DOI: 10.18372/2410-7840.20.12450
УДК 681.391

СИСТЕМАТИЧЕСКИЕ БАЙТ-ОРИЕНТИРОВАННЫЕ КОДЫ

Анатолий Белецкий, Дмитрий Конюший, Дмитрий Полторацкий

Порядок (число разрядов или длина) классических циклических кодов как правило не является кратным целому числу байтов, что приводит к непроизводительным затратам вычислительных ресурсов при их аппаратно-программной реализации. В силу указанной причины переход к байт-ориентированным кодам, в которых как длина k информационных слов $\boldsymbol{I}$, так и число r проверочных разрядов $\boldsymbol{R}$ кратны целому числу байтов, представляется наиболее целесообразным для практического применения. Отличительная особенность предлагаемого подхода к синтезу (кодированию информации) и анализу кодов (декодированию сообщений) состоит в отказе от образующих $\boldsymbol{G}$ и проверочных матриц $\boldsymbol{H}$, обычно сопровождающих систематические циклические коды, и их замене на единственную матрицу $\boldsymbol{P}$ проверочных символов (МПС), меньшую по объёму по сравнению с применяющимися матрицами $\boldsymbol{G}$ и $\boldsymbol{H}$. Основу формирования МПС циклических $(n, k, t)-$ кодов, где n – длина кода и t – кратность устраняемых ошибок в кодовых словах, составляют образующие (порождающие) полиномы (одномерные двоичные векторы), обозначаемые символом $\boldsymbol{b}$. Двоичный полином $r-$ й степени тогда и только тогда является образующим полиномом примитивного циклического $(n, k, t)-$ кода, когда так называемая «контрольная» $(k+1)-$ я строка s_{k+1}, являющаяся продолжением матрицы проверочных символов $\boldsymbol{P}$ кода и вычисляемая по правилам формирования строк этой матрицы, но не входящая в неё, определяется соотношением $s_{k+1} = 0^{[r-1]}1$ (необходимые условия), причём вес v каждой строки матрицы проверочных символов не меньше чем $2t$, а расстояние Хемминга $d_{i,j}$ между любыми парами строк (s_i, s_j) матрицы $\boldsymbol{P}$ таково, что $d_{i,j} \geq 2t-1$ (достаточные условия). Двойственные матрицы проверочных символов, т.е. матрицы, порождаемые двойственными двоичными полиномами, взаимно связаны операторами инверсной перестановки строк и столбцов матриц. Разработан систематический помехоустойчивый $(16, 8, 2)-$ код, порождаемый симметричным неприводимым полиномом восьмой степени $\boldsymbol{b} = 100111001$, являющийся уникальным (т.е. единственным в своём роде) и по ряду критериев оптимальным в классе байт-ориентированных кодов. Приводится развёрнутая характеристика алгоритма синдромного декодирования байт-ориентированных кодов.

Ключевые слова: *байт-ориентированные коды, образующие и проверочные матрицы, матрицы проверочных символов, синдромное декодирование.*

1. Введение

Как правило любая информация (текстовая, графическая, музыкальная и пр.) хранится в компьютерах в виде массивов бинарных данных, упорядоченных в форме последовательности байтов, являющихся базовыми элементами вычислительных систем (структур). В том случае, когда возникает необходимость извлечь из памяти системы не весь байт, а только его некоторую часть, например, шесть или семь битов, то на выполнение такой операции затрачиваются дополнительные ресурсы, что приводит к снижению скорости вычислений или другим потерям.

Обратимся к широко используемым в теории и приложениях помехоустойчивого кодирования систематическим примитивным $(n, k, t)-$ кодам, устраняющих ошибки кратности t, размер которых $n = 2^m - 1$, $m \geq 3$, составляет нечётное число бит и, как следствие нечётности n, размер информационных k, или проверочных $r = n - k$ компонентов кодовых слов $\boldsymbol{C}$ становится не кратным восьми битам. К подобным относятся, например, циклические $(15, 7, 2)-$ коды, длина k информационных слов $\boldsymbol{I}$ которых равняется семи битам. А это означает, что для обеспечения помехоустойчивой передачи сообщений исходные данные, сохраняемые в ЭВМ в форме байт структурированного массива, должны быть предварительно «нарезаны» семибитными блоками с последующим присоединением к каждому блоку байта проверочных символов. Такая процедура семибитного разбиения байт организованных данных не совсем удобна как на этапе кодирования, так и декодирования сообщений. По этой причине представляется целесообразным переход к так называемым *байт-ориентированным кодам*.

Определение. *Байт-ориентированными помехоустойчивыми кодами будем называть такие систематические $(n, k, t)-$ коды, устраняющие ошибки кратности t, в которых число как информационных k, так и проверочных $r = n - k$ разрядов, а, следовательно, и длина n кодовых слов $\boldsymbol{C}$ кратны восьми битам (байту).*

Основная цель данной работы состоит в решении задач синтеза (построения) и анализа двоичных байт-ориентированных кодов (байт-ОК), последняя из которых (анализ) сводится, по сути, к декодированию сообщений такими кодами.

2. Общие соотношения [1]

Кодовые слова (векторы) $\boldsymbol{C}$ длины n получаются из k – битных векторов информационных символов $\boldsymbol{I}$ отображением

$$\boldsymbol{C} = \boldsymbol{I} \cdot \boldsymbol{G}, \qquad (1)$$

где $\boldsymbol{G}$ – прямоугольная двоичная (k, n) – матрица, содержащая k строк и n столбцов, и называемая *образующей* (или *порождающей*) *матрицей*, составленная конкатенацией единичной k – го порядка матрицы $\boldsymbol{E}$ и (k, r) – *матрицы* $\boldsymbol{P}$ *проверочных символов* (МПС), то есть

$$\boldsymbol{G} = \boldsymbol{E} \circ \boldsymbol{P}, \qquad (2)$$

где $\circ$ – знак конкатенации.

Коды (1), образующие матрицы которых $\boldsymbol{G}$ представлены в форме (2), являются *систематическими кодами*. Заметим, что у кодового слова (1), порождаемого матрицей (2), первые k символов $\boldsymbol{C}$ совпадают с информационными символами $\boldsymbol{I}$, так как подставив (2) в (1), имеем

$$\boldsymbol{C} = \boldsymbol{I} \circ \boldsymbol{I} \cdot \boldsymbol{P} = \boldsymbol{I} \circ \boldsymbol{R}, \qquad (3)$$

где $\boldsymbol{R}$ – *группа проверочных символов*, отвечающих информационному слову $\boldsymbol{I}$.

Поставим в соответствие образующей матрице $\boldsymbol{G}$ из (2) такую *проверочную* (n, r) – *матрицу* $\boldsymbol{H}$, предназначенную для тестирования принятых кодовых слов (установления факта наличия или отсутствия искажения), которая обеспечивает равенство

$$\boldsymbol{G} \cdot \boldsymbol{H} = \boldsymbol{0}. \qquad (4)$$

Соотношение (4) означает, что любое разрешённое кодовое слово $\boldsymbol{C}$, составленное линейной комбинацией строк образующей матрицы $\boldsymbol{G}$ (или состоящее из одной строки этой матрицы), при умножении на проверочную матрицу $\boldsymbol{H}$ формирует нулевой r – битный вектор $\boldsymbol{0}$. Условие (4) выполняется, если только

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{P} \\ \boldsymbol{E} \end{pmatrix}. \qquad (5)$$

В самом деле, формально перемножив правые части выражений (2) и (5), получим

$$\boldsymbol{G} \cdot \boldsymbol{H} = (\boldsymbol{E} \circ \boldsymbol{P}) \cdot \begin{pmatrix} \boldsymbol{P} \\ \boldsymbol{E} \end{pmatrix} = \boldsymbol{E} \cdot \boldsymbol{P} + \boldsymbol{P} \cdot \boldsymbol{E} = 2\boldsymbol{P} = \boldsymbol{0}, \qquad (6)$$

поскольку все алгебраические операции выполняются над полем F_2.

