, поэтому переход в п.1. Формируем новое множество S1–S2–S3–S4 (строки 17÷20). После удаления строк 17 и 20, в которых значения столбца S4 отсутствуют в <u>*SE*</u>, остаются строки 18 и 19, представляющие ошибки (7,4) в строке 7 и (2,6) в строке 2. После удаления из списков *SX*, *SY*, *SE* и <u>*SE*</u> содержащихся в этих строках значений в списках *SX* и *SY* элементов нет. Следовательно, все ошибки обнаружены.

6. Останов алгоритма.

Алгоритм поиска ошибок по результатам проверок на четность

0. Принимается pr := 0.

1. Анализируется состояние списков *SX*, *SY*, *SE* и $\underline{SE}$. Если во всех списках нет элементов, то это означает отсутствие ошибок, переход в п.11. Если в каких-либо двух из списков *SX*, *SY*, *SE*, $\underline{SE}$ нет элементов, или двойные ошибки есть более чем в двух координатах (имеются в виду строки, столбцы, главные диагонали, вспомогательные диагонали), то это означает недостаточность данных и невозможность обнаружения ошибок, переход в п. 11. Если и в *SX* и в *SY* имеются элементы, переход в п. 3, иначе переход в п. 2.

2. Если в *SX* отсутствуют элементы или число элементов в *SX* меньше, чем в *SE*, то переход в п. 8, если в *SY* нет элементов или число элементов в *SY* меньше, чем в *SE*, то переход в п. 6, в противном случае переход в п. 3.

3. Сформируем множество S1–S2–S3–S4 из элементов списков *SX*, *SY*, *SE* и $\underline{SE}$, где S1–S2 представляют все пары элементов из *SX* и *SY*, а S3 и S4 – соответствующие им номера главных и вспомогательных диагоналей.

3.1. Если в *SE* имеются элементы, то переход в п. 3.2, иначе из S1–S2–S3–S4 удаляются строки, в которых значения столбца S4 отсутствуют в списке $\underline{SE}$. Если среди оставшихся имеются пары строк с одинаковыми значениями в столбце S3, то эти строки представляют адреса ошибочных символов, из *SX*, *SY* и $\underline{SE}$ удаляются присутствующие в этих строках элементы, сами строки удаляются из S1–S2–S3–S4 и переход в п. 1; иначе переход в п. 4.

3.2. Если в $\underline{SE}$ есть элементы, то переход в п. 4, иначе из S1–S2–S3–S4 удаляются строки, в которых значения столбца S3 отсутствуют в списке *SE*. Если среди оставшихся имеются пары строк с одинаковыми значениями в столбце S4, то они представляют адреса ошибочных символов, из *SY* и *SE* удаляются присутствующие в этих строках элементы, сами строки удаляются из S1–S2–S3–S4 и переход в п. 1, иначе переход в п. 4.

4. Из множества S1–S2–S3–S4 последовательно удаляются строки, в которых значения столбцов S3 и S4 отсутствуют в списках *SE* и $\underline{SE}$ соответственно, и строки, имеющие одинаковые значения в столбце S4.

Если после удалений в S1–S2–S3–S4 строк не осталось, то pr := pr + 1. Если pr = 3, то переход в п. 11, иначе переход в п. 2.

5. Из *SX*, *SY*, *SE* и $\underline{SE}$ удаляются присутствующие в этих строках элементы, если в этих строках значения столбцов S1, S2, S3, S4 отсутствуют в *SX*, *SY*, *SE*, $\underline{SE}$, то эти значения заносятся в списки *SX*, *SY*, *SE*, $\underline{SE}$ соответственно. Сами строки удаляются из S1–S2–S3–S4. Переход в п. 1.

6. Составляются все пары элементов из списков *SX* и *SE* и записываются в столбцы S1 и S3, по значениям этих пар вычисляются соответствующие номера столбцов и вспомогательных диагоналей, которые записываются в S2 и S4. Из S1–S2–S3–S4 удаляются строки, в которых значение в S4 отсутствует в $\underline{SE}$, строки, имеющие одинаковые значения в S4 и S1. Если в S1–S2–S3–S4 не осталось строк, то pr := pr + 1; если pr = 3, переход в п. 11, иначе переход в п. 8.

7. Если в S1–S2–S3–S4 остались только одна или несколько пар строк, имеющих одинаковые значения в столбце S2, то эти строки представляют адреса ошибочных элементов в столбцах матрицы, номера которых указаны в S2. Элементы, содержащиеся в этих строках, удаляются из *SX*, *SE* и $\underline{SE}$, сами строки удаляются из S1–S2–S3–S4 и переход в п. 1; в противном случае из S1–S2–S3–S4 удаляются строки, в которых значения S2 отсутствуют в *SY*.

