

УДК 004.052.2

Е.С. Лепёшкина

## Построение и исследование проверочных матриц кодов, исправляющих одиночные и двойные смежные ошибки

Одним из следствий воздействия космической радиации на аппаратуру бортовых систем космических аппаратов являются сбои в памяти. В связи с уменьшением нормы проектирования цифровой электронной компонентной базы увеличивается количество множественных сбоев в памяти, в частности двойных. Для защиты от двойных сбоев в памяти используются коды с исправлением одиночных и двойных смежных ошибок. В работе представлен математический базис применения кодов данного вида. Предложено и доказано утверждение о структуре проверочной матрицы, необходимой для исправления одиночных и двойных смежных ошибок. Сформулирован общий алгоритм нахождения и исправления одиночных и двойных смежных ошибок в кодовом слове памяти. Доказано утверждение о минимальном количестве проверочных бит для информационного слова любой длины. Проведена общая оценка сложности операций кодирования и исправления ошибок в кодовом слове в зависимости от числа проверочных бит и структур проверочных матриц. Результаты работы могут найти применение при проектировании устройств обнаружения и коррекции ошибок в бортовой памяти космических аппаратов.

**Ключевые слова:** космическая электроника, множественные сбои, помехоустойчивые коды, коды с исправлением одиночных и двойных смежных ошибок.

**DOI:** 10.21293/1818-0442-2024-27-2-37-43

Бортовая электроника космических аппаратов эксплуатируется в сложных условиях внешней среды, потенциально вызывающих ее сбои и отказы. Среди множества негативных факторов космического пространства особое место занимает космическая радиация. Если от ряда негативных факторов, таких как вакуум, температура и др., найдены эффективные технические решения, то поиск решения проблем от действия космической радиации продолжается.

Одним из видов негативного воздействия радиации являются события SEU (Single-Event Upset, одиночный сбой), вызывающие изменения в записанной в памяти информации под воздействием высокоэнергетических частиц (тяжелых заряженных частиц, ТЗЧ) от Солнца, звезд и других дальних космических объектов [1, 2]. ТЗЧ воздействуют на ячейки хранения бит статической памяти и вызывают изменения записанных данных на противоположные значения: хранимые значения логической 1 заменяются на значения 0, и, наоборот, 0 инвертируются в 1. Если подобное инвертирование происходит в одном бите слова памяти, то происходит одиночная ошибка (однократное SEU), если в нескольких битах, то происходит множественная ошибка (мультибитные SEU) [3].

Современная схемотехника (схемы Error Detection and Correction, EDAC) бортовой электроники рассчитана на парирование одной ошибки на основе применения модифицированных кодов Хэмминга [4, 5]. До недавнего времени проблема множественных сбоев не стояла. Однако в ряде работ показано [6], что в последнее время наблюдается увеличение фиксируемых множественных ошибок. Причиной этого считают уменьшение типоразмеров проектирования цифровой электронной компонентной базы (менее 100 нм) [7]. При уменьшении раз-

меров КМОП-транзисторов, уменьшаются размеры создаваемых на их основе ячеек памяти. Поэтому ТЗЧ при попадании на кристалл микросхемы памяти будет воздействовать не на одну ячейку памяти, а сразу на несколько. В зависимости от энергии частицы и от «сечения» ее попадания на матричную структуру памяти в этом случае могут появляться не только одиночные, но и смежные двойные, тройные и даже ошибки большей кратности [8]. Смежной является ошибка, которая произошла в соседних битах одного слова памяти. Наиболее вероятна двойная смежная ошибка, нежели события большей кратности. Несмежные ошибки еще более маловероятны, так как это ошибки, происшедшие в одном слове памяти в разное время.

В последнее время проблеме обнаружения и исправления двойных ошибок посвящены работы многих авторов. Данные коды получили название SEC-DED-DAEC (Single Error Correction, Double Error Detection and Double Adjacent Error Correction), далее по тексту для простоты записи – SDAEC (Single and Double Adjacent Error Correction). Известны корректирующие коды по именам авторов Дутта (39,32) [9], Датта (42,32) [10], Нила (42,32) [11], Ревирриго (39,32) [12], Ча-Юна (39,32) [13], Хоюна-Йонгсурка (41,32) [14]. Разработаны предложения по применению подобных кодов в бортовой аппаратуре космических аппаратов [15, 16]. Вместе с тем не известны работы, которые бы поставили и решили бы в целом основные вопросы разработки и применения кодов SDAEC с учетом их возможной избыточности и получаемой при этом производительности. Данная работа направлена на решение этих задач.

### Основные определения

Процесс кодирования можно описать следующим образом: к блоку из  $k$  символов сообщения (информационных символов) по определенному

правилу добавляется  $m$  избыточных (проверочных) символов. Совокупность информационных и проверочных символов образует кодовое слово длиной  $n$ . Кодовые слова, соответствующие всем возможным сообщениям, образуют  $(n, k)$ -код. При этом если сумма любых двух кодовых слов также является кодовым словом, то код является линейным. Код, сообщения которого состоят из символов 0 и 1, называется двоичным кодом. В общем случае код задается в векторном пространстве над полем  $GF(q)$ .