Предположим, что в канале передачи информации кодовое слово $\boldsymbol{C}$ возможно искажается аддитивной помехой $\boldsymbol{e}$ и на вход приёмного устройства поступает сообщение

$$\boldsymbol{C}^* = \boldsymbol{C} + \boldsymbol{e}. \qquad (7)$$

Решающим устройством приёмника вычисляется так называемый *синдром* $\boldsymbol{S}$ (варианты определений «синдрома» приводятся ниже по тексту) входного кодового слова $\boldsymbol{C}^*$, индицирующий состояние (поражён или не поражён вектор $\boldsymbol{C}$ помехой), по формуле

$$\boldsymbol{S} = \boldsymbol{C}^* \cdot \boldsymbol{H} = \boldsymbol{C} \cdot \boldsymbol{H} + \boldsymbol{e} \cdot \boldsymbol{H} = \boldsymbol{e} \cdot \boldsymbol{H}, \qquad (8)$$

так как, согласно (3) и (5),

$$\boldsymbol{C} \cdot \boldsymbol{H} = (\boldsymbol{I} \circ \boldsymbol{I} \cdot \boldsymbol{P}) \cdot \begin{pmatrix} \boldsymbol{P} \\ \boldsymbol{E} \end{pmatrix} = \boldsymbol{I} \cdot \boldsymbol{P} + (\boldsymbol{I} \cdot \boldsymbol{P}) \cdot \boldsymbol{E} = 2(\boldsymbol{I} \cdot \boldsymbol{P}) = \boldsymbol{0}.$$

Таким образом, синдром $\boldsymbol{S}$ зависит исключительно только от вектора помехи $\boldsymbol{e}$ и совершенно не зависит от исходного (неискажённого) кодового слова $\boldsymbol{C}$; причём, если $\boldsymbol{S} = \boldsymbol{0}$, то это означает, что слово $\boldsymbol{C}^*$ поступило на вход приёмного устройства без ошибок ($\boldsymbol{e} = \boldsymbol{0}$), тогда как при $\boldsymbol{S} \neq \boldsymbol{0}$ – входное слово содержит ошибку ($\boldsymbol{e} \neq \boldsymbol{0}$).

Представим вектор $\boldsymbol{C}^*$ на входе приёмника в виде конкатенации возможно искажённых информационной $\boldsymbol{I}^*$ и проверочной $\boldsymbol{R}^*$ компонент, то есть пусть

$$\boldsymbol{C}^* = \boldsymbol{I}^* \circ \boldsymbol{R}^*. \qquad (9)$$

Тогда с учётом (5), (8) и (9) получим

$$\boldsymbol{S} = \boldsymbol{C}^* \cdot \boldsymbol{H} = \oplus \begin{matrix} \boldsymbol{S}_{I^*} \\ \boldsymbol{R}^* \end{matrix} = \oplus \begin{matrix} \boldsymbol{I}^* \cdot \boldsymbol{P} \\ \boldsymbol{R}^* \end{matrix}. \qquad (10)$$

Следовательно, *синдром* $\boldsymbol{S}$ *циклического кода*, как и в любом систематическом коде, определяется суммой по модулю 2 элементов группы $\boldsymbol{S}_I$, сформированных из принятых информационных символов $\boldsymbol{I}$ преобразованием $\boldsymbol{S}_I = \boldsymbol{I} \cdot \boldsymbol{P}$, которую (группу $\boldsymbol{S}_I$) назовём *синдромом информационной группы*, и *группы проверочных символов* $\boldsymbol{R}$ входного кодового слова $\boldsymbol{C}$ [2].

Соотношения (3) и (10) приводят к заключению, что матрицы проверочных символов $\boldsymbol{P}$ играют конструктивную роль как на этапе кодирования информации (3), так и декодирования сообщений (10). Применение МПС $\boldsymbol{P}$ вместо традиционно используемых образующей $\boldsymbol{G}$ и проверочной

$\boldsymbol{H}$ матриц обеспечивают, по крайней мере, существенное сокращению размерности операндов, участвующих в матричных вычислениях.

Основу формирования МПС циклических (n, k, t) – кодов составляют *образующие* (*порождающие*) полиномы (одномерные двоичные векторы), которые обозначим символом $\boldsymbol{b}$. На строки матриц $\boldsymbol{P}$ накладываются такие ограничения [3-5]. Во-первых, вес ν_p каждой строки МПС должен быть не меньше чем $2t$, то есть

$$\nu_p \geq d_{\min} - 1 = 2t, \qquad (11)$$

где $d_{\min} = 2t+1$ – минимально допустимое расстояние Хемминга между произвольными парами кодовых слов $(\boldsymbol{C}_i, \boldsymbol{C}_j)$; $i \neq j$; $i, j = \overline{0, 2^n - 1}$. С учётом единицы, расположенной на главной диагонали матрицы $\boldsymbol{E}$, вес кодового слова $\boldsymbol{C}_i$, представляющего собой любую из строк матрицы $\boldsymbol{G}$, достигает требуемого значения, равного $2t+1$. И, во-вторых, расстояние Хемминга $d_{i,j}$ (совпадающее с весом кода, образованного поразрядной суммой по модулю 2 элементов этих строк) между всевозможными парами (i, j) строк МПС должно удовлетворять условию:

$$d_{i,j} \geq d_{\min} - 2 = 2t - 1. \qquad (12)$$

Ниже изложен достаточно простой алгоритм построения матриц $\boldsymbol{P}$, нумерация строк которых осуществляется по направлению снизу вверх, а столбцов – справа налево, начиная с номера 1. Предлагаемый алгоритм сводится к выполнению последовательности таких операций:

1. В нижней (первой) строке формируемой матрицы проверочных символов $\boldsymbol{P}$ размещается образующий r – й степени полином $\boldsymbol{b}$ за исключением его старшей единицы;

2. Содержимое младших $r-1$ разрядов предыдущей i – й строки (на старте заполнения МПС таковой является первая строчка) переписывается в последующую $(i+1)$ – ю строку матрицы со сдвигом на один разряд влево, т.е. размещается в разрядах r, $r-1$, …, 2, а в первый (правый, младший) разряд этой строки записывается нуль;

3. Если содержимое r – го разряда предыдущей строки, из которой выбирались младшие $r-1$ разрядов, равно 0, то переходим к п. 5, иначе – к п. 4;

4. Если же содержимое r – го разряда предыдущей строки равно 1, то осуществляется поразрядное сложение по модулю 2 строки МПС, составленной в п. 2, с первой строкой формируемой матрицы $\boldsymbol{P}$;

5. Последующие операции приостанавливаются в том случае, если не соблюдается одно из двух вышеприведенных условий (11) или (12), иначе – возврат к п. 2.

Причина, по которой следует выполнять операцию поразрядного сложения по модулю 2, указанную в п. 4, заключается в следующем. Предположим, что содержимое r – го разряда некоторой i – й строки МПС оказалось равным единице. Вышерасположенная $(i+1)$ – я строка матрицы $\boldsymbol{G}$ образуется в результате циклического сдвига i – й строки на один разряд влево. И, как следствие такого сдвига, появляется единичка в $(r+1)$ – м разряде строки формируемой матрицы $\boldsymbol{G}$, искажающая структуру расположенной слева от МПС матрицы $\boldsymbol{E}$. Для устранения искажения в $(i+1)$ – й строке матрицы $\boldsymbol{E}$, соответственно и $\boldsymbol{G}$, достаточно поразрядно просуммировать эту строчку матрицы $\boldsymbol{G}$ с её нижней (первой) строкой, которая в $(r+1)$ – м разряде также содержит единичку – правое замыкание главной диагонали матрицы $\boldsymbol{E}$. Подобная операция допустима, поскольку блочные циклические коды является линейными кодами. После завершении суммирования в $(r+1)$ – м разряде $(i+1)$ – й строки матрицы $\boldsymbol{G}$ вместо единички появляется нуль, что и приводит к восстановлению структуры единичной матрицы.

3. Синтез байт-ориентированных кодов

Переходим к построению байт-ориентированных (или байт-структурированных) кодов, совсем не обязательно циклических, начиная с кодов минимальной длины, устраняющих ошибки кратности t, то есть обратимся к задаче синтеза $(16, 8, t)$ – кодов. В качестве образующих полиномов, которые потенциально рассматриваются как претенденты на формирование МПС таких кодов, воспользуемся нечётными (и, следовательно, заканчивающимся на 1, полиномами восьмой степени с весом $\nu = 2t+1$. Число полиномов r – й степени $L_{r,\nu}$ с весом ν определяется числом сочетаний из $r-1$ по $\nu-2$, то есть $L_{r,\nu} = C_{\nu-2}^{r-1}$.

Все нечётные полиномы восьмой степени с весом, равном пяти, т.е. полиномы, которые потенциально могут оказаться образующими $(16, 8, t)$ – кодов, сведены в табл. 1.