Если в S1–S2–S3–S4 еще остались строки, то они представляют адреса ошибочных элементов. Из *SX*, *SY*, *SE* и $\underline{SE}$ удаляются входящие в эти строки элементы, при этом, если в *SY* отсутствует значение столбца S2 в удаляемой строке, то это значение записывается в *SY*, а строка удаляется из S1–S2–S3–S4. Если в *SX* и в *SY* больше нет элементов, то переход в п. 11, иначе переход в п. 1.

8. Составляются все пары элементов из *SY* и *SE* и записываются в S2 и S3, по их значениям вычисляются соответствующие номера строк и вспомогательных диагоналей, которые записываются в столбцы S1 и S4.

Из S1–S2–S3–S4 последовательно удаляются строки, в которых: значения столбца S4 отсутствуют в списке $\underline{SE}$; значение столбца S1 превышает (m–2); строки с одинаковыми значениями в столбце S4; строки с одинаковыми значениями в столбце S2.

9. Если в S1–S2–S3–S4 остались только одна или несколько пар строк с одинаковыми значениями в S1, то эти строки указывают адреса ошибочных элементов в матрице. Из *SY*, *SE* и $\underline{SE}$ удаляются элементы, содержащиеся в этих строках, сами строки удаляются из S1–S2–S3–S4, переход в п. 1, в противном случае из S1–S2–S3–S4 удаляются строки, в которых значения столбца S1 отсутствуют в *SX*, переход в п.10.

Если в S1–S2–S3–S4 не осталось строк, то pr := pr+1. Если pr = 3, переход в п. 11, иначе переход в п. 6.

10. Оставшиеся строки представляют адреса ошибочных элементов. Из *SX*, *SY*, *SE* и $\underline{SE}$ удаляются содержащиеся в этих строках элементы, при этом, если в *SX* нет значения столбца S1 в удаляемой строке, то это значение записывается в *SX*, сама строка удаляется из S1–S2–S3–S4. Переход в п. 6.

11. Останов алгоритма.

Рассмотрим работу алгоритма на примере матрицы M2, представленной на рис. 3, в которой имеется одна строка с тремя ошибками и две строки с одиночными ошибками. В соответствии с п. 1 алгоритма образуем множество строк S1–S2–S3–S4. Здесь и далее курсивом будем выделять значения тех столбцов, которые являются образующими для нахождения остальных элементов строки. В данном случае это значения столбцов S1 и S2. Удаляем строки в соответствии с п. 4 (отмечены символами --,---). Остались строки 2 и 6 (отмечены знаком +), представляющие адреса ошибок. Из списков *SX*, *SY*, *SE* и $\underline{SE}$ удаляем элементы, содержащиеся в строках 2 и 6. Оставшиеся в списках значения показаны под штриховой линией. Переход в п. 1.

Формируем новое множество S1–S2–S3–S4 (строки 16–18), удаляем из него строки 16 и 17, в которых значения столбца S4 отсутствуют в списке $\underline{SE}$. Остается только строка 18, указывающая адрес ошибки (7,7). Удаляем из *SX*, *SY*, *SE* и $\underline{SE}$ значения, содержащиеся в строке 18. Переход в п. 1.

Поскольку в списке *SX* нет элементов, то переход в п. 8. Формируется множество S1–S2–S3–S4 (строки 19–22), на основе значений списков *SY* и *SE* (выделены в строках 19–22 курсивом). После удаления строк 20 и 21, в которых значения столбца S4 отсутствуют в списке $\underline{SE}$, остается только одна пара строк – 19 и 22, в которых значения столбца S1 одинаковы. Эти строки представляют адреса двух ошибок в строке 4. Удаляем из *SY*, *SE* и $\underline{SE}$ значения строк 19 и 22, а сами строки – из S1–S2–S3–S4. Переход в п. 1. Поскольку в списках *SX*, *SY*, *SE* и $\underline{SE}$ больше нет элементов, то это означает, что все ошибки обнаружены.

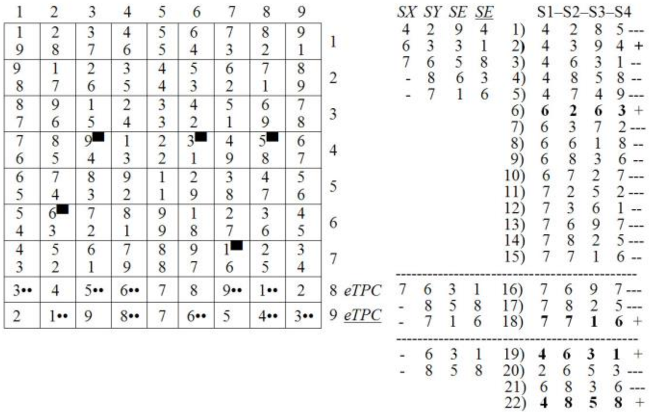

Рис. 3. Матрица M2

Другой пример представлен на рис. 4. Матрица M3 содержит двойные ошибки в строке 6 и в главной диагонали 3. Выполнение пунктов 1–5 позволяет найти ошибку (4,7). Повторное выполнение этих же пунктов (строки 9–11), и пунктов 6, 7 (строка 12) не дает результата. Последующее выполнение п.п. 8–10 (строки 13–15) приводит к нахождению ошибки (6,3). При этом обнаруживается, что в списке *SX* нет значения 6, имеющегося в строке 13, поэтому в список *SX* заносится значение 6.