Рассмотрим линейный блочный код вида  $\mathbf{c}=(n, k)$ , где  $\mathbf{c}=(c_1, c_2, \dots, c_n)$  – кодовый вектор длиной  $n$ ,  $k$  – длина исходного информационного вектора  $\mathbf{i}=(i_1, i_2, \dots, i_k)$ .

Кодовый вектор  $\mathbf{c}$  связан с информационным вектором  $\mathbf{i}$  следующим образом:  $\mathbf{c}=\mathbf{i} \cdot \mathbf{G}$ , где  $\mathbf{G}$  – порождающая матрица линейного кода  $\mathbf{c}$  размерностью  $k \times n$ . То есть любое кодовое слово линейного блочного кода  $(k, n)$  можно получить умножением информационного вектора  $\mathbf{i}$  на порождающую матрицу  $\mathbf{G}$  [17].

Для проверки принятого кодового слова на отсутствие ошибок необходимо умножить его на проверочную матрицу  $\mathbf{H}$ , так как проверочная матрица  $\mathbf{H}$  линейного кода  $\mathbf{c}$  такова, что для любого кодового слова и только для него выполняется равенство вида  $\mathbf{c}\mathbf{H}^T=0$ , где  $\mathbf{H}^T$  – транспонированная матрица  $\mathbf{H}$ . Эта процедура называется вычислением синдрома  $\mathbf{s}$ . Суммируются столбцы проверочной матрицы  $\mathbf{H}$ , соответствующие единичным координатам принятого сообщения. Если принятое сообщение является кодовым словом, то равенство выполняется и синдром  $\mathbf{s}$  равен 0. Если равенство не выполняется, то это сигнализирует о том, что произошла ошибка, определяемая вектором ошибки  $\mathbf{s}$ . В случае кода, исправляющего одиночные ошибки, порядковый номер строки транспонированной проверочной матрицы  $\mathbf{H}^T$ , совпадающий со значением синдрома  $s$ , является номером искаженного ошибкой разряда кодового слова.

Порождающая  $\mathbf{G}$  и проверочная  $\mathbf{H}$  матрицы связаны между собой:  $\mathbf{G} \cdot \mathbf{H}^T=0$  [18].

Таким образом, линейный код  $\mathbf{c}=(n, k)$  может быть определен одной из двух матриц: либо порождающей  $\mathbf{G}$ , либо проверочной  $\mathbf{H}$ . Зная проверочную матрицу, можно найти порождающую матрицу. Причем справедливо и обратное.

#### Требования к проверочной матрице

В следующей теореме формулируются достаточные условия, при наложении которых на проверочную матрицу получится код, исправляющий одиночные и двойные смежные ошибки (SDAEC).

*Теорема 1.* Если все столбцы и суммы (по модулю 2) соседних столбцов проверочной матрицы  $\mathbf{H}$  кода различны, то соответствующий  $\mathbf{H}$  код исправляет одиночные и двойные смежные ошибки.

*Доказательство.* Пусть проверочная матрица  $\mathbf{H}$  порядка  $m \times n$  ( $m$  – количество строк,  $n$  – количество столбцов и длина кода) с элементами  $\mathbf{h}_{ij}$ .

Через  $\mathbf{h}^j$  будем обозначать  $j$ -й столбец матрицы  $\mathbf{H}$ :

$$\mathbf{h}^j = \begin{pmatrix} h_{1j} \\ \vdots \\ h_{mj} \end{pmatrix}.$$

Если  $\mathbf{c}=(c_1, \dots, c_n)$ ,  $c_1, \dots, c_n \in Z=\{0, 1\}$  – произвольный кодовый вектор, то

$$\mathbf{H}\mathbf{c}^T = c_1\mathbf{h}^1 \oplus \dots \oplus c_n\mathbf{h}^n = \bar{\mathbf{0}}, \quad (1)$$

где  $\bar{\mathbf{0}}$  – нулевой вектор-столбец размерности  $m$ . Отсюда следует, что, во-первых, если вектор  $\tilde{\mathbf{c}}=(\tilde{c}_1, \dots, \tilde{c}_n)$  отличается от кодового вектора  $\mathbf{c}$  в одной из координат, например  $i$ -й, то  $\tilde{c}_i = c_i \oplus 1$  и поэтому ввиду (1)

$$\begin{aligned} \mathbf{H}\tilde{\mathbf{c}}^T &= \tilde{c}_1\mathbf{h}^1 \oplus \dots \oplus \tilde{c}_n\mathbf{h}^n = \tilde{c}_1\mathbf{h}^1 \oplus \dots \oplus (c_i \oplus 1) \times \\ &\times \mathbf{h}^i \oplus \dots \oplus c_n\mathbf{h}^n = c_1\mathbf{h}^1 \oplus \dots \oplus c_n\mathbf{h}^n \oplus \mathbf{h}^i = \mathbf{h}^i. \end{aligned} \quad (2)$$