Таблица 1

Множество полиномов восьмой степени с весом $\nu = 5$

№	Номер разряда									№	Номер разряда								
	9	8	7	6	5	4	3	2	1		9	8	7	6	5	4	3	2	1
										18	1	0	1	1	0	0	1	0	1
1	1	0	0	0	0	1	1	1	1	19	1	0	1	1	0	1	0	0	1
2	1	0	0	0	1	0	1	1	1	20	1	0	1	1	1	0	0	0	1
3	1	0	0	0	1	1	0	1	1	21	1	1	0	0	0	0	1	1	1
4	1	0	0	0	1	1	1	0	1	22	1	1	0	0	0	1	0	1	1
5	1	0	0	1	0	0	1	1	1	23	1	1	0	0	0	1	1	0	1
6	1	0	0	1	0	1	0	1	1	24	1	1	0	0	1	0	0	1	1
7	1	0	0	1	0	1	1	0	1	25	1	1	0	0	1	0	1	0	1
8	1	0	0	1	1	0	0	1	1	26	1	1	0	0	1	1	0	0	1
9	1	0	0	1	1	0	1	0	1	27	1	1	0	1	0	0	0	1	1
10	1	0	0	1	1	1	0	0	1	28	1	1	0	1	0	0	1	0	1
11	1	0	1	0	0	0	1	1	1	29	1	1	0	1	0	1	0	0	1
12	1	0	1	0	0	1	0	1	1	30	1	1	0	1	1	0	0	0	1
13	1	0	1	0	0	1	1	0	1	31	1	1	1	0	0	0	0	1	1
14	1	0	1	0	1	0	0	1	1	32	1	1	1	0	0	0	1	0	1
15	1	0	1	0	1	0	1	0	1	33	1	1	1	0	0	1	0	0	1
16	1	0	1	0	1	1	0	0	1	34	1	1	1	0	1	0	0	0	1
17	1	0	1	1	0	0	0	1	1	35	1	1	1	1	0	0	0	0	1

Среди 35 полиномов, представленных в табл. 1, не нашлось ни одного полинома, порождающего $(16, 8, t)$ – коды. Вместе с тем, эта таблица содержит три таких полинома, которые хотя и образуют циклические коды, но размеры информационных компонентов этих кодов не кратны байту. Таковыми являются пара двойственных полиномов $\boldsymbol{b}_2 = 100010111 = 11111 \cdot 11001$ и $\boldsymbol{b}_{33} = 111010001 = 11111 \cdot 10011$, составленных из произведений элементов разложения двучлена $x^{15}+1$ и порождающих классические примитивные циклические $(15, 7, 2)$ – коды, а также симметричный (и поэтому не имеющий дуального эквивалента) неприводимый полином (НП) восьмой степени

$$\boldsymbol{b}_{10} = 100111001, \qquad (13)$$

формирующих $(17, 9, 2)$ – код. Индексы i полиномов $\boldsymbol{b}_i$ совпадают с номерами полиномов, которые представлены в табл. 1.

Если первые два полинома $\boldsymbol{b}_2$ и $\boldsymbol{b}_{33}$ не оставляют каких-либо шансов для синтеза байт-ОК, поскольку длина k информационных слов $\boldsymbol{I}$ кодов, совпадающая с высотой МПС $\boldsymbol{P}$, равна семи, то третий полином $\boldsymbol{b}_{10}$ такую перспективу сохраняет. Других полиномов восьмой степени, кроме НП $\boldsymbol{b}_{10} = 100111001$, порождающих $(16, 8, 2)$ – байт-ОК, не существует. МПС, отвечающая полиному $\boldsymbol{b}_{10}$, показана в табл. 2.

Таблица 2

Матрица проверочных символов циклического $(17, 9, 2)$ – кода, порождаемого неприводимым полиномом $\boldsymbol{b} = 100111001$

№ строки		Разряды МПС							
		8	7	6	5	4	3	2	1
9		1	0	0	1	1	1	0	0
8		0	1	0	0	1	1	1	0
7		0	0	1	0	0	1	1	1
6		1	0	0	0	1	1	1	1
5		1	1	0	1	1	0	1	1
4		1	1	1	1	0	0	0	1
3		1	1	1	0	0	1	0	0
2		0	1	1	1	0	0	1	0
1	**1**	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**1**

В том, что полиномы $\boldsymbol{b}_2$ и $\boldsymbol{b}_{33}$ оказались составными полиномами, образуемыми произведением *двух* элементов разложения двучлена $x^{15}+1$, нет ничего неожиданного, поскольку это согласуется с хорошо известным в теории кодирования положением, согласно которому порождающий полином $\boldsymbol{b}$ примитивного циклического (n, k, t) – кода, устраняющего ошибки кратности не более t, может быть составлен из произведения t элементов разложения $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_t$ двучлена x^n+1, $n=2^m-1$, где m – натуральное число, не меньшее трёх, т.е. $\boldsymbol{b}=\boldsymbol{b}_t\cdot\boldsymbol{b}_t\cdot\ldots\cdot\boldsymbol{b}_t$.

Любой линейный циклический (n, k, t) – код (такой, что $r=n-k=8i$, где i – натуральное число), в котором $k>8$, но не кратно целому числу байтов, всегда можно привести к байт-ОК, исключая из МПС столько l верхних строк (что влечёт за собой так же обнуление l старших информационных символов слова $\boldsymbol{I}$), сколько необходимо, чтобы привести высоту матрицы $\boldsymbol{P}$ к целому числу байтов. Образуемые посредством таких сокращений матриц $\boldsymbol{P}$ коды называются *укороченными кодами*. При этом линейность вновь образованного кода сохраняется, но утрачивается свойство цикличности, поскольку приведение исходного кода к байт-ориентированной форме сопровождается удалением из информационных слов $\boldsymbol{I}$ их l старших символов. На этапе декодирования ранее выброшенные старшие разряды информационных компонент $\boldsymbol{I}$ кодовых слов $\boldsymbol{C}$ можно восстановить (записав нули в этих позициях кода), так что декодирование будет осуществляться по полной длине кода.

Таким образом, исключив из МПС, представленной в табл. 2, верхнюю (девятую) строку, приходим к возможности построения байт-ориентированного $(16, 8, 2)$ – кода, порождаемого полиномом $\boldsymbol{b}=100111001$, матрица проверочных символов которого такова

$$\boldsymbol{P}=\begin{pmatrix}1&0&0&1&1&1&0&0\\0&1&0&0&1&1&1&0\\0&0&1&0&0&1&1&1\\1&0&0&0&1&1&1&1\\1&1&1&1&0&0&0&1\\1&1&1&0&0&1&0&0\\0&1&1&1&0&0&1&0\\0&0&1&1&1&0&0&1\end{pmatrix}. \qquad (14)$$

Обратим внимание на то, что верхняя половина МПС (14) является зеркальным отображением её нижней половины с инверсно переставленными столбцами.

Скорость выполнения операций вычисления байт-ориентированных кодовых слов $\boldsymbol{C}$ по формуле (3) можно существенно повысить, воспользовавшись табличным способом (см. 16-ричную табл. 3, составленную на основании матрицы (14)) определения проверочных символов $\boldsymbol{R}$, отвечающих информационному слову $\boldsymbol{I}=\boldsymbol{i}_2\circ\boldsymbol{i}_1$.

Таблица 3

Проверочные символы $(16, 8, 2)$ – байт-ОК, порождаемого НП $\boldsymbol{b}=100111001$

$\longrightarrow \boldsymbol{i}_1$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	39	72	4B	E4	DD	96	AF	F1	C8	83	BA	15	2C	67	5E
1	DB	E2	A9	90	3F	06	4D	74	2A	13	58	61	CE	F7	BC	85
2	8F	B6	FD	C4	6B	52	19	20	7E	47	0C	35	9A	A3	E8	D1
3	54	6D	26	1F	B0	89	C2	FB	A5	9C	D7	EE	41	78	33	0A
4	27	1E	55	6C	C3	FA	B1	88	D6	EF	A4	9D	32	0B	40	79
5	FC	C5	8E	B7	18	21	6A	53	0D	34	7F	46	E9	D0	9B	A2
6	A8	91	DA	E3	4C	75	3E	07	59	60	2B	12	BD	84	CF	F6
7	73	4A	01	38	97	AE	E5	DC	82	BB	F0	C9	66	5F	14	2D
8	4E	77	3C	05	AA	93	D8	E1	BF	86	CD	F4	5B	62	29	10
9	95	AC	E7	DE	71	48	03	3A	64	5D	16	2F	80	B9	F2	CB
A	C1	F8	B3	8A	25	1C	57	6E	30	09	42	7B	D4	ED	A6	9F
B	1A	23	68	51	FE	C7	8C	B5	EB	D2	99	A0	0F	36	7D	44
C	69	50	1B	22	8D	B4	FF	C6	98	A1	EA	D3	7C	45	0E	37
D	B2	8B	C0	F9	56	6F	24	1D	43	7A	31	08	A7	9E	D5	EC
E	E6	DF	94	AD	02	3B	70	49	17	2E	65	5C	F3	CA	81	B8
F	3D	04	4F	76	D9	E0	AB	92	CC	F5	BE	87	28	11	5A	63