В списке *SE* нет элементов, поэтому выполняются п.п. 1–3, 3.1 (строки 16–19) и после удаления строк 17, 18, в которых значения столбца S4 отсутствуют в $\underline{SE}$, в S3 остаются только две строки (16 и 19) с одинаковыми значениями в столбце S3, которые представляют ошибочные элементы (2,4) и (6,8) в главной диагонали с номером 3. После удаления из *SX*, *SY*, *SE* и *SE* значений строк 16 и 19 в этих списках не осталось элементов, что является свидетельством того, что все ошибки обнаружены.

1	2	3	4	5	6	7	8	9	
1 9	2 8	3 7	4 6	5 5	6 4	7 3	8 2	9 1	1
9 8	1 7	2 6	3 5	4 4	5 3	6 2	7 1	8 9	2
8 7	9 6	1 5	2 4	3 3	4 2	5 1	6 9	7 8	3
7 6	8 5	9 4	1 3	2 2	3 1	4 9	5 8	6 7	4
6 5	7 4	8 3	9 2	1 1	2 9	3 8	4 7	5 6	5
5 4	6 3	7 2	8 1	9 9	1 8	2 7	3 6	4 5	6
4 3	5 2	6 1	7 9	8 8	9 7	1 6	2 5	3 4	7
3	4••	5	6	7••	8	9	1	2	8 *eTPC*
2••	1	9••	8	7	6••	5••	4	3	9 *eTPC*

SX	*SY*	*SE*	*SE*	S1–S2–S3–S4					
2	3	7	2	1)	2	3	2	6	--
4	4	4	5	2)	2	4	3	5	--
-	7	-	9	3)	2	7	6	2	--
-	8	-	6	4)	2	8	7	1	--
				5)	4	3	9	4	--
				6)	4	4	1	3	--
				7)	**4**	**7**	**4**	**9**	+
				8)	4	8	5	8	--
2	3	7	2	9)	2	3	2	6	--
-	4	-	5	10)	2	4	3	5	--
-	8	-	6	11)	2	8	7	1	--
				12)	2	8	7	1	--
				13)	**6**	**3**	**7**	**2**	+
				14)	7	4	7	9	--
				15)	2	8	7	1	--
2	4	-	5	16)	**2**	**4**	**3**	**5**	+
6	8	-	6	17)	2	8	7	1	--
				18)	6	4	8	1	--
				19)	**6**	**8**	**3**	**6**	+

Рис. 4. Матрица М3

Поиск ошибок не всегда заканчивается успешно. Например, не удается определить адреса ошибок в случае, представленном на рис. 5. В матрице М4 имеются только две ошибки, но после удаления из множества S1–S2–S3–S4 строк, имеющих одинаковые значения в столбце S4, не остается ни одной строки и поэтому найти адреса ошибочных элементов невозможно. А вот в более сложном случае, при наличии в матрице двух строк с групповыми ошибками (рис. 6, матрица М5), легко определяются адреса всех ошибочных элементов.

1	2	3	4	5	6	
1 6	2 5	3 4	4 3	5 2	6 1	1
6 5	1 4	2 3	3 2	4 1	5 6	2
5 4	6 3	1 2	2 1	3 6	4 5	3
4 3	5 2	6 1	1 6	2 5	3 4	4
3 2	4 1	5 6	6 5	1 4	2 3	5
2••	3	4	5••	6	7	7 *eTPC*
7	6	5••	4	3	2••	8 *eTPC*

SX	*SY*	*SE*	*SE*
-	2	5	2
-	5	2	5

S1–S2–S3–S4				
4	2	5	2	--
1	2	2	5	---
1	5	5	2	--
4	5	2	5	---

Рис. 5. Матрица М4

1	2	3	4	5	6	7	8	9	
1 9	2 8	3 7	4 6	5 5	6 4	7 3	8 2	9 1	1
9 8	1 7	2 6	3 5	4 4	5 3	6 2	7 1	8 9	2
8 7	9 6	1 5	2 4	3 3	4 2	5 1	6 9	7 8	3
7 6	8 5	9 4	1 3	2 2	3 1	4 9	5 8	6 7	4
6 5	7 4	8 3	9 2	1 1	2 9	3 8	4 7	5 6	5
5 4	6 3	7 2	8 1	9 9	1 8	2 7	3 6	4 5	6
4 3	5 2	6 1	7 9	8 8	9 7	1 6	2 5	3 4	7
3	4••	5••	6••	7	8••	9••	1••	2	8 *eTPC*
2••	1	9	8	7••	6••	5••	4••	3••	9 *eTPC*