Во-вторых, если вектор  $\tilde{\mathbf{c}}$  отличается от кодового вектора  $\mathbf{c}$  в двух смежных координатах, например,  $i$  и  $i+1$ , то  $\tilde{c}_i = c_i \oplus 1$ ,  $\tilde{c}_{i+1} = c_{i+1} \oplus 1$  и, значит,

$$\begin{aligned} \mathbf{H}\tilde{\mathbf{c}}^T &= \tilde{c}_1\mathbf{h}^1 \oplus \dots \oplus \tilde{c}_n\mathbf{h}^n = c_1\mathbf{h}^1 \oplus \dots \oplus (c_i \oplus 1) \times \\ &\times \mathbf{h}^i \oplus (c_{i+1} \oplus 1) \cdot \mathbf{h}^{i+1} \oplus \dots \oplus c_n\mathbf{h}^n = c_1\mathbf{h}^1 \oplus \dots \oplus \\ &\oplus c_n\mathbf{h}^n \oplus \mathbf{h}^i \oplus \mathbf{h}^{i+1} = \mathbf{h}^i \oplus \mathbf{h}^{i+1}. \end{aligned} \quad (3)$$

По условию все столбцы и суммы соседних столбцов матрицы  $\mathbf{H}$  различны, следовательно, все элементы множества

$$\{h^1, \dots, h^n, h^1 \oplus h^2, h^2 \oplus h^3, \dots, h^{n-1} \oplus h^n\} \quad (4)$$

различны, и возможно идентифицировать по полученной сумме (2) или (3), какая ошибка (одиночная или двойная смежная) произошла, и определить ее местоположение. Теорема 1 доказана.

#### Алгоритм нахождения и исправления одиночных и двойных ошибок

Используя доказанную теорему, можно сформулировать общий алгоритм идентификации (определения) и исправления одиночных и двойных смежных ошибок.

Шаг 1. По принятому вектору  $\tilde{\mathbf{c}}$  вычислить вектор  $\mathbf{h}=\mathbf{H}\tilde{\mathbf{c}}^T$ . Если  $\mathbf{h}=\bar{\mathbf{0}}^T$ , то вектор  $\tilde{\mathbf{c}}$  – кодовый, завершить работу. Иначе, шаг 2.

Шаг 2. Сравнить вектор  $\mathbf{h}$  с векторами  $\mathbf{h}^1, \dots, \mathbf{h}^n$ . Если  $\mathbf{h}=\mathbf{h}^i$  для некоторого  $i$ , то изменить  $i$ -й бит вектора  $\tilde{\mathbf{c}}$  на противоположный и завершить работу. Иначе, шаг 3.

Шаг 3. Сравнить вектор  $\mathbf{h}$  с векторами  $\mathbf{h}^1 \oplus \mathbf{h}^2, \dots, \mathbf{h}^{n-1} \oplus \mathbf{h}^n$ . Если  $\mathbf{h}=\mathbf{h}^i \oplus \mathbf{h}^{i+1}$  для некоторого  $i$ , то изменить  $i$ -й и  $(i+1)$ -й бит вектора  $\tilde{\mathbf{c}}$

на противоположный и завершить работу. Иначе, сигнализировать об ошибках и завершить работу.

Проанализирована работа алгоритма. Если вектор  $\tilde{c}$  отличается от вектора  $c$  в одном или двух соседних координатах, что означает наличие однократной или двукратной смежной ошибки, то алгоритм работает корректно. Алгоритм работает корректно и в том случае, если ошибок не произошло. В остальных случаях алгоритм либо не изменит вектор  $\tilde{c}$ , либо будет пытаться исправлять ошибки на шагах 2 или 3. Последнее будет возможно, если некоторая комбинация ошибок будет давать синдром, равный одному из векторов множества (4).

**Оценка числа проверочных бит SDAEC кода**

В следующей теореме получена оценка снизу на число проверочных бит SDAEC кода.

*Теорема 2.* Если  $\mathbf{H}$  – проверочная матрица кода, исправляющего одиночные и двойные смежные ошибки порядка  $m \times n$ , то справедливы неравенства:

$$m \geq \lceil \log_2 n \rceil + 2, \tag{5}$$

если число  $n$  не является степенью числа 2, и

$$m \geq \lceil \log_2 n \rceil + 1 = k + 1, \tag{6}$$

когда  $n = 2^k$ .

*Доказательство.* По теореме 1 столбцы матрицы  $\mathbf{H}$ , коих  $n$ , и суммы двух соседних ее столбцов, коих  $n-1$ , должны быть различны и, конечно, отличны от столбца из нулей. Таким образом, чтобы построить матрицу  $\mathbf{H}$ , необходимо иметь в запасе не менее  $2n-1$  различных ненулевых столбцов размерности  $m$ .

С другой стороны, из 0 и 1 можно построить  $2^m-1$  различных ненулевых столбцов размерности  $m$ . Значит, должно выполняться неравенство  $2^m-1 \geq 2n-1$  или равносильное ему  $2^{m-1} \geq n$ .

Логарифмируя его далее по основанию 2, получим  $m-1 \geq \log_2 n$ .

Наконец, беря целую часть от правой и левой частей последнего неравенства и учитывая, что число  $m-1$  целое, окончательно получаем  $m-1 \geq \lceil \log_2 n \rceil + 1$  или  $m \geq \lceil \log_2 n \rceil + 2$ , если  $n$  не является степенью числа 2, и  $m-1 \geq \log_2 2^k = k$  или  $m \geq k+1$ , когда  $n = 2^k$ . Теорема 2 доказана.

Далее составлена таблица, первая строка которой содержит значения  $n$ , а вторая – значение  $\lceil \log_2 n \rceil + 2$ , если  $n$  не степень двойки, и значение  $k+1$ , если  $n = 2^k$ .