$\downarrow \boldsymbol{i}_2$

Очередной (по длине) байт-ОК может быть синтезирован на основе примитивного БЧХ $(31, 16, 3)$ – кода [1, 4], устраняющего ошибки кратности $t = 3$. Образующим полиномом $\boldsymbol{b}$ такого кода является полином 15-й степени

$$\boldsymbol{b} = 1000111110101111, \qquad (15)$$

которому отвечает МПС

$$\boldsymbol{P} = \begin{pmatrix} 1&0&0&0&1&1&1&1&1&0&1&0&1&1&1 \\ 1&1&0&0&1&0&0&0&0&1&1&1&1&0&0 \\ 0&1&1&0&0&1&0&0&0&0&1&1&1&1&0 \\ 0&0&1&1&0&0&1&0&0&0&0&1&1&1&1 \\ 1&0&0&1&0&1&1&0&1&0&1&0&0&0&0 \\ 0&1&0&0&1&0&1&1&0&1&0&1&0&0&0 \\ 0&0&1&0&0&1&0&1&1&0&1&0&1&0&0 \\ 0&0&0&1&0&0&1&0&1&1&0&1&0&1&0 \\ 0&0&0&0&1&0&0&1&0&1&1&0&1&0&1 \\ 1&0&0&0&1&0&1&1&0&0&0&1&1&0&1 \\ 1&1&0&0&1&0&1&0&0&0&1&0&0&0&1 \\ 1&1&1&0&1&0&1&0&1&0&1&1&1&1&1 \\ 1&1&1&1&1&0&1&0&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&1&0&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0&1&0&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1&0&1&0&1&1&1&1 \end{pmatrix}. \qquad (16)$$

Поскольку полином $\boldsymbol{b}$ не симметричный, то инвертированием правой части вектора (15) приходим к полиному $\boldsymbol{b}^*$, двойственный (дуальный) полиному $\boldsymbol{b}$, то есть

$$\boldsymbol{b}^* = 1111010111110001. \qquad (17)$$

Матрицу $\boldsymbol{P}^*$, сопряжённую МПС (16), можно составить или на основании её образующего полинома (17), или, что ещё проще, инвертированием строк и столбцов матрицы (16). Дополнив справа матрицы $\boldsymbol{P}$ и $\boldsymbol{P}^*$ столбцом контроля чётности, разместив в элементах столбцов цифру 0, если вес соответствующих строк матриц нечётный, и 1 – если вес чётный, приходим к расширенным МПС, порождающими сопряжённые байт-ориентированные $(32, 16, 3)$ – коды. Отметим, что значение элементов столбца контроля чётности определяется с учётом единичного элемента строки матрицы $\boldsymbol{E}$, являющейся компонентой образующей матрицы $\boldsymbol{G}$. Такие коды сохраняют свойство цикличности, но при условии, что из кодовых слов $\boldsymbol{C}$ удалены биты контроля чётности.

Как $(32, 16, 3)$ – байт-ОК, так и ранее полученные того же класса $(16, 8, 2)$ – коды обладают одинаковой относительной скоростью $R = k / n = 0.5$, но различаются средним значением $n_t = t / n$ числа устраняемых ошибок t на единицу длины n кода. Какой из этих двух байт-ОК является более предпочтительным, зависит от конкретных условий применения кодов.

На основании компьютерных расчётов могут быть получены и другие полиномы 16-й степени, приемлемые для построения укороченных циклических байт-ориентированных (n, k, t) – кодов. Отметим среди них, например, образующие полиномы (табл. 4), обеспечивающие в $(24, 8, t)$ – кодах коррекцию ошибок кратности $t = 3$.

Таблица 4

Множество образующих полиномов 16-й степени, порождающих укороченные $(24, 8, 3)$ – коды

№ п/п	Образующий полином	Вес	№ п/п	Образующий полином	Вес
1	10010101000100011	7	5	11010010101001011	9
2	11000100010101001		6	11100011010110001	
3	10001011000010011		7	11101010110100001	
4	10011000010001011		8	11110100010101001	

4. Основные свойства МПС

Обратим внимание на такие особенности матрицы проверочных символов, представленной в табл. 2. В нижней (первой) строке этой матрицы размещён порождающий её полином $\boldsymbol{b}_{10} = 100111001$, из которого удалена старшая (левая) единичка, тогда как в верхней строке находится тот же полином $\boldsymbol{b}_{10}$, но утративший младшую (правую) единичку. И, как следствие отмеченных особенностей граничных строк (нижней и верхней) матриц $\boldsymbol{P}$, если воспользоваться формальным правилом составления матрицы проверочных символов и вычислить её $(k+1)-$ю строчку (обозначим эту строчку как s_{k+1} и назовём *контрольной строкой* МПС), то приходим к такому результату

$$s_{k+1} = 0^{[r-1]}1 = \underbrace{00\ldots0}_{r-1}1, \qquad (18)$$

где $r = n - k$ – число проверочных разрядов кода.

Соотношение (18) является следствием таких эмпирически установленных положений:

Утверждение 1. *Двоичный полином $r-$й степени тогда и только тогда является образующим полиномом примитивного циклического $(n, k, t)-$кода, когда так называемая «контрольная» $(k+1)-$я строка s_{k+1}, являющаяся продолжением матрицы проверочных символов $\boldsymbol{P}$ кода и вычисляемая по правилам формирования строк этой матрицы, но не входящая в неё, определяется соотношением $s_{k+1} = 0^{[r-1]}1$ (необходимые условия), причём вес v каждой строки матрицы проверочных символов не меньше чем $2t$, а расстояние Хемминга $d_{i,j}$ между любыми парами строк (s_i, s_j) матрицы таково, что $d_{i,j} \geq 2t-1$ (достаточные условия).*

Утверждение 2. *Если нижняя s_1-я строка МПС содержит образующий полином $\boldsymbol{b}$ двоичного примитивного $(n, k, t)-$кода, из которого выброшена старшая (левая) единичка, то в верхней s_k-й строке размещается этот же полином $\boldsymbol{b}$, но из которого выброшена младшая (правая) единичка.*

Утверждение 3. *Двойственные матрицы проверочных символов, т.е. матрицы, порождаемые двойственными двоичными полиномами, взаимно связаны операторами инверсной перестановки строк и столбцов матриц.*

Легко убедиться в том, что Утверждения 1–3 соблюдаются так же и для всех МПС классических кодов, к числу которых отнесены коды Хемминга и циклические коды, включая коды БЧХ и коды Голея [1, 4].

В качестве примера, подтверждающего основные положения, сформулированные в п. 4, ниже приведены МПС $\boldsymbol{P}_2$ и $\boldsymbol{P}_2^*$ циклических $(15, 7, 2)-$кодов, порождаемых двойственными образующими полиномами $\boldsymbol{b}_2 = 100010111$ и $\boldsymbol{b}_2^* = 111010001$.

$$\boldsymbol{P}_2 = \begin{pmatrix} 1&0&0&0&1&0&1&1 \\ 1&1&0&0&1&1&1&0 \\ 0&1&1&0&0&1&1&1 \\ 1&0&1&1&1&0&0&0 \\ 0&1&0&1&1&1&0&0 \\ 0&0&1&0&1&1&1&0 \\ 0&0&0&1&0&1&1&1 \end{pmatrix}, \quad \boldsymbol{P}_2^* = \begin{pmatrix} 1&1&1&0&1&0&0&0 \\ 0&1&1&1&0&1&0&0 \\ 0&0&1&1&1&0&1&0 \\ 0&0&0&1&1&1&0&1 \\ 1&1&1&0&0&1&1&0 \\ 0&1&1&1&0&0&1&1 \\ 1&1&0&1&0&0&0&1 \end{pmatrix}. \qquad (19)$$

Из структурных форм МПС $\boldsymbol{P}_2$ и $\boldsymbol{P}_2^*$, представленных системой (19), следует, что двойственные матрицы не только взаимно связаны операторами инверсной перестановки строк и столбцов, но и каждой из них отвечает контрольная строка, определяемая соотношением (18).

5. Синдромное декодирование байт-ориентированных кодов

Декодирование сообщений, образованных байт-ОК, является более сложным процессом, чем кодирование. Поэтому ограничимся в дальнейшем пояснением лишь способа *синдромного декодирования* кодов, как более простого в программной и аппаратной реализации по сравнению с методами алгебраического декодирования [6].