SX	*SY*	*SE*	*SE*	S1–S2–S3–S4				
2	1	8	7	2	1	9	8	x
3	2	9	6	2	2	1	7	---
	3	1	5	2	3	2	6	x
	5	4	4	**2**	**5**	**4**	**4**	+
	6	5	3	2	6	5	3	---
	7	6	2	**2**	**7**	**6**	**2**	+
				3	**1**	**8**	**7**	+
				3	2	9	6	---
				3	3	1	5	+
				3	5	3	3	x
				3	6	4	2	---
				3	7	5	1	x
2	2	9	6	2	2	1	7	x
3	6	5	3	**2**	**6**	**5**	**3**	+
				3	**2**	**9**	**6**	+
				3	6	4	2	x

Рис. 6. Матрица М5

Причиной неудачного поиска ошибок является наличие большого количества одинаковых номеров диагоналей, соответствующих различным элементам матрицы. Ведь в каждой строке матрицы содержатся номера диагоналей от 1 до *n*, поэтому имеется *n* множеств элементов матрицы, в каждом из которых элементам матрицы соответствуют одинаковые номера диагоналей. А поскольку поиск ошибок основан на анализе вариантов размещения (адресов) ошибочных элементов, которым могут соответствовать элементы матрицы с одинаковыми номерами диагоналей, то поэтому в множестве S1–S2–S3–S4 появляются строки с одинаковыми значениями в столбцах S3 и S4, что создает неоднозначность в определении адресов ошибок.

Заключение

1. Предлагаемый способ является «ситуационным», поскольку при некоторых вариантах размещения ошибок в матрице нахождение всех ошибок может оказаться невозможным из-за совпадения номеров диагоналей в рассматриваемых вариантах возможного размещения ошибок. С увеличением размеров матрицы (точнее – числа столбцов) вероятность обнаружения ошибок повышается, поскольку при этом уменьшается вероятность появления нескольких строк с одинаковыми значениями в столбцах S3, S4. Действительно, вероятность выбора некоторого произвольного элемента матрицы, которому соответствует номер диагонали в диапазоне от 1 до n, равна $1/n$, а вероятность выбора второго элемента матрицы с таким же номером диагонали равна $1/n^2$.

2. Способ поиска обеспечивает высокую скорость кода. Для матрицы, имеющей n столбцов и m строк, в которой последний столбец представляет проверочные символы четности по горизонтали, а последние три строки – проверочные символы четности по вертикали, главной диагонали и вспомогательной диагонали соответственно, скорость кода равна $(m–3)*(n–1) / m*n$. Например: при $n = 7$ и $m = 6$ скорость кода равна $(6–3)*(7–1) / 6*7 = 0,43$; при $n = 30$ и $m = 27$ скорость кода $= (27–3)*(30–1) / 27*30 = 0,86$. Следовательно, с увеличением числа столбцов в матрице скорость кода приближается к единице.

CODEWORDS CORRECTION ON THE BASIS OF PARITY CHECKS

A.S. POLJAKOV, V.E. SAMSONOV

Abstract

To find errors in codewords the parity check on axis and diagonals (main and auxiliary) of code plain is proposed. The results of such checks are used in algorithm to find multiple errors in lines and columns of code plane.

Keywords: codewords, code plain, errors, parity on diagonals.

Список литературы

1. *Поляков А.С., Самсонов В.Е.* Устранение ошибок в кодовых словах по результатам проверок на четность. // Докл. БГУИР. 2015 . № 8 (94). С. 51–56.
2. *Вернер М.* Основы кодирования. М., 2004.
3. *Питерсон У., Уэлдон Э.* Коды, исправляющие ошибки. М., 1976.
4. *Морелос-Сарагоса Р.* Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. М., 2005.
5. Сорока Н.И., Кривиченко Г.А. Телемеханика. Часть 2. Коды и кодирование. Минск. [Электронный ресурс]. – Режим доступа: http://www/bsuir.by/m/12_100229_1_62250.pdf/. – Дата доступа: 16.06.2015.
6. "Enchanced" Turbo Product Codes (eTPC). [Электронный ресурс]. – Режим доступа: www.aha.com/Uploads/ANtpc12_03063.pdf. – Дата доступа: 10.07.2015.