Из табл. 1 следует, что:

- 1) код, имеющий ровно 8 информационных бит, имеет длину не меньшую чем 13;
- 2) код, имеющий ровно 16 информационных бит, имеет длину не меньшую чем 22;
- 3) код, имеющий ровно 32 информационных бита, имеет длину не меньшую чем 39;
- 4) код, имеющий ровно 64 информационных бита, имеет длину не меньшую чем 72.

В общем случае справедливо следствие 1.

*Следствие 1.* Код, исправляющий одиночные и двойные смежные ошибки и имеющий  $2^k$  информационных бит, имеет длину не меньшую чем  $2^k + k + 2$ .

Вопрос о существовании кодов с указанными в следствии 1 числом информационных бит и длиной решается положительно, если возможно привести проверочную матрицу  $\mathbf{H}$  порядка  $(k+2) \times (2^k + k + 2)$ , удовлетворяющую условиям теоремы 1, и отрицательно, если возможно доказать, что такой матрицы не существует.

**Оценка сложности кодирования и коррекции ошибок**

Данные оценки проведены на примерах полученных матриц  $\mathbf{H}$ .

Ниже приводятся требуемые матрицы  $\mathbf{H}$  для  $k=3, 4, 5, 6$ , и тем самым устанавливается существование SDAEC-кодов. Матрицы приведены в систематическом виде.

*Случай  $k=3$ .* Тогда  $n = 2^3 + 3 + 2 = 13$ .

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{7}$$

В кодовом векторе  $c = (c_1, \dots, c_{13})$  биты  $c_1, \dots, c_8$  являются информационными, а  $c_9, \dots, c_{13}$  – проверочными. Процесс кодирования – вычисление проверочных бит по информационным – осуществляется по формулам:

$$c_9 = c_1 \oplus c_3 \oplus c_5,$$

$$c_{10} = c_2 \oplus c_4 \oplus c_7,$$

$$c_{11} = c_2 \oplus c_5 \oplus c_6 \oplus c_8,$$

$$c_{12} = c_2 \oplus c_3 \oplus c_6 \oplus c_7,$$

$$c_{13} = c_1 \oplus c_4 \oplus c_6 \oplus c_8.$$

Количество операций сложения равно 13.

Таблица 1

«Зависимость количества проверочных бит от количества информационных бит»

$n$	4	5	...	8	9	...	16	17	...	32	33	...	64	65	...	128
	3	4	...	4	5	...	5	6	...	6	7	...	7	8	...	8

Чтобы быстро проверить, что матрица  $\mathbf{H}$  удовлетворяет условиям теоремы 1, нужно сопоставить каждому вектору  $(r_1, \dots, r_5)$  целое число  $z$  по правилу  $z = r_1 2^0 + r_2 2^1 + r_3 2^2 + r_4 2^3 + r_5 2^4$ .

Вычисления показывают, что столбцам матрицы  $\mathbf{H}$  соответствует набор чисел  $\{17, 14, 9, 18, 5, 28, 10, 20, 1, 2, 4, 8, 16\}$ .

Суммам соседних столбцов – набор чисел:  $\{31, 7, 27, 23, 25, 22, 30, 21, 3, 6, 12, 24\}$ , и среди них нет одинаковых.

Рассмотрен вопрос о минимальности числа операций сложения. Количество операций сложения,

необходимых при кодировании, равно сумме всех единиц матрицы  $\mathbf{H}$  минус удвоенное количество строк:

$$S_k = \text{Sum}(\mathbf{H}) - 2 \cdot S(\mathbf{H}), \quad (8)$$

где  $S_k$  – количество операций сложения,  $\text{Sum}(\mathbf{H})$  – сумма всех элементов матрицы  $\mathbf{H}$ ,  $S(\mathbf{H})$  – количество строк в матрице  $\mathbf{H}$ .

Очевидно, что при фиксированном количестве строк количество операций сложения можно уменьшать, используя при построении матрицы  $\mathbf{H}$  столбцы с минимальным количеством единиц.

*Лемма 1.* Функция (8) достигает минимума на матрице (7).

Доказательство. При построении матрицы  $\mathbf{H}$  использовались все  $\binom{5}{1} = 5$  столбцов с одной единицей, все  $\binom{5}{2} = 10$  столбцов с двумя единицами (6

как столбцы и 4 как суммы), два оставшихся столбца имеют по три единицы. Лемма 1 доказана.

Возможно ли уменьшить число операций и построить более структурированную матрицу увеличивая количество строк, а значит, увеличивая число проверочных бит?

Пусть количество строк увеличится до шести. Тогда в распоряжении будет  $\binom{6}{1} = 6$  столбцов по одной единице и  $\binom{6}{2} = 15$  столбцов по две единице.

Следующий пример показывает, что матрицу, удовлетворяющую теореме 1 порядка  $6 \times 14$ , построить можно:

$$\mathbf{H} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (9)$$

Количество операций сложения при кодировании равно  $S_k = \text{Sum}(\mathbf{H}) - 2 \cdot S(\mathbf{H}) = 22 - 2 \cdot 6 = 10$ . Количество операций сложения, необходимых для вычисления вектора синдромов, равно  $S_c = \text{Sum}(\mathbf{H}) - S(\mathbf{H}) = 22 - 6 = 16$ .