Идея синдромного декодирования заключается в следующем. Пусть $\boldsymbol{c}(x)$ – исходное (неискажённое) $n-$разрядное систематическое кодовое слово (первичное сообщение), составленное конкатенацией $k-$ битного информационного слова

$\boldsymbol{i}(x)$ и совокупности $r=(n-k)$ проверочных разрядов, которые условимся называть *синдромом* $\boldsymbol{s}_{i,b}(x)$, *порождаемым образующим полиномом* $r-$й степени $\boldsymbol{b}(x)$ для выбранного слова $\boldsymbol{i}(x)$, т.е.

$$\boldsymbol{c}(x)=\boldsymbol{i}(x)\circ\boldsymbol{s}_{i,b}(x). \qquad (20)$$

Синдром $\boldsymbol{s}_{c,b}(x)$ неискажённого полинома кодового слова $\boldsymbol{c}(x)$, заданного соотношением (20), равен нулю

$$\boldsymbol{s}_{c,b}(x)=\boldsymbol{c}(x)\,(\mathrm{mod}\,\boldsymbol{b}(x))\;=\boldsymbol{0}. \qquad (21)$$

Обозначим через $\boldsymbol{c}_e(x)$ полином (вектор) циклического кода, поражённого аддитивной помехой $\boldsymbol{e}(x)$, называемого также *вектором ошибок* или *вектором коррекции*, полагая тем самым

$$\boldsymbol{c}_e(x)=\boldsymbol{c}(x)+\boldsymbol{e}(x). \qquad (22)$$

Вполне очевидно, что синдром $\boldsymbol{s}^{[e]}_{c,b}(x)$ искажённого полинома $\boldsymbol{c}_e(x)$, как следует из соотношений (21) и (22), однозначно определяется лишь вектором ошибок $\boldsymbol{e}(x)$, то есть

$$\boldsymbol{s}^{[e]}_{c,b}(x)=\boldsymbol{e}(x)\,(\mathrm{mod}\,\boldsymbol{b}(x)).$$

Поясним способ синдромного декодирования на примере укороченного байт-ориентированного $(16,8,2)-$кода, порождаемого полиномом восьмой степени $\boldsymbol{b}=100111001$, ранее обозначенного в (13) как $\boldsymbol{b}_{10}$. Для рассматриваемого варианта кода необходимо предварительно составить таблицу *синдромов* $\boldsymbol{S}_i$ *информационных символов* $\boldsymbol{I}_i$, $i=\overline{1,k}$, представляющие собой $i-$ю строку матрицы $\boldsymbol{E}$, кодового слова $\boldsymbol{C}$, которые для $(16,8,2)-$байт-ОК размещены в нижних восьми строках МПС табл. 2 и для удобства перенесены в табл. 5.

Таблица 5

Синдромы $\boldsymbol{S}_i$ информационных символов укороченного $(16,8,2)-$кода, порождаемого неприводимым полиномом $\boldsymbol{b}=100111001$

Номер i бита вектора $\boldsymbol{I}_i$		Разряды синдромов $\boldsymbol{S}_i$							
		8	7	6	5	4	3	2	1
8		0	1	0	0	1	1	1	0
7		0	0	1	0	0	1	1	1
6		1	0	0	0	1	1	1	1
5		1	1	0	1	1	0	1	1
4		1	1	1	1	0	0	0	1
3		1	1	1	0	0	1	0	0
2		0	1	1	1	0	0	1	0
1		0	0	1	1	1	0	0	1

Обратим внимание на то, что термин «синдром $\boldsymbol{S}_i$ информационных символов $\boldsymbol{I}_i$» не следует путать с введенными ранее терминами «синдром $\boldsymbol{S}$ кодового слова $\boldsymbol{C}$» и «синдром $\boldsymbol{S}_I$ информационной группы».

На основании данных табл. 5 составим табл. 6, в жирной рамке которой разместим *синдромы ошибок* (ещё один синдромный термин) $\boldsymbol{S}_{i;\,j}=\boldsymbol{S}_i\oplus\boldsymbol{S}_j$ одиночных $(i=0)$ и всевозможных двойных $(i>0)$ ошибок, представленных для компактности 16-ричными значениями

Таблица 6

Синдромы ошибок $\boldsymbol{S}_{i;\,j}$ укороченного $(16,8,2)-$кода

	1	2	3	4	5	6	7	8	$\rightarrow j$
0	39	72	E4	F1	DB	8F	27	4E	
1		4B	DD	C8	E2	B6	1E	77	
2			96	83	A9	FD	55	3C	
3				15	3F	6B	C3	AA	
4					2A	7E	D6	BF	
5						54	FC	95	
6							A8	C1	
7								69	

$\downarrow i$

Безотносительно к конкретным значениям образующих полиномов 16-й степени (например, тех, которые входят в табл. 4), порождающих укороченные $(24,8,3)-$ коды и устраняющих ошибки кратности $t=3$, полное множество синдромов ошибок $\boldsymbol{S}_{i;\,j}$ показано в виде затенённых элементов табл. 7.

Принципиальное отличие синдромов ошибок $\boldsymbol{S}_{i;\,j}$, представленных в табл. 6 и 7, состоит в следующем. Если синдромы $\boldsymbol{S}_{i;\,j}$ в табл. 6, соответствующие $(16,8,2)-$коду и исправляющие не более двух ошибок, состоят из двух групп, причём первая группа, для которой индекс $i=0$, объединяет синдромы одиночных ошибок, а вторая $(i>0)$ – синдромы ошибок кратности $t=2$, то синдромы $\boldsymbol{S}_{i;\,j}$ $(24,8,3)-$кодов (табл. 7) включают дополнительно ещё одну группу, отвечающих ошибкам кратности $t=3$. Индекс i третьей группы синдромов содержит две разделённые запятой цифры, совпадающие с номерами строк МПС над выбранным полиномом 16-й степени.

Проиллюстрируем (табл. 8), в качестве примера, способ вычисления синдрома $\boldsymbol{S}_{1,4;\,7}$ для $(24,8,3)-$кода, образуемого первым полиномом из табл. 4.

Таблица 7

Разметка синдромов ошибок $S_{i;j}$ укороченных $(24, 8, 3)$ – кодов над НП 16-й степени

	0	1	2	3	4	5	6	7		1,2	1,3	1,4	1,5	1,6	1,7	$\rightarrow i$
1	■															
2	■	■														
3	■	■	■							■						
4	■	■	■	■						■	■					
5	■	■	■	■	■					■	■	■				
6	■	■	■	■	■	■				■	■	■	■			
7	■	■	■	■	■	■	■			■	■	■	■	■		
8	■	■	■	■	■	■	■	■		■	■	■	■	■	■	
$\downarrow j$																
	2,3	2,4	2,5	2,6	2,7	3,4	3,5	3,6	3,7	4,5	4,6	4,7	5,6	5,7	6,7	$\rightarrow i$
4	■															
5	■	■				■										
6	■	■	■			■	■			■						
7	■	■	■	■		■	■	■		■	■		■			
8	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
$\downarrow j$																

Таблица 8

К вычислению синдрома ошибок $S_{1,4;7}$ $(24, 8, 3)$ – кода Над НП $\boldsymbol{b} = 10010101000100011$

Номер бита вектора $\boldsymbol{I}$		Разряды синдрома $S_{1,4;7}$															
		16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
8		0	1	1	0	0	1	0	1	0	1	0	1	1	0	0	1
7		1	0	1	0	0	1	1	1	1	0	1	1	1	1	0	1
6		1	1	0	0	0	1	1	0	1	1	0	0	1	1	1	1
5		1	1	1	1	0	1	1	0	0	1	1	1	0	1	1	0
4		0	1	1	1	1	0	1	1	0	0	1	1	1	0	1	1
3		1	0	1	0	1	0	0	0	1	0	0	0	1	1	0	0
2		0	1	0	1	0	1	0	0	0	1	0	0	0	1	1	0
1		0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	1
$S_{1,4;7} =$		1	1	1	1	0	1	1	0	1	0	1	0	0	1	0	1

Затенением в табл. 8 выделены строки, номера которых совпадают с индексами вычисляемого синдрома $S_{1,4;7}$, равного 0xF6A5.

Скорость выполнения процесса декодирования можно существенно повысить, если перейти от преобразований $(i,\ j) \Rightarrow S_{i;j}$ (табл. 5) к обратным преобразованиям $S_{i;j} \Rightarrow (i,\ j)$, представленным в табл. 9.

Покажем на числовых примерах способы устранения ошибок в $(16, 8, 2)$ – коде, приняв в качестве информационного слова вектор

$$\boldsymbol{I} = 10010110, \qquad (23)$$

воспользовавшись для расчётов синдрома S_I слова (23) таблицей 10.

Таблица 9

Взаимосвязь синдрома ошибок $S_{i;j}$ с номерами (i, j) искажённых информационных символов $(16, 8, 2)$ – кода над НП $\boldsymbol{b} = 100111001$

$S_{i;j}$	$i,\ j$	$S_{i;j}$	$i,\ j$	$S_{i;j}$	$i,\ j$	$S_{i;j}$	$i,\ j$	$S_{i;j}$	$i,\ j$	$S_{i;j}$	$i,\ j$
15	3, 4	3F	3, 5	6B	3, 6	95	5, 8	BF	4, 8	DD	1, 3
1E	1, 7	4B	1, 2	72	2	96	2,3	C1	6, 8	E2	1, 5
27	7	4E	8	77	1, 8	A8	6, 7	C3	3, 7	E4	3
2A	4, 5	54	5, 6	7E	4, 6	A9	2, 5	C8	1, 4	F1	4
39	1	55	2, 7	83	2, 4	AA	3, 8	D6	4, 7	FC	5, 7
3C	2, 8	69	7, 8	8F	6	B6	1, 6	DC	5	FD	2, 6

Таблица 10

К вычислению синдрома $\boldsymbol{S}_I$ информационного слова $\boldsymbol{I} = 11001101$

Номер бита вектора $\boldsymbol{C}$	Номер бита вектора $\boldsymbol{I}$	Вектор $\boldsymbol{I}$		Разряды синдрома $\boldsymbol{S}_I$							
				8	7	6	5	4	3	2	1
16	8	1		0	1	0	0	1	1	1	0
15	7	1		0	0	1	0	0	1	1	1
14	6	0		1	0	0	0	1	1	1	1
13	5	0		1	1	0	1	1	0	1	1
12	4	1		1	1	1	1	0	0	0	1
11	3	1		1	1	1	0	0	1	0	0
10	2	0		0	1	1	1	0	0	1	0
09	1	1		0	0	1	1	1	0	0	1
			$\boldsymbol{S}_I =$	0	0	0	0	0	0	1	1

Поразрядным сложением по модулю 2 выделенных элементов табл. 10 определяем синдром $\boldsymbol{S}_I$, отвечающий выбранному слову $\boldsymbol{I}$.

Представим вектор неискажённого кодового слова $\boldsymbol{C}$ (первичного сообщения) в форме

$$\boldsymbol{C} = \boldsymbol{I} \circ \boldsymbol{S}_I = 10010110'00000011, \qquad (24)$$

где апостроф ' отделяет информационную компоненту $\boldsymbol{I} = 10010110$ от соответствующего ему синдрома $\boldsymbol{S}_I = 00000011$ (будем обозначать синдром $\boldsymbol{S}_I$, там где это представляется более предпочтительным, также символом $\boldsymbol{R}$).

Возможны такие альтернативные варианты (V) сбоев в кодовом слове $\boldsymbol{C}^*$ на входе приёмного устройства:

– *VI* – помеха поражает не более двух и только информационные разряды $\boldsymbol{I}$ слова $\boldsymbol{C}^*$;

– *VR* – помеха поражает не более двух и только проверочные разряды $\boldsymbol{R}$ слова $\boldsymbol{C}^*$;

– *VIR* – помеха поражает по одному разряду компонент $\boldsymbol{I}$ и $\boldsymbol{R}$ слова $\boldsymbol{C}^*$.

В качестве классификационного критерия вариантов сбоя выберем приращение синдрома кодового слова $\boldsymbol{C}^*$, обозначив его как $\Delta\boldsymbol{S}$, определяемое поразрядной суммой по модулю 2 синдрома $\boldsymbol{S}_I$ информационной группы и группы проверочных символов $\boldsymbol{R}$ кодового слова $\boldsymbol{C}^*$, то есть

$$\Delta\boldsymbol{S} = \oplus \begin{matrix} \boldsymbol{S}_I \\ \boldsymbol{R} \end{matrix}. \qquad (25)$$

При этом если вес ν приращения $\Delta\boldsymbol{S}$ не превышает 2, то возможными результатами исходов являются:

$$\nu = \begin{cases} 0 & - \text{ отсутствие сбоя;} \\ 1-2 & - \text{ вариант сбоя } VR. \end{cases}$$

Если же окажется, что $\nu > 2$, то это будет означать, что имеют место такие варианты сбоя входного кодового слова $\boldsymbol{C}^*$: или *VI*, или *VIR*.

Рассмотрим ситуацию, в которой воздействию помехи оказываются подверженными два разряда систематического подблока $\boldsymbol{I}$ слова $\boldsymbol{C}^*$ (вариант *VI*) и пусть, для примера, таковыми будут 16-й и 12-й разряды входного слова $\boldsymbol{C}^*$. Полноформатный код помехи $\boldsymbol{e}$ представим в виде 16-битного бинарного вектора

$$\boldsymbol{e} = 0\mathbf{1}000\mathbf{1}00'00000000, \qquad (26)$$

который, будучи поразрядно просуммированным с кодовым словом (24), приводит к образованию искажённых как информационной компоненты $\boldsymbol{I}$, так и всего слова

$$\boldsymbol{C}^* = \boldsymbol{I}^* \circ \boldsymbol{R} = \mathbf{0}001\mathbf{1}110'00000011. \qquad (27)$$

Биты, поражённые помехой, выделены в (26) и (27) жирным шрифтом.

Несколько модифицировав табл. 10, произведём оценку синдрома $\boldsymbol{S}_I^*$ компоненты $\boldsymbol{I}^*$, выполненную в табл. 11.

По формуле (25) находим

$$\begin{matrix} \boldsymbol{S}_I^* = 10111100 \\ \oplus \quad \boldsymbol{R} = 00000011 \\ \hline \Delta\boldsymbol{S} = 10111111 = 0\text{xBF}. \end{matrix} \qquad (28)$$

Таблица 11

К вычислению синдрома $\boldsymbol{S}_I^*$

информационного слова $\boldsymbol{I}^* = \mathbf{0}0011\mathbf{1}110$

Номер бита вектора $\boldsymbol{C}^*$	Номер бита вектора $\boldsymbol{I}^*$	Вектор $\boldsymbol{I}^*$		Разряды синдрома $\boldsymbol{S}_I$							
				8	7	6	5	4	3	2	1
16	8	1		0	1	0	0	1	1	1	0
15	7	1		0	0	1	0	0	1	1	1
14	6	0		1	0	0	0	1	1	1	1
13	5	0		1	1	0	1	1	0	1	1
12	4	1		1	1	1	1	0	0	0	1
11	3	1		1	1	1	0	0	1	0	0
10	2	0		0	1	1	1	0	0	1	0
09	1	1		0	0	1	1	1	0	0	1
			$\boldsymbol{S}_I^* =$	1	0	1	1	1	1	0	0

Обращаясь к табл. 8, убеждаемся в том, что вектор ошибки (26) соотношением (28) вычислен правильно.

И, наконец, переходим к пояснению алгоритма устранения ошибок для варианта *VIR*, которым предполагается, что однократным искажениям подвергнуты как информационный вектор $\boldsymbol{I}$, так и вектор проверочных разрядов $\boldsymbol{R}$ кодового слова $\boldsymbol{C}^*$. Суть алгоритма сводится к такой последовательности операций:

1. Установим нулевым значение индекса i, т.е. пусть $i = 0$;

2. Увеличиваем на единицу индекс i, полагая $i := i + 1$;

3. Инвертируем $i-$й бит в информационном слове $\boldsymbol{I}$, обозначив это слово как $\boldsymbol{I}_i$;

4. Используя табл. 4, вычислим синдром $\boldsymbol{S}_i = \boldsymbol{S}(\boldsymbol{I}_i)$;

5. Определяем вес ν_i приращения синдрома $\Delta\boldsymbol{S}_i = \boldsymbol{S}_i \oplus \boldsymbol{R}$;

6. Если $\nu_i > 1$, то переходим к п. 2, иначе – к п. 7;

7. Конец вычислений; при этом последний вектор $\boldsymbol{I}_i$ оказывается совпадающим с неискажённым информационным словом $\boldsymbol{I}$.

Проиллюстрируем предлагаемый алгоритм численным примером. Итак, пусть неискажённое кодовое слово $\boldsymbol{C}$ задано выражением (24), а *VIR* – помеха – вектором

$$\boldsymbol{e} = 000001\mathbf{0}\,'00\mathbf{1}00000\,.$$

Тем самым имеем

$$\boldsymbol{C}^* = \boldsymbol{I}^* \circ \boldsymbol{R}^* = 10010\mathbf{1}\mathbf{0}0\,'00\mathbf{1}00011\,. \qquad (29)$$

Инвертируем, согласно п. 3 вышеприведенного алгоритма, младший (первый) бит компоненты $\boldsymbol{I}^*$ в (29) и вычислим с помощью табл. 4 синдром $\boldsymbol{S}_1 = 01001000$ информационного вектора $\boldsymbol{I}_1 = 10010101$. Поразрядно суммируя по модулю 2 синдром $\boldsymbol{S}_1$ с вектором $\boldsymbol{R}^*$ из (29), получим

$$\begin{array}{rl} & \boldsymbol{S}_1 = 01001000 \\ \oplus & \underline{\boldsymbol{R}^* = 00100011} \\ & \Delta\boldsymbol{S}_1 = 01101011\,. \end{array} \qquad (30)$$

Как следует из соотношения (30) вес ν_1 приращения синдрома $\Delta\boldsymbol{S}_1$ превышает единицу, поэтому инвертируем второй разряд компоненты $\boldsymbol{I}^*$ и производим вычисления аналогичные предыдущим, но уже над вектором $\boldsymbol{I}_2 = 10010110$, которому отвечает синдром $\boldsymbol{S}_2 = 00000011$ и приращение $\Delta\boldsymbol{S}_2 = 00100000$. Поскольку вес приращения $\Delta\boldsymbol{S}_2$ оказался равным единице, то это означает, что не искажённое информационное слово $\boldsymbol{I}$ совпадает с вектором $\boldsymbol{I}_2$. Полный алгоритм декодирования укороченного байт-ориентированного $(16, 8, 2)$ – кода представлен на рис. 1.