Для матрицы (7) имеется  $S_c = 23 - 5 = 18$ , а для матрицы (9)  $S_c = 22 - 6 = 16$ .

Последующее увеличение числа проверочных бит будет давать увеличение и для  $S_k$ , и для  $S_c$ , поскольку количество столбцов с двумя единицами всегда будет оставаться постоянным и равным 8, а количество столбцов с одной единицей будет расти. Таким образом, по количеству операций сложения матрица (9) предпочтительней. Также матрица (9) имеет структуру, она состоит из 4 блоков, каждый из которых получается циклическим сдвигом первого столбца.

Ниже приведены два набора чисел, соответствующих столбцам и суммам соседних столбцов матрицы (9), по правилу

$$z = 1 \cdot r_1 + 2 \cdot r_2 + 4 \cdot r_3 + 8 \cdot r_4 + 16 \cdot r_5 + 32 \cdot r_6,$$

которые показывают, что матрица (9) удовлетворяет теореме 1:  $\{34, 5, 10, 20, 40, 9, 18, 36, 1, 2, 4, 8, 16, 32\}$  и  $\{39, 15, 30, 60, 33, 27, 54, 37, 3, 6, 12, 24, 48\}$ .

Общая ситуация. Всего существует

$$\binom{m}{2} = \frac{m(m-1)}{2}$$

векторов с двумя единицами. Векторы с двумя единицами, стоящими рядом, использовать нельзя, поскольку они получаются как сумма двух соседних столбцов единичной матрицы. Всего таких векторов  $m-1$ . Таким образом, для построения проверочной матрицы в распоряжении имеется  $m$  векторов с одной единицей и  $\frac{m(m-1)}{2} - m - 1 = \frac{(m-1)(m-2)}{2}$  векторов с двумя единицами. Если все они исчерпаются, то далее нужно будет использовать вектора с 3, 4 и т.д. единицами. Необходимо вычислить число операций, которое будет затрачиваться на кодирование и вычисление синдрома для 16, 32 и 64 информационных бит, с учетом сказанного выше.

*Случай  $k = 4$ , 16 информационных бит.*

Расчеты для  $m = 6$ . Число векторов с одной единицей равно  $m = 6$ . Число векторов с двумя единицами за вычетом  $m-1$  равно  $\frac{(m-1)(m-2)}{2} = 10$ .

Чтобы составить матрицу из 22 столбцов, необходимо еще 6 векторов. Будут использоваться векторы из трех единиц, их  $\frac{m(m-1)(m-2)}{3!} = 20$ .

$$\text{Sum}(\mathbf{H}) = 6 \cdot 1 + 10 \cdot 2 + 6 \cdot 3 = 44,$$

$$S_k = \text{Sum}(\mathbf{H}) - 2 \cdot S(\mathbf{H}) = 44 - 2 \cdot 6 = 32,$$

$$S_c = \text{Sum}(\mathbf{H}) - S(\mathbf{H}) = 44 - 6 = 38.$$

Расчеты для  $m = 7$ . Число векторов с одной единицей равно  $m = 7$ . Число векторов с двумя единицами за вычетом  $m-1$  равно  $\frac{(m-1)(m-2)}{2} = \frac{6 \cdot 5}{2} = 15$ .

Чтобы построить матрицу из 23 столбцов, необходимо еще один вектор с тремя единицами, в этом случае:  $\text{Sum}(\mathbf{H}) = 7 \cdot 1 + 15 \cdot 2 + 1 \cdot 3 = 40$ , откуда  $S_k = 40 - 2 \cdot 7 = 26$  и  $S_c = 40 - 7 = 33$ .

Расчеты для  $m = 8$ . Имеем  $\frac{(m-1)(m-2)}{2} = 28$ .

Поэтому

$$\text{Sum}(\mathbf{H}) = 8 \cdot 1 + 16 \cdot 2 = 40;$$

$$S_k = 40 - 2 \cdot 8 = 24;$$

$$S_c = 40 - 8 = 32.$$

Результаты остальных расчетов приведены в табл. 2. Матрицы с параметрами  $m, n$  из табл. 2, удовлетворяющие условиям теоремы 1, существуют. Для первых двух случаев  $m = 6$  и  $m = 7$  соответствующие таблицы приведены ниже (табл. 3 и 4).

Таблица 2

Параметры для $k = 4$								
$n$	22	23	24	25	26	...	48	
$m$	6	7	8	9	10	...	32	
$S_k$	32	26	24	23	22	...	0	
$S_c$	38	33	32	32	32	...	32	

Случай  $k = 5$ . Число информационных бит равно 32 и 64. Результаты вычислений представлены в табл. 5.

Случай  $k = 6$ . Число информационных бит равно 64. Результаты вычислений представлены в табл. 6.

Таблица 3

Проверочная матрица для  $m = 6$

0	1	1	0	1	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0	0	0
0	0	1	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0
1	0	0	0	1	1	0	0	0	0	1	1	0	0	1	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1	0
1	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1	0	0	0	0	0	1

Столбцам и суммам столбцов соответствуют следующие наборы различных чисел:  
 {44, 17, 7, 34, 9, 28, 36, 18, 49, 5, 10, 44, 33, 20, 26, 40, 1, 2, 4, 8, 16, 32};  
 {61, 22, 37, 43, 21, 56, 54, 35, 52, 15, 38, 13, 53, 14, 50, 41, 3, 6, 12, 23, 48}.

Таблица 4

Проверочная матрица для  $m = 7$

1	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0	1	0	0
0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0
0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1
1	0	0	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1

Столбцам и суммам столбцов соответствуют следующие наборы различных чисел:  
 {65, 22, 33, 66, 17, 68, 34, 9, 18, 36, 72, 5, 10, 20, 40, 80, 1, 2, 4, 8, 16, 32, 64};  
 {87, 55, 99, 83, 85, 102, 43, 27, 52, 108, 77, 15, 30, 60, 120, 81, 3, 6, 12, 24, 48, 96}.

Таблица 5

Параметры для  $k = 5$

$n$	39	40	41	42	43	45	...	95	96
$m$	7	8	9	10	11	12	...	63	64
$S_k$	74	67	59	54	53	52	...	1	0
$S_c$	81	75	68	64	64	64	...	64	64

Таблица 6

Параметры для  $k = 6$

$n$	72	73	74	75	76	...	191	192
$m$	8	9	10	11	12	...	127	128
$S_k$	163	157	146	136	125	...	1	0
$S_c$	171	166	156	147	137	...	128	128

Матрицы, удовлетворяющие условиям теоремы 1, для всех указанных в таблицах значений параметров  $m$  и  $n$  существуют.

**О декодировании построенных кодов.** Далее отмечены замечания, полезные при декодировании исследуемых кодов SDAEC. Пусть проверочная матрица  $\mathbf{H}$  кода, исправляющего одиночные и двойные смежные ошибки, построена только из столбцов с одной и двумя единицами и у любых двух соседних столбцов нет единиц на одинаковых позициях (см., например, матрицу (9)). Тогда вес Хемминга (количество единиц) столбцов матрицы  $\mathbf{H}$  и сумм по модулю 2 соседних столбцов равен 1, 2, 3 или 4. В последних двух случаях имеется двойная смежная ошибка, в первом случае – одиночная ошибка. Если вес Хемминга равен 2, то двойная смежная ошибка возникает, когда единицы стоят на соседних позициях, иначе, ошибка однократная. Таким образом,

ошибку (ошибки) можно определить по весу Хемминга вектора синдрома и по позициям единиц. Также можно определить функцию от числа единиц вектора синдрома и их позиций, значением которой будет позиция (позиции) ошибочных бит. В случае когда при построении матрицы  $\mathbf{H}$  используются столбцы с тремя единицами, алгоритм определения позиций ошибочных бит, отличный от простого перебора, тоже существует, но имеет более сложную структуру.

**Заключение.** В работе представлены математические основы, необходимые для использования кодов класса SDAEC. Сформулировано и доказано утверждение о структуре проверочной матрицы  $\mathbf{H}$ , необходимой для исправления одиночных и двойных смежных ошибок в кодовом слове. Для этого все столбцы и суммы (по модулю 2) соседних столбцов проверочной матрицы  $\mathbf{H}$  кода должны быть различны.

Сформулирован общий алгоритм нахождения и исправления одиночных и двойных смежных ошибок в кодовом слове памяти. Алгоритм состоит из трех шагов: первый шаг определяет отсутствие ошибок; второй шаг при наличии исправляет одиночную ошибку; третий шаг при наличии исправляет двойную смежную ошибку; в случае ошибки большей кратности алгоритм сигнализирует об этой ситуации и завершает работу.

Доказано утверждение о минимальном количестве проверочных бит в кодовом слове любой длины. Так, когда число информационных бит есть сте-

пени двойки (4, 8, 16, 32, 64 и т.д.), минимальное количество проверочных бит должно быть на 2 больше показателя степени информационного слова (соответственно, 4, 5, 6, 7, 8 и т.д.).

Проведена общая оценка сложности операций кодирования и исправления ошибок в кодовом слове. Показано, что при увеличении количества проверочных бит и, соответственно, увеличении проверочной матрицы SDAEC-кода уменьшается количество операций сложения по модулю 2, необходимых для проведения кодирования информационных слов памяти и последующего исправления кодового слова при обнаружении одиночных и двойных смежных ошибок. Последнее обстоятельство позволяет, при необходимости, находить компромиссное решение между избыточностью и производительностью реализуемого SDAEC-кода, используемого в конкретном случае обеспечения целостности информации в бортовой памяти.