Кратко поясним алгоритм декодирования. Каждое кодовое слово $\boldsymbol{C} = \boldsymbol{I} \circ \boldsymbol{R}$, поступившее на вход приёмника, проходит цепочку тестовых испытаний, а именно:

1. Вычисляется (можно воспользоваться табл. 3) синдром $\boldsymbol{S}_I$ информационной группы $\boldsymbol{I}$ слова $\boldsymbol{C}$ и если окажется, что $\boldsymbol{S}_I$ совпадает с вектором проверочных разрядов $\boldsymbol{R}$, то это означает, что кодовое слово $\boldsymbol{C}$ не искажено. Приёмник извлекает компоненту $\boldsymbol{I}$ и переходит к обработке очередного входного слова. Иначе

2. Производится поразрядное сложение по модулю 2 синдрома $\boldsymbol{S}_I$ и вектора проверочных разрядов $\boldsymbol{R}$, образуя приращение $\Delta\boldsymbol{S}$. Если вес ν приращения синдрома $\Delta\boldsymbol{S}$ не превышает 2, то это означает, что помеха исказила только одно или два разряда $\boldsymbol{R}$ кодового слова $\boldsymbol{C}$, не затронув его информационную группу $\boldsymbol{I}$. Исправлять вектор $\boldsymbol{R}$ нет необходимости. Как и в п. 1, приёмник извлекает из кодового слова $\boldsymbol{C}$ компоненту $\boldsymbol{I}$ и переходит к обработке очередного входного слова. Иначе

3. Проверяется принадлежность приращения $\Delta\boldsymbol{S}$ одному из элементов $\boldsymbol{T}$ табл. 8 и если окажется, что значение $\Delta\boldsymbol{S}$ присутствует в таблице, то это будет означать, что помехой поражены один или два разряда информационного слова $\boldsymbol{I}$, номера которых представлены в табл. 8 в колонке (i, j). Производится корректировка (инвертирование) информационных разрядов i, j и приёмник переходит к обработке очередного входного слова $\boldsymbol{C}$. Иначе

4. Если условие $\Delta\boldsymbol{S} \in \boldsymbol{T}$ не выполняется, то это означает, что помехой поражены по одному разряду обеих систематических компонент $\boldsymbol{I}$ и $\boldsymbol{R}$ слова $\boldsymbol{C}$. На данном этапе тестирования выполняется последовательное инвертирование в цикле $i-$го бита, $i=\overline{1,8}$, информационного слова $\boldsymbol{I}$, которое обозначается как $\boldsymbol{I}_i$. На каждом $i-$м шаге цикла вычисляется синдром $\boldsymbol{S}_i$ компоненты $\boldsymbol{I}_i$ и определяется приращение синдрома $\Delta\boldsymbol{S}_i = \boldsymbol{S}_i \oplus \boldsymbol{R}$. Цикл заканчивается, как только достигается равенство $\nu(\boldsymbol{S}_i)=1$. Выполнение равенства означает, что инвертированием $i-$го бита информационного слова $\boldsymbol{I}$ устраняется единственная, присутствовавшая в этом слове ошибка. На этом завершается тестирование входного кодового слова $\boldsymbol{C}$.

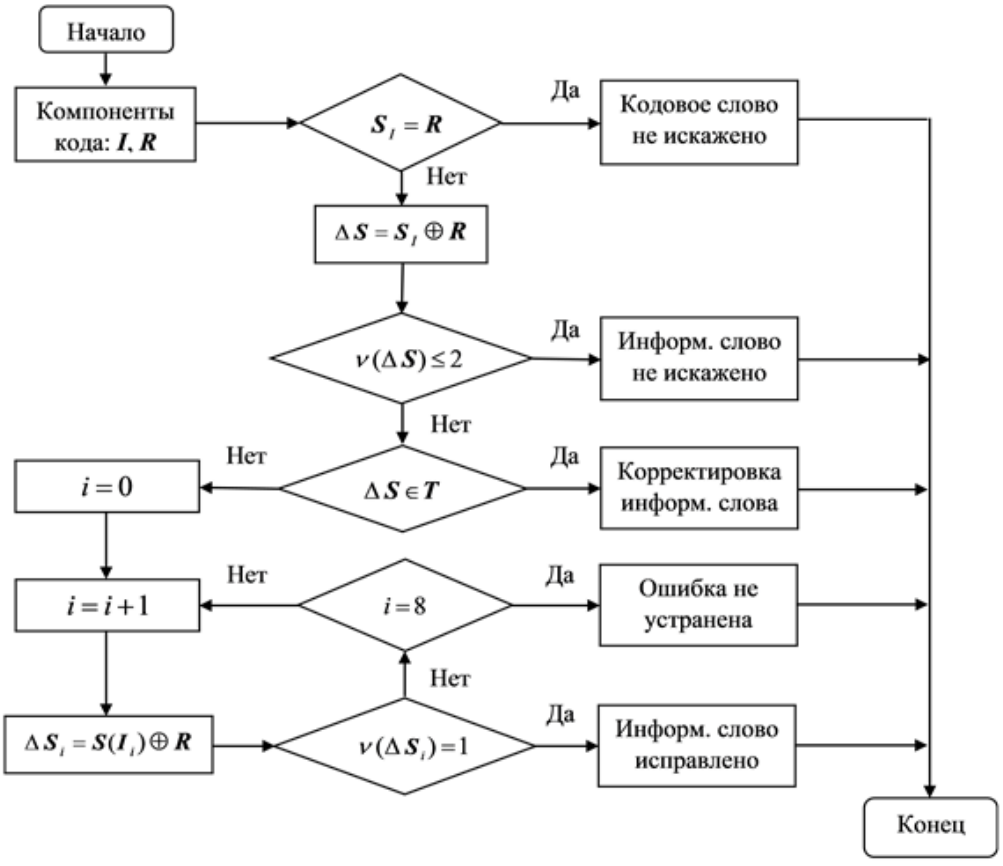

Рис. 1. Структурно-логическая схема алгоритма (16, 8, 2) – декодирования

Рассмотренный алгоритм ориентирован на обнаружение и устранение двукратных ошибок. Если в принятом кодовом слове число ошибок больше двух, такие входные слова не корректируются.

6. Обсуждение результатов

Разработанный систематический помехоустойчивый $(16, 8, 2)$ – код, порождаемый симметричным неприводимым полиномом восьмой степени $\boldsymbol{b} = 100111001$, является *уникальным* (т.е. единственным в своём роде) и по ряду критериев *оптимальным* в классе байт-ОК.

Введём обозначение $(\bar{n}, \bar{k}, t)$ для байт-ориентированных кодов, в котором $\bar{n}$ и $\bar{k}$ – число байт, содержащихся в кодовом $\boldsymbol{C}$ и информационном $\boldsymbol{I}$ словах соответственно, а t – максимальное число ошибок, корректируемых кодом. В частности, анализируемый $(16, 8, 2)$ – байт-ОК может быть записан как $(\bar{2}, \bar{1}, 2)$ – код.

Выберем в качестве критериев оптимальности байт-ОК такие числовые характеристики:

1) среднее число байтов n_t кодового слова $\boldsymbol{C}$, приходящихся на одну устраняемую $(\bar{n}, \bar{k}, t)$ – кодом ошибку. Имеем

$$n_t = \bar{n} / t;$$

2) среднее число байтов n_s таблицы синдромов ошибок $\boldsymbol{S}_{i;\,j}$ (подобной табл. 5 или 6), затрачиваемых на устранение одной ошибки в $(\bar{n}, \bar{k}, t)$ – кодах, то есть

$$n_s = V_S / t,$$

где V_S – объём (в байтах) таблиц синдромов $\boldsymbol{S}_{i;\,j}$.

Для произвольных параметров $\bar{n}$, $\bar{k}$ и t байт-ОК соответствующие им числовые характеристики легко могут быть получены аналитически. В частности, согласно данным табл. 5 и 6 приходим к таким оценкам объёмов V_S таблиц синдромов ошибок $\boldsymbol{S}_{i;\,j}$: для $(\bar{2}, \bar{1}, 2)$ – кода $V_{S1} = 36$, тогда как для $(\bar{3}, \bar{1}, 3)$ – кода $V_{S2} = 92$. Несмотря на то, что как в $(16, 8, 2)$ –, так и в $(24, 8, 3)$ – кодах параметр $n_t = 1$, первый из них является более предпочтительным, поскольку превосходит второй по критерию n_s.