#### Литература

- Осипенко П.Н. Одиночные сбои – вызов современных микропроцессоров // *Электронные компоненты*. – 2009. – № 7. – С. 12–15.
- Максименко С.Л. Анализ проблемы построения радиационно стойких информационно-управляющих систем / С.Л. Максименко, В.Ф. Мелехин, А.С. Филиппов // *Информационно-управляющие системы*. – 2012. – № 2 (57). – С. 18–25.
- Механизмы многократных сбоев в микросхемах памяти / А.И. Чумаков, А.В. Сокоян, А.Б. Боруздина, А.А. Смолин, А.А. Печенкин // *Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС)*. – 2016. – № 4. – С. 145–152.
- Hamming R.W. Error detecting and error correcting codes // *Bell System technical journal*. – 1950. – № 29 (2). – P. 147–160.
- Hsiao M.Y. A class of optimal minimum odd-weight column SEC-DED codes // *IBM J. Res. Develop.* – 1970. – № 14 (4). – P. 395–401.
- Смутьский А.В. К вопросу обеспечения устойчивости бортовой аппаратуры перспективных космических аппаратов к множественным сбоям от действия отдельных ядерных частиц космического пространства / А.В. Смутьский, С.И. Алексеев, Ю.Е. Кудрявцев // *Вестник НПО им. С.А. Лавочкина*. – 2014. – № 4 (25). – С. 97–102.
- Investigation of increased multi-bit failure rate due to neutron induced SEU in advanced embedded SRAMS / G. Georgakos, P. Huber, M. Ostermayr, E. Amirante, F. Ruckerbauer // *2007 IEEE Symposium on VLSI Circuits*. – 2007. – P. 80–81.
- Краснюк А.А. Особенности применения методов помехоустойчивого кодирования в суб-100-нм микросхемах памяти для космических систем / А.А. Краснюк, К.А. Петров. – М.: Национальный исследовательский ядерный университет «МИФИ», ФГБУН НИИ системных исследований РАН. – 2012. – 4 с.
- Dutta A. Multiple Bit Upset Tolerant Memory Using a Selective Cycle Avoidance Based SEC-DED-DAEC Code / A. Dutta, N.A. Touba // *25th IEEE VLSI Test Symposium (VTS'07)*. – URL: <https://ieeexplore.ieee.org/document/4209937>, свободный (дата обращения: 28.02.2024).
- Datta R. Exploiting unused spare columns to improve memory ECC / R. Datta, N.A. Touba // *27th IEEE VLSI Test Symposium*. – 2009. – P. 47–52.
- Neale A. A new SEC-DED error correction code subclass for adjacent MBU tolerance in embedded memory / A. Neale, M. Sachdev // *Device and Materials Reliability, IEEE Transactions*. – 2013. – Vol. 13, No. 1. – P. 223–230.
- Unequal error protection codes derived from SEC-DED codes / P. Reviriego, S.S. Liu, A. Sánchez-Macián, L. Xiao, J.A. Maestro // *Electron. Lett.* – 2016. – No. 52(8). – P. 619–620. – URL: <https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/el.2016.0077>, свободный (дата обращения: 28.02.2024).
- Cha S. Efficient implementation of single error correction and double error detection code with check bit pre-computation for memories / S. Cha, H. Yoon // *JSTS: J. Semiconductor Technol. Sci.* – 2018. – No. 12(4). – P. 418–425.
- Hoyoon J. Protection of On-chip Memory Systems against Multiple Cell Upsets Using Double-adjacent Error Correction Codes / J. Hoyoon, L. Yongsurk // *Int. J. Computer Inform. Technol.* – 2014. – № 3 (6). – P. 1316–1320.
- A method for protecting telemetry in the processor memory from single failures / E. Lepeshkina, A. Shakhmatov, N. Kustov, V. Khanov // *AIP Conference Proceedings*. – 2021. – С. 50033.
- Лепёшкина Е.С. Применение кодов с исправлением двух ошибок для защиты конфигурационной памяти программируемой логики от действия космической радиации / Е.С. Лепёшкина, Н.Д. Кустов, В.Х. Ханов // *Russian Technological Journal*. – 2023. – Т. 11, № 5. – С. 54–62.
- Кудряшов Б.Д. Основы теории кодирования: учеб. Пособие. – СПб.: БХВ-Петербург, 2016. – 400 с.
- Питерсон У.У. Коды, исправляющие ошибки. – М.: Мир, 1964. – С. 264.

#### Лепёшкина Екатерина Сергеевна

Ст. преподаватель каф. безопасности информационных технологий (БИТ) Сибирского государственного университета науки и технологий им. М.Ф. Решетнева (СибГУ им. М.Ф. Решетнева)  
Им. газеты «Красноярский рабочий» пр-т, 31,  
г. Красноярск, Россия, 660037  
ORCID: 0000-0001-5116-6260  
Тел.: +7-923-313-57-13  
Эл. почта: klepkal111.93@mail.ru

Lepeshkina E.S.

#### Construction and analysis of check matrices correcting single and double adjacent errors codes

One of the consequences of the impact of space radiation on the on-board system equipment of spacecraft is a memory faults. Due to the decrease in the design standards of digital electronic component base, the number of multiple memory faults, in particular double ones, is increasing. To protect against double memory faults, the codes that correct single and double adjacent errors are used. The paper presents the mathematical basis for the use of codes of this type. A statement about the structure of the check matrix necessary to correct single and double adjacent errors is proposed and proven. A general algorithm for finding and correcting single and double adjacent errors in a memory code word is formulated. The statement about the minimum number of check bits for an information word of any length is proven. A general assess-

ment of the complexity of coding operations and error correction in a code word has been carried out depending on the number of check bits and the structures of check matrices. The results of the work can be used to design the devices detecting and correcting errors in the on-board memory of spacecraft.

**Keywords:** space electronics, multiple faults, error-control codes, single and double adjacent errors correction codes.