На основании компьютерных расчётов удалось эмпирически подтвердить преимущество $(\bar{2}, \bar{1}, 2)$ – кода по сравнению с байт-ориентированными кодами с другими числовыми параметрами.

ЗАКЛЮЧЕНИЕ

1. Вместо традиционно используемых на этапах синтеза и анализа помехоустойчивых циклических кодов образующей $\boldsymbol{G}$ и проверочной $\boldsymbol{H}$ матриц рекомендуется применять матрицу проверочных разрядов $\boldsymbol{P}$, выполняющую на этапах кодирования/декодирования сообщений те же самые функции, что и матрицы $\boldsymbol{G}$ и $\boldsymbol{H}$, но занимающую в процессорах обработки информации гораздо меньший объем памяти.

2. Предложен помехоустойчивый $(16, 8, 2)$ – байт-ОК, порождаемый симметричным образующим неприводимым полиномов $\boldsymbol{b} = 100111001$, обеспечивающий б´ольшую скорость кодирования/декодирования сообщений по сравнению с классическими примитивными циклическими кодами, устраняющими в кодовых словах ошибки кратности, не превышающими 2.

3. Разработан оригинальный, экономный по требуемым для своей реализации аппаратно-программным ресурсам, алгоритм синдромного декодирования сообщений в байт-ориентированных кодах.

ЛИТЕРАТУРА

[1]. Р. Блейхут, *Теория и практика кодов, контролирующих ошибки: Пер. с англ.*, М.: Мир, 1986, 576 с.

[2]. А. Овсянников, А. Ямович, *Теория информации: Уч. Пособие*, Самар. гос. аэрокосм. ун-т., Самара, 2005, 131 с.

[3]. Р. Хемминг, *Теория кодирования и теория информации Пер. с англ*, М.: Мир, 1986, 576 с.

[4]. У. Питерсен, Э. Уэндон, *Коды, исправляющие ошибки: Пер. с англ.,* М.: Мир, 1976, 593 с.

[5]. Э. Берлекэмп, *Алгебраическая теория кодирования: Пер. с англ.*, М.: Мир, 1971, 478 с.

[6]. С. Федоренко, *Методы быстрого декодирования линейных блоковых кодов: Моногр.*, СПб: ГУАП, 2008, 199 с.

СИСТЕМАТИЧНІ БАЙТ-ОРІЄНТОВАНІ КОДИ

Порядок (число розрядів або довжина) класичних циклічних кодів як правило не є кратним цілому числу байтів, що призводить до зайвих витрат обчислювальних ресурсів при їх апаратно-програмної реалізації. З огляду на зазначене перехід до байт-орієнтованих кодів, в яких як довжина k інформаційних слів $\boldsymbol{I}$, так і число r перевірочних розрядів $\boldsymbol{R}$ кратні цілому числу байтів, здається більш доцільним до практичного застосування. Відмінна особливість пропонованого підходу до синтезу (кодуванню інформації) та аналізу кодів (декодуванню повідомлень) полягає у відмові від утворюючих $\boldsymbol{G}$ і перевірочних матриць $\boldsymbol{H}$, що зазвичай супроводжують систематичні циклічні коди, і їх заміну на єдину матрицю $\boldsymbol{P}$ перевірочних символів

(МПС), меншу за обсягом у порівнянні з матрицями $\boldsymbol{G}$ і $\boldsymbol{H}$. Основу формування МПС циклічних $(n, k, t)-$ кодів, де $n-$ довжина коду і $t-$ кратність помилок в кодових словах, що усуваються, складають (породжують) поліноми (одномірні вектори), які позначимо символом $\boldsymbol{b}$. Двійковий поліном $r-$го ступеня $\boldsymbol{b}$ тоді і тільки тоді є утворюючим поліномом примітивного циклічного $(n, k, t)-$ коду, коли так званий «контрольний» $(k+1)-$й рядок s_{k+1}, що є продовженням матриці перевірочних символів $\boldsymbol{P}$ коду і обчислюється за правилами формування рядків цієї матриці, але не входить до неї, визначається співвідношенням $s_{k+1} = 0^{[r-1]}1$ (необхідні умови), причому вага ν кожного рядка матриці перевірочних символів $\boldsymbol{P}$ не менш ніж $2t$, а відстань Хеммінга $d_{i,j}$ між будь-якими парами рядків (s_i, s_j) матриці $\boldsymbol{P}$ така, що $d_{i,j} \geq 2t-1$ (достатні умови). Двоїсті матриці перевірочних символів, тобто матриці, що породжуються двоїстими двійковими поліномами, взаємно пов'язані операторами інверсної перестановки рядків і стовпців матриць. Розроблено систематичний перешкодостійкий код, який породжується симетричним незвідним поліномом восьмого ступеню $\boldsymbol{b} = 100111001$, що є унікальним (тобто єдиним у своєму роді) і по ряду критеріїв оптимальним в класі байт-орієнтованих кодів. Наводиться розгорнута характеристика алгоритму синдромного декодування байт-орієнтованих кодів.

Ключові слова: байт-орієнтовані коди, утворюючі і перевірочні матриці, матриці перевірочних символів, синдромне декодування.

SYSTEMATIC BYTE-ORIENTED CODES

The order (number of bits or length) of classical cyclic codes is usually not a multiple of an integer number of bytes, which leads to unproductive expenditures of computing resources with their hardware-software implementation. For this reason, the transition to byte-oriented codes, in which both the length k of information words $\boldsymbol{I}$ and the number r of test bits are multiples $\boldsymbol{R}$ of an integer number of bytes, seems most appropriate for practical use. A distinctive feature of the proposed approach to synthesis (information coding) and code analysis (message decoding) is the rejection of generators $\boldsymbol{G}$ and verification matrices $\boldsymbol{H}$, usually accompanying systematic cyclic codes, and their replacement by a single matrix $\boldsymbol{P}$ of parity symbols (MAP), smaller in volume compared to using matrices $\boldsymbol{G}$ and $\boldsymbol{H}$. The basis for the formation of MPS cyclic $(n, k, t)-$ codes, where $n-$ the code length and $t-$ the multiplicity of eliminated errors in codewords, are generators (generating) polynomials (one-dimensional binary vectors), denoted by the symbol $\boldsymbol{b}$. A binary polynomial of $r-$ degree is a generating polynomial of a primitive cyclic $(n, k, t)-$ code if and only if the so-called "control" $(k+1)-$ string s_{k+1}, which is an extension of the matrix of code $\boldsymbol{P}$ parity symbols and computed according to the rules of forming the rows of this matrix, but not entering into it, is determined by the relation $s_{k+1} = 0^{[r-1]}1$ (necessary conditions), and the weight ν of each row of the parity matrix is not less than $2t$, and the Hamming $d_{i,j}$ distance between any pairs of rows (s_i, s_j) of the matrix $\boldsymbol{P}$ is such that $d_{i,j} \geq 2t-1$ (sufficient conditions). The operators of inverse permutation of rows and columns of the matrices mutually relate dual matrixes of parity symbols, i.e. the matrices generated by the dual binary polynomials. A systematic noise-proof $(16, 8, 2)-$ code is generated, generated by a symmetric irreducible polynomial of the eighth degree $\boldsymbol{b} = 100111001$, which is unique (that is, unique in its kind) and optimal in a class of byte-oriented codes for a number of criteria. A detailed characteristic of the algorithm for syndrome decoding of byte-oriented codes is given.

Keywords: byte-oriented codes, generators and verification matrixes of codes, matrix of parity symbols, syndrome decoding.

Белецкий Анатолий Яковлевич, доктор технических наук, профессор, заслуженный деятель науки и техники Украины, лауреат Гос. премии Украины в области науки и техники, профессор кафедры электроники Национального авиационного университета.
E-mail: abelnau@ukr.net.

Білецький Анатолій Якович, доктор технічних наук, професор, заслужений діяч науки і техніки України, лауреат Державної премії України в галузі науки і техніки, професор кафедри електроніки Національного авіаційного університету.

Beletsky Anatoly, Doctor of Science, Professor, Honored Scientist of Ukraine, Laureate of the State Prize of Ukraine in Science and Technology, Professor of Department Electronics of National Aviation University.

Конюший Дмитрий Вадимович, студент кафедры электроники Национального авиационного университета.
E-mail: dima.konushiy.95@mail.ru.

Конюший Дмитро Вадимович, студент кафедри електроніки Національного авіаційного університету.

Koniushyi Dmytro, Student of Department Electronics of National Aviation University.

Полторацкий Дмитрий Анатольевич, студент кафедры электроники Национального авиационного университета.
E-mail: damonpolt@gmail.com.

Полторацький Дмитро Анатолійович, студент кафедри електроніки Національного авіаційного університету.

Poltoratskyi Dmytro, Student of Department Electronics of National Aviation University.