**DOI:** 10.21293/1818-0442-2024-27-2-37-43

### References

- Osipenko P.N. *Odinokhnye sboi – vyzov sovremennykh mikroprotessorov* [Single failures – a challenge for modern microprocessors]. *Electronic Components*, 2009, no. 7, pp. 12–15 (in Russ.).
- Maksimenco S.L., Melekhin V.F., Filippov A.S. *Analiz problemy postroeniya radiatsionno-stoikikh informatsionno-upravlyayushchikh sistem* [Analysis of problems in constructing radiation-resistant information and control systems]. *Information and Management Systems*, 2012, no. 2(57), pp. 18–25 (in Russ.).
- Chumakov A.I., Sogoyan A.V., Boruzdina A.B., Smolin A.A., Pechenkin A.A. [Mechanisms of Multiple Cell Upsets in Memory]. *Problems of Advanced Micro- and Nano-electronic Systems Development*, 2016, no. 4, pp. 145–152 (in Russ.).
- Hamming R.W. Error detecting and error correcting codes. *Bell System Technical Journal*, 1950, no. 29(2), pp. 147–160.
- Hsiao M.Y. A class of optimal minimum odd-weight column SEC-DED codes. *IBM Journal of Research and Development*, 1970, no. 14(4), pp. 395–401.
- Smulskiy A.V., Alekseev S.I., Kudryavcev U.E. *K voprosu obespecheniya ustoychivosti bortovoi apparatury perspektivnykh kosmicheskikh apparatov k mnozhestvennym sboyam ot deistviya ot del'nykh yadernykh chastits kosmicheskogo prostranstva* [On the issue of ensuring the stability of onboard equipment of promising spacecraft to multiple failures from the action of individual nuclear particles in outer space]. *Vestnik NPO im. S.A. Lavochkina*, 2014, no. 4 (25), pp. 97–102 (in Russ.).
- Georgakos G., Huber P., Ostermayr M., Amirante E., Ruckerbauer F. Investigation of increased multi-bit failure rate due to neutron induced SEU in advanced embedded SRAMS. 2007, *IEEE Symposium on VLSI Circuits*, 2007, pp. 80–81.
- Krasnyuk A.A., Petrov K.A. *Osobennosti primeneniya metodov pomekhoustoichivogo kodirovaniya v sub-100-nm mikroskhemakh pamyati dlya kosmicheskikh sistem* [Features of the application of noise-resistant coding methods in sub-100 nm memory chips for space systems]. *National Research Nuclear University MEPhI*, 2012, 4 p. (in Russ.)
- Dutta A., Touba N.A. Multiple Bit Upset Tolerant Memory Using a Selective Cycle Avoidance Based SEC-DED-DAEC Code. *25th IEEE VLSI Test Symposium (VTS'07)*. Available at: <https://ieeexplore.ieee.org/document/4209937>, free (Accessed: February 28, 2024).
- Datta R., Touba N.A. Exploiting unused spare columns to improve memory. *27th IEEE VLSI Test Symposium*, 2009, pp. 47–52.
- Neale A., Sachdev M. A new SEC-DED error correction code sub-class for adjacent MBU tolerance in embedded memory. *Device and Materials Reliability, IEEE Transactions*, 2013, vol. 13, no. 1, pp. 223–230.
- Reviriego P., Liu S.S., Sánchez-Macián A., Xiao L., Maestro J.A. Unequal error protection codes derived from SEC-DED codes. *Electronics Letters*, 2016, no. 52(8), pp. 619–620. Available at: <https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/el.2016.0077>, free (Accessed: February 28, 2024).
- Cha S., Yoon H. Efficient implementation of single error correction and double error detection code with check bit pre-computation for memories. *JSTS – Journal of Semiconductor Technology and Science*, 2018, no. 12 (4), pp. 418–425.
- Hoyoon J., Yongsurk L. Protection of On-chip Memory Systems against Multiple Cell Upsets Using Double-adjacent Error Correction. *International Journal of Computer and Information Technology*, 2014, no. 3 (6), pp. 1316–1320.
- Lepeshkina E., Shakhmatov A., Kustov N., Khanov V. A method for protecting telemetry in the processor memory from single failures. *AIP Conference Proceedings*, 2021, p. 50033.
- Lepeshkina E.S., Kustov N.D., Khanov V.Kh. *Primenenie kodov s ispravleniem dvukh oshibok dlya zashchity konfiguratsionnoi pamyati programmiruemoi logiki ot deistviya kosmicheskoi radiatsii* [Application of double-error correction codes to protect configuration programmable logic memory against space radiation]. *Russian Technological Journal*, 2023, vol. 11, no. 5, pp. 54–62 (in Russ.).
- Kudryashov B.D. *Osnovy teorii kodirovaniya* [Basics of Coding Theory]. SPb, BHV-Peterburg, 2016, 400 p. (in Russ.).
- Peterson U.U. *Kody, ispravlyayushchie oshibki* [Error correcting codes]. Moscow, Mir, 1964, 264 p. (in Russ.).

---

### Ekaterina S. Lepeshkina

Assistant Professor, Information Technology Security Department, Siberian State University of Science and Technology named after M.F. Reshetneva 31, «Krasnoyarsky Rabochy» pr., Krasnoyarsk, Russia, 660037  
 ORCID: 0000-0001-5116-6260  
 Phone: +7-923-313-57-13  
 Email: klepka1111.93@mail.